# Master's Thesis : Hyperparameters and features selection for forecasting energy generation and consumption

**Auteur :** Lempereur, Audrey
**Promoteur(s) :** Cornélusse, Bertrand
**Faculté :** Faculté des Sciences appliquées
**Diplôme :** Master en ingénieur civil en informatique, à finalité spécialisée en "intelligent systems"
**Année académique :** 2019-2020
**URI/URL :** http://hdl.handle.net/2268.2/9061

University of Liège - Faculty of Applied Science


**Master Thesis**

# Hyperparameters and Features Selection for Forecasting Energy Generation and Consumption

Graduation Studies conducted for obtaining the Master's degree in Computer sciences by Lempereur Audrey

**Audrey Lempereur**

**Supervised by Bertrand Cornélusse**

Academic year 2019-2020

# Abstract

Microgrids are electrical power systems made of consumption, generation, and storage devices. Their emergence raises some questions concerning the generation of reliable energy forecasts. Indeed, each microgrid has its characteristics, it can integrate different energy resources and be subjected to different loads. Moreover, the production of these energy resources and the load can be subjected to many variations over time (due to the weather, the inherent characteristics of the resource, socio-economic factors). Therefore, the automatic generation of energy forecasts is a challenging task. This thesis develops a methodology for selecting the hyperparameters and features of existing forecasting models and for using those optimized models for autonomously generating forecasts.

To this end, several forecasting models were optimized. These models were statistical time series models such as the naive forecaster or the exponential smoothing technique and some artificial intelligence-based models such as the linear regression, the gradient boosting, or the multilayer perceptron. Since the performances of different models can vary over time, the usage of a multi-model forecasting system is then analyzed. Multi-model forecasting enables to periodically calibrate the weights assigned to the different models and thus improves the final forecast. The multi-model methods tested are based on the linear combination of the different models. Firstly, some simple multi-models producing the mean or the median of the forecasts. Then, some methods based on inverse MSE weighting and linear regression weighting have been tested.

The individual models that produced the best results are the gradient boosting regression and the multilayer perceptron. The multi-model forecasting method that led to the best results is the multi-model selecting the median of the forecasts. It is also the method with the lowest costs in terms of computational time.

# Acknowledgement

I would like to express my gratitude to all those who contributed to the completion of this thesis. First, I would like to thank my supervisor Bernard Cornélusse. I would also like to thank Jonathan Dumas for his continuous support, the numerous meetings, and his rereading of this report. He was always available for answering my questions and to steer me in the right direction. Then, I would also like to acknowledge my brother Nicolas Lempereur as the second reader of this thesis. Finally, I would like to express my gratitude to my parents for providing me continuous encouragement throughout these years of study and the process of writing this thesis.

# Contents

# List of Figures

# List of Tables

# Introduction

## Context

In the sector of energy, efficient use of the resources relies on a good knowledge of the demand and of the supply. Forecasts of the production and of the consumption of electricity can be used to make decisions in terms of operational planning. At the scale of a microgrid, the ability to forecast future energy consumption and production for operating the system in an efficient way is also a challenge.

Microgrids are localized groups of interconnected loads and energy resources that normally operate connected to and synchronous with the traditional centralized grid, but can disconnect and operate autonomously[11]. Typically, solar energy can be one of the resources that operate within a microgrid. It is an intermittent and unstable production that can depend on many factors such as the solar irradiation, the shading, or the orientation of the panels. The load consumption can also be affected by various factors such as the weather, human behaviors, or economic factors.

Thanks to the development of smart meters, it is possible to store the historical values of energy production and load consumption of a microgrid. This historical information can be used jointly to weather forecasts to produce forecasts of consumption and production. In this work, such tools will be used to measure the load consumption of a microgrid and the production of electricity through photovoltaic panels.

Traditional forecasting techniques are based on time series or regression analysis. Time-series models widely used include the exponential smoothing techniques and the autoregressive integrated moving average (ARIMA). Other traditional forecasting techniques include artificial intelligence-based methods such as regression trees or neural networks. Forecasting can be done thanks to single output models that produce one forecast per training samples, but it can also be done with multi-output models that produce multiple forecasts (e.g. all forecasts for 24 hours ahead).

Most artificial intelligence-based methods need to be finely tuned in order to optimize the performances of the model created. The task of automatically selecting these hyperparameters is challenging. The selection of the input features for these models is also an important choice that needs to be studied. Indeed, the input features could be composed of weather information such as the temperature, the solar irradiation, the wind speed. It could also include some seasonality variables or even lagged observations of the quantities to forecast.

As the distribution of the load and of the Photovoltaic production (PV) can change, the performances of a model can degrade as time passes, which is called model drift. Thus, periodical retraining of the models should be done to ensure that they adapt to the new observations. The choice of the retraining frequency should be based on a trade-off between computational cost and predictive performances.

The selection of the number of training samples is also an important parameter. Indeed, the performances of a model trained on a too large training set can decrease because of the change of data distribution over time. Moreover, the number of training samples required could vary depending on the complexity of the model used, the amount of noise in the data, and so on. Therefore, the training set size is an important parameter that needs to be selected carefully.

Once all these parameters have been determined several strategies can be used for generating forecasts. The first strategy consists of the selection of one model for producing the forecasts. Typically, the forecasting models are evaluated on a dataset and the model with the best performances is selected. A first problem related to this solution is the fact that the distribution of the data may change over time. Thus, the best model used may not be the best one in the future. Moreover, in some situations, several models can have similar performances and it may not be possible to identify a single "best" model. These drawbacks can be overcome with multi-model forecasting which is the second strategy that can be used in this context. With this method, the strength of several models can be combined together. Typically, multi-model forecasts can be obtained by using a linear combination of the forecasts from the individual models. In 1969, Granger and Bates[3] studied a model that constructs a weighted average of the forecasts that minimized the mean square error of the combined forecast. In more recent studies, it was shown that simple equal-weighted combinations might in practice perform better than more sophisticated methods[5].

# Overview

In this thesis, a methodology for building an automated forecasting system is introduced. This methodology is tested thanks to the usage of existing time series of the PV production and load consumption of a real microgrid. The construction of this system involves several steps. Firstly, some forecasting models need to be built and tuned and the selection of the input features for training the models needs to be done. The models used are the seasonal naive forecaster (using the rule "today equals tomorrow"), the exponential smoothing technique, the linear regression, the gradient boosting method and the multilayer perceptron. After that, the amount of data necessary for training each model is selected and the impact of the retraining frequency on the performances of the models is analyzed. Once the models' parameter and hyperparameters are known, they can be used to produce a set of forecasts that are stored in a dataset. Finally, the forecasts of the individual models are analyzed and used to produce multi-model ensemble forecasts. Some simple combination methods such as the mean or the median forecast are first tested. Then, adaptative methods such as inverse MSE weighting or the weighted linear regression are used.

# Results

**Input features selection** For forecasting the load, the best performances were observed when the input features were simultaneously composed of weather variables (solar irradiation, temperature, relative humidity, wind speed, the quantity of precipitations) and variables for capturing the seasonality (the day of the week, the month of the year and the hour of the day). For the forecasting of the PV, the usage of weather variables specific to the PV (solar irradiation, temperature, and solar irradiation squared) has led to the best performances for most of the models. The introduction of a variable for capturing the hour of the day improved the performances for some other models.

**Analysis of the training set size** For forecasting the load, training the models on 20 to 32 days of data in the past was optimal. For forecasting the PV production, the multi-output models required shorter training periods (between 6 to 13 days of data) while the single output models required longer training periods (between 21 and 26 days of data).

**Analysis of the retraining frequency** For forecasting the load, weekly retraining of the models provides a maximum degradation of the performances[1] of 5%. For forecasting the PV production, the single output models could be retrained weekly and the multi-output models required a higher retraining frequency (retraining every 2 or 6 days).

Before the combination of the individual models together, the models based on gradient boosting and the multilayer perceptron are the models that provided the best performances both for forecasting the load and the PV.

**Multi-ensemble forecasting** For the forecasting of the load, the multi-model ensemble forecasting technique that produced the best performances in terms of mean absolute error is the multi-model forecast constructed by selecting the median of the forecasts from different models. In terms of mean squared error, the usage of a linear regression trained on the forecasts of the individual models produced the best performances. For forecasting the PV production, the multi-model that provide the best performances in term of mean absolute or mean squared error was the median. The usage of this method also provided better performances than all individual model used alone.

# Organization

Chapter 1 discusses the problem formulation from an abstract and then from a concrete point of view. The chapter 2 describes the load and PV time series of the system studied. The chapter 3 provides the theoretical background and the chapter 4 explains the methodology followed in this thesis in order to find a solution to the

[1] in term of NRMSE (Eq. 3.3)

problem formulated. The chapter 5 summarizes the optimization of the models. In chapter 6, a dataset is constructed with the forecasts of the different models and the performances of the different models are analyzed. Finally, in chapter 7, the multi-model ensemble forecasting techniques are tested and compared in several ways.

# Chapter 1

# Problem specifications

## 1.1 Abstract Problem

The goal of this thesis is to develop and validate an innovative methodology to select the hyperparameters and features of existing forecasting models for forecasting energy generation and consumption of microgrids. The input features used are based on weather variables and observations of the load and PV time series.

Suppose $\mathbf{x_T} = \{x_1, x_2, ..., x_T\}$ is a time series of the observed load and $\mathbf{y_T} = \{y_1, y_2, ..., y_T\}$ is a time series of the observed PV production of a microgrid. The goal of the forecasting problem is to produce forecasts of $x_{T+h}$ and $y_{T+h}$ made at time $T$ for $h$ steps ahead. Let's consider that the available weather forecasts at time $T$ are given by $\mathbf{w_{T+h'}} = \{w_1, ..., w_{T+h'}\}$ with $h' > h$. Then, the forecasting problem can be expressed in a generic way as finding the function $f$ and $g$ such that:

$$\hat{x}_T(h) = f(\mathbf{x}_T, \mathbf{w_{T+H'}}) \tag{1.1}$$

$$\hat{y}_T(h) = g(\mathbf{x}_T, \mathbf{w_{T+H'}}) \tag{1.2}$$

that minimizes the loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$.

$$\mathcal{L}_1(X_{T+h} - \hat{x}_T(h)) \tag{1.3}$$

$$\mathcal{L}_2(Y_{T+h} - \hat{y}_T(h)) \tag{1.4}$$

where $X_{T+h}$ and $Y_{T+h}$ are the real observations.

The forecast $\hat{x}_N(h)$ can be either a deterministic value or a random variable with an explicit density distribution.

The forecasting horizon is the number of steps ahead for which forecasts need to be made. Depending on the forecast horizon, forecasts can be characterized as short-term (from a few minutes up to a few days ahead), middle-term (from a few days to a few months ahead), or even long-term (from a few months to a few years). In this thesis, a short-term horizon of one day is selected. One day ahead forecasts can be used for operational planning.

The forecasting resolution is the distance in time between two forecasts. In this problem, the forecasting resolution is selected to be 15 minutes because it corresponds

to the Belgium market period.

The forecasting frequency represents the frequency at which the forecasts are produced. Since the forecasts are produced for one day ahead, the forecasting should be at least done daily to produce forecasts all the time. In this work, the forecasting was chosen to be performed once per day as it reduces the computational load.

## 1.2 Concrete Problem

Concretely, this thesis focuses on the development of a methodology for building an automated forecasting system. This system is represented in figure 1.1.

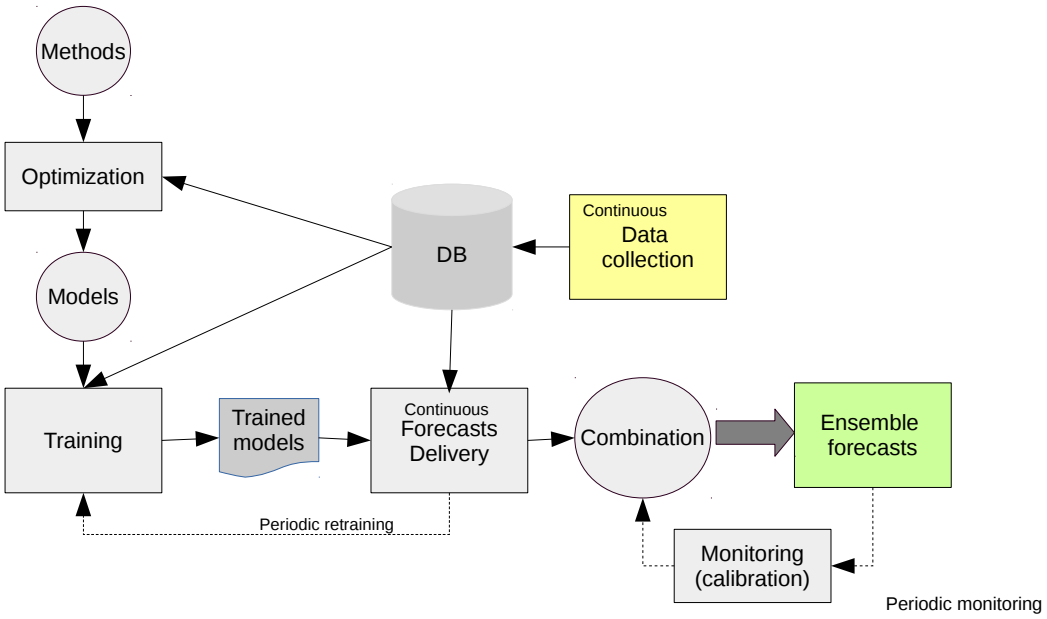



Figure 1.1: Forecasting system

This system is based on a continuous collection of data. The observations of the load and PV production as well as the weather information would be continuously collected.

The first stage of this system involves the selection of forecasting methods and their usage for producing optimized models. Firstly, for each method, the good combination of input variables must be selected and the hyperparameters of the forecasting models need to be optimized.

The number of training samples for training the forecasting models needs to be determined as well. Indeed, the training set size can have an impact on the performances of the models and it must be selected as a trade-off. If the training set size is too small, the training is performed on a small number of samples and it may not well generalize. By contrast, if the training set size is too large, it increases the chance that the data-generating process has changed over the period covered by the

window. Thus, the oldest data would no longer be representative of the system's current behavior. Depending on the complexity of the model used, the size of the training set that leads to the best performance of the model can be different. An additional question concerns the choice of a training set of fixed length (i.e. training done with a sliding window) or a training set size of increasing length (i.e. training done with an expanding window). In this thesis, the training set will be a sliding window of fixed size.

As time passes, the data distributions may deviate from those of the original training set. Thus, the performances of the models could degrade, which is called model drift. If a model's predictive performance has fallen due to changes in the environment, the solution is to retrain the model on a new training set which reflects the current reality[14]. It can be done thanks to two strategies. The first strategy consists in periodically retraining the models. In this case, the selection of the retraining frequency is an important parameter. If the retraining period is too short, the model will be updated unnecessarily which will increase the costs in terms of computation. If the retraining period is too long, the model would be suboptimal. The second strategy is to automate the retraining of the model when a model drift is observed. With this strategy, a threshold of divergence must be selected and will be used to trigger model retraining. The threshold is a very sensitive parameter and must be selected very carefully. In this thesis, the strategy used is periodic retraining to decrease the overhead linked to the management of model drift.

Once all the parameters and hyperparameters have been determined, the individual models can be used to automatically generate forecasts. Two main strategies can then be used. The first strategy consists in selecting a single model for producing the forecasts. The second strategy consists in combining the individual models to produce forecasts. Thanks to this method, the strengths of the different models can be exploited together. In this thesis, multi-model forecasting is used and a period recalibration of the model is used to produce the multi-model forecasts.

Finally, several constraints must be taken into account during the construction of the system. It should notably be able to run on a personal computer and the targeted computational time should be reasonable.

# Chapter 2

# Data

## 2.1 Introduction

This chapter introduces the load and the PV time series. It starts by a description of the two time series and then discusses the different factors influencing the load consumption and the PV production. The time series are studied during the period from the 2nd of September 2019 to the 31st of December 2019 with a 15-min resolution. The forecasting models will produce forecasts during the period from the 14th of October 2019 to the 31st of December 2019.

The system studied is a three-phase system. Initially, the measured observation (of the load or of the PV) $\tilde{x}_i^j$ on phase $j$ at time step $i$ is an instantaneous value available with a resolution of 10 ms. The values $x_t^j$ with a resolution of 15 mins is obtained by averaging the measured observations $[\tilde{x}_{N\times t}, \tilde{x}_{N\times t+1}, ..., \tilde{x}_{N\times t+N-1}]$. The data points with a 15 min resolution are thus obtained by doing:

$$x_t^j = \frac{1}{N} \sum_{i=0}^{N-1} \tilde{x}_{N\times t+i} \tag{2.1}$$

where $N = \frac{15*60*1000}{10} \frac{[\text{ms}]}{[\text{ms}]}$.

Once this values have been obtained, the missing data points have been replaced by linear interpolation and negative points were set to 0.

The load (resp. PV) described below are actually the sum of the load (resp. PV) on each phase. If $x_i^j$ represents the load (resp. PV) at time step $i$ on phase $j$, then the total load (resp. PV) is given by:

$$x_t = x_t^1 + x_t^2 + x_t^3$$

## 2.2 Load Consumption

Over the period from the 2nd of September 2019 to the 31st of December 2019, the mean of the load is 778.4 W and its standard deviation is 1226 W[1]. It is interesting to

[1]The mean and standard deviations during the period from the 14th Oct. to the 31st of Dec. are given by 971.7 W and 1412.5 W

note that the standard deviation is quite high compared to its mean. In figure 2.1, the daily mean and standard deviation of the load can be seen during the whole studied period. We can observe that the mean and standard deviation change significantly as time passes and that they tend to increase over time. The fact that the load increases is expected given that the studied period starts in late summer and ends in early winter. Thus, the sunshine duration and the temperature decreases which might lead to increases in the heating, lighting,...

From this figure, it can also be observed that there is a weekly seasonal pattern with high mean and standard deviation of the load during the weekdays and lower load during the weekends. In figure 2.2, box plots of the load observations by day of the week are shown. It can again be observed that there are some variations of consumption based on the day of the week. For example, on Wednesdays, 75% of the data points are lower than 460 W while on Tuesdays, 75% of the datapoints are lower than 2525W. During weekends and on Wednesdays, the load seems to be significantly lower than the rest of the week.

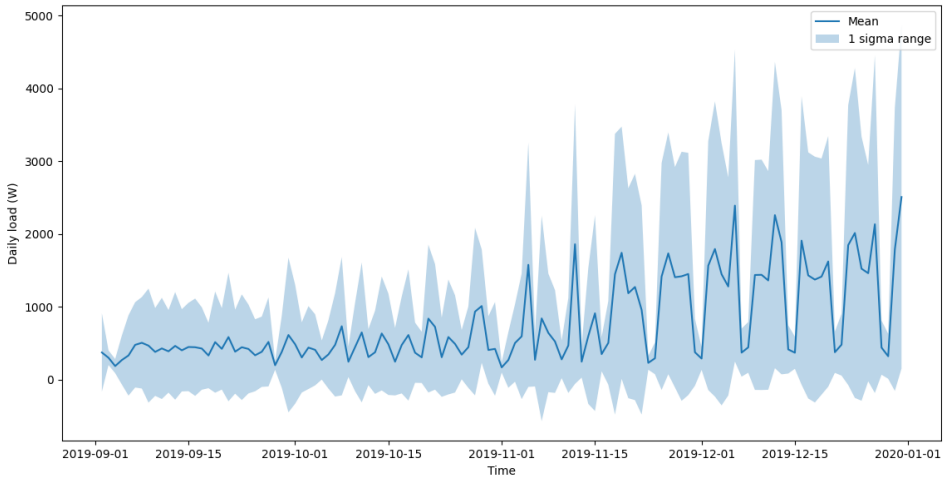



Figure 2.1: Daily mean and standard deviation of the load (W)

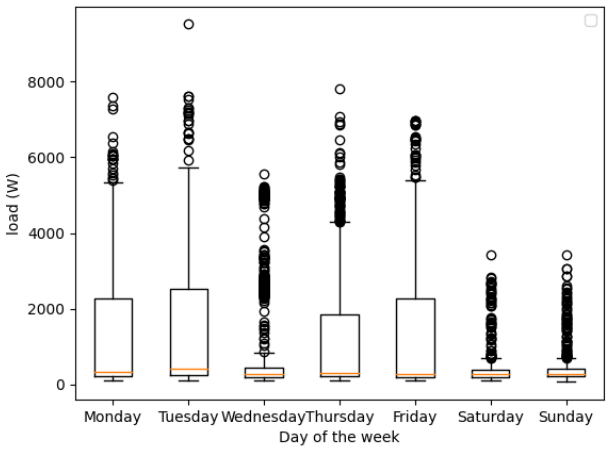



Figure 2.2: Boxplot of the load by day of the week

### 2.2.1 Analysis of the daily data

Figure 2.3 represents the daily distribution of the data under the form of box plots. Thanks to this figure, it can be observed that the median of the observations at a given time is always situated between 100 and 500 W independently from the time of the day. Between 10 PM and 7 AM, the box plots are small compared to the box plots of the rest of the day. It suggests that there are high-level chances that the load will be in this range during this period of the day. During the rest of the day (from 7 AM to 10 PM), the range of values that can take the load can reach huge values up to 3300 W for the 75 percentile and 7550 W for the maximum whisker. This uneven distribution shows that the load in the most positive quartile group can vary considerably.

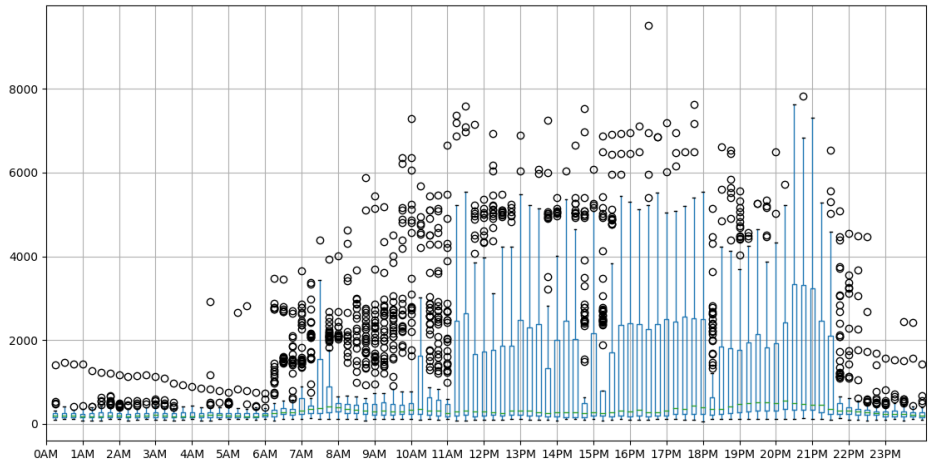
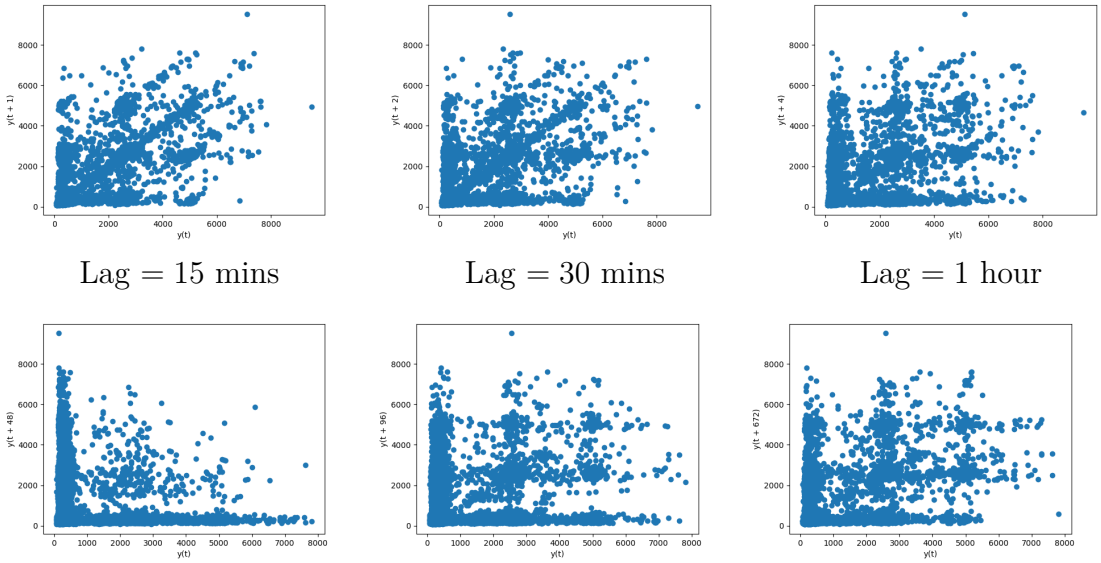



Figure 2.3: Boxplot of the daily load (W)

### 2.2.2 Correlation of lagged values

In figure 2.4, scatter plots represent the correlation between the load at time $t$ on the x-axis and the lagged loads on the y-axis. On the plot with a lag which is equal to 15 mins, the representation of a single line of best fit is complicated. It is an indication of the fact that the correlation between an observation at time $t$ and at time $t + 1$ is low. The figures obtained when using a lag of 15 or 30 mins are relatively similar. When the lag size increases, the correlation between the loads starts to have an L shape (especially for a lag of 12 hours[2]).

[2]Indeed, during the night the load is very close to 0. Thus, 12 hours later during the day, the load can take a wide range of values.




Lag = 15 mins | Lag = 30 mins | Lag = 1 hour

Lag = 12 hours | Lag = 1 day | Lag = 7 days

Figure 2.4: Scatter lagged plots for different lags for forecasting the load

## 2.3 Photovoltaic production

The mean PV production over the whole period is given by 252.7 W and its standard deviation is given by 493 W[3]. In figure 2.5, the mean and standard deviation of the daily PV production are provided. At the beginning of the studied period, the standard deviation is important compared to the mean daily production. However, as time moves on, the mean and especially the standard deviation of the PV production decrease. Since the PV production depends on the solar irradiance, this is an expected behavior as the daily solar irradiance decreases during most of the studied period[4].

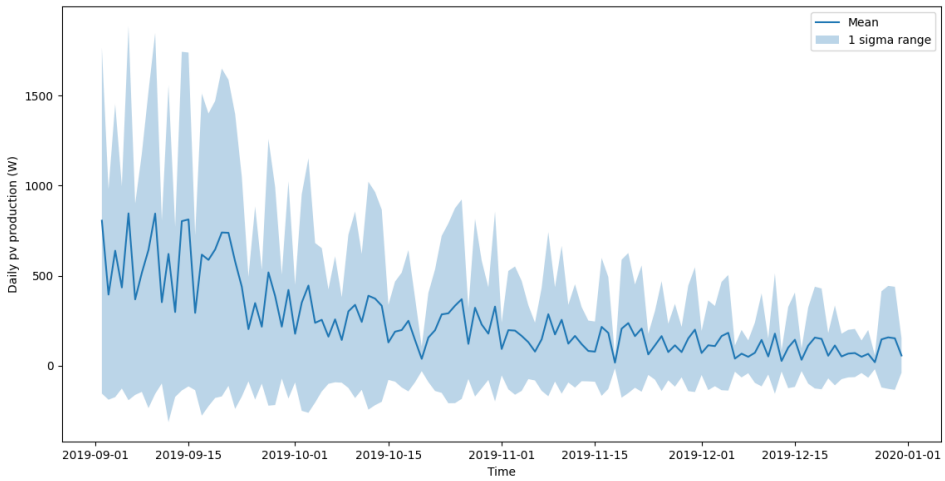



Figure 2.5: Daily mean and standard deviation of the PV production (W)

[3]The mean and standard deviations during the period from the 14th of Oct. 2019 to the 31st of Dec. 2019 are given by 146.4 W and 293.3 W

[4]From the 2nd of September to the 22nd of December (winter solstice)

### 2.3.1 Analysis of the daily data

In the figure 2.6, a box plot of the daily PV production is provided. From this data, it can be observed that the PV production is zero between 9 PM and 6 AM. The median of the PV production increases until 12:45 PM and then the median decreases. In the middle of the day, the data distribution is more spread and the higher observed PV production reaches 3330 W at 1:30 PM.

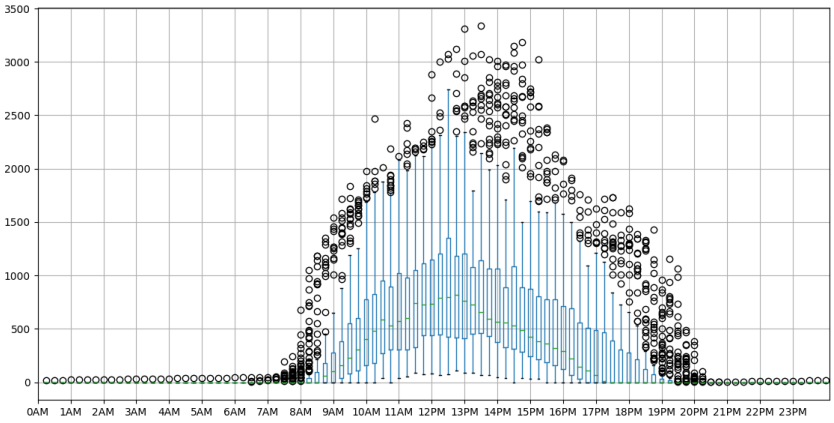



Figure 2.6: Boxplot of the daily PV production (W)

As can be seen in figure 2.7, the observations by day of the week are very similar to each other. The higher 75 percentile reaches 200 W so it means that at least 75% of the data points have values lower than 200W.

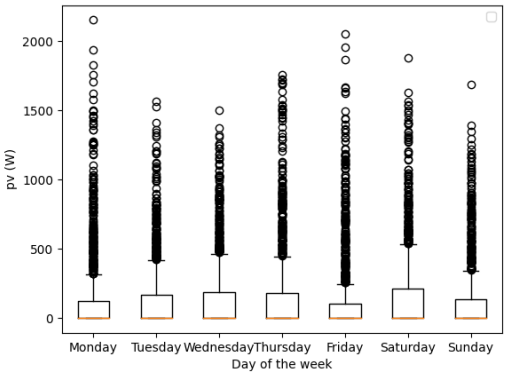



Figure 2.7: Boxplot of the PV production by day of the week

### 2.3.2 Scatter lagged plot

In the figure 2.8, several scatter lagged plots are used for representing the correlation between the PV production at time $t$ on the x-axis and the PV production at time $t + 1$ (e.g. 15 minutes later) on the y-axis. From this figure, it can be observed that the PV production at a given time, and the PV production 15 minutes later are two variables that are positively correlated. Indeed, as one variable increases, the

other one increases and, conversely, when one variable decreases, the other variable decreases as well. The same observations can be made when the lag size is 30 mins or 1 hour but the correlation decreases slightly since the points are more spread onto the plots. When considering the plot with a lag of 12 hours, it can be observed that the PV production following or preceding a 0 PV production by 12 hours can take a wide range of values. Indeed, the PV production should be 0 during the night and 12 hours it can change greatly.

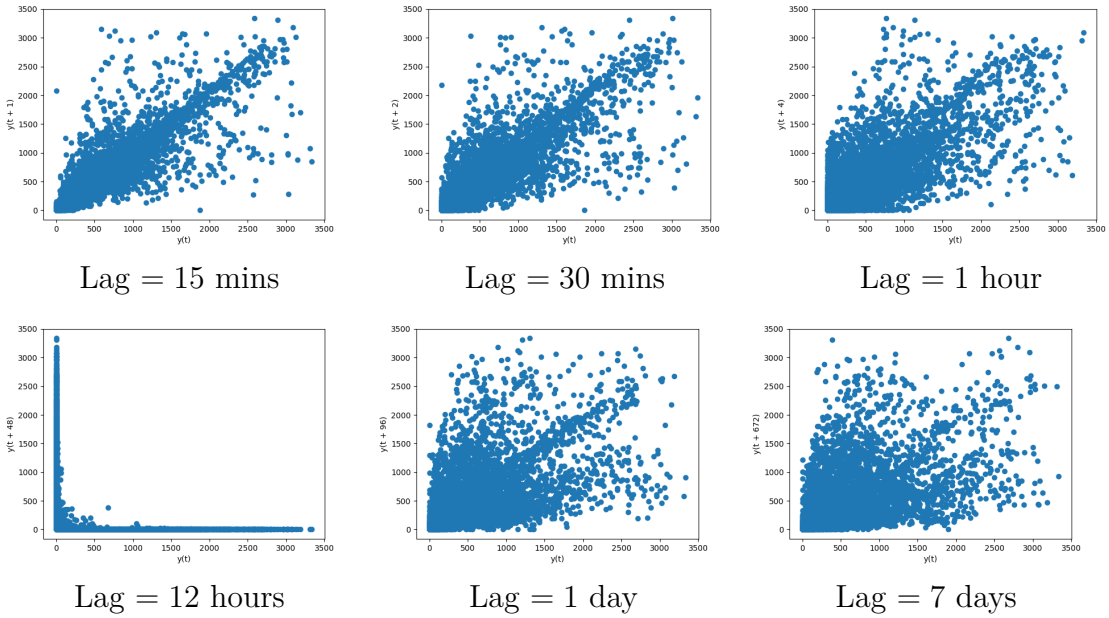



Figure 2.8: Scatter lagged plots for different lags

## 2.4 Factors influencing the load consumption

In this section, the main factors influencing the load consumption are described. These factors can be categorized as time factor, weather, economy and random disturbances[8].

**Time** The electric load of a microgrid is characterized by some seasonal pattern. It can depend on the season of the year, the day of the week or the hour of the day.

**Weather** The weather is also an important component that can have an impact on the load. Depending on the heating system used and on the presence of cooling systems, the *temperature* can be an important parameter. The *solar irradiation* can influence load consumption due to lighting systems. *Humidity* can affect the apparent temperature while it has no effect on the temperature and impact the behavior of consumers. The *precipitations* can also have an effect on the load directly and indirectly. It can have a direct effect on the behavior of individuals and have an indirect effect due to the decrease in temperatures linked with precipitations. At low humidity rates, the *speed of wind* lowers the apparent temperature and increases the rate of evaporation of perspiration from the human body therefore it gives the cooling effect[8]. Finally, the *cloudiness* can affect the temperature as well as the

light intensity at daytime.

**Socio-economic factors** Some socio-economic factors can impact load consumption. Such factors could include, the price of the electricity, the people's buying capabilities, the difference in pricing of the electricity during the day and during the night, and so on.

## 2.5 Factors influencing the PV production

In this section, the main factors influencing the PV production are described. These factors can be classified as environmental factors, PV system factors, installation factors, cost factors, and miscellaneous factors[9].

**Solar irradiation** The first environmental factor that affects PV production is solar irradiation. Indeed, the efficiency $\eta$ of a PV is given by:

$$\eta = \frac{P_{\max}}{G \times A}$$

where $P_{\max}$ is the maximum power point, $G$ is the input solar irradiance under standard test conditions and $A$ is the surface area of the panel.

**Temperature** A second factor that can affect the PV production is the temperature. Its effects can depend on the kind of panel used.

**Dust** Some of the sunlight can be blocked from the PV module due to the presence of dust which thus causes loss in the generated power since the solar irradiance is scattered on the surface of the solar panel.

**Shading.** The PV production can be reduced because of shading. The losses due to shading depend on the number of shaded cells, the type of panels used, the connections between the panels, and several other factors such as the size of the nearby buildings and the presence of trees or cross-shading from other panels.

**Additional factors** PV production can also be affected by the inherent characteristics of the PV panels, by some installation factors (e.g. angle of inclination, and orientation of the PV panels), or by some miscellaneous factors (e.g. degradations, glass breakage, presence of hot spots).

## 2.6 Conclusion

The load time series can be subjected to wide variations and is characterized by some seasonality patterns based on the hour of the day or the day of the week. By contrast, the PV time series takes a lower range of values and has some seasonality patterns based on the hour of the day. Both time series are affected by a variety of factors including the weather. In the next chapter, the methodology used in this thesis is summarized.

# Chapter 3

# Theoretical background

## 3.1 Forecasting techniques

For producing forecasts from a set of historical data and explanatory variables, several methods can be used. Traditionally, statistical time series forecasting methods such as the exponential smoothing technique or ARIMA are used for producing forecasts. More recently, artificial intelligence-based methods have been widely used as well. In this thesis, both types of models are used. The statistical time series methods used are the seasonal naive forecaster and the exponential smoothing technique. The artificial intelligence-based methods are the linear regression, the gradient boosting, and the multilayer perceptron. All those models are described in detail in appendix A.

## 3.2 Metrics

Several criteria can be considered when choosing a forecasting method. The first criteria that is typically considered is the forecasting accuracy.

### 3.2.1 Forecasting accuracy

The forecasting accuracy can be represented in different ways depending on the fact that the forecast is given as a point forecast or a probabilistic forecast. Let's suppose a model produces forecasts $\hat{x}_1, ..., \hat{x}_N$ during $N$ time steps. If the actual observations at those time steps are given by $X_1, ..., X_N$, then, several metrics exist to measure the error. Different choices in the metric could lead to different parameter estimates and could even result in a different model structure.

In this thesis, the main metrics used for point forecasts are the mean absolute error(MAE) and the root mean squared error(RMSE). These metrics are scale-dependent errors. Thus, the error is on the same scale as the data and they cannot be used to make comparisons between series that involve different scales.

**MAE** The mean absolute error is a measure that assumes that the loss function is linear. It is given by the average of the absolute error:

$$MAE = \sum_{i=1}^{N} \frac{|X_i - \hat{x}_i|}{N} \tag{3.1}$$

**RMSE** The root mean squared error is a measure that assumes that the loss function is quadratic. With this measure, large errors are more penalized. The RMSE is given by:

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{(X_i - \hat{x}_i)^2}{N}} \tag{3.2}$$

Since the distribution of the observed variables may vary over time, these metrics have also been normalized in some experiments by dividing by the mean observation. These normalized versions are thus given by:

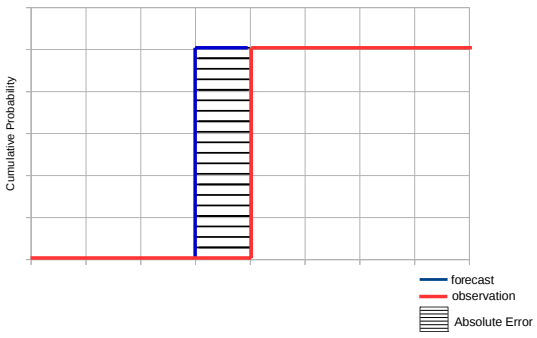
$$NMAE = \frac{1}{\overline{X}} MAE \tag{3.3}$$

$$NRMSE = \frac{1}{\overline{X}} RMSE \tag{3.4}$$

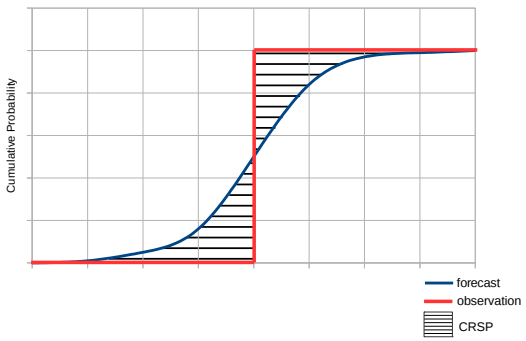
$$\text{with } \overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{3.5}$$

**CRPS** In some experiments described in the appendix, the gaussian process regressor has been tested. Since it produces probabilistic forecasts, different metrics must be used for measuring the accuracy. The Continuous Ranked Probability Score (CRPS) is a scoring rule that can be used in that context. The CRPS evaluates the area between the predictive cumulative distribution function (CDF) and that of the observation[1]. It can be directly compared to the MAE criterion used for point forecasts, since the CRPS is its generalization in a probabilistic forecasting framework[7].

$$CRPS(F, y) = \int_{-\infty}^{+\infty} (F(x) - 1\{y \leq x\})^2 \, dx = E_F|Y - y| - \frac{1}{2} E_F|Y - Y'|$$




Absolute error of a deterministic forecast




CRSP of a probabilistic forecast

[1]It is minimal when the true distribution of events is used as predictive density.

### 3.2.2 Time and complexity

The time an algorithm takes to train a model can be of great importance in the algorithm selection process. Indeed, an algorithm with a short fitting time could be chosen over an algorithm that has a longer fitting time but which produce more accurate forecasts. The training time depends on the algorithm selected as well as the number of training samples that are used to train the model.

The space complexity is also an important parameter for selecting a model as the models are trained on architecture with limited capabilities.

## 3.3 Model evaluation techniques

Three main methods exist to evaluate the performances of forecasting models. The first method is the simple splitting of the data in a training and a validation set. The second method is based on cross-validation and the third method is the Akaike's Information Criterion(AIC) ( see appendix A.3). The splitting of the data in a training and a validation set is simpler and faster to apply than the other methods but the estimate of the error on the validation set can be biased. The cross-validation is more complex and requires to retrain the different models several times on different training sets which means that the time needed to perform the evaluation of the model can be longer. However, the bias of the estimate of the error can be lower. The AIC is useful for selecting between models in the same class[2] but it is not useful for selecting models from different classes. For these reasons, the performances are mainly evaluated based on cross-validation in this thesis.

In figure 3.1, a typical example of cross-validation for time series is provided. In this example, the training samples are represented in blue and the validation samples are represented in red. The training samples are used for training the model and the validation samples are used for evaluating the performances of the model on new data. In figure 3.1, the training set is always prior to the validation set. Indeed, with time series, the data used for training the model should not be posterior to the data used for evaluation. Since it is not possible to obtain a reliable forecast based on a small training set, the earliest observations are not considered as validation sets. The forecast accuracy can be computed by averaging over the validation sets.

This procedure can be generalized to allow multi-step ahead errors to be used. The validation set should ideally be at least as large as the maximum forecast horizon required. Another variation of the cross-validation procedure developed in figure 3.1 is a procedure in which the training is of fixed size. In this situation, the cross-validation is a sliding window procedure.

[2]For example, it can be used to select an ARIMA model between candidate ARIMA models or an ETS model between candidate ETS models.

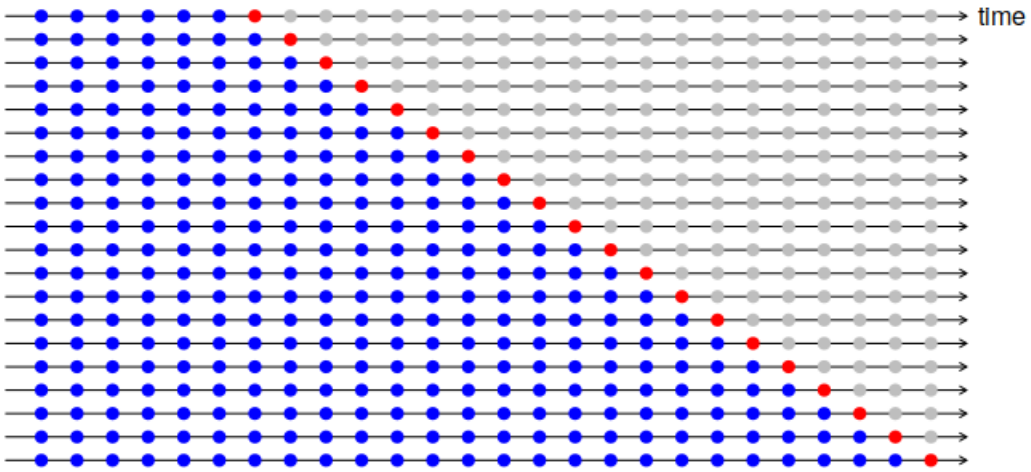



Figure 3.1: Time-series cross validation[10]

This procedure can be generalized to allow multi-step ahead errors to be used.

## 3.4 Multi-model forecasting

The usage of metrics for comparing forecasting models is a great tool for assessing the performances of different models. However, selecting the model with the best performances will not necessarily provide the best results. Some studies showed that the combination of forecasts from multiple models outperforms individual models in a wide range of forecasting problems[1][4].

Technically, the combined forecast $g(f^{(1)}, f^{(2)})$ dominates individual forecasts $f_1$ and $f_2$ if

$$E[\mathcal{L}(f^{(i)}, y)] > \min_{f(.)} E[\mathcal{L}(g(f^{(1)}, f^{(2)}), y] \text{ for } i = 1, 2 \tag{3.6}$$

where $y$ is the observation, $\mathcal{L}(.,.)$ is a loss function .

Let's consider the combination scheme for point forecasts. The objective is to find the optimal combination of forecasts that minimizes an error loss. Mathematically, if $f_i^{(1)}, ..., f_i^{(N)}$ are the forecasts from $N$ individual models for time step $i$ and if $y_i$ is the real observation at time step $i$, then the optimal combination is given by the function $g$:

$$g(f_i^{(1)}, ..., f_i^{(N)}) \tag{3.7}$$

that minimizes

$$E[\mathcal{L}(g(f_i^{(1)}, ...f_i^{(N)}), y_t)] \tag{3.8}$$

This function $g$ can be reduced to a simple linear combination of the individual forecast. In this case, the combined forecast for time step $i$ is given by:

$$g_i = w_0 + \sum_{j=1}^{N} w_i^{(j)} f_i^{(j)} \tag{3.9}$$

where $w_i^{(j)}$ are the combination weights.

**Simple combination** In some simple implementations, the forecasts combination can be done without observing the data underlying the forecasts. A first example of this methodology is average forecasting in which the multi-model forecast is given as the average of the individual forecasts. The combination weights are then given by:

$$w_0 = 0 \qquad\qquad w^{(j)} = \frac{1}{n} j \in [1, N] \tag{3.10}$$

Another simple method involves the usage of the median of the forecasts. Let's suppose the forecasts of the $N$ models for time step $i$ are ordered such that $f_i^{(1)} \leq f_i^{(2)} \leq ... \leq f_i^{(N-1)} \leq f_i^{(N)}$ , then, the median forecast is given by:

$$g_i^{median} = \frac{f_i^{(\lfloor \frac{N+1}{2} \rfloor)} + f_i^{(\lceil \frac{N+1}{2} \rceil)}}{2} \tag{3.11}$$

Similarly, the truncated mean can be used. If $f_i^{(1)}, ..., f_i^{(N)}$ is an ordered list, then the truncated mean can be obtained by trimming the top and the bottom $\lambda\%$.

$$g_i^{trunc} = \frac{1}{N(1-2\lambda)} \sum_{j=\lfloor \lambda N+1 \rfloor}^{\lceil (1-\lambda)N \rceil} f_i^{(j)} \tag{3.12}$$

**Estimated weight combination** In other methods, the data underlying the forecasts are used to compute the combination weights. Firstly, the inverse MSE weighting is a method in which the weight relative to a model is inversely proportional to its MSE computed on a past window of time. Mathematically, the weight $w_{T+1}^{(j)}$ computed at time step $T$ for time step $T+1$ for model $j$ is given by:

$$\hat{w}_{T+1}^{(j)} = \frac{1}{MSE_{T,w}^{(j)}} \times \frac{1}{\sum_{j=1}^{N} 1/MSE_{T,w}^{(j)}} \tag{3.13}$$

$$MSE_{T,w}^{(j)} = \frac{\sum_{i=1}^{w} (f_i^{(j)} - y_i)^2}{w} \tag{3.14}$$

The multimodel superensemble technique was introduced in 1999 by Krishnamurti et al.[2]. This method is based on the minimization of a mean squared error which can be denoted by $G$. If $g_i^{LR}$ represents the superensemble forecast at time $i$ and $y_i$ represents the observed values at time $i$, G is thus given by:

$$G = \sum_{t=T-w}^{T} (g_t^{SE} - y_t)^2 \tag{3.15}$$

where $w$ represents the size of the training period and $T$ represents the time of the last values from the training period.
If the number of model is given by $N$ and the model forecasts are denoted by $(f_i^{(1)}, ..., f_i^{(N)})$ at time $i$, then the superensemble forecasts is given by

$$g_i^{SE} = \bar{y} + \sum_{j=1}^{N} w_j (f_i^{(j)} - \bar{f}^{(j)}) \tag{3.16}$$

where $\bar{y}$ is the mean of the observations and $\bar{f}^{(j)}$ is the forecast mean for a model $j$ for a given training period.

# Chapter 4

# Methodology

## 4.1 Introduction

This chapter introduces the methodology followed to analyze an automated forecasting system for forecasting PV production or load consumption. The approach followed three main stages.

The first stage involves the collection of the data, the preparation of the data and the construction of the default models to be studied.

The second stage concerns the optimization of the models. Firstly, the input features that provide the best performances are selected for each model. Then, the hyperparameters of the models are tuned. The impact of the training set size is analyzed and the best training set size is selected for each model. Finally, a study of the retraining frequency is done for each model.

The last stage concerns the creation of a dataset of forecasts for a given period and the study of multi-model forecasting techniques that are periodically recalibrated.

## 4.2 Data collection

In this section, the collection of the observations of load consumption and PV production as well as the collection of the weather data is discussed.

**Load and PV time series** The load and PV data are obtained through requests to the feed API[12] provided by Emoncms.org[13]. Even if the values are available at a frequency of 10ms, the load and time series are collected to produce data points every 15 mins. These data points are not instantaneous values but rather the mean of the values over the period starting at the time provided. As mentioned in chapter 2, if $\tilde{x}_i^j$ represents the instantaneous load or PV at time step $i$ on phase $j$, the data points with a 15 min resolution are obtained through:

$$x_t^j = \frac{1}{N} \sum_{i=0}^{N-1} \tilde{x}_{N \times t+i} \tag{4.1}$$

The data is available from August 2019 up to 2020. An important part of the values during the first month is missing. As a consequence, this data was not taken into account for the study.

**Weather data** The weather data that were collected are the temperature at 2 meters from the ground, the solar irradiation on the ground, the wind speed at 2 meters from the ground, the relative humidity at 2 meters from the ground and the precipitation (caused by rain or snow). These values were available during the year 2019. This weather information are factors that can influence both the load and PV consumption as mentioned in chapter 2.

## 4.3 Data preparation

The data preparation follows three main steps: the data preprocessing step, the feature engineering step, and finally the feature scaling and selection step.

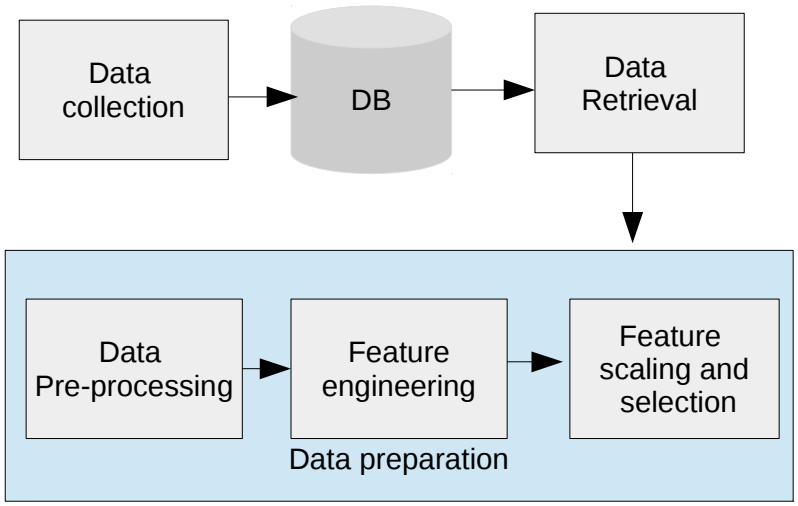



Figure 4.1: Data collection and preparation

### 4.3.1 Data preprocessing

This step focuses on the cleaning of the data. Firstly, the missing values in the load and PV time series have been replaced with linear interpolation. Secondly, the negative values have been replaced with 0. Finally, the total load and PV values have been produced by summing their values on the three phases. If $x_i^j$ represents the load (resp. PV) at time step $i$ on phase $j$, then the total load (resp. PV) is given by:

$$x_t = x_t^1 + x_t^2 + x_t^3$$

### 4.3.2 Feature engineering

The feature engineering is an important step as it enables to construct the inputs that can be used to produce forecasts. This step involves the creation of lagged

features as well as the creation of dummy variables for capturing seasonality. It is only necessary for artificial intelligence-based methods.

**Lag features** The generation of lag features consists of using the observation at given times, as input features for the prediction of the values at a future time. It can be viewed as a sliding window process. For instance, if the observations during 6 time steps are given by the values of the table 4.1, then the lag features with a window size of 3 are the one from the table 4.2. As can be seen, the usage of a window of size N implies that the N first raw of data needs to be discarded as there are some missing values.

| t | value |
|---|---|
| 0 | a |
| 1 | b |
| 2 | c |
| 3 | d |
| 4 | f |
| 5 | g |

Table 4.1: Initial table

| t | $t-3$ | $t-2$ | $t-1$ |
|---|---|---|---|
| 1 | NaN | NaN | NaN |
| 2 | NaN | NaN | a |
| 3 | NaN | b | c |
| 4 | b | c | d |
| 5 | c | d | f |

Table 4.2: Table with lagged features

In the experiments, the window was of 1 hour so at time $t$, the inputs features are based on observations at time $t - 15$ mins, $t - 30$ mins, $t - 45$ mins and $t - 60$ mins. Since the experiment was done by producing forecast for one day ahead, these features induced a bias and were thus used mainly for reference.

**Dummy variables for seasonality** The different seasonality patterns can be captured by the usage of dummy variables. Variables were created for representing the day of the week, the month of the year, and the hour of the day. In the case of the creation of dummy variables for forecasting the PV production, the only dummy variable used is the hour of the day. These dummy variables were constructed either by using variables that take integer values or binary values.

In the first case, the seasonality information is viewed as integer values: the week of the day is an integer value in $[0, 6]$, the month of the year is an integer value in $[1, 12]$ and the hour of the day is an integer in the range $[0, 23]$. An example is provided in table 4.3.

| Month | Day of week | Time of day |
|---|---|---|
| 9 | 3 | 6 |

Table 4.3: Representation of Thursday 12 September 2010 at 6:15 AM

Given that the variables can be seen as categorical variables, a binary encoding of the variables is possible. When using variables that take binary values, the creation of 7 binary variables is necessary for capturing the day of the week, 12 binary variables are necessary for representing the month of the year and 12 binary variables have

been used to represent the time of the day (the variables representing : 0-2 AM, 2-4 AM,...,10-12 PM). An example is provided in table 4.4.

| Month of the year | | | | | Day of the week | | | | | Time of the day | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| J | ... | S | ... | D | Mon | ... | Thu | ... | Sun | 0-2 | ... | 6-8 | .. | 22-24 |
| 0 | ... | 1 | ... | 0 | 0 | ... | 1 | ... | 0 | 0 | ... | 1 | ... | 0 |

Table 4.4: Representation of Thursday 12 September 2010 at 6:15 AM using binary values

This second way of producing dummy variables was only tested with the neural network architecture. Indeed, it induces a larger number of inputs for representing the same amount of information. It could thus lead techniques such as linear regression to overfit the data.

### 4.3.3 Feature scaling

Feature scaling has been used with artificial intelligence-based models. The scaling has been made thanks to the min-max normalization and standard scaling.
Formally, the min-max normalization $\mathbf{X}'$ of a feature vector $\mathbf{X}$ is given by:

$$\mathbf{X}' = \frac{\mathbf{X} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})} \tag{4.2}$$

Similarly, standard scaling enables to create a feature $\mathbf{X}'$ from $\mathbf{X}$ by computing:

$$\mathbf{X}' = \frac{\mathbf{X} - \mu}{\sigma} \tag{4.3}$$

where $\mu$ and $\sigma$ are the mean and standard deviation of $\mathbf{X}$.

The min-max normalization has been applied with the linear regression and the gradient boosting. The multilayer perceptron was tested with min-max and standard scaling for its inputs and its outputs and the scaling method that led to the best performances was selected.

### 4.3.4 Feature selection

With statistical time series approaches, the only features needed are the historical time series. When using artificial intelligence-based methods, the weather information, the lagged values, and the dummy variables for seasonality can also be used. The feature selection is a very important step in data preparation. Section 4.5.1 describes the procedure used for selecting the input features. In this section, the different combinations of input features that have been used in the experiments are highlighted.

#### Weather variables

The first input features used are the set of every weather variables. The information taken into account is thus the solar irradiation on the ground (swd), the temperature at 2 meters of the ground (tt2m), the relative humidity at 2 meters of the ground

(rh2m), the wind speed at 2 meters of the ground (ws2m), the precipitation of snow or rain (prec). These inputs were tested for both the forecasting of load consumption and PV production.

| Weather variables | | | | |
|---|---|---|---|---|
| swd | tt2m | rh2m | ws2m | prec |

### Weather and lagged observations

Then, the weather variables were tested in combination with the lags of the observed variables. If the time is $T$, the observations taken into account are the observations at time $T-60$ mins ($o_{T-4}$), $T-45$ mins ($o_{T-3}$), $T-30$ mins ($o_{T-2}$) and $T-15$ mins ($o_{T-1}$). These inputs were tested for both the forecasting of the load consumption and of the PV production.

| Weather variables | | | | | Lagged observations | | | |
|---|---|---|---|---|---|---|---|---|
| swd | tt2m | rh2m | ws2m | prec | $o_{T-4}$ | $o_{T-3}$ | $o_{T-2}$ | $o_{T-1}$ |

### Weather variables and dummy variables for seasonality

Another variation of the input features used is obtained by the combination of the weather information with the dummy variables for capturing seasonality. The seasonality variables taken into account for the forecasting of the load was the day of the week, the month of the year and the hour of the day.

| Weather variables | | | | | Seasonality variables | | |
|---|---|---|---|---|---|---|---|
| swd | tt2m | rh2m | ws2m | prec | Month of the year | Day of the week | Hour of the day |

In the case of the prediction of the PV production, only one dummy variable was used to capture the hour of the day (as can be seen in the table below).

| Weather variables | | | | | Seasonality variables |
|---|---|---|---|---|---|
| swd | tt2m | rh2m | ws2m | prec | Hour of the day |

### Subset of weather variables specific to PV

As the PV production depends mainly on the solar irradiation and on the temperature, a subset of the weather variables can be used for forecasting.

| Weather variables | | |
|---|---|---|
| swd | tt2m | $\text{swd}^2$ |

In the case of the multilayer perceptron, the seasonality variables were tested by representing them with binary or integer values.

**Subset of weather variables specific to PV with seasonality variables**

The subset of the weather variables was also used as input in combination with the seasonality variable.

| Weather variables | | | Seasonality variable |
|---|---|---|---|
| swd | tt2m | $swd^2$ | Hour of the day |

## 4.4 Forecasting model construction

This section describes the forecasting models used and the methodology used for training those models. The first method tested was the naive forecaster. It was used as a baseline for comparison of the different models. The exponential smoothing technique from the family of statistical approaches was studied.

Then, artificial intelligence-based methods were explored. The linear regression, the gradient boosting regressor, and the multilayer perceptron were selected to produce point forecasts. The Gaussian process regression was selected for generating probabilistic forecasts but was not used in the multi-model forecasting system.

**Single output mode** All the artificial intelligence-based models have been tested in a single output mode in which the input features are used for predicting a single dependent variable (i.e. the load/PV at a given time). If the training samples are given during $T$[1] time steps and if $\mathbf{i}_t$ is the input feature vector and $o_t$ is the load or PV, then the training examples are given by:

$$\{(\mathbf{i}_0, o_0), ..., (\mathbf{i}_{96}, o_{96}), ..., (\mathbf{i}_{T-96}, o_{T-96}), ..., (\mathbf{i}_T, o_T)\}$$

**Multi-output mode** The models were also tested in a multi-output mode in which the input features from one entire day were concatenated to form a single entry and which was used to forecast the load or PV production for this day (i.e. for forecasting 96 load or PV values). In this case, the training samples are thus given by:

$$\{([\mathbf{i}_0, ..., \mathbf{i}_{96}], [o_t, ..., o_{t+96}]), ..., ([\mathbf{i}_{T-96}, ..., \mathbf{i}_T], [o_{T-96}, ..., o_T])\}$$

Now, let's describe the model used with more details.

**Naive forecaster** The naive forecaster model is a seasonal naive model (see eq. A.2) in which the seasonal period is equal to 1 day (so 96 data points). The model is retrained once per day and it uses a training size of 1 day of data.

**Exponential Smoothing Technique** The exponential smoothing model was constructed thanks to the `ExponentialSmoothing` from `statsmodels`. This model is dependent on several parameters (the trend, the damping of the trend, the seasonality, and the length of the seasonality period).

[1] T is a multiple of 96 (so it represents an integer number of days)

**Linear regression** The linear regression model was created thanks to the `LinearRegression` from `sklearn`[6].

**Gradient boosting regression** The gradient boosting regression model was created thanks to the `GradientBoostingRegressor` model from `scikit-learn`. Its default values were a maximal depth of trees of 5, a minimum number of samples per split and per leaf of 5, 100 estimators, and a learning rate of 0.1. Then, all these parameters have been optimized.

The gradient boosting was tested by doing single output and multi-output which has been done by using a pipeline using the `MultioutputRegressor` from `scikit-learn`.

**MultiLayer Perceptron** The model was built thanks to `Keras`. The model was simply composed of one hidden layer with a number of neurons which was twice the number of input features. The default activation function of this layer was the ReLu. This layer was followed by a dropout layer with a default value of 0.4. Finally, the model was composed of a Dense layer with a default ReLu activation function. The optimization was done thanks to the Adam algorithm using a default learning rate of 0.005. The loss function of the model was the mean squared error.
A single output model was created with one neuron in the output layer and a multi-output model was created with 96 neurons in the output layer.

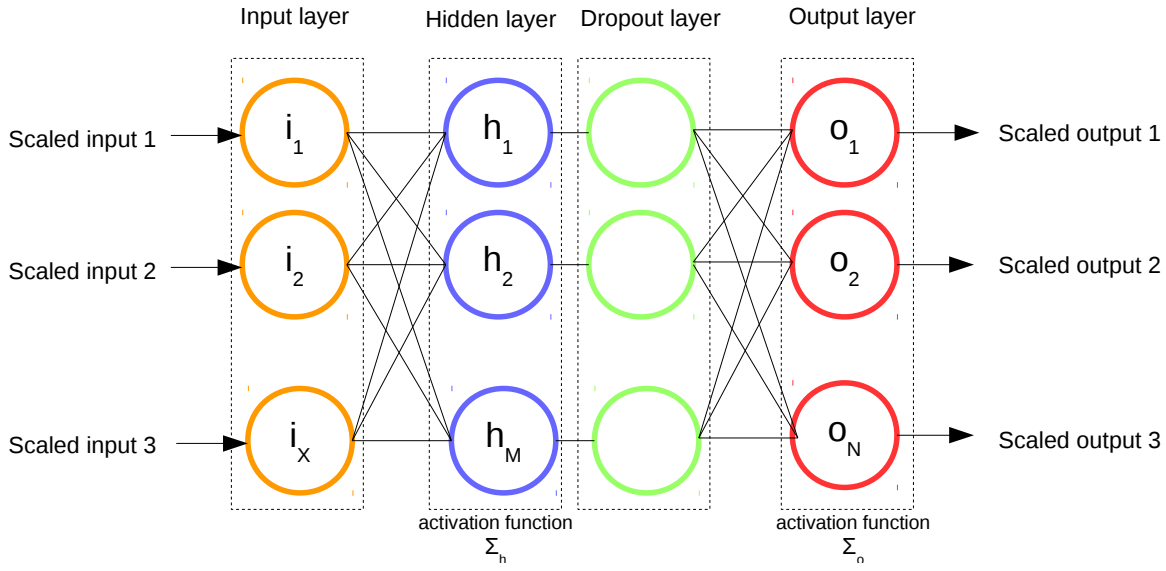



Figure 4.2: Neural network architecture

**Gaussian Process Regression** The model was created thanks to the `GaussianProcessRegressor` from `scikit-learn`. The default kernel used was the RBF kernel.

## 4.5 Optimization

The optimization of the individual models follows the steps represented in figure 4.3.

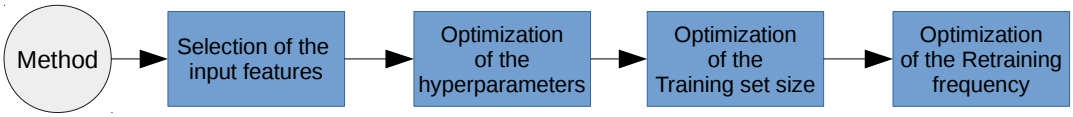



Figure 4.3: Optimization details

### 4.5.1 Input features selection

This optimization step concerns the selection of input features for each forecasting model. The input features tested are described in 4.3.4.

The performances of the forecasting models trained with different combinations of input features were evaluated thanks to a sliding window cross-validation procedure as can be seen in figure 4.4. The model is trained on a training set composed of 35 days of data and is evaluated on a validation set composed of 14 days of data. The training and validations sets are then slid by 14 days and the procedure is repeated. The training and evaluation steps are thus done five times until reaching the end of the studied period. The forecasting model producing point forecasts were evaluated based on their mean MAE, RMSE, and mean training time. The forecasting model producing probabilistic forecasts was evaluated based on its CRPS and mean training time.

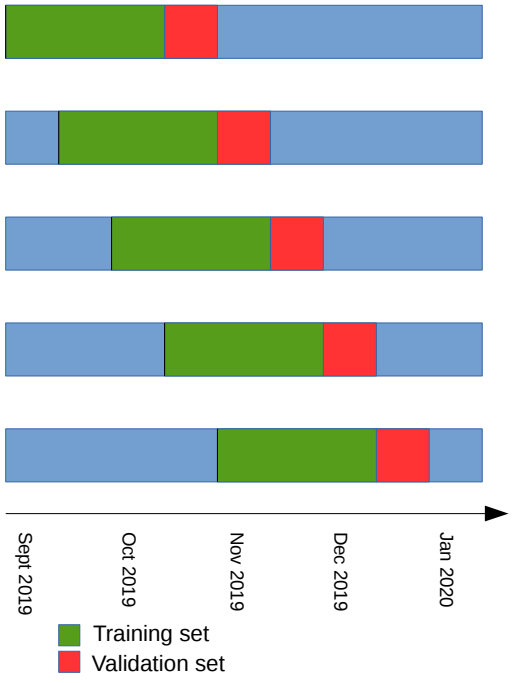



Figure 4.4: Sliding window usage for performance evaluation

Thanks to these metrics, the best input features are selected for each model. These input features are then fixed and are used for all the following experiments. The result are summarized in chapter 5 and are detailed in detail in appendix B.

### 4.5.2 Hyperparameter Optimization

Most forecasting methods are affected by a set of hyperparameters. Their selection can have an impact on the performances of the forecasting models generated. As a matter of fact, an important step in building the models implies hyperparameter optimization. The sliding window cross-validation strategy developed in section 4.5.1 was also used for assessing the performances of the models. The analysis of the results is summarized in chapter 5 and discussed in depth in appendix C. Once the models have been optimized, the hyperparameters were not modified for the rest of the experiments.

The models which were optimized are the exponential smoothing technique (EST), the gradient boosting regression(GBR), the multilayer perceptron(MLP), and the gaussian process regression (GPR).

**EST** The hyperparameters optimized are the seasonality and the trend patterns (which can be either absent, multiplicative or additive) and the fact that the trend is damped or not and the length of the seasonality.

**GBR** The model was optimized by tuning the maximal depth of trees, the number of estimators, the minimum number of samples per split and per leaf, and the learning rate.

**MLP** With the multilayer perceptron, the first parameter which was tuned is the choice of the scaling on the inputs and on the outputs (Min-max normalization or standard scaling). Then, the following components are tuned: (i) the number of neurons in the hidden layer, (ii) the variation of the learning rate and the number of epochs, (iii) the dropout rate, (iv) the batch size, (v) the number of layers and finally (vi) the activation functions.

**GPR** The Gaussian process regressor is tested with various kernels.

### 4.5.3 Training set size analysis

The learning set size can have an impact on the performances of a model. In the case of a naive forecaster, the learning set size is fixed to one day. However, for other methods, it can be tricky. A problematic concerns the analysis of the impact of the learning set size for each possible method.

In order to do this analysis, several training periods of various sizes can be used to train a given model as shown in figure 4.5. This model will be used to predict a validation period that follows the training period. Thus, the performances can be evaluated as a function of the training set size.

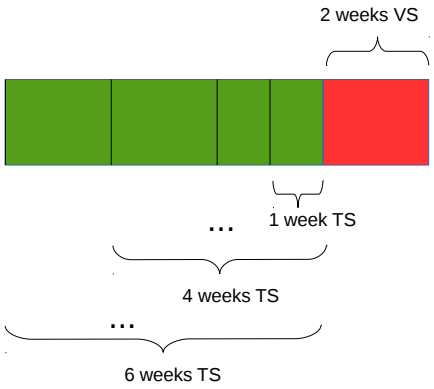



Figure 4.5: Methodology to study the impact of the training period

In order to decrease the possible bias on the performance measured on the validation set, this procedure can be reproduced several times by the mean of a sliding window as shown in figure 4.6.

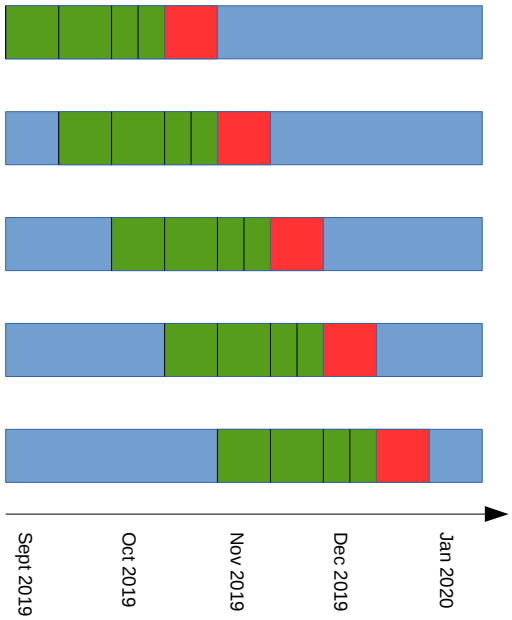



Figure 4.6: Methodology to study of the impact of the training period thanks to a sliding window method

The experiments have been done by varying the training set size from 1 day to 42 days of data. The metrics used for evaluating the performances of the point forecasts were the MAE and the RMSE. The results are summarized in chapter 5 and are fully described in appendix D.

### 4.5.4 Analysis of the retraining frequency

The frequency at which the models are trained is also of great importance. Indeed, the distribution of the data can vary over time and the performance of the models may thus degrade as time passes. To select a suitable retraining period for each forecasting model, an analysis of the retraining frequency is done.

The retraining frequency can be selected based on the performance of a forecasting model trained on different training set for a given validation set as shown on the picture 4.7.

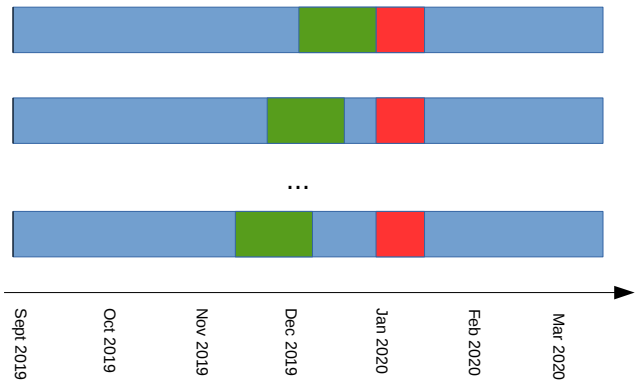



Figure 4.7: Retraining frequency analysis methodology

The validation set used was composed of 7 consecutive days randomly selected on the available dataset. To reduce the bias induced by the choice of the validation set, this procedure is repeated several times to forecast different randomly selected validation periods.

The metric used to evaluate the performances of the models on the validation set is the NRMSE. This metric was chosen because the different forecast periods can be different in terms of load. Thus the usage of the RMSE would lead to unwanted variations depending on the forecast period.

The retraining period is chosen as the longest period smaller or equal to one week such that the performances decrease at most by 5 %. Mathematically, with $\mathbf{x}$ being the vector of NRMSE, containing elements $x_i$ where $i$ represents the number of days between the training and the forecasting, it can be written as:

$$s = \arg \max_{x_i < x_0 \times 1.05} \mathbf{x} \tag{4.4}$$

In this way, a compromise between the computational load and the performances is done. The results are presented in detail in appendix E.

A pseudo-code of the methodology used to study the retraining frequency of the different models is provided (Algorithm 1).

**Algorithm 1** Retraining analysis algorithm

**procedure** RETRAININGFREQUENCYANALYSIS($f$, $M$, $N$)
    *// $f$ is the forecasting model*
    *// $M$ is the maximum number of days separating the TS from the VS*
    *// $N$ is the number of times the procedure is repeated*
    scores $\leftarrow$ array filled with 0 of dimensions $[M, N]$
    $P_t \leftarrow$ the best training period for model $f$
    **for** $i = 0$ **to** $M$ **do**
        $t \leftarrow$ random day in $(2019/10/14, 2019/12/24)$
        $(\mathbf{i_V}, \mathbf{o_V}) \leftarrow$ $generate\_inputs(t, t + 7)$
        **for** $j = 0$ **to** $N$ **do**
            $(\mathbf{i_T}, \mathbf{o_T}) \leftarrow generate\_inputs(t - j - P_t, t - j)$
            $f \leftarrow train\_model(\mathbf{i_T}, \mathbf{o_T})$
            $\mathbf{y} \leftarrow f(\mathbf{i_V})$
            $\text{scores}(i, j) \leftarrow NRMSE(\mathbf{o_V}, \mathbf{y})$
    $\mu(i) \leftarrow \frac{\sum_{j=0}^{N} \text{scores}(i,j)}{N}$
    $\sigma(i) \leftarrow \sqrt{\frac{\sum_{j=0}^{N} \text{scores}(i,j) - \mu(i)}{N}}$
**end procedure**

## 4.6 Dataset Construction

To be able to construct and use the monitoring system, a dataset of forecasts obtained with different models has been built.

Forecasts are created in the period from the 14th of October 2019 to the 31st of December 2019. The knowledge of the optimal training set size and of the optimal retraining frequency enables to produce forecasts for the entire period as detailed in algorithm 2.

**Algorithm 2** Dataset construction

Let $m_i$ be a forecasting model
$P_t \leftarrow$ the best training period for model $m_i$
$P_r \leftarrow$ the best retraining period for model $m_i$
Let $y$ be an array used to contain the forecasts
$t \leftarrow 2019/10/14$
**while** $t < 2020/1/1$ **do**
    $(\mathbf{i}, \mathbf{o}) \leftarrow generate\_inputs\_training(t - P_t, t)$
    $m_i \leftarrow train\_model(\mathbf{i}, \mathbf{o})$
    **while** $t < t + P_r$ **do**
        $\mathbf{i} \leftarrow$ $generate\_inputs\_forecasting(t, t + 1)$
        $\mathbf{y_{[t:t+1]}} \leftarrow m_i(\mathbf{i})$
        $t \leftarrow t + 1$

## 4.7 Analysis of a monitoring system

Once the dataset of forecasts has been created, the final step consisting in the creation of multi-model for managing the production of forecasts has been studied. In this last step, several multi-models ensemble forecasting techniques are tested.

Firstly, simple methods that require no training of the models are used. These methods are the model averaging, the multi-model that uses a truncated mean, and the multi-model that produces the median of the forecasts. Then, models that need to be trained for producing forecasts have been used. The first model tested is the model that simply selects the "best" model. The multi-model reevaluates the performances of the different models each day and the one that produced the best forecasts during a given period in the past.

Mathematically, if $f_i^{(1)}, ..., f_i^{(N)}$ are the forecasts from $N$ individual models for time step $i$ and if $y_i$ is the observation at time step $i$, the model will update its weights at time $T$ by doing:

$$\hat{w}_{T+1}^{(j)} = \begin{cases} 1 & \text{if } RMSE_{T,w}^{(j)} > RMSE_{T,w}^{(i)} \quad \forall i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

or:

$$\hat{w}_{T+1}^{(j)} = \begin{cases} 1 & \text{if } MAE_{T,w}^{(j)} > MAE_{T,w}^{(i)} \quad \forall i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

with

$$RMSE_{T,w}^{(j)} = \sqrt{\sum_{i=T-w}^{T} \frac{(f_i^{(j)} - y_i)^2}{w}} \tag{4.7}$$

$$MAE_{T,w}^{(j)} = \sum_{i=T-w}^{T} \frac{|f_i^{(j)} - y_i|}{w} \tag{4.8}$$

The window size $w$ indicates the number of days that are taken into account for computing the performance metric.

The second multi-model tested is the multi-model based on inverse MSE weighting. The equation 3.14 describes the update of its weights. Such as the previous model, the model based on inverse MSE weighting needs to evaluate the performances of the individual models on a past window of time. Thus, the impact of the size of the window is a parameter which needs to be studied.

After the inverse MSE weighting, the multi-model that does linear regression weighting has been used. With this strategy, the multi-model is updated daily by applying linear regression on the past forecasts of the individual models. Mathematically,

the multi-model performs:

$$g_i^{LR} = w_0 + \sum_{j=1}^{N} w_i^{(j)} f_i^{(j)} \tag{4.9}$$

$$\mathcal{L}(g_i^{LR}, y_i) = \sum_{i=T-W}^{T} (g_i^{LR} - y_i)^2 \tag{4.10}$$

where $F_i$ is the multi-model forecast at time step $i$ and $\mathcal{L}(F_i, y_i)$ is the loss function to minimize.

Once these classical models have been tested, some tests were done by taking into account extra information. In practice, it is common that some forecasting models work well during some periods and much less during other periods. Thus, the first multi-model tested is a model that takes into account the weekly seasonality of the load. When combining weights for forecasting a given day of the week (e.g. Monday), the combination weights are determined by using the individual forecasts from the same day of preceding weeks (preceding Mondays). This training method was tested with the multi-model that select models with best performances, with the inverse MSE weighting multi-model and with the linear regression weighting.

After that, the last method studied is a linear regression weighting that uses additional explanatory variables. For forecasting the load, the explanatory variables used are the hour of the day and the day of the week. For forecasting the PV, the explanatory variable used is the hour of the day. Mathematically, this methodology is given by:

$$g_i^{LR2} = \sum_{j=1}^{N} w_i^{(j)} f_i^{(j)} + \sum_{j=1}^{M} w_i^{(N+j)} \eta_i^j \tag{4.11}$$

$$\mathcal{L}(g_i^{LR2}, y_i) = \sum_{i=T-W}^{T} (g_i^{LR2} - y_i)^2 \tag{4.12}$$

where $F_i$ is the multi-model forecast at time step $i$, $\mathcal{L}(F_i, y_i)$ is the loss function to minimize, and $\eta_i^1, ...\eta_i^M$ are the values of the explanatory variables at time step $i$.

As mentioned before, these multi-models are updated every day and their performances are compared in terms of MAE and RMSE. Additionally to the usage of the MAE and of the RMSE, a specific normalization was also used in this experiment for evaluating the PV production. It is obtained by dividing the error metrics by the maximum observation of the PV production:

$$\frac{MAE}{\max X_i} \tag{4.13}$$

$$\frac{RMSE}{\max X_i} \tag{4.14}$$

To view the impact of the retraining frequency of the multi-models, tests were also performed by performing weekly updates of the multi-models rather than daily updates. In this situation, the combination weights of the different models are thus determined once per week. The multi-models have finally been compared based on their training time and on the time necessary for combining the different forecasts.

# Chapter 5

# Optimization

In this chapter, the results obtained by optimizing the parameters as explained in section 4.5 are summarized. Firstly, the input features of the forecasting models are optimized, then the hyperparameters of the forecasting models are optimized. The training set size is optimized and finally, the retraining period is selected.

## 5.1 Input features selection

The first optimization step concerns the selection of the input features to be used for training the artificial intelligence-based models. The different combinations of input features tested are described in section 4.3.4. The artificial intelligence-based models analyzed are the multiple linear regression(MLR), the gradient boosting regression(GBR), and the multilayer perceptron(MLP). These methods have been tested in a single output mode or a multi-output mode as described in section 4.4. In appendix B, the performances of the models trained with different combinations of input features are detailed.

**Load forecasting** Let's consider the impact of the selection of the input features for the forecasting of the load. For the single output MLP, the best performances were observed when the input features where composed of all the weather variables and the seasonality variables expressed as binary values. For all the other models[1], the best performances were obtained by using as input features the weather variables and the seasonality variables expressed as integer values. A summary of the best input features for all the models and their respective performances are summarized in table B.17. At this point, the best performances are obtained with the single output GBR and the worst performances are obtained with the multi-output MLR.

**PV forecasting** Now, let's consider the impact of the selection of the input features for the forecasting of PV production. With the MLR[2], the input features based on the weather variables relative to the PV (i.e. solar irradiation, temperature, and solar irradiation squared) provide the best performances. About the single output GBR, the usage of the combination of weather variables relative to the PV with the seasonality variable (i.e. the hour of the day) provides the best performances.

[1] i.e. single and multi-output MLR and GBR and multi-output MLP

[2] i.e. the single output MLR and the multi-output MLR

With the multi-output GBR, the simple usage of the weather variables specific to the PV provides the best performances. Finally, with the single output MLP, the best performances are reached when all the weather variables are used with the seasonality variable as input. For the multi-output model, the best performances are observed with the weather variables specific to the PV and the seasonality variables expressed as binary values. The performances of the models used with their best input features are summarized in table B.19.

## 5.2 Hyperparameter optimization

In this section, the forecasting models depending on several hyperparameters have been optimized. These models are the exponential smoothing technique, the gradient boosting regression, and the multilayer perceptron. The description of their default parameters is given in section 4.4 and the description of the hyperparameters being optimized is provided in section 4.5.2. The performances of the different models can be compared to the naive forecaster. When evaluating the naive forecaster on the validation sets for the forecasting of the load, the MAE is equal to 776 W and the RMSE is equal to 1458 W. For the forecasting of the PV production, the naive forecaster has a MAE of 104 W and a RMSE of 238 W.

**EST** For the forecasting of the load, the EST with no trend, a multiplicative seasonality, and a seasonal period of 7 days provides the best performances. With this configuration, the performances are better than the one of the naive forecaster. For the forecasting of the PV production, the exponential smoothing technique is not appropriate because of the intermittence of the PV production.

**GBR** The hyperparameters providing the best performances for the different models are summarized in table F.1. For the forecasting of the load, the single output GBR provides very good performances with a MAE of 591 W and a RMSE of 1064W. The performances of the multi-output GBR are also significantly better than the performances of the naive forecaster with a MAE of 628 W and a RMSE of 1122 W.

For the forecasting of the PV, the RMSE of the single output model and of the multi-output model are significantly lower than the RMSE of the naive forecaster. Regarding the MAE, the performances of the single output model are similar to the one of the naive forecaster but the performances of the multi-output model are relatively better.

**MLP** The hyperparameters that provide the best performances for the multilayer perceptron are given in the table F.2 in appendix. In the case of the single output models, the selection of the scaler for the inputs and outputs was an important parameter[3] The second parameter that had the largest impact on the performances

[3]The single output model for forecasting the load provided better performances with a min-max scaling of the inputs and a standard scaling of the outputs. With this scalers, the MAE was equal to 712 W and the RMSE was equal to 1128 W. By contrast, with standard scalers for the inputs and the outputs, the MAE is equal to 1196 W and the RMSE is equal to 1172 W. With the single output models for forecasting the PV production, the observations are similar.

is the activation function used. In the end, after optimization of all the parameters, the models for forecasting the load had good performances. For the forecasting of the PV, the single output model has performances similar to the multi-output GBR but the multi-output model still provides worst performances than the naive forecaster.

## 5.3 Training set size analysis

Once the hyperparameters of the models have been determined, the size of the training set size on which the models need to be trained has been analyzed. In the previous experiments, the training set size was set by default to 35 days of data. In this experiment, training set sizes composed of 1 day of data to 42 days of data have been tested. The full analysis of the training set size for the different models is provided in the appendix D.

**Load forecasting** For the forecasting of the load, the training set sizes that provided the best performances for all models are sets composed of 20 to 32 days of data. For all of the models, training set sizes composed of less than one week of data provided power performances. However, with too large training set sizes, the performances were decreasing.

**PV forecasting** For the forecasting of the PV production, the optimization of the training set size has a greater impact on the performances. In general, the best training set sizes for the multi-output models were made of 6 to 13 days of data. With single output modes, the training set sizes that provided the best performances are composed of 21 to 26 days of data. For the multi-output MLR, the single output GBR, and the multi-output MLP, the modification of the training set size provided significant improvements in the performances.

## 5.4 Analysis of the retraining frequency

The final step in the optimization process is the selection of the retraining frequency. The full analysis of the retraining frequency is detailed in chapter E. The maximum retraining period was selected to be done weekly.

**Load forecasting** For most of the models, weekly retraining was sufficient to provide a maximum degradation of the performances[4] of 5%. The exceptions are the EST which needs to be trained daily and the multi-out GBR which needs to be trained every 6 days.

**PV forecasting** Regarding the forecasting of the PV production, there are some variations in the retraining period based on the fact that the model is a multi-output model or not. With single output models, weekly retraining is also sufficient to have a maximum degradation of the NRMSE of 5%. Concerning the multi-output models, the MLR and the GBR need to be retrained every 2 days and the GBR need to be retrained every 6 days.

[4] in term of NRMSE (Eq. 3.3)

# Chapter 6

# Dataset of forecasts

## 6.1 Introduction

To be able to construct and use the monitoring system, a dataset of forecasts obtained with different models has been built. This dataset has been constructed by using the learned optimal training period and retraining period to produce forecasts in a window of time from the 14 October 2019 to the 31 December 2019. More formally, the dataset is created by following the methodology described in the algorithm 2. The forecasts of the load and of the PV are respectively analyzed in the sections 6.2 and 6.3.

## 6.2 Analysis of the forecasts of the load

The table 6.1 presents the MAE and RMSE obtained by the different models for all the study period. The models based on GBR and MLP provides the best performances. Specifically, the single output GBR outperforms all other models in terms of MAE and RMSE. Globally, the model that provides the best performances on the whole dataset is the single output GBR. It can be observed that in terms of MAE and RMSE, the GBR using a single output provides better performances on the global dataset. The multi-output MLR is the only model that provides worst performances than the naive forecaster.

| **Model** | **Output mode** | **MAE (W)** | **RMSE (W)** |
|---|---|---|---|
| Naive forecaster | / | 819 | 1534 |
| EST | / | 733 | 1364 |
| MLR | single | 801 | 1271 |
| | multi | 907 | 1619 |
| GBR | single | 613 | 1073 |
| | multi | 671 | 1172 |
| MLP | single | 644 | 1139 |
| | multi | 690 | 1191 |

Table 6.1: Performances of different models for the forecasting of the load ordered by RMSE

In the picture 6.1, a display of the forecasts of the different models for a random week in the study period is provided.

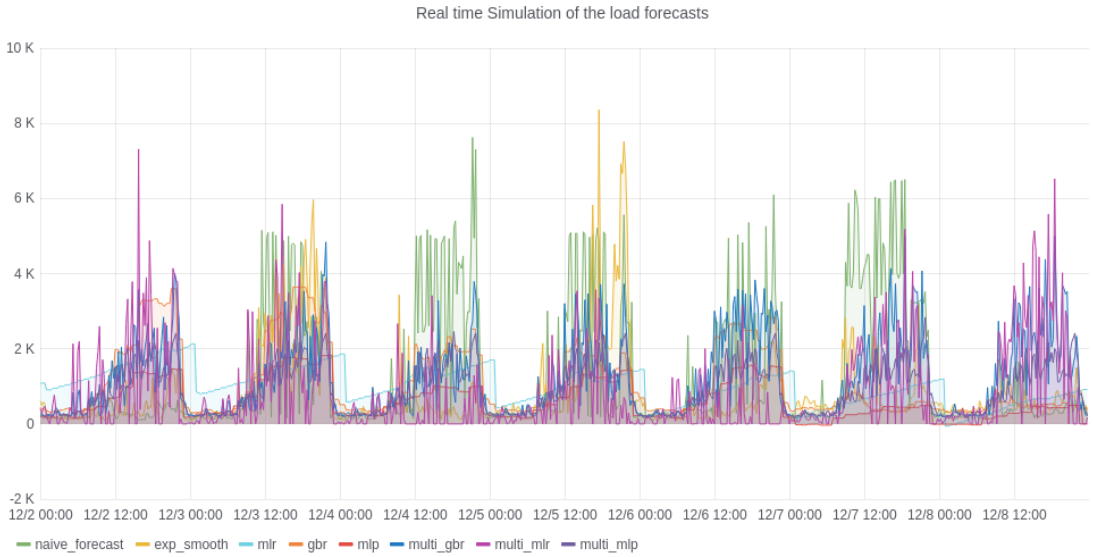



Figure 6.1: Forecasts of the load by different models for the week of the 2nd of December 2019

### 6.2.1 Daily performances

In figure 6.2, the daily RMSE of the forecasts are shown for all the study period. Except on some particular days, the magnitude of the error of the different models tends to be similar. Indeed, some days the mean RMSE is high for all models (e.g. RMSE between 2000 and 3000W), and some other days the mean RMSE is low for all models (e.g. RMSE between 0 and 1000W).

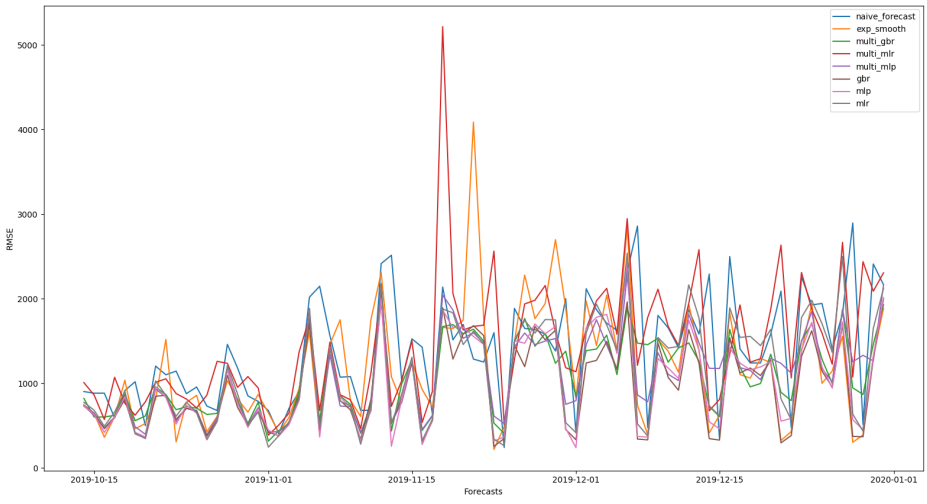



Figure 6.2: Daily RMSE of the different models during the study period

The percentage of days during which a particular model outperforms all the other

models is displayed in the table 6.2. This percentage is computed based on the MAE and based on the RMSE. The single output MLP produces a better daily MAE than the other models 20% of the days from the studied period. In terms of RMSE, it is the single output GBR that had the best performances 28% of the days.

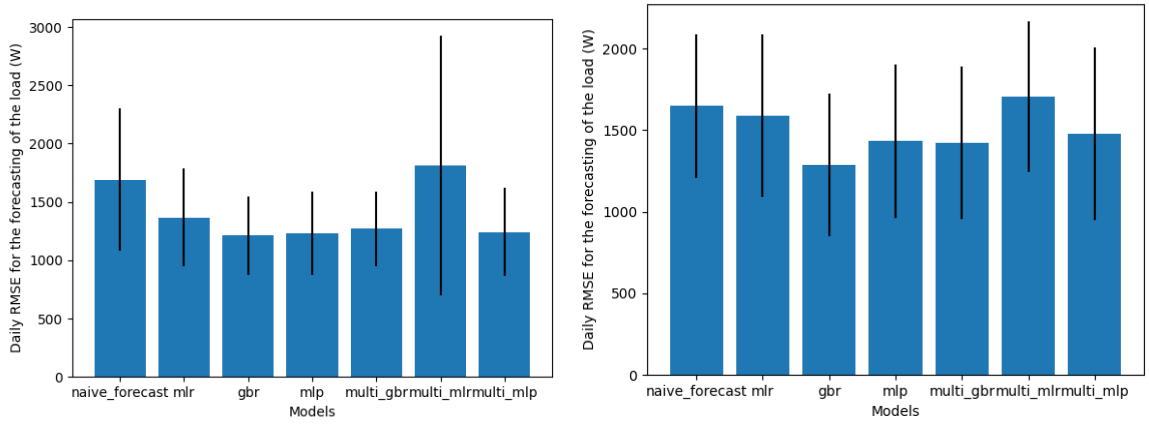
| **Model** | **Output mode** | **MAE** % | **RMSE** % |
|---|---|---|---|
| Naive forecaster | / | 16 | 5 |
| EST | / | 22 | 18 |
| MLR | single | 5 | 8 |
| | multi | 3 | 4 |
| GBR | single | 14 | 28 |
| | multi | 10 | 10 |
| MLP | single | 20 | 15 |
| | multi | 10 | 12 |

Table 6.2: Percentage of the days during the study period during which one model outperforms in the sense of the daily MAE or daily RMSE

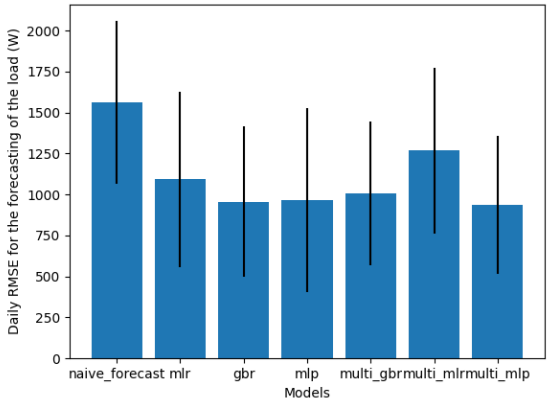
### 6.2.2 Performances by day of the week

In figure 6.3, the mean RMSEs of the models by day of the week are shown. Each plot represents the performances of the models for a given day of the week. Each bar corresponds to the performances of a different model. The performances are represented thanks to the mean RMSE and the standard deviation of the RMSE.

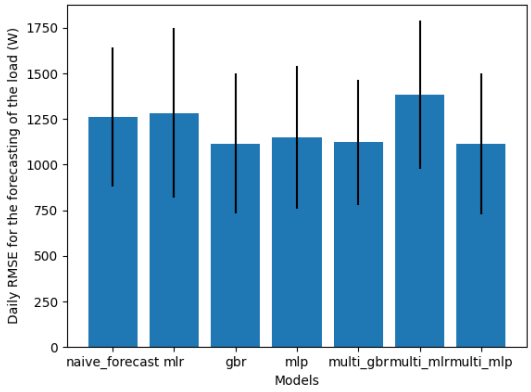
As can be seen, the performances of the different models depend greatly on the day of the week. During some days of the week, all the models have similar performances and during other days of the week, some models outperform the other models. For example, on Mondays, the models with the best performances are the multi-output MLP (bar on the right), while on weekend days, the model that provides the best forecasts is the GBR (4th bar).



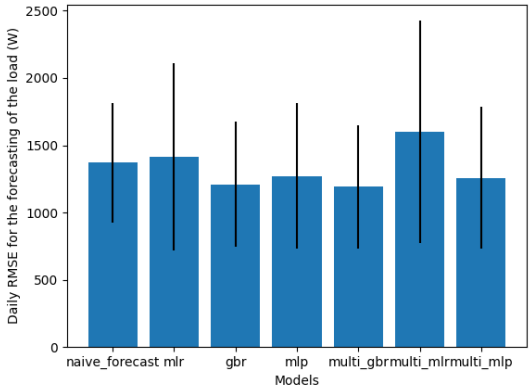

Monday Tuesday



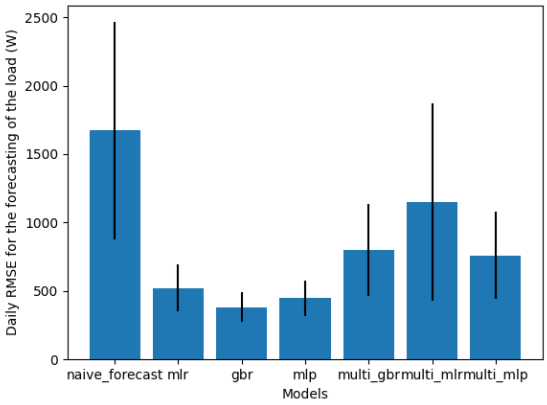

Wednesday



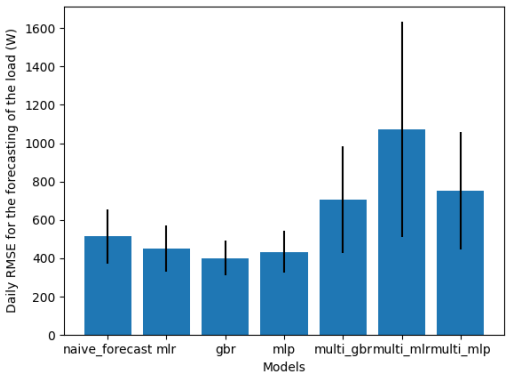

Thursday




Friday




Saturday




Sunday

Figure 6.3: mean RMSE per day of the week for different models

### 6.2.3 Performances per time of the day

In the figure 6.4, the mean error by time of the day is displayed. As can be seen, there are variations of the performances of the models depending on the time of the day. For example, the MLR (green curve) is the model with the biggest RMSE in the

interval from 10 PM to 5 AM. However, during the rest of the day, its performances are in the average.

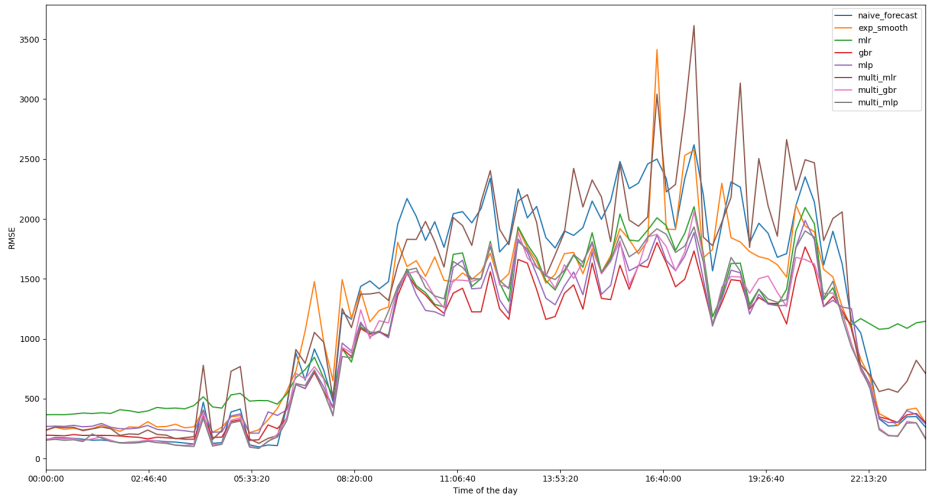



Figure 6.4: Mean RMSE at each time interval of a day

## 6.3 Analysis of the forecasts of the PV

In table 6.3, the performances of the models on the whole studied period are shown. The two first metrics are the MAE and RMSE described before. The two last metrics are normalized versions of the MAE and RMSE where the divisor is chosen to be the maximal production of the PV observed on the period studied (being 2152 W). In terms of RMSE, the best model is the single output MLP. In terms of MAE, the best performances are obtained with the multi-output GBR which slightly outperforms the single output MLP. The single output GBR provides good performances as well.

| **Model** | **Output Mode** | **MAE (W)** | **RMSE (W)** | $\frac{\textbf{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\textbf{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|---|
| Naive forecaster | / | 98 | 228 | 4.5 | 10.6 |
| MLR | single | 95 | 179 | 4.4 | 8.3 |
| | multi | 108 | 272 | 5.0 | 12.7 |
| GBR | single | 76 | 164 | 3.5 | 7.6 |
| | multi | 72 | 171 | 3.4 | 7.9 |
| MLP | single | 75 | 162 | 3.5 | 7.5 |
| | multi | 83 | 181 | 3.8 | 8.4 |

Table 6.3: Performances of different models for the forecasting of the PV production ordered by RMSE

In the figure 6.1, the forecasts of the models are presented for three consecutive days from the dataset.

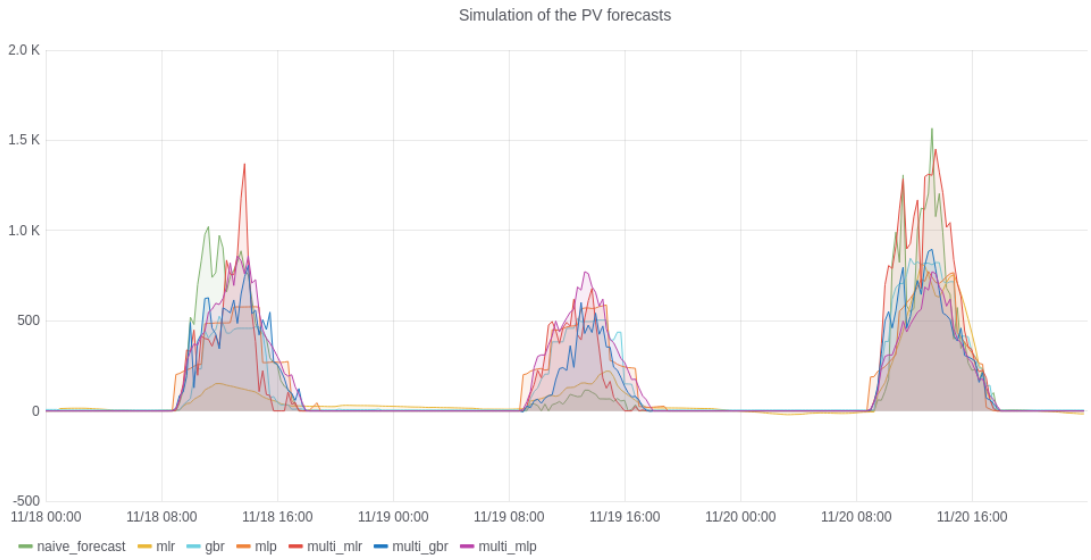



Figure 6.5: Forecasts of the PV production by different models for 3 days in November 2019

### 6.3.1 Daily performances

In figure 6.6, the daily RMSEs of the forecasts are computed for the studied period. As can be seen in brown, the performances of the multi-output MLR are relatively low compared to the other models during some particular days of the studied period. The naive forecaster (blue curve) seems to have a higher RMSE than other models most of the time.

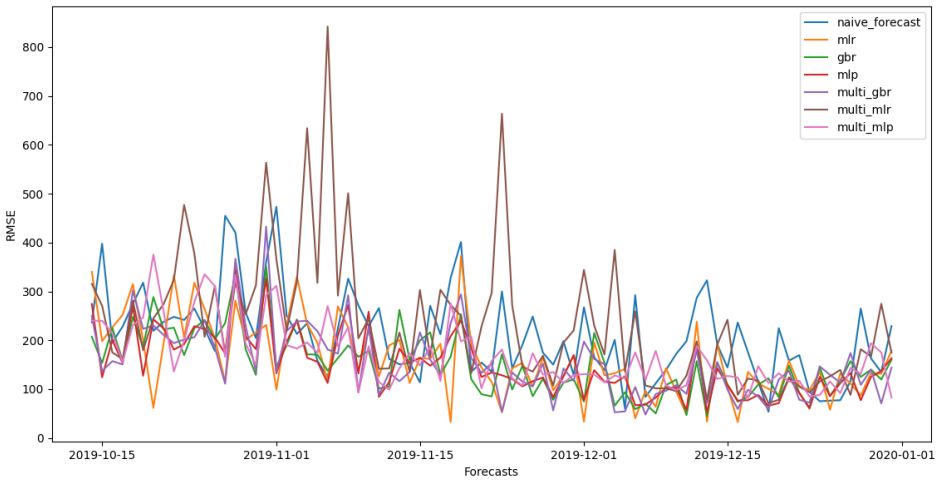



Figure 6.6: Daily RMSE of the different models during the study period

In the table 6.2, the percentage of the days a particular model had the best performances compared to all the other models is shown. The models that provided a lower MAE 27.85% of the day is the multi output GBR. In terms of RMSE, the model that produced the lower daily RMSE most of the time is the single output GBR.

| Model | Output Mode | MAE (%) | RMSE (%) |
|---|---|---|---|
| Naive forecaster | / | 13.9 | 6.3 |
| MLR | single | 8.9 | 17.7 |
| | multi | 3.8 | 2.5 |
| GBR | single | 13.9 | 21.5 |
| | multi | 27.9 | 19.0 |
| MLP | single | 13.9 | 16.5 |
| | multi | 17.7 | 16.5 |

Table 6.4: Percentage of the days during the study period during which one model outperforms in the sense of the daily MAE or daily RMSE

### 6.3.2 Performance by time of the day

In figure 6.7, the mean performances of the models per time of the day are shown. As was observed with the forecasting of the load, the MLR provided worse performances than the other models during the night (7 PM to 8 AM). During the period from 9 AM to 3 PM, the worst model in mean is the multi-output MLR and the second worst model is the naive forecaster.

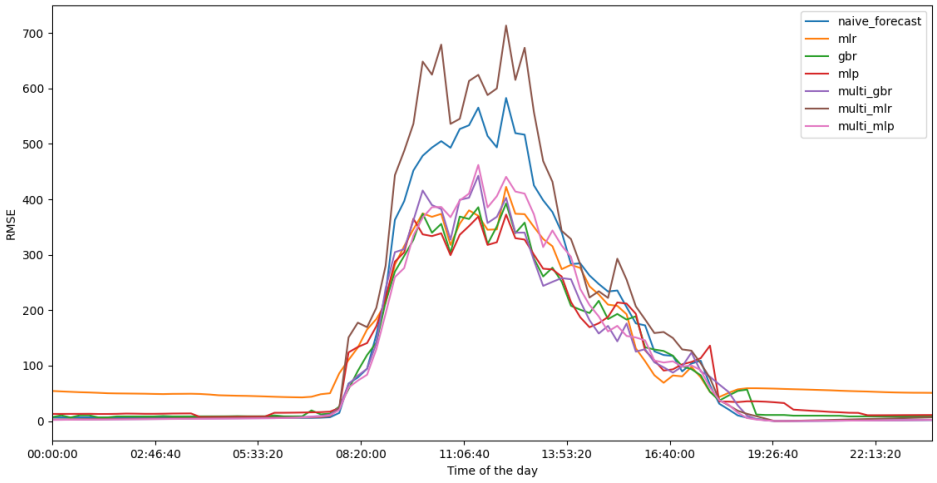



Figure 6.7: Mean RMSE at each time interval of a day

## 6.4 Conclusion

In this chapter, the dataset used for simulating forecasts has been created and analyzed. In the next chapter, a study of the strategy employed to combine this forecast in multi-model forecasts is done.

# Chapter 7

# Monitoring system analysis

## 7.1 Introduction

In this chapter, several multi-model ensemble forecasting techniques were analyzed. These multi-model forecasts were tested on the dataset of forecasts created in the previous chapter.

In section 3.4, the background under multi-model forecasting is developed and in section 4.7, the methodology under these experiments is detailed.

In the section 7.2 , different forms of linear combinations using a set of combination weights $w_1, ...., w_N$ are tested.

$$F_i = w_o \sum_{j=1}^{n} w_i^{(j)} f_i^{(j)} \tag{7.1}$$

Firstly, some simple combinations scheme based on fixed combination weights are tested. These methods are the average forecasting method, the truncated mean method, and the median method. Then some methods that use time-varying weights are analyzed.

## 7.2 Analysis of the multi-models

### 7.2.1 Model averaging

In this experiment, the model averaging technique has been used. The tests were done by averaging all the models together and also by combining with equal weights the three models with the best performances in terms of RMSE. For the forecasting of the load and of the PV, the three models with the best performances are the single output GBR and MLP and the multi-output MLP.

**Load forecasting** For the forecasting of the load, the averaging of all individual models together results in a MAE of 647W and a RMSE of 1102 W. In terms of RMSE, the average model thus ranks just after the single output GBR. In terms of MAE, the average model follows the single output GBR and the single output MLP. When combining the forecasts of the three best performing models, the MAE

is 614 W and the RMSE is 647 W. These values are slightly higher than the MAE and RMSE obtained with the forecasts from the single output GBR.

**PV forecasting** For the forecasting of the load, the averaging of all the models together provides a MAE of 72 W and a RMSE of 154 W. This is an improvement on all the individual models used alone as the best model was the single output MLP which had a MAE of 75 w and a RMSE of 162 W. When combining only the three best models, the MAE reaches 70 and the RMSE keeps the value of 154W.

### 7.2.2 Truncated mean

In this experiment, the truncated mean[1] was used on the forecasts from the individual models. The percentage of points to discard on both ends of the mean was tested with different values in the range[0,0.48] with steps of 0.02. In the appendix, the figure G.1 and G.2 show the impact of the selection of the cut off on the performances of the multi-model.

**Load forecasting** From figure G.1, it can be observed that the increase of the cut off decreases the MAE (from 646 W to 607 W) while increasing the RMSE (from 1102 W to 1117 W). It thus shows that the usage of the mean decreases the RMSE while the usage of the median decreases the MAE.

**PV forecasting** From figure G.2, it can be observed that the increase of the cut off improves both the MAE and the RMSE. The MAE reaches 66W and the RMSE reaches 151.5 W with $\lambda \in [0.3, 0.42]$. Then, they increase very slightly. In the case of the forecasting of the PV, the usage of the median provides better performances over the usage of the mean.

### 7.2.3 Median forecast

**Load forecasting** Let's consider the usage of the median for the forecasting of the load. As mentioned in the previous section, the usage of the median provides a MAE of 607W and a RMSE of 1117W. In terms of MAE, the median provides better performances than all individual models used alone. In terms of RMSE, the median forecast underperforms the single output GBR but outperforms all other individual models.

**PV forecasting** Considering the forecasts of the PV, the resulting MAE is 66 W (3%) and the resulting RMSE is 152 W (7%). The usage of the median thus provides better results in terms of MAE and RMSES than all individual models alone and it also provides better performances than the average model.

### 7.2.4 Best model selection

In this experiment, the performances of the individual models are evaluated on a past window of time and the model with the best performances is selected to produce

[1] see Eq. 3.12

forecasts. The combination weights are updated in this way at the end of each day for the next day. Two main parameters have to be taken into account. The first parameter is the metric with which to evaluate the performances of a model and the second parameter is the window of time in the past on which the performances are evaluated. As an example, with a window of time of one day and a performance metric to optimize which is the RMSE, the multi-model evaluates the RMSE of the individual forecasts on the previous day and uses the model with the lowest RMSE to produce forecasts for the next day. The impact of these two parameters will thus be analyzed in the next sections and the models selected with this strategy will be observed as well.

## Load forecasting

**Performance metric** Let's consider the impact of the performance metric used for forecasting the load. First, the situation in which all the available past forecasts are used to evaluate the performances of the different models can be analyzed. The results obtained with this technique are shown in table 7.1. As can be seen, when the performance metric to optimize is the RMSE, the multi-model forecasts have performances very close to the single output GBR model. When the performance metric to optimize is the MAE, the multi-model forecasts have worst performances but it is still better than most forecasts created by individual models.

| **Metric minimized** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| RMSE | 617 | 1076 |
| MAE | 623 | 1109 |

Table 7.1: Performances by minimizing the RMSE/MAE with a window size that covers all the preceding observations

**Window of time** In figure 7.1, the performances of the multi-model forecasts are shown as a function of the size of the window of time on which the performance metric was computed at each update of the model. The performance metric used for optimization in this case is the RMSE. The usage of small windows of time for computing the performances of the different models leads to lower performances. Above windows of 14 days, the performances of the multi-model don't change any more. Indeed, when performances are evaluated on a small window, the variations of the performances of the individual models can be more important. When the window passes above a given threshold, the model that gives the best performances in the long term will always be selected.

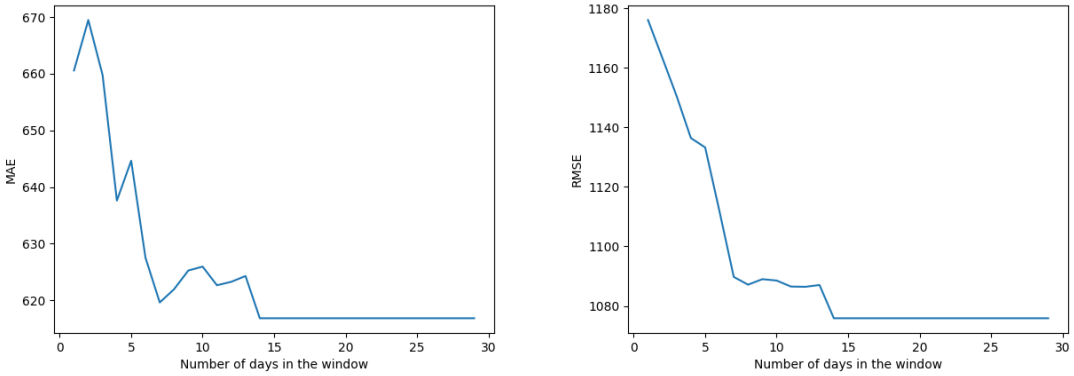



Figure 7.1: Performances of the multi-model forecast by selecting each day the model that had the lowest RMSE on a past time window

In figure 7.2, the performance metric used for optimization is the MAE. In this case, the increase of the window size leads to an improvement of the performances until the window size is composed of 14 days of data. With a window size of 14 days, the MAE is of 611 W and the RMSE is of 1083W, the multi-model has thus similar performances than the single output GBR which is the best individual model. Above this value, the performances start to decrease until reaching a plateau when the window is composed of thirty days.

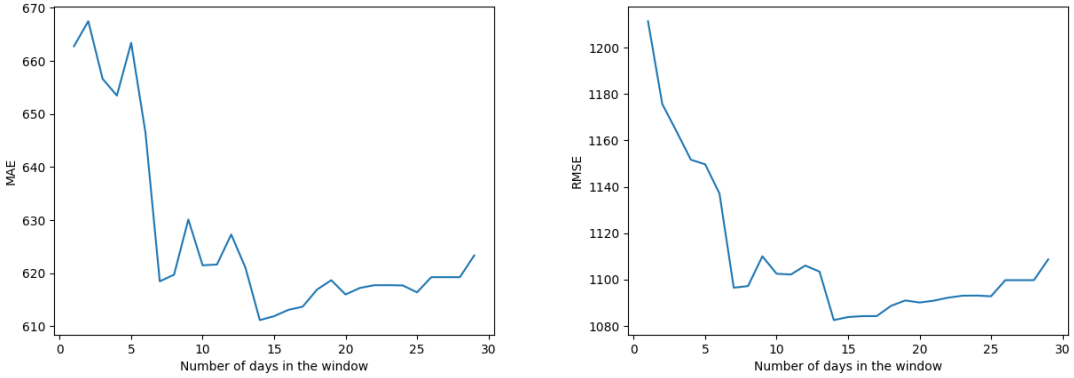



Figure 7.2: Performances of the multi-model forecast by selecting each day the model that had the lowest MAE on a past time window

**Models selected** Finally, it is interesting to observe the models selected at each iteration by the multi-model. When the window size is of 30 days and the performance metric is the RMSE, the models selected at each iteration are the following:

EST $\rightarrow$ m.o. MLP $\rightarrow$ EST $\rightarrow$ EST $\rightarrow$ s.o. MLP $\rightarrow$ s.o. MLP $\rightarrow$ s.o. GBR $\rightarrow$ ... $\rightarrow$ s.o. GBR

with "s.o." representing a single output model and "m.o." representing a multi-output model.
In the beginning, the model selected varies from day to day. After one week, the model selected is the single output GBR (7th model in the example) and this model is selected at each following iteration.

## PV forecasting

**Performance metric** Now, let's consider the impact of the choice of the performance metric for the forecasting of the PV. First, the performances of the individual models can be assessed on an expanding window that uses all the available past data. In table 7.2, the performances of this model optimized based on the MAE and based on the RMSE are shown. As can be seen, the multi-models built don't outperform the performances obtained with forecasts made by the individual models. It provides performances in the average compared to the individual models.

| **Metric optimized** | **MAE** (W) | **RMSE** (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| RMSE | 80 | 171 | 3.7 | 7.9 |
| MAE | 77 | 179 | 3.6 | 8.3 |

Table 7.2: Performances by minimizing the RMSE/MAE with a window size that covers all the preceding observations

**Window of time** The usage of smaller windows can improve the performances but the MAE never drop down 67 W and the RMSE never drops down below 167 W. The tables summarizing the performances of the models with different window sizes and the related figures are shown in section G.3.

**Models selected** With a window size of 30 days and optimization of the RMSE, the individual model selected at each iteration vary from day to day during all the studied period. The multi-model never selects forecasts from the naive forecaster, the single and multi-output MLR, and the multi-output MLP. The forecasts from the GBR and MLP are mainly selected[2] and the forecasts from the multi-output GBR are also used some days[3]. The models selected are thus the 3 models with the best performances on the whole dataset.

### 7.2.5 Inverse MSE weighting

The inverse MSE weighting involves the computation of the MSE of the individual models on a past window of time and its usage for creating combination weights inversely proportional to the MSE. Mathematically, it is given by the equation 3.14 in section 3.4. With this strategy, the only parameter is the window of time that is used for computing the MSE. Let's consider its impact on the forecasting of the load and then on the forecasting of the PV.

**Load forecasting** Figure 7.3 presents the performances of the multi-model with window sizes from 1 day to 30 days. As can be seen in the figure, the performances are optimized when the multi-model uses a window size of 7 days. In this case, the MAE is of 636 W and the RMSE is of 1087W. The model has poorer performances than the single output GBR but better performances than all the other models.

[2]The single output GBR is selected 43% of the days and the single output MLP is selected 53% of the days.

[3]The multi-output GBR is selected 4% of the time

With smaller or longer windows, the performances of the multi-models are poorer but the variations are not large (less than 10 W for both metrics).

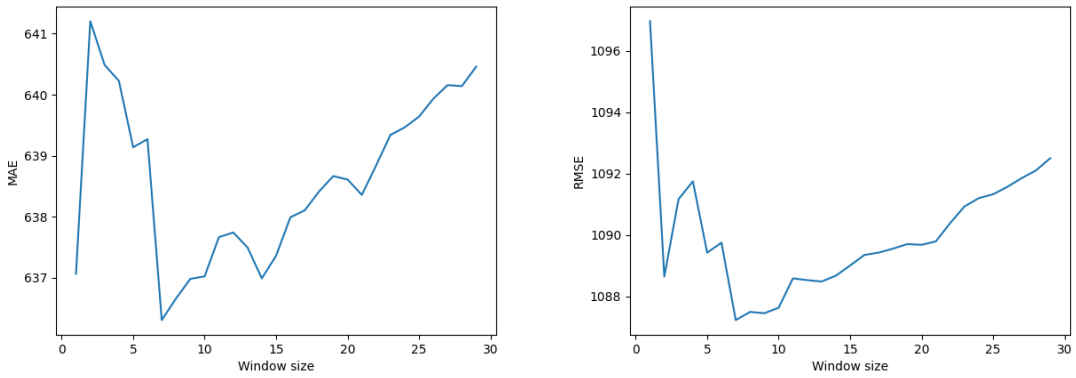



Figure 7.3: Performances of the multi-model with different windows of time for the computation of the MSE

**PV forecasting** Let's consider the usage of inverse MSE weighting for forecasting the PV. Once again, the multi-model is tested with windows from 1 to 30 days. As can be seen in figure 7.4, when the MSE is computed on a window of time larger than 1 day, the MAE is approximately equal to 71 W and the RMSE is approximately equal to 152W. Thus, when the MSE is computed on a window of time above 1 day, the performances of the multi-model in terms of MAE and RMSE are better than the performances of the forecasts of all individual models.

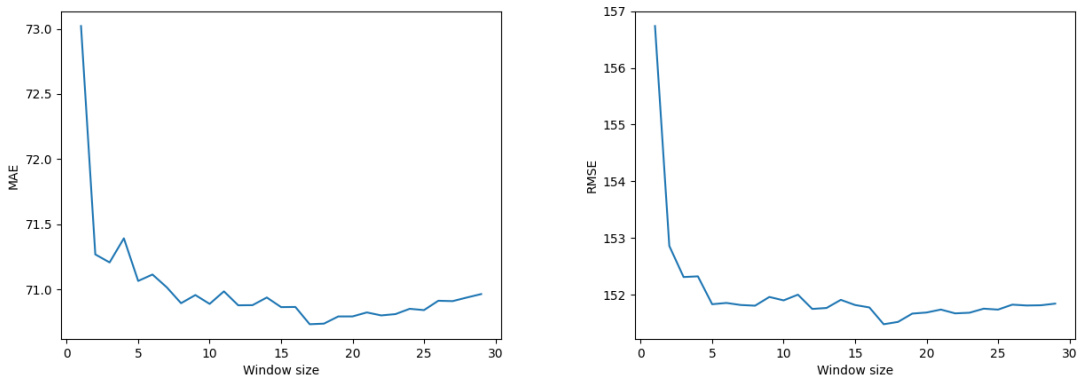



Figure 7.4: Performances of the multi-model with different windows of time for the computation of the MSE

### 7.2.6 Weighted linear regression

With this strategy, the combination weights are obtained by minimizing the residual sum of squares between the observed load or PV and the multi-model forecast. The parameters that can affect the performances are the size of the window on which the multi-model is trained and the usage of a constant in the combination of the models.

**Load forecasting** In table 7.3, the performances of the model with a window that uses all the past data are shown. The multi-model outperforms all individual models in terms of RMSE but in terms of MAE it has performances similar to the single output MLP. In figure 7.5, the performances of the multi-models are shown as a function of the size of the window that is used for training the model. As can be seen, the error decreases until reaching a minimum when the window is composed of 10 days. After that, the error fluctuates slightly.

| **Parameters** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| $w_0$ not fixed | 667 | 1053 |
| $w_0 = 0$ | 652 | 1049 |

Table 7.3: Performances of the weighted linear regression for the forecasting of the load

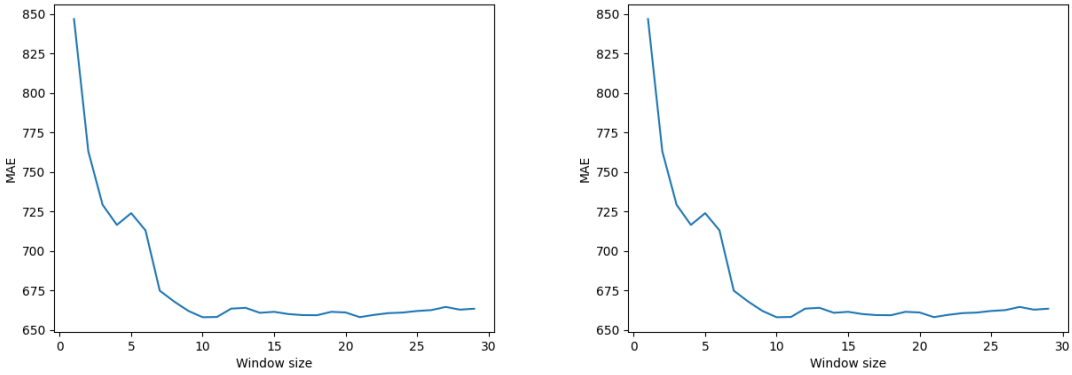



Figure 7.5: Performances of the multi-model with a window size that covers all the preceding observations

**PV forecasting** In table 7.4, the performances of the model with a window that uses all the past data are shown. In terms of RMSE, the multi-model will thus outperform all individual models. In terms of MAE, the performances of the multi-model are very good as well since the model that has the lower MAE is of 72 W and is obtained with the multi-output GBR. In figure 7.6, the performances of the multi-models are shown as functions of the size of the window that is used for training the model. As can be seen, the MAE and the RMSE decrease when the window size increases until the size of the window is of 18 days. Then, the MAE and the RMSE slightly increase.

| **Parameters** | **MAE** (W) | **RMSE** (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| $w_0$ not fixed | 75 | 157 | 3.47 | 7.29 |
| $w_0 = 0$ | 73 | 159 | 3.40 | 7.37 |

Table 7.4: Performances by minimizing the RMSE/MAE with a window size that covers all the preceding observations

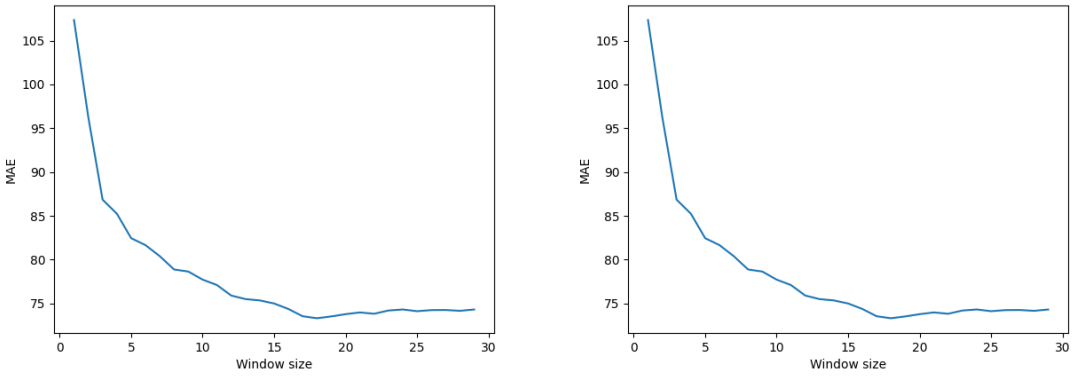



Figure 7.6: Performances of the multi-model with a window size that covers all the preceding observations

### 7.2.7 Model combination with training based on the day of the week

As has been seen in the previous chapter, the performances of the individual models for forecasting the load may vary depending on the day of the week. In this experiment, the combination weights for a given day are obtained by training the multi-model only on the forecasts made the same day from the previous weeks. The multi-models tested with this method are the multi-model that selects the best model, the multi-model using inverse MSE weighting, and the weighted linear regression. The results are provided in table 7.5.

The multi-models that selects the model with minimum RMSE or MAE for producing forecasts provided better performances in terms of MAE and worst performances in terms of RMSE compared to a model in which the training is done on all the past data. In terms of MAE, it even outperformed all individual models. The multi-model that uses inverse MSE weighting improved its MAE (625W compared to 641W when trained on all data) and its RMSE ( 1088W compared to 1094W when trained on all data). Finally, the multi-models based on linear regression led to degradation in terms of RMSE.

| **Weighting** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| Min RMSE selection | 607 | 1097 |
| Min MAE selection | 620 | 1144 |
| Inverse MSE weighting | 625 | 1088 |
| weighted linear regression | 677 | 1117 |
| weighted linear regression with $w_0 = 0$ | 648 | 1089 |

Table 7.5: Performances of models taking only into account the data from this day of the week

### 7.2.8 Weighted linear regression with additional explanatory variables

In this section, a weighted linear regression is used for updating the weights with some additional explanatory variables. For forecasting the load, the explanatory variables used are the hour of the day and the day of the week. For forecasting the PV, the explanatory variable used is the hour of the day.

**Load forecasting** In table 7.6, the results for the forecasting of the load with different seasonality variables are shown. Compared to the simple linear regression, the introduction of the day of the week alone or combined with the hour of the day improved slightly both the MAE and the RMSE.

| **Additional inputs** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| hour, day of the week | 650 | 1046 |
| hour | 657 | 1052 |
| day of the week | 652 | 1046 |

Table 7.6: Performance of linear regression with $w_0 = 0$ for forecasting the load

**PV forecasting** The introduction of the hour of the day in the inputs does not lead to a significant change in the performances compared to a simple linear regression.

| **Additional inputs** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| hour | 74 | 157 |

Table 7.7: Performance of linear regression with $w_0 = 0$ for forecasting the PV

## 7.3 Impact of the frequency of calibration

To decrease the computational cost, the multi-models that require periodic updates can be trained weekly rather than daily. This strategy has been adopted and the performances of this models are shown in the table 7.8 and 7.9.

**Load forecasting** As can be seen in table 7.8, the multi-model that selects the model that produced the lower MAE on a past window of time has better performances when it is updated weekly rather than daily. With the other methods, the choice of updating weekly the models increase the MAE and/or the RMSE. This increase is at most of 15 W for the RMSE and of 10 W for the MAE.

| Multi-model weighting | Weekly update | | Daily update | |
|---|---|---|---|---|
| | MAE (W) | RMSE (W) | MAE (W) | RMSE (W) |
| Min RMSE selection | 630 | 1105 | 620 | 1090 |
| Min MAE selection | 614 | 1089 | 618 | 1097 |
| Inverse MSE weighting | 636 | 1089 | 636 | 1087 |
| LR weighting | 677 | 1063 | 667 | 1052 |

Table 7.8: Performances of different model summarized

**PV forecasting** In this case, the highest increase of MAE is of 4W, and the highest increase of RMSE if of 5W. Therefore, since the variations in the performances are slights, it would be more interesting to update the multi-models weekly to reduce the computational cost.

| Multi-model weighting | Weekly update | | Daily update | |
|---|---|---|---|---|
| | MAE (W) | RMSE (W) | MAE (W) | RMSE (W) |
| Min RMSE selection | 81 | 168 | 80 | 171 |
| Min MAE selection | 81 | 184 | 77 | 179 |
| Inverse MSE weighting | 71 | 152 | 71 | 152 |
| LR weighting | 75 | 158 | 75 | 157 |

Table 7.9: Performances of different model summarized

## 7.4 Computational cost

In the table 7.10, the training time and the forecasting time of the multi-models are shown. These times are computed by averaging the training time and the time of construction of the forecasts over the period from the 14th of October to the 31st of December when the multi-models are updated daily. The multi-models that require training all used the same window of time of 30 days.

The average model, truncated mean model, and median model had a 0 training time as they require no training. The multi-model using inverse MSE weighting leads to the highest training time. The time required to construct the forecasts for the next day was minimal for the multi-model based on the median and on the truncated mean. The multi-model based on linear regression leads to a slightly larger time compared to all of the other models.

| **Combination method** | Parameters | **Training time** (ms) | **Time of construction of forecasts** (ms) |
|---|---|---|---|
| Average model | / | 0.0 | 1.58 |
| Truncated mean | $\lambda = 0.2$ | 0.0 | 1.28 |
| Median | / | 0.0 | 1.28 |
| Min RMSE selection | $w = 30$ | 7.5 | 1.58 |
| Inverse MSE | $w = 30$ | 5.6 | 1.60 |
| LR weighting | $w = 30$ | 5.1 | 2.14 |

Table 7.10: Comparison of the times of the multi-model forecasting techniques for the forecasting of the load for multi-models which are updated daily

## 7.5 Comparison of the multi-models

In the tables 7.11 and 7.12, the performances obtained with the different multi-models techniques are summarized for the forecasting of the load and of the PV production. In both cases, the model that gave the lower MAE among all multi-models is the model based on the selection of the median forecast.

**Load forecasting**

The model that leads to the minimal RMSE is the model in which the weights are obtained thanks to linear regression. The usage of additional seasonality variables even improved the performances.

| **Combination method** | MAE (W) | RMSE (W) |
|---|---|---|
| Average model | 647 | 1102 |
| Median model | 607 | 1117 |
| Min RMSE selection | 620 | 1090 |
| Min MAE selection | 618 | 1097 |
| Inverse MSE weighting | 636 | 1087 |
| LR weighting with $w_0 = 0$ | 652 | 1049 |
| LR weighting with $w_0 = 0$ and additional inputs | 650 | 1046 |

Table 7.11: Performances of different model summarized

**Pv forecasting**

For the forecasting of the PV, the method that produced the lower RMSE is the multi-model based on the median and the model based on inverse MSE weighting. The best multi-model to use for forecasting the PV would thus be the median as it both minimizes the MAE and the RMSE and as it does not require training.

| **Combination method** | **MAE** (W) | **RMSE** (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| Average model | 72 | 154 | 3.3 | 7.2 |
| Median model | 67 | 152 | 3.1 | 7.1 |
| Min mae selection | 77 | 179 | 3.6 | 8.3 |
| Min rmse selection | 80 | 171 | 3.7 | 7.9 |
| Inverse MSE weighting | 71 | 152 | 3.3 | 7.1 |
| LR weighting with $w_0$ not fixed | 75 | 157 | 3.5 | 7.3 |
| LR weighting with $w_0 = 0$ | 73 | 159 | 3.4 | 7.4 |

Table 7.12: Performances of different model summarized

## 7.6 Conclusion

In this chapter, several multi-model forecasting techniques were used. They were compared in terms of their performances and training time.

For the forecasting of the load, the method that led to the best performances in terms of MAE is the median forecast. However, the method that leads to the best performances in terms of RMSE is the linear regression method. While the median is the method that produces the fastest the forecasts, the method based on linear regression requires training and the time of construction of the forecasts of the forecasts is the slowest.

The method that leads to the best performances on the PV forecasting is the multi-model that selects the median of the individual forecast. A big advantage of this method is also the fact that it requires no training and it is the fastest method for producing forecasts.

# Conclusion

In this thesis, the problem of generating short-term forecasts of the energy consumption and production of a microgrid was addressed. A set of models have been created and optimized and these models have been used in a multi-model forecasting system.

The individual models used in this thesis range from statistical time series models to artificial intelligence-based models. The statistical time series models used are the seasonal naive forecaster and the exponential smoothing technique. The artificial intelligence-based models are the linear regression, the gradient boosting regression, the multilayer percetron, and the Gaussian regression. These models were tested in a single output mode and a multi-output model. In a single output mode, input samples are used to produce forecasts for one time step at a time. In a multi-output mode, the input samples are modified and are used to produce forecasts for one day ahead.

The first challenge faced was the selection of the best input features for training the artificial intelligence-based models. Regarding the load forecasting, the input features that provided the best performances were the weather information combined with the seasonality variables for capturing the day of the week, the hour of the day, and the month of the year. When considering the models used for forecasting the PV production, the more relevant input features were the subset of the weather variables specific to the PV (i.e. the solar irradiation, temperature, and solar irradiation squared).

After this phase of discovery of the input features, a tuning of the hyperparameters of the models has been done. Firstly, the parameters of the exponential smoothing techniques were tuned. This model had bad performances for the forecasting of the PV because of the intermittence of this time series. The methods based on gradient boosting and multilayer perceptron were optimized and provided good performances. The last models being optimized are the models based on Gaussian regression. These models did not provide good performances and their training time was relatively long especially for the single output models.

Then, the problem of the selection of the amount of data for training the models was addressed. This can also be a challenging task as the distribution of the data may vary over time. For the forecasting of the load, training the models on 20 to 32 days of data in the past was optimal. For the forecasting of the PV production, the multi-output models required shorter training periods (between 6 to 13 days of data) while the single output models required longer training periods (between 21 and 26 days of data).

After that, an analysis of the retraining frequency was done. With most of the models, weekly retraining of the model provides forecasts with at most 5% of degradation of the NRMSE compared to daily retraining. With the multi-output models for forecasting the PV production, the retraining should be done at a higher frequency (every 2 to 6 days).

Once all the model parameters and hyperparameters were tuned, a dataset was created with the simulated forecasts from the different models for the period from the 14th of October 2019 to the 31st of December 2019. The single output gradient boosting regression and the single output multilayer perceptron were the best models for forecasting the load and the PV.

Finally, several multi-model forecasting techniques were explored for combining the forecasts of the different models. Firstly, some simple methods that require no training of the multi-models were tested. These methods are the mean forecast, the truncated mean forecast, and the median forecast. Then, some methods that require training of the multi-models were explored. Firstly, a multi-model that selects the forecasts from the best model was tested, then inverse MSE weighting and methods based on weighted linear regression were used. For the forecasting of the load, the MAE was minimized by using the multi-model selecting the median forecast and the RMSE was minimized by using a linear regression strategy for the weighting of the forecasts. For the forecasting of the PV, the usage of the median leads to the best performances. It minimizes simultaneously the MAE, the RMSE, and the computational time linked to the usage of the method.

Since the load is a time series with big variations, the final results obtained for its forecasting are still inaccurate. However, for the forecasting of the PV production that is a much more predictable quantity, the models provide quite good performances.

# Appendices

# Appendix A

# Forecasting techniques

There are two types of forecasting methods: The *qualitative forecasting methods* and the *quantitative forecasting methods*. Quantitative forecasting can be applied when numerical information about the past is available and when it is reasonable to assume that some aspects of the past patterns will continue into the future. As a consequence, this thesis will only consider quantitative methods.

Quantitative methods can be divided into point forecasting and probabilistic forecasting. Point forecasting is based on the creation of a parametric function $g : \mathcal{X} \to \mathcal{Y}$. Probabilistic forecasting is based on the specification of a distribution over function $g$.

Quantitative methods can also be classified based on the fact that the method used is a statistical time series approach or an artificial intelligence based method.

Quantitative methods can be approached with 3 types of models: (i) explanatory models, (ii) time-series models, (iii) mixed models.

Traditional statistical time series approaches include *naïve methods* (today equals tomorrow, data of same hour last week), *exponential smoothing techniques* (Holt's linear trend method, Holt's winters seasonal method), *ARIMA models*.
In this kind of methods, each time a forecast is produced, the model needs to do rebuild based on new observations that arrived.

## A.1 Statistical time series approaches

### A.1.1 Simple methods

Some simple methods are classically used to produce forecasts.

**Naive approach** The first approach is the naive approach. If $x_T$ is an observation at time T, the naive approach produces the forecast $\hat{x}_{T+h|T}$:

$$\hat{x}_{T+h|T} = x_T \tag{A.1}$$

**Naive seasonal approach** If the data is highly seasonal and the seasonal period is given by $m$,the seasonal naive forecast is:

$$\hat{x}_{T+h|T} = x_{T+h-m(k+1)} \tag{A.2}$$

$$\hat{x}_{T+h|T} = x_{T+h-m(k+1)}$$

**Drift method** A variation on the naive method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the drift) is set to be the average change seen in the historical data.

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right).$$

### A.1.2 Decomposition models

Decomposition models are models which decomposes the time series into several components. When assuming an additive decomposition, the forecast can be given by:

$$y_t = S_t + T_t + R_t,$$

where $y_t$ is the data, $S_t$ is the seasonal component, $T_t$ is the trend-cycle component, and Rt is the remainder component, all at period t. Alternatively, a multiplicative decomposition would be written as

$$y_t = S_t \times T_t \times R_t.$$

### A.1.3 Exponential smoothing models

Depending on the component of the time series (trend and seasonal) and the way in which these components enter the smoothing method (e.g., in an additive, damped or multiplicative manner), different exponential smoothing techniques can be applied.

| **Trend** | **Seasonal** | | |
|---|---|---|---|
| | **N** | **A** | **M** |
| **N** | $\hat{y}_{t+h\|t} = \ell_t$<br>$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$ | $\hat{y}_{t+h\|t} = \ell_t + s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$<br>$s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$ | $\hat{y}_{t+h\|t} = \ell_t s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$<br>$s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$ |
| **A** | $\hat{y}_{t+h\|t} = \ell_t + hb_t$<br>$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ | $\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$<br>$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$ | $\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$<br>$s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$ |
| $\mathbf{A_d}$ | $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$<br>$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ | $\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$<br>$s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$ | $\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$<br>$\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$<br>$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$<br>$s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$ |

Figure A.1: Formulas for recursive calculations and point forecasts. In each case, $l_t$ denotes the series level at time t, $b_t$ denotes the slope at time t, st denotes the seasonal component of the series at time t, and m denotes the number of seasons in a year; $\alpha, \beta, \gamma$ and $\phi$are smoothing parameters, $\phi_h = \phi + \phi^2 + ... + \phi^h$ and k is the integer part of $\frac{h-1}{m}$[10]

| Index | Method |
|---|---|
| (N,N) | Simple exponential smoothing |
| (A,N) | Holt's linear method |
| (Ad,N) | Additive damped trend method |
| (A,A) | Additive Holt-Winters' method |
| (A,M) | Multiplicative Holt-Winters' method |
| (Ad,M) | Holt-Winters' damped method |

Table A.1: Name associated with different exponential smoothing techniques[10]

### A.1.4 ARIMA Models

**Non-seasonal ARIMA models** ARIMA stands for autoregressive integrated moving average.The combination of differencing and autoregression and a moving average model. The model can be written:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{A.3}$$

It is called an ARIMA(p,d,q) model where p is the order of the autoregressive part, d is degree of first differencing involved and q is the order of the moving average part.

**Seasonal ARIMA models** A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$\text{ARIMA} \quad (p,d,q) \quad (P,D,Q)_m$$

where the first part is a non seasonal part of the model and the second term is a seasonal part of the model and m is the number of observations per year.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period.

## A.2 Artificial intelligence-based methods

### A.2.1 Linear regression

A multiple linear regressor model is a model of the form:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + var\epsilon_t$$

where $y$ is the variable to be forecast and $x_1 \ldots, x_k$ are the k predictor variables.

### A.2.2 Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). Parameters of a MLP include its architecture (number of layers, number of

nodes per layer), the activation function that is used, the loss function,... Hyperparameters of MLP are the number of epochs during which the algorithm will passes through the training dataset, the learning rate of the algorithm, the batch size,...

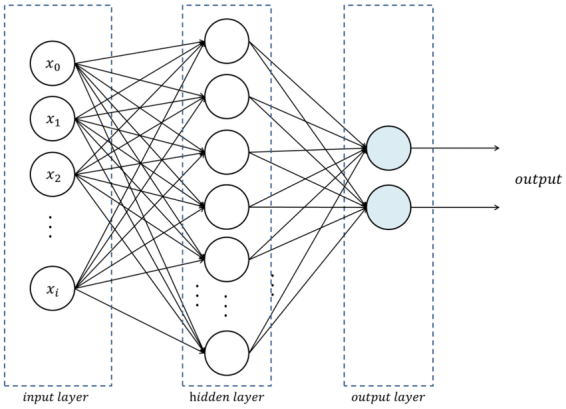



Figure A.2: Representation of a Multilayer Perceptron

**Number of nodes in input layer.** The number of nodes in the input layer is fully determined by the number of independent input in the dataset.

**Number of nodes in the output layer** The number of output nodes is also dependent on the problem. In one-step ahead forecasting, only one output node is needed, in multi-step ahead forecasting, the number of output node should correspond to the forecasting horizon. Alternatively, a single output node can be used and all the future forecasts required determined in the iterative steps.

**Number of hidden layers.** Lippmann (1987), in his paper on neurocomputing, stated clearly that a three-layer perceptron can form arbitrarily complex decision regions and can separate meshed classes, which means that no more than three network layers are needed in perceptron-like feedforward nets.

**Number of neurons in hidden layer.** There is no straight-forward methodology for selecting the *number of hidden neurons* but there exist some rules of thumb. It is generally selected in the range 0.5 to 3 times the number of input nodes.

**Activation functions** Several activation functions can be added on the output of a layer from a neural network. This activation functions can be categorized as linear activation functions and non linear activation functions such as the Sigmoid, the ReLu, the tanh.

**Dropout rate.** In order to reduce overfitting, a dropout can be applied to the network. At each training stage, individual nodes are either dropped out of the net with probability 1-p or kept with probability p.

**Learning rate and number of epochs.** The learning rate is the step size or the magnitude of weight updating. The number of epochs is the number of times the whole training data is shown to the network while training.

**Batch-size.** The batch size is the number of sub samples given to the network after which parameter update happens.

### A.2.3 Gradient Boosting Regressor

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

The parameters of the Gradient boosting method include tree-specific parameters, boosting parameters and miscellaneous parameters. Tree-specific parameters include parameters such as the minimum number of samples per split or per leaf and the maximum depth of a tree. Some parameters that can affect boosting are the learning rate and number of estimators.

### A.2.4 Gaussian Process Regressor

A Gaussian process is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution.

It is completely specified by its mean function $m(x)$ and the covariance function (kernel) $k(x, x')$. The choice of the kernel function is a parameter that needs to be selected. Example of kernels are the white noise kernel, the exponential quadratic kernel, the Matérn kernel, the periodic kernel.

## A.3 AIC

Akaike's Information Criterion is a model selection criterion and is defined as

$$\text{AIC} = T \log\left(\frac{\text{SSE}}{T}\right) + 2(k+2)$$

$$\text{SSE} = \sum_{t=1}^{T} e_t^2$$

where T is the number of observations used for estimation and k is the number of predictors in the model.
For small values of T, the AIC tends to select too many predictors, and so a bias-corrected version of the AIC has been developed,

$$\text{AIC}_\text{c} = \text{AIC} + \frac{2(k+2)(k+3)}{T-k-3}.$$

As with the AIC, the AICc should be minimised.

The AICc is useful for selecting between models in the same class. For example, we can use it to select an ARIMA model between candidate ARIMA models or an ETS model between candidate ETS models. However, it cannot be used to compare between ETS and ARIMA models because they are in different model classes, and the likelihood is computed in different ways[10].

# Appendix B

# Selection of the input features

In this chapter, a short analysis of the performances of the artificial intelligence-based method is done based on the choice of the inputs for training the model. The input features are detailed in section 4.5.1.

## B.1 Linear Regression

### B.1.1 Single output model

**Load forecasting** The performances of the linear regression model for the forecasting of the load are presented in table B.1. It can be observed from table B.1 that the usage of weather information together with dummy variables for capturing seasonality improves the performances compared to the model using only weather information. As a consequence, those inputs will be used to study the linear regression for forecasting the load.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 790 | 1317 | 2 |
| Weather variables and lagged observations | 451 | 874 | 3 |
| Weather and seasonality variables | 763 | 1219 | 10 |

Table B.1: Performances of the MLR model for forecasting the load

**PV forecasting** As can be seen in table B.2, the introduction of dummy variables does not improve the performances of the model. The best performances are obtained with the weather information specific to the PV production. This input was thus selected for the rest of the study.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 114 | 198 | 2 |
| Weather variables (specific to PV) | 104 | 192 | 2 |
| Weather variables and lagged observations | 51 | 106 | 3 |
| Weather and seasonality variables | 116 | 198 | 3 |

Table B.2: Performances of the MLR model for forecasting the PV

### B.1.2 Multi-output model

**Load forecasting** The usage of a multi-output MLR provides results that are below the results obtained with a naive forecaster. The best performances are obtained by using as input the weather information and the dummy variables.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 1153 | 2026 | 420 |
| Weather variables and lagged observations | 488 | 815 | 540 |
| Weather and seasonality variables | 1084 | 1821 | 560 |

Table B.3: Performances of the multi-output MLR trained on different input features for the forecasting of the load

**PV forecasting** The best performances are obtained by using the weather variables specific to the PV.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 170 | 405 | 370 |
| Weather and seasonality variables | 170 | 405 | 360 |
| Weather variables (specific to PV) | 127 | 268 | 370 |
| Weather variables and lagged observations | 138 | 837 | 450 |

Table B.4: Performances of the multi-output MLR trained on different inputs

## B.2 Gradient Boosting

### B.2.1 Single output model

**Load forecasting** In table B.5, the performances are shown for different inputs. As can be seen, the introduction of the seasonality variables decreases sharply the MAE and the RMSE.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 805 | 1343 | 584 |
| Weather variables and lagged observations | 479 | 933 | 1140 |
| Weather and seasonality variables | 643 | 1114 | 700 |

Table B.5: Performances of the Single output gradient boosting trained on different inputs for forecasting the load

**PV forecasting** In this section, another variation of the input is introduced. This variation is based on the usage of inputs composed of the temperature, solar irradiation, the solar irradiation squared, and the dummy variables for the seasonality. We can observe that this setting produced the best performances for the model.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 97 | 203 | 580 |
| Weather variables and lagged observations | 46 | 112 | 840 |
| Weather and seasonality variables | 98 | 202 | 650 |
| Weather variables (specific to PV) | 90 | 194 | 390 |
| Weather (specific to PV) and seasonality variables | 89 | 187 | 390 |

Table B.6: Performances of the Single output gradient boosting trained on different inputs for forecasting the PV production

### B.2.2 Multi-output model

**Load forecasting** The inputs selected will be the weather and the dummy variables since it provides better performances compared to the model using only weather variables.

| Input Features | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| Weather variables | 719 | 1219 | 25 |
| Weather and seasonality variables | 687 | 1166 | 31 |
| Weather and lagged variables | 283 | 600 | 46 |

Table B.7: Performances of the multi-output gradient boosting trained on different inputs

**PV forecasting** The usage of the inputs based on the weather specific to the PV provides the best performances.

| Input Features | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| Weather variables | 100 | 214 | 24 |
| Weather and seasonality variables | 100 | 214 | 24 |
| Weather and lagged variables | 24 | 55 | 34 |
| Weather (specific to PV) | 90 | 195 | 12 |

Table B.8: Performances of the multi-output gradient boosting trained on different inputs features

## B.3 MultiLayer Perceptron

In order to train the multilayer perceptron, another variation of the inputs has been tested. In these new inputs, the dummy variables are encoded as binary variables.

### B.3.1 Single output model

**Load forecasting** Here, it can be observed that the model provides better performances when the weather variables and the dummy variables encoded as binary values are used. The training time with these inputs is also the longest.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 784 | 1315 | 630 |
| Weather and seasonality variables | 784 | 1315 | 690 |
| Weather variables and lagged observations | 710 | 1175 | 710 |
| Weather and seasonality variables (binary) | 773 | 1268 | 930 |

Table B.9: Performance of the single MLP for the forecasting of the load with different selection of input features

**PV forecasting** The model which has been selected is the model using the weather and dummy variables expressed as binary values. However, the model that uses only the weather information specific to the PV and the dummy variables expressed as binary values provides similar performances.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 122 | 234 | 540 |
| Weather variables and lagged observations | 50 | 118 | 980 |
| Weather and seasonality variables | 116 | 201 | 650 |
| Weather variables (specific to PV) | 141 | 212 | 830 |
| Weather and seasonality variables (binary) | 89 | 189 | 1170 |
| Weather (specific to PV) and seasonality variables | 119 | 205 | 1300 |
| Weather (specific to PV) and seasonality variables (binary) | 92 | 184 | 1610 |

Table B.10: Performance of the single output MLP for the forecasting of the PV with different selection of input features

### B.3.2 Multi-output model

**Load forecasting** In terms of MAE, the model that provides the best performances is the model using as input the weather and the dummy variables expressed as binary values. In terms of RMSE, the best performances are obtained with the model using weather and dummy variables expressed as integers. This second model has also a smaller training time. Thus, these later inputs were selected for the rest of the study.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 673 | 1186 | 539 |
| Weather variables and lagged observations | 673 | 1188 | 1018 |
| Weather and seasonality variables | 669 | 1185 | 670 |
| Weather and seasonality variables (binary) | 665 | 1190 | 2800 |

Table B.11: Performance of the multioutput MLP for the forecasting of the load with different selection of input features

**PV forecasting** The two choices of inputs that provide the best performances are the weather information and dummy variables expressed as binary values and the

weather information specific to the PV and dummy variables expressed as binary values. In practice, the second choice was used for the rest of the study.

| Input Features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|
| Weather variables | 136 | 262 | 530 |
| Weather variables and lagged observations | 160 | 633 | 1030 |
| Weather and seasonality variables | 134 | 257 | 643 |
| Weather and seasonality variables (binary) | 132 | 254 | 1350 |
| Weather (specific to PV) | 134 | 255 | 830 |
| Weather (specific to PV) and seasonality variables | 134 | 255 | 1376 |
| Weather (specific to PV) and seasonality variables (binary) | 132 | 251 | 1800 |

Table B.12: Performance of the multioutput MLP for the forecasting of the PV with different selection of input features

## B.4 Gaussian process regressor

### B.4.1 Single output model

With the single output model, the Min-Max scaler has been removed and the normalized_y argument from the gaussian process has been set to True.

**Load forecasting** The default kernel used in this experiment is the Rational Quadratic kernel. As can be seen in table B.13, the best performances are obtained when the input features are simply composed of the weather variables.

| Input Features | CRPS (W) | mean training time (s) |
|---|---|---|
| Weather variables | 830 | 136 |
| Weather and seasonality variables | 843 | 84 |
| Weather and lagged observations | 965 | 65 |

Table B.13: Performance of the multioutput GBR for the forecasting of the Load with different selection of input features

**PV forecasting** The default kernel used in this experiment is the Matern kernel. The best performances are obtained by using only the weather variables as input features.

| Input Features | CRPS (W) | mean training time (s) |
|---|---|---|
| Weather variables | 219 | 62 |
| Weather and seasonality variables | 230 | 56 |
| Weather variables (specific to PV) | NaN | 8 |
| Weather(specific to PV) and seasonality variables | 263 | 8 |
| Weather and lagged observations | 202 | 62 |

Table B.14: Performance of the multioutput GBR for the forecasting of the PV with different selection of input features

### B.4.2 Multi-output model

**Load forecasting** The best input features are the weather variables combined with the seasonality variables.

| Input Features | CRPS (W) | mean training time (s) |
|---|---|---|
| Weather variables | 749 | 0.18 |
| Weather and seasonality variables | 743 | 0.17 |
| Weather and lagged observations | 788 | 0.15 |

Table B.15: Performance of the multioutput GBR for the forecasting of the Load with different selection of input features

**PV forecasting** The input features leading to the lowest CRPS are the weather variables and the weather variables used with seasonality variables. The input features selected for the rest of the experiments are the weather variables because it has a smaller training time.

| Input Features | CRPS (W) | mean training time (s) |
|---|---|---|
| Weather variables | 205 | 0.13 |
| Weather and seasonality variables | 205 | 0.20 |
| Weather variables (specific to PV) | 217 | 0.17 |
| Weather(specific to PV) and seasonality variables | 217 | 0.20 |
| Weather and lagged observations | 210 | 0.16 |

Table B.16: Performance of the multioutput GBR for the forecasting of the PV with different selection of input features

## B.5 Summary of the results

### Load forecasting

| Model | Output mode | Input features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|---|
| MLR | single | WS | 763 | 1219 | 10 |
| | multi | WS | 1084 | 1821 | 560 |
| GBR | single | WS | 643 | 1114 | 700 |
| | multi | WS | 687 | 1166 | 31000 |
| MLP | single | WSb | 773 | 1268 | 930 |
| | multi | WS | 669 | 1185 | 670 |

Table B.17: Performances of the method with their best input features used for the forecasting of the load. "WS" represents the input made of weather and seasonality variables and "WSb" represents the input made of weather and seasonality variables expressed as binary values

In the table B.18, the performances of the gaussian process regressor using simple or multi-output are shown. The CRPS obtained with a multi-output GPR is lower than the one obtained with the single output model. It can also be noted that the training time is relatively important when using the single output model.

| Model | output | Input Features | CRPS (W) | MeanTraining time (s) |
|---|---|---|---|---|
| GPR | single | Weather variables | 830 | 136.0 |
| | multi | Weather and seasonality variables | 743 | 0.2 |

Table B.18: Performances of the Gaussian Process with their best input features used.

### PV forecasting

| Model | Output mode | Input features | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|---|
| MLR | single | PV weather | 104 | 192 | 2 |
| | multi | PV weather | 127 | 268 | 370 |
| GBR | single | PV weather and seasonality | 89 | 187 | 390 |
| | multi | PV weather | 90 | 195 | 12.3k |
| MLP | single | Weather and hour | 89 | 189 | 1.2k |
| | multi | PV weather, seasonality (binary) | 132 | 251 | 1.8k |

Table B.19: Performances of the method with their best input features used for the forecasting of the PV.

In the table B.20, the performances of the gaussian process regressor are shown for forecasting the PV production. The best performances are obtained with the multi-output model.

| Model | output | Input Features | CRPS (W) | MeanTraining time (s) |
|---|---|---|---|---|
| GPR | single | Weather variables | 219 | 61.6 |
| | multi | Weather and seasonality variables | 205 | 0.2 |

Table B.20: Performances of the Gaussian Process with their best input features used for the forecasting of the PV.

## B.6 Conclusion

In this chapter, the impact of the selection of the input features has been analyzed with different artificial intelligence-based models to forecast the load and the PV. In the next chapter, the hyperparameters of the models will be tuned in order to optimize their performances.

# Appendix C

# Hyperparameter optimization

## C.1 Naive forecaster

In order to have a reference for comparison, the naive forecaster is created and its performances are computed on the validation period (table C.1).

| Forecast | MAE (W) | RMSE (W) |
|---|---|---|
| PV | 104 | 238 |
| load | 776 | 1458 |

Table C.1: Performances of the naive forecaster

## C.2 Exponential smoothing techniques

**Load forecasting** The model with the lowest MAE is the model with no trend and a multiplicative seasonality in which the seasonal periods are composed of 7 days. The model with the lower RMSE is also the model with no trend and a multiplicative seasonality but the seasonal period is only 1 day. It can be observed that the results of both models in terms of MAE and RMSE are better compared to the naive forecaster. The training time is also reasonable. The results of both models are summarized in tableC.2.

| Seasonal period (days) | Mean training time (s) | MAE (W) | RMSE (W) |
|---|---|---|---|
| 7 | 0.52 | 648 | 1239 |
| 1 | 0.43 | 744 | 1214 |

Table C.2: Performance of the models with no trend and a multiplicative seasonality for forecasting the load

**PV forecasting** Since the PV production is intermittent, the exponential smoothing techniques with multiplicative components for the trend or seasonality cannot be tested. By eliminating these models, the best model is the simple model with neither trend nor seasonality which gives the performances summarized in table C.3. However, it is the performances of the model that predicts 0 at each time step and it is thus not relevant to use it in the remaining of the experiments. ARIMA models

are also inaccurate models for PV forecasting because of the same reason.

| Mean training time (s) | MAE (W) | RMSE (W) |
|---|---|---|
| 0.024 | 154 | 340 |

Table C.3: Performance of the models with no trend and a multiplicative seasonality for forecasting the PV

## C.3 Single Output Gradient Boosting Regression

### C.3.1 Impact of the maximal depth

**Load forecasting** The MAE and the RMSE are minimal with a maximal depth equal to 7. With this depth, the MAE equals 607 and the RMSE is 1095. When the max depth is greater than this value, the performances tend to decrease as the model starts to overfit the data.

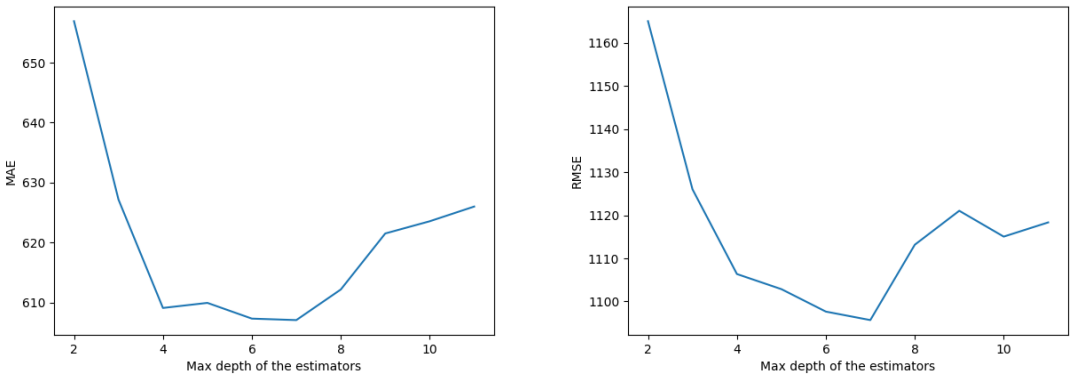



Figure C.1: Performances of the GBR with different max depths for the forecasting of the load

**PV forecasting** The maximal depth of trees that gives the best performances is 4, the MAE equals 88 and the RMSE equals 182. Over this value, the performances decrease as can be seen in figure C.1.

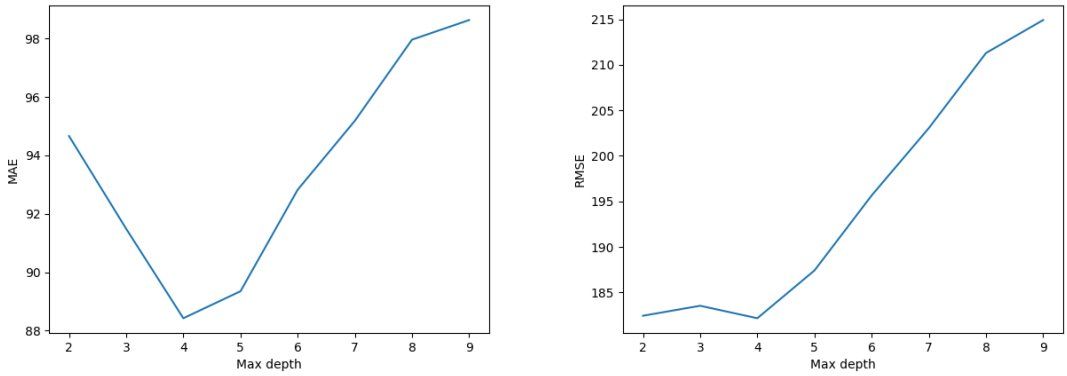



Figure C.2: Performances of the GBR with different max depths for the forecasting of the PV

### C.3.2 Impact of the number of estimators

**Load forecasting** The best performances are reached when using 19 estimators, the MAE then equals 607 W and the RMSE equals 1095. Above 20 estimators, the model overfits the input data, and the MAE and RMSE increase. The performances in terms of the number of estimators can be found in figure C.3.

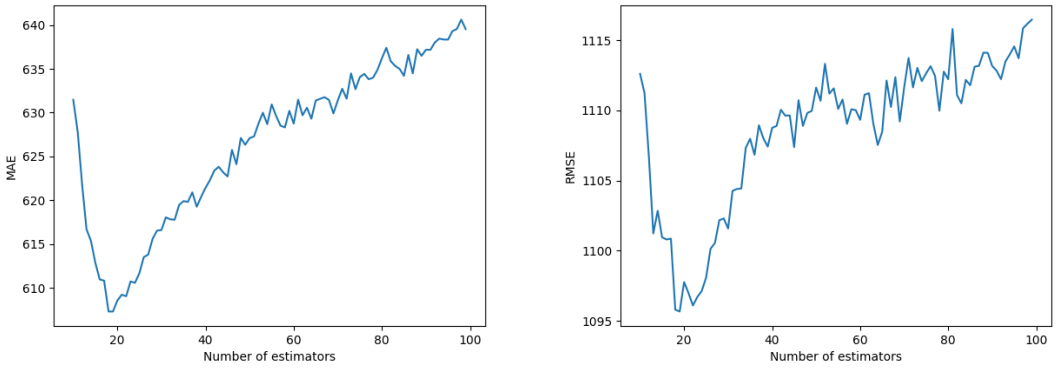



Figure C.3: Performances of the GBR with different number of estimators for the forecasting of the load

**PV forecasting** In this case, the MAE and the RMSE are not minimized simultaneously for the same number of estimators. The MAE is minimal when using 70 estimators and reaches 87.6 W while the RMSE is minimal when using 20 estimators and reaches 172W. To find a compromise between the selection of these two values, the choice of the selection of 40 estimators was done. With this value, the MAE equals 89 W and the RMSE equals 176 W.The performances in terms of the number of estimators can be found in the figure C.4.

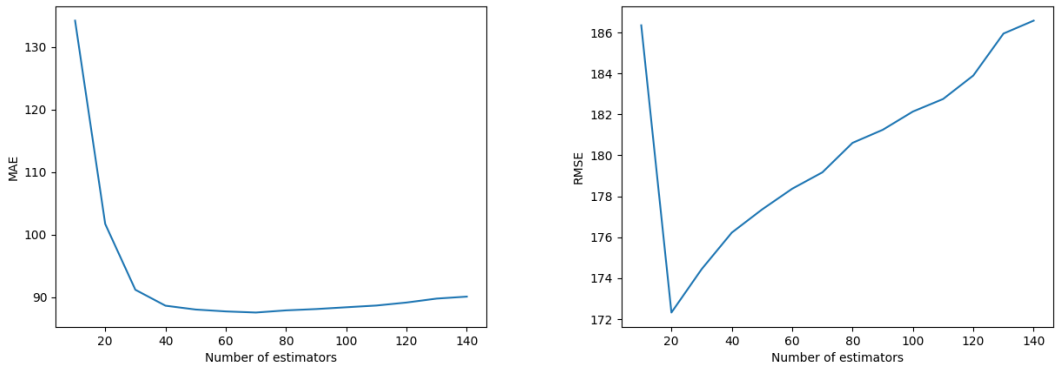



Figure C.4: Performances of the GBR with different number of estimators for the forecasting of the PV

### C.3.3 Impact of the minimum number of samples per split

**Load forecasting** The minimum RMSE is obtained with a minimum number of splits of 7 and is equal to 1094W. The MAE is minimal for a minimum number of split of 4 and is then 607W. The selected minimum number of samples per splits for the other experiments is 5.

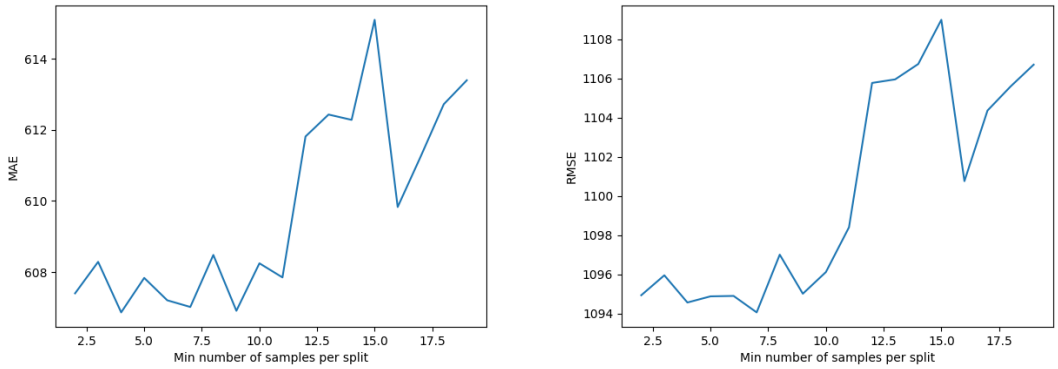



Figure C.5: Performances of the GBR with different minimum number of samples per split for the forecast of the load

**PV forecasting** The number of samples per split does not affect the MAE and RMSE. The default value of 5 is thus kept in the rest of the experiments.

### C.3.4 Impact of the minimum number of samples per leaf

**Load forecasting** The MAE is minimal with a minimum number of samples per leaf equal to 51 and it is then equal to 589W. With a minimum number of samples per split which is equal to 52, the RMSE is minimal and is equal to 1060. Thus, in the rest of the experiment, the minimum number of samples per leaf was chosen to be 51.

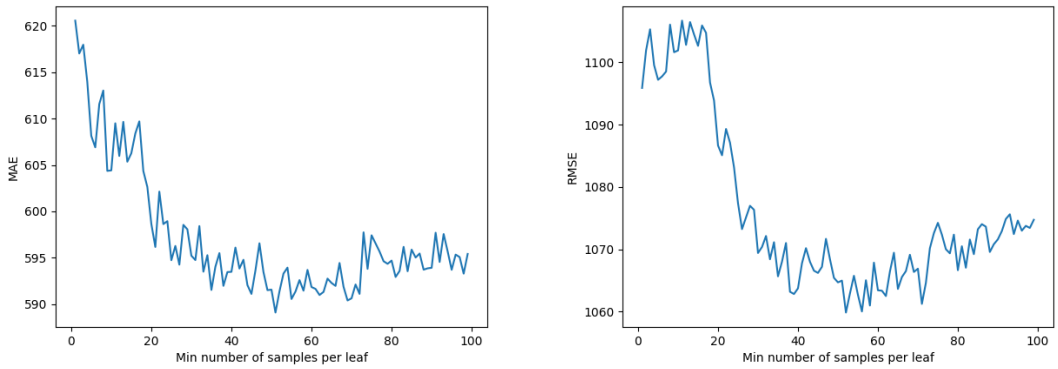



Figure C.6: Performances of the GBR with different minimum number of samples per leaf for forecasting the load

**PV forecasting** The number of samples per leaf does not affect much the MAE and RMSE. Like in the case of the selection of the minimum number of samples per split, the default value of 5 is selected for the rest of the experiments.

### C.3.5 Impact of the learning rate

**Load forecasting** In terms of MAE, the best performances are observed with a learning rate of 10%. With this value, the MAE is equal to 591 W and the RMSE is equal to 1064 W. In terms of RMSE, the best performances are observed with a learning rate of 20%. With this value, the MAE is equal to 595 and the RMSE is equal to 1058W. Above those values, the performances decrease because the magnitude of the change in the update of the model is too large. The selected learning rate for the rest of the experiments is the learning rate of 10%. The performances as a function of the learning rate can be viewed in figure C.7.

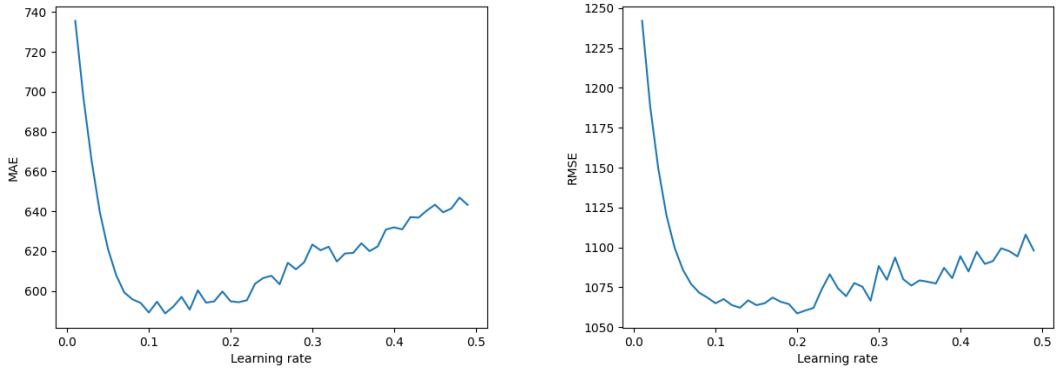



Figure C.7: Performances of the GBR with max depth of 7, 19 estimators, mininum number of samples per split of 5, min number of samples per lead of 51 and different learning rates for the forecast of the load

**PV forecasting** The lowest RMSE is obtained with a learning rate of 0.05 and reaches 172 W. With a learning rate of 0.05, the MAE equals 103 W. When using a learning rate of 0.1, there is a best compromise between the MAE and the RMSE

because the MAE equals 89 W and the RMSE equals 176 W. It is the selected learning rate for the rest of the experiments. The performances as a function of the learning rate can be viewed in figure C.8.

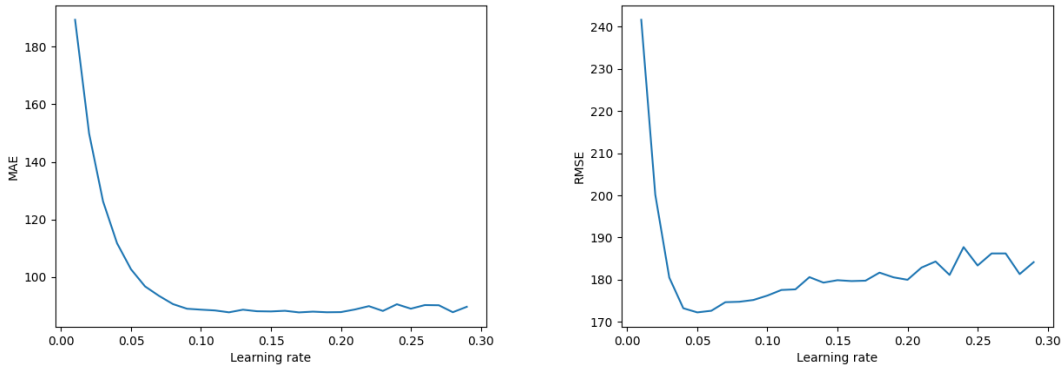



Figure C.8: Performances of the GBR with max depth of 4, 40 estimators, mininum number of samples per split of 5, min number of samples per lead of 5 and different learning rates for the forecast of the PV

## C.4 Multiple Output Gradient Boosting Regression

### C.4.1 Impact of the number of estimators

**Load forecasting** With 12 estimators, the MAE equals 640 W and the RMSE equals 1137 W. Above this value, both the MAE and the RMSE continuously increase as can be viewed in picture C.9 where the number of estimators are tested in [10,140] by step of 10 estimators.

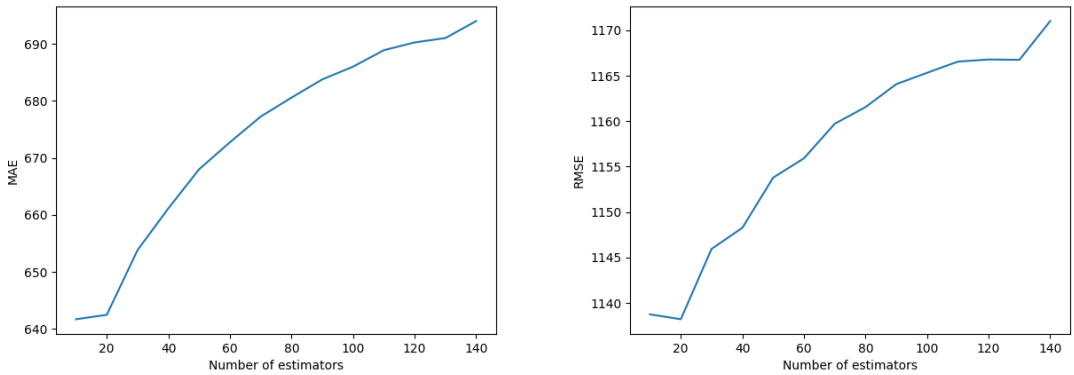



Figure C.9: Performances of the multi-output GBR with different number of estimators for the prediction of the load

**PV forecasting** In picture C.10, the number of estimators are also tested in [10,140] by step of 10 estimators. Below and above 30 estimators the performances tend to decrease caused by underfitting or overfitting of the data. When testing the

number of estimators in the range ]20,40[, the best number of estimators seems to be 29 and produced a MAE equal to 88 W and a RMSE equal to 188 W.

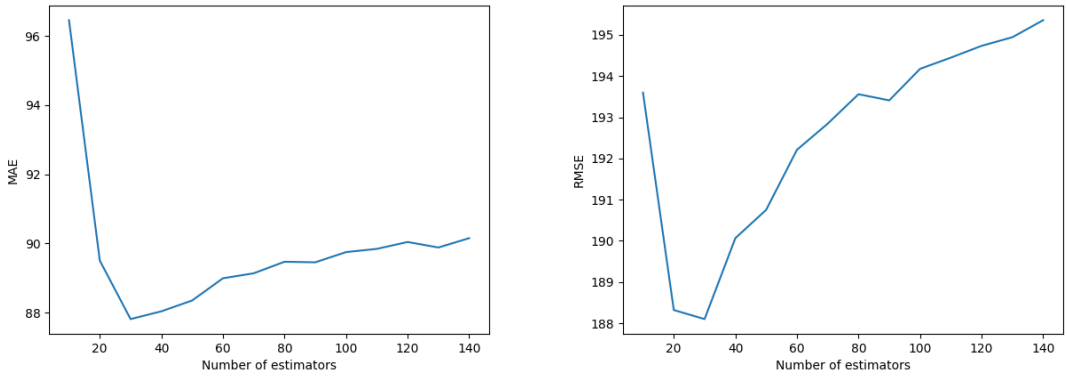



Figure C.10: Performances of the multi output GBR with different number of estimators for the prediction of the PV

### C.4.2 Impact of the maximal depth of trees

**Load and PV forecasting** The maximal depth of trees does not impact significantly the performances. We can thus select a maximal depth of 5.

### C.4.3 Impact of the minimum number of samples per leaf

**Load forecasting** The minimum number of samples per leaf that gives the best performances is 10 and gave a MAE of 629 W and a RMSE of 1124 W. When the minimum number of samples per leaf passes 18, the performances reach a plateau as the number of inputs (35 days of data) prevents the model from being able to divide the samples into several groups composed of at least 18 samples. The performances of the model as a function of the minimum number of samples per leaf can be seen in picture C.11.

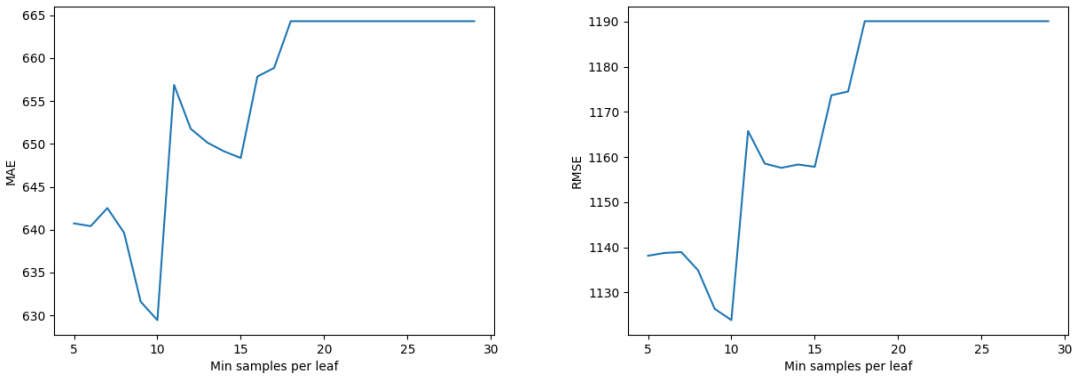



Figure C.11: Performances of the multi output GBR with different minimum number of samples per leaf for the prediction of the load

**PV forecasting** The minimum number of samples per leaf that gives the best performances is 10 and gave a MAE of 83 W and a RMSE of 177 W. Like in the case of the forecast of the load, above a value of 17, the performances reaches a plateau.

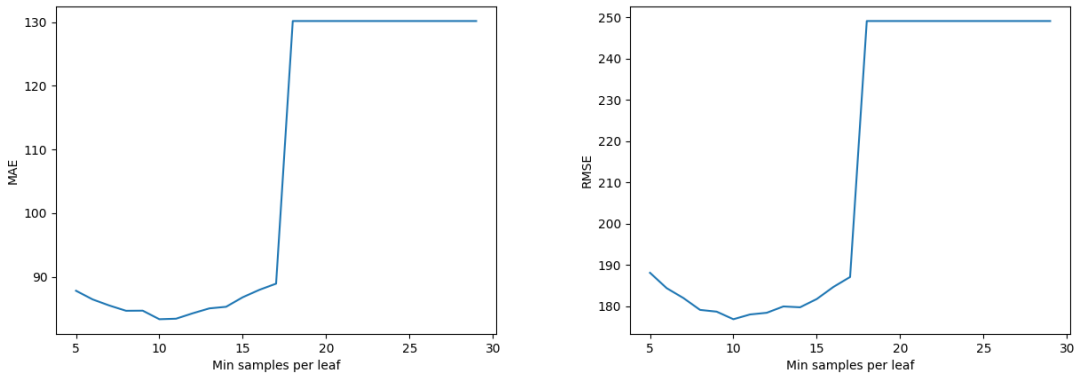



Figure C.12: Performances of the multi output GBR with different minimum number of samples per leaf for the prediction of the PV

### C.4.4 Impact of the minimum number of samples per split

**Load and PV forecasting** The minimum number of samples per split does not affect significantly the performances of the model. A value superior to 35 should not be selected because it is the number of training samples. The default value of minimum 5 samples per split can be selected.

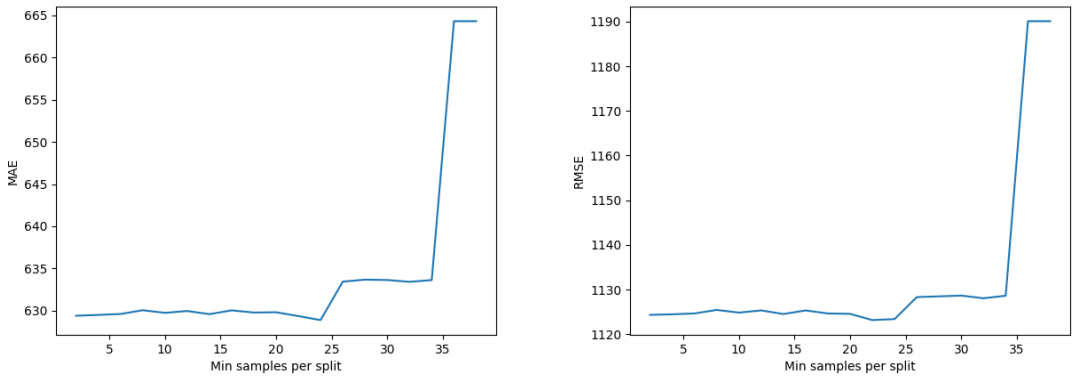



Figure C.13: Performances of the multi output GBR with different minimum number of samples per split for the prediction of the load

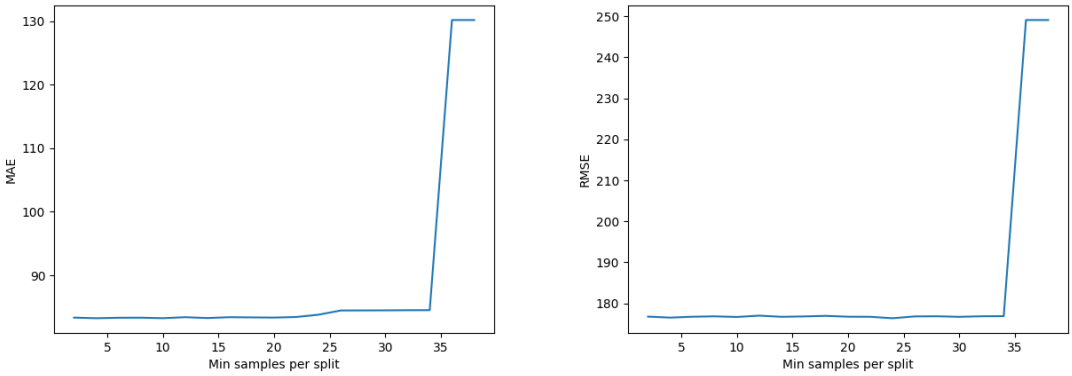



Figure C.14: Performances of the multi output GBR with different minimum number of samples per split for the prediction of the PV

### C.4.5 Impact of the learning rate

**Load forecasting** The performances of the model as a function of the learning rate have a parabolic shape with a minimum reached when the learning rate is of 15% (see figure C.15). With this value, the MAE equals 628 W and the RMSE equals 1122 W.

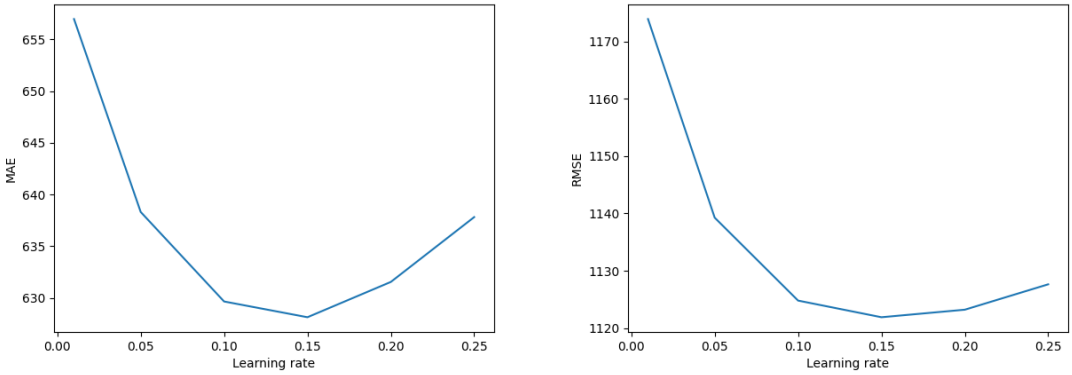



Figure C.15: Performances of the multi output GBR with different learning raes for the prediction of the load

**PV forecasting** A learning rate of 0.1 minimizes the MAE and the RMSE as can be seen in figure C.16.

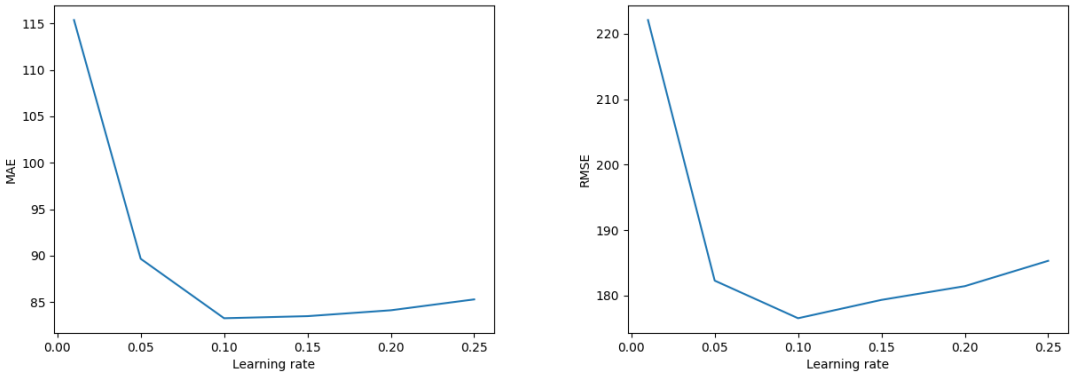



Figure C.16: Performances of the multi output GBR with different learning rates for the prediction of the PV

## C.5 MLP single output

### C.5.1 Variation of the normalization/scaling

**Load forecasting** As can be seen in C.4, the scalers that leads to the best performances in terms of RMSE is the min-max scaler in input and the standard scaler in output (table C.4). We can observe that the usage of standard scalers both on the inputs and on the outputs produces very bad performances and the usage of the min-max scaler on both the inputs and the outputs provides quite bad performances as well.

| Scaler input | Scaler output | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|
| StandardScaler | StandardScaler | 1196 | 1431 | 740 |
| MinMaxScaler | StandardScaler | 712 | 1128 | 1430 |
| StandardScaler | MinMaxScaler | 651 | 1172 | 1100 |
| MinMaxScaler | MinMaxScaler | 767 | 1334 | 1220 |

Table C.4: Performances of the MLP used with different scalers for the forecasting of the load

**PV forecasting** The scaler that gives the best performances for the PV forecasting is the min-max scaler for the output and the standard scaler for the input (table C.5). All the other combinations of scalers provide relatively bad performances.

| Scaler input | Scaler output | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|
| StandardScaler | StandardScaler | 350 | 390 | 540 |
| MinMaxScaler | StandardScaler | 247 | 288 | 650 |
| StandardScaler | MinMaxScaler | 84 | 175 | 830 |
| MinMaxScaler | MinMaxScaler | 237 | 307 | 1000 |

Table C.5: Performances of the MLP used with different scalers

### C.5.2 Variation of the number of neurons in the hidden layer

**Load forecasting** With 80 neurons in the hidden layer, the MAE is equal to 709 W and the RMSE is equal to 1112W.

**PV forecasting** With 35 neurons, there is a good compromise between the performances in terms of MAE, RMSE and training. With this value, the MAE is equal to 89 W and the RMSE is equal to 182.

### C.5.3 Variation of the learning rate and number of epochs

Different combinations of numbers of epochs and learning rates have been tested, the number of epochs being tested in the range $[10, 210]$ by step of 10, and the learning rates being tested in $\{0.0005, 0.001, 0.005, 0.01\}$.

**Load forecasting** As the learning rate increases, the mean squared error and the mean absolute error decreases until reaching a learning rate of 0.005 and then it reincreases slightly with a learning rate of 0.01. For a learning rate of 0.005 and 200 epochs, the MAE and the RMSE are minimal.

**PV forecasting** When the learning rate equals 0.0005, the error is more important and then with higher learning rate, the error decreases and it is minimized for the MAE and the RMSE when the number of epochs is 100 and the learning rate is 0.001.

### C.5.4 Dropout rate

**Load forecasting** With a dropout rate of 0.3, the MAE reaches 712 W and the RMSE reaches 1106 W

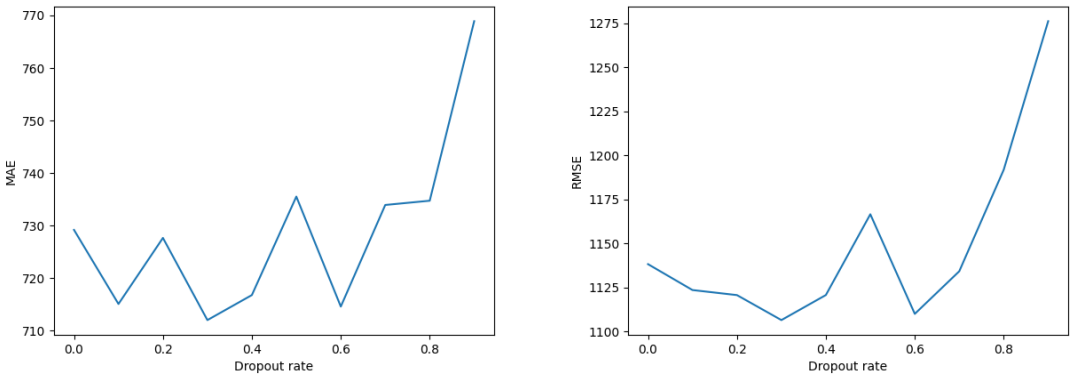



Figure C.17: Performances of the MLP with different dropout rate for forecasting the load

**PV forecasting** With a dropout rate of 50%, the MAE equals 85 W and the RMSE equals 177 W.

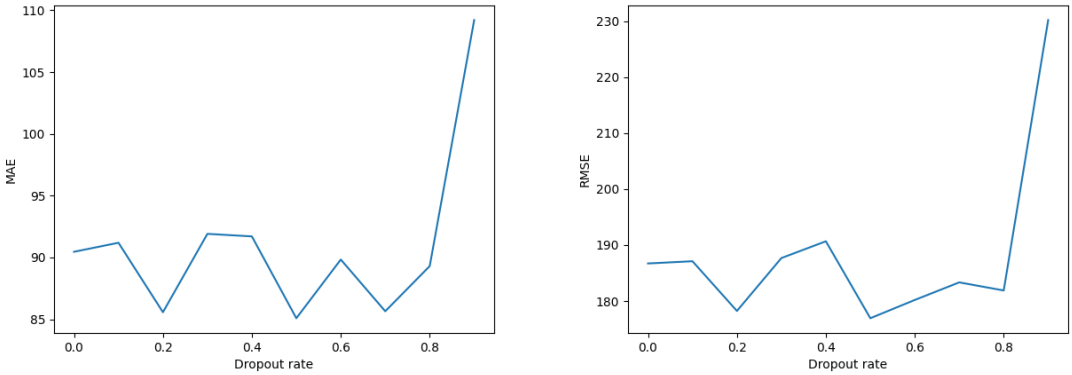



Figure C.18: Performances of the MLP with different dropout rates for forecasting the PV

### C.5.5 Batch size

**Load forecasting** With a batch size of 20, the MAE is equal to 712W and the RMSE is equal to 1118 W.

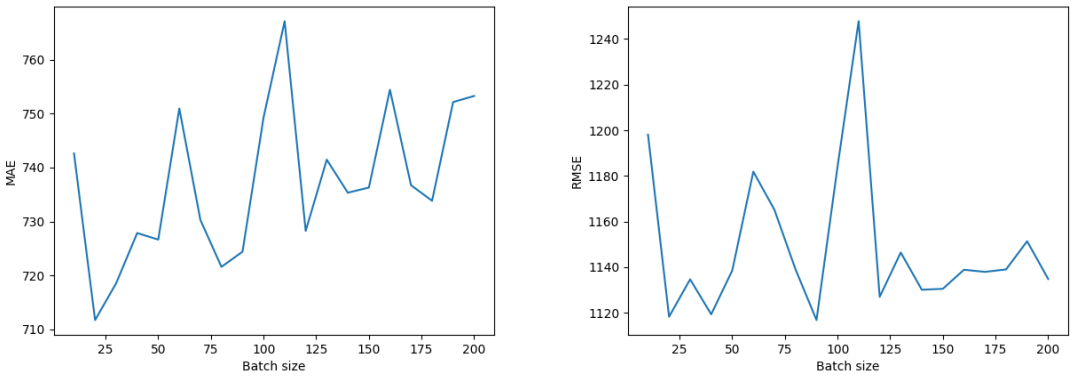



Figure C.19: Performances of the MLP with different batch size for forecasting the load

**PV forecasting** With a batch size of 30, the MAE is equal to 86 and the RMSE is equal to 176.

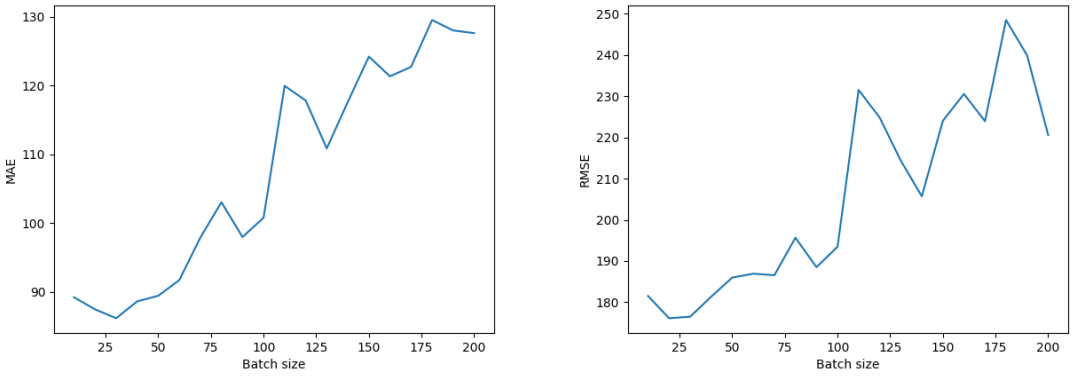



Figure C.20: Performances of the MLP with different batch size for forecasting the PV

### C.5.6 Number of layers

In this experiment, a second hidden layer is added to the structure of the MLP. This layer is simply a dense layer with a number of neurons which is the half of the one of the first hidden layer and an equal dropout rate.

**Load forecasting** The introduction of the second layer leads to an increase in the mean training time but also a decrease in the performances. In the rest of the experiments, the initial structure using only one hidden layer is thus used.

| Number of layers | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| 1 | 714 | 1123 | 0.7 |
| 2 | 746 | 1245 | 1.0 |

Table C.6: Performances of the MLP with different number of layers for forecasting the load

**PV forecasting** In this case, the introduction of a second layer leads to a small decrease of the RMSE and an increase in the MAE. The choice was thus made to keep the original structure with one hidden layer as the improvement in the RMSE was small with 2 layers and it leads to an increase in the training time.

| Number of layers | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| 1 | 89 | 179 | 0.75 |
| 2 | 100 | 176 | 0.96 |

Table C.7: Performances of the MLP with different number of layers for forecasting the PV

### C.5.7 Variation of the activation functions

The models have been tested with different activation functions both for the hidden layer and for the output layer. These activation functions where the linear activation function, the ReLu activation function, the tanh, and the softmax activation function.

**Load forecasting** In the remaining of the experiments, the tanh is used as activation function of the output and the ReLu is used as activation function of the hidden layer.

| Activation output | Activation hidden layer | MAE (W) | RMSE (W) | Mean training period (s) |
|---|---|---|---|---|
| linear | linear | 695 | 1139 | 0.66 |
| relu | linear | 747 | 1152 | 0.78 |
| tanh | linear | 682 | 1140 | 0.98 |
| softmax | linear | 1257 | 1445 | 1.15 |
| linear | relu | 656 | 1116 | 1.39 |
| relu | relu | 710 | 1137 | 1.62 |
| tanh | relu | 650 | 1121 | 1.82 |
| softmax | relu | 1257 | 1445 | 1.94 |
| linear | tanh | 706 | 1163 | 2.17 |
| relu | tanh | 755 | 1165 | 2.34 |
| tanh | tanh | 675 | 1145 | 2.58 |
| softmax | tanh | 1257 | 1445 | 2.79 |
| linear | softmax | 648 | 1165 | 3.04 |
| relu | softmax | 767 | 1222 | 3.30 |
| tanh | softmax | 656 | 1179 | 3.50 |
| softmax | softmax | 1257 | 1445 | 3.59 |

Table C.8: Performances of the model with different activation functions for forecasting the load

**PV forecasting** The best performances were obtained by using the ReLu as activation function for the output and the linear function as activation function for the hidden layer.

| Activation output | Activation hidden layer | MAE (W) | RMSE (W) | Mean training period (s) |
|---|---|---|---|---|
| linear | linear | 120 | 199 | 0.77 |
| relu | linear | 85 | 182 | 0.83 |
| tanh | linear | 121 | 202 | 1.03 |
| softmax | linear | 2379 | 2460 | 1.06 |
| linear | relu | 109 | 193 | 1.25 |
| relu | relu | 89 | 182 | 1.78 |
| tanh | relu | 108 | 195 | 1.81 |
| softmax | relu | 2379 | 2460 | 2.07 |
| linear | tanh | 120 | 194 | 2.12 |
| relu | tanh | 93 | 190 | 2.32 |
| tanh | tanh | 123 | 199 | 2.77 |
| softmax | tanh | 2379 | 2460 | 3.12 |
| linear | softmax | 209 | 272 | 2.90 |
| relu | softmax | 206 | 281 | 3.14 |
| tanh | softmax | 213 | 274 | 3.50 |
| softmax | softmax | 2379 | 2460 | 3.69 |

Table C.9: Performances of the model with different activation functions for forecasting the PV

## C.6 MLP multi output

### C.6.1 Variation of the normalization/scaling

**Load forecasting** In comparison to what was observed with the single output MLP, with the multi-output MLP, the choice of the scaler does not change much the performances. The scaling chosen is the standard scaler for the input and the output.

| Scaler output | Scaler input | MAE (W) | RMSE (W) | mean training time (s) |
|---|---|---|---|---|
| StandardScaler | StandardScaler | 668 | 1184 | 0.79 |
| StandardScaler | MinMaxScaler | 669 | 1185 | 0.93 |
| MinMaxScaler | StandardScaler | 669 | 1183 | 1.08 |
| MinMaxScaler | MinMaxScaler | 669 | 1185 | 1.30 |

Table C.10: Performances of the MLP used with different scalers

**PV forecasting** The choice of the scaler doesn't seem to affect much the performances. The scalers selected are the standard scalers both for the input and for the output.

| Scaler output | Scaler input | MAE (W) | RMSE (W) | mean training (s) time |
|---|---|---|---|---|
| StandardScaler | StandardScaler | 132 | 252 | 0.89 |
| StandardScaler | MinMaxScaler | 132 | 252 | 1.09 |
| MinMaxScaler | StandardScaler | 132 | 251 | 1.47 |
| MinMaxScaler | MinMaxScaler | 133 | 254 | 2.97 |

Table C.11: Performances of the MLP used with different scalers

### C.6.2 Variation of the number of neurons

**Load forecasting** By varying the number of neurons in the hidden layer, we can observe that the MAE remains in the range [670,680] and the RMSE remains in the range [1170,1180]. As a consequence, the number of neurons in the hidden layer was selected to be 2 the number of input neurons by default.

**PV forecasting** By varying the number of neurons in the hidden layer between 100 and 2880, it can be seen that the MAE starts with a value of 130 and then stabilize to 120 whatever the number of neurons. The tendency is the same for the RMSE which is equal to 248 whatever the number of neurons. The hidden layer size was thus again selected to be 2 times the number of input neurons.

### C.6.3 Variation of the Learning rate and of the number of epochs

**Load forecasting** Different combinations of numbers of epochs and learning rates have been tested. The number of epochs has been tested in the range $[10, 190]$ by step of 10 and the learning rates being tested in $\{0.0005, 0.001, 0.0020.005, 0.01\}$. With a learning rate of 0.002, the MAE and RMSE are smaller and with 130 epochs, the MAE reached 663 W and the RMSE reached 1186W.

**PV forecasting** Different combinations of numbers of epochs and learning rates have been tested, the number of epochs being tested in the range $[10, 190]$ by step of 10, and the learning rates being tested in $\{0.0005, 0.001, 0.0020.005, 0.01\}$. The learning rate that provided the better performances is 0.001 and the number of epochs didn't seem to impact very much the performances but it was selected to be 130.

### C.6.4 Variation of the dropout rate

**Load forecasting** The dropout rate was selected to be at 60% as it gives a MAE of 664 W and a RMSE of 1187W.

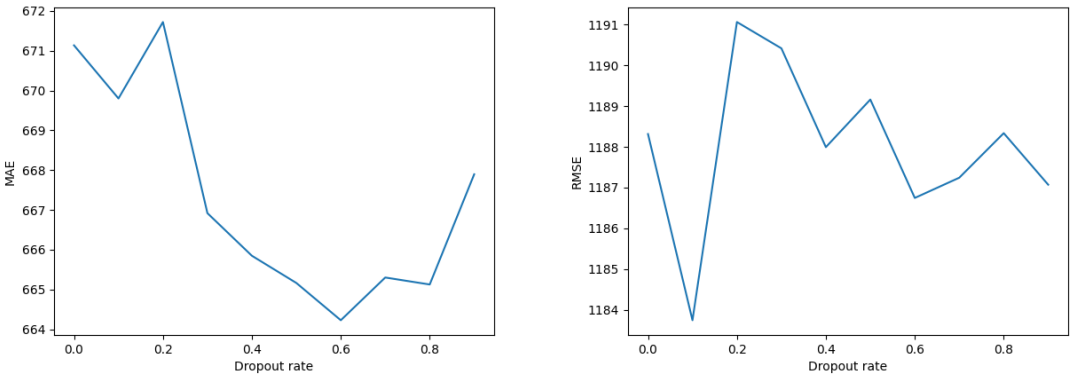



Figure C.21: Performances of the multi output MLP with different dropout rate

**PV forecasting** As can be seen in figure C.22, the choice of the dropout rate does not impact very much the performances but it was selected to be 40% as it is the dropout rate that provides the best performances.

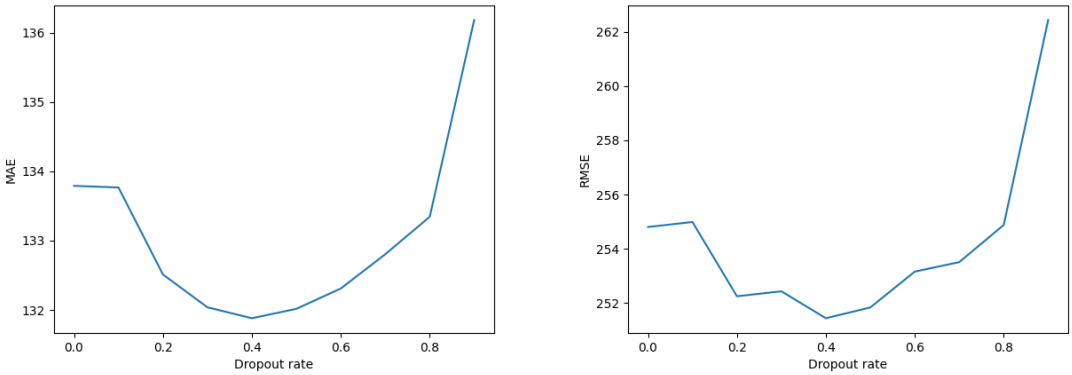



Figure C.22: Performances of the multi output MLP with different dropout rate

### C.6.5 Variation of the batch size

**Load forecasting** The batch size was tested in the range [10,200] with steps of 10 and it showed that independently from the selected batch size, the MAE was always in the range [664,665] and the RMSE was always in the range [1186, 1191]. The batch size selected is 20 and it gave a MAE equal to 664W and a RMSE equal to 1189W.

**PV forecasting** The batch size was tested in the range [10,200] with steps of 10 and it showed that independently from the selected batch size, the MAE was always in the range [131,133] and the RMSE was always in the range [251, 255]. As a consequence, the batch size was simply selected to be 50.

### C.6.6 Variation of the activation functions

**Load forecasting** The activation functions that provided the best performances both in terms of RMSE and MAE are the ReLu activation function both for the

hidden and for the output layer.

| Activation output | Activation hidden layer | MAE (W) | RMSE (W) | Mean training period (s) |
|---|---|---|---|---|
| linear | linear | 2038 | 3225 | 0.59 |
| relu | linear | 667 | 1190 | 0.67 |
| tanh | linear | 806 | 1339 | 0.84 |
| softmax | linear | 666 | 1188 | 1.07 |
| linear | relu | 1003 | 1643 | 1.15 |
| relu | relu | 664 | 1187 | 1.33 |
| tanh | relu | 757 | 1305 | 1.46 |
| softmax | relu | 665 | 1187 | 1.65 |
| linear | tanh | 1378 | 2173 | 1.80 |
| relu | tanh | 665 | 1189 | 1.99 |
| tanh | tanh | 757 | 1293 | 2.13 |
| softmax | tanh | 666 | 1189 | 2.30 |
| linear | softmax | 664 | 1190 | 2.49 |
| relu | softmax | 664 | 1190 | 2.64 |
| tanh | softmax | 664 | 1190 | 2.83 |
| softmax | softmax | 665 | 1187 | 3.02 |

Table C.12: Performances of the multi-output MLP with different activation functions for the forecasting of the load

**PV forecasting** The softmax was selected as activation function of the hidden layer and the ReLu was selected as activation function of the output.

| Activation output | Activation hidden layer | MAE (W) | RMSE (W) | Mean training period (s) |
|---|---|---|---|---|
| linear | linear | 147 | 289 | 1.10 |
| relu | linear | 134 | 255 | 0.79 |
| tanh | linear | 134 | 265 | 1.00 |
| softmax | linear | 131 | 250 | 1.23 |
| linear | relu | 133 | 258 | 1.34 |
| relu | relu | 132 | 253 | 1.53 |
| tanh | relu | 133 | 256 | 1.64 |
| softmax | relu | 131 | 250 | 1.83 |
| linear | tanh | 142 | 278 | 2.44 |
| relu | tanh | 133 | 254 | 2.10 |
| tanh | tanh | 136 | 265 | 2.55 |
| softmax | tanh | 131 | 250 | 2.51 |
| linear | softmax | 130 | 248 | 2.79 |
| relu | softmax | 130 | 248 | 3.00 |
| tanh | softmax | 130 | 248 | 3.10 |
| softmax | softmax | 130 | 250 | 3.89 |

Table C.13: Performances of the multioutput MLP with different activation functions for the forecasting of the PV

### C.6.7 Number of layers

**Load forecasting** As can be seen in table C.14, the usage of only one hidden layer provided slightly better performances and was thus selected.

| Number of layers | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| 1 | 665 | 1188 | 0.57 |
| 2 | 669 | 1189 | 0.86 |

Table C.14: Performances of the MLP with different number of layers for forecasting the load

**PV forecasting** As can be seen in table C.15, the performances in terms of MAE and RMSE are equivalent. As the structure which used one layer leads to shorter training time, it was selected.

| Number of layers | MAE (W) | RMSE (W) | Mean training time (s) |
|---|---|---|---|
| 1 | 129 | 248 | 0.70 |
| 2 | 129 | 248 | 1.14 |

Table C.15: Performances of the MLP with different number of layers for forecasting the load

## C.7 Gaussian process regression

The gaussian process was tested with different kernels and is evaluated in terms of its CRPS score.

### C.7.1 Single output model

With the single output model, the limiting factor is the training time which can be relatively high depending on the kernel selected.

#### Load forecasting

| Kernel | CRPS (W) | Mean training time (s) |
|---|---|---|
| RationalQuadratic | 829 | 151 |
| RBF | 813 | 10 |
| Matern | 813 | 15 |

Table C.16: CRPS in function of the type of kernel for the forecasting of the load

**PV forecasting** The model built with a Matern kernel is better than the other models but the training time is large.

| Kernel | CRPS (W) | Mean training time (s) |
|---|---|---|
| RationalQuadratic | 256 | 95 |
| RBF | 268 | 10 |
| Matern | 219 | 64 |

Table C.17: CRPS in function of the type of kernel for the forecasting of the load

### C.7.2 Multi-output model

In comparison to the single output gaussian regressor, the multi-output model leads to significantly lower training time and it provides also better performances.

**Load forecasting**

| Kernel | CRPS (W) | Mean training time (s) |
|---|---|---|
| RBF | 841 | 0.13 |
| Rational Quadratic kernel | 808 | 0.15 |
| Matern | 825 | 0.15 |
| ExpSineSquared(periodicity = 96) | 895 | 0.08 |
| 1.0 * RBF(1.0) | 743 | 0.16 |
| 1.0 * Matern(length_scale=1, nu=1.5) | 732 | 0.18 |

Table C.18: CRPS in function of the type of kernel for the forecasting of the load

**PV forecasting**

| Kernel | CRPS (W) | Mean training time (s) |
|---|---|---|
| RBF | 218 | 0.09 |
| Rational Quadratic kernel | 266 | 0.16 |
| Matern | 231 | 0.11 |
| ExpSineSquared(periodicity = 96) | 154 | 0.06 |
| Rational Quadratic kernel | 237 | 0.20 |
| 1.0 * RBF(1.0) | 205 | 0.14 |
| 1.0 * Matern(length_scale=1, nu=1.5) | 214 | 0.14 |

Table C.19: CRPS in function of the type of kernel for the forecasting of the load

## C.8 Summary of the results

**Load forecasting** In table C.20, the performances of the different models after tuning of their hyperparameters are provided. The models based on gradient boosting have the lowest MAE and RMSE. Then, the methods based on MLP and EST seem to have also good performances compared to the naive forecaster. The models based on linear regression comparatively have worst performances and in the case of the multiple output linear regression, the performances are worst than the ones of the naive forecaster.

| Model | Output mode | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|
| Naive forecaster | / | 776 | 1458 | / |
| EST | / | 648 | 1239 | 520 |
| MLR | single | 763 | 1219 | 10 |
| | multi | 1084 | 1821 | 560 |
| GBR | single | 591 | 1064 | 150 |
| | multi | 628 | 1122 | 2.1k |
| MLP | single | 650 | 1121 | 1.8k |
| | multi | 664 | 1187 | 1.3k |

Table C.20: Performances of the models for forecasting the load after optimization of their hyperparameters

**PV forecasting** In table C.21, the performances of the different models for forecasting the PV production are summarized. The exponential smoothing not being adapted, its performances are worst than the one of the naive forecaster. As a consequence, this model was not used in the remainder of the experiments. The linear regression model has a lower RMSE than the naive forecaster. The multi-output linear regressor has worst performances than the naive forecaster.

| Model | Output mode | MAE (W) | RMSE (W) | Mean training time (ms) |
|---|---|---|---|---|
| Naive forecaster | / | 104 | 238 | / |
| EST | / | 154 | 340 | 24 |
| MLR | single | 104 | 192 | 2 |
| | multi | 127 | 268 | 370 |
| GBR | single | 103 | 172 | 120 |
| | multi | 89 | 176 | 120 |
| MLP | single | 85 | 182 | 830 |
| | multi | 130 | 248 | 3k |

Table C.21: Performances of the models for forecasting the PV after optimization of their hyperparameters

## C.9 Analysis of a change of resolution

The performances of all of the models trained for forecasting the load consumption are relatively low. These performances are notably due to the inherent characteristics of the load time-series as it has usually a high standard deviation). To view if a decrease in the resolution of the forecasts could improve the performances, a test was performed by using a resolution of one hour rather than a resolution of 15 minutes. To do so, the cross-validation procedure (fig. 4.4) was used similarly to the previous experiments. The linear regression, the gradient boosting regression, and the multilayer perceptron were tested with their optimal hyperparameters and the performances of those models are shown in table C.22. By using this resolution, the performances of the MLR and the GBR are improved by $\sim$ 100 W. It is a

good improvement but the error with this resolution is still relatively high. The performances of the MLP are reduced with this new resolution but it would certainly have required some retuning of the model.

| Model name | outputs | MAE (W) | RMSE (W) | Mean Training Time (ms) |
|---|---|---|---|---|
| MLR | single | 687 | 1045 | 1 |
| | multi | 984 | 1611 | 79 |
| GBR | single | 533 | 912 | 47 |
| | multi | 545 | 928 | 380 |
| MLP | single | 870 | 1445 | 1050 |
| | multi | 777 | 1251 | 69 |

Table C.22: Performances of the model by using a hourly resolution

## C.10 Conclusion

In this chapter, several models were tuned. The models that produced point forecasts were tuned and most of them had good performances except the multiple output linear regression for forecasting the load and the PV and the exponential smoothing technique which was not adapted for forecasting the PV production. About the gaussian regressor which produces probabilistic forecast, it was quite complex to tune and the single output model had the drawback to be very long to train compared to other models.

In the next chapter, the impact of the number of days of data for training the models is studied for the different models

# Appendix D

# Impact of the training set size on performances of the models

## D.1 Introduction

In this chapter, the impact of the training set sizes will be analyzed on the different models. Thanks to these values, an optimal training period can be determined for all forecasting models. This training period is a trade-off between the mean absolute error and the mean squared error. The sizes of the training sets are studied in the range from 1 day of data to 41 days of data.

## D.2 Exponential Smoothing Techniques

**Load forecasting** As can be seen in the figure D.1, the best performances are obtained when the training set size reaches 31 days.

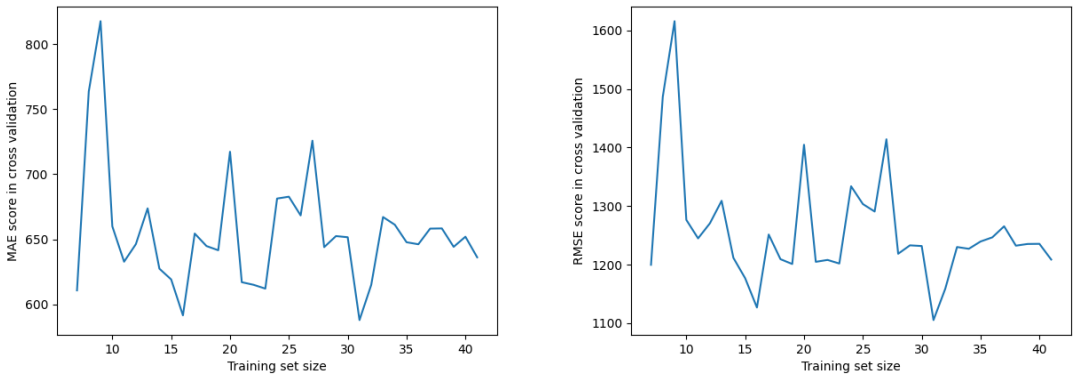



Figure D.1: Performances of the EST with different training set size for the forecasting of the load

## D.3 Linear regression

### D.3.1 Single output model

**Load forecasting** Since the performances are very bad, the figure D.2 contains the graph of the MAE and the RMSE with training set size in [1,42[ and also a zoom in which the training set size are in the range [10,42[. The MAE is minimal for a training period of 32 days. Then, the MAE is equal to 760W and the RMSE is equal to 1218W. The RMSE is minimal for a training period of 25 days and then slightly increases. To make a compromise, the training period is selected to be 32 days in the rest of the experiments.

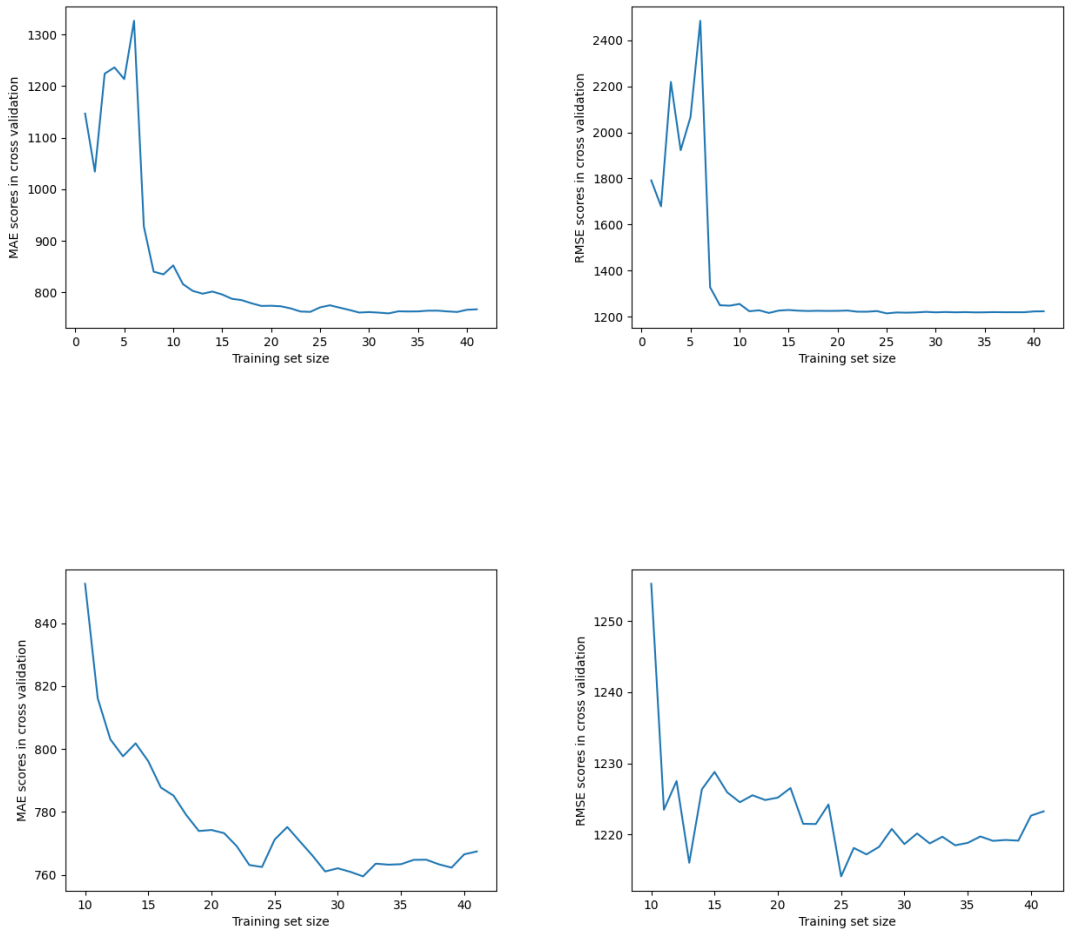



Figure D.2: Performances of the LR with different training set size for the forecasting of the load (zoom)

**PV forecasting** The minimum RMSE is obtained with a training set size of 26 days and equals 184 W. The performances in terms of MAE when the training set size takes values between 19 and 29 days are quite stable ($\sim$ 98 W). The selected training period has thus been chosen to be 26 days.

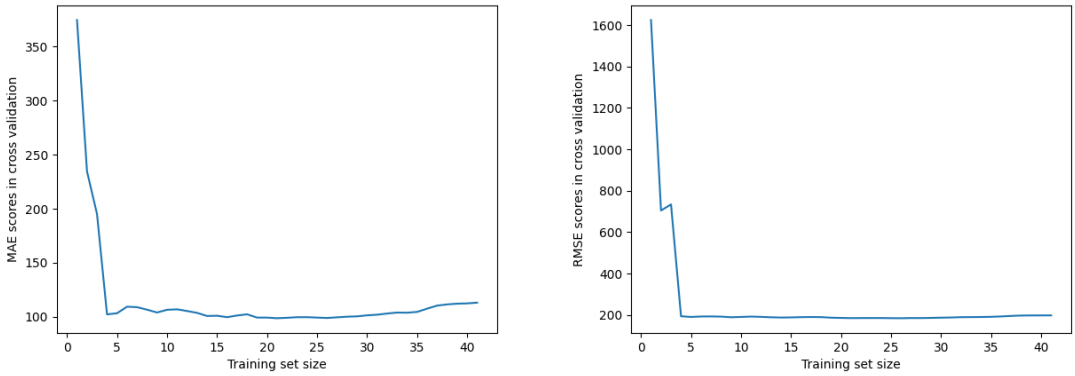



Figure D.3: Performances of the MLR with different training set size for the forecasting of the PV

### D.3.2 Multi-output model

**Load forecasting** The increase in the training set size produces an increase in the MAE and RMSE which reach a peak with 7 days of data and then vary. The performances are relatively bad compared to the simple naive forecaster, it thus shows that it is not a good model.

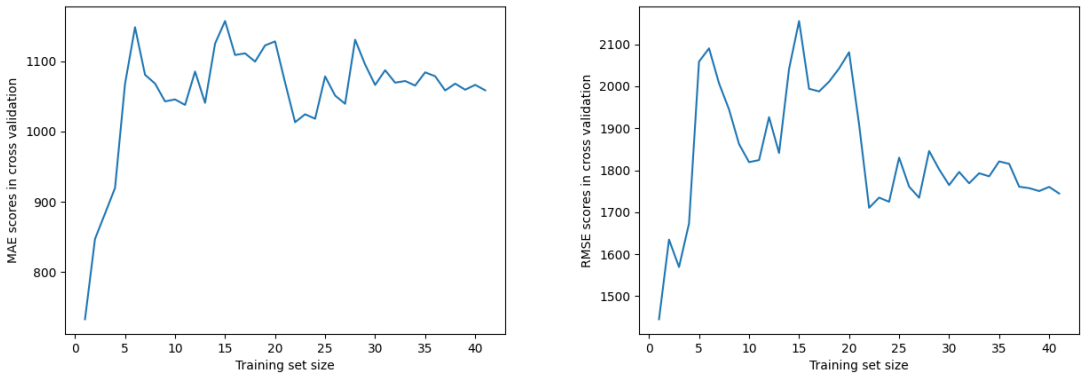



Figure D.4: Performances of the LR with different training set size for the forecasting of the load

**PV forecasting** The MAE and the RMSE are minimized with a training set composed of 6 days of data. Then, the MAE equals 105 and the RMSE equals 241.

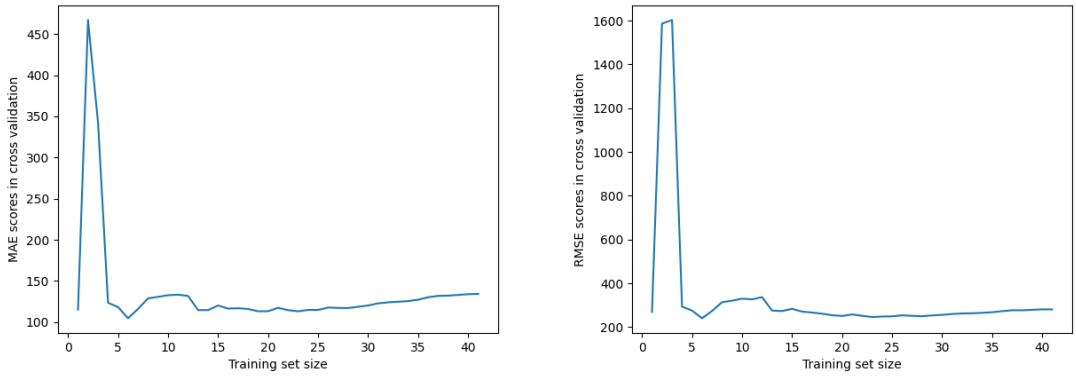



Figure D.5: Performances of the multi output LR with different training set size for the forecasting of the PV

## D.4 Gradient Boosting Regression

### D.4.1 Single output model

**Load forecasting** The MAE is equal to 589W and is minimal for a training period of 32 days. For a training period of 11 days, the RMSE is minimal and equals 1032W. A training set size of 20 days is selected which corresponds to a MAE of 592 W and a RMSE of 1042W.

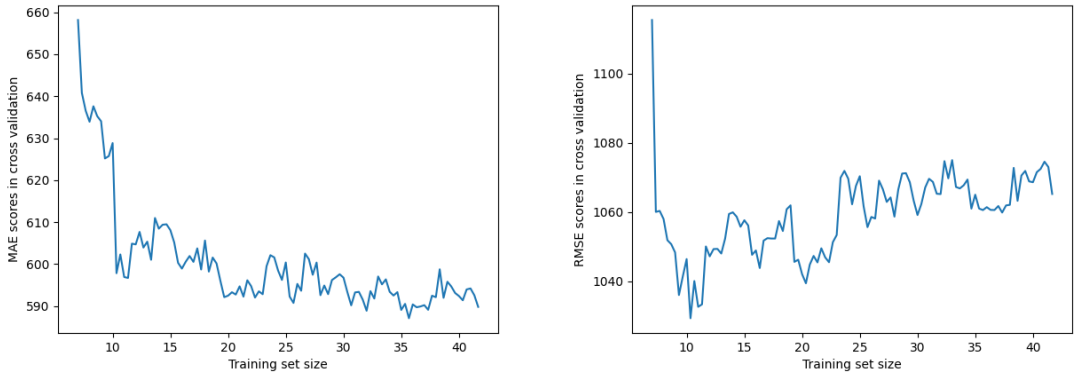



Figure D.6: Performances of the GBR with different training period for the forecast of the load

**PV forecasting** When the training period reaches 18 days, the MAE reaches its minimum and is equal to 79 W. When the training period reaches 21 days, the RMSE reaches its minimum and equals 170 W (while the MAE is equal to 83 W). The training set size was chosen to be 21 days.

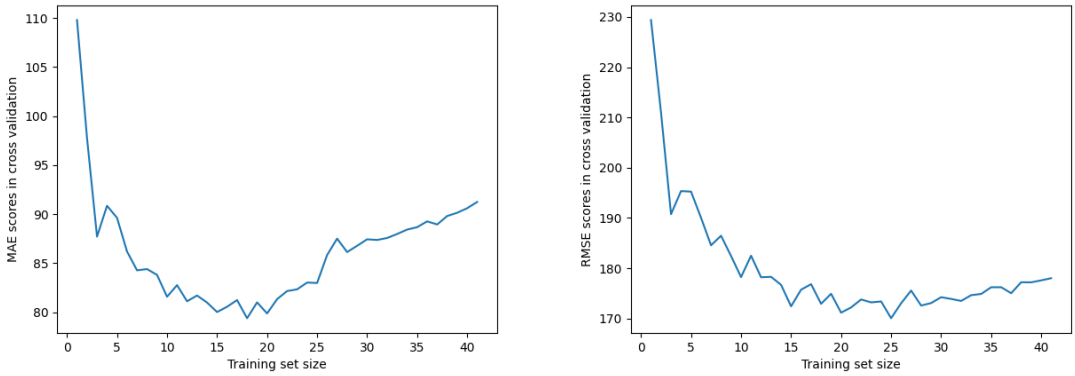



Figure D.7: Performances of the GBR with different training period for the forecast of the PV

### D.4.2 Multi-output model

**Load forecasting** A training period of 23 days minimizes the MAE as well as the RMSE. With this value, the MAE equals 626 W and RMSE equals 1119 W.

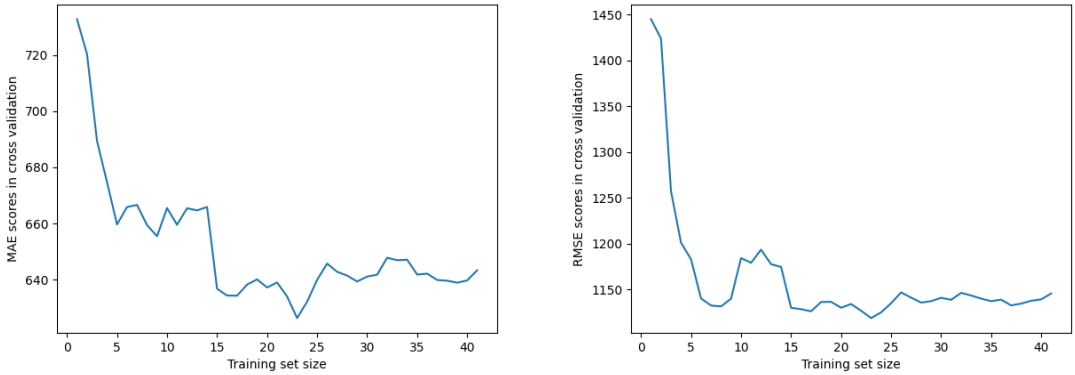



Figure D.8: Performances of the multi output GBR with different training set sizes for the prediction of the PV

**PV forecasting** The MAE reaches a minimum when 11 days of data are included in the training set and the RMSE reached a minimum when 13 days of data are included in the training set. Above this value, the performances of the model decrease. For the remaining of the experiments, a training set size of 13 days was selected. With this value, the MAE equals 78W and the RMSE equals 177W.

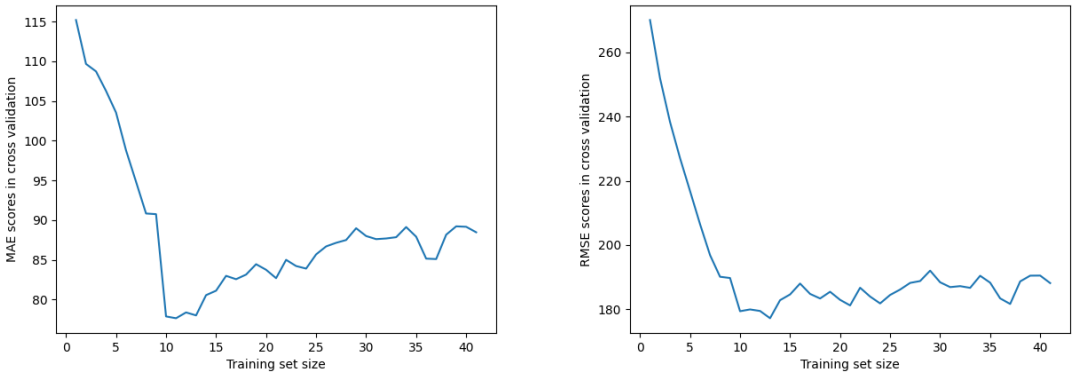



Figure D.9: Performances of the multi output GBR with different training set sizes for the prediction of the PV

## D.5 MultiLayer Perceptron

### D.5.1 Single output model

**Load forecasting** When observing the RMSE as a function of the training set size, we can view that it decreases sharply until being composed of 1 week of data and that above this value, the RMSE restarts to increase slightly. By contrast, the MAE seems to decrease until using 35 days of data and then starts to increase slightly. To have a trade-off between the performances in term of MAE and RMSE, a training set size composed of 26 days of data was selected. With this value, the MAE is equal to 704 W and the RMSE is equal to 1472 W.

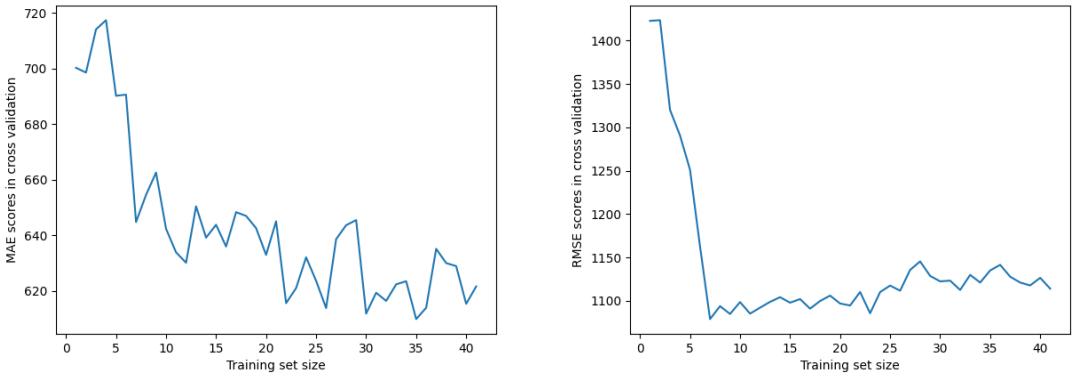



Figure D.10: Performances of the MLP with different training period of the load

**Pv forecasting** Both the MAE and the RMSE are minimized with a training set composed of 22 days of data. With this value, the MAE is equal to 77 W and the RMSE is equal to 168W.

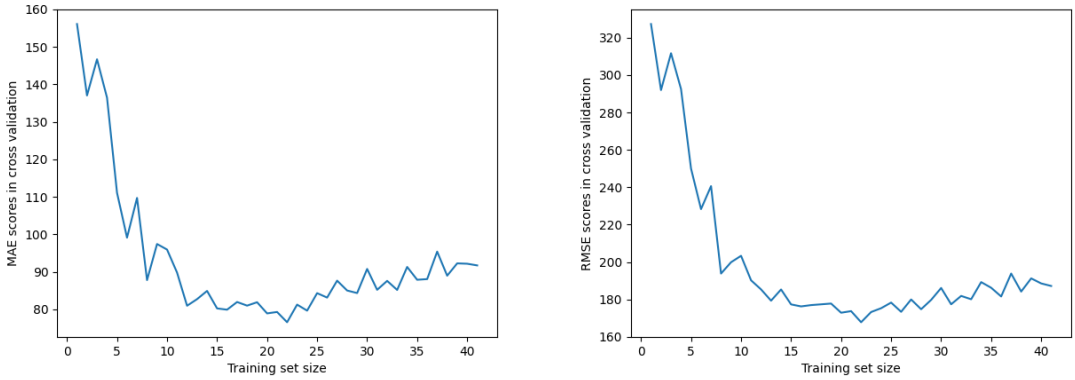



Figure D.11: Performances of the MLP with different training period of the PV

### D.5.2 Multi-output model

**Load forecasting** The error reaches very high values when trained on a small number of days of data. Then, it decreases until reaching a minimum in terms of MAE and RMSE when 22 days of data are used. Above this value, the error restarts to increase. With a training set size of 22 days, the MAE is equal to 656 W and the RMSE is equal to 1152 W.

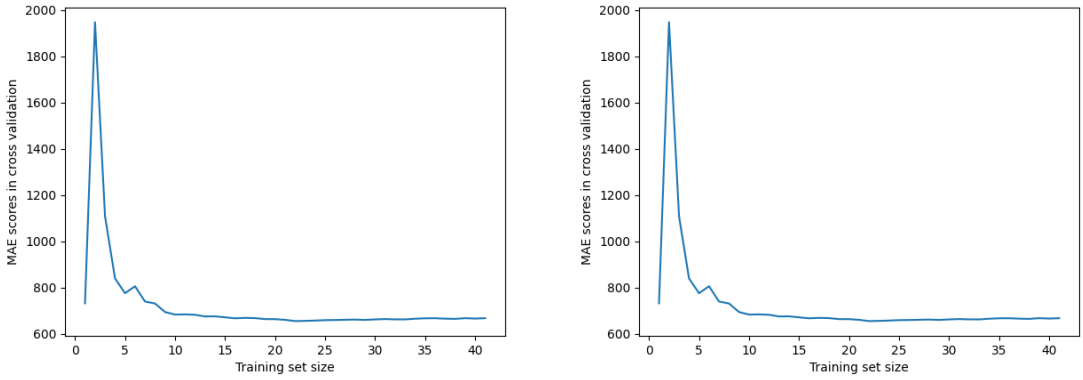



Figure D.12: Performances of the MLP with different training period of the load

**PV forecasting** As can be seen in figure D.12, a training period 10 days yields the lower MAE and RMSE which respectively reach 89 W and 187W.

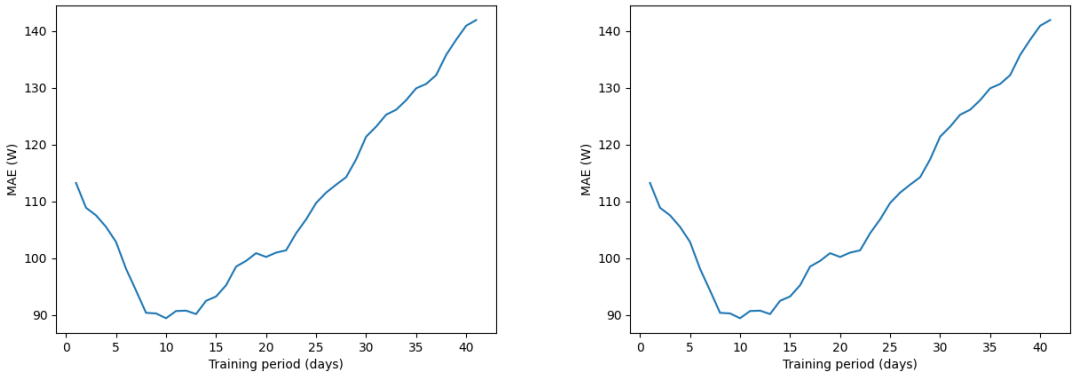



Figure D.13: Performances of the MLP with different training period of the PV

## D.6 Gaussian process regression

### D.6.1 Single output Model

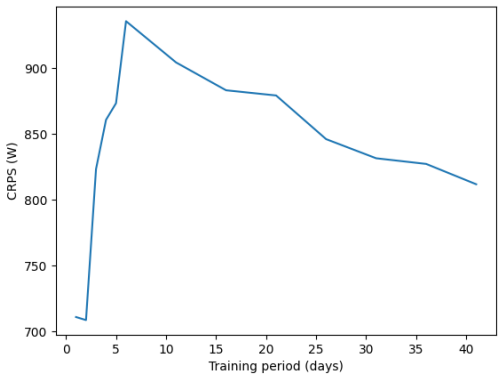



Figure D.14: CRPS for different training set sizes for forecasting of the load

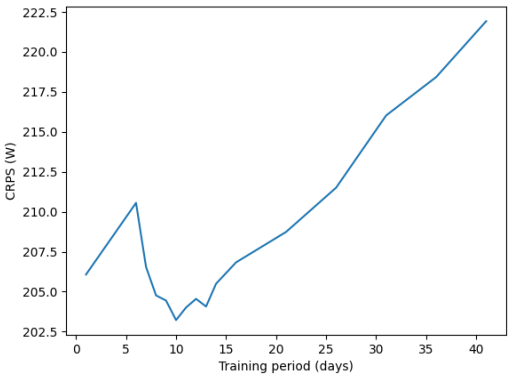



Figure D.15: CRPS for different training set sizes for forecasting of the PV

### D.6.2 Multi-output Model

**Load forecasting** As can be seen in figure D.16, the CRPS reaches a minimum when the training period is composed of 38 days of data.

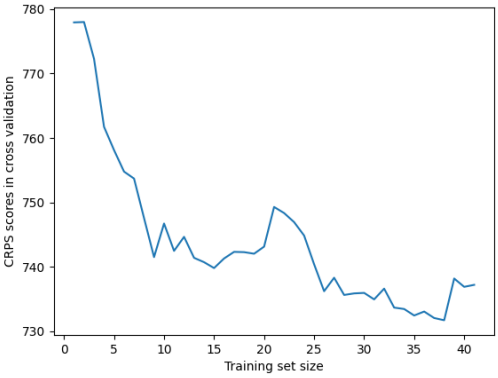



Figure D.16: CRPS multioutput GBR for the forecasting of the load

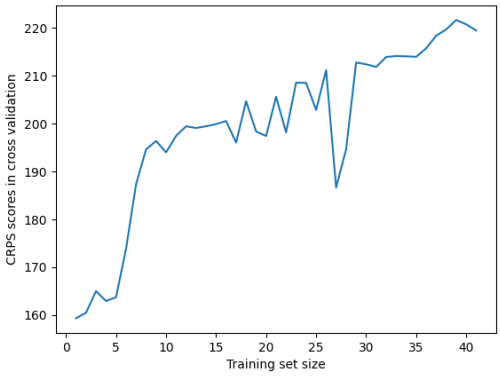



Figure D.17: CRPS multioutput GBR for the forecasting of the PV

## D.7 Summary the results

### Load forecasting

| **Model** | **Output mode** | **TSS (days)** | **MAE (W)** | **RMSE (W)** |
|---|---|---|---|---|
| EST | / | 31 | | |
| MLR | single | 32 | 760 | 1218 |
| | multi | 24 | 1018 | 1724 |
| GBR | single | 20 | 592 | 1042 |
| | multi | 23 | 626 | 1119 |
| MLP | single | 26 | 704 | 1472 |
| | multi | 22 | 656 | 1152 |

Table D.1: Performances of the models in cross validation for forecasting the load with respect to the training period selected

| Model | outputs | Training period (days) | CRPS (W) |
|---|---|---|---|
| GPR | single | 41 | 812 |
| | multi | 38 | 732 |

Table D.2: Performances of the GPR models in cross validation for forecasting the load with respect to the training period selected

**PV forecasting**

| **Model** | **Output mode** | **TSS (days)** | **MAE (W)** | **RMSE (W)** |
|---|---|---|---|---|
| MLR | single | 26 | 98 | 184 |
| | multi | 6 | 105 | 241 |
| GBR | single | 21 | 83 | 170 |
| | multi | 13 | 78 | 177 |
| MLP | single | 22 | 77 | 168 |
| | multi | 10 | 89 | 187 |

Table D.3: Performances of the models in cross validation for forecasting the PV with respect to the training period selected

| Model | outputs | Training period (days) | CRPS (W) |
|---|---|---|---|
| GPR | single | 10 | 203 |
| | multi | 27 | 187 |

Table D.4: Performances of the GPR models in cross validation for forecasting the PV with respect to the training period selected

## D.8 Conclusion

Thanks to the study made in this chapter, the adequate training set size for each model was studied. Now that this training set size is known, the next chapter can be devoted to the study of the retraining frequency.

# Appendix E

# Retraining frequency analysis

In this chapter, the strategy developed in section 4.5.4 was used to evaluate the impact of the retraining frequency on the performances of the different models. This reduction of performances is represented graphically as can be seen on the various pictures from this chapter. The blue line from the charts represents the performance of the models depending on the time separating the training of the models from the forecasting. These performances are expressed in terms of the NRMSE ( eq. 3.3). The red line from the charts represents the level at which the performance of a model declines by 5% compared to a model that is trained daily (i.e. by training the model each time that forecasts are produced).

## E.1 Linear regression

### E.1.1 Single output model

With the model used for forecasting the load, the 5 % threshold is reached after 19 days without retraining. For the forecasting of the pv, the 5% threshold is reached after 15 days without retraining.

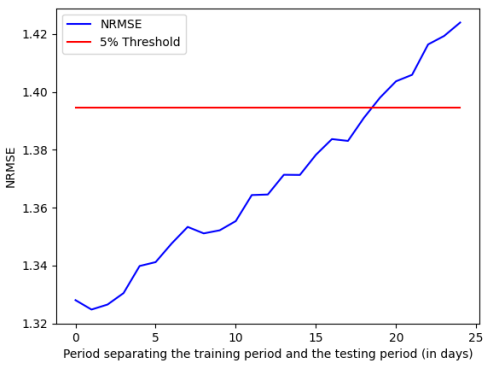



Figure E.1: NRMSE for forecasting the load with a LR

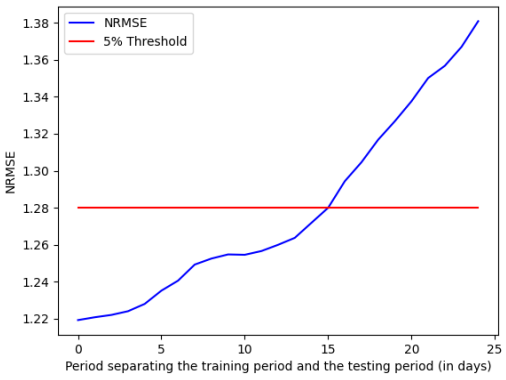



Figure E.2: NRMSE for forecasting the PV with a LR

### E.1.2 Multi-output model

For the forecasting of the load, the 5% is never reached so the retraining period has been chosen to 7 days. For the forecasting of the PV, the increase being too sharp at the beginning, the retraining was selected to be performed each day.

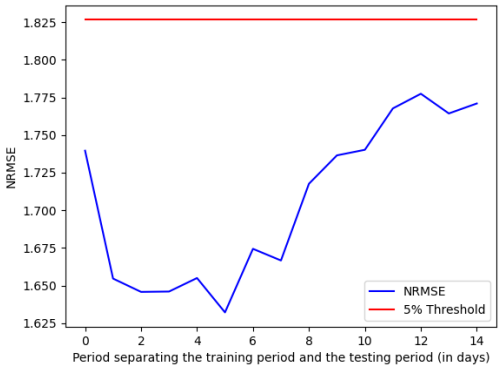



Figure E.3: NRMSE for forecasting the load with a MLP

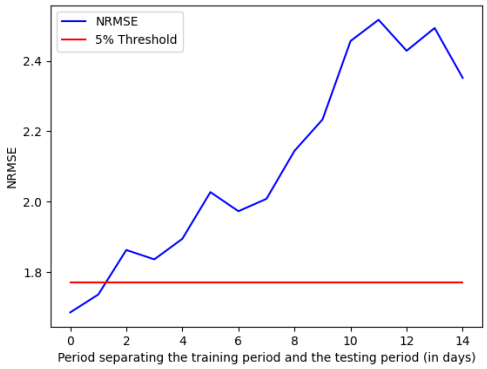



Figure E.4: NRMSE for forecasting the PV with a multioutput MLP

## E.2 Gradient Boosting Regression

### E.2.1 Single output model

For the forecasting of the load, the 5% threshold is reached after 10 days. For the forecasting of the PV, the 5% threshold is reached after 13 days.

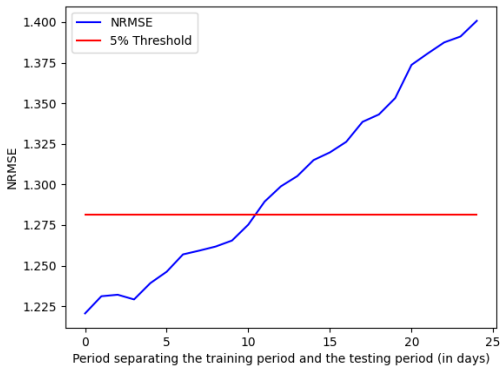



Figure E.5: NRMSE for forecasting the load with a GBR

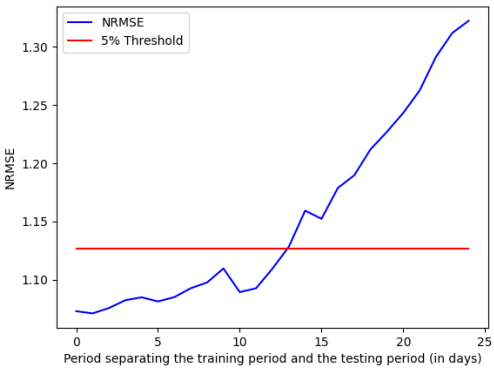



Figure E.6: NRMSE for forecasting the PV with a GBR

### E.2.2 Multi-output model

For the forecasting of the load, the threshold is reached after 8 days. For the forecasting of the PV, the threshold is reached after 5 days.

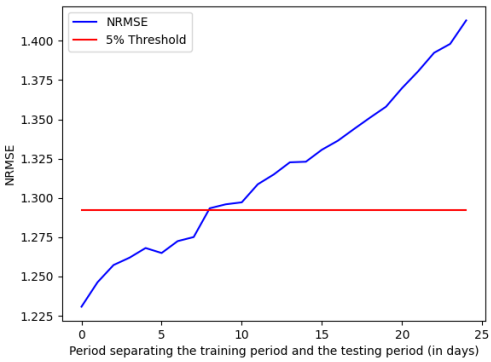



Figure E.7: NRMSE for forecasting the load with a multi output GBR

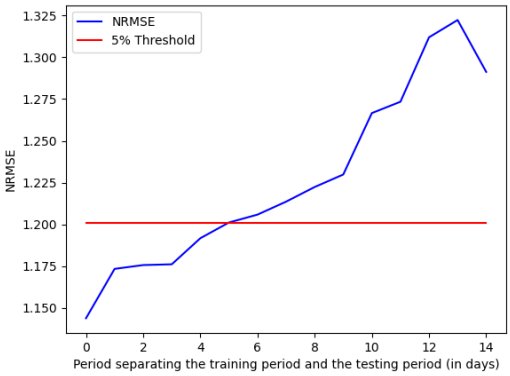



Figure E.8: NRMSE for forecasting the PV with a multi GBR

## E.3 MultiLayer Perceptron

### E.3.1 Single output model

For the forecasting of the load, the threshold is reached after 5 days. For the forecasting of the PV, the retraining should be performed every 8 days.

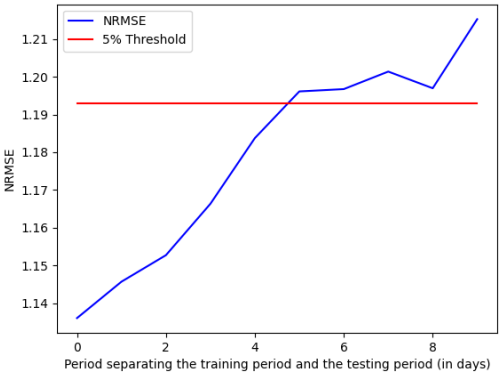



Figure E.9: NRMSE for forecasting the load with a MLP

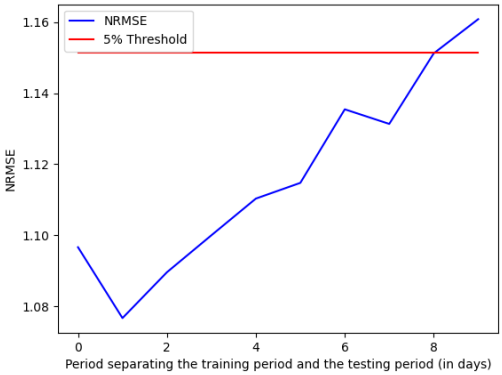



Figure E.10: NRMSE for forecasting the PV with a MLP

### E.3.2 Multi-output model

For the forecasting of the load, the threshold is reached after 7 days. For the forecasting of the PV, the threshold is reached after 1 day.

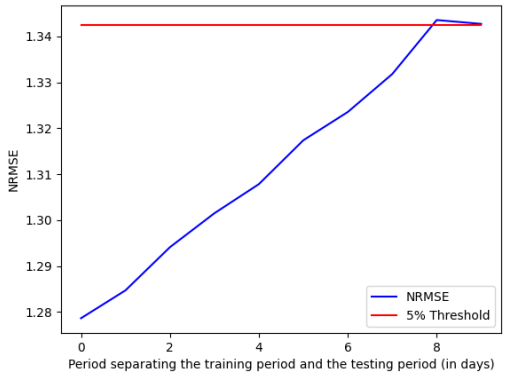



Figure E.11: NRMSE for forecasting the load with a MLP

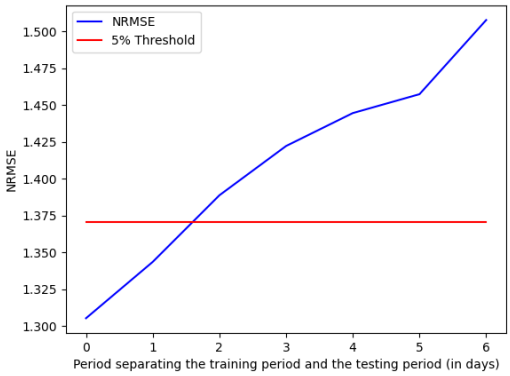



Figure E.12: NRMSE for forecasting the PV with a MLP

## E.4 Summary of the results

**Load forecasting** The selected retraining period for each model are shown in table E.1. For most of the models, the retraining could be done at a low frequency. A weekly retraining of the models provides a maximum degradation of the performances[1] of 5%.

**PV forecasting** The selected retraining period for each model are shown in table E.2. It can be seen that in the case of the forecasting of the PV production, the single output models can be trained at a lower frequency compared to the multi output models. Indeed, they can be retrained weekly while the multi output models need to be retrained every 2 to 6 days.

| Model | Output mode | Retraining time (days) |
|---|---|---|
| EST | / | 1 |
| MLR | single | 7 |
| | multi | 7 |
| GBR | single | 7 |
| | multi | 6 |
| MLP | single | 7 |
| | multi | 7 |

Table E.1: Best selected retraining period for different models for forecasting the load

| Model | Output mode | Retraining time (days) |
|---|---|---|
| MLR | single | 7 |
| | multi | 2 |
| GBR | single | 7 |
| | multi | 6 |
| MLP | single | 7 |
| | multi | 2 |

Table E.2: Best selected retraining period for different models for forecasting the PV

[1] in term of NRMSE (Eq. 3.3)

# Appendix F

# Hyperparameters of forecasting models

| | Load forecasting | | PV forecasting | |
|---|---|---|---|---|
| Output mode | single | multi | single | multi |
| Maximal depth | 7 | 5 | 4 | 5 |
| Number of estimators | 19 | 12 | 40 | 29 |
| Minimum number of samples per split | 5 | 5 | 5 | 5 |
| Minimum number of samples per leaf | 51 | 5 | 5 | 5 |
| Learning rate | 0.1 | 0.15 | 0.1 | 0.1 |

Table F.1: Hyperparameters of the Gradient boosting model

| | Load forecasting | | PV forecasting | |
|---|---|---|---|---|
| Scaling input | Min-Max | Standard scaling | Standard scaling | Standard scaling |
| Scaling output | Standard scaling | Standard scaling | Min-Max | Standard scaling |
| X | 36 | $8 \times 96$ | 17 | $15 \times 96$ |
| M | 80 | $2 \times X$ | 35 | $2 \times X$ |
| N | 1 | 96 | 1 | 96 |
| Learning rate | 0.005 | 0.002 | 0.001 | 0.001 |
| $N_{\text{epochs}}$ | 200 | 130 | 100 | 130 |
| Dropout rate | 0.3 | 0.6 | 0.5 | 0.4 |
| Batch size | 20 | 20 | 30 | 50 |
| $\Sigma_h$ | ReLu | ReLu | Linear | Softmax |
| $\Sigma_o$ | Tanh | ReLu | ReLu | ReLu |

Table F.2: Hyperparameters of the MLP. X is the number of input neurons, M is the number of neurons in the hidden layer, N is the number of output neurons, $\Sigma_h$ is the activation function of the hidden layer and $\Sigma_o$ is the activation fucntion of the output layer.

# Appendix G

# Monitoring system

## G.1 Model averaging

| **Combined models** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| Single output GBR and MLP and multi output GBR | 614 | 1080 |
| All models | 647 | 1102 |

Table G.1: Performances of the model obtained by aggregating several models for the forecasting of the load

| **Combined models** | **MAE** (W) | **RMSE** (W) |
|---|---|---|
| Single output MLP and GBR and multi output GBR | 70 | 154 |
| All models | 72 | 154 |

Table G.2: Performances of the model obtained by aggregating several models for the forecasting of the PV

## G.2 Truncated mean

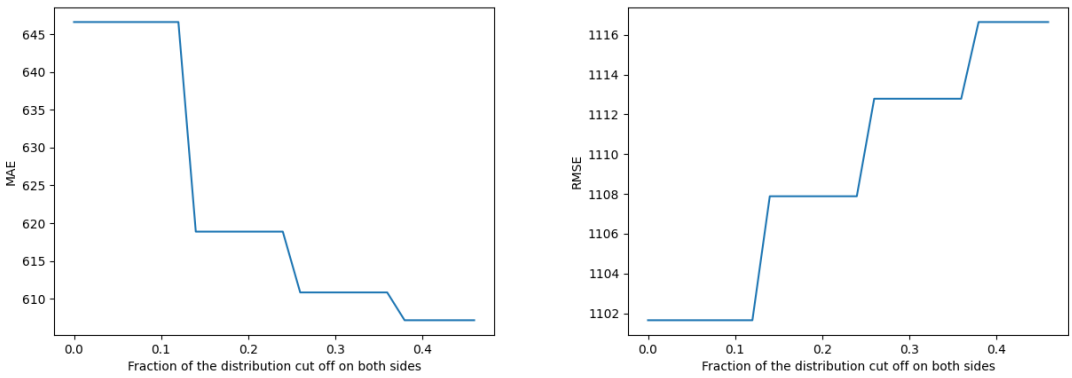



Figure G.1: Variation of the cut off fraction for the forecasting of the load

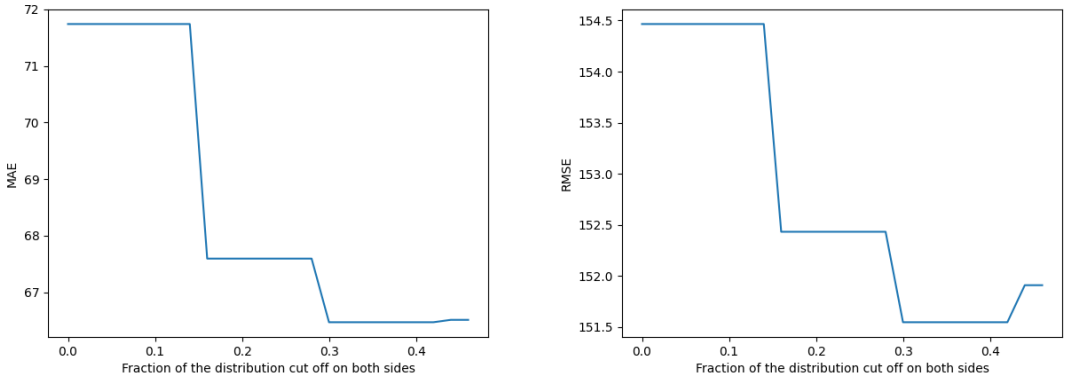



Figure G.2: Variation of the cut off fraction for the forecasting of the PV

## G.3 Best model selection

### G.3.1 Load forecasting

| window size (days) | MAE (W) | RMSE (W) |
|---:|---:|---:|
| 1 | 661 | 1176 |
| 7 | 620 | 1090 |
| from 14 to all past data | 617 | 1076 |

Table G.3: Minimization of the RMSE for the forecasting of the load with different window sizes

| window size (days) | MAE (W) | RMSE (W) |
|---:|---:|---:|
| 1 | 663 | 1097 |
| 7 | 618 | 1096 |
| 14 | 611 | 1083 |
| 21 | 617 | 1091 |
| 30 to all past data | 623 | 1109 |

Table G.4: Minimization of the MAE for the forecasting of the load with different window sizes

### G.3.2 PV forecasting

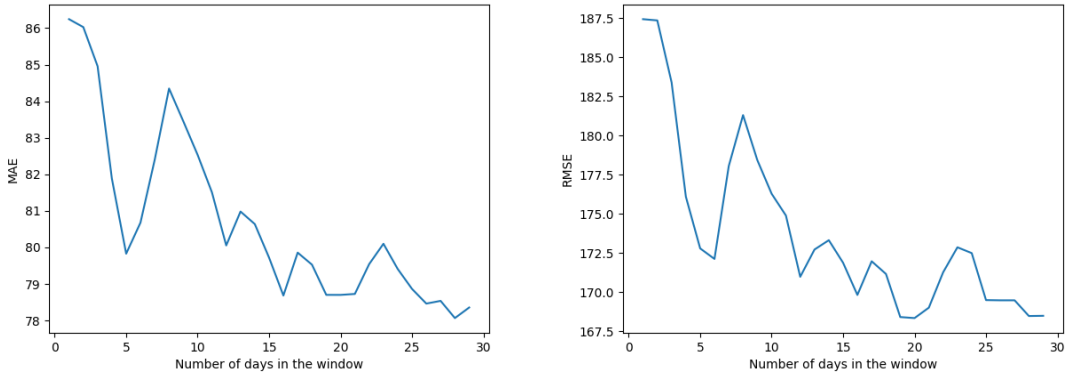



Figure G.3: Performances of the multi-model forecast by selecting each day the model that had the lowest RMSE on a past time window

| Window size (days) | MAE (W) | RMSE (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| 5 | 80 | 173 | 3.71 | 8.03 |
| 10 | 83 | 176 | 3.84 | 8.19 |
| 15 | 80 | 172 | 3.70 | 7.99 |
| 20 | 79 | 168 | 3.66 | 7.82 |
| 25 | 79 | 170 | 3.66 | 7.88 |
| 30 | 79 | 170 | 3.66 | 7.89 |
| all previous data | 80 | 171 | 3.70 | 7.95 |

Table G.5: Minimization of the RMSE for the forecasting of the load with different window sizes

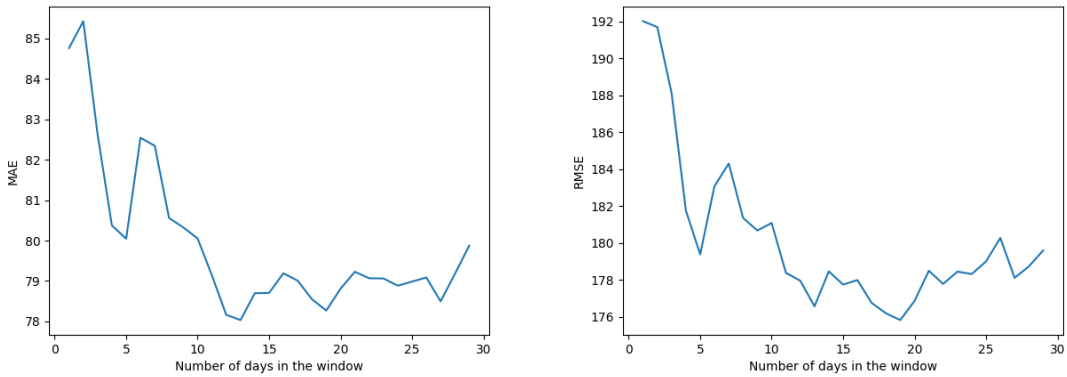



Figure G.4: Performances of the multi-model forecast by selecting each day the model that had the lowest MAE on a past time window

| Window size (days) | MAE (W) | RMSE (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| 5 | 80.0 | 179 | 3.72 | 8.34 |
| 10 | 80.0 | 181 | 3.72 | 8.42 |
| 15 | 79.0 | 178 | 3.66 | 8.26 |
| 20 | 79.0 | 177 | 3.66 | 8.22 |
| 25 | 79.0 | 179 | 3.67 | 8.32 |
| 30 | 79.0 | 180 | 3.67 | 8.36 |

Table G.6: Minimization of the MAE for the forecasting of the load with different window sizes

## G.4 inverse MSE Weighting

### G.4.1 Load forecasting

| window size (days) | MAE (W) | RMSE (W) |
|---|---|---|
| 1 | 637 | 1097 |
| 7 | 636 | 1087 |
| 14 | 637 | 1089 |
| 21 | 638 | 1090 |
| 30 | 641 | 1093 |
| all past data | 641 | 1094 |

Table G.7: Inverse mse weighting with different window sizes

### G.4.2 PV forecasting

| Window size (days) | MAE (W) | RMSE (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---|---|---|---|---|
| 10 | 71 | 152 | 3.29 | 7.06 |
| 15 | 71 | 152 | 3.29 | 7.05 |
| 20 | 71 | 152 | 3.29 | 7.05 |
| 25 | 71 | 152 | 3.29 | 7.05 |
| 30 | 71 | 152 | 3.30 | 7.06 |
| all past data | 71 | 152 | 3.30 | 7.06 |

Table G.8: Inverse mse weighting with different window sizes for the forecasting of the PV

## G.5 OLS weighting

### G.5.1 Load forecasting

| window size (days) | MAE (W) | RMSE (W) |
|---:|---:|---:|
| 1 | 847 | 1498 |
| 7 | 675 | 1082 |
| 14 | 661 | 1060 |
| 21 | 658 | 1052 |
| 30 | 664 | 1055 |
| all past data | 667 | 1053 |

Table G.9: OLS weighting with different window sizes

### G.5.2 PV forecasting

| Window size (days) | MAE (W) | RMSE (W) | $\frac{\text{MAE}}{\max \mathbf{P}}$ (%) | $\frac{\text{RMSE}}{\max \mathbf{P}}$ (%) |
|---:|---:|---:|---:|---:|
| 5 | 82 | 174 | 3.83 | 8.10 |
| 10 | 78 | 167 | 3.61 | 7.76 |
| 15 | 75 | 161 | 3.48 | 7.49 |
| 20 | 74 | 159 | 3.43 | 7.37 |
| 25 | 74 | 158 | 3.44 | 7.33 |
| 30 | 75 | 158 | 3.46 | 7.34 |
| all past data | 75 | 157 | 3.47 | 7.29 |

Table G.10: OLS weighting with different window sizes for the forecasting of the PV

# References


[1] Robert T. Clemen. **?**Combining forecasts: A review and annotated bibliography**?** In: *International Journal of Forecasting 5* (1989), pp. 559–583.

[2] T. N. Krishnamurti et al. **?**Improved Weather and Seasonal Climate Forecasts from Multimodel Superensemble**?** In: (1999). DOI: 10.1126.

[3] J. Bates and C. Granger. **?**The combination of forecasts.**?** In: *International Journal of Forecasting 16* (2000), pp. 451–476.

[4] Spyros Makridakis and Michèle Hibon. **?**The M3-Competition: results, conclusions and implications**?** In: *International Journal of Forecasting 16* (2000), pp. 451–476.

[5] G. Elliot. **?**Averaging and the Optimal Combination of Forecasts**?** In: (2011).

[6] F. Pedregosa et al. **?**Scikit-learn: Machine Learning in Python**?** In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[7] Juan M. Morales et al. *Integrating Renwables in Electricity Markets. Operational Problems*. Springer Science & Business Media, 2013.

[8] Arbab N. Fahad MU. **?**Factor affecting short term load forecasting.**?** In: *Journal of Clean Energy Technologies* (2014), pp. 305–309. DOI: 10.7763/JOCET.2014.V2.145.

[9] M.M. Fouad, Lamia A. Shihata, and ElSayed I. Morgan. **?**An integrated review of factors influencing the performance of photovoltaic panels**?** In: *Renewable and Sustainable Energy Reviews* (2017), pp. 1499–1511. DOI: 10.1016/j.rser.2017.05.141.

[10] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. Ed. by OTexts: Melbourne, Australia. 2018. Accessed on 13.03.2020.

[11] 2019 Microgrids at Berkeley Lab. *About Microgrid*. 2019. URL: https://building-microgrid.lbl.gov/about-microgrids (visited on 03/20/2020).

[12] *API Documentation - Feed API*. URL: https://emoncms.org/site/api#feed. Accessed on 02.06.2020.

[13] *Emoncms.org*. URL: https://emoncms.org/. Accessed on 02.06.2020.

[14] *The Ultimate Guide to Model Retraining*. URL: https://mlinproduction.com/model-retraining/. (accessed: 15.03.2020).