Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

ML4AQ meeting – ML-MOS in Mexico City

PETETIN Hervé

XX/06/2019

Plan

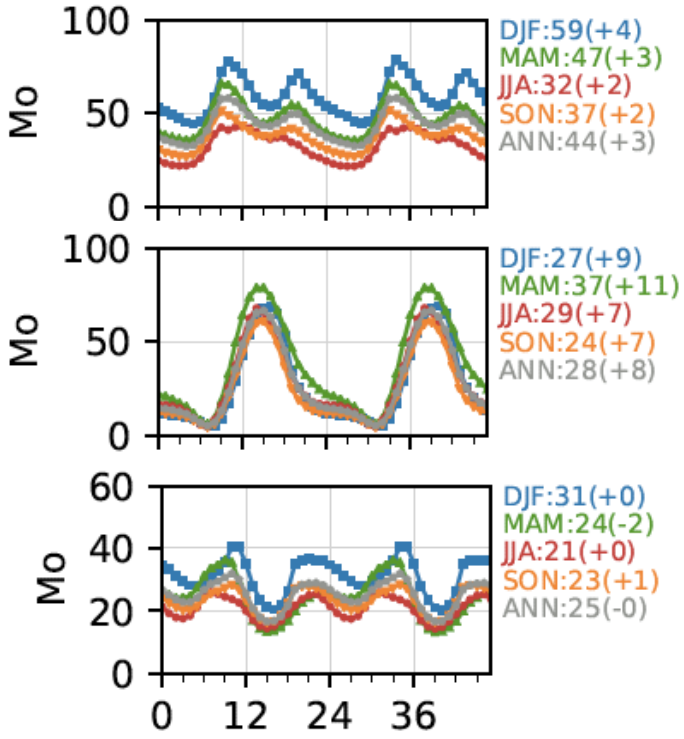
- MOS correction of the operational AQ forecasting system of Mexico City (16 months in 2017-2018)
- Influence of hyperparameters tuning
- Uni- versus multi-station MOS correction
- Features importance
- Probabilistic forecasts with quantile regression

Data : AQ forecasts in Mexico City

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Mexico City dataset

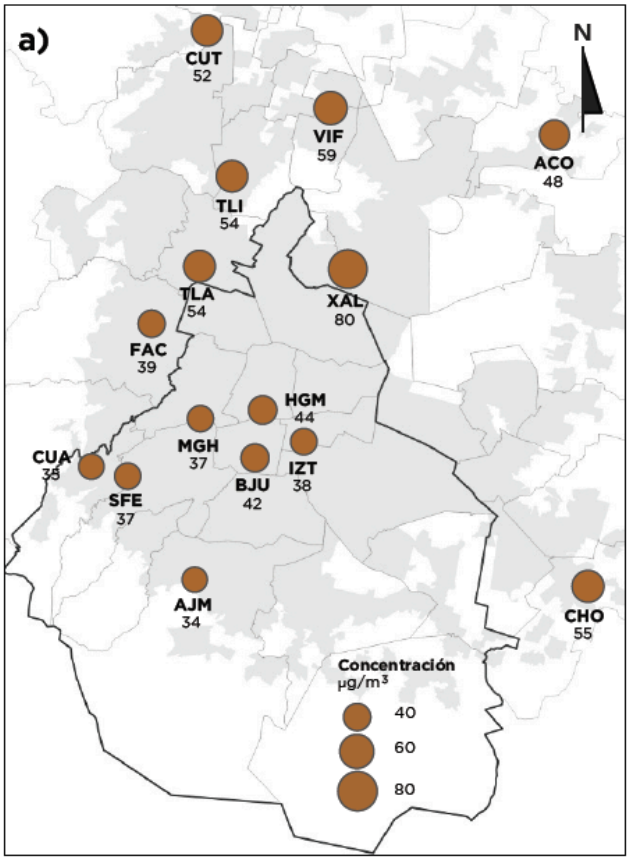
- Period : 16 months in 2017-2018
  (8 August 2017 – 31 December 2018)
- Network of surface stations (35 stations), with at least 50% hourly data :
  - PM10 : 20 stations
  - O3 : 32 stations
  - NO2 : 24 stations
- Mean observed diurnal profiles :



PM10 (44 ug/m3)

O3 (28 ug/m3)

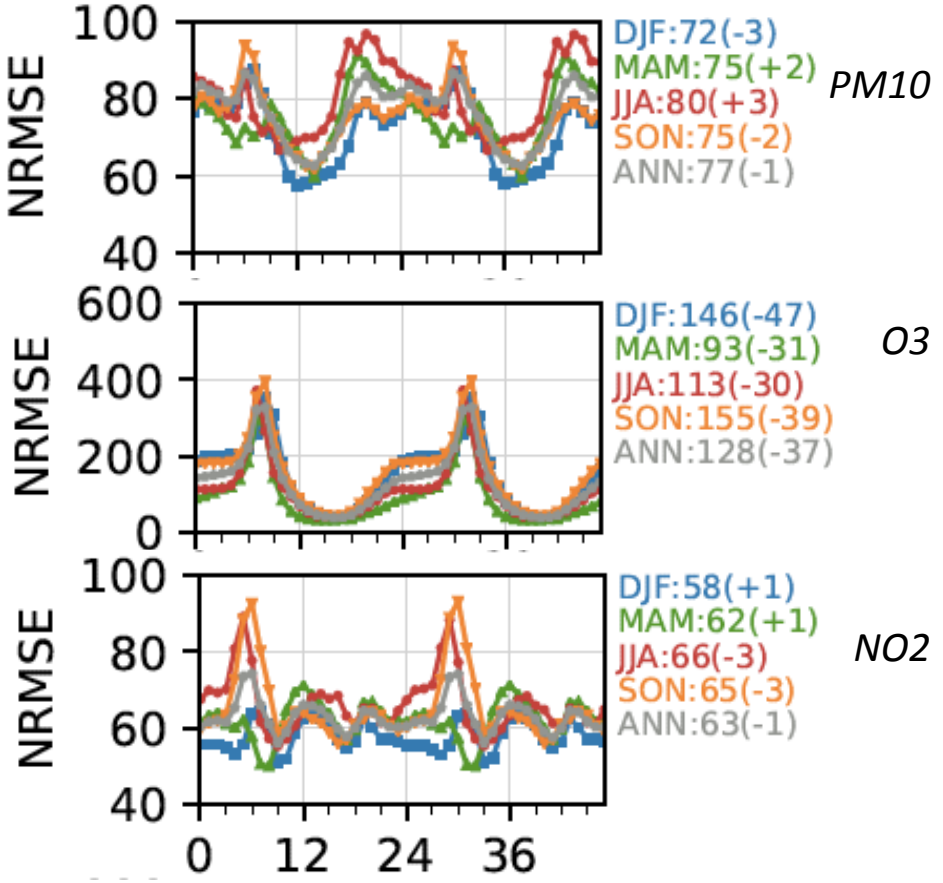
NO2 (25 ug/m3)



Average PM_{10} concentrations in Mexico

Performance of raw AQ forecasts

- Substantial underestimation of PM10
- Strong overestimation of O3
- Quite poor correlations for all pollutants



	PM10	O3	NO2
nMB (%)	-34%	+81%	-10%
nRMSE (%)	77%	128%	63%
PCC	0.42	0.43	0.50
N (hours)	184,824	314,256	230,832

Methodology for ML-based MOS correction

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Machine learning set-up

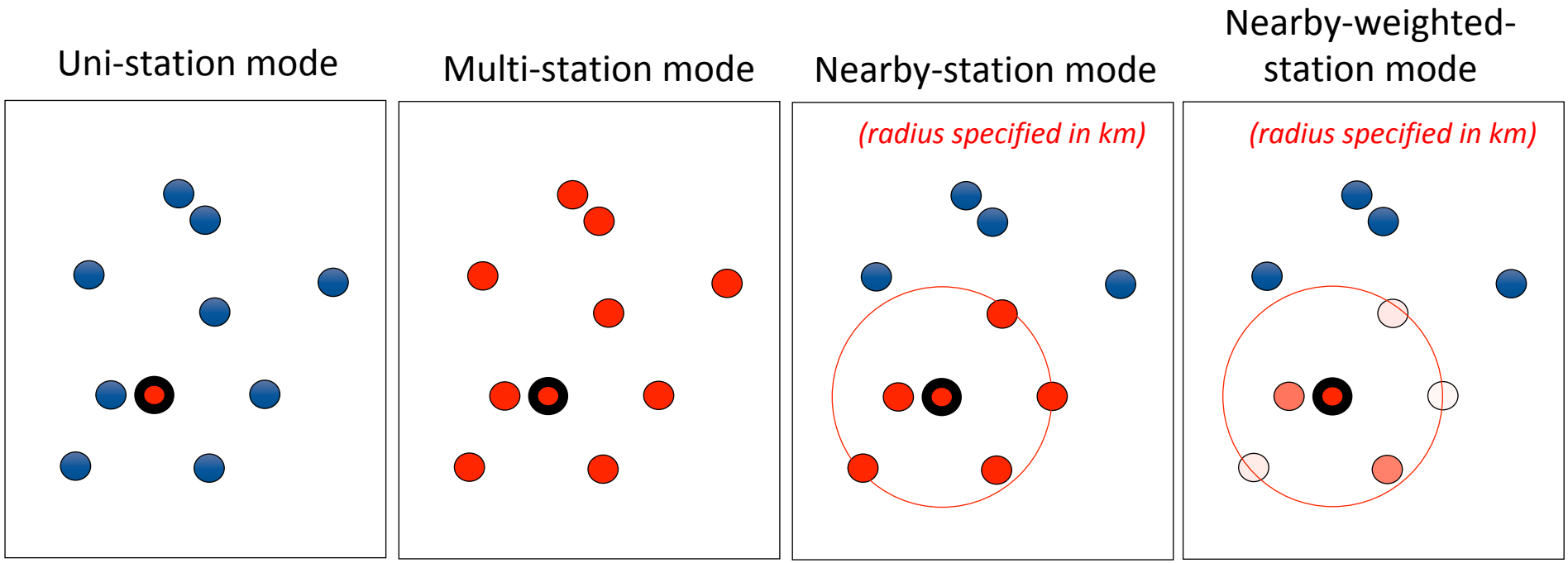
- **Strategy :** We mimic an operational system in which new forecasts and observations are obtained continuously. A new machine learning model is trained every month and used to correct the forecast during the coming month. Application to several pollutants : particles (PM_{10}), ozone (O_3), nitrogen dioxide (NO_2)

- **Target** : observed concentration (*supervised regression problem*)
- **Features** :
  - From chemistry-transport model : forecasted and past* concentrations
  - From meteorological model : forecasted and past* meteorological variables (temperature, wind, pressure) and their temporal gradients
  - From observations : past* concentrations
  - Other : julian date, hour of the day, day of week, month

- **Algoritms** : GBM (gradient boosting machine), RF (random forest)

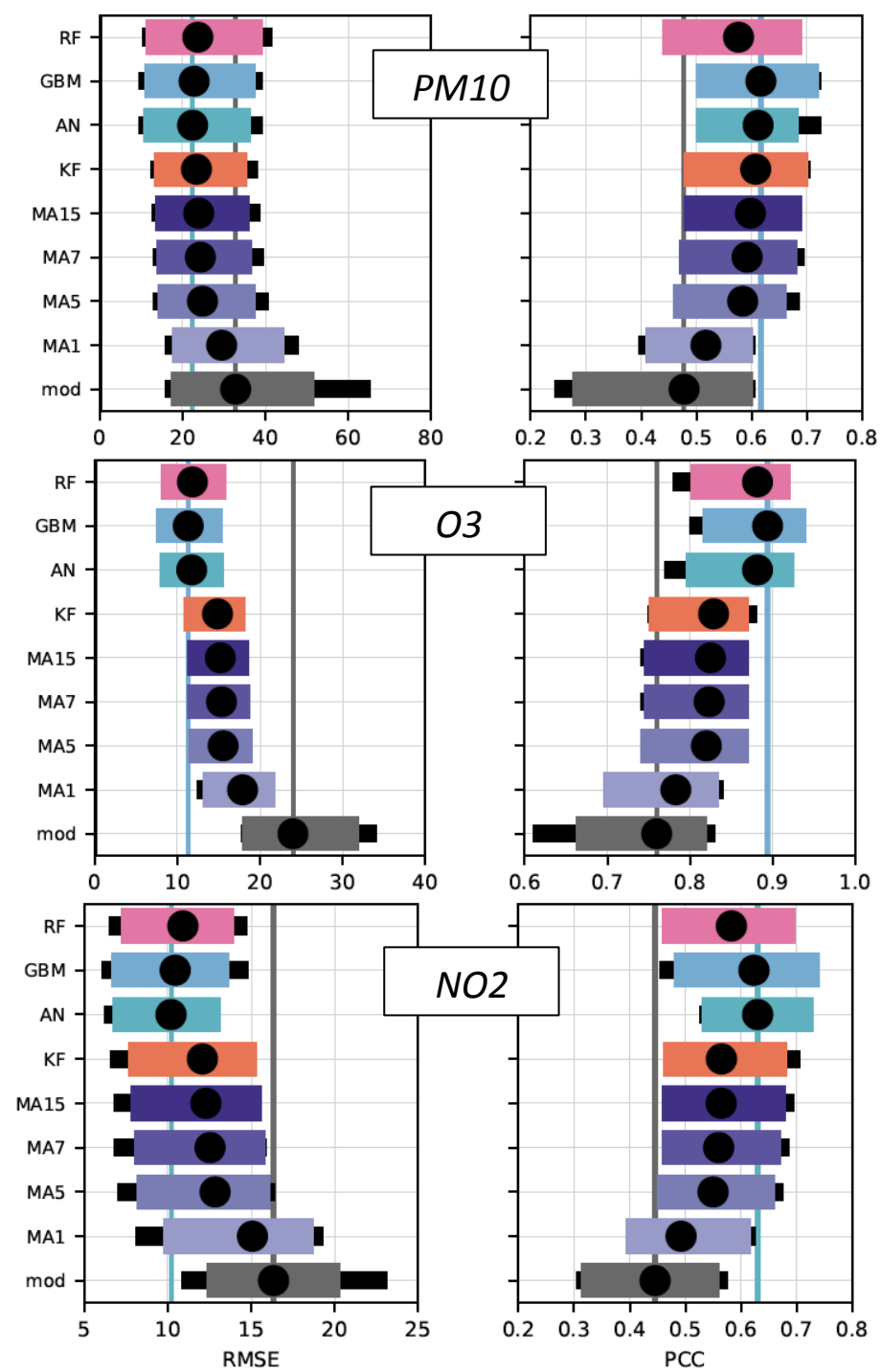
*"past" : one and two days before

Possible configurations



Uni-station MOS correction

Barcelona Supercomputing Center
Centro Nacional de Supercomputación



Uni-station results

- NB : Results with default ML hyperparameters
- Best performance obtained with AN and ML-GBM

Uni- versus multi-station MOS correction

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

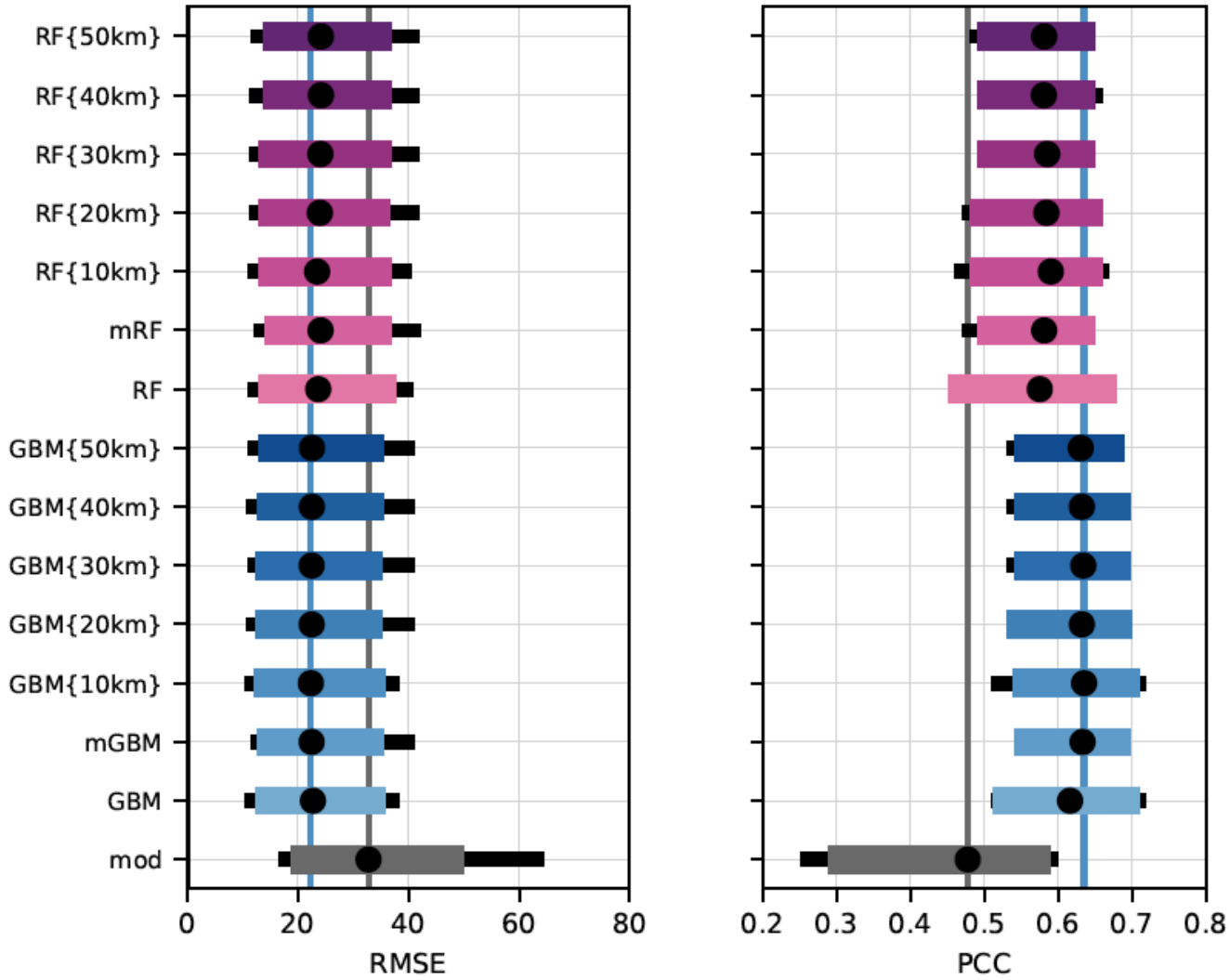
Preliminary remarks

- Multi-station approach explored with the ML-based MOS methods
- All or near-by stations taken into account (so far, no weights are applied)
- No additional features included (may be useful to include some features describing somehow the differences between the stations, e.g. station type)

- Approach potentially useful if large observational data gap at the beginning of the period at one given station (or if a new station installed in the city) ➔ in this case, the correction at this station can benefit from the information accumulated at the other stations

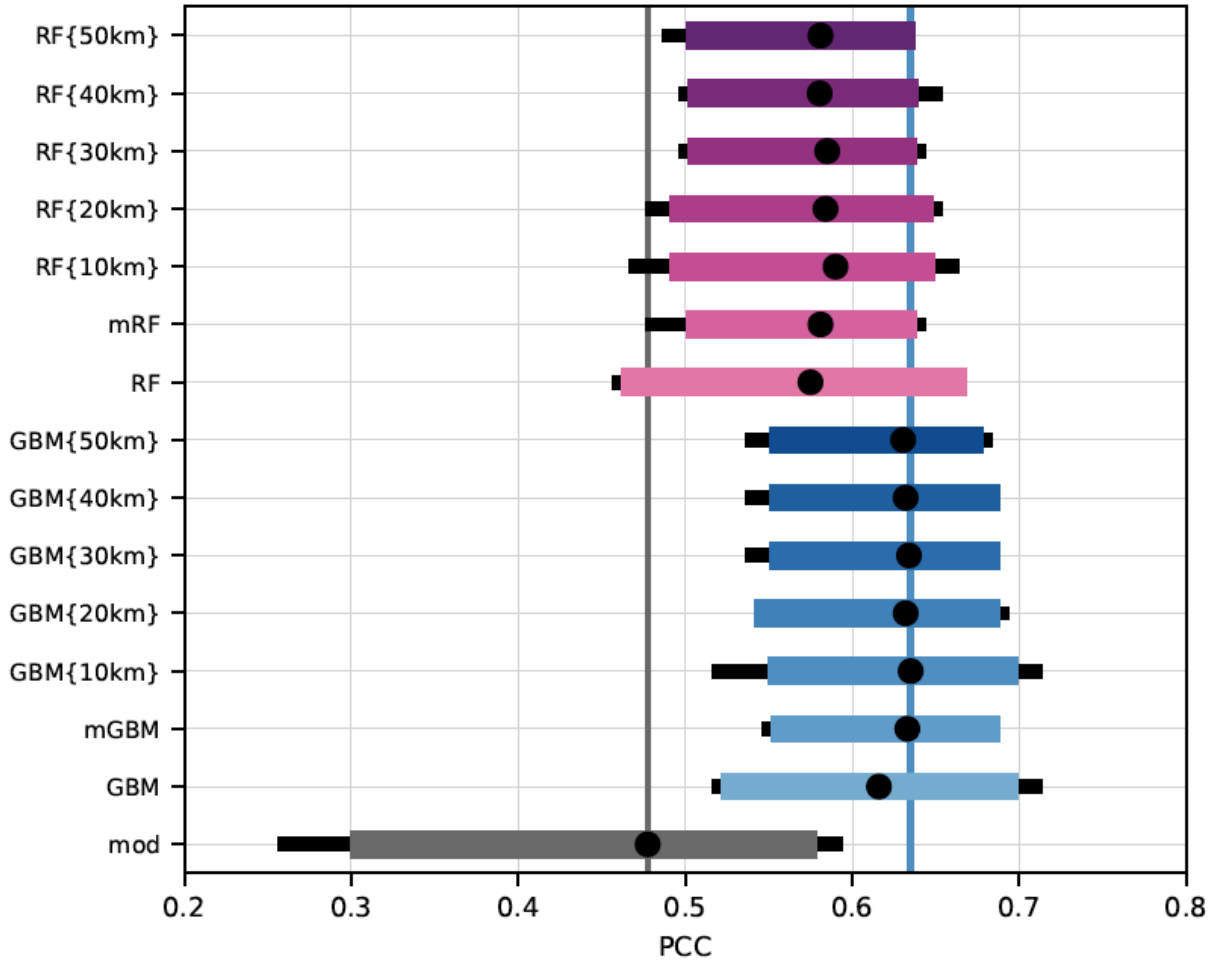
Results (here shown for PM10)

- Changes of performance concern more the correlation



Results (here shown for PM10)

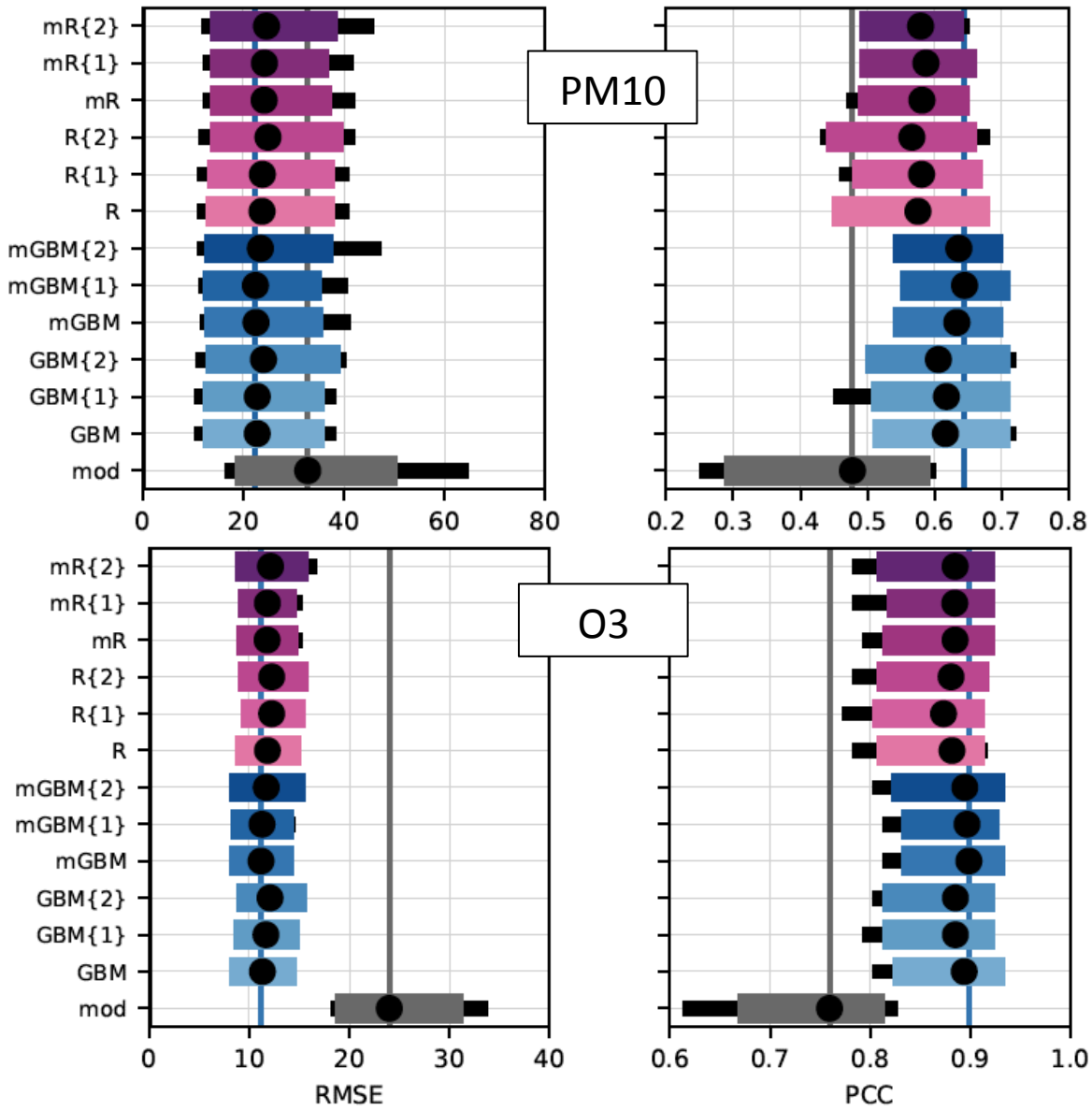
- Multi-station mode ➔ among the stations, less dispersion of the performance, i.e. best results are slightly deteriorated, worst results are improved
- Nearby-station mode ➔ quite similar results than multi-station with dispersion of the performance reduced with larger distances



Target variable (preliminary results)

Impact of the choice of target variable

- Options tested : (0) observed concentration, (1) model error, (2) log of observed concentration
- No strong differences, but best results are obtained by mGBM{1} for PM10 and mGBM{0} for O3



Features importance

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

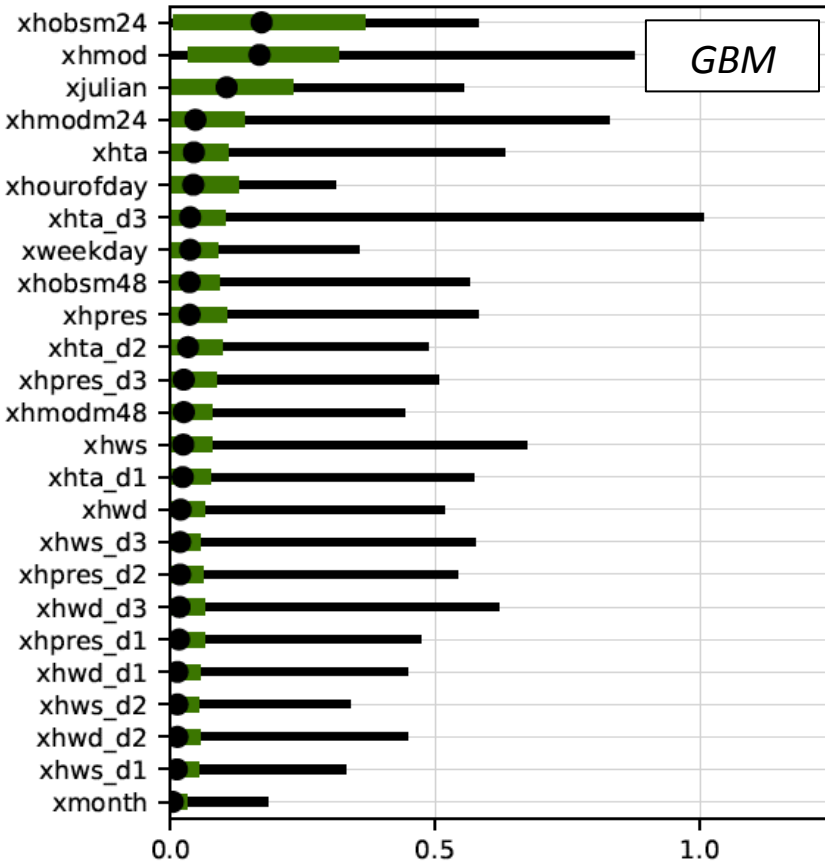
Features importance

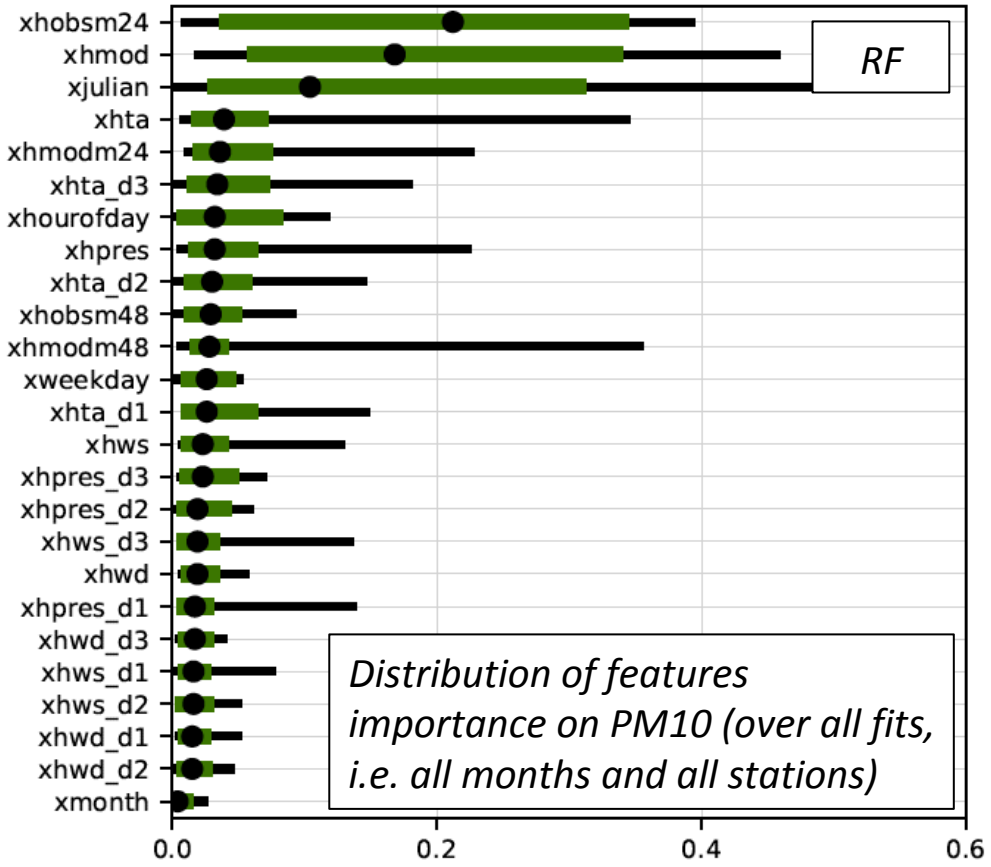
- Advantage of ensemble decision tree-based algorithms (GBM, RF) : some interpretability through the explicit quantification of the features importance
- Importance provides a score that indicates how useful or valuable each feature was in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance.
- NB : Features importance differs in GBM and RF

- Method : For a given pollutant, each time a ML is trained (i.e. every month, at every station), we get the corresponding features importance, and we can combine all these numbers into a distribution

Features importance : overall results for PM10

- Most important parameters :
  - **Observed concentration at day-1**
  - **Forecasted concentration**
  - **Julian date**
  - Concentration simulated at day-1
  - Temperature
  - …

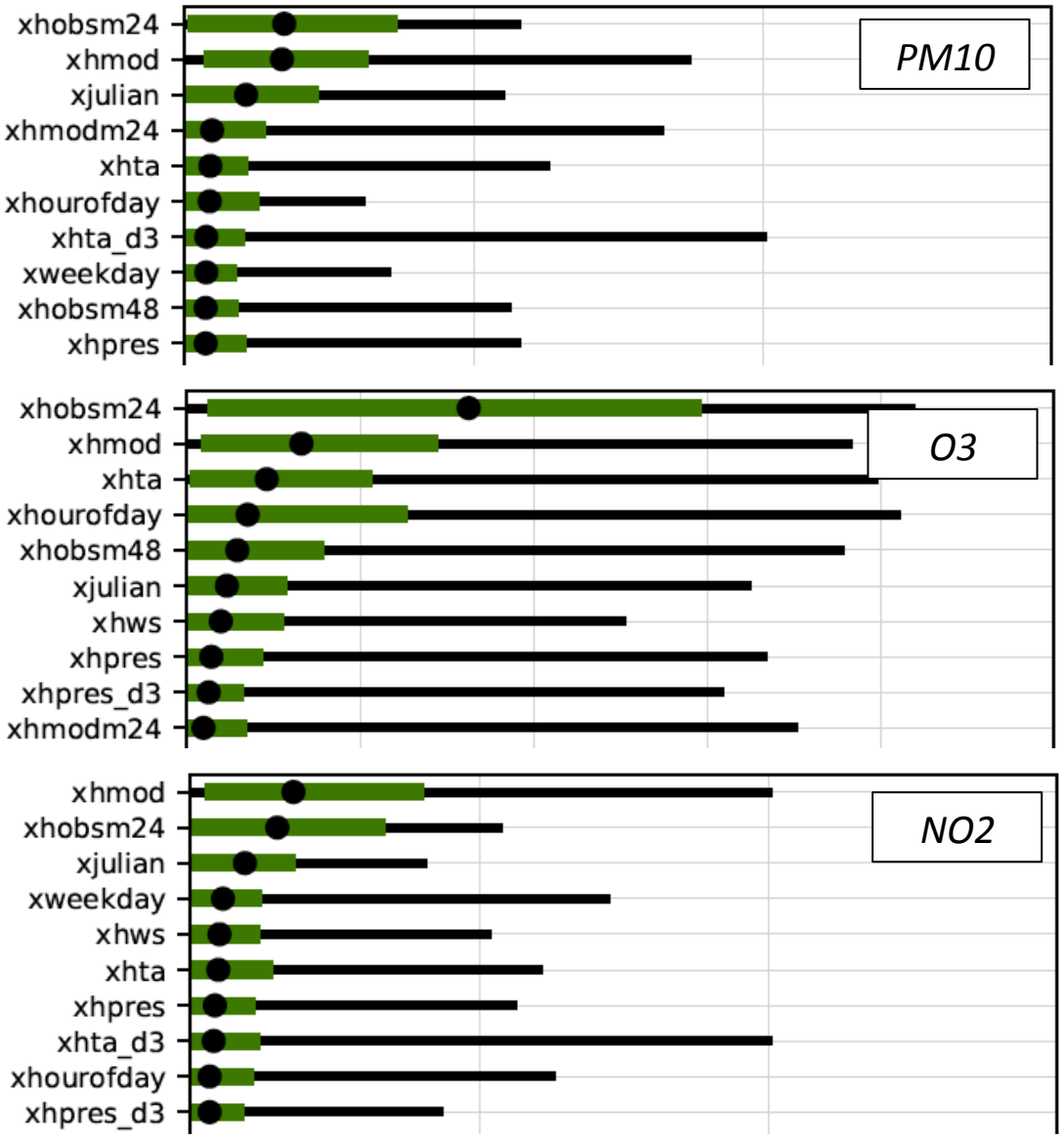




Difference among pollutants (here for GBM)

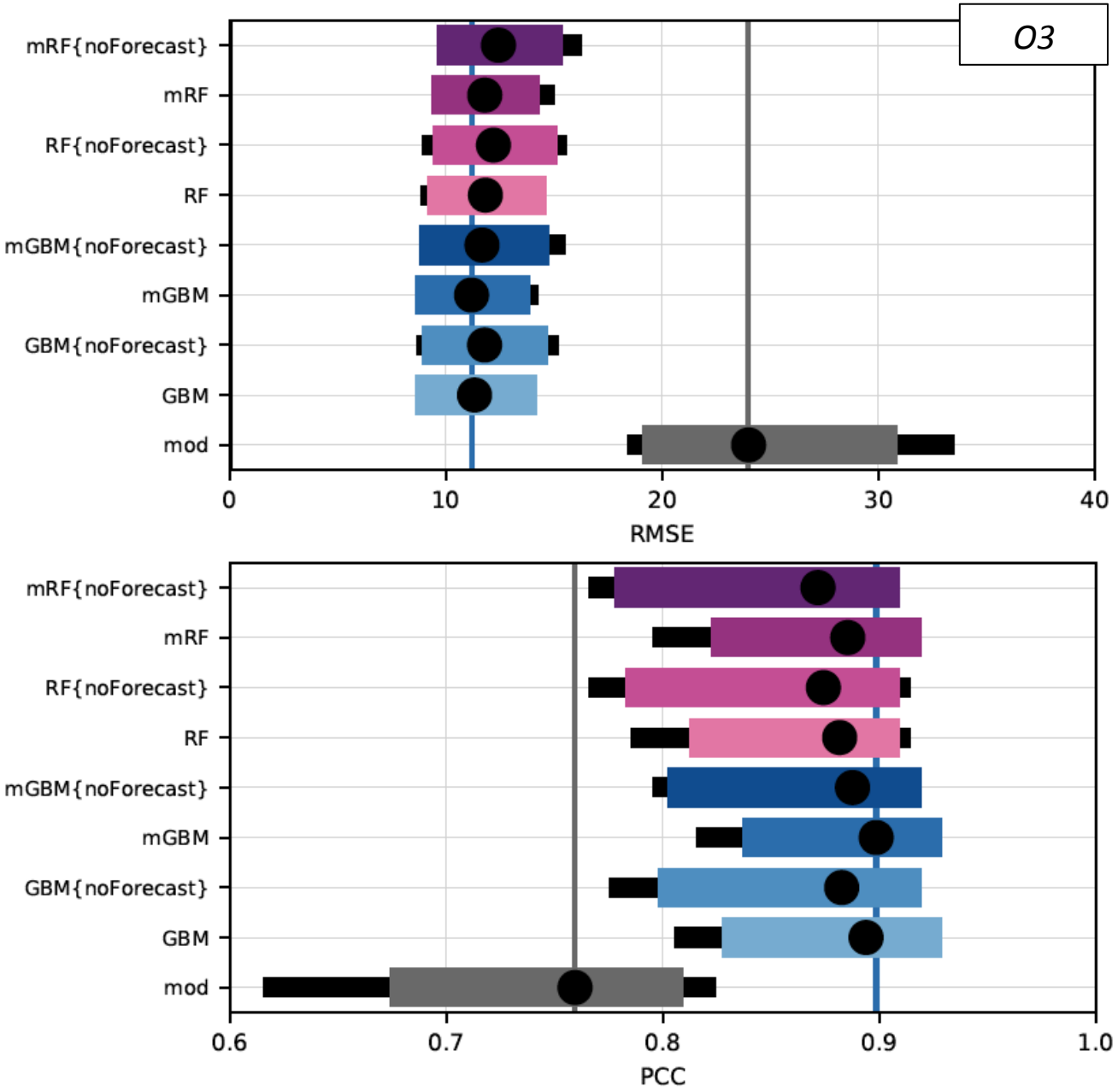
Compared to PM10 :

- O3 :
  - Obs at day-1 substantially more important than raw forecast, likely explained by the stronger autocorrelation of O3
  - Ttemperature
- NO2 :
  - Raw forecast slightly more important than obs at day-1, maybe due to its shorter lifetime
  - Importance of weekday



So are raw chemical forecasts really useful?... YES!

- Including them in the set of features clearly improves the results (as compared with "noForecast")
- **Particularly for the correlation** ➔ geophysical models help to better capture the dynamics of pollutant concentrations
- Idem for other pollutants

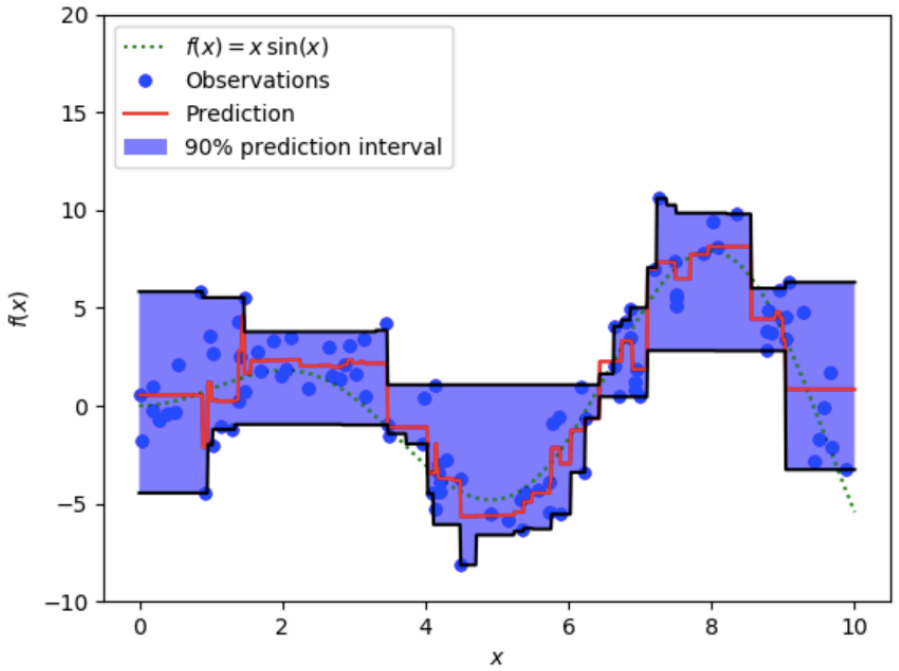


Probabilistic forecasts

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Quantile regression

- So far, all raw and corrected forecasts are deterministic ➔ provide the conditional mean concentration
- By using a different loss function, the GBM algorithm (implemented in sklearn python module) offers the possibility of performing quantile regression (QR) ➔ provide the conditional percentile for any given percentile
- Computational constraint : one ML model needs to be trained for each percentile



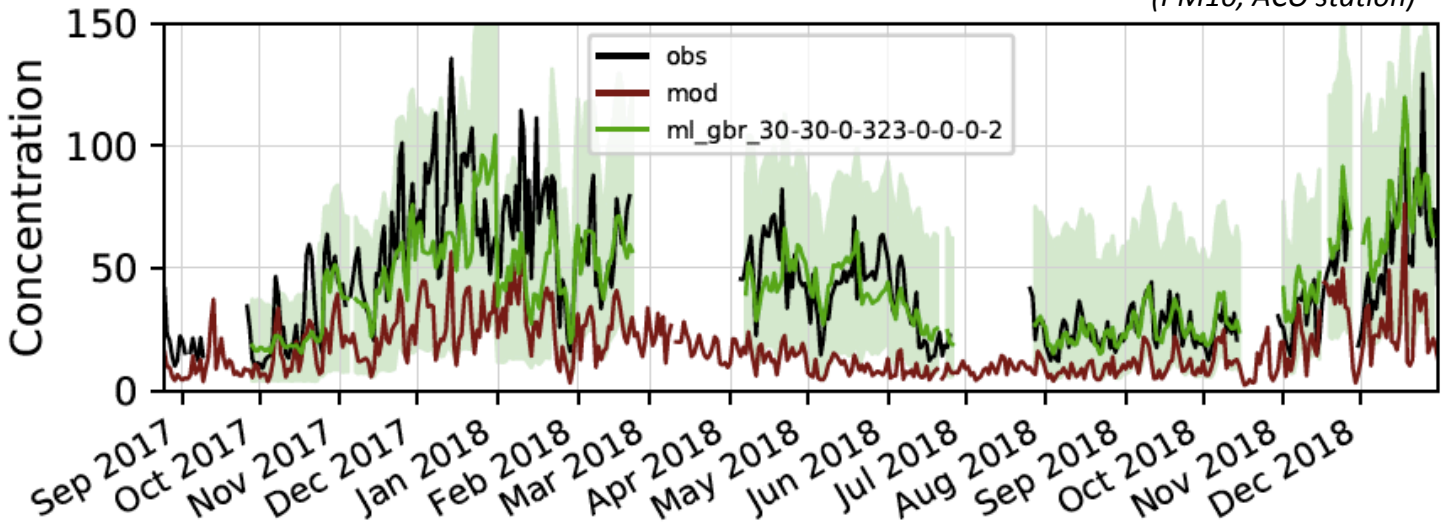
Here, 3 ML models trained, for :

- mean
- 5th percentile
- 95th percentile

Quantile regression

- So far, all raw and corrected forecasts are deterministic ➔ provide the conditional mean concentration
- By using a different loss function, the GBM algorithm (implemented in sklearn python module) offers the possibility of performing quantile regression (QR) ➔ provide the conditional percentile for any given percentile
- Computational constraint : one ML model needs to be trained for each percentile

(PM10, ACO station)



Here, 3 ML models trained, for :

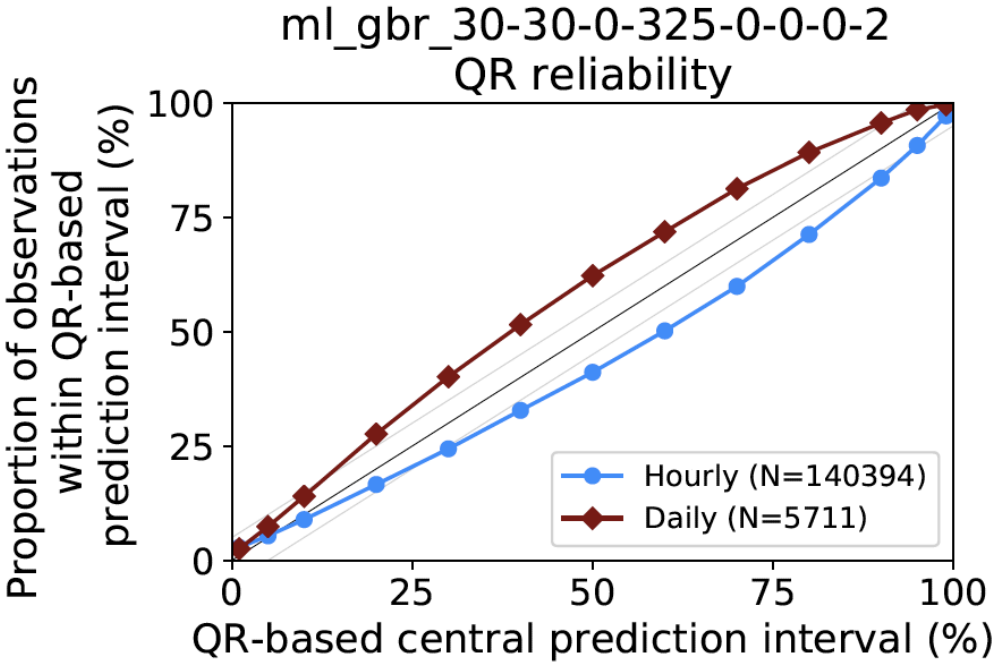
- mean
- 2.5th percentile
- 97.5th percentile

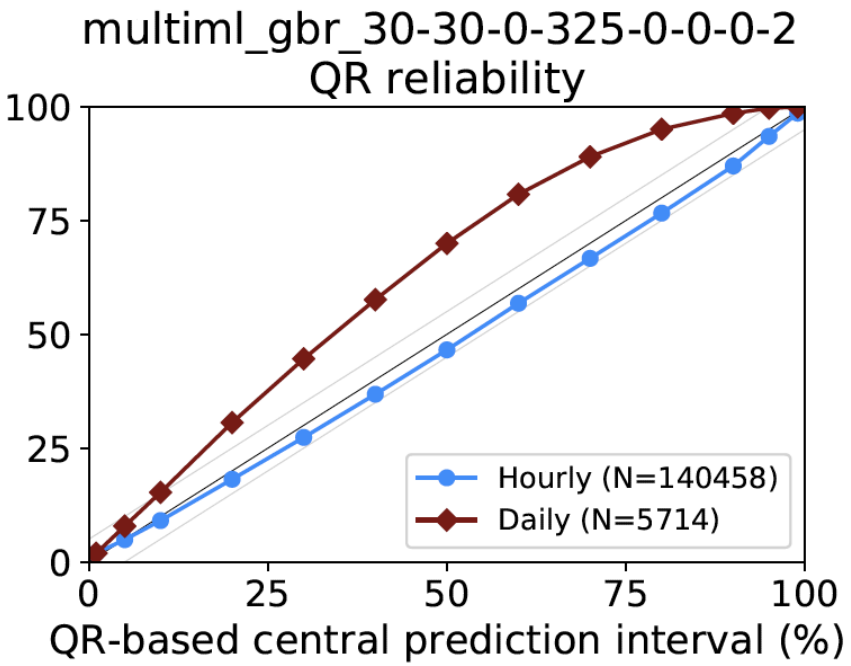
Quantile regression reliability : method

- Ideally : if the 90% (central) prediction interval obtained by QR is the concentration range [C_5,C_{95}], we want that 90% of the observations fall within this prediction interval; and idem for other central prediction intervals ➔ is it really the case?
- Method :
  - Besides the mean, QR performed on a set of (26) percentiles : 0.5, 2.5, 5, 10, 15, 20, 25, 30, 35, 40, 45, 47.5, 49.5, 50.5, 52.5, 55, 60, 65, 70, 75, 80, 85, 90, 95, 97.5, 99.5
  - By considering different pairs of these percentiles, we check the proportion of observations falling within the different prediction intervals : 1, 5, 10, 20… 80, 90, 95, 99%

Quantile regression reliability : results on PM10

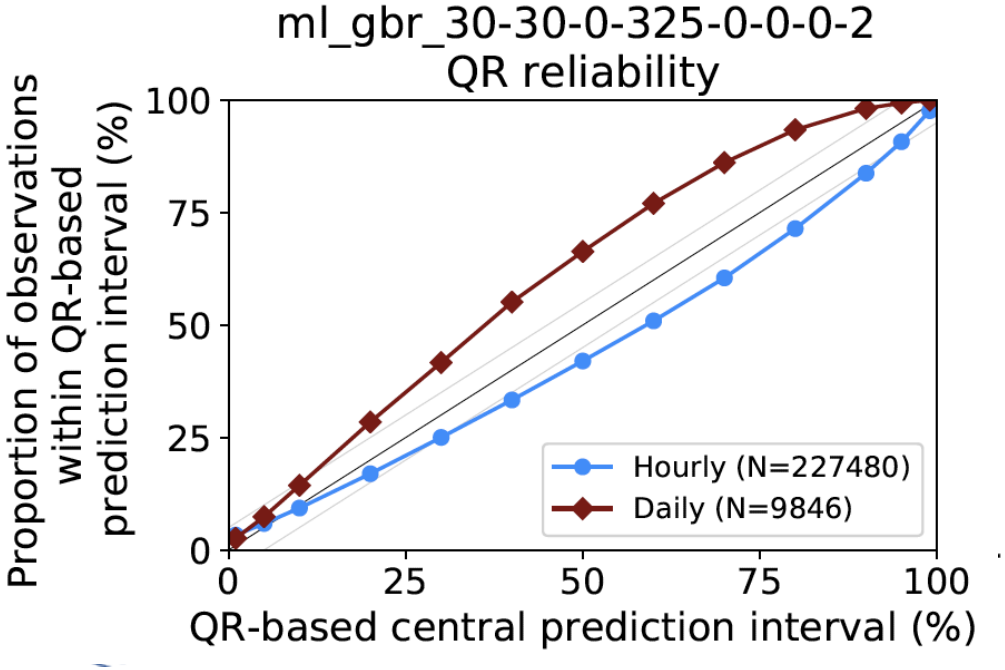
- For hourly concentrations (on which percentiles are fitted) : the QR reliability is reasonably good (although not always perfect)
- For daily concentrations : the QR-based prediction intervals are not well calibrated (they are far too conservative!)
- Better reliability on hourly data with multi-station model

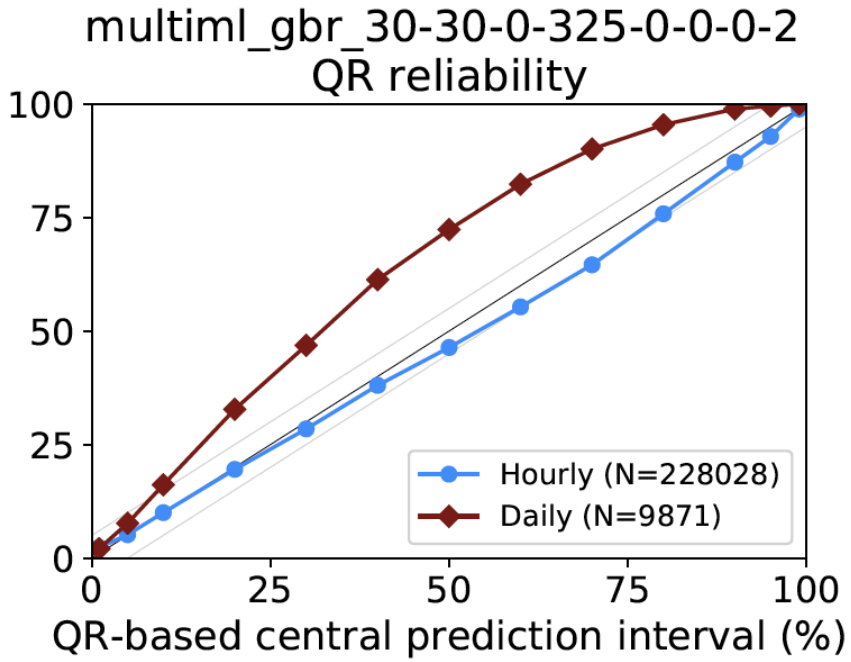




Quantile regression reliability : results on O3

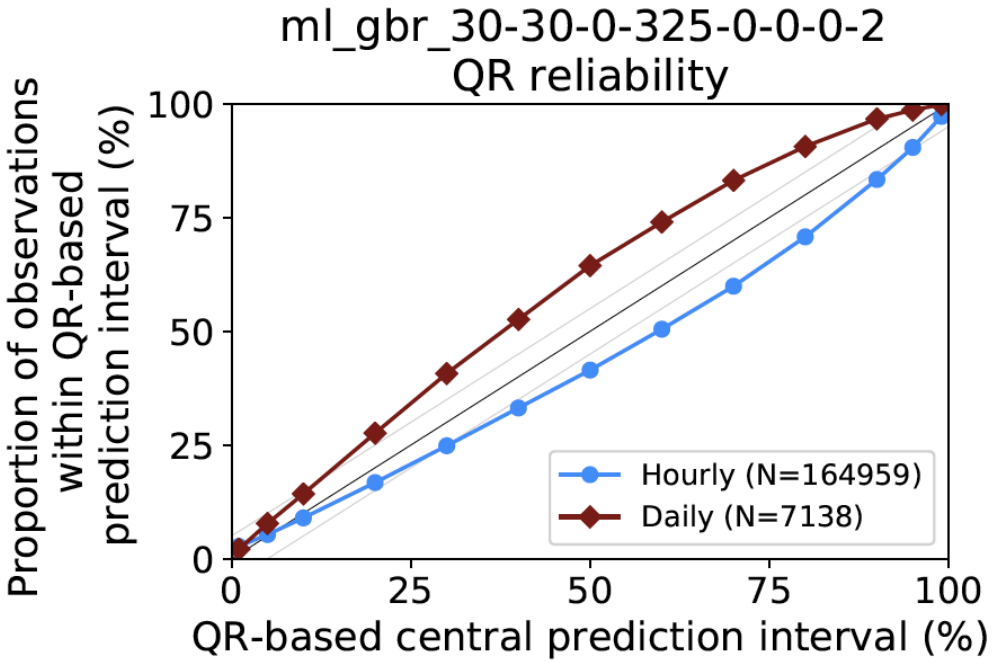
- For hourly concentrations (on which percentiles are fitted) : the QR reliability is reasonably good (although not always perfect)
- For daily concentrations : the QR-based prediction intervals are not well calibrated (they are far too conservative!)
- Better reliability on hourly data with multi-station model

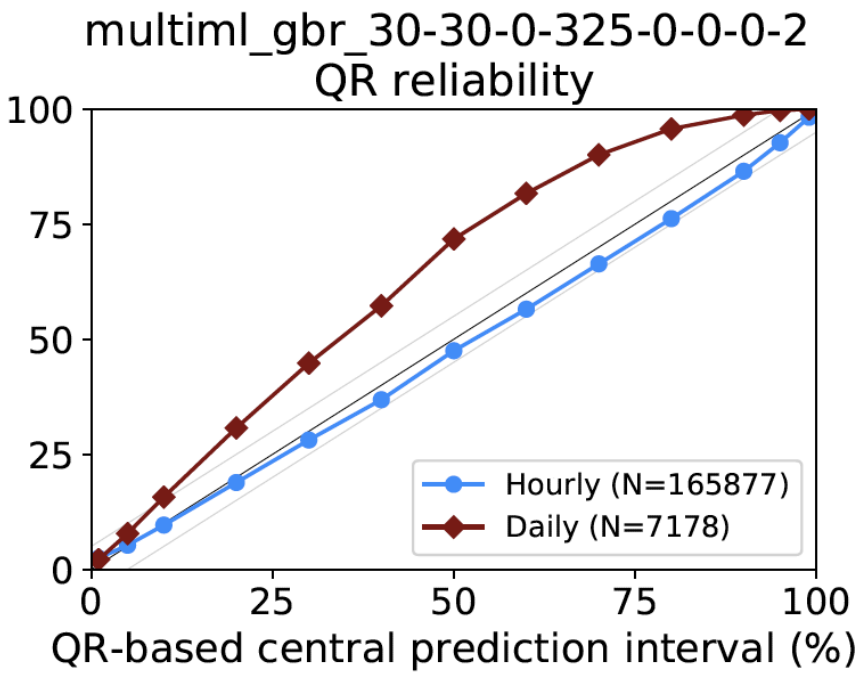




Quantile regression reliability : results on NO2

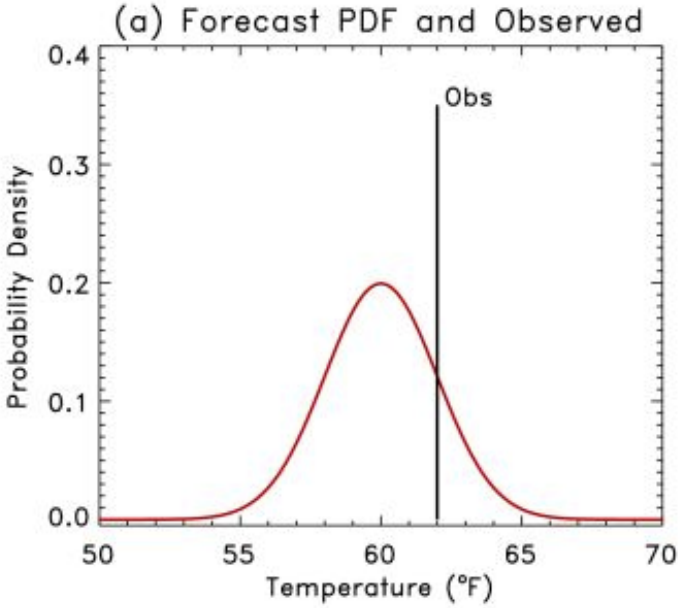
- For hourly concentrations (on which percentiles are fitted) : the QR reliability is reasonably good (although not always perfect)
- For daily concentrations : the QR-based prediction intervals are not well calibrated (they are far too conservative!)
- Better reliability on hourly data with multi-station model

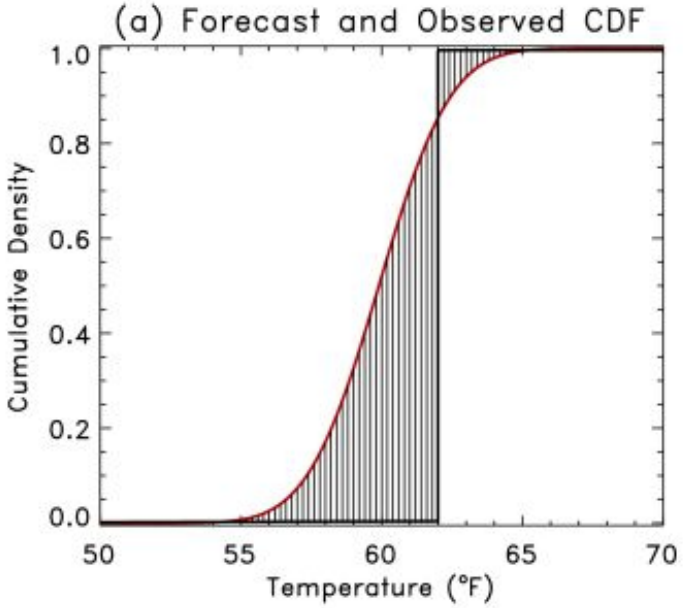




Continuously ranked probability score (CRPS)

- CRPS easily interpretable since generalization of MAE for probabilistic forecasts





Remarks

- Evaluation of probabilistic forecast requires more than a single observations (the more the better)
- Rare events (low probability) require more data
- Desired properties for probabilistic forecast of binary event : reliability (1:1 line best), resolution, sharpness (sharp probability density)
- Reliability diagram : show the full joint distribution of forecasts and observations for probability forecast of a binary predictand (counterpart of conditional quantile plot for nonprobabilistic forecasts of cotinuous predictands

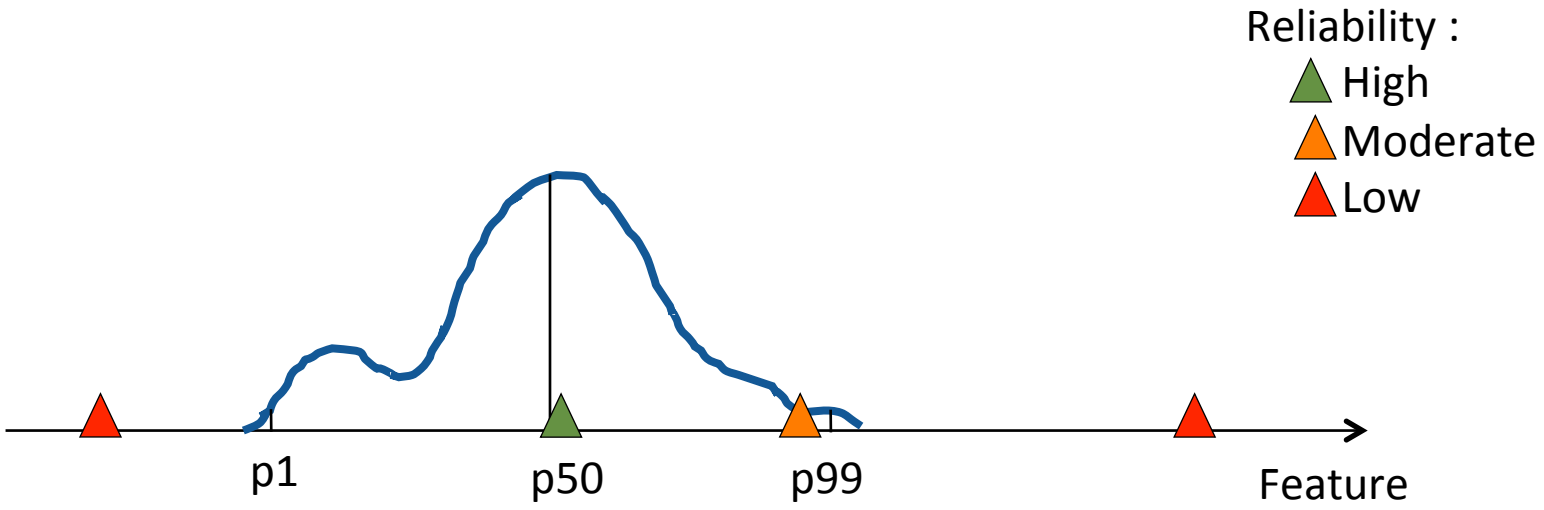
Reliability of the ML prediction and problem of extrapolation

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Degree of extrapolation

- Given a set of features, ML models can always predict a value for the target but what if these features have very different values than in the training set?
- In other words : **how to assess objectively if we are in an extrapolation mode (in which prediction are likely more doubtful)?**

➔ Idea 1 : simple parameter, the degree of extrapolation Γ, that quantifies the position of the features values in the features distribution



This can be done for each feature separately, and then we compute a sum weighted by the features importance

Degree of extrapolation

- Given a set of features, ML models can always predict a value for the target but what if these features have very different values than in the training set?
- In other words : **how to assess objectively if we are in an extrapolation mode (in which prediction are likely more doubtful)?**

➔ Idea 1 : simple parameter, the degree of extrapolation Γ, that quantifies the position of the features values in the features distribution

We introduce the degree of extrapolation Γ^j of a given prediction j defined as :

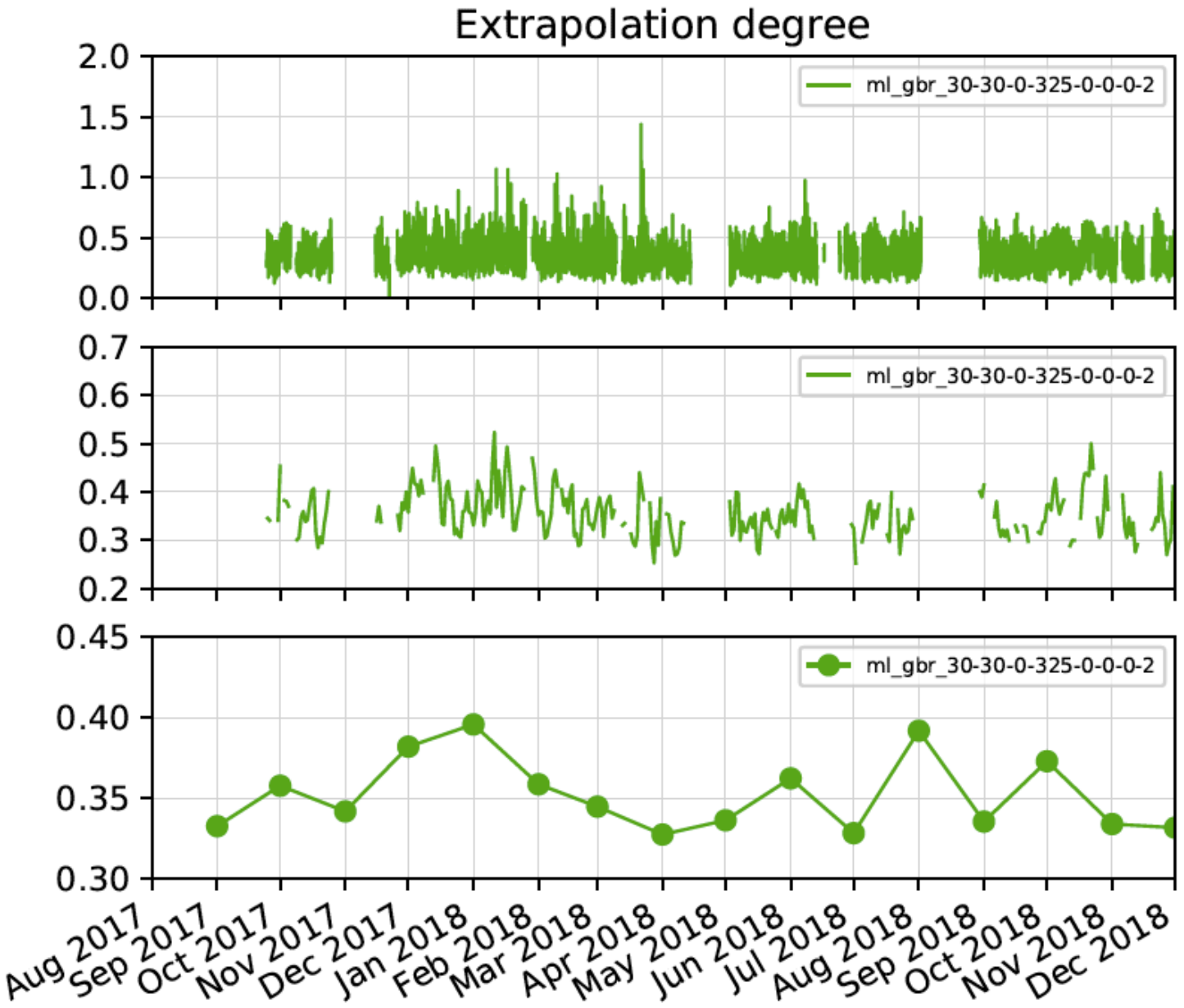
$$\Gamma = \sum_{i=1}^{N} \alpha_i \gamma_i^j \tag{22}$$

with N the total number of features, α_i the importance of feature i (note that $\sum_i^N \alpha_i = 1$), and γ_i^j the degree of extrapolation of the feature value $C^{i,j}$, here defined as :

$$\begin{cases} \gamma_i^j = \dfrac{C_i^j - C_{i,p50}}{C_{i,p99} - C_{i,p50}} \text{ if } C_i^j \geqslant C_{i,p50} & (23) \\ \gamma_i^j = \dfrac{C_{i,p50} - C_i^j}{C_{i,p50} - C_{i,p01}} \text{ if } C_i^j < C_{i,p50} & (24) \end{cases}$$

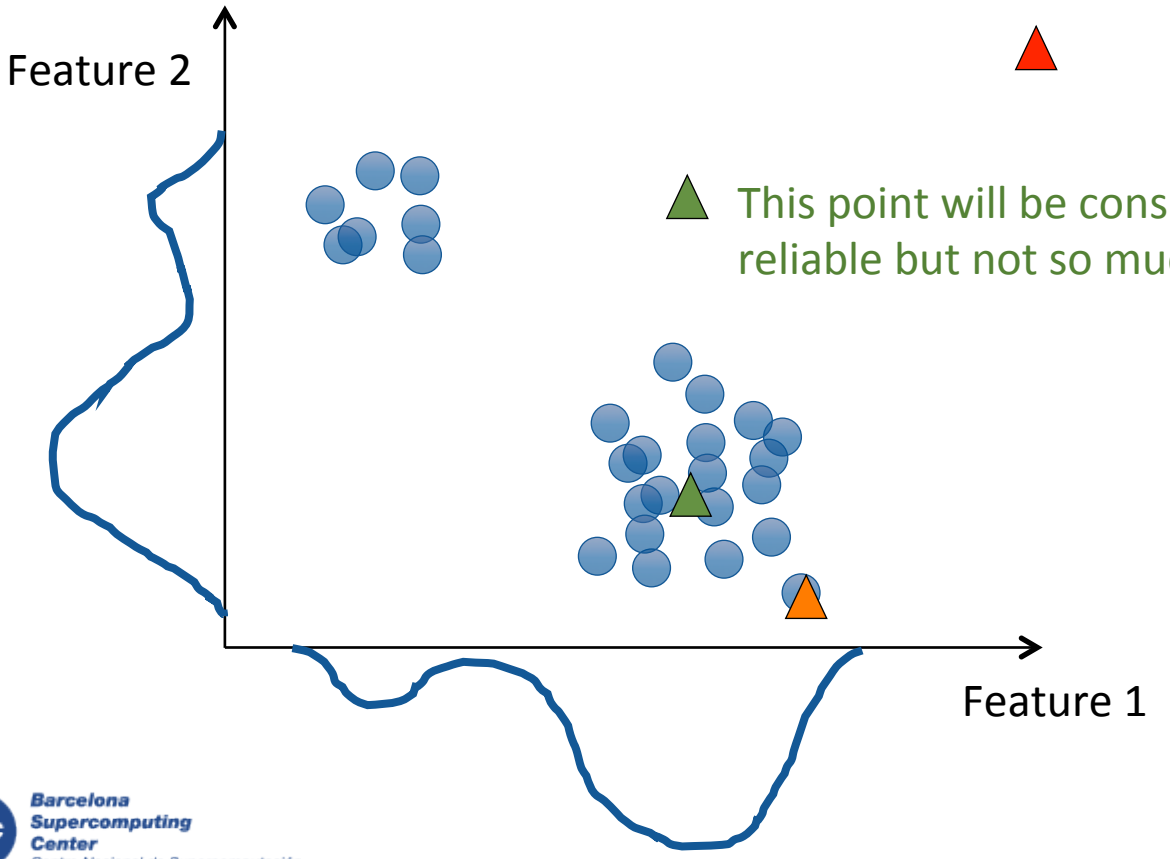
where $C_{i,p01}$, $C_{i,p50}$ and $C_{i,p99}$ are respectively the 1^{st}, 50^{th} and 99^{th} percentiles of the feature i calculated based on the last training dataset.

Illustration for PM10 at FAC station

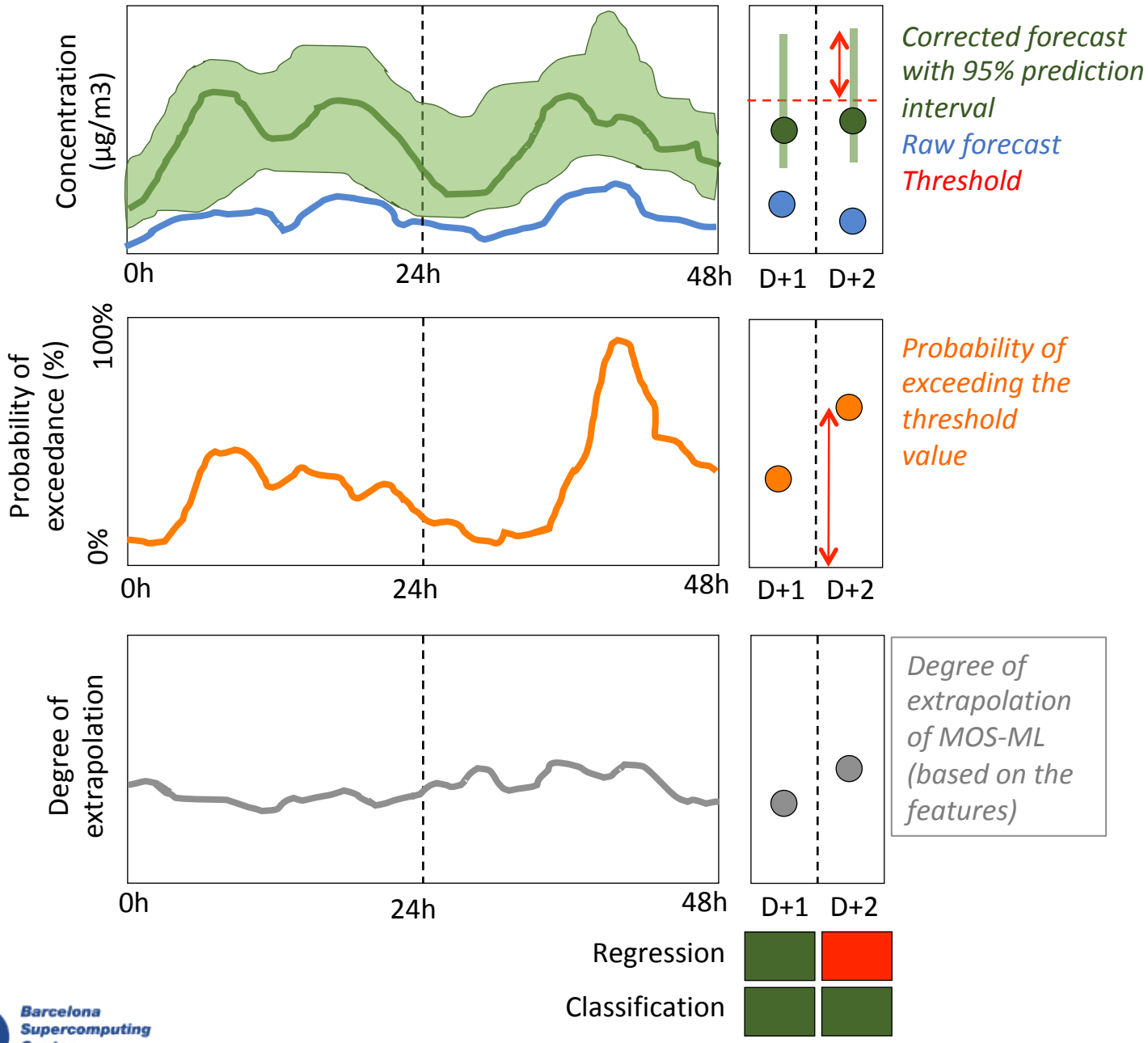


Degree of extrapolation

- Nice, simple, quite easy to interprete…
- …but some limitations, in particular the fact that all dimensions (i.e. all features) are treated separatly



➔ Other ideas to be investigated…


Concentration (µg/m3)
0h
24h
48h
D+1
D+2
Corrected forecast with 95% prediction interval
Raw forecast
Threshold
Probability of exceedance (%)
100%
0%
0h
24h
48h
D+1
D+2
Probability of exceeding the threshold value
Degree of extrapolation
0h
24h
48h
D+1
D+2
Degree of extrapolation of MOS-ML (based on the features)
Regression
Classification

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA

Thank you

herve.petetin@bsc.es

Episode detection metrics

	Event observed	Event not observed
Event forecasted	a	b
Event not forecasted	c	d

- **Error** : ERROR = (b+c)/(a+b+c+d) complement of **Accuracy** : A=(a+d)/(a+b+c+d)
  - ➔ *How many events or non-events are erroneously classified?*
- **Probability of detection** : POD = a/(a+c)
  - ➔ *How many observed events have been well predicted by the model?*
- **Probability of false detection** : POFD = b/(b+d)
  - ➔ *How many observed non-events are erroneously classified as events by the model?*
- **Hit rate** : HR = a/(a+b) complement of **False Alarm Ratio** : FAR = b/(a+b)
  - ➔ *Over all events forecasted by the model, how many are indeed observed?*
- **Critical success index** : CSI = a/(a+b+c)
  - ➔ *If we ignore the (numerous) non-events, how many events are correctly detected?*
- **Bias** : B=(a+b)/(a+c)
  - ➔ *Are we forecasting the correct number of events? (no matter when they occur or if the are correct)*

Ensemble decision-tree based algorithms

- Ensemble decision tree-based algorithms developed for improving the bias-variance trade-off of decision trees, for instance :

  - **Random Forest** (RF) : average a large number of decisions trees fitted (i) over a bootstrap sample of the training dataset and (ii) in which each split is done not based on all features but based on a randomly selected subset of the features

  - **Gradient Booting Machine** (GBM) : sequential ensemble of small decision trees (weak learners; trunks) applied to the initial dataset and the subsequent residuals (in practise : fit a first small tree on the data, compute the residuals, fit a second tree on these residuals, compute new residuals, fit a third tree on these new residuals...)

➔ Much better prediction skills than simple decision trees!