1 Context

The measurement network is not evenly distributed: it is intrinsically linked to emission sources (population, human activities, etc.), and this varies depending on the pollutants and regions. Even without considering topographical configuration, measurement conditions vary greatly between stations, and it is important to take this high degree of spatial heterogeneity in the measurement network into account when using the data for model validation or data assimilation. This raises the issue of spatial representativeness, which is particularly difficult to grasp.

The metadata provided by measurement network installers and available through EEA API is very valuable, but although it is based on rigorous criteria, it remains subjective. Joly and Peuch (2012) showed that this information could be supplemented by a classification of the data itself, based on a sufficiently deep history. This article showed that it was useful to treat each time series separately, as a measurement site is influenced by different emission sources depending on the pollutants.

This classification has been updated annually to take into account the most recent data and is used by the operational services of the European regional CAMS project. Since Joly and Peuch (2012), the method has been completely revised in 2017 (unpublished) in the framework of the CAMS_50 European project, and extended to more species.

> The idea behind the present study is to use new machine learning methods to try to simplify and further improve this classification method.

2 Previous method

Since 2017, the classification method run in operations for the CAMS project comprises the following steps:

1. Data selection: removal of values above fixed thresholds set for each pollutant. Stations above 1,400 metres in altitude are excluded.
2. Calculation of indicators describing the time series (if sufficient data is available). Since the 2017 revision of the method, 14 indicators are considered:
    - to characterise the distribution of values,
    - to characterise variability (high frequency and autocorrelation),
    - to characterise the diurnal cycle,
    - to characterise the annual cycle,
    - to characterise the “weekend effect”.
3. Transformation, normalisation and outliers detection.

- Transformation of indicators with highly asymmetric distribution: automatic technique from Box and Cox (1964).
- Normalisation of indicators.
- Detection of outliers for each group separately, multivariate (supervised learning method named SVM).

4. Principal Component Analysis: reduction in the number of dimensions.
5. Linear Discriminant Analysis: LDA favours the directions that best separate different groups, maximizing inter-group distance and minimising intra-group variance. Since 2017, a "multiple LDA" has been used to separate the four groups R, S, U, and T.
6. Projection onto the first axis and distribution into 10 classes. Since 2017, the proportion of different families of stations is now taken into account when projecting into the 10 classes.

The idea behind the present study is to replace this entire process with a three-step method based on machine learning algorithms:

1. Data pre-treatment
2. Unsupervised detection of outliers
3. Supervised classification

3 Input dataset

As in Joly and Peuch (2012), 8 years of data are considered: 2018-2025 in the present case. All the available data from EEA API has been downloaded (E1a and E2a fluxes), using "real time" data when/where "validated" data where unavailable.

Based on the existing metadata (also available through EEA API), the following groups have been derived:

- Stations with 27° $\leq$ latitude $\leq$ 72° and -32° $\leq$ longitude $\leq$ 46°
  - **T** = traffic urban
  - **U** = background urban
  - **S** = background suburban
  - **R** = background rural
- **X** = others (outside / inside + traffic suburban, traffic rural, industrial *) : not considered

We found questionable ozone data for some stations due to a large number of zero values. Rather than eliminating all zero values for all variables, we decided to keep the series as they were. One of the objectives of the method is precisely to detect automatically abnormal time series.

Missing data is problematic when applying deep learning methods. The following strategy was therefore adopted:

- Gaps $\leq$ 24 consecutive hours are considered small gaps and are interpolated using a PCHIP interpolator (Piecewise Cubic Hermite Interpolating Polynomial), that preserves monotonicity in the interpolation data and does not overshoot if the data is not smooth. The interpolant uses monotonic cubic splines to find the value of new points.
- Gaps $\leq$ 30x24 consecutive hours are considered large gaps. To keep a diurnal cycle, a running mean is computed *for the given hour of the day*, with a window size of 91 values (days), if minimum 20 values are available.

For larger gaps, values are left missing. Each civil year, for each station, is considered separately. Annual time-series with remaining missing values are discarded.

> It is not perfect: interpolation cannot invent missing data, but the results are much more realistic than zero padding, and in any case allow us to integrate a larger number of "complete" time series into our dataset.

4 Outlier Detection

`tsfresh` Python package automatically calculates a large number of time series characteristics, the so called "features". The extracted features can be used to describe the time series. They can also be used to cluster time series and to train machine learning models that perform classification or regression tasks on time series.

The Features Extraction is a reduction dimension that makes possible the Outlier Detection. Two methods from the python library `scikit learn` have been tested: `IsolationForest` and `LocalOutlierFactor`. For both methods, the outliers fraction (optional parameter) is left "automatic".

We found that `IsolationForest` method tends to form a single cluster of data points that exhibit similar behavior in the main features with high covariance (first axes of the PCA). In contrast, `LocalOutlierFactor` method seems more effective at isolating atypical patterns in features with lower variance.

Besides, plotting outlier data on a map showed that `IsolationForest` method tends to group outliers within specific regions (e.g., Northern Italy for ozone, Turkey for PM10, Polland for PM2.5). However, our goal is not to exclude entire regions, but rather to detect erroneous data at a more localized level.

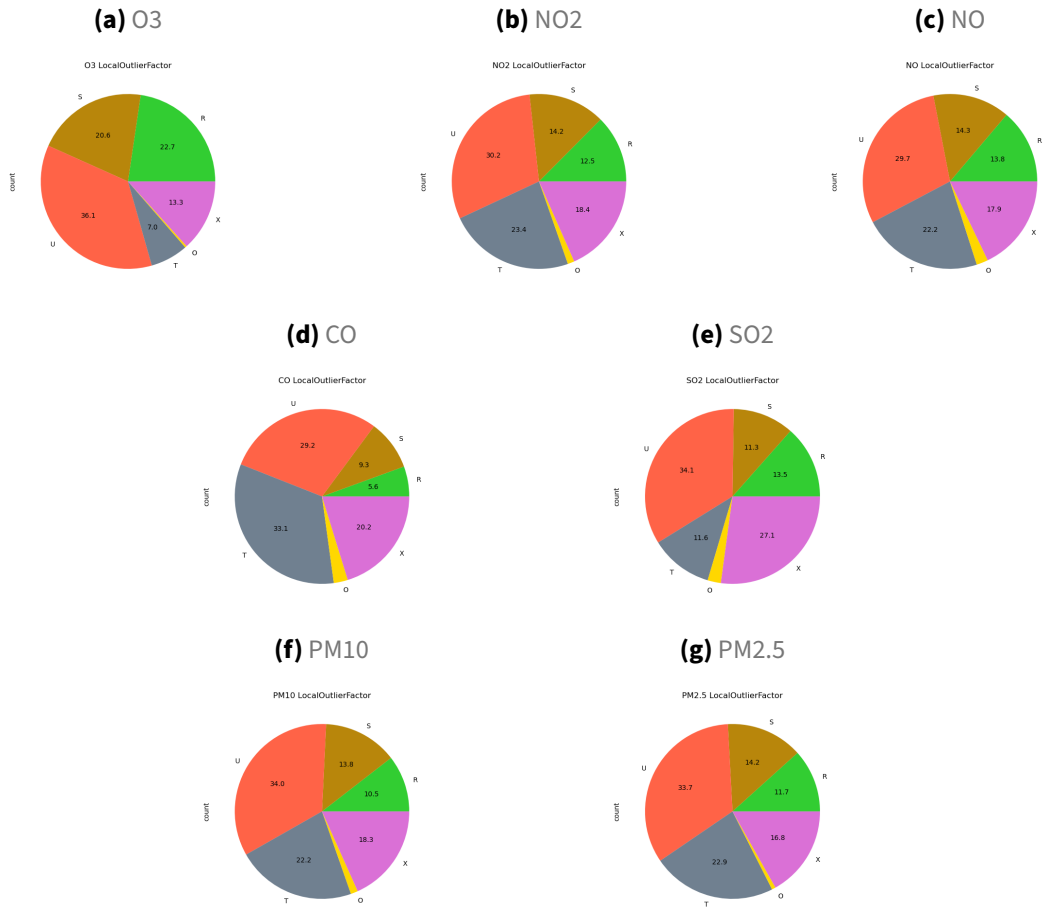


Figure 1: Piechart of the groups obtained with "LocalOulierFactor" method

> Due to these results, `LocalOutlierFactor` method appears to be more suitable.

Figure 1 illustrates the data typology after outlier detection. The groups are defined as in Section 3, with the addition of “O” representing outliers. The fraction of outlier data is lowest for O3 and highest for CO, but remains reasonable (a few percents).

> In conclusion, the outlier detection process removes only a relatively small proportion of the available data. Nonetheless, the typology of the input data varies considerably depending on the pollutant, especially O3, CO, and SO2.

5 Classification

We use `tsai` which is an open-source `Python` deep learning package built on top of `Pytorch` & `fastai` focused on state-of-the-art techniques for time series tasks like classification, regression, forecasting, imputation, etc.

5.1 Statistical scores

A bunch of scores are used in the following to monitor the performance of the process:

- `accuracy`: fraction of correct predictions.
- `precision`: ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.
- `recall`: ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.
- `f1 score`: harmonic mean of the precision and recall.
- `roc_auc`: Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
- `corrcoef`: Matthews correlation coefficient (MCC). The Matthews correlation coefficient is used in machine learning as a measure of the quality of binary and multiclass classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction. The statistic is also known as the phi coefficient.
- `smooth accuracy`: performance indicator defined for our purpose. Weighted accuracy based on the normed (over the rows) confusion matrix. The weights (cf. matrix 1) are maximum on the diagonal, but they are not zero on either side, which allows for greater tolerance for small differences between predicted and true values.

$$\begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \tag{1}$$

5.2 Preliminary study

5.2.1 Methodology

The purpose of the preliminary study is to develop a method that yields satisfactory results with our dataset. To do this, the data is divided into two parts:

- **"High-potential (HP) data"**: the measurement sites we trust the most. To do this, we compare the metadata and previous objective classifications obtained for the last three years with the method currently used in operations for the CAMS project. We select stations with a good coherence with metadata:
  - metadata="R" and $1 \leq$ class ≤ 4
  - metadata="S" and $3 \leq$ class ≤ 6
  - metadata="U" and $5 \leq$ class ≤ 8
  - metadata="T" and $7 \leq$ class ≤ 10
- **"Low-potential (LP) data"**: remaining measurement sites, for which there is no consistency, or no stability in the relationship between metadata and objective classifications.

The new classification method will be implemented (selection of the algorithm and its parameters) by performing training on HP data and validation on LP data.

5.2.2 First selection of algorithms

The following algorithms have been compared with a fixed learning rate and batch size:

- InceptionTime (Fawaz, 2019)
- XceptionTime (Rahimian, 2019)
- MiniRocket (Dempster, 2021)
- ResCNN - 1D-ResCNN (Zou , 2019)
- ResNet - Residual Network (Wang, 2016)
- FCN - Fully Convolutional Network (Wang, 2016)
- mWDN - Multilevel wavelet decomposition network (Wang, 2018) : 4 levels
- LSTM-FCN (Karim, 2017), 2 flavours: default, shuffle: False
- LSTM (Hochreiter, 1997a), 6 flavours: varying layers number 1 to 3, and bidirectional True/False

At this stage, the algorithms that gave the best performances were: InceptionTime, XceptionTime, Minirocket, ResCNN, ResNet, mWDN.

5.2.3 Tuning of the parameters

The 6 selected algorithms are now tested with different combinations of learning rate (lr) and batch size (bs), and 140 epochs:

- bs = 64, lr = 1e-4
- bs = 32, lr = 1e-4
- bs = 64, lr = 1e-3
- bs = 32, lr = 1e-3
- bs = 64, lr = 1e-2
- bs = 32, lr = 1e-2

Table 1 gives the selected combinations of algorithms and parameters, for each species.

Species	Algorithm	Batch Size	Learning Rate
O3	ResNet	32	1e-4
NO	XceptionTime	64	1e-3
NO2	InceptionTime	32	1e-3
CO	InceptionTime	32	1e-3
SO2	InceptionTime	32	1e-3
PM10	InceptionTime	32	1e-2
PM2.5	ResNet	32	1e-3

Table 1: Best combination of algorithms and parameters

5.3 Applying the classification

5.3.1 Methodology

The annual time-series (both HP and LP) are now distributed into 10 subsets with similar properties. Using the algorithms and hyperparameters previously selected (cf. table 1), the method is applied to predict the classes of one subset, based on the data from remaining 9 subsets (cf. figure 2).

At this stage, each measurement is assigned 1 to 8 different predicted labels. A weighted average is calculated in order to give more importance to the most recent time series.

5.3.2 Results

Figures 3, 4, and 5 show a marked improvement with this new method, especially for S and U labels. The confusion matrix is much more diagonal. Best results are obtained for NO2, NO, and O3. Figures 6

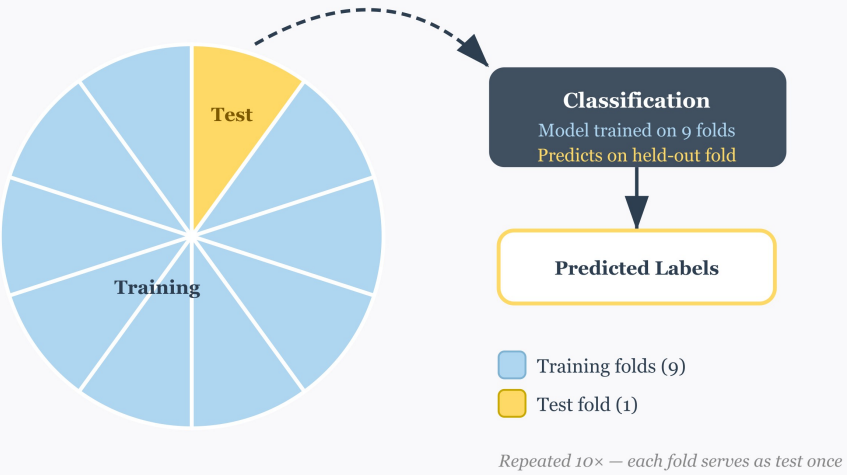


Figure 2: Illustration of the k-fold classification

and 7 confirm that performances are lower for SO2 and PM2.5, but significantly improved compared to the previous method. On that point of view, the improvement is quite remarkable for PM10.

5.3.3 Impact for use in the CAMS project

In the CAMS project, only the background sites are selected, according to the metadata. Then, for operational tasks of assimilation and verification, the sites with classes 1-7 (i.e. predicted R, S, or U, with the new method) are selected, according to the objective classification.

Species	Background	LDA Rejected	ML Rejected
O3	2111	198 (9%)	30 (1%)
NO	1618	262 (16%)	97 (6%)
NO2	2257	377 (17%)	138 (6%)
CO	574	196 (34%)	78 (14%)
SO2	1240	318 (26%)	76 (6%)
PM10	1873	512 (27%)	160 (9%)
PM2.5	1242	359 (29%)	148 (12%)

Table 2: Among Background sites (according to metadata), number of sites rejected (outlier or predicted T) with the old LDA method, and with the new ML method.

Table 2 shows that the number of rejected sites is much lower with the new classification (inferior to 15%). Figures 10, 11, and 12 show that some regions are much better represented with the new classification (e.g., Turkey for SO2 and PM10).

The impact for use in the CAMS project seems thus positive.

Figure 3: Comparison of the old *versus* new classifications: O3, NO2, NO. Confusion matrices of the predicted *versus* True labels.

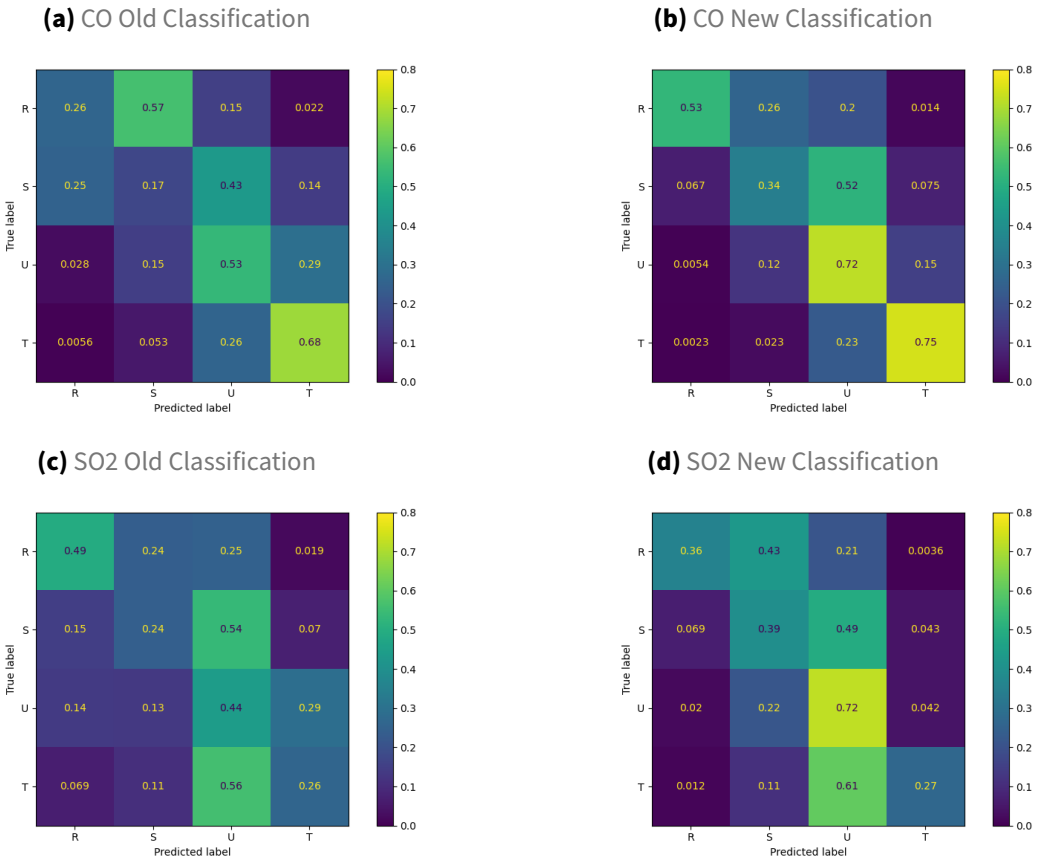


Figure 4: Comparison of the old *versus* new classifications: CO, SO2. Confusion matrices of the predicted *versus* True labels.

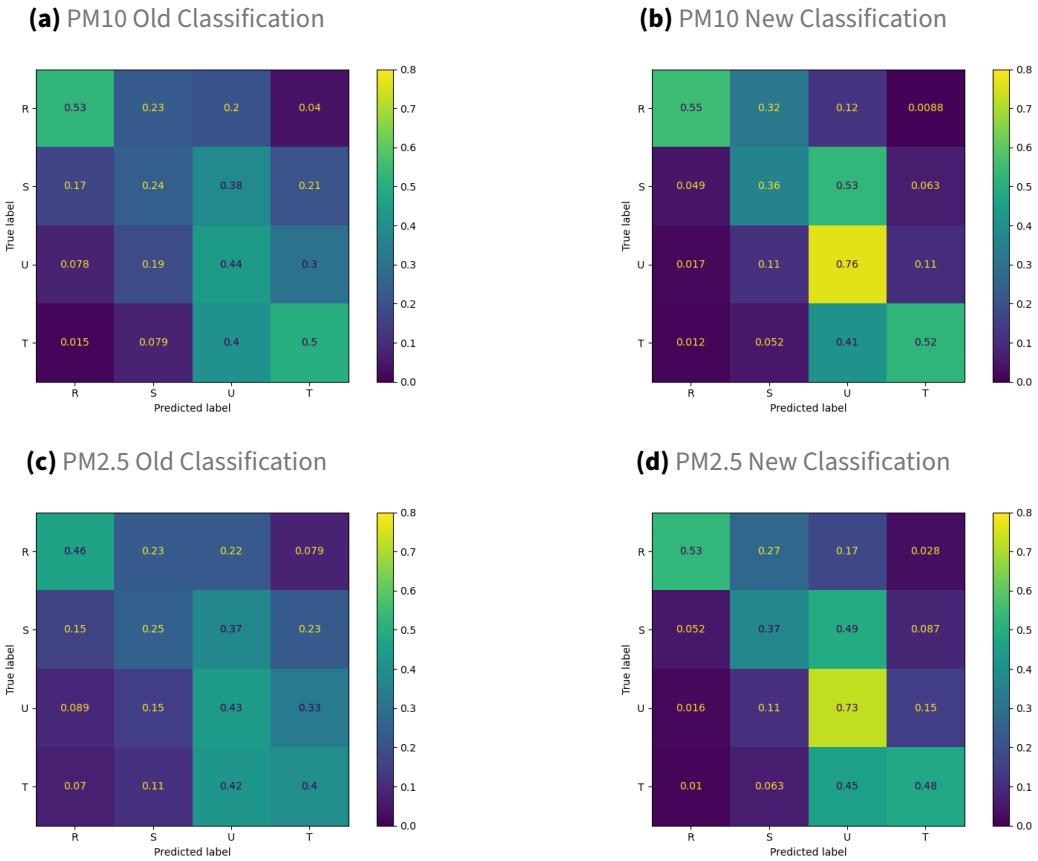


Figure 5: Comparison of the old *versus* new classifications: PM10, PM2.5. Confusion matrices of the predicted *versus* True labels.

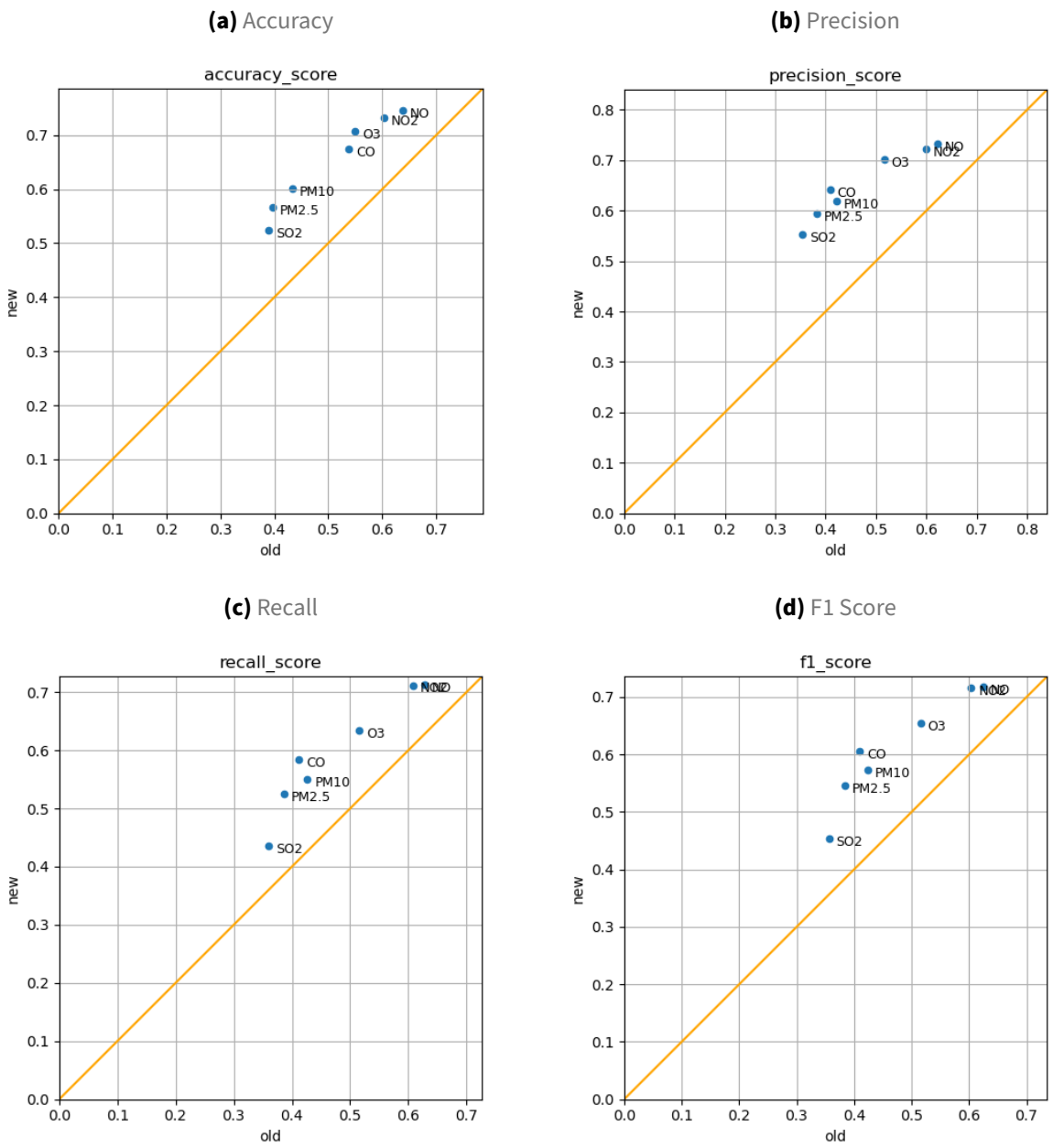
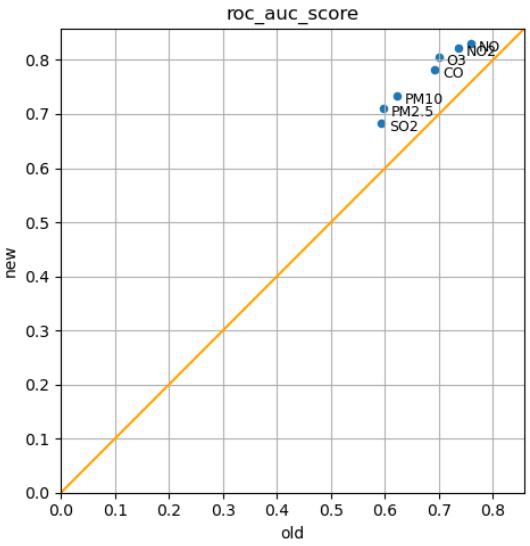
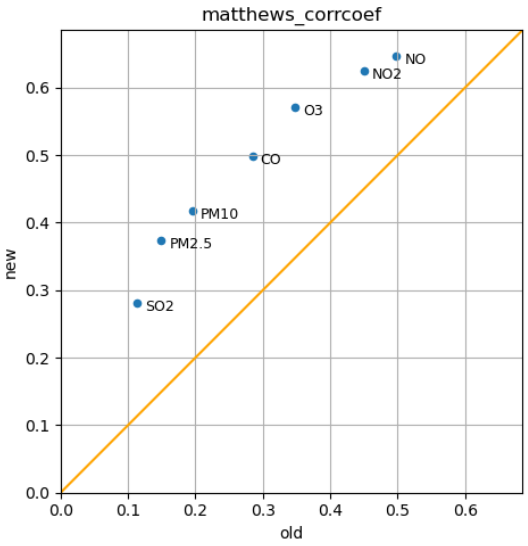


Figure 6: Comparison of the old *versus* new classifications. Scores defined in section 5.1.

(a) ROC AUC



(b) Matthews Correlation



(c) Smooth Accuracy

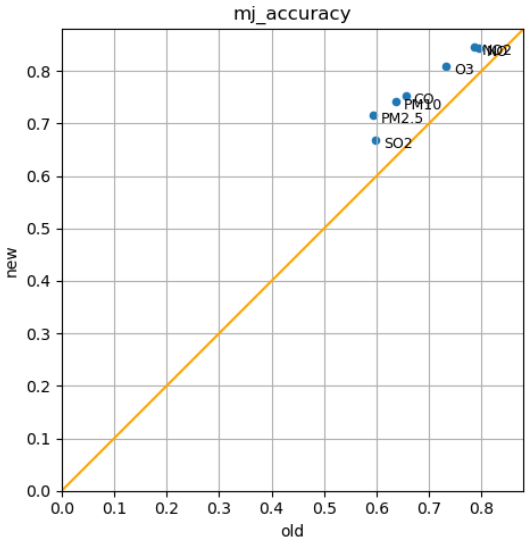


Figure 7: Comparison of the old *versus* new classifications. Scores defined in section 5.1.

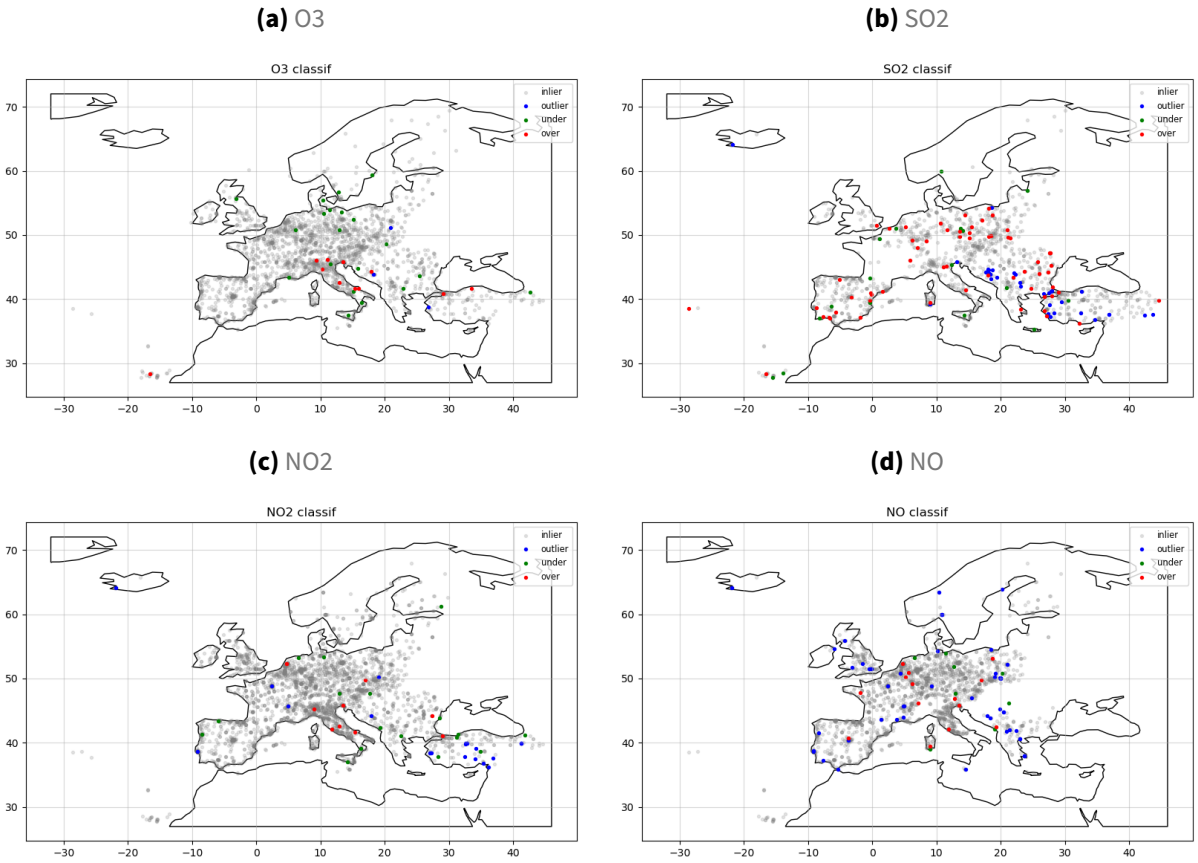
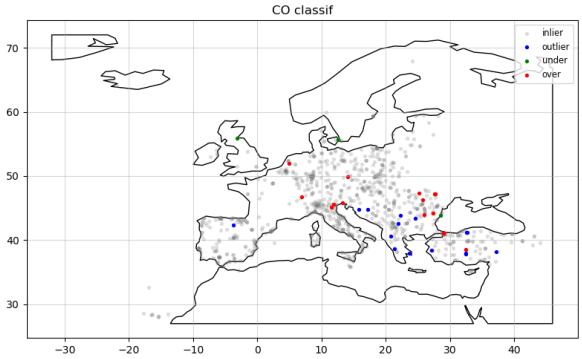
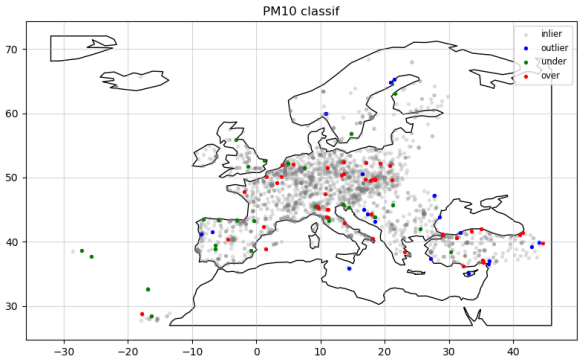


Figure 8: Map of the main discrepancies between the predicted *versus* true labels: O3, SO2, NO2, NO. *over* is for R sites predicted U or T, and *under* is for U or T sites predicted R.

(a) CO



(b) PM10



(c) PM2.5

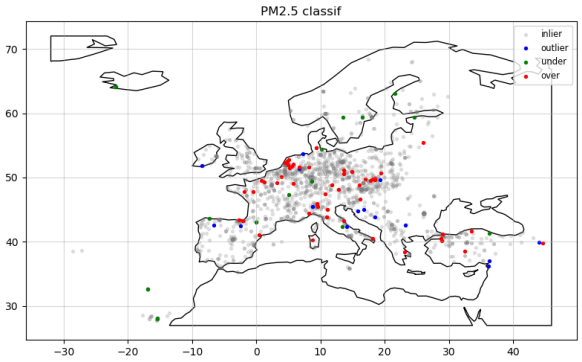
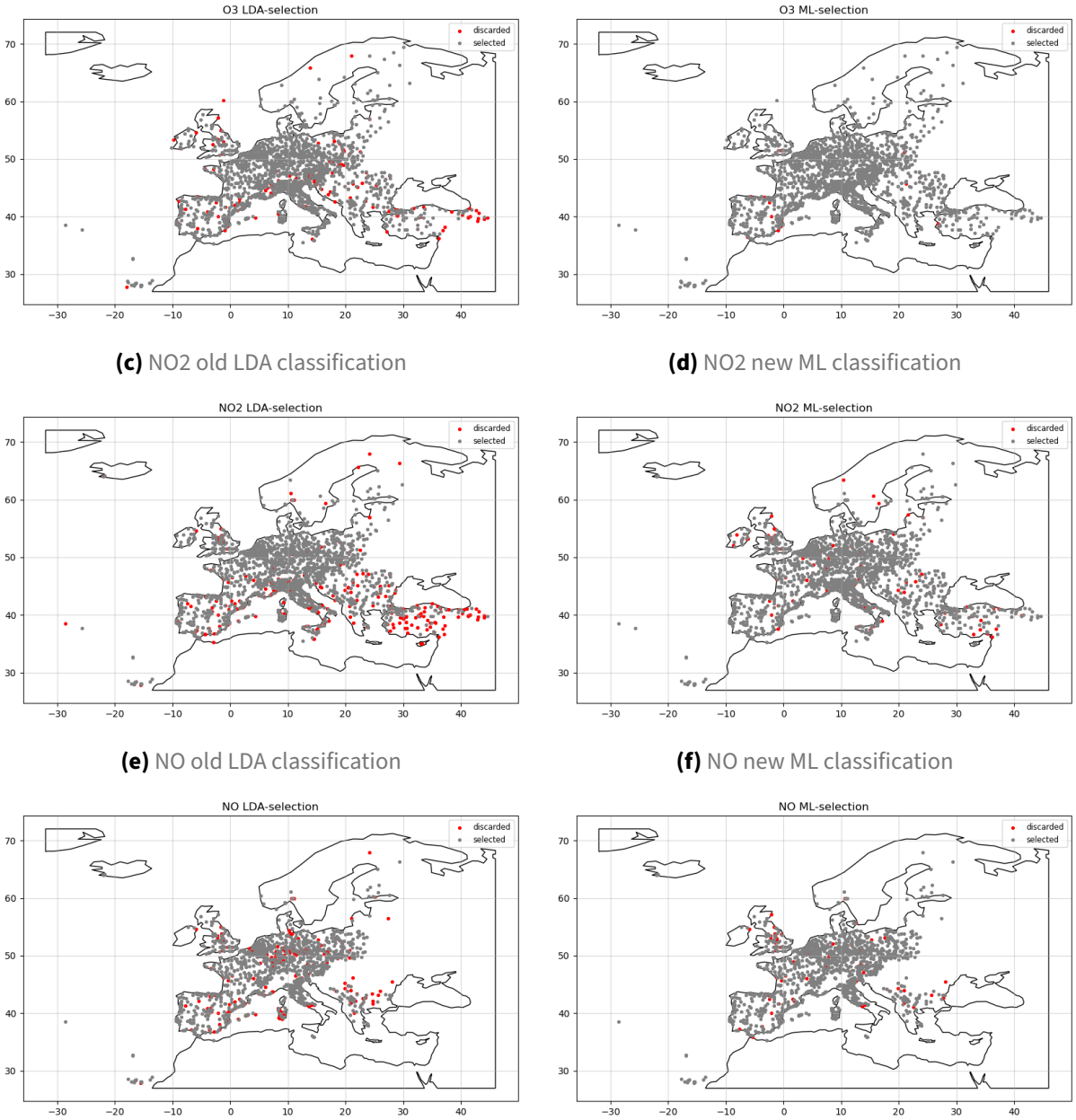


Figure 9: Map of the main discrepancies between the predicted *versus* true labels: CO, PM10,PM2.5. *over* is for R sites predicted U or T, and *under* is for U or T sites predicted R.

(a) O3 old LDA classification

(b) O3 new ML classification



(c) NO2 old LDA classification

(d) NO2 new ML classification

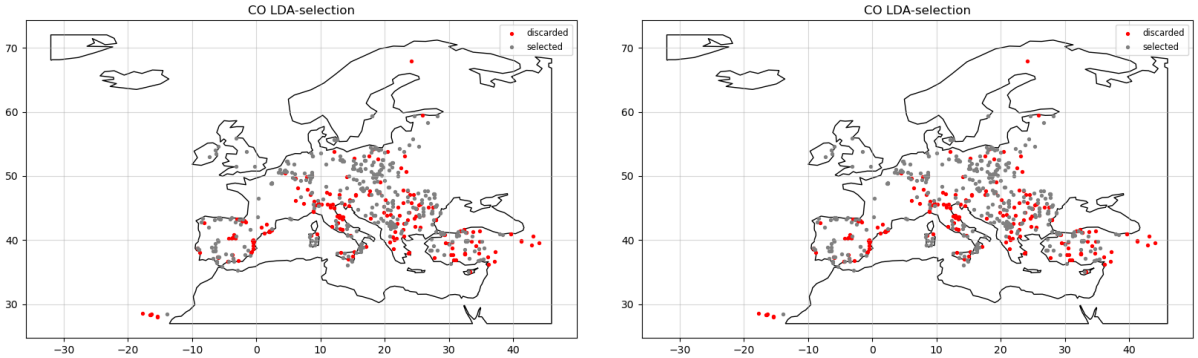
(e) NO old LDA classification

(f) NO new ML classification

Figure 10: Localization of Background sites (according to metadata) selected for CAMS, or rejected (outlier or class >7 according to the objective classification): O3, NO2, NO

(a) CO old LDA classification

(b) CO new ML classification



(c) SO2 old LDA classification

(d) SO2 new ML classification

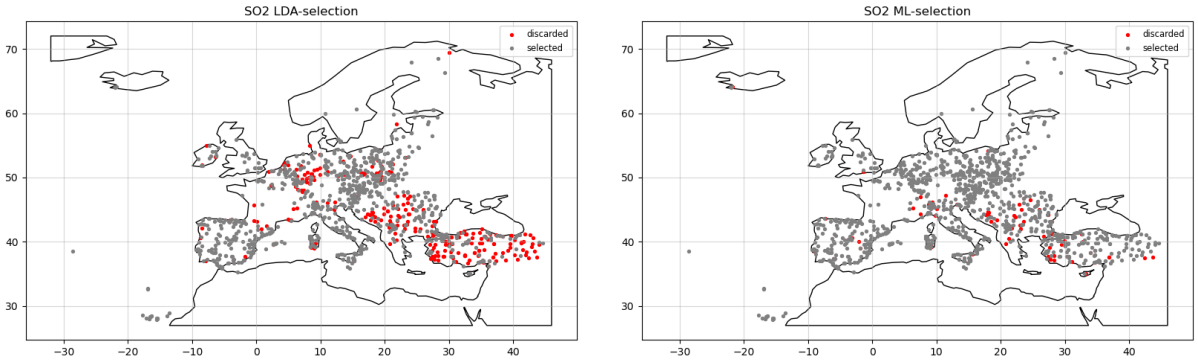
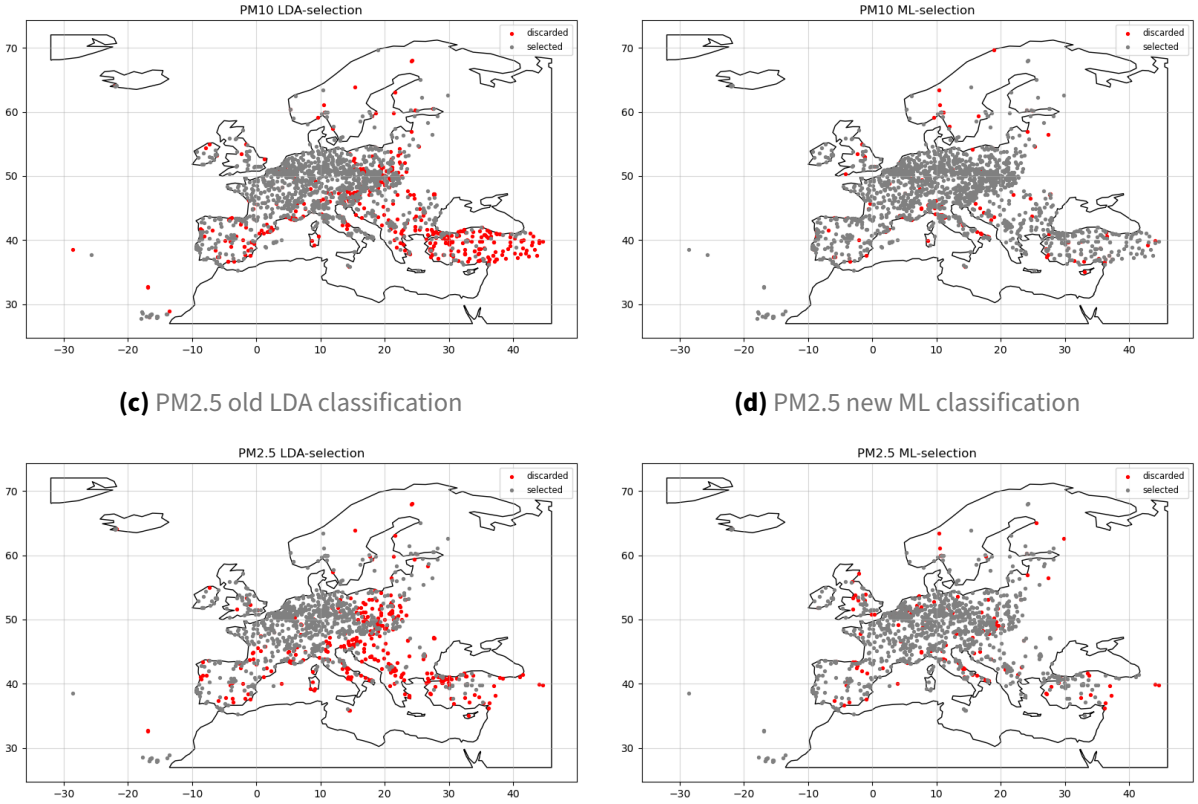


Figure 11: Localization of Background sites (according to metadata) selected for CAMS, or rejected (outlier or class >7 according to the objective classification): CO, SO2

(a) PM10 old LDA classification

(b) PM10 new ML classification



(c) PM2.5 old LDA classification

(d) PM2.5 new ML classification

Figure 12: Localization of Background sites (according to metadata) selected for CAMS, or rejected (outlier or class >7 according to the objective classification): PM10, PM2.5