Data Challenge: Validation Aid for Air Quality Microsensor Data.

Submitted By:

Jules EXBRAYAT |Henrique LEFUNDES DA SILVA |Amrin AKTER |Aynur ASI

Supervised By:

Phlypo RONALD |Louhichi SANA |Yann FORTIER

Grenoble INP - Ensimag, UGA

Grenoble INP - Phelma, UGA

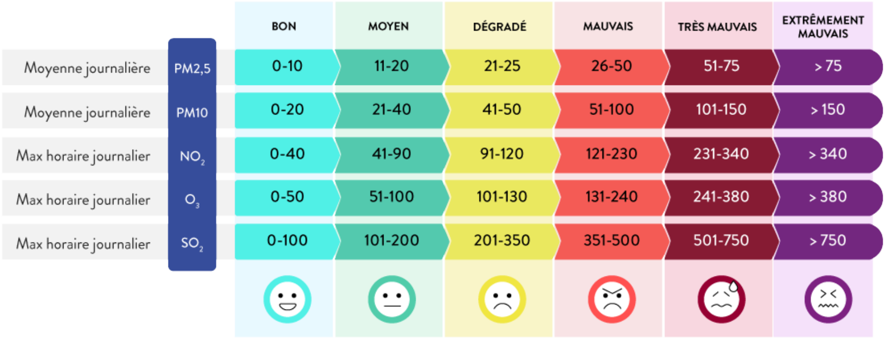
19th January, 2025

Contents

1 Introduction

Air quality monitoring plays an important role in understanding and addressing different environmental challenges, especially in urban and industrial areas. The Atmo Auvergne-Rhône-Alpes observatory, which oversees air quality in one of France's most diverse regions, has long relied on a network of regulatory measurement stations to produce reliable data. However, with the advancement of compact and affordable microsensors, the potential to enhance monitoring capabilities has grown noticeably. These microsensors have been set across various locations, offering the opportunity to collect more granular data than before. However, their inherent variability and susceptibility to environmental factors create challenges for ensuring data reliability.

To provide clear communication of air quality levels to the public, Atmo utilizes the **ATMO Index**, a color-coded system that categorizes air quality into six levels: *Bon*, *Moyen*, *Dégradé*, *Mauvais*, *Très Mauvais*, and *Extrêmement Mauvais*. This system is widely used across France and helps citizens (*Suivre et améliorer la qualité de l'air*, n.d.) understand air pollution levels in real time. The ATMO Index is depicted in Figure 1, which illustrates its intuitive color legend.



		BON	MOYEN	DÉGRADÉ	MAUVAIS	TRÈS MAUVAIS	EXTRÊMEMENT MAUVAIS
Moyenne journalière	PM2,5	0-10	11-20	21-25	26-50	51-75	> 75
Moyenne journalière	PM10	0-20	21-40	41-50	51-100	101-150	> 150
Max horaire journalier	NO_2	0-40	41-90	91-120	121-230	231-340	> 340
Max horaire journalier	O_3	0-50	51-100	101-130	131-240	241-380	> 380
Max horaire journalier	SO_2	0-100	101-200	201-350	351-500	501-750	> 750

Figure 1: ATMO Index Legend (ATMO France, 2024)

The 2024 Data Challenge aims to address the challenges posed by microsensors in three stages. The first stage involves analyzing the variability of microsensor measurements against co−located regulatory analyzers. The second stage focuses on developing methods to validate (or invalidate) microsensor data by using comparisons with these regulatory analyzers. Finally, the third stage extends the validation framework to microsensors deployed in locations without co−located regulatory analyzers. We aim to integrate the technical, environmental, and geographical data to suggest some validation frameworks.

2 Data Description and Preprocessing

The dataset provided for this challenge is comprehensive and includes multiple sources of air quality and environmental data collected over two years in the Grenoble area. It encompasses measurements from 40 microsensors, three regulatory stations, and supplementary meteorological data. The data is structured to facilitate comparisons between microsensors and reference analyzers, enabling the development of validation frameworks.

The microsensor data includes hourly measurements of particulate matter (PM2.5), along with meteorological data such as temperature and relative humidity. And these can be used to understand the variability and behavior of microsensors under different environmental conditions. Additionally, measurements from three regulatory stations — Saint Martin d'Hères, Les Frênes, and Rocade Sud — provide validated reference values for pollutants like particulate matter (PM2.5). These reference data serve as a benchmark for evaluating microsensor performance.

Feature Engineering

Target Variable: We engineered the target variable for tree-based models by computing the difference between the microsensor measurement and the reference measurement of PM2.5. For classification tasks, we created a categorical target variable, where a value of 1 indicates that the PM2.5 difference exceeds a specified threshold (e.g., 3 µg/m³), and 0 otherwise.

Explanatory Variables: We generated categorical time-based variables (hour, month, season) to assess the probability of the microsensor providing inaccurate measurements. These variables enable the model to capture patterns based on specific times of the day or year.

Scaling: To normalize the range of the explanatory variables, we applied a min-max scaler, which transforms the values to fall between 0 and 1.

One-Hot Encoding: We one-hot encoded categorical variables to make them suitable for logistic regression. To reduce dimensionality, time variables with many categories, such as the hour of the day (24 categories), were grouped into broader time periods (night, morning, afternoon, evening) before encoding.

3 Methodology and Performance Analysis

3.1 Breakdown of Trends

In this section, we worked with the PM2.5 concentration data for the Saint-Martin-d'Hères station over the years 2023 and 2024, as shown in figure 2, to better understand the data. The safe limit for PM2.5 in France is 25 µg/m³, as indicated in Figure 1. This limit is represented by the red dashed line in Figure 2. To improve visualisation and exclude the extreme value from December 2024, we set the y-axis limit to 0-100 µg/m³.

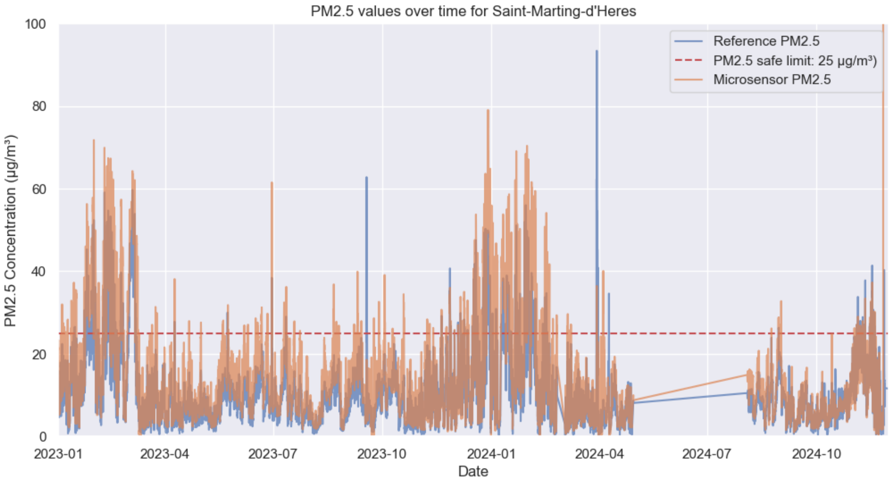


Figure 2: PM2.5 values for both the Reference and Microsensor stations at Saint-Martin-d'Hères over the years 2023 and 2024

From Figure 2, we observe that PM2.5 concentrations often exceed the safe limit, with 7.57% of the reference sensor values and 15.47% of the microsensor values crossing this threshold. We noticed that the microsensor reports exceedances more frequently. Between May 2024 and August 2024, we observed flat lines for both sensors, which might be due to issues in data collection or unusually consistent readings. Aside from this, the data show periodic spikes and dips, indicating seasonality and trends.

In Figure 3, we presented the autocorrelation plot to analyse temporal dependencies, showing strong initial correlations at lower lags that decrease sharply, indicating short-term dependencies in PM2.5 levels. The similar autocorrelation patterns for both sensors suggest that the microsensor effectively captures the temporal behavior of the reference sensor. The rise and fall of the autocorrelation values suggest potential seasonality, likely driven by daily cycles influenced by environmental conditions. Based on these findings, we identified the need to further explore seasonal patterns and consider applying time series models.

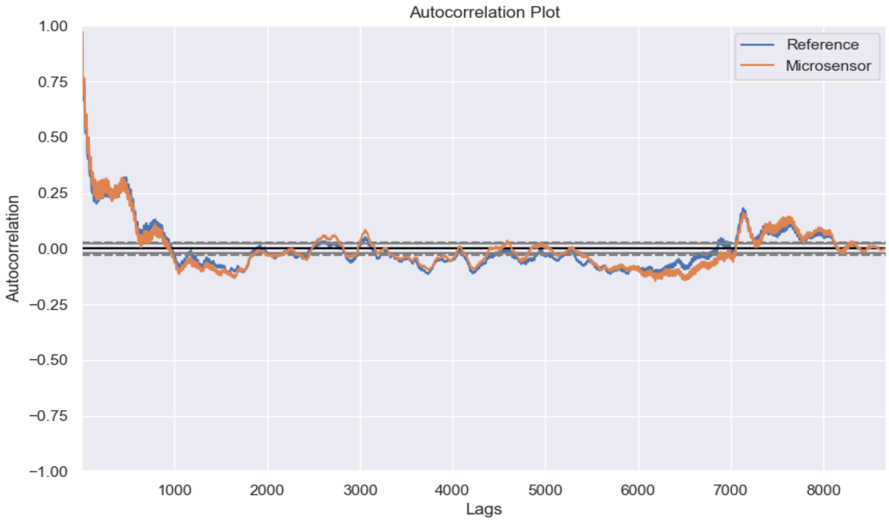
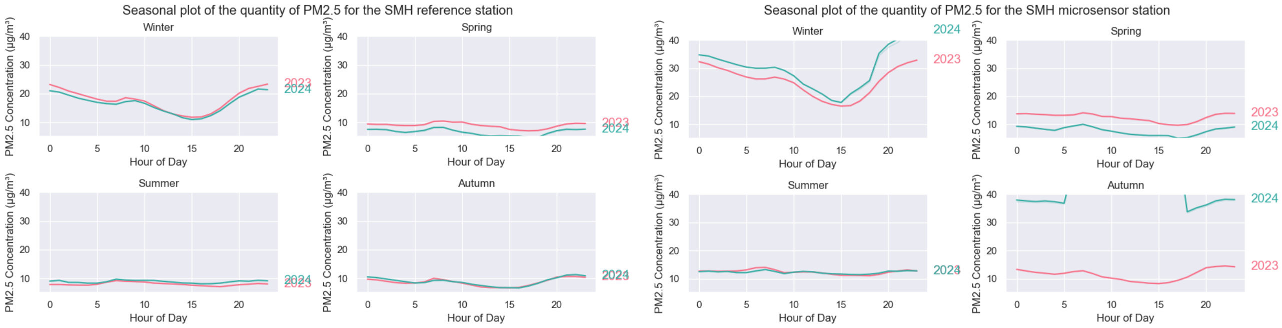


Figure 3: Autocorrelation plot of the reference and microsensor station

Exploring Seasonal Patterns

In this section, we present the seasonal plots in Figure 4, showing hourly PM2.5 concentration trends across seasons (Winter, Spring, Summer, Autumn) for 2023 and 2024. These plots compare data from the Saint-Martin-d'Hères reference and microsensor stations.



(a) Seasonal plot for the Reference station

(b) Seasonal plot for the Microsensor station

Figure 4: Seasonal plots for the Reference and Microsensor stations at Saint-Martin-d'Hères for the years 2023 and 2024.

The y-axis represents the mean PM2.5 concentration for each hour, averaged across all days in the season and year. For example, hour = 7 in Spring 2023 reflects the average PM2.5 at 7 AM for all days in Spring 2023. A consistent y-scale is used for both stations.

For the **reference station**, we observe that PM2.5 levels peak in winter, with the highest concentrations at midnight (0:00) and a sharp rise in the evening around 20:00, and levels drop during the day, reaching their lowest around 15:00 PM. This trend is consistent for both 2023 and 2024. In spring, PM2.5 levels are lower but remain steady throughout the day, with slight increases around 7:30 AM and in the evening, though 2023 shows slightly higher peaks. Summer shows the best air quality, with low and stable PM2.5 levels. In autumn, PM2.5 levels slightly increase in the morning around 7:30 AM and in the evening, and the trends for both 2023 and 2024 are closely aligned, showing consistent air quality patterns.

The **microsensor station** captures similar seasonal trends but with more variability. In winter, sharper decreases at 15:00 and stronger evening peaks are evident, especially in 2024, with levels exceeding 40 µg/m³. During the day, measurements fluctuate more compared to the reference station. In spring, trends align with the reference station but show larger differences between 2023 and 2024. Summer readings are stable and similar to the reference station. In autumn, significant differences are observed between 2023 and 2024, particularly in the morning and afternoon, with some 2024 values exceeding 40 µg/m³. This variability and the larger differences in autumn are more pronounced compared to the reference station. Overall, the seasonal patterns indicate that PM2.5 levels are higher during winter and autumn and lower during spring and summer.

Analyzing Environmental Influences on PM2.5 Concentrations

Figure 5 illustrates the relationship between PM2.5 concentrations and two environmental factors: temperature and humidity.

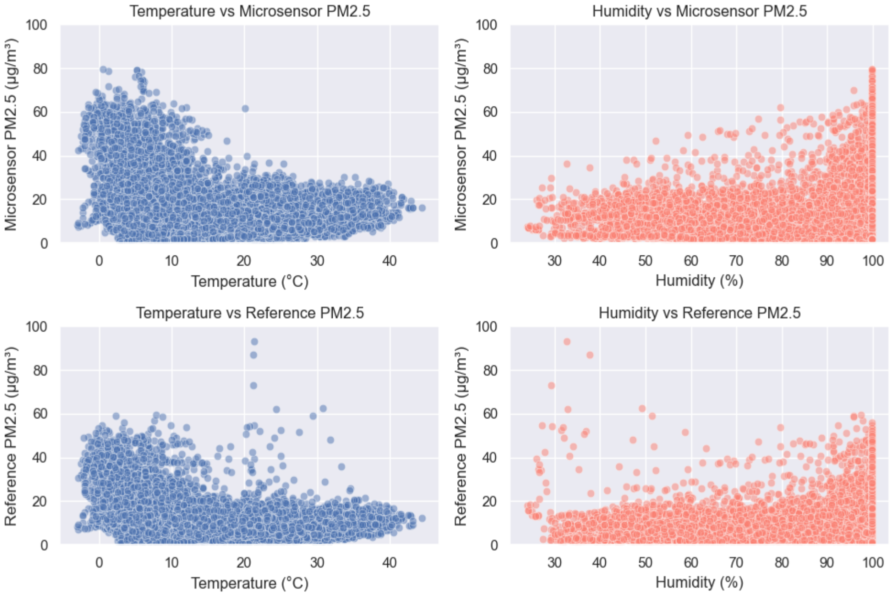


Figure 5: Scatter plots showing the relationship of PM2.5 concentrations with temperature and humidity for the Reference and Microsensor stations at Saint-Martin-d'Hères

The scatter plots on the left depict PM2.5 levels against temperature, while those on the right show PM2.5 levels against humidity for both the microsensor and reference stations. We have limited the y-axis scale to 0–100 µg/m³ to make the patterns clearer and avoid outliers from distorting the overall trends.

For both the microsensor and reference stations, we observe an inverse relationship between temperature and PM2.5 concentrations. Higher PM2.5 levels are concentrated at lower temperatures, particularly below 10°C, where most values cluster between 20 and 60 µg/m³. As the temperature increases above 10°C, PM2.5 concentrations gradually decrease, indicating better air quality at higher temperatures. This pattern is consistent across both sensors, reflecting the impact of reduced atmospheric dispersion and increased emissions during colder conditions.

On the other hand, the relationship between humidity and PM2.5 concentrations shows a positive correlation for both sensors. As humidity increases, PM2.5 concentrations tend to rise gradually. A dense cluster of high PM2.5 levels is observed near 100% humidity, likely due to the accumulation of particulate matter under high moisture conditions. While both sensors capture this trend, the microsensor data exhibits slightly more variability compared to the reference station, indicating potential differences in measurement sensitivity.

3.2 ARIMA Models for Time Series Forecasting

The AutoRegressive Integrated Moving Average (ARIMA) model is extensively used in the field of time series forecasting and was first introduced by Box and Jenkins, 1970. This model has become one of the most popular techniques in forecasting due to its flexibility in handling various types of time series data, including environmental pollutants.

3.2.1 Model Specification

ARIMA models are specified by three parameters: p, representing the number of autoregressive components (AR); d, the number of differencing operations required to achieve stationarity (Integrated, I component); and q, which denotes the number of lagged forecast errors in the prediction equation (moving average components, MA). The mathematical formulation of the ARIMA model is given by:

$$\nabla^d y_t = \alpha + \sum_{i=1}^{p} \phi_i \nabla^d y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \tag{1}$$

where $\nabla^d y_t$ represents the differenced series with d orders of differencing, y_t is the value of the time series at time t, α is a constant or the mean of the time series, ϕ_i are the coefficients of the autoregressive terms, θ_j are the coefficients of the moving average terms, and ε_t is the white noise error term.

To use an ARIMA model for forecasting, the time series data must first be checked for trends or seasonal changes and made stationary through differencing. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are then analyzed to determine the appropriate values for p (autoregressive terms) and q (moving average terms). Maximum Likelihood Estimation (MLE) is employed to find the best values for these terms, guided by the Akaike Information Criterion (AIC). The model is validated by comparing its forecasts against actual observations and checking the residuals for any patterns, confirming its accuracy and reliability for future predictions.

3.2.2 Training and Validation

Using the Saint Martin d'Hères reference, the PACF suggests two significant autoregressive components, while the ACF shows the data is highly correlated, with a slight peak after 24 lags, which is comprehensible as the data tends to repeat the same pattern throughout the day.

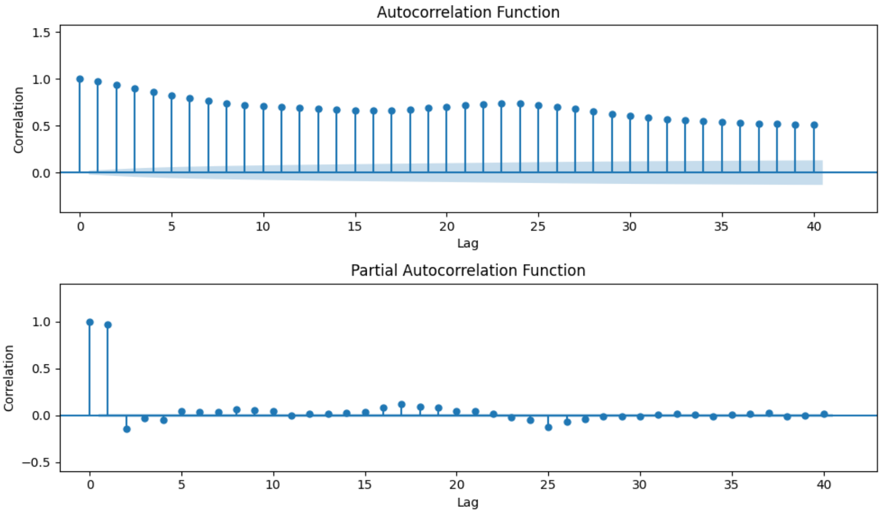


Figure 6: Partial autocorrelation and autocorrelation.

Although the analysis of PACF and ACF may show the ARMA behavior of the time series, the best model was done automatically by the library `pmdarima` with the function `auto_arima` giving the first four months of 2023. The algorithm searches for the combination that minimizes the Akaike information criterion. The algorithm suggested p = 2, d = 1, q = 1, and for the seasonal parameters (1, 0, 0) while keeping a seasonality of 24 samples (24 hours).

When forecasting, the ARIMA model follows closely the values but with one sample delay. Yet, the idea of validation persists, looking for the points in the microsensor that fall within the confidence interval. Due to the lag this validation is not done properly, as can be seen in Fig. 7, but the lag can be removed by hand in this case.

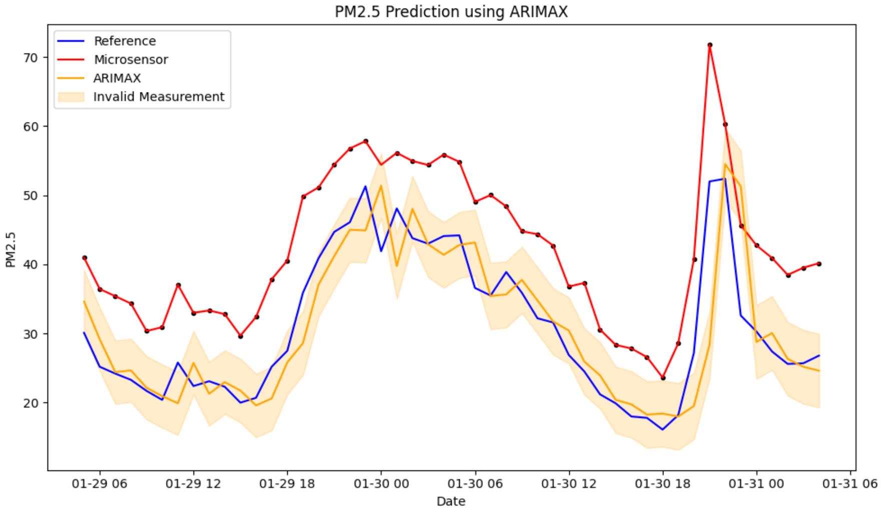


Figure 7: ARIMA Forecast using sliding window (N = 672 hours) and one step ahead prediction.

The results for the microsensor can be improved by a Ridge Regressor calibration, to remove the systematic bias that often disturbs the measure. Using the same sliding window, the bias and coefficient can be estimated and used to correct the next microsensor acquisition, at the same time it's validated or not. By implementing the calibration and removing the lag we get a better validation algorithm (Fig. 8) with $R^2 = 0.98$ for the ARIMA model. Removing the systematic error, the model would consider only four points invalid.

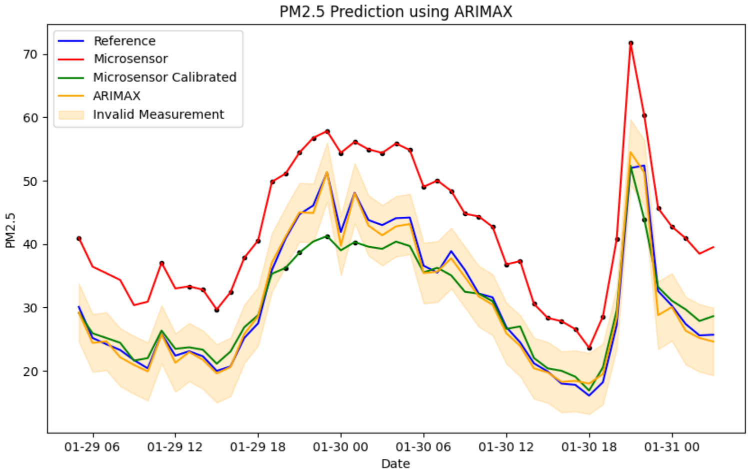


Figure 8: ARIMA Forecast with lag corrected. The calibration window is the same as the ARIMA window.

As the ARIMA model does not properly fit the data without lag, other methods, more advanced were searched, such as the Temporal Fusion Transformer (TFT). It would overcome the flaw found in the ARIMA model, which is the difficulty of adding auxiliary variables apart from the actual time series. The TFT model showed promising results, but given its size, implementation complexity, data handling, and time management, the results could not be explored in detail.

3.3 Anomalies Detection Using Autoencoder Neural Network

We implemented an anomaly detection framework using autoencoder neural networks. Autoencoders are unsupervised learning models designed to compress input data into a lower-dimensional representation and reconstruct it, minimizing reconstruction errors (Hinton and Salakhutdinov, 2006). The main goal is to reduce the data's dimensionality while retaining its essential features.

3.3.1 Model Architecture

The autoencoder consists of the below key components:

Encoder: The encoder is the first part of the autoencoder. It compresses the input data into a smaller, dense representation. The encoder typically consists of one or more layers of neurons that gradually reduce the dimensions of the input data. The final output of the encoder is a vector, known as the ”latent space” or ”bottleneck,” which represents a compressed version of the input data.

Latent Space: This is the compressed representation that captures the essential information about the input. It lies at the bottleneck of the network, where the dimensionality is the smallest. The size and quality of the latent space are crucial for the autoencoder's performance in capturing important features.anomaly detection performance.

Decoder: The decoder takes the latent representation and tries to reconstruct the original input data. The decoder network essentially performs the inverse of the encoding process, attempting to reconstruct the data from the compressed representation as accurately as possible.

Loss Function: The loss function measures the difference between the original input and the reconstructed input. A common loss function is Mean Squared Error (MSE), which calculates the squared difference between the input and the output. The network is trained to minimize this loss.

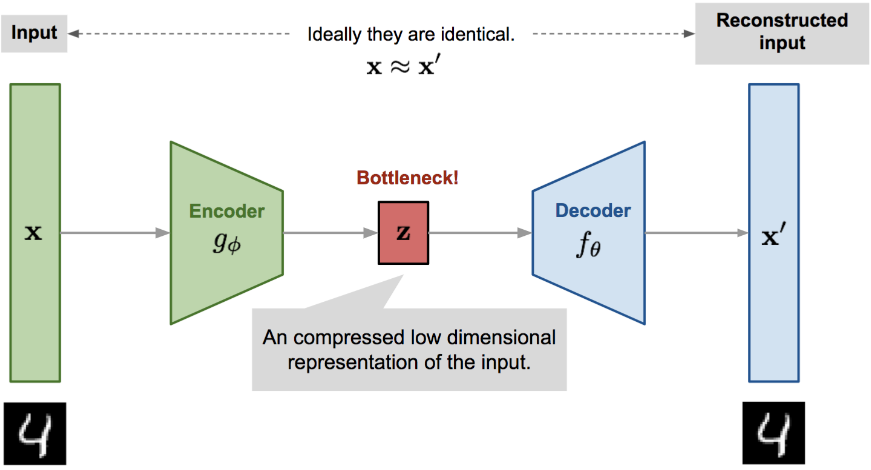

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 9: Structure of the Autoencoder used for anomaly detection.(Source: Weng, 2018).

Anomalies can be detected by measuring the reconstruction error. If the model cannot accurately reconstruct certain inputs, these inputs may be anomalous. After encoding, the decoder can be used to reconstruct the data from its compressed form. This is helpful in tasks like denoising or anomaly detection, where you may want to compare the reconstructed data to the original data to find discrepancies.
The model was implemented using the TensorFlow library and trained on data from reference stations, using PM2.5 concentrations, temperature, and humidity as input features. Temporal attributes, such as year and month, were excluded to avoid introducing noise.

3.3.2 Training and Validation

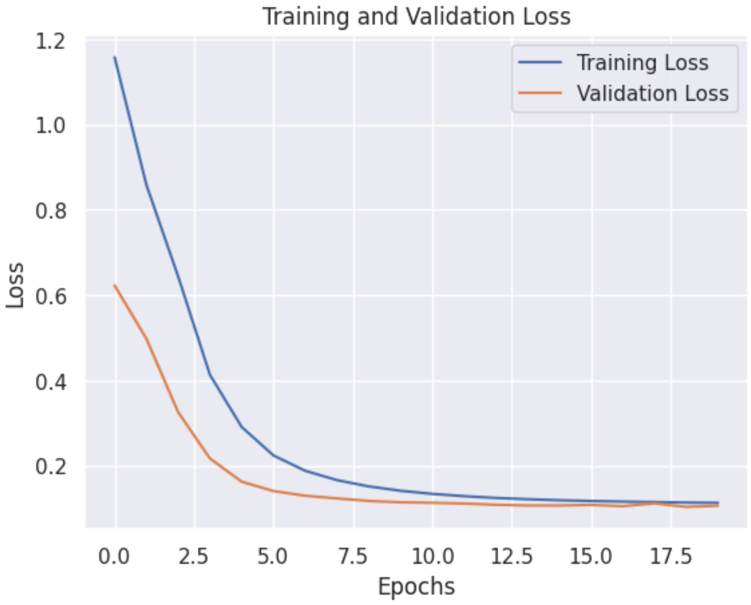


Figure 10: Training and validation loss curves for the autoencoder.

The training process aimed to minimize the Mean Squared Error (MSE) between the input and reconstructed data. Key observations from the training phase include:

- Both training and validation loss curves converged (see figure 10) without overfitting, indicating the model generalized well to unseen data.
- The validation loss stabilized around epoch 15, confirming optimal training.
- The latent space exhibited a Gaussian-like distribution, suggesting meaningful feature extraction.

3.3.3 Anomaly Detection Results

The model was tested on microsensor data, and anomalies were identified in figure 11 based on high reconstruction errors:

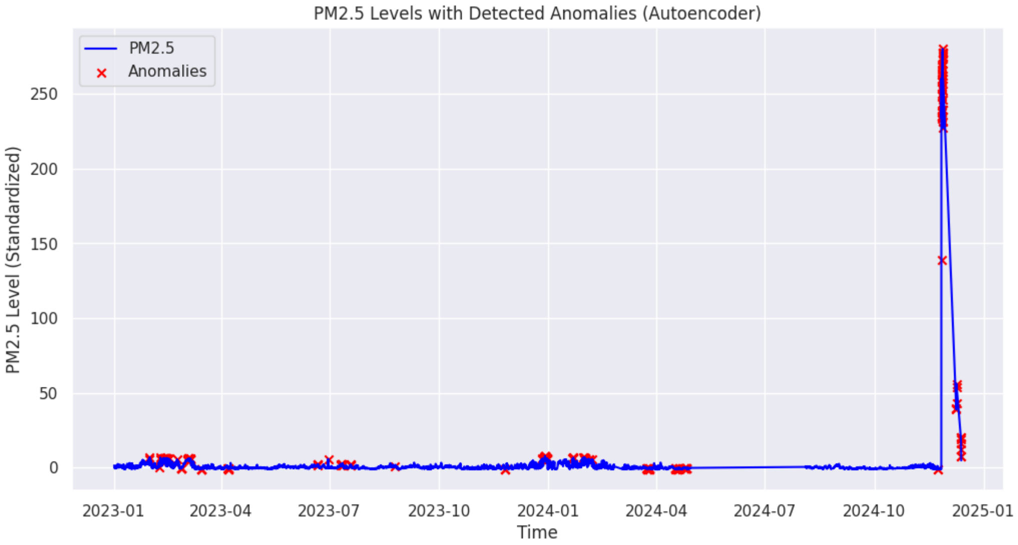


Figure 11: Detected anomalies in microsensor data using autoencoder reconstruction errors.

- **Significant Outliers:** A prominent spike near the dataset's end was detected as an anomaly, likely corresponding to a genuine outlier due to sudden pollutant level increases.
- **Minor Deviations:** Smaller anomalies at lower PM2.5 levels were also flagged, potentially caused by sensor malfunctions or legitimate deviations from normal patterns.

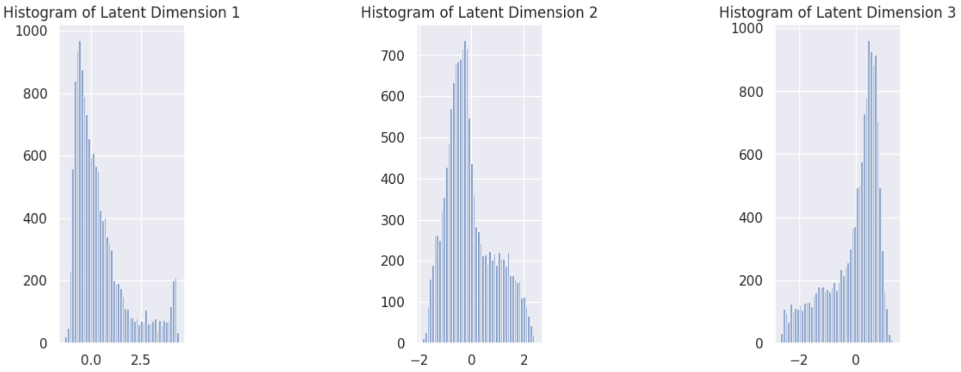


Figure 12: Histograms of reconstruction errors for anomaly detection.

The histogram of reconstruction (see figure 12) errors showed a near-normal distribution centered around zero.

This pattern is desirable and suggests that latent dimensions have converged well. A balanced, Gaussian-like distribution typically indicates that the encoder is learning meaningful and stable features.

3.4 Decision Tree Classifier

3.4.1 Motivation

The decision tree classifier was chosen for its high explainability and the control it offers over model training. Notably, we can limit the maximum number of nodes (i.e., decision rules), ensuring the model's output remains interpretable and avoiding overfitting. A smaller tree, such as one with three nodes, helps capture essential relationships without unnecessary complexity. Decision forests were excluded from this report as they aggregate multiple decision trees, making the model harder to interpret, without significantly improving results compared to a single tree.

Since decision trees are supervised learning algorithms, we formulated the problem as a binary classification task, requiring labeled data. We labeled data as correct when microsensor measurements were within 3 µg/m³ of the PM2.5 reference value, and incorrect otherwise.

By analyzing the decision rules (see Figure 16), we aim to understand how factors like season, time of day, and meteorological conditions affect microsensor reliability.

3.4.2 Training

The training features included season, time of day, the microsensor's PM2.5 average for the previous half day, temperature, and humidity. Data was collected from the Saint-Martin-d'Hères and Les Fresnes stations in 2023. Given the nature of our classification task, failure prediction, the dataset is unbalanced. Fortunately, microsensors are mostly correct, with only 17% of measures being incorrect. Due to the limited amount of data, we used the Synthetic Minority Over-sampling Technique (SMOTE, Chawla et al., 2002) to balance the dataset.

Data was scaled and one-hot encoded, as detailed in Section 2.

A grid search with cross-validation optimized parameters such as the maximum number of nodes (set to 3) and the minimum impurity decrease. We focused on the impurity decrease parameter, as it prevents overfitting and overly complex decision rules that might obscure the actual relationships between context and microsensor validity. The optimal parameter value found was 0.03. The formula for impurity decrease (ΔI) is:

$$\Delta I = I_{\text{parent}} - \left(\frac{N_{\text{left}}}{N_{\text{parent}}} I_{\text{left}} + \frac{N_{\text{right}}}{N_{\text{parent}}} I_{\text{right}} \right)$$

Where:

- I_{parent} is the impurity of the parent node,
- N_{left} and N_{right} are the number of samples in the left and right child nodes,
- N_{parent} is the number of samples in the parent node,
- I_{left} and I_{right} are the impurities of the left and right child nodes.

After the grid search, a final training was performed, and the results are presented below.

3.4.3 Result

The results presented here show the model's predictions on the test set after final training. Predictions were considered correct if the predicted probability exceeded 0.7, ensuring a balance between the false positive and false negative rates (see Figure 14).

Model	Precision	Accuracy	Recall	False Positive Rate
Dummy Classifier	0.8219	0.4925	0.4942	0.5466
Decision Tree	0.9478	0.6485	0.6090	0.1615

Table 1: Performance metrics for decision tree model and dummy classifier.

The decision tree's precision is notably higher, improving by 12% over the uniform dummy classifier. The model excels in minimizing the false positive rate, which is crucial for detecting incorrect measures. However, this leads to a higher false negative rate, lowering both accuracy and recall.

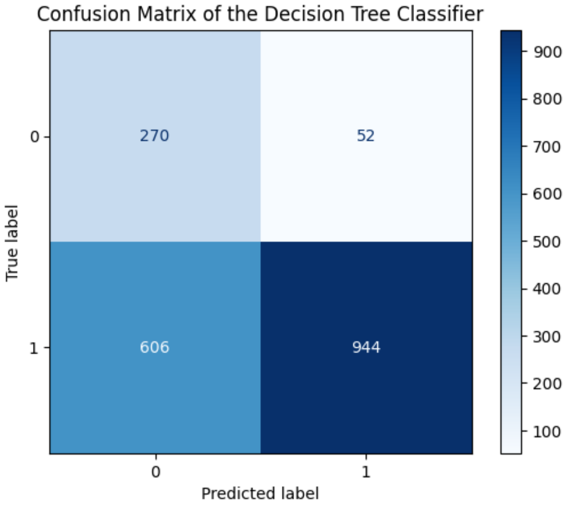


Figure 13: Confusion matrix for the decision tree model.

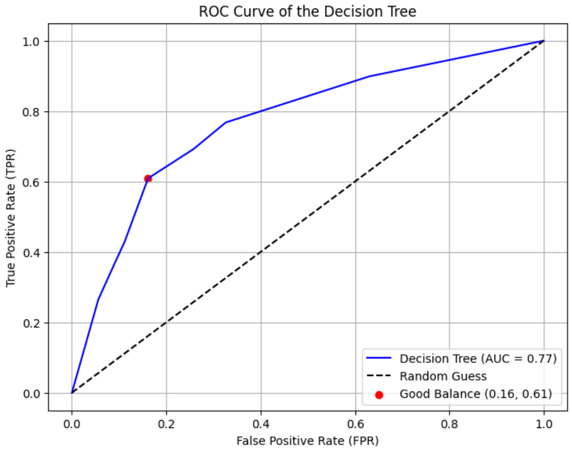


Figure 14: ROC curve for the decision tree model. The red dot indicates the chosen false positive rate corresponding to a probability threshold of 0.7.

The most important feature is the average PM2.5 level over the last half day, followed by temperature. Humidity and the spring season dummy variable have lower importance, and the model did not use other features. This behavior was expected since the number of nodes was limited to three. Increasing the number of nodes did not improve results and reduced interpretability.

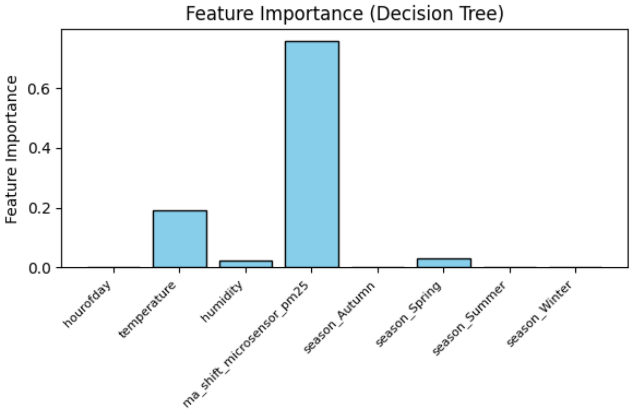


Figure 15: Feature importance, computed as the sum of the impurity decrease of each node where the feature is used.

Figure 16 shows the decision rules of the model. Each node displays the following information (from top to bottom):

- The Gini score,
- The sample size for the node,
- The number of observations falling into each category,
- The predicted label.

The darker the node color, the more distinct the classification.

The decision tree suggests that microsensors fail more often when PM2.5 levels are high, temperature is low, and humidity is high. These conditions typically occur in anticyclonic weather, where low wind and high pressure trap moisture in the lower atmosphere.

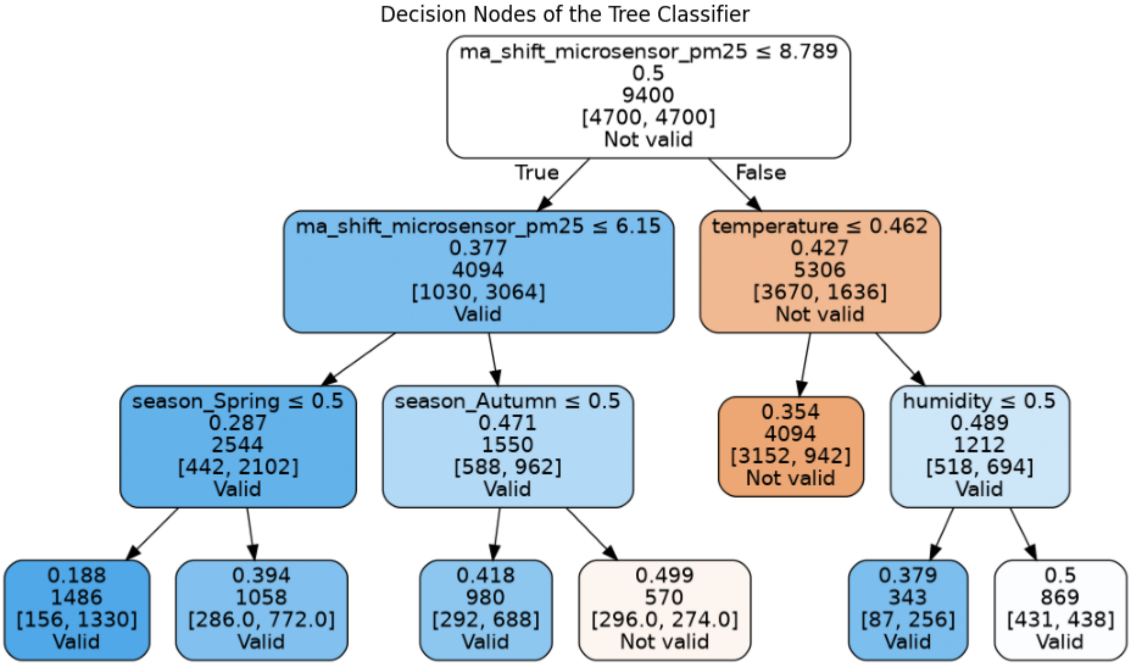


Figure 16: Decision tree rules of prediction.

3.5 Kriging: A Statistical Method

3.5.1 Motivations

Kriging is a statistical method of interpolation based on normally distributed variables in a physical space. This model is well-suited for the inference of unknown data that depends on neighboring known data points. More precisely, Kriging weights the prediction of the unknown variable according to its distance from each of the known measurements. To achieve this, the model leverages the variance-covariance matrices of the Gaussian variables as well as the physical distances between them.

3.5.2 Principle

Let us denote, for all $i \in (1, \ldots, n)$, x_i the geographical coordinates of the microsensor i and $Z(x_i)$ the random variable representing its PM2.5 measure. Then the interpolation $\hat{Z}(x_0)$ is computed similarly as classic linear regressor: $\hat{Z}(x_0) = \sum_{i=1}^{n} w_i \times Z(x_i)$

But the essential feature of Kriging resides in the computation of the weights w_i, which depend on the distances between the target point and the known data points:

$$\begin{bmatrix} \hat{W} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathrm{Var}_{x_i} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathrm{Cov}_{x_i x_0} \\ 1 \end{bmatrix}$$

μ is a Lagrange multiplier used to honor the unbiasedness condition of the estimator. Var_{x_i} and $\mathrm{Cov}_{x_i x_0}$ are respectively the variogram and covariogram of the stochastic field formed by the microsensors.

3.5.3 Method

The procedure utilizing the Kriging model aims to compute a confidence interval for a microsensor's measurement of PM2.5 at time t, denoted $Z(x_0)$, based on a chosen number n of neighboring measurements $(Z(x_i))_{i\in\{1,\dots,n\}}$ taken at the same time t. If the observed measured value of PM2.5, $Z(x_0)$, falls within the prediction confidence interval, then the microsensor value is deemed valid.

3.5.4 Result

In our case, the Kriging method relies on interpolating spatially ground-truth points. The ground truth used was the three fixed stations, although this is not enough to properly fit a model with such regional variation. Another problem is that "Rocade Sud" and "Les Frenes" are close to each other, which makes generalization harder. To overcome those problems, the microsensor "ET00857" at Saint Bruno was also used as ground-truth for the model.

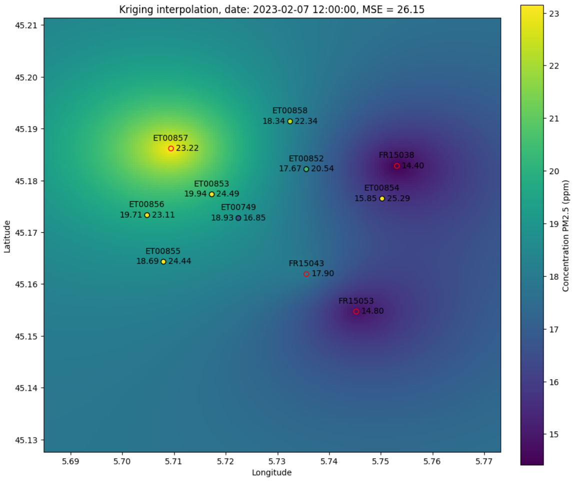


Figure 17: Kriging interpolation results. The value on the left of the point is the measured value, right side corresponds to the predicted value by the Kriging model. Red dots shows the ground-truth stations.

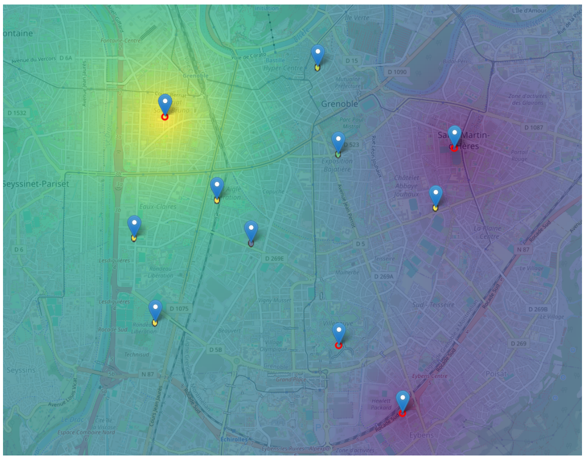


Figure 18: Map of Grenoble overlapped with the Kriging results. Red dots shows the ground-truth stations.

Even if the model could not predict properly the values, the model seemed promising, especially if more points could be considered as true ground truth or if few microsensors could be calibrated and reliable. Using this model we could also check if any point could be considered invalid, given the model variance and confidence interval (2 standard deviations).

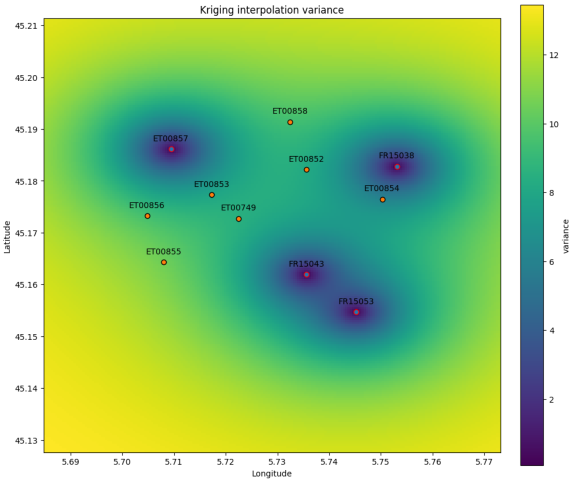


Figure 19: Kriging interpolation variance spread around the plane, the further from the point the bigger the variance.

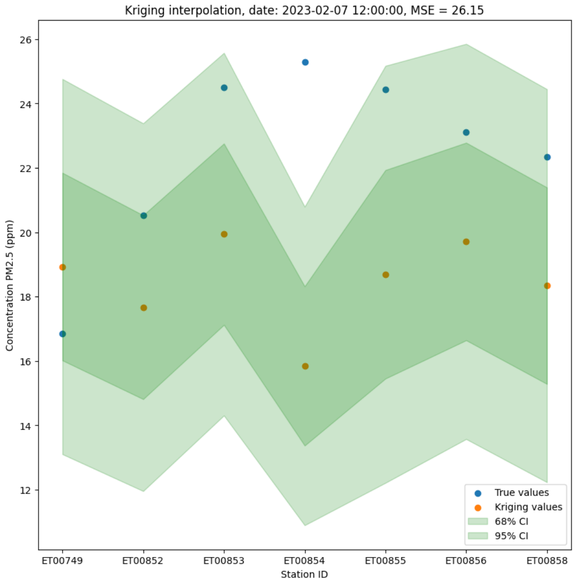


Figure 20: Map of Grenoble overlapped with the Kriging results. Red dots represent the ground-truth stations.

Using this model, we can see that the measure obtained by ”ET00854” would be considered as invalid, given it’s close to the fixed station and measures a value 78% bigger, even though it’s not close to a major road or typical pollutant of PM2.5.

4 Acquired Competences and Contributions by Team Members

Jules

Facing the Atmo Data Challenge was both a test of skill and collaboration. My main focus was refining my statistical learning knowledge, particularly model interpretability. This was challenging because we had to balance efficiency with understanding our models. Engineering the dataset with Henrique was also interesting, as it required us to anticipate our needs in terms of data, modeling, and other aspects. I worked on decision trees and decision forests due to their high explainability. I had to carefully optimize my hyperparameters, keeping in mind that the number of nodes should be limited to ensure the decision tree is easy to interpret. Finally, teamwork was really enriching and required frequent synchronization and reporting. I collaborated with Henrique on the Kriging model and data engineering. The workload was also made faster, for instance, thanks to Amrin writing the formatted LaTeX report while I contributed raw text or Aynur plotting graphs and making the results look fancier.

Henrique

The ATMO Data Challenge was an interesting experience, working with a diverse group from different backgrounds, which provided good discussions. Initially, I focused on data acquisition and data handling along with Jules, creating a generic data loader, which allowed our group to acquire and manipulate data easily. Along with the group, we explored the data statistically, looking for trends and correlations. I was responsible for the ARIMA model used in time-series forecasting, which is simpler than Machine Learning models. A second time, I also looked independently for a Temporal Fusion Transformer (TFT), but time was a constraint. Lastly, I was responsible for implementing the Kriging interpolation model based on the stations' locations. All this work would not be possible without the help and organization of my colleagues, who made amazing work. In this process, it was possible to reaffirm the importance of communication, collaboration, and management.

Amrin

From last year's Data Challenge, I knew it would be a hard challenge to face. This year, one of the main difficulties I faced was dealing with the data acquisition from the server. Thankfully, Henrique and Jule handled managing the data, which made it easier for us to move forward. I started with a short literature review, identified potential models, and focused on descriptive analysis, such as creating seasonal plots and analyzing meteorological impacts using code initially adapted by Jule. Other than these, I also attempted to implement the isolation forest model, but we found it difficult to interpret and dropped the idea. I also managed the report and LaTeX files, and throughout this experience, I learned the importance of communication, teamwork, and time management in a collaborative, time-constrained environment and how to work under pressure.

Aynur

As someone without prior practical experience in data science, the ATMO challenge provided an invaluable opportunity to gain practical skills in the field. Through this challenge, valuable expertise was acquired in extracting data from APIs, performing descriptive analysis, and training models to address the given task. My contributions to the project included conducting a detailed descriptive statistical analysis of the data, which involved visualizing key trends through graphs and identifying relationships between data variations and seasonal patterns. Additionally, an autoencoder model was trained to detect anomalies in the microsensor data, enabling the identification of inconsistencies in the data used for validation against the reference measurements. This achievement would not have been possible without the exceptional collaboration and effort of the team. Special thanks go to my teammates for their dedication in preparing a clean and well-structured dataset, which significantly streamlined the process and made tackling the task more efficient and rewarding.

5 Conclusion

To sum up, this 2024 ATMO data challenge began with acquiring and preprocessing the data, followed by analyzing data variability and using different validation methods for two scenarios: co-located microsensors with regulatory analyzers and non-co-located microsensors. Methods such as ARIMA for time series forecasting, autoencoders for anomaly detection, decision trees for classification, and Kriging for interpolation were applied. Kriging proved to be our best model because of its quality of being interpretable while delivering more consistent results than ARIMA, or decision tree. This approach could be extended to create a meta-model by combining the outputs of ARIMA and Kriging through techniques like averaging or linear regression to achieve better results.

A Link of GitHub Repository

Here, you will find the link to our github repository, where we have uploaded all of our code for the three stages. Here is the link:

DataChallenge: Air Quality Monitoring

References

Suivre et améliorer la qualité de l'air. (n.d.). https://www.grenoble.fr/138-suivre-et-ameliorer-la-qualite-de-l-air.htm

ATMO France. (2024). *L'indice atmo: Comprendre et mesurer la qualité de l'air* [Accessed: 2024-12-02]. https://www.atmo-nouvelleaquitaine.org/article/lindice-atmo-un-outil-precis-et-complet-de-la-qualite-de-lair

Box, G. E., & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. Holden-Day.

Hinton, G. E., & Salakhutdinov, R. R. (2006). *Reducing the dimensionality of data with neural networks* (Vol. 313). American Association for the Advancement of Science. https://doi.org/10.1126/science.1127647

Weng, L. (2018). *From autoencoder to variational autoencoder: A tutorial*. https://lilianweng.github.io/posts/2018-08-12-vae/

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *Smote: Synthetic minority over-sampling technique* [Journal of Artificial Intelligence Research, AI Access Foundation, Volume 16, Pages 321–357. DOI: 10.1613/jair.953. Accessed: 2024-12-02]. http://dx.doi.org/10.1613/jair.953