REFERENCE: amt-2024-127- "CC1"

Title: "*Improving Raw Readings from Low-Cost Ozone Sensors Using Artificial Intelligence for Air Quality Monitoring"*

Authors: *Guillem Montalban-Faet, Eric Meneses-Albala, Santiago Felici-Castell, Juan J. Perez-Solano and Jaume Segura-Garcia*

Departament de Informàtica, ETSE, Universitat de València, Avd. de la Universidad S/N, 46100 Burjassot, (Valencia), Spain

Dear editor and reviewer,

Thank you for giving us the opportunity to address the comments provided by the anonymous reviewers. We have made every effort to respond thoroughly to their feedback. Attached is a response letter with our responses highlighted **in blue**. The revised manuscript also uses **blue text** to indicate the changes made. In some answers, this **blue text** is highlighted if there is more than one answer.

We would also like to express our gratitude to the anonymous reviewers for their valuable comments and suggestions. We appreciate the time and effort they have invested in improving our work. We firmly believe that this manuscript is now suitable for publication and an excellent contribution to share with the broader research community.

Reviewer's comments (A. Kourtiche Referee #2 CC1, 24 Jan 2025)

The article focuses on leveraging low-cost sensors (LCS), specifically the ZPHS01B module, to monitor ground-level ozone (O_3), a critical air pollutant and urban pollution indicator. Due to the limited accuracy of LCS, the authors applied advanced Machine Learning (ML) methods, including Gradient (GB), (RF), (ADA), and (DT), for calibration.

The dataset spans 165 days, with optimal results obtained using a 10 min monitoring interval. The GB model achieved the best performance, reducing the estimation error by 94.05%, while other models reduced errors by more than 89%. (HPO) and feature selection techniques (FIA, FS) improved model performance.

First of all, we would like to sincerely thank you for your thoughtful review and comments, which have greatly contributed to improving our work.

In the following sections, we will address all your comments, queries, and suggestions. This is an extended version of an answer provided to, we guess the same reviewer, as RC2 from Feb 4 2025 about the same manuscript.

The authors plan to extend the dataset and include additional parameters like wind speed and traffic density in future work.

I.Questions:

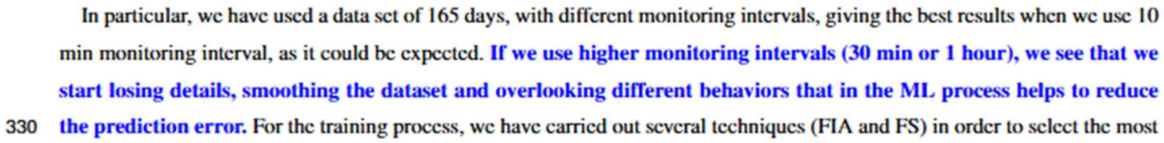
1. Why was a 10 min interval more effective than 30 min or 1 hour ? Could other time intervals (e.g., 5 or 15 minutes) be explored?

Response 1: Thank you for this comment. 10 minutes was the minimum interval given by the references, since it is a standardized monitoring interval for outdoor official air quality monitoring in normal conditions as depicted in Section 3.2.

And among the datasets given by the 10 min, 30 min and 1-hour intervals, after training the models as it is explained in Section 4, we see that the minimum prediction error is achieved by the 10 minutes interval, as it is shown in Figure 7 for RMSE and MAE in the new version of the manuscript (Figure 8 in the first version).

Notice that this is due to several reasons. On one hand, the 10 min interval is determined by environmental researchers as the interval that better gathers the different outdoor air quality behaviors, with higher detail under normal conditions. Higher sampling frequencies (lower monitoring intervals) create oversampling and redundancy. On the other hand, if we use higher monitoring intervals (30 min or 1 hour), we see that we start losing details, smoothing the dataset and overlooking different behaviors that in the Machine Learning (ML) process helps to reduce the prediction error.

We have clarified this issue in the new version of the manuscript as follows:

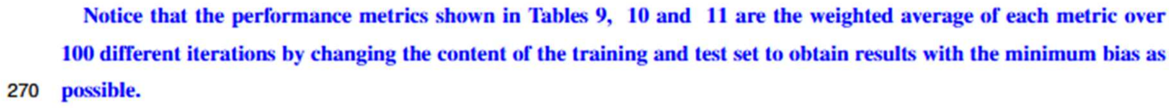


In particular, we have used a data set of 165 days, with different monitoring intervals, giving the best results when we use 10 min monitoring interval, as it could be expected. **If we use higher monitoring intervals (30 min or 1 hour), we see that we start losing details, smoothing the dataset and overlooking different behaviors that in the ML process helps to reduce the prediction error.** For the training process, we have carried out several techniques (FIA and FS) in order to select the most

2. Were the ensemble models compared statistically to determine significance in performance differences?

How do these models generalize to new datasets or different geographical locations?

Response 2: Yes, these models were tested and compared in Section 4. We evaluated the performance metrics in terms of R^2, RMSE, MAE and MAPE as shown in Tables 9-11. These results are the weighted average of each metric over 100 different iterations by changing the content of the training and test set to obtain results with the minimum bias as possible as explained in the manuscript. This explanation is included in the new version as follows:



Notice that the performance metrics shown in Tables 9, 10 and 11 are the weighted average of each metric over 100 different iterations by changing the content of the training and test set to obtain results with the minimum bias as possible.

With regard to the generalization of different datasets, this is considered by taking a sufficient dataset, as it is detailed in the reference "Machine Learning in Environmental Research: Common Pitfalls and Best Practices" by Zhu, et al. In particular, as it is explained in Section 3.2, the recommended relationship (ratio) between Sample size and Feature size (Sample-size to Feature-size Ratio (SFR)) is higher than 500. In our datasets, we have a sample size of 23496, 7843 and 3922 for 10 min, 30 min and 1 hour interval, that is a SFR of 4699.2, 1568.6 and 784.4, since we only use 4 features, as it is depicted in Section 4.

About extending the dataset with more data, notice that the fusion of the different datasets from different locations as a first approach is not recommended, since they could change the environmental conditions. This merging process would require refinement in the datasets as well as in the models, that in this case, given the available datasets are not necessary. It is better to work with different datasets from different locations separately, independently.

Nevertheless, in order to answer the reviewer, we have created another dataset (Dataset 2) with new samples from another deployment with two different LCS nodes (called AQ IoT Node 1 and 2) in a different location, in Valencia city. In particular, the new dataset is from the official AQ monitoring station called Moli del Sol (Valencia, Spain) placed at 39.48113875, -0.40855865, managed by Generalitat Valenciana (GVA) and its data is retrieved from https://rvvcca.gva.es/estatico/46250048, for O3 calibration. This station is 4.1 km away from the previous official station used for the dataset in the manuscript. In this case, this dataset is from May 31, 2024, till January 25, 2025**,** it has 239 days and includes data from different seasons as suggested by the reviewer. Notice that in our case, to carry out all these deployments, it is required to ask for permission to the official institutions in charge of Air Quality.

Thus, with this new dataset (Dataset2), we have repeated the same process as explained in the manuscript, achieving nearly the same results as shown below. We show the HPO results over 100 different iterations by changing the content of the training and testing set (with the best results given by 90%/10% ratio as already discussed in Section 4) to obtain results with the minimum bias as possible, for both nodes (AQ IoT Node 1 and 2):

NODE 1

```
GradientBoostingRegressor(criterion='squared_error', max_depth=None,
learning_rate=0.1,max_features=1.0, n_estimators=900, subsample=1.0)
R2 = 0.9405841973910234
RMSE = 6.107097433579371
MAE = 4.336455961006405
MAPE = 0.1679585719053396
time = 102.18236994743347

RandomForestRegressor(max_depth=None,max_features=1.0, n_estimators=100)
R2 = 0.9046692614127114
RMSE = 7.735712909738179
MAE = 5.23282966066717
MAPE = 0.20992345469839893
time = 27.86010217666626

AdaBoostRegressor(estimator=DecisionTreeRegressor(max_features=1.0),
 n_estimators=50, learning_rate=0.01, loss='exponential')
R2 = 0.9090424941272316
RMSE = 7.556194639834324
MAE = 4.564039465946062
MAPE = 0.16874010491994965
time = 11.807359457015991

DecisionTreeRegressor(max_depth=None, max_features=1.0, splitter='best')
R2 = 0.8191113924173187
RMSE = 10.655883718994565
MAE = 6.295906305813436
MAPE = 0.2235395149139127
time = 0.33399152755737305
```

NODE 2

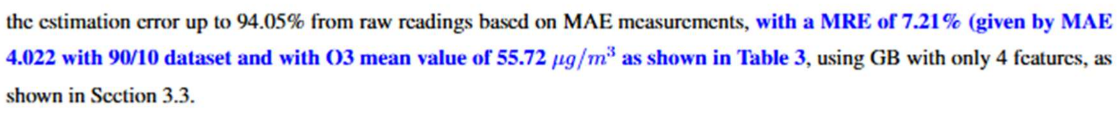
```
GradientBoostingRegressor(criterion='squared_error', max_depth=None,
learning_rate=0.1,max_features=1.0, n_estimators=900, subsample=1.0)
R2 = 0.9547003457380135
RMSE = 5.332505456267162
```

```
MAE = 3.7416539078656776
MAPE = 0.14152286529664848
time = 82.03594541549683

RandomForestRegressor(max_depth=None,max_features=1.0, n_estimators=100)
R2 = 0.934358633720318
RMSE = 6.419078264005403
MAE = 4.187047365360581
MAPE = 0.15794878544527777
time = 20.572300910949707

AdaBoostRegressor(estimator=DecisionTreeRegressor(max_features=1.0),
 n_estimators=50, learning_rate=0.01, loss='exponential')
R2 = 0.9287003904552755
RMSE = 6.690020376986309
MAE = 3.8299511364469465
MAPE = 0.13586980540686971
time = 8.766397953033447

DecisionTreeRegressor(max_depth=None, max_features=1.0, splitter='best')
R2 = 0.8745869394552789
RMSE = 8.872688771740032
MAE = 4.974713868475632
MAPE = 0.16654468197115013
time = 0.23625636100769043
```

As seen in this new dataset, both AQ IoT nodes exhibit similar behavior. However, Node 2 performs slightly better than Node 1, likely due to manufacturing variations associated with their low cost. It is important to emphasize that these results closely resemble those already presented in the manuscript. In the following table we compare and summarize these results from *Dataset 1*, the one used in the manuscript, and *Dataset 2*, the new data set analyzed here in the review.

GB optimized	**Dataset1**	**Dataset2 (Node1)**	**Dataset2 (Node2)**
R^2	0.938	0.940	0.954
RMSE	6.492	6.107	5.332
MAE	4.022	4.336	3.741
MAPE	0.194	0.167	0.141
Time [s]	66.937	102.182	82.035

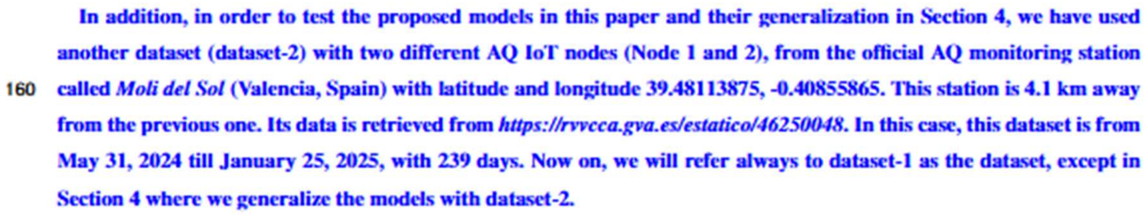
As we can see, Node 1 works worse than Node 2, and the previous results obtained from Dataset1 are between these two. In this case, with Dataset 2, the Mean Relative Error (MRE) is 6,71% for Node 2 and for Node 1 is 7.78%, and with Dataset 1 was 7.21%. The estimation of the MRE discussion is at the end of Section 4 in the new version of the manuscript as follows:



the estimation error up to 94.05% from raw readings based on MAE measurements, **with a MRE of 7.21% (given by MAE 4.022 with 90/10 dataset and with O3 mean value of 55.72 $\mu g/m^3$ as shown in Table 3**, using GB with only 4 features, as shown in Section 3.3.

Thus, based on this information, we conclude that for the ZPHS01B module, 165 days of Dataset 1 provide sufficient information to generalize the proposed calibration models. This aligns with the SFR recommended values, as stated earlier. In other words, given the features and characteristics of this module, the original dataset (165 days) contains enough information to generalize the behavior of the O3 sensors and their response. Thus, better results cannot be achieved with other datasets given the constraints of this module.

This information has been included in the new version of the manuscript with these modifications, in Section 3.1, describing the dataset-2 as follows:



In addition, in order to test the proposed models in this paper and their generalization in Section 4, we have used another dataset (dataset-2) with two different AQ IoT nodes (Node 1 and 2), from the official AQ monitoring station called *Moli del Sol* (Valencia, Spain) with latitude and longitude 39.48113875, -0.40855865. This station is 4.1 km away from the previous one. Its data is retrieved from *https://rvvcca.gva.es/estatico/46250048*. In this case, this dataset is from May 31, 2024 till January 25, 2025, with 239 days. Now on, we will refer always to dataset-1 as the dataset, except in Section 4 where we generalize the models with dataset-2.

In Section 4, in the results with:

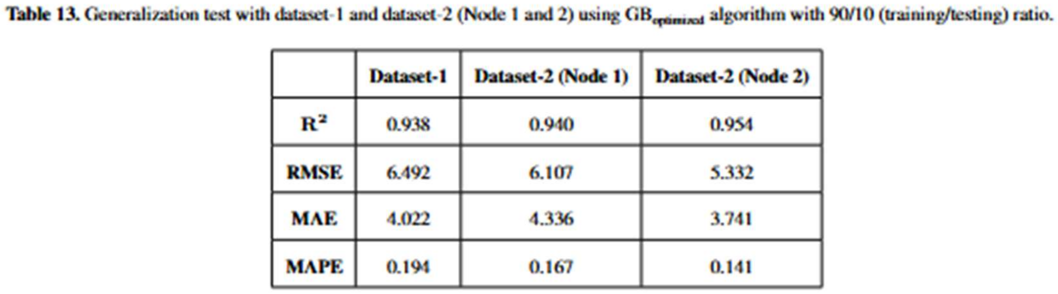
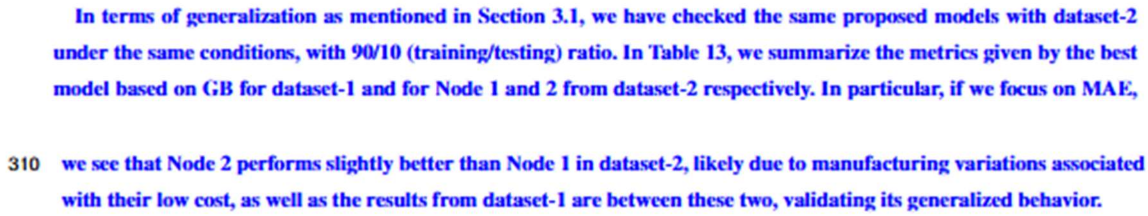


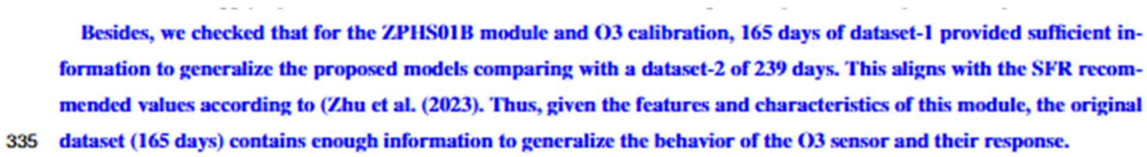
Table 13. Generalization test with dataset-1 and dataset-2 (Node 1 and 2) using $GB_{optimized}$ algorithm with 90/10 (training/testing) ratio.

	Dataset-1	**Dataset-2 (Node 1)**	**Dataset-2 (Node 2)**
R^2	0.938	0.940	0.954
RMSE	6.492	6.107	5.332
MAE	4.022	4.336	3.741
MAPE	0.194	0.167	0.141



In terms of generalization as mentioned in Section 3.1, we have checked the same proposed models with dataset-2 under the same conditions, with 90/10 (training/testing) ratio. In Table 13, we summarize the metrics given by the best model based on GB for dataset-1 and for Node 1 and 2 from dataset-2 respectively. In particular, if we focus on MAE, we see that Node 2 performs slightly better than Node 1 in dataset-2, likely due to manufacturing variations associated with their low cost, as well as the results from dataset-1 are between these two, validating its generalized behavior.

As well as in the conclusion section:



Besides, we checked that for the ZPHS01B module and O3 calibration, 165 days of dataset-1 provided sufficient information to generalize the proposed models comparing with a dataset-2 of 239 days. This aligns with the SFR recommended values according to (Zhu et al. (2023). Thus, given the features and characteristics of this module, the original dataset (165 days) contains enough information to generalize the behavior of the O3 sensor and their response.

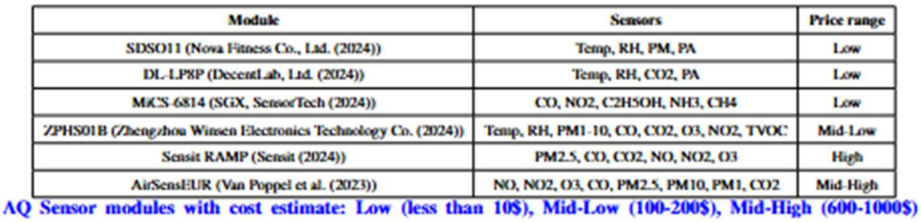
3. Why were only 4 features used for the GB model? Were additional environmental factors considered initially but excluded?

Response 3: Thank you for your comment. Based on the analysis conducted in Section 3.3 for feature selection, we proceeded in Section 4 with the features that provided the best performance metrics. These selected features are [date, O3, T, RH], as initially indicated in Section 4. We omitted other features that led to poorer results. It is important to note that adding less significant features can reduce the importance of key parameters, ultimately affecting the overall performance. For instance, including both T and PM results in worse performance compared to using only T, leading to less effective models.

4. How does the proposed approach balance cost savings with performance?

Response 4: Low-cost sensors, as detailed in Section 2, are much cheaper than official equipment but with lower accuracy.

Taking this information into account, Table 15 of the manuscript (in the new version and 17 in the first one), compares our approach to O3 calibration with similar related work. It is important to note that the starting point of the selected studies for comparison differs slightly from ours, as these studies used more reliable and significantly more expensive low-cost sensors—approximately ten times the cost of the ZPHS01B module as depicted in Table 1 of Section 2 with its price range. This was already included in the manuscript.



Module	Sensors	Price range
SDSO11 (Nova Fitness Co., Ltd. (2024))	Temp, RH, PM, PA	Low
DL-LP8P (DecentLab, Ltd. (2024))	Temp, RH, CO2, PA	Low
MiCS-6814 (SGX, SensorTech (2024))	CO, NO2, C2H5OH, NH3, CH4	Low
ZPHS01B (Zhengzhou Winsen Electronics Technology Co. (2024))	Temp, RH, PM1-10, CO, CO2, O3, NO2, TVOC	Mid-Low
Sensit RAMP (Sensit (2024))	PM2.5, CO, CO2, NO, NO2, O3	High
AirSensEUR (Van Poppel et al. (2023))	NO, NO2, O3, CO, PM2.5, PM10, PM1, CO2	Mid-High

Table 1. **AQ Sensor modules with cost estimate: Low (less than 10$), Mid-Low (100-200$), Mid-High (600-1000$) and High (≈<4000$).**

Thus, we consider that this is a fair balance, highlighting the improvements for the O3 calibration by using this module.

II.Improvements Needed:

The current dataset covers 165 days. Increasing the dataset size and covering different seasons or regions could improve generalizability.

Response 5: Thank you for this comment.

As already answered in Response 2, we repeated the analysis with a new dataset (Dataset 2) from May 31 2024 till January 25, 2025. This dataset has 239 days and includes data from different seasons as suggested by the reviewer.

As a first approach, we agree that increasing the dataset size and covering different seasons or regions could improve the generalizability of ensemble machine learning algorithms. In general, a larger dataset typically provides more diverse examples, allowing the model to learn from a wider variety of patterns and reducing the risk of overfitting to specific data characteristics. Even when the data spans different locations, the model also may become more robust by capturing the seasonal variations and regional trends that might not be present in a smaller, localized dataset.

However, as stated in Response 2, 165 days of Dataset 1 provided sufficient information to generalize the proposed O3 calibration model and better results cannot be obtained with other datasets given the constraints of this module.

Besides, it must be stressed that there is a trade-off between accuracy and life-time of the low cost sensor. And this is the main reason we cannot last the different deployments for years. In particular, these low cost sensor modules degrade fast and their accuracy is reduced in months.

Adding complementary parameters, such as traffic patterns, industrial activities, and meteorological conditions, could enhance the model’s robustness.

Response 6: Thank you for your interesting comment. Although this approach is very interesting and valid for some scenarios, in our case we focus only on Air quality information obtained directly from the low-cost sensor modules. Of course we could include other related information in more theoretical studies, but not in a real scenario as the one proposed. This type of information (traffic patterns, industrial activities) is not available easily in real time, assuming the low cost IoT AQ nodes, described in this paper. As it is explained in Section 2, usually, these nodes have limited communications and only can gather local information from their directly connected sensors. And when the information is processed, they can run the ML models to improve the accuracy of the readings. Finally, they can upload this information to other external servers, but always with constraints due to their features.

Besides, other meteorological sensors (such as wind speed and direction) could be interesting, but in the end they will modify the different diffusion models of the different gasses, but in practice they do not alter the direct readings of the low cost Air quality sensors, if they are properly housed as we did in deployment.

Nevertheless, this discussion has been included in Section 5 in the conclusion as future work, but more focus on theoretical studies rather than on real deployments with constrained devices as the ones used for Air quality monitoring with low-cost features.

-While GB is identified as the best-performing model, a statistical comparison of model performances (e.g., paired t-tests on errors) should be included to support conclusions.

+Explain why ADA and RF performed similarly or differently from GB.

-Discuss the trade-off between GB’s higher execution time and its improved accuracy.

-Propose optimizations for deployment scenarios requiring real-time predictions.

Response 7: Thank you for your comments. Next, we provide an extended explanation about these issues. The different key points about this explanation have been used in order to improve the wording in different parts of the manuscript.

Find next a detailed discussion about all these items.

As we mentioned below, the guidelines to process this kind of data is shown in reference "Machine Learning in Environmental Research: Common Pitfalls and Best Practices" by Zhu, et al.. Thus, in particular, about the mentioned "paired t-tests on errors", these tests are used to test if the means of two paired measurements are significantly different, but this does not apply in our experiments, since the different models are carried out independently and using different data-sets, as it is explained in Section 4 and different "training-test" ratio percentages from these datasets: 60%-40%, 70%-30%, 80%-20% and 90%-10%. Besides, during the training process, each performance metric depicted in Section 4 based on R^2, RMSE, MAE and MAPE is obtained with 100 different iterations by changing the content of the training and test set to obtain results with the minimum bias as possible.

About the behavior of ADA and RF vs GB, although all of them are ensemble ML algorithms, their algorithms are based on slightly different approaches. In particular, as explained in Section 3.4, Adaptive Boosting (AdaBoost) and Gradient Boosting differ in how they improve performance. AdaBoost focuses on re-weighting the training data, assigning higher weights to misclassified examples, so subsequent weak learners focus on these harder cases. It combines weak learners using weighted voting, emphasizing the most accurate ones. In contrast, Gradient Boosting focuses on minimizing a specific loss function by fitting each new weak learner to the residual errors (differences between actual and predicted values) of the previous model. This makes Gradient Boosting more flexible, allowing it to handle custom loss functions and more complex learners. While AdaBoost is simpler and faster, but sensitive to noise, Gradient Boosting is more powerful and robust for complex tasks, but it requires higher execution time.

Similarly, Random Forest and Gradient Boosting are both ensemble learning algorithms and use decision trees as base models, but differ significantly in their approach. Random Forest builds multiple decision trees independently by randomly sampling data and features, then aggregates their predictions (via majority vote for classification or averaging for regression). This parallel training makes it robust, fast, and less prone to overfitting. In contrast, Gradient Boosting trains decision trees sequentially, where each tree attempts to correct the residual errors of the previous ones by optimizing a specified loss function. This iterative process makes Gradient Boosting more flexible and capable of fine-tuning but slower. While Random Forest excels in robustness and simplicity, Gradient Boosting often achieves higher accuracy in complex tasks due to its ability to learn from mistakes adaptively.

Besides, these algorithms (ADA, RF and GB) have different hyperparameters and with different optimized values, adjusted independently by HPO techniques, as shown in Table 5-8 in Section 3.4.

Regarding the execution time, the trade-off between Gradient Boosting's higher execution time and its improved accuracy compared to Adaptive Boosting and Random Forest comes down to the balance between computational cost and predictive performance. Gradient Boosting builds trees sequentially, optimizing a specific loss function at each step, which allows it to capture complex patterns and often achieve superior accuracy. However, this iterative process makes it computationally intensive and slower, especially for large datasets or when fine-tuning hyperparameters. Adaptive Boosting, while also sequential, is generally faster because it uses simpler learners (like decision stumps) and focuses on re-weighting misclassified points rather than optimizing a loss function as mentioned before. Random Forest, in contrast, trains trees independently and in parallel, making it much faster, but it sacrifices some accuracy because it relies on averaging predictions instead of iterative error correction. While Gradient Boosting excels in tasks where accuracy is paramount, its higher execution time may not be justified for less complex problems or time-sensitive applications, where Random Forest or Adaptive Boosting could provide a faster, more practical solution.

And finally, about the optimizations to be applied on the deployments for real-time predictions, it must be stressed that once these models are trained, they can be ported to the low cost AQ node that is based on a microcontroller. Then, with these models we can improve the accuracy of the direct readings immediately.

Notice that these details have been used to enrich the new wording in Section 4 when dealing with the different algorithms. Besides, it has been included in the future work, since in practice, this is a very interesting point for the whole AQ monitoring network.

III-Proposed Best Method

-Explore DL models like LSTMs or Temporal Convolutional Networks (TCNs) for time-series prediction to capture long-term dependencies.

Response 8: Thank you for this interesting comment.

Gradient Boosting algorithms are often more practical, efficient, and interpretable for time-series prediction tasks, especially when datasets are small-to-medium-sized, contain noise, or require explicit domain knowledge. While DL models like LSTMs and TCNs excel in capturing long-term dependencies in very large datasets, Gradient Boosting flexibility, lower data requirements and ease of use make it a strong choice for real-world time-series applications.

Nevertheless, as it is mentioned before, the lifetime of these low cost sensors and their performance degrade over the time (aging), due to their manufacturing process. In particular, this is more critical in the ZPHS01B module and that is the reason we focused on these ensemble algorithms.

Of course, there is a tradeoff between ML and DL in these scenarios, but pros and cons made us conduct the test with these ML techniques, with good results.

It is worth mentioning that we also used DL techniques, but we observed that they are not able to generalize as the ML approach did. And for this reason, the results using DL techniques are not so robust and reliable, mainly due to overfitting even with bigger datasets in this context and scenario. These results are shown below for a simple Sequential Neural Network from TensorFlow/Keras using an optimizator *stochastic gradient descent* with an input of 4 features and two layers. These two layers are a dense layer with four neurons and a linear activation, followed by a second layer with a neuron that provides the output. The network scheme is shown in **Figure 1**, below.

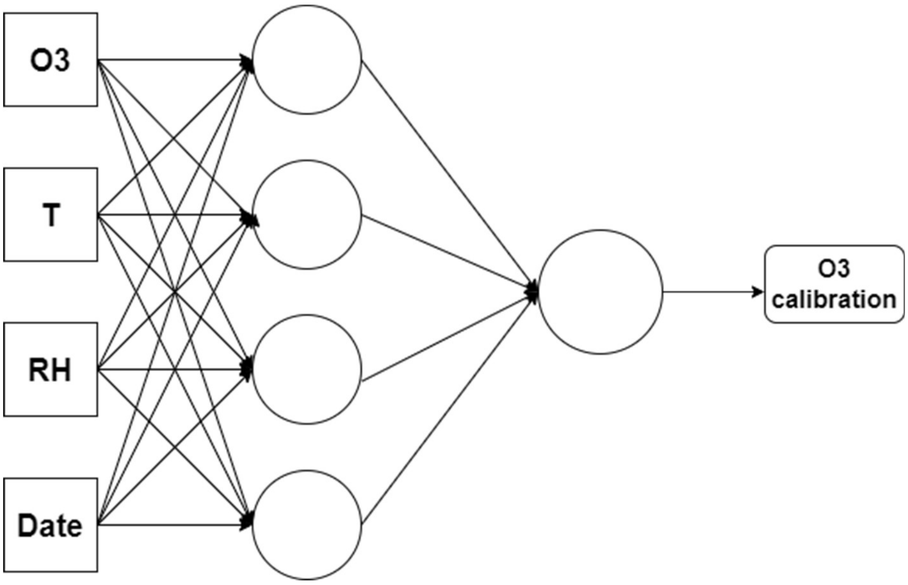


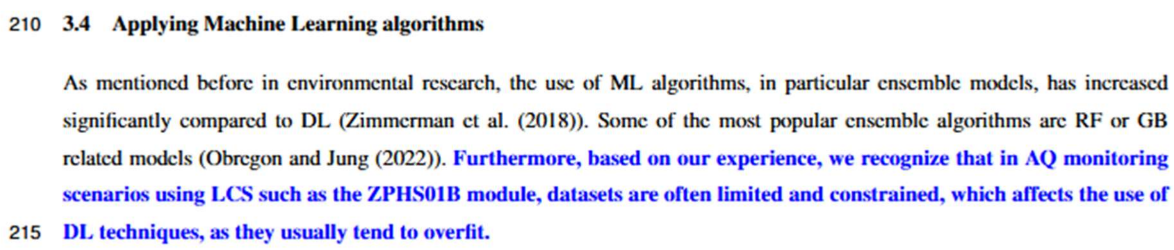
Figure 1: *Scheme of the Sequential Neural Network from TensorFlow/Keras using an optimizator stochastic gradient descent with an input of 4 features and two dense layers.*

The results from this *Sequential Neural Network* are:

```
R2 Score: 0.9999999999976741

RMSE: 3.514481801502949e-05

MAE: 2.9925663790820442e-05
```

As we can see, these results show that these techniques learn and memorize the whole dataset and we cannot generalize. That is the reason we focused on ML since they adapt and perform better in this scenario, given by the AQ monitoring stations and the ZPHS01B low cost sensor module for O3 calibration.

We have included this explanation also in the revised version, in order to justify the selection of these ML techniques instead of other techniques. This information is included as follows:



3.4 Applying Machine Learning algorithms

As mentioned before in environmental research, the use of ML algorithms, in particular ensemble models, has increased significantly compared to DL (Zimmerman et al. (2018)). Some of the most popular ensemble algorithms are RF or GB related models (Obregon and Jung (2022)). **Furthermore, based on our experience, we recognize that in AQ monitoring scenarios using LCS such as the ZPHS01B module, datasets are often limited and constrained, which affects the use of DL techniques, as they usually tend to overfit.**

-Combine GB with DL methods for feature extraction and refinement, especially if additional parameters are included.

Response 9: Thank you for this comment.

From our experience, combining Gradient Boosting algorithms with Deep Learning methods for feature selection is often unnecessary due to several reasons. Gradient Boosting algorithms as the ones proposed in this research, are inherently capable of handling feature selection through their built-in mechanisms, such as calculating feature importance and automatically ignoring irrelevant or redundant features during training as it was shown in Section 3.4. These algorithms excel in structured data tasks and effectively model complex, non-linear relationships without requiring additional feature selection methods as depicted in Section 3.4. Furthermore, Deep Learning-based feature selection is computationally expensive, requiring significant resources and larger datasets to avoid overfitting, which may not justify the effort when Gradient Boosting can already achieve competitive results. Additionally, Gradient Boosting provides interpretable outputs which offer clear insights into feature importance, unlike Deep Learning methods, which often function as black boxes. Finally, introducing Deep Learning adds unnecessary complexity to the pipeline, increasing training time and resource demands without guaranteed improvements in predictive performance, especially when Gradient Boosting already performs well on the given dataset.

This explanation and justification have been considered in the new version of the manuscript in Section 3.4.

-Use advanced ensemble techniques like Stacked Generalization (Stacking) to blend predictions from GB, RF, and ADA for better accuracy.

Response 10: Thank you for this comment.

It must be pointed out that using Stacked Generalization (Stacking) to blend predictions from Gradient Boosting, Random Forest and AdaBoost may not be ideal due to several reasons. First, it adds complexity by introducing a meta-learner, making the workflow harder to interpret and manage, often for marginal accuracy gains. Gradient Boosting already iteratively optimizes predictions and often outperforms combinations with simpler models like Random Forest or AdaBoost, making the stack redundant. Additionally, stacking increases the risk of overfitting, especially with small datasets, as the meta-learner can overfit to the base models' predictions. It also significantly increases training time and computational demands, while the lack of diversity among tree-based models reduces the potential benefits of combining them. Simpler alternatives, such as weighted averaging or selecting the best-performing model, often achieve comparable results without the added complexity.

Several scientific references support the arguments against using Stacked Generalization (Stacking) to combine predictions from Gradient Boosting, Random Forest, and AdaBoost. The increased complexity and risk of overfitting associated with stacking are highlighted in "A guide to ensemble learning," which notes that ensemble methods can lead to computational complexity and overfitting risks [1]. Additionally, the article "Stacking to Improve Model Performance: A Comprehensive Guide" discusses how utilizing too many base models in a stacked ensemble can result in overfitting and increased computing complexity [2]. Furthermore, the article "Gradient Boosting vs Random Forest" explains that Gradient Boosting focuses on sequential correction of errors, while Random Forest relies on the diversity of independently trained trees, suggesting that combining these models may not provide significant additional benefits [3].

Thus, based on these reasons, it shows that for this case, stacking these particular models may introduce unnecessary complexity and overfitting risks without substantial improvements in predictive performance.

However, this comment has been included in the new version of the manuscript, to justify this explanation.

References:

[1] https://serokell.io/blog/ensemble-learning-guide

[2] https://medium.com/@brijesh_soni/understanding-boosting-in-machine-learning-a-comprehensive-guide-bdeaa1167a6

[3] https://www.geeksforgeeks.org/gradient-boosting-vs-random-forest/

IV-Recommendations :

Expand the dataset and include more parameters to increase model accuracy.

Conduct real-world validation to demonstrate scalability and robustness.

Compare ML and DL approaches to assess their suitability for time-series AQ calibration.

Provide open-source tools for replicating and extending the proposed calibration process.

By addressing these improvements and exploring advanced methodologies, the study can significantly contribute to cost-effective and scalable air quality monitoring solutions.

Response 11: Thank you for feedback.

We consider that all these issues have been discussed during this review, and some of them, the more interesting, have been included in the new version of the manuscript improving its wording.

In summary, about the dataset, we have discussed this in Response 2 and 5 with detail, as well as using other locations. About the real-world validation, all our trials and measurements come from real deployments. We have not used anything simulated. About the comparison between ML and DL, as it was discussed previously, we have included this discussion as well as their worse results, in favor of ML in this case. Also, about the open-source tools, all our datasets are available online, as it is indicated in the last part of the manuscript with the following statement “Please feel free to contact to the authors for further information: http://www.uv.es/eco4rupa/dataset.html”.

Finally, thank you for your thoughtful review and comments which will enable us to improve this work. We appreciate the time and effort invested in your review.