A calibration methodology of low-cost air pollutant sensor using neural networks

Aymane Souani, Alexandre Hucher, Vincent Vigneron, Hichem Maaref

A calibration methodology of low-cost air pollutant sensor calibration using neural networks

[‡]A. Souani, A. Hucher
ECOMESURE
Saclay, France
{aymane.souani,alexandre.hucher}@ecomesure.com

V. Vigneron, H. Maaref
IBISC EA 4526
Univ Evry, Université Paris-Saclay
Evry, France
{vincent.vigneron,hichem.maaref}@univ-evry.fr

***Abstract*—Air quality Low cost sensors (LCSs) are cheap and can map extensive areas. They alert people about pollution spikes in smart city buildings (schools, universities, hospitals ...) or industrial areas. Before using them for a specified task, they must be calibrated to give accurate readings, *i.e.* they must be aligned with a measure based on a reference machine. Unfortunately, classic calibration is limited by interferences with other pollutants or can be affected by atmosphere constants in the case of uncontrolled environments. This paper proposes a calibration solution based on artificial neural networks (ANN).**

***Index Terms*—Calibration, LCS, air pollutant, regression, neural networks, SVR.**

I. Introduction

Pollutant LCSs are used everywhere, at any time; most are not resistant to critical situations, *e.g.* environment changes, instability outside lab conditions, temperature, humidity, ... [1]. Two sensors from the same manufacturer could have different readings and behaviors (slightly similar but still different). Hysteresis and time drift also affect sensor responses, which requires permanent calibration. It is not the case for reference instruments: their measurements are accurate and provide real-time data. However, their high cost remains a barrier to their use for daily tasks.

Calibration is here to rectify the LCSs response concerning the reference machine response. For this purpose, the *reference* instrument recording the (supposedly accurate) pollutant's concentration provides the actual concentration value. The response of the sensor to calibrate shall fit the reference value.

[‡]A CIFRE grant from ANRT supports A. Souani, number 2020/1127.

This work is not about pollutant concentration forecasting but a calibration task.

Current pollutant sensor calibration methods usually make two assumptions : (*i*) The output of the sensor is linearly related to the reference measurement. (*ii*) The environmental conditions are under control.

Sensors may interfere with other pollutants (*e.g.* O_3 with NO_2) or with temperature and (relative) humidity. The most common approach to consider interferences is linear regression (see Tab. I).

Mijling *et al.* [2] and Spinelle [3] propose to calibrate simultaneously NO_2 and O_3 sensors. Pollutant sensor responses are not necessarily linear, even if it is current practice. In addition, these models are corrupted with electronic noise.

Table I: Some multivariate linear calibration models in the case of interferences [3], [4]. R_S is the reference measurement, a, b, c, d, e and $\beta_{\ell i}, \ell = 0, \ldots, 3$ are parameters, T and H are temperature and humidity respectively, and O_3 is the ozone concentration.

Sensor name	Multivariate linear model
NO2B4	$NO_2 = \dfrac{R_S - bO_3 - cT - dH - e}{a}$
PMS5003	$PM_x = \dfrac{R_S - \beta_{2i}T - \beta_{3i}H - \beta_{0i}}{\beta_{1i}}$

Machine learning algorithms have been proposed to improve the performance of air quality monitoring sensors. Zimmerman *et al.* [5], and Malings [6], for instance, explored and compared multivariate linear regression (MLR), random forest (RF), and support vector regressor (SVR).

Calibration data selection, collinearity, non-normality, non-selectivity, and non-linearity are the

main calibration difficulties. For instance, coefficient estimates of collinear independent variables would be susceptible to the change in the model, even for a tiny change. Collinearity will inflate the variance and standard error of coefficient estimates. This, in turn, will reduce the reliability of our model (see Draper [7]). The variable selection can fix collinearity issues by *(a)* Excluding variables that have high variance inflation factor (VIF) value (see section III-B) from our regression model. *(b)* Reducing the input space dimensionality (with PCA for instance).

In the following, the efficiency of the neural network calibration strategy is demonstrated on an accurate dataset, introduced in section II. Section III presents the methodology, and section IV compares several models. A short conclusion concludes this work.

Notations: In this paper, small Latin letters $a, b, \ldots$ represent integers. Small bold letters $\mathbf{a}, \mathbf{b}$ are put for vectors, and capital letters A, B for matrices or tensors, depending on the context. The dot product between two vectors is denoted $< \mathbf{a}, \mathbf{b} >$. We denote by $\|\mathbf{a}\| = \sqrt{< \mathbf{a}, \mathbf{a} >}$, the ℓ_2 norm of a vector. $X_1, \ldots, X_n$ are variates, $x_1, \ldots, x_n$ observations. A hat $\widehat{\ }$ will denote empirical estimates of model parameters.

II. Data

A. Experimental setup

This paper will focus on calibrating the pollutant $PM_{2.5}$ (particle matters under 2.5μm). The indoor calibration needs completely controlled and very stable environmental conditions. We have prepared an experimental setup to capture high-quality data for $PM_{2.5}$, which may vary and interfere with the outside climate changes. Kumar [8] has proven that in-situ calibration is more interesting than performing only the lab tests. Our setup was conducted using 12 different ECOMESURE optical partical counters (OPCs) to capture $PM_{2.5}$ concentrations[1], alongside with the reference machine 1405 TEOM which is a precise continuous ambient particulate monitor that can only measure one pollutant (see Tab. II).

Each PM measurement is accompanied by temperature and relative humidity measures. 45-day data were collected from May 30th to July 14th, 2022. It totalizes 58,887 measures.

[1]The 12 sensors can also measure PM_1 and PM_{10}.

Table II: PM measurement devices used in the campaign.

Device	Pollutant	Freq. (Hz)
™ECOMESURE OPC	PM_x [a]	0.016
1405 TEOM	$PM_{2.5}$	0.00027

[a]Includes PM_1, $PM_{2.5}$, PM_{10}

The reference machine gives concentration levels based on a 1-hour average. Hence, our procedure also considers the averaged measurements of $PM_{2.5}$ on one hour.

Figure 1: The experimental setup containing the 12 sensors (framed by green) and reference instrument measurement (sampling $PM_{2.5}$ framed by red).

III. Methods

A. Calibration model description

Suppose we have a set of calibration data with a rapid measurement vector $\boldsymbol{x} = (x_1, \ldots, x_d)^T$ and a reference measurement $\boldsymbol{y}$ taken on each N sample. $x_i, i = 1, \ldots, d$ are independent variables.

Multivariate calibration can usually not be handled by multi-linear regression based on the ordinary least square (OLS) regression because of near-multicollinearity among the variables in the matrix X. When the variables are multicollinear, the matrix $X^T X$ is singular, and the OLS solution becomes not unique.

1) Multivariate linear regression: the target value $\boldsymbol{y}$ is expected to be a linear combination of the features $\boldsymbol{x}$, *i.e.* $\widehat{\boldsymbol{y}} = \boldsymbol{x}^{+T}\boldsymbol{w}$, where $\boldsymbol{x}^+$ and $\boldsymbol{w}$ are vectors of size $(d+1)$ to include the bias. The estimated parameter vector is $\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \|\boldsymbol{x}^{+T}\boldsymbol{w} - \boldsymbol{y}\|_2^2$.

However, when $(X^TX)^{-1}$ is nearly singular, $\boldsymbol{w}$ becomes sensitive to errors, resulting in overfitting. We can avoid it by giving a penalty to $\boldsymbol{w}$. This gives birth to ridge, lasso, and elastic-net regression.

2) Long-short term memory: Hochreiter *et al.* [9] introduced a particular memory cell capable of retaining information for long periods. The long-short term memory (LSTM) can read and write to its memory. More importantly, this memory never goes through an activation function. This effectively combats the trailing gradient problem [10] and makes the formation of this pattern very stable.

The original LSTM works with a series of input signals $\boldsymbol{x}_t$. It has a so-called hidden state $\boldsymbol{h}_t$ and cell state $\boldsymbol{c}_t$ of the same size as $\boldsymbol{x}_t$. The cell state $\boldsymbol{c}_t$ is the model's memory. The hidden state $\boldsymbol{h}_t$ is the model's prediction of $\boldsymbol{x}_t$.

The LSTM equations are defined by the following set of matrix equations;

$$\begin{aligned} A &= \boldsymbol{h}_t \|^1 \boldsymbol{x}_t && (1)\\ \boldsymbol{f}_t &= \sigma(W_f A + \boldsymbol{b}_f) && (2)\\ \boldsymbol{i}_t &= \sigma(W_i A + \boldsymbol{b}_i) && (3)\\ \boldsymbol{o}_t &= \sigma(W_O A + \boldsymbol{b}_O) && (4)\\ \boldsymbol{d}_t &= \tanh(W_d A + \boldsymbol{b}_d) && (5)\\ \boldsymbol{c}_{t+1} &= \boldsymbol{f}_t \circ \boldsymbol{c}_t + \boldsymbol{i}_t \circ \boldsymbol{d}_t && (6)\\ \boldsymbol{h}_{t+1} &= \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_{t+1}) && (7) \end{aligned}$$

where $\|^1$ is the concatenation operator, $\circ$ is put for the Hadamar product, σ is the logistic function, W are weight matrices, and $\boldsymbol{b}$ biases. The basic idea is that the model takes the input $\boldsymbol{x}_t$ and the previous prediction of the current input $\boldsymbol{h}_t$, updates its internal memory $\boldsymbol{c}_t$ to $\boldsymbol{c}_{t+1}$ and then makes a new prediction $\boldsymbol{h}_{t+1}$ based on $\boldsymbol{c}_{t+1}$, $\boldsymbol{h}_t$ and $\boldsymbol{x}_t$.

3) Support vector regressor: uses the same principle as SVM for regression problems [11]. We have the following objective for regression

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{d}\xi_i \quad \text{s.t. } |\boldsymbol{y}_i^T - \boldsymbol{w}^{+T}\boldsymbol{x})| \leq \epsilon - \xi_i, \quad (8)$$

where $\xi_i \geq 0, i = 1, \ldots, d$ is the observation distance from their correct decision boundary. the SVR tries to fit the best line within a threshold value, the ϵ-tube. Support vectors are data points closer to the hyperplane that influence the position and orientation of the hyperplane.

4) Shallow networks: can be viewed as misspecified nonlinear regression models simply taking the regression function as the output of a multi-layer perceptron (MLP). It concatenates 2 mappings: the first maps the data $\boldsymbol{x}_t$ into a regression vector $\boldsymbol{\phi}_t = \phi(\boldsymbol{x}_t)$ of fixed dimension, the second mapping, parameterized with θ maps the regression vector $\boldsymbol{\phi}_t$ onto the output $\boldsymbol{y}_t$:

$$\boldsymbol{y}_t = f(\boldsymbol{\phi}_t, \boldsymbol{\theta}), \quad t = 1, \ldots, N. \quad (9)$$

A useful choice of $\phi(\boldsymbol{x}_t)$ could be $\boldsymbol{\phi}_t = (\boldsymbol{x}(t-\Delta t), \boldsymbol{x}(t-2\Delta t), \ldots, \boldsymbol{x}(t-k\Delta t))^T$ where k sets the observation horizon and Δt is the sampling period. The estimation problem's complexity depends on the model structure's character $f(\cdot)$, usually chosen as a known continuous function on a compact $\boldsymbol{\theta}$ of Euclidean space.

The choice of approximation for the mapping $f(\cdot)$ is called the model selection.

The parameter vector $\boldsymbol{\theta}$ is estimated by minimizing the least mean square error (LMSE):

$$L_2(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \frac{1}{2N}\sum_{i=1}^{N}\|\boldsymbol{y}_i - \widehat{\boldsymbol{y}}_i\|^2. \quad (10)$$

which measures the quality of the approximation on the training set. L_2 is convex but returns large values when outliers are present.

The non-linear functions in the hidden and the output layers can be chosen as sigmoid or ReLU functions, the latter being given by $\mathrm{ReLU}(x) = \max(0, x)$. The parameters in the network are the weights that connect two consecutive layers.

Data normalization is mandatory when learning with shallow networks. This normalization considers the observations as normally distributed random variables with zero mean and diagonal covariance matrix. If X is the data matrix (observations in rows, variables in columns), then:

$$\widetilde{X} = \Sigma^{-1/2}(X - \mathbf{1}\boldsymbol{\mu}^T) \quad (11)$$

where $\mathbf{1}$ is a $N-$vector of 1, $\boldsymbol{\mu}, \Sigma$ resp. the mean vector and the variance-covariance of the data, calculated from $\Sigma = (X - \mathbf{1}\bar{\boldsymbol{x}}^T)^T(X - \mathbf{1}\bar{\boldsymbol{x}}^T)/(N-1)$.

Simulations were realized using Pytorch optimizer Adam[2] which was first introduced in [12].

[2]ADAM=Adaptive Moment Estimation.

B. Variable selection

The data are issued from reference and non-calibrated pollutant sensors. Both provide time series. To remove collinearity, we can exclude independent variables with high VIF values from our regression model, which, for the variable i, is written:

$$VIF_i = 1/(1 - R_i^2), \tag{12}$$

with R_i^2 is the squared correlation coefficient between $x(i)$ and the regression predictor $\widehat{x}_i$. The VIF directly measures how much the variance of each coefficient is inflated as compared to a situation with uncorrelated explication variables.

The variable selection method results in a stable and reliable calibration equation based on the less multicollinear variables. Many methods have been reported in the literature to select useful variables [13], such as, for instance, the discrete auto-correlation function Γ_{xx}:

$$\Gamma_{xx}(\tau) = \frac{\sum_{i=\tau+1}^{T}(x(i)-\overline{x})(x(i-\tau)-\overline{x})/(T-\tau)}{\sum_{i=1}^{T}(x(i)-\overline{x})^2/T}, \tag{13}$$

where T in the time horizon, $\overline{x}$ is the sample mean of x, $\overline{x}(t-\tau) = \sum_{i=\tau+1}^{T} x(t-\tau)/(T-\tau)$, τ is the discrete time-lag, can help with evaluating lag variables (see Fig. 2). The closer neighboring data observations are to each other, the higher degree of correlation they have.

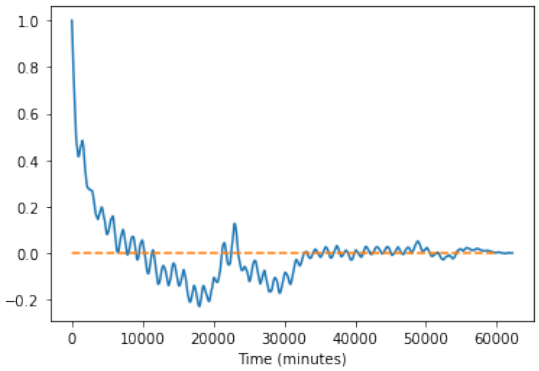

Figure 2: Half auto-correlation function for $PM_{2.5}$ signal.

To exclude the strongly linearly related data, the Pearson correlation coefficient between pairs of variables is calculated and presented in a matrix format Fig. 3. PM_1, $PM_{2.5}$ and PM_{10} are highly correlated

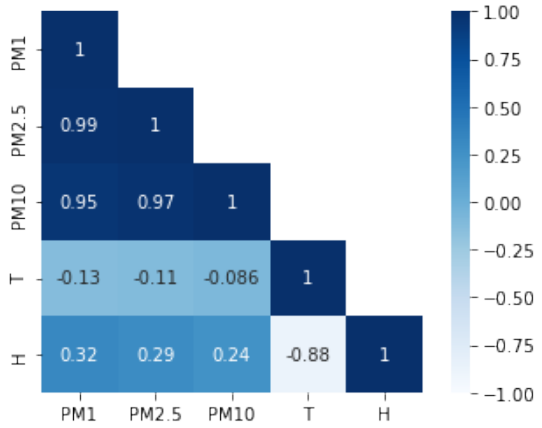

Figure 3: Correlation matrix of the different available variables.

(r >0.95), excluding the option taking all these variables as input variables for neural network calibration.

Algorithm 1 proposes to select the lagged variables used in the model. Suppose we have already determined a series of p lag periods $t_1, t_2, \ldots, t_p$. To add a new lag period, t_{p+1} is chosen such that $x_{t-t_{p+1}}$ have a high degree of correlation to x_t while a low degree of correlation to $x_{t-t_1}, x_{t-t_2}, \ldots$. Algorithm 1 details how to obtain a series of lag periods with these objectives in mind.

Algorithm 1: Time lag determination.

Result: a series of lag periods
initialization: Set upper limit of lag period N.
Let $t_1 = 1$ and $p = 1$;
$\Gamma_{xx}(k)$ given in Eq. (13)
while $t_p < N$ **do**
  $t_{p+1} = \arg_k \max \left(\frac{|\Gamma_{XX}(k)|}{\sum_{i=1}^{p} |\Gamma_{XX}(k) - t_i|} \right)$,
  $k = t_p + 1, t_p + 2, \ldots, N$;
  $p = p + 1$;
end

The approach is data-driven in that there is no *a priori* assumption about the models for the time series under study. It is beneficial for many practical problems because it is often easier to have data than to have good theoretical guesses about the underlying laws governing the systems from which data are generated.

Several tests were proposed in the literature for

choosing the number of input variables: for feed-forward networks, the explainable variables (in our case, temperature and humidity) are duplicated in as many as the primary measure. For example, if we choose to have p inputs for $PM_{2.5}$, the perfect fit is to have p coinciding inputs of temperature and p inputs of humidity.

C. Data preparation

This consists of the following steps:

1) Choose the size of the entries of the model.
2) Generate the $N \times (d+1)$ data matrix M from the original time-series. Each line is an observation. Mathematically, $M_0 = (\mathbf{m}_1, \ldots, \mathbf{m}_N)^T$, and $\mathbf{m}_k = (x_k, \ldots, x_{k+d-1})$.
3) Normalize the variables (standardization).
4) Shuffle the previous observations all together with a random permutation $\{\sigma_{(1)}, \ldots, \sigma_{(N)}\}$. Hence, $M_1 = (\mathbf{m}_{\sigma_{(1)}}, \ldots, \mathbf{m}_{\sigma_{(N)}})^T$ keep them in memory to recuperate the right order of the signal whenever we want.
5) Split then the data into training, validation, and test sets.

Note that the generalization of this process to more than one time series (*i.e.* several pollutants) is trivial.

D. Criterion that should be used for comparing models

To compare two losses with the same number p of variables, it is generally reasonable to compare the residual sum of squares (RSS) or R^2 – the smaller the value, the better the fit. The RSS is the average measure of the goodness of fit of the line to the data to get a predicted value $\widehat{y}_i$: RSS$= \sum_{i=1}^{N} \|\boldsymbol{y}_i - \widehat{\boldsymbol{y}_i}\|^2$. The best loss with $p+1$ variable is usually smaller than the best one with p. So using RSS inevitably leads to a model with too many variables, but not necessarily to improved predictions.

For models with various numbers of parameters, it is not. Mallow proposes to penalize additional variables by using C_p [14], which is defined by:

$$C_p = \frac{\text{RSS}}{\widehat{\sigma}^2} + 2p - N, \tag{14}$$

where $\widehat{\sigma}^2$ estimates the corrected residual variance of the model, p is the number of variables. C_p is a particular case of the Akaike Information Criterion, which may be used for choosing between statistical models in a more general setting. Good models will give small values of C_p [15]. Mallows has suggested that good models have a C_p value close to p. F-tests are also often used for comparing models of different sizes; if one of the models is a sub-model of the other [7].

R-squared (also known as the coefficient of determination) is a statistical measure of how close the data fit the regression line:

$$R^2 = \frac{\text{SSE}}{\text{SST}} = \frac{\text{SST} - \text{RSS}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{N}(\widehat{Y}_i - \bar{Y})^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}, \tag{15}$$

where SST, the total variance, is the sum of the variance explained by the regression SSE and the mean of the squares of the residuals RSS. In the case of the perfect fit, $R^2 = 1$.

The mean absolute error (MAE) measures the average of the absolute values of the differences between predicted and reference measurement:

$$MAE(Y, \widehat{Y}) = \frac{1}{N}\sum_{i=1}^{N} |Y_i - \widehat{Y}_i|. \tag{16}$$

A traditional grid search for hyperparameters tune-up was pursued to find the best SVR that fits into training data. The best kernel found and used for this study is radial basis functions (RBFs) for the best SVR.

IV. Results

The models used in this section are described in section III. Table III summarizes the model's architecture. For instance, the `3-25-25-1` ANNs means 3 input variables, 2 hidden layers composed each of 25 neurons, and a single output neuron with a linear activation function. As for LSTM, the components are LSTM cells. A summary of the number of estimated parameters is also given in Tab. III to compare the models' complexity.

$PM_{2.5}$, temperature and humidity feed the MLR, SVR, MLP Model1 and LSTM. But MLP-2 takes 9 inputs (3 of each $PM_{2.5}$, temperature, and humidity). These 3 inputs were chosen with a lag computed from the auto-correlation function in Fig 2).

To test the performance of the proposed calibration models, they were applied to the testing data never seen during the training.

Fig 4 shows the calibration marks on four different instances from the test set. MLR remains the weakest model studied as it does not include the various relationships of the feature space. The other algorithms

Table III: Model's structure for calibration with atmospheric conditions.

Models	Structure	# of params	Inp. var
MLR	ND	4	3
MLP 1	3-25-25-1	800	3
MLP 2	9-25-25-1	950	9
LSTM 1	3-25-25-1	8,226	3
LSTM 2	9-25-25-1	8,826	9

Table IV: Error metrics for calibration results.

Model | Scores	MSE	MAE	R^2
MLR	7.46	2.73	0.53
SVR	1.78	0.76	0.89
MLP 1	3.53	1.44	0.77
MLP 2	1.42	0.90	0.91
LSTM 1	1.98	1.04	0.88
LSTM 2	1.47	0.91	0.91

have comparable results that can be discerned by the metrics values, detailed in Tab. IV. The reference signal of $PM_{2.5}$ is too fluctuating, and the sensor's response is not learned at all standards (otherwise, the R^2 score will be 1). This also happens when the training data is not well sampled or not sufficient to contain all the needed information.

We should consider that even if results are good for a model, some technical constraints can not be valid every time in a deployability aim. The MLP2 has good performances but needs a 50-hour horizon, *i.e.* $\{x(t-3000), x(t1000), x(t)\}$ (time expressed in minutes). The SVR shall be better in these conditions, even if it does not outperform in this study. LSTM should be considered as the potential main algorithm for learning the sensor's historical dependence on time, which is not the case for other models. Its response strongly follows the reference time series under the presented time intervals. Furthermore, some artifacts are seen in Fig. 4. They are due to a lack of learning time, as we did not train it for the needed number of epochs.

One may also consider hybrid models, as proposed by Zimmerman *et al.* [5], to overcome too much complexity in "simple" cases.

V. Conclusion

This study demonstrates that ANNss and SVRs are many better-calibrating sensors than multivariate linear models due to their non-linearities. However, there are still some issues to be addressed, the effects of which can be seen in the result figures. In the majority of our experiments, the peak amplitudes are underestimated, the prediction of which could be improved by developing new metrics with constraints on high amplitudes.

How to choose suitable models? The data structure, including the choice of inputs and lags,requires many observations. Sampling could reduce the complexity of the system. Another amelioration to consider is the design of LSTM architectures: LSTM has almost the same results as MLP with many more parameters.

References

[1] H. Kim, M. Müller, S. Henne, and C. Hüglin, "Long-term behavior and stability of calibration models for no and NO_2 low-cost sensors," *Atmospheric Measurement Techniques*, vol. 15, no. 9, pp. 2979–2992, 2022. [Online]. Available: https://amt.copernicus.org/articles/15/2979/2022/

[2] B. Mijling, Q. Jiang, D. de Jonge, and S. Bocconi, "Field calibration of electrochemical NO_2 sensors in a citizen science context," *Atmospheric Measurement Techniques*, vol. 11, no. 3, pp. 1297–1312, 2018. [Online]. Available: https://amt.copernicus.org/articles/11/1297/2018/

[3] L. Spinelle, "Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide," *Sensors and Actuators B: Chemical*, vol. 65, 03 2015.

[4] R. Y., A. V. R.M., and J. Noel, "Development of a multiple regression model to calibrate a low-cost sensor considering reference measurements and meteorological parameters," *Environmental Monitoring and Assessment*, vol. 192, pp. 1–11, 2020.

[5] N. Zimmerman, "A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring," *Atmospheric Measurement Techniques*, vol. 11, pp. 291–313, 01 2018.

[6] C. Malings, "Development of a general calibration model and long-term performance evaluation of low-cost sensors for air pollutant gas monitoring," *Atmospheric Measurement Techniques Discussions*, pp. 1–30, 08 2018.

[7] N. Draper and H. Smith, *Applied Regression Analysis*. New York, USA: John Wiley & Sons, 1981.

[8] K. Vikas, M. Vasudev, and S. Manoranjan, "Significance of meteorological feature selection and seasonal variation on performance and calibration of a low-cost particle sensor," *Atmosphere*, vol. 13, no. 4, p. 587, 2022.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[11] I. Steinwart and A. Christmann, *Support Vector Machines*. New York: Springer, 2014.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[13] C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, 1st ed. Springer Publishing Company, Incorporated, 2018.

[14] A. Khuri, "Introduction to linear regression analysis," *International Statistical Review*, vol. 81, 2013.

[15] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 6, pp. 716–723, December 1974.

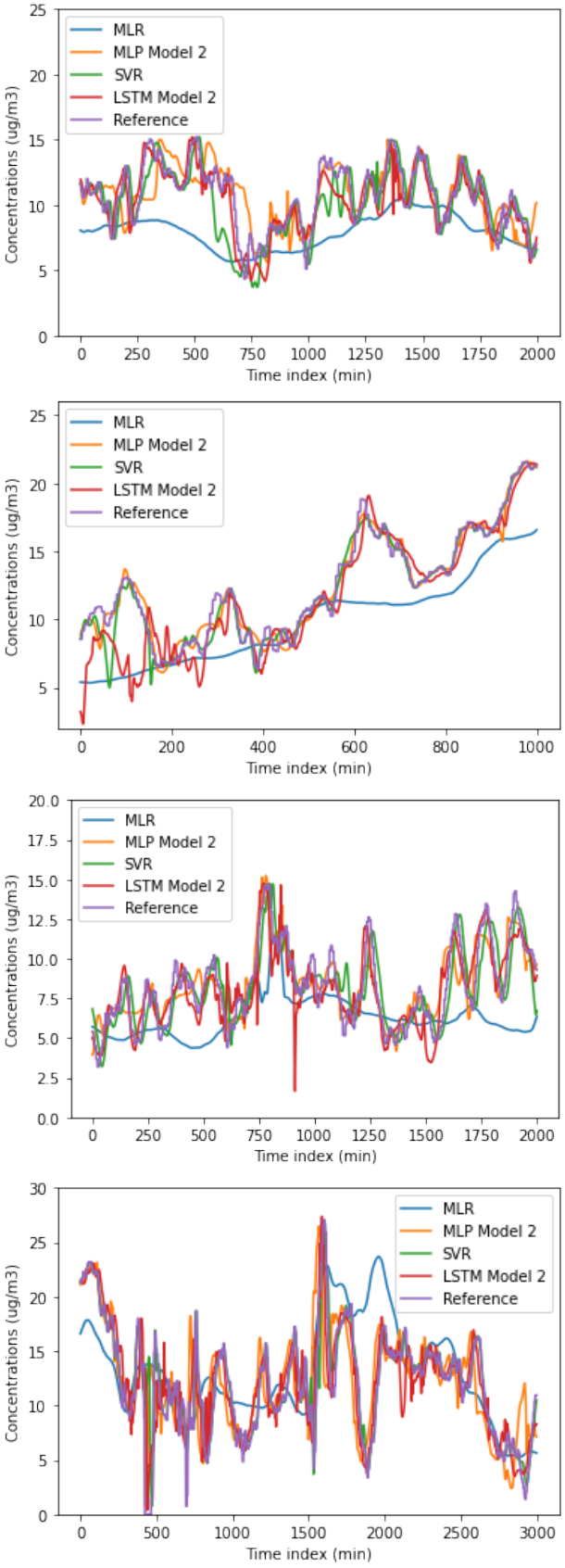

Figure 4: Results issued from the five models. They are presented in different periods from the test set.