$[][]^{\dagger}$

$[][]$

$\downarrow \hat{c}\hat{S}\langle\rangle equation($

T

$$(1)$$

deqnarrayTr tr$\infty], \phi||], \phi\hat{U}\phi||\phi\hat{Z}\hat{d}\mathcal{D}\bar{d}\bar{c}$

eqnarrayG

An open-source tool for evaluating calibration techniques used in low-cost air pollutant monitors

Daniel Tatsch[1], Alejandro Ramirez[1], Fernando Campo[2], Leonardo Hoinaski[2], and Evelio González-Dalmau[3]

[1]UNIVALI
[2]UFSC
[3]Cuban Neuroscience Center

January 6, 2023

Abstract

Low-cost air pollutant sensors suffer several interferences due to the variation of climatic elements. Recent studies look for calibration solutions based on different regression and classification machine learning algorithms. The present work brings together the implementation and extraction of performance metrics from these algorithms in a single open-source tool. Both the input data and parameters for each algorithm are automatically configured. This feature makes the tool compatible with any input dataset and removes the need to interact with complex codes.

Introduction

Low-cost air quality monitoring systems are usually composed of controllers without large processing capabilities. The reduction in production costs also compromises the electronic complexity and the robustness of their cases or compartments.

According to (*Pmid: 2902*, 2017), low-cost equipment mostly uses electrochemical gas sensors. The concern in using this sensor technology lies in the cross-interference between the pollutant being monitored and the other pollutants present in the same air sample. Besides, these sensors have a high sensitivity to the variation of climatic elements, such as temperature, relative humidity (RH), atmospheric pressure and wind speed (Mead, 2013).

A common solution to fix the errors caused by low-cost pollutant sensors is calibration. A calibration model can be built both based on the sensors response obtained in laboratories, outside environments and prior knowledge, such as information in the datasheets of the components or data from reference equipment. In field calibration (after the equipment has been installed) the focus is to develop algorithms that minimizes the sources of errors that compromise the performance of sensors previously calibrated. In this step, climatic elements sensors can be used to get a cross-analysis with their data and the monitored pollutant concentration (missing citation).

A bibliographic research was conducted to find the algorithms commonly applied in this type of analysis in electrochemical gas sensors monitored data. 18 articles, between 2010 and 2022 were selected and registered in a remote directory[1]. Figure 1 presents the ratio of the procedures and algorithms used in the selected papers. Most projects perform initial processing in hardware and firmware, such as the application of

[1]ff

correction equations provided by electrochemical sensor manufacturers that consider the signals generated by their electrodes.

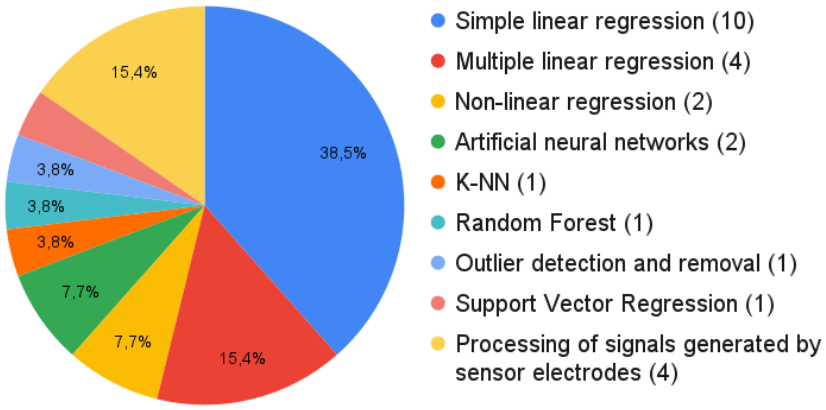

Figure 1: Graphic representation of the procedures and algorithms used in related works.

The authors used different metrics to evaluate and compare data obtained by the regressors and classifiers in their papers, such as the determination coefficient (R^2) (TIAN, n.d.; missing citation), Root Mean Square Error (RMSE) (ZIMMERMAN, n.d.; 2019) and Mean Absolute Error (MAE) (missing citation). Additionally, precision and recall metrics were also calculated to evaluate the performance of classification algorithms.

It was also identified in the literature review that this type of work typically focuses on implementing some specific algorithms to create your classification or regression models under very ideal laboratory conditions and taking these metrics into field calibrations (WEI & humidity conditions on electrochemical sensor response in ambient air quality monitoring. Sensors, 2018; *Issn 1424*, 2020). This approach by itself, in addition to being complex, is not enough to effectively calibrate devices used in open environments. Furthermore, the unavailability of the software tools used in the calibration process hinders both the understanding of the applied algorithms and the adaptation to other applications.

The need to invest in software that automatically generates calibration models is also linked to the evolution of the processing capacity of microcontrolled embedded systems used for data monitoring. (*Issn 2331*, 2021) shows, for example, a tiny Machine Learning (tinyML) application, where an artificial neural network was developed on an Espressif ESP32 (32-bit microcontroller) development platform. This application shows that regression and classification models can be adapted and applied to edge devices. This can ensure real-time calibration of data generated by sensors installed in the field, without the need for interaction with users or other devices for data collection and processing.

Structure of the algorithms analysis tool

A prompt tool was developed to simplify the analysis of the different algorithms found in the literature review and get a fair comparison between their results. The project was implemented in Python programming language and the principal libraries used were Pandas (read and analyze datasets), Matplotlib (generate graphics), Scikit-learn and Tensorflow (get statistical analysis and implement regressors and classifier algorithms).

Table 1 shows the algorithms implemented and their principal parameters. These specifications are also available in a configuration file (json format) and follow the default values recommended in Python libraries. This makes it possible to easily modify these values and, if necessary, append new parameters to the algorithms.

Table 1: Algorithms and parameters available in the tool.

Algorithm	**Parameters**
Simple linear regression	Single feature data
K-Nearest Neighbors (KNN)	Max number of neighbors
Random Forest	Max number of trees
Support Vector Machine (SVM)	Kernel function
	Regularization parameter (C)
Artificial Neural Network	Number of dense layers
	Dense layers size (list)
	Dropout size
	Activation and loss function
	Optimizer function
K-Nearest Regression (KNR)	Max number of neighbors
	Weight
	Epsilon
Support Vector Regression (SVR)	C-step
	Max C value
	kernel function

The operation of the algorithm analysis tool can be divided into two major parts: setup and execution. In the setup part user needs to select the desired pollutant and the classification or regression algorithm that will be used in the analysis. The path to the pollutant dataset file and the algorithm parameters are defined in the configuration file, which also set the target columns for both the pollutant and features (independent or reference data) datasets. A pre-processing step is performed in the following parts:

- Application of a moving average both in the features and pollutant data (according to the windows defined in the configuration file);
- data processing and cleaning:
  - Calculate previous statistical data (e.g. mean, median, standard deviation and correlation with features data);
  - Remove outliers and noisy values (outside sensor range);
  - Re-calculate statistical data after the clean-up.

For applications that use the classification algorithms, this pre-processing step also creates categorical values based on the target value distribution. 4 pollutant concentration levels are defined according to the quartile threshold values obtained in the processed statistical data.

The execution part includes the separation of the pre-processed dataset into training and testing data and the application of the selected algorithm. The first set of data is used to train the classifier or regressor while the second set is used to evaluate its performance. Both the graphs and the metrics obtained in the test step are saved in a separate file structure according to the selected pollutant and algorithm. Figure 2 shows de complete execution flow structured in the algorithm analysis tool considering both the setup and execution parts.

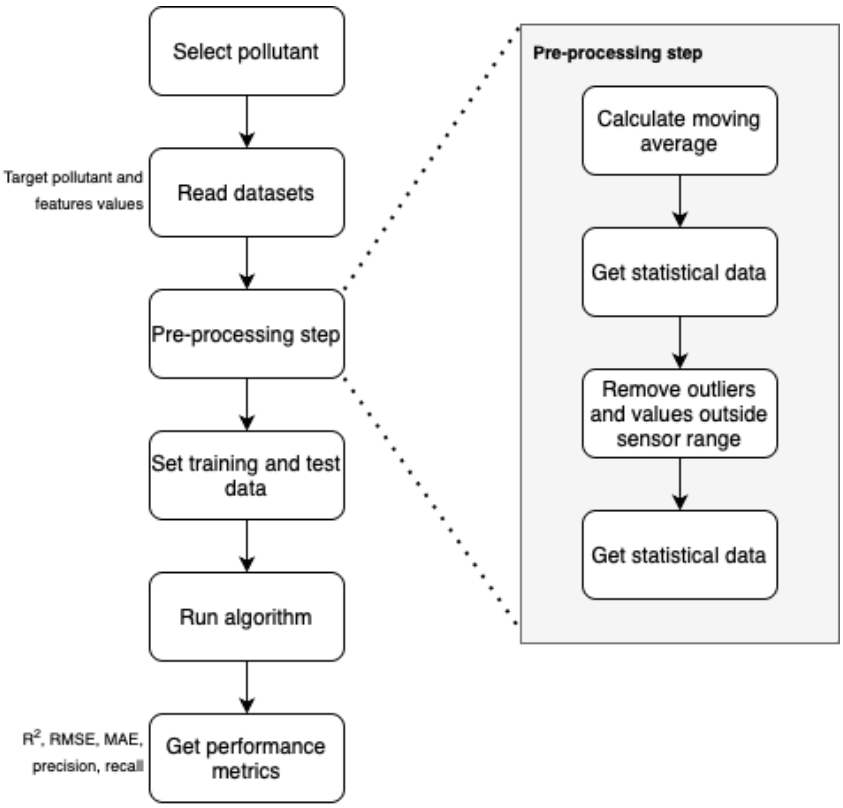

Figure 2: Execution flow of the algorithm analysis tool.

Setup and data analysis

A dataset obtained by a low-cost monitoring system installed in an external environment and developed by the Air Quality Control Laboratory (LCQAr) of the Federal University of Santa Catarina (UFSC) - Brazil were used to evaluate the tool workflow. The equipment has eletrochemical sensors from the Alphasense company (B4-family) for carbon monoxide (CO), hydrogen sulfide (H_2S), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), and ozone (0_3). In addition, it also has temperature and RH sensors attached to its internal part. The data provided by the pollutant sensors was also pre-processed on the firmware device, with Alphasense correction equations to temperature and RH. An initial analysis was performed considering 0_3 data as the target value and temperature and RH as features or independent values. The pollutant dataset has 4998 concentration data (in parts per billion - ppb) collected every minute from 2020/07/14 to 2020/07/18.

The datasets paths, their names, columns under analysis, sensor ranges and a list with different moving average windows were set in the tool configuration file. Figure 3 exemplifies the tool setup for 0_3 pollutant analysis with a moving average window equals to 60, to minimize noisy and abrupt changes.

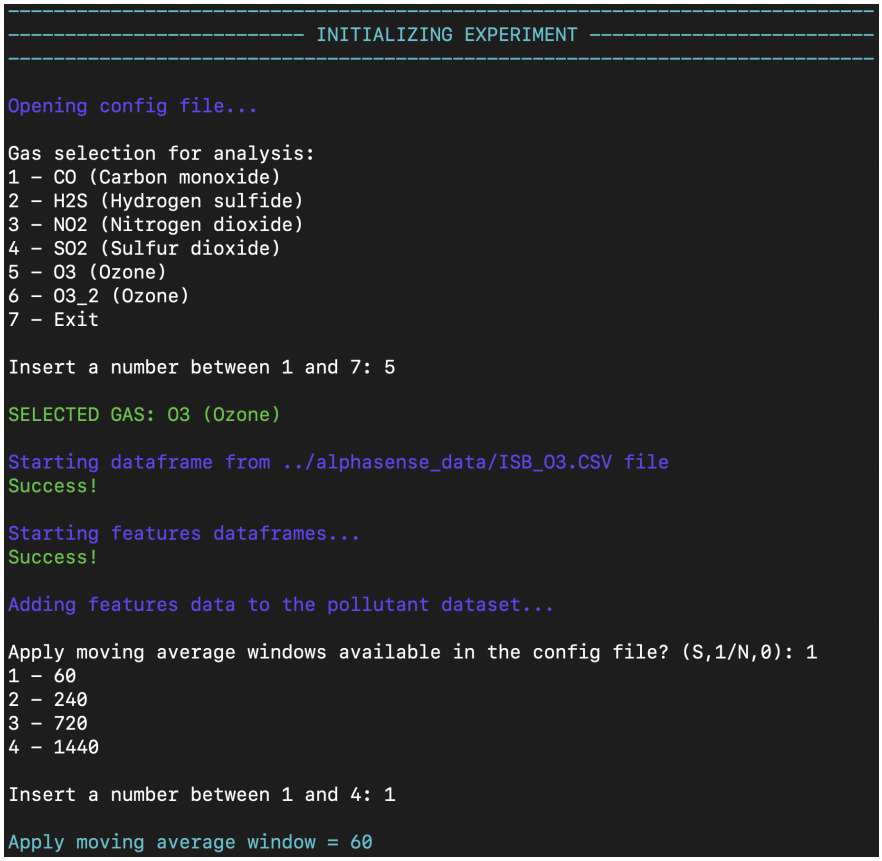

Figure 3: Tool execution - set pollutant and moving average window.

For this application, the temperature and RH datasets were used as feature data, enabling the analysis between them and the selected pollutant. However, any type of data, such as a calibrated reference sensor could be used to evaluate the electrochemical sensors installed in the equipment.

After setting up the previous information, the prompt tool automatically generates the following comparison graphics in Figures 4 and 5 from 0_3 data and each of the features selected.

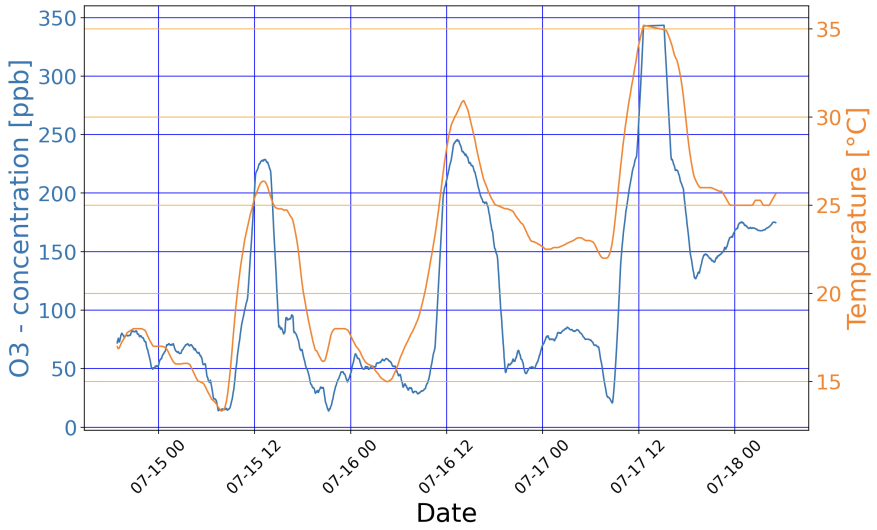

Figure 4: Ozone concentration [ppb] x temperature [°C].

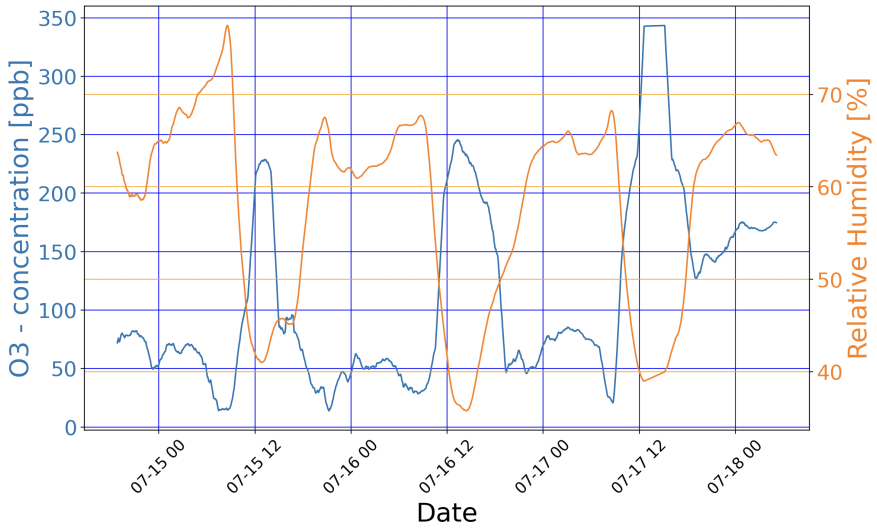

Figure 5: Ozone concentration [ppb] x Relative Humidity [%].

The pollutant data behavior shows clearly dependent on the climatic elements variation. In case of temperature data (Figure 4), this relation presents a direct proportional influence, while the pollutant relashion with RH (Figure 5) is inversely proportional, that is, as the RH value increases the pollutant concentration seems to decrease.

Following the procedures presented in Figure 2, the next pre-processing step is to calculate the statistical data from the dataset under analysis without any previous modification and also after removing the invalid data (outliers and concentration values outside the sensor range). Table introduces a comparison between these results.

The values highlighted in Table shows that even without the occurrence of data outside the sensor range, the process of removing the outliers improves the metrics obtained considerably. There is a significant decrease in

Statistic metrics extracted from algorithm analysis tool before and after remove invalid data.

Statistical metric	Pre-results	After remove invalid data
Mean [ppb]	112.63	104.43
Median [ppb]	77.43	75.82
Standard deviation	82.69	69.86
Variance	6837.73	4880.3
Outliers	153	0
Outside sensor range	0	0
Temperature correlation	0.874	0.852
RH correlation	-0.754	-0.728

the dispersion of the dataset to its mean (standard deviation) and in the correlations with temperature and RH. Furthermore, the negative value of the pollutant and RH correlation reinforces the inversely proportional relation between the monitored data.

Algorithms execution

The tool was developed to provide the application of one algorithm per execution. After the user input, the parameters algorithms calibration and the train and test operations are performed. These steps are executed considering the attributes presented in Table 1 and defined in the configuration file. As the simple linear regression only takes one feature in its analysis and this experiment considers two (temperature and RH), only multiple linear regression was applied.

The calibration process of KNN and Random Forest classifiers was performed considering the test accuracy metric. A loop was executed to variate their parameters and find 1 as the best number of nearest neighbors for classification in KNN and 11 as the number of trees used in Random Forest that achieved the highest accuracy. The SVM classifier was executed with the default value of C (1) and linear kernel function.

The artificial neural network structure was implemented according to the computer processing limitations. It was considered 1 dense layer with 64 neurons and an additional layer with a fraction of the input units to drop in 0.2 (dropout layer). The output dense layer was set with size 4 due to the previous categorical values calculated and the activation functions, optimizer and loss function were kept the same as shown in Table 1.

For the KNR regressor, the same procedure explained in the KNN classifier was executed to find the optimal number of nearest neighbors. In addition, the weight was configured as uniform so that the nearest and farthest neighbors have the same weight.

The SVR algorithm was configured to use its default values in the kernel function (Radial Basis Function - RBF) and the epsilon-tube within which no penalty is associated in the training loss function (0.1). The regularization parameter C was calibrated according to the epsilon value and the C value that minimized the MAE metric.

Results

Table shows the performance metrics of all algorithms. Precision and recall metrics are applicable only for classification algorithms and focus on the evaluation of false positives and negatives.

As noted in Table , even with a high R^2, all regression algorithms had error values much higher than those calculated in the classifiers. This difference can be explained by the large variance of data still present in

Statistic metrics extracted from the algorithm analysis tool.

Algorithm	R^2	**MAE**	**RMSE**	**Precisi**
Multiple linear regression	0.74	28.79	34.75	-
KNN	0.969	0.0392	0.198	0.86
Random Forest	0.961	0.0493	0.225	0.95
SVM	0.581	0.458	0.733	0.63
Artificial Neural Network	0.56	0.506	0.751	0.56
KNR	0.997	1.132	3.914	-
SVR	0.81	24.39	31.687	-

the pollutant dataset even after the pre-processing step. Furthermore, as the classifiers act on categorical values grouped according to the distribution of measurements, the error metrics tended to become smaller.

In the analysis of the classification algorithms, KNN and Random Forest obtained the best performances. Both got accuracies greater than 96% and despite having the MAE and RMSE slightly higher than the KNN, the Random Forest classifier obtained greater precision, reflecting the lower occurrence of false positives. SVM and artificial neural network obtained similar performances in the classifications, both in the calculation of the R^2 and in the MAE and RMSE errors. For the SVM classifier, the creation of a linear hyperplane did not separate classes optimally, as shown in Figure 6. Classifying data from a given class into another increased the occurrence of false positives, while not considering values outside the margins of each region increased the occurrence of false negatives.

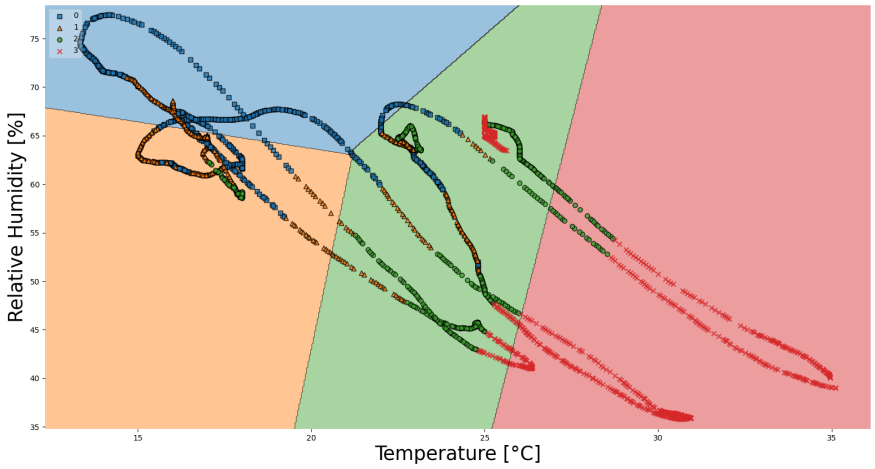

Figure 6: SVM - decision regions.

Conclusions

Here, a tool for analyzing different algorithms used in the calibration of data from low-cost air pollutant monitors was developed. This software was built to be compatible with any input dataset and allows full configuration of the implemented algorithms. It was validated using a dataset containing concentrations of the pollutant ozone (O_3), together with temperature and relative humidity data.

It was possible to demonstrate the dependence of electrochemical sensors on the variation of climatic elements and to apply pre-processing techniques to minimize this interference. The automatic generation of graphs and performance metrics made the analysis of the algorithms more efficient and enabled a fair comparison between them. The Random Forest classifier obtained the best results in the analysis of O_3, with accuracy, precision and recall equal or greater than 95%.

The source code of the tool is available in a public repository[2] for consultation and contributions. Some suggestions for new implementations are the analysis of the LCQAr datasets by varying the parameters of each algorithm and also the definition of the categorical variables of the classifiers based on the pollutant emission limits defined in the legislation of the corresponding region. The regression and classification models can also be extracted and adapted for applications that seek to perform autonomous calibration directly in the embedded systems present in the measurement equipment.

This work has been supported by The Coordination of Improvement of Higher Education Personnel (CAPES) project 139/11, the Foundation for Research and Innovation Support of Santa Catarina (FAPESC) project 2018TR499 and National Council for Scientific and Technological Development (CNPq) grant 315338/2018-0.

References

(2017).

The Use of Electrochemical Sensors for Monitoring Urban Air Quality in Low-Cost, High-Density Networks. (2013).

(2019).

Issn 1424. (2018).

(2020).

(2021).

[2] ff