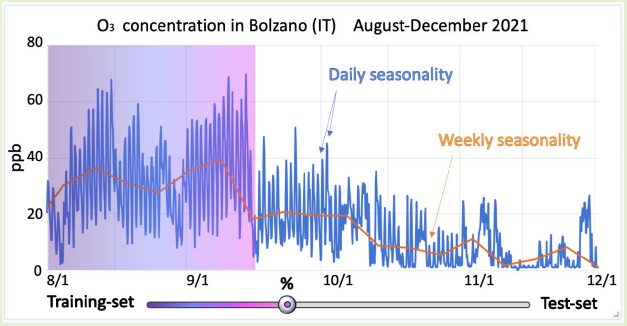
Minimized Training of Machine Learning-Based Calibration Methods for Low-Cost O_3 Sensors

Stefano Tondini, *Member, IEEE*, Riccardo Scilla, and Paolo Casari, *Senior Member, IEEE*

***Abstract*—Low-cost sensors (LCSs) show a huge potential toward enabling the pervasive and continuous monitoring of crucial environmental parameters, supporting environment preservation, and informing citizens' well-being through ubiquitous air quality data. The main drawback of LCSs is that their data is usually biased, even if LCSs are calibrated by their manufacturer at production time. More accurate in-field calibration methods based on machine learning (ML) and neural networks (NNs) are being considered in some recent studies. They typically imply LCSs colocation with reference measurement stations certified by environmental agencies. Due to seasonality effects, however, the correlation between LCSs and their reference may rapidly degrade once the LCSs are moved from the calibration site, making even really accurate calibrations useless. In this work, we specifically target this problem by optimizing the training settings of the most popular ML and NN calibration models for LCSs when a sequential split schema is adopted to separate training and test sets. Then, we assess the degradation of the calibration over time based on the R^2 score, when the splitting of the dataset between training and test sets is different from the classical 80%–20% ratio. This method is applied to real data gathered from an O_3 sensor deployed in co-location with a certified reference station for a period of six months. Eventually, we show that, in the case of long-short term memory NNs, using 20% of the dataset for the training is a trade-off condition that minimizes the calibration effort and still yields a robust and long-lasting calibration.**



***Index Terms*—Calibration method, harmful pollutants, low-cost sensors (LCSs), machine learning (ML), neural networks (NN), prophet forecasting, sequential split training, time series.**

I. Introduction

AIR quality is crucial to citizens' well-being as well as to environment preservation. According to the most recent WHO reports, the emission of harmful pollutants is one of the main causes of health-related diseases [1]. The impact of air quality, both at a local and global scale, demands distributed and continuous monitoring [2]. Current attempts rely on networks of air quality monitoring stations managed by regional environmental agencies, which make it possible to track the most common pollutant gases with certified accuracy [3]. However, these stations typically host high-resolution equipment [4] and are thus expensive both to establish and to maintain. As a result, installing a dense network of certified air quality monitoring stations is an ineffective approach for dense or ubiquitous monitoring, especially in urban areas. Rather, it is common to deploy only one or a few acquisition points per city, which are not sufficient to discriminate, e.g., the different pollution levels expected from different districts, worse air quality in the surroundings of industrial clusters, relatively higher air quality at residential areas located far from crowded streets, and so on [5]).

The problem could be overcome by means of networks of autonomous sensor nodes, namely, wireless sensors networks (WSNs) [6], [7]. This approach is getting more and more relevant not only among the scientific community but also at the level of citizen science [8]. In some cases, WSNs can be also made of mobile monitoring devices deployed in public areas to expand the survey domain and limit the hardware needs at the same time [9].

In accordance with Internet of Things (IoT) principles [10], [11], [12] and high-efficiency information/power management

Manuscript received 31 October 2023; accepted 16 November 2023. Date of publication 12 December 2023; date of current version 31 January 2024. This work was supported in part by the Cassa di Risparmio di Bolzano Foundation in collaboration with the NOI Techpark Bolzano, the Südtiroler Wirtschaftsring and Rete Economia Alto Adige under the Fusion Grant scheme; and in part by the European Union's Horizon 2020 programme under Grant 99987. The associate editor coordinating the review of this article and approving it for publication was Dr. Xiaojin Zhao. *(Corresponding author: Stefano Tondini.)*

Stefano Tondini was with the Center for Sensing Solutions of Eurac Research, 39100 Bolzano, Italy. He is now with the Department of Electrical Engineering, Eindhoven University of Technology, 5612AZ Eindhoven, The Netherlands (e-mail: s.tondini@tue.nl).

Riccardo Scilla is with the Center for Sensing Solutions of Eurac Research, 39100 Bolzano, Italy (e-mail: rscilla@eurac.edu).

Paolo Casari is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: paolo.casari@unitn.it).

Digital Object Identifier 10.1109/JSEN.2023.3339202

schemes [13], [14], [15], [16], [17], [18], WSNs happen to be way more affordable than traditional monitoring stations, especially when they are made of low-cost sensors (LCSs) [19], [20]. Being typically smaller, they are also easier to install in urban areas. LCSs can be acquired by private citizens and associations thereof and networked together to cover areas that would be otherwise sparsely sampled, as is done already for RF spectrum utilization sensing [21]. By doing so, accurate and geographically dense environmental sensing can be expanded also to peri-urban areas.

The main drawback of LCSs is that collected data are not as accurate and unbiased as those of calibrated environmental stations. In fact, most LCSs for pollution monitoring are based on metal–oxide–semiconductor (MOS) or electrochemical (EC) working principles [22], [23], [24], which often lack accuracy and reliability [25]. This is because manufacturers calibrate their LCSs within a gas testbench just against the target pollutant, and only rarely they do provide corrections, e.g., for temperature variations [24]. However, calibration setups are very different from real operational conditions, where the influence of external (environmental) parameters strongly affects the measurements. As an example, LCSs for harmful pollutants monitoring are sensitive to relative humidity (RH), as well as temperature, and often suffer from cross-sensitivity [25], [26], [27], i.e., they react also to the presence of other nontarget gases leading to measurement alterations. Specifically, it has been reported that O_3 sensors undergo an oxidation process in the presence of NO_2 [28], and CO measurements are correlated with H_2 in urban environments [27]. Another issue is the stability of LCSs over time. Recent studies have demonstrated that the consumption of the reagents in some specific EC sensors leads to read-out drifting (aging) [28], [29]. Still, EC sensors age faster when exposed to high-temperature and low RH [30]. Additionally, some gas sensors have cross-sensitivity to other gases that are not included in their datasheet. For example, CO sensors can be triggered by alcoholic isopropyl base solvents.

Many strategies can be adopted to cope with the above issues. For instance, recursive calibrations can be put in place by periodically comparing LCS outputs against some reference (REF) instruments in a supervised environment. By surveying different environmental conditions, it would then be possible to extract a (linear or nonlinear) correction function for the LCSs [31]. However, the full mapping of the sensors' response to a large variety of environmental conditions is highly time- and resource-consuming, making this a nonviable approach. For this reason, the most recent approaches to LCS calibration are based on machine learning (ML) techniques. The main idea is to place the LCSs near a certified monitoring station so that they share the same environmental conditions and, ideally, sample the same pollutant concentrations. Then, it is possible to train a supervised ML model to correct the LCSs' output. In this context, ML makes it possible to factor in any set of environmental parameters, including additional pollutants besides the target one, in order to make the calibration model sensitive to such interferences [32]. Moreover, based on a data-driven approach, such a system can be implemented without the need for a complex experimental setup. It also eases the mitigation of unwanted effects such as drifting over time, if proper periodic recalibration policies are adopted. ML-based calibrations can be operated directly on the deployed sensors (edge scenario) [33], [34] or on the time series stored in a database (cloud scenario) [35], [36], [37].

It is worth mentioning that the application of remote calibration to low-cost IoT sensors is taking momentum. Among others, paradigms like distant calibration [38], hierarchical networks and mobile buddies serving as calibration devices [39], and calibration model transfer over space [40], and over different pollution regimes [41] are emerging and being benchmarked.

A. Motivation of the Work

Considering the above context, in this article we aim at improving LCS accuracy by means of ML-based methods. Urban monitoring is a particularly suitable use case, as certified air quality stations are available at least at a regional level, and offer an excellent reference to compare sensing capability in an operational environment. However, installing LCSs close to reference stations is not always possible, either for formal reasons (limited hosting permissions from the environmental agencies that manage the sites) or for practical reasons. (There would be no point in a permanent deployment of LCSs near the monitoring station, besides the validation of the process itself.)

Given the last points, our investigation focuses on how to minimize the calibration time (i.e., training) for ML-assisted LCSs. In this work, we consider the specific case of O_3 sensing, but the same methodology can easily be extended to other pollution sensors, as well. For this purpose, we first benchmark the most popular AI-based calibration models, in order to select the most convenient approach for our problem. This task allows us to fine-tune the AI model selection based on the characteristics of O_3 sensors. Then we analyze the splitting ratio of training and test sets over a period of three months in order to estimate if a trade-off condition can be found for which the LCS calibration holds (i.e., the correlation between the LCS and the reference lies above a threshold value) even if the time the LCS is colocated with the reference instrument is limited. We can compare the performance of different splitting ratios on the same LCS as it was colocated with the reference station for the whole duration of the study. We can mimic the sensor being moved far apart from the calibration site when we consider only a subset of the reference time series (ground truth) to train the ML models. Due to the duration of our field campaign, we are also able to address the onset of calibration issues when seasonal effects appear. Our overarching goal is to enhance the efficiency of LCS-based sensing, as well as limit the need for human intervention on low-cost WSNs.

Specifically, the contributions of this article are as follows:

1) setting up a preprocessing pipeline for time series, which encompasses data harmonization, outlies and gaps removal, as well as features selection;
2) benchmarking different ML- and neural network-based calibration models, namely, multiple linear regression (MLR), random forest regression (RFR), support vector

TABLE I
STATE-OF-THE ART AI-BASED O_3 SENSOR CALIBRATION APPROACHES

Auth	Model	Targ	Feature (pollut)	Feature (atmos)	Train (day)	Test (day)	R^2
[43]	MLR	O_3	O_3	T, RH	6	2	0.81
[43]	RF	O_3	CO, CO_2 NO_2, O_3	T, RH	6	2	0.99
[44]	MLP	O_3	CO, NO O_3	RH	7	7	0.92
[54]	MLP	O_3	CO, CO_2 NO, NO_2 SO_2, O_3 PM10	T, RH	12		0.86
[54]	RF	O_3	CO, CO_2 NO, NO_2 SO_2, O_3 PM10	T, RH	12		0.97
[62]	LR	O_3	O_3				0.92
[63]	MLR	O_3	O_3, VOC	T	2	14	0.16
[63]	MLP	O_3	O_3, VOC	T	2	14	0.91

regression (SVR), multilayer perceptron (MLP), long-short-term memory (LSTM), and convolutional neural network (CNN);

3) studying the effect of temporal pivoting and of random splitting and sequential splitting between the training and test sets;
4) assessing a trade-off condition for the time interval spanned by the training set that yields a long-lasting calibration.

The latter two points characterize our study, since, in contrast with previous works (see Table I), we report the experimental outcomes corresponding to varying the ratio between training and test, investigating other conditions besides the classical 80%–20%, by also applying a sequential splitting schema to sort training and test sets. It is also worth mentioning that, unlike the majority of the studies in this field, the LCSs have been deployed in such a way that their sensing condition is identical to that of the REF instruments, i.e., in supervised constant air-flow (see Section II-A for further details), that guarantees the reliability of outcomes. After an extensive benchmarking, we show that the resilient calibrations for an O_3 sensor can be achieved with LSTM even if the part of the dataset devoted to the model training is reduced down to 20%. The metric we use to assess the calibration degradation over time is the weekly averaged R^2 score.

In this article, we do not address the analysis of the calibration performance when the LCSs are moved to different locations with respect to the calibration site with different distributions of pollutants. We refer to the relevant work of deSouza et al. [41] on the topic for such a discussion.

B. Structure of the Paper

In the following, we survey related work in Section II, before reporting on the materials and methods used to collect and preprocess the experimental data in Sections III and IV, respectively. The operations related to the implementation, optimization and training of the ML models and NNs are also detailed in Section IV. Section V analyzes the results obtained from the comparison of the different approaches, and discusses the minimum the training to achieve a reasonable/reliable calibration. Finally, Section VI outlines some future perspectives on the topic of AI applied to sensor time series.

II. RELATED WORK

In the recent literature, several ML-based strategies have been proposed with the aim of calibrating LCSs. The ML methods used in these works are mainly linear models such as linear regression (LR) and MLR, but also nonlinear models such as RFR, SVR, and neural networks (NNs) [42]. These models have to be trained against a reference data source during the calibration time period.

The majority of the works exploit MOS [43] or EC [44] sensors that are located nearby certified REF instruments/stations, which share the same environmental conditions and serve as ground truth for multiple pollutant concentrations at a time [45]. Environmental parameters such as RH, temperature (T) [46], [47], or wind speed (WS) [22] are often included in the training process. Some studies carry out such calibration in indoor environments as well [48], [49]. Commonly, the target pollutants are CO, CO_2, NO, NO_2, O_3, and SO_2 and particulate matter PM1, PM2.5, and PM10.

There is still no agreement on the sensor sampling frequency, which mostly varies between 15 s [44] and 30 s [19], up to 1 min [11]. However, such values are oversampling the reference measurements, which are made available (after averaging) on a 10-min or 1-h basis. To cope with this aspect, resampling or averaging is also applied to the LCS time series. A usual choice is a 1-min average [22], [45], while other studies increase the duration of the averaging window to 1 h [50]. In this study, De Vito et al. [50] addressed adaptive ML strategies to extend the validity of calibration models for air quality multisensor systems. In such a scheme, the authors implemented continuous learning through periodical calibration updates on two algorithms, namely the standard shallow, feed-forward, neural network and the extreme learning machine. In any case, it is crucial to achieve temporal coherence between the LCS and the reference time series [51]. Because LCS measurements often include saturated values or outliers, such operations as smoothing, filtering and outlier detection are typically applied during a preprocessing phase [17], [32], [50] in order not to limit the performance of ML-based calibration models.

The debate about whether or not the performance depends on the training length remains open. The point has been tackled since the introduction of ML approaches for in-field sensor calibration [52], [53], [54]. Many works on multisensor calibration reported comprehensive tables on the outcomes of ML and NN models applied to datasets from different field campaigns [32], [55], [56]. However, the training verus test splitting ratio is not always reported as a parameter. In turn, it is not straightforward to compare the results of different studies, as the percentage of each dataset used for training and test sets may be noticeably different from case to case.

The 80/20 and 75/25 splitting ratios are still a widespread standard for testing ML approaches applied to LCSs calibration, e.g., [46], [51], [57], [58], [59], [60]. Moving apart

from classical splitting ratios, Spinelle et al. [45] used a cross-validation approach for ANNs, dividing the dataset into subsets of two weeks, the first used for training (50%) and the second for test (50%) to avoid artifacts from the training process. 50/50 splitting ratio was also used by Aula et al. [59] and De Vito et al. [61] for CO and NO_2 sensor calibration, and by Han et al. [62] to validate NN performances on CO, O_3, NO_2, and SO_2 sensors. Still Lin et al. [63] explored 1/2, 1/3 and 2/3 as splitting between test and training for calibration of $NO_2 + O_3$ sensors.

Maag et al. [64] tested different recalibration frequencies over a period of 12 months, using a training dataset of four weeks. They obtained an increasing error by decreasing the recalibration frequency, suggesting a trade-off value for the calibration duration from 12 weeks (67%) to 16 weeks (75%). 20/80 and even lower ratios were also searched from several recent studies, for instance to emulate the condition where the LCSs are collocated with a reference station for a certain period and then deployed in the field for standalone pollution monitoring [65], [66] and to perform pervasive air quality monitoring through high-resolution hybrid networks leveraging ML and IoT principles [61].

Stratified split is also gaining momentum to ensure that the distribution of classes is preserved in both the training and test sets [67], [68].

For short-term operations, most authors converged on four weeks of training for three–four months of maximum deployment, as the onset of seasonal influences (especially in periods with low concentration of pollutants) calls for periodic recalibration routines.

The LR model, together with its extension to MLR, is typically used as a comparison baseline for more complex models [23], [44], [51]. For instance, Lin et al. [63] exploited LR for the calibration of NO_2 and O_3. As LR models have a single explanatory variable, the NO_2 and O_3 measurements were input separately, without the possibility to address cross-sensitivity issues. Zimmerman et al. [44] compared an RFR model with an MLR model, choosing CO, CO_2, NO_2, and O_3 as targets. For the RFR all the pollutants (CO, CO_2, NO_2, and O_3) and T and RH were considered as input features to train the algorithm, while for the MLR the single pollutants (and T and RH) were processed one at a time. Bigi et al. [23] showed in their considered conditions and scenario that nonlinear models SVR and RFR outperform MLR in the case of NO and NO_2.

It must be mentioned that this holds for short-term calibrations, but for-long term ones (i.e., spanning over multiple seasons) MLR may outperform nonlinear approaches. For instance, Zimmerman et al. [44] proposed a hybrid calibration model combining RF and MLR to possibly cope with higher concentrations than those encountered in the training window by RF. The issue of long-term performance assessment for EC sensors calibrated in-field through ML approaches has been extensively addressed by De Vito et al. [61], concluding that yearly recalibration is necessary to prevent unacceptable worsening of accuracy.

NN models have also been extensively searched for the calibration of LCSs, showing promising results. In particular, Yamamoto et al. [69] used an MLP to calibrate a temperature LCS using environmental factors such as solar radiation (SR), RH, WS, and rainfall. However, the study reports that the calibration reliability drops when the LCSs have moved far apart from the calibration site. To solve this issue, Park et al. [70] developed a hybrid model, combining an MLP with an LSTM recurrent neural network (RNN). The resulting model, named HybridLSTM, embeds a component that enables the extraction of time dependencies from the analyzed data. This novel model was compared to MLR and MLP, showing an increase in the correction (i.e., calibration) quality. A similar result was achieved by Yu et al. [46], who incorporated a CNN and a gated recurrent unit (GRU) for CO and O_3 calibration. The authors report that the CNN makes it possible to capture short-term patterns in the input features, while the GRU allows extracting long-term periodic patterns. This approach outperformed LR, MLR, SVR, and LSTM, showing improvements in all metrics. Lee et al. [51] developed a model called segmented model and residual treatment (SMART) for PM2.5 (particulate matter of 2.5 μm diameter), using ambient light (AL) together with T and RH as additional input features. SMART aims to lower the weaknesses of the MLR and MLP models (linear and nonlinear). At first, the MLR and MLP are trained against a set of AL, T, and RH conditions. A residual map is then calculated to point out the models' performance against the different environmental conditions. Eventually, the best model is selected for each AL, T, and RH combination.

Antonini et al. [49] followed a different approach based on a classification problem. Their goal was to benchmark eight CO sensors by comparing them against a discrete range of concentrations (0, 2, 5, 10, 15, 20, and 25 ppm) as a function of RH. They used an MLP NN to associate the measurement from the raw sensor with a discrete concentration within the range. Interestingly, the authors exploited some genetic algorithms (PSO and NSGA-II) for the optimization of the MLP.

A summary of the state-of-the-art approaches applied to O_3 LCSs is shown in Table I.

III. Materials

A. Data Collection

The LCSs benchmarked in this work have been hosted within a certified air quality monitoring station located in the city center of Bolzano, Italy (lat. 46.495633, long. 11.340195, elev. 262 m). The station is managed by the local environmental agency (APPA Bolzano, whose help we acknowledge) and is equipped with several high-resolution instruments that comply with the protocol for standardized acquisition released by the European Environment Agency [71].

Fig. 1 shows the installation of the LCSs, which mimics the sampling conditions of the REF instruments. The air monitoring station includes two suction ducts, which let in a constant airflow from the roof and release it to the ground [Fig. 1(a)]. This brings similar air samples to the LCSs and the REF, respectively, so as to yield the same measuring conditions.

From now on, we will use the following shorthands.

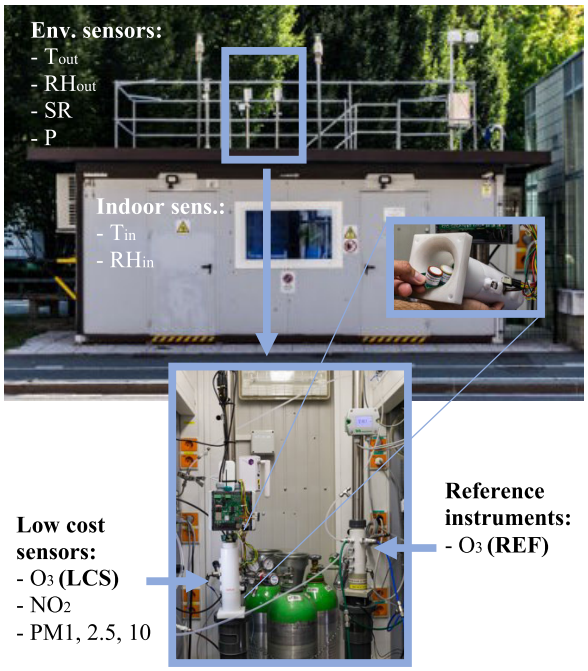


Fig. 1. Installation of the LCSs within the monitoring station of Piazza Adriano, Bolzano, Italy. The two duction pipes feed the REF instruments and the set of LCSs with the same controlled airflow, which is then released toward the ground. An indoor picture is shown as an inset, together with a magnification of the LCSs hosting. The different parameters available for the current study are also listed.

1) O_3REF: The reference high-resolution O_3 analyzer ThermoSCIENTIFIC 49i hosted by the monitoring station.
2) O_3LCS: the EC LCSs Alphasense OX-A431, deployed inside the monitoring station.

The core unit of the O_3LCS acquisition system is a Raspberry Pi 4, connected to the sensors via the I[1]C bus. It also encompasses additional pollution sensors, namely, NO_2 (Alphasense NO2-A431) and PM1, 2.5, 10 (Alphasense OPC-N3), and some sensors of microclimatic parameters, i.e., Tin/Rhin (Sensirion SHT31), Tout/Rhout (Galtech PM15PS), atmospheric pressure (Bosch Sensortech BMP388), and SR (Apogee SP 420 Smart). The latter sensors are placed on the roof of the monitoring station within a radiation shield, beside the pyranometer that is directly exposed to the sunlight. The time series originating from the extra sensors will be also exploited as additional input features for the ML models tested in our work.

Concerning the sampling rate, the O_3LCS sensors are acquired every 10 s, while the O_3REF measurements are made available already with a 10-min average. The time series analyzed in this study range from August 1, 2021 to December 30, 2021 [72], which represent the period during with the LCSs have been hosted by the air quality monitoring station.

B. Data Workflow

The collected data from the O_3LCS setup, together with the ones from the O_3REF system, are sent in real time (via an HTTPS POST request through a wireless connection) to an InfluxDB[2] instance, within two separate measurement tables. This is one of the best options to date to collect and handle large volumes of time-stamped data gathered by multiple sensor sources. This way, the O_3LCS's and O_3REF's time series can be made available to a common processing environment by another HTTPS call to InfluxDB's API.

The processing environment chosen for this study is JupyterLab[1], a standard for data scientists, which provides an interactive notebook layout to embed text, graphics, and executable source code in the same resource. The JupyterLab is mounted on the Docker container orchestrator Kubernetes, which supports automated resource management and scaling when a multiuser approach is implemented.

For a quick check on the time series integrity, as well as for a basic visualization of the raw data, a Grafana[3] visualization dashboard is coupled to InfluxDB. This web application makes it possible to compare the different data sources on the same or different panels created by interactive query builders. Moreover, this tool has an early warning feature that notifies the users about time series flow interruption via e-mail or instant messaging applications such as Telegram. For this specific implementation, the warning threshold has been set to send a notification when the sensor posts no new data for more than 10 min. This, in turn, speeds up possible restoration intervention and reduces the chance to lose time series continuity.

Finally, a GitLab[4] repository is used to collect and share the pipelines configured for our analysis.

IV. Methods

As previously introduced, calibration is implemented as a supervised regression task, where O_3LCS measurements are input features, whereas O_3REF measurements provide ground-truth values for the target harmful pollutant. The ML models' output is then an engine that calibrates the O_3LCS' measurements.

In such a framework, we have set up different ML pipelines, to select the model that yields the best performance, even when training data spans a limited period of time. Each pipeline is composed of different operational steps that sequentially preprocess the raw data, implement, optimize and train the ML models, generate predictions based on the real data, and finally assign different performance scores for easier comparison. A simplified pipeline schematic is shown in Fig. 2. Here, the "EXTRA" data source corresponds to the time series available from sensors for additional pollutants other than O_3. In the following, we detail all the above steps.

A. Data Preprocessing

1) Harmonization: This operation prepares the raw input data for the predictive ML models. As O_3LCS data may contain irregularities or corrupted data points (e.g., noisy or inconsistent values), it is fundamental to carry out a data integrity analysis to identify issues, make decisions on the

[1]https://jupyter.org/

[2]https://www.influxdata.com/

[3]https://grafana.com/

[4]https://gitlab.inf.unibz.it/

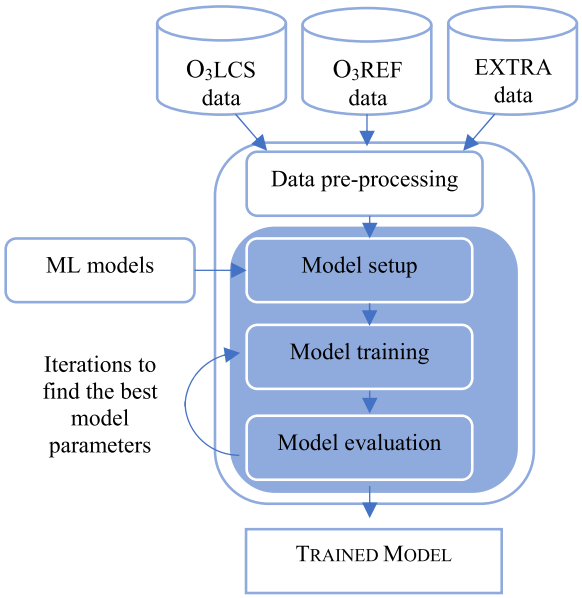


Fig. 2. Schematics of the pipeline implementation from the preprocessing of the input data for the ML models' optimization (in the light blue highlight).

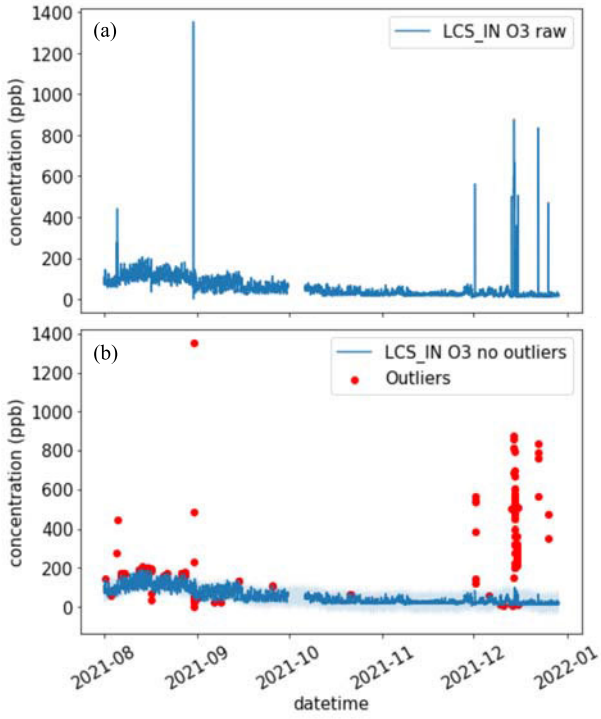


Fig. 3. LCS time series: (a) before and (b) after applying Prophet's outlier removal process.

processing steps, and accordingly clean up the raw data. Conversely, O_3REF data are already preprocessed by the Local Environmental Agency of Bolzano, before their release.

Moreover, the O_3LCS and O_3REF sensors have to be aligned in terms of sampling rate and units. To do so, O_3LCS sensor data are time-averaged over a window of 1 h, and the pollutant concentrations are converted to parts-per-billion, for compliance with O_3REF values.

Finally, it is possible to aggregate all data (O_3LCS and O_3REF) by joining all the data frames into a unique structure, so as to facilitate further processing.

2) Outliers and Gaps Removal: At this stage, we remove outliers and gaps from the time series, as they could otherwise distort and mislead the ML training process, resulting in longer training time and lower accuracy [73], [74], [75]. In this work, such problems have been addressed through Facebook's Prophet tool [76]. This tool decomposes the time series into three main components: trend, seasonality, and holidays (anomalies that must be considered unique events as they deviate from normal behavior). Based on the specifications described in [76], Prophet is more flexible and faster than the more common ARIMA model, as it easily allows to set multiple seasonal frequencies by defining an additional input parameter. Unlike ARIMA, Prophet is robust against missing values. After analyzing the nonlinear trends in the time series through yearly, weekly, and daily seasonality effects, it can reconstruct the full time series by adding the different time series components with an uncertainty interval that factors in reconstruction errors. This confidence interval makes it possible to discriminate outliers in the measurements, by considering those data points which do not lie in the uncertainty range. Fig. 3 shows how Prophet performs on the O_3LCS time series.

This procedure is not applied to the REF time series, as data from the monitoring station are already preprocessed by the local environmental agency. Conversely, we observe that the red dots in Fig. 3(b) represent sensor anomalies, as the O_3 peak lues seen in Fig. 3(a) correspond to unrealistic concentrations compared to the O_3REF time series. A possible cause of anomalies is that the sensors get a small negative value, which can happen for many reasons, and in case of an unsighted integer it just becomes a huge value. A more detailed overview on different outliers/anomalies detection methods is given in [77], [78], [79], and [80].

The second issue that needs to be addressed is the removal of gaps in the time series. This can be solved by filling in the missing data points or removing the gaps by cutting/merging the time series. Both solutions present additional issues: filling the missing data usually requires introducing synthetic data from imputation techniques that account for the time series values before and after the gap. This solution is not suitable for our purpose, because we focus on real data coming from in-field LCSs, and introducing synthetic data in a time series of real measurement may lead to artifacts in the ML model outcomes.

A more suitable solution is then to remove the gaps. However, this approach has to be performed carefully because the pollutant measurements are affected by heavy seasonal trends, with both daily and weekly cycles. Therefore, we perform the cut and merge procedure by aligning the last data point before the gap with the one sharing the same time-of-day (but on a different date) after the gap. Such a choice possibly leads to losing multiple data points. Therefore, if the gap only involves a few measurements, we prefer linear interpolation. Otherwise, if the gap is bigger than a preset threshold value (e.g., 1 h), we resort to cut-and-merge as described above.

TABLE II
FEATURE SELECTION

Feature selection strategy	Selected features
Correlation matrix	O_3, RH_{in}, T_{in}, SR
Univariate selection	O_3, RH_{in}, T_{in}, SR, NO_2
Sequential forward selection	O_3, RH_{out}, RH_{in}, T_{in}, SR, NO_2
Subsequential backward elimination	O_3, T_{in}, SR, NO_2
Best features (which emerge from at least two strategies)	
O_3, RH_{in}, T_{in}, SR, NO_2	

In our extensive tests, we observed that a threshold of 1 h represents the best trade-off between preserving time series data and reducing the impact of interpolation on the ML training process.

3) Feature Selection and Extraction: Feature selection [81] is the process of extracting the most relevant features among those available in the training dataset. From the point of view of training, reducing the number of features to process improves the accuracy of the resulting model, limits its complexity, and mitigates the chance of overfitting [82]. More broadly, fewer input features also lead to a lower training time. Different techniques can be exploited for feature selection on the LCS time series, considering each measurement as an input feature (O_3, NO_2, PM, T_{in}, RH_{in}, T_{out}, RH_{out}, SR, and P) and the REF measurement as a target [83], [84]. In this study, multiple approaches have been considered, namely, Pearson correlation, univariate selection, and the wrapper methods of sequential forward selection and sequential backward elimination. The outcomes are then combined through voting and the best features are selected/extracted to be included in the pipelines. Table II summarizes the feature selection outcome.

B. Calibration Models' Setup

Among the ML models used in similar studies, we often find the classical LR and MLR [85], or other nonlinear models like SVR [86] and RFR [87], which commonly perform better than linear ones. Instead, among the NN-based approaches [88], the most common are MLP, CNN, and RNN, which enable the exploitation of temporal correlations by means of convolution operations and memory cells, respectively.

The classical ML models' pipeline is implemented using Scikit-learn[5], which provides built-in functions for the above models. These can be optimized to the need of a specific problem by means of tunable hyperparameters. The NN models are built via TensorFlow[6] by creating a suitable NN topology, e.g., dense layers (DLs) plus 2-D Convolution layers (for CNN) and LSTM layers (for RNN). The methods implemented in our work are fully analogous to those used in the prior art (e.g., MLR, RFR, and SRV [23], [44], MLP [45], [55], [56], and LSTM [70]).

Since our aim is to solve a regression task, all NN models require a single output neuron, while the number of input neurons depends on the number of training features. For the MLP, only DLs are used, while for the CNN and RNN models, their own specific layer is used first, ending up in a sequence of DLs [89]. The number of layers in the NN topology is limited, as the LCS calibration problem involves modest amounts of data, and having larger networks would increase the likelihood of overfitting [82]. The overfitting problem is handled also by implementing early stopping during the network training if the error [mean average error (MAE)] on the validation set starts increasing with the number of epochs. In the specific case of our implementation, if the MAE does not improve over 50 epochs, we trigger an early termination.

C. Training Approaches

1) Random Versus Sequential Sampling: Once the ML models are implemented, a common step to take is splitting the time series into training and test sets. A typical approach is to consider 80% of the dataset as the training set, and the remaining 20% as the test set. In the scope of this work, we examined different splitting ratios in order to evaluate a utility relationship between the amount of data used for training, and the goodness of the results obtained from the ML-based O_3LCS calibration. Further details on this aspect follow in Section V.

We remark that selecting training and test data sequentially or at random significantly affects the predictive performance of the resulting ML model. Typically, random sampling (RS) is chosen, so that the training and test sets have the same statistics [85]. In the case of a real application, where ML pipelines are supposed to be embedded into IoT systems (either in the edge or in the cloud), sequential sampling (SS), i.e., considering the first part of the dataset for the training and the remaining part for the test, is typically preferred. In fact, the typical operating conditions of time series sensors imply that a model should learn for an initial time period, and then calibrate the low-cost sensor for a predefined number of subsequent measurements. This may come at the cost of having statistically unbalanced training and test sets, as is especially true for harmful pollutant sensors, which are affected by strong seasonality trends.

The RS and SS outcomes in the case of an 80% splitting ratio are shown in Fig. 4.

From the boxplots in the insets, we observe that RS leads to balanced training and test sets, whereas in the SS case, the test set features a more concentrated range of data points around the median than the training set, which includes higher concentrations as well. This, in turn, affects the ML model's performance, because the learning phase occurs during a period of higher O_3 concentration, whereas the correction phase (calibration) is performed on a test set mainly characterized by lower O_3 concentrations.

2) Temporal Pivoting: Temporal pivoting is generally a better choice than providing the input value at the current time step as input to a prediction model. Temporal pivoting requires to arrange the input features of the ML model so that not only the data at the current time step t, but also the tuple of the surrounding time steps $(t - N, \ldots, t - 2, t - 1, t, t + 1, t + 2, \ldots, t + N)$ are taken into consideration in order to output calibrated measurements at time step t [90].

[5]https://scikit-learn.org/stable/
[6]https://www.tensorflow.org/

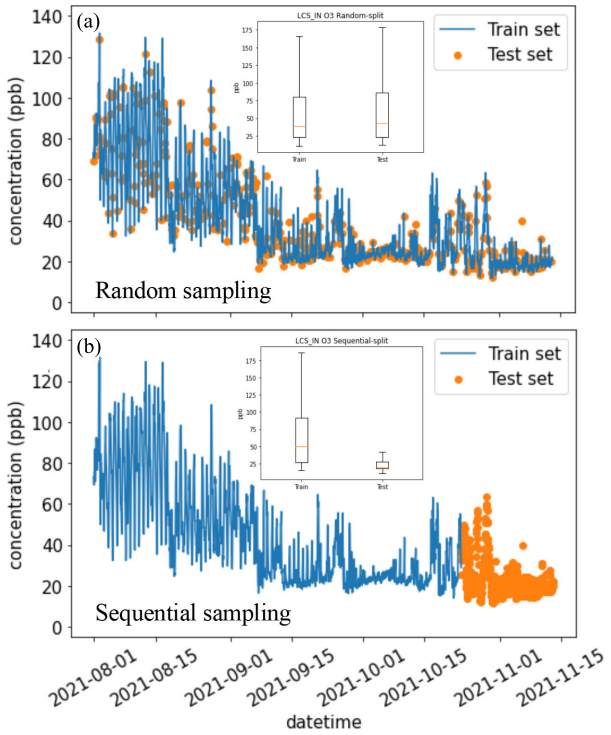


Fig. 4. 80% splitting between training and test set when different sampling conditions, i.e., (a) random and (b) sequential, are applied. The insets convey the dataset split distribution for each case.

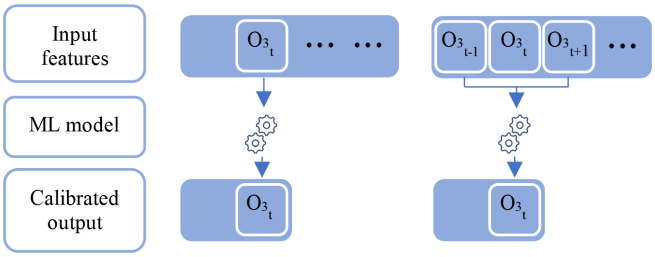


Fig. 5. Temporal pivoting representation, on the right, where three adjacent timestamp measurements are used to predict the current calibrated output.

TABLE III
MODELS TESTED IN THIS WORK AND CORRESPONDING HYPERPARAMETERS

Model	Hyperparameter	Value	Param
MLR	no hyperparameters		
SVR	Kernel	radial basis function	
	C	1e2	
	Gamma	1e-4	
RFR	n_estimators	40 ; 10 (RS or SS*)	
	max_features	0.5 ; 0.4*	
	min_sampling_split	2 ; 16*	
MLP**	# of dense layers (DS)	2	185 ;
	# of units in DS 1	16 ; 4*	77*
	# of units in DS 2	16 ; 4*	
	activation function	ReLU ; tanh*	
	learning rate	1e-2	
	optimizer	Adam	
RNN**	# of LSTM layers	1 ; 2*	10270 ;
	# of units in LSTM 1	32 ; 8*	501*
	# of units in LSTM 2	0 ; 2*	
	bidirectional layer 1	true ; false*	
	bidirectional layer 2	false; true*	
	# of dense layers (DS)	2	
	# of units in DS 1	4 ; 16*	
	# of units in DS 2	4 ; 2*	
	activation function	ReLU	
	learning rate	1e-2 ; 1e-3*	
	optimizer	Adam	
CNN**	# of 2D Conv layers	1	2325 ;
	# of filters Conv layer 1	10 ; 30*	1235*
	filter size	3 ; 4*	
	# of dense layers (DS)	2 ; 1*	
	# of units in DS 1	16 ; 2*	
	# of units in DS 2	16 ; 0*	
	activation function	ReLU ; tanh*	
	learning rate	1e-1 ; 1e-3*	
	optimizer	Adam ; SGD*	

*Different values come from the different training approaches used, namely random sampling (RS) or sequential sampling (SS). Please, see Section III.C.1 for more details.
**To evaluate the performance during the training, a validation set of 10% out of the data provided in the training set is considered. Each model is trained for 500 epochs, implementing early stopping if no improvement in the validation set occurs for more than 50 epochs. The data is arranged in batches of 128 samples.

An implementation scheme is shown in Fig. 5, where a sliding window on the O_3 input feature includes the adjacent time steps of the current measurement. All three values are passed to the ML model, which outputs the calibrated O_3 reading at the time step t. This is the way that we can account for the dependency that ties adjacent time steps. However, in doing so, we delay the models' output, as we have to wait for all the following $N/2$ measurements to be collected before predicting the calibrated output at time t. A way to overcome this issue is to consider only previous adjacent time steps ($t-1$, $t-2$, and so on), as is done in time series forecasting problems [71].

Since the analysis reported in this work is based on already acquired time series, we considered three different configurations for the purpose of evaluating the best temporal pivoting setting, namely, current time only (from now on [0,0]), one data point before and after the current time ([1,1]), two data points before and after the current time ([2,2]). In our extensive tests, we tried also combinations encompassing only past values, but such trials did not work properly.

D. D Models' Optimizaiton

The optimization of the hyperparameters' setting is one of the most critical steps to take in ML [88]. Depending on the method used, different ways to optimize the hyperparameters can be followed [91], [92]. A summary of the models benchmarked in this study together with their final hyperparameters is given in Table III. In the Appendix, we also list the search ranges used to optimize the hyperparameter values.

To optimize the hyperparameters of each model, we resort to a fivefold cross-validation procedure [93]. Moreover, to fine-tune the hyperparameters based on their starting values, the halving grid search and genetic algorithms (Gas) are broadly exploited in literature specifically to identify the best-suited NN architecture for a specific task [94].

It is worth mentioning that NNs are often harder to train than simpler ML algorithms like MLR and SVR. However,

TABLE IV
NSGA-II SETTINGS

Parameter	Value
Population size	64
Offspring population size	32
Number of generations	500
Crossover operation	Single point
Crossover probability	1.0
Mutation operation	Integer polynomial
Mutation probability	1/# parameters

they outperform the other approaches in the case of long time series. To relieve the training effort, NN architectures have to be as small as possible in terms of parameters, but should still yield good regression scores (R^2) for the calibration task. Smaller architectures are also advantageous in terms of portability. Especially in an edge computing scheme, IoT devices are often limited in terms of RAM and computing power, thus preventing the training of large NN architectures. The problem then becomes a multiobjective optimization problem that has to minimize the architecture size (in terms of parameters) and maximize the regression metric (R^2 score).

Regarding Gas, we choose the nondominated sorting algorithm (NSGA-II), which is widely used in recent works. The NGSA-II is implemented in the jMetalPy[7] framework, which enables the users to design their own individual genome and fitness functions [95]. Being a heuristic optimization technique with stochastic components, Gas such as NSGA-II may generate the same individuals over and over by crossover and mutation operations. Evaluating the same NN architecture multiple times can then slow down the convergence of such an algorithm. Hence, we keep track of individuals who have been already evaluated, along with their fitness. If a newly generated individual is present in the set, then the same fitness function is assigned to it, without retraining the model [50].

The NSGA-II setup conditions used to optimize the NN models considered in this work are reported in Table IV.

E. Models' Performance Evaluation

The ML models are evaluated over the test set through different loss metrics. In regression tasks, the coefficient of determination R^2, the MAE, the mean squared error (MSE), the root mean squared error (RMSE), and the mean absolute percentage error (MAPE) are the most common figures to compare the model output (LCS calibrated measurements) against the model input (REF measurements from the air quality monitoring station) [32].

V. RESULTS AND DISCUSSION

A. Calibration Models Benchmark

This section details the calibration performance of the different methods. We remark that the measurement time series have been preprocessed as described in Section IV-A, resulting in a 1-h averaging window and in using O_3, RH_{in}, T_{in}, SR, and NO_2 as input features. For training, both RS and SS are included in the comparison, with a splitting ratio of 80% for the training and 20% for the test set. We chose temporal pivoting [1,1] because it basically yields the same results as [2,2] or deeper pivoting, but is computationally lighter.

[7]https://github.com/jMetal/jMetalPy

TABLE V
PERFORMANCE OF THE IMPLEMENTED ML MODELS

ML model	MAE (init.)	RMSE (init.)	R^2 (init.)	MAE (calib)	RMSE (calib)	R^2 (calib)
Random Sampling						
MLR				3.66 ppb	5.08 ppb	0.86
SVR	25.21	28.82	-3.46	2.88 ppb	4.35 ppb	0.9
RFR	ppb	ppb		2.81 ppb	4.26 ppb	0.9
Sequential Sampling						
MLR				3.22 ppb	4.24 ppb	0.68
SVR	17.46	18.26	-4.98	2.29 ppb	3.52 ppb	0.78
RFR	ppb	ppb		2.97 ppb	3.7 ppb	0.75

TABLE VI
PERFORMANCE OF THE IMPLEMENTED NN

NN	MAE (init)	RMSE (init)	R^2 (init)	MAE (calib)	RMSE (calib)	R^2 (calib)
Random Sampling						
MLP				3.36 ppb	5.01 ppb	0.87
LSTM	25.21	28.82	-3.46	3.16 ppb	4.67 ppb	0.88
CNN	ppb	ppb		3.24 ppb	4.75 ppb	0.88
Sequential Sampling						
MLP				2.53 ppb	3.32 ppb	0.8
LSTM	17.46	18.26	-4.98	2.11 ppb	3.19 ppb	0.82
CNN	ppb	ppb		2.62 ppb	3.43 ppb	0.79

The obtained results are reported in Table V for the ML models (MLR, SVR, and,RFR) and in Table VI for the NN (MLP, LSTM, and CNN). All methods are evaluated through the MAE, RMSE, and R^2 scores. The initial MAE, RMSE, and R^2 obtained on the raw data (i.e., noncalibrated test set versus reference test set) are also reported in the tables[8].

1) ML Model Performance: As a first consideration, we observe that the calibrations obtained with the RS-based training are overall better than those obtained with the SS. Besides the starting correlation R^2 between O_3LCS and O_3REF being higher in the RS case, RS performs better because, in the SS case, the models are trained mainly during a period of high O_3 concentrations, and then tested when a notable O_3 drop occurs (winter season in the Northern Hemisphere). SVR and RFR achieve the same calibration accuracy with the RS splitting, while SVR shows the best performance with the SS splitting.

Figs. 6 and 7 show the calibration outcome in the case of SVR with the RS splitting and the SS splitting, respectively.

The initial dataset (noncalibrated data) is reported in Figs. 6(a) and 7(a), where we recall that RS takes test data from the whole dataset, whereas SS uses only the last 20% of the dataset (see Fig. 4). Figs. 6(b) and 7(b) report the

[8]It is worth mentioning that, with longer time series than those analyzed in this work, it would be possible to evaluate how the ML and NN models' performance degrades in the long-term (more than two years), which may lead to different outcomes with respect to the short-term ones.

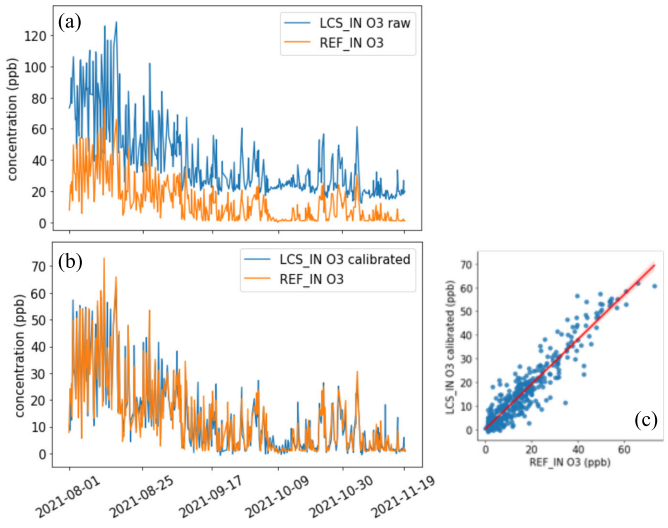


Fig. 6. SVR method implemented with RS splitting approach: (a) starting test set carrying non-calibrated O_3LCS and O_3REF; (b) calibrated O_3LCS and O_3REF; and (c) LR between O_3REF and calibrated O_3LCS.

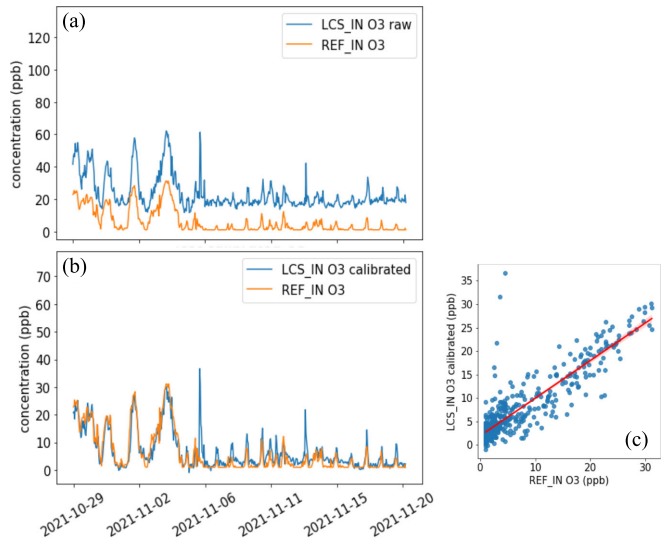


Fig. 7. SVR method implemented with SS splitting approach: (a) starting test set carrying non-calibrated O_3LCS and O_3REF; (b) calibrated O_3LCS and O_3REF; and (c) LR between O_3REF and calibrated O_3LCS.

calibrated O_3LCS time series, in blue solid line, and the O_3REF time series, in orange solid line, as reference.

Figs. 6(c) and 7(c) show the LR between O_3REF and calibrated O_3LCS. While in Fig. 6(c) the alignment of the data along the bisector line indicates an overall successful calibration, in Fig. 7(c) only the data representing higher O_3 concentrations lie on the line, whereas the LR degrades for low/intermediate concentrations. Again, the main reason is that the model is trained during a period of high O_3 concentration and then tested after O_3 drops (as is typical for the winter season of the Northern hemisphere). This point will be further discussed in Section V-B.

2) NNs Performance: The overall observation about the worsening of the calibration by passing from the RS to the SS splitting holds also for the NNs. In this case, the performances of the three networks are always aligned, with the LSTM

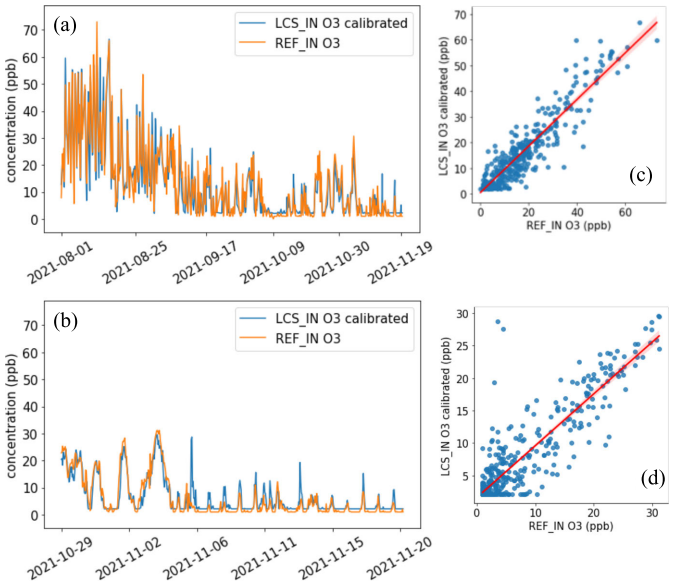


Fig. 8. LSTM calibration implemented with: (a) and (b) RS splitting approach and (c) and (d) SS splitting approach. The plots carry the calibrated O_3LCS time series in blue and the O_3REF time series as reference in orange. The LR between O_3REF and calibrated O_3LCS are shown for the (b) RS splitting and (d) SS splitting, respectively.

architecture in the lead. However, with an SS splitting, all three MLP, LSTM, and CNN output a better calibration than the SVR (best among ML models).

Fig. 8 reports the calibrated O_3LCS time series in the blue solid line and the O_3REF time series in the orange solid line for the LSTM NN model in both the cases of RS (a) and SS (b) splitting. The plots on the rightmost part of the figure represent the LR between O_3REF and calibrated O_3LCS.

In this case, the scatter points are aligned to the bisector line for the RS splitting as well as for the SS splitting. Even if Fig. 8(c) shows a smaller dispersion then Fig. 8(d), no offsets are present also in the SS case and the calibration is overall better than the one reported in Fig. 7(b) and (c).

It is interesting to notice that some sharp overshoot starts to appear in the final part of the time series, for instance looking at Figs. 7(b) and 8(b). This might have not physical but rather an electronic cause. In any event, such localized effect does not have a remarkable contribution to R^2 worsening. For a matter of comparing the different reactions of the methods to the small features along the REF time series, in Fig. 9, we show four plots focusing on the period October 29, 2021 to November 20, 2021 for the performance of SVR–RS, SVR–SS, LSTM–RS, and LSTM–SS, respectively.

B. Training Minimization and Convergence Analysis

The focus of this section is to assess the effect of different SS splitting ratios between training and test sets on the calibration model's performance. To do this, the percentage of the dataset employed for training is varied (simulating a variation of the length of the calibration period), in order to mimic a real calibration procedure. In fact, for practical purposes, the LCSs should realistically be installed close to a reference station just for a few days/weeks and, once the

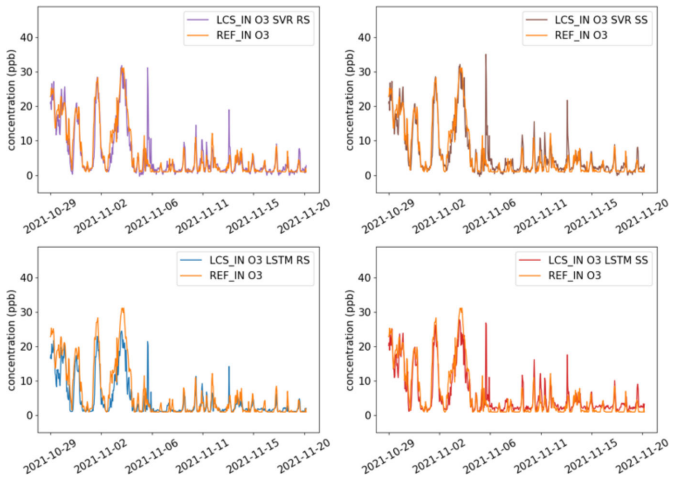


Fig. 9. Outcome of the four different methods, namely, SVR–RS, SVR–SS, LSTM–RS, and LSTM–SS, on the period October 29, 2021 to November 20, 2021.

model training is accomplished, the calibration should last as much as possible within a certain quality range.

In this case, the tests are carried out with the LSTM architecture, which has been found to yield the best performance among the models analyzed in Section V-A. The evaluation of the effect of the different SS splitting ratios is carried out through the correlation of O_3LCS and O_3REF over time. More precisely, in the test set the R^2 parameter is averaged on a weekly basis, so as to indicate any drops in the correlation on a timescale. For the purpose of this specific test, we colocated the LCSs with the reference station for two additional timeslots, one from January 25 to February 4, 2022 and another from February 23 to March 25, 2022. By doing so, we could verify the reliability of the calibration also in periods far from the training.

Fig. 10 reports the results of decreasing the training-set duration from the 80% of the overall dataset to the 40%, 20%, and 10%. The upper panel shows the noncalibrated O_3LCS and the O_3REF time series, where the two additional field campaign periods have been appended following the merging procedure described in Section IV-A. The yellow, green, red, and blue vertical lines represent the SS splitting ratios of 80/20, 60/40, 20/80, and 10/90 between training and test sets, respectively. The lower panels show the weekly R^2 trend for all the above combinations.

As a general comment, we observe that the LSTM performance clearly degrades only in the 10/90 case, whereas the trends observed in the other cases are practically similar. Also, all the trials suffer from a dramatic correlation drop in the first half of October 2021. By focusing on the gray vertical bar, we observe that concurrently with the first serious drop of O_3 concentration in the time series, only the 80/60 and the 60/40 maintain a reasonably high R^2 score, as the models are still being trained at this time. However, this is not sufficient to counteract what happens within the pink vertical bar, where R^2 goes almost to 0 for all the splitting ratios. The reason for this behavior is that the O_3 concentration also basically drops to 0, and thus the sensor's and reference

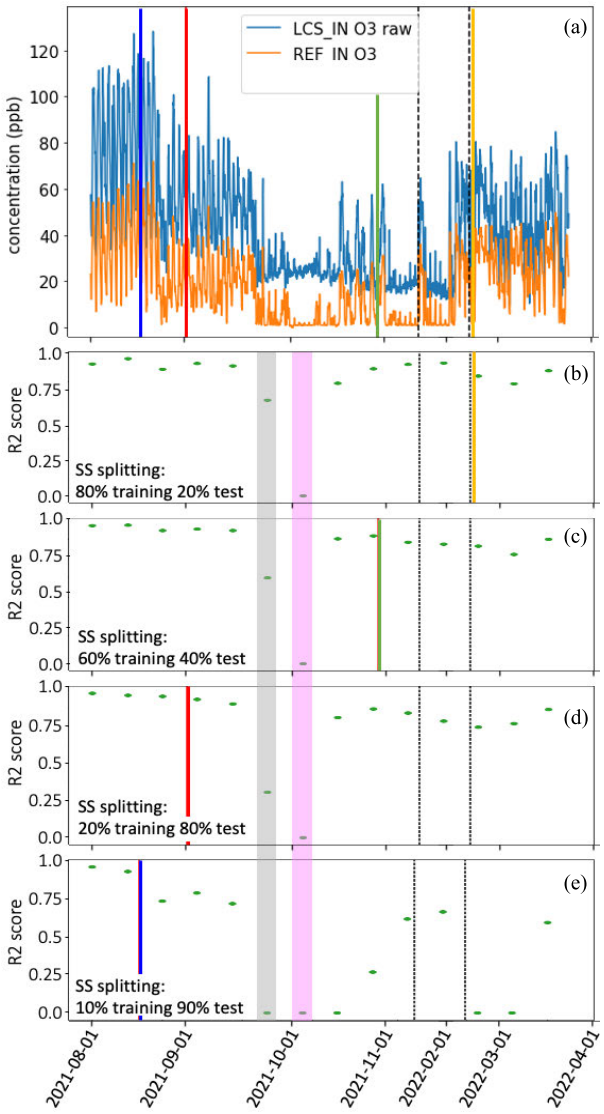


Fig. 10. (a) Noncalibrated O_3LCS and O_3REF time series extended to January 25–February 4, 2022 and February 25–March 25, 2022. (b)–(e) Weekly averaged R^2 score trends associated with different SS splitting ratios, namely, 80/20, 60/40, 20/80, and 10/90.

instrument's noise floor become predominant. In this particular time period, the calibration model analyzes two signals with a completely different physical origin, which are most likely not correlated at all. In fact, the ThermoSCIENTIFIC 49i O_3 analyzer exploited as a reference instrument is based on dual cell photometer mainly affected by shot noise and afterpulsing [96], [97] at full-scale, while the Alphasense OX-A431 is mainly affected by the electronic noise brought by the host-board Analogue Front End—A4 [98]. For completeness, we report that the temperature ranged between 11.26 °C and 32.6 °C whereas the RH ranged between 10.8% and 79.8% during the field campaign.

From this analysis, we can conclude that, in the case of the LSTM NN with an SS splitting approach, the splitting ratio between training and test can be lowered down to 20/80 with no serious consequences on the calibration expiration over time.

On the other hand, it must be noticed that also the classical splitting ratio 80/20 is not immune to a dramatic R^2 degradation when strong seasonal effects set in. A more in-depth analysis on the dependence on the performance on the actual month in which the calibration is carried out will be explored in a future work, when a dataset spanning over multiple years will be available.

The training minimization analysis has been carried out with the LSTM method, as it yielded the best performance among all the methods surveyed. Such an outcome is supported by the fact that nonlinear methods can better capture complex patterns and relationships from time series. Indeed, time series often exhibit nonlinear temporal dependencies, which can be modeled effectively by NNs, for instance. In particular, deep learning techniques, such as RNNs and LSTM networks, are specifically designed for sequential data. Indeed, they can capture intricate dependencies over time, like seasonality, trends, time-varying drifts, and irregular patterns, which make them effective for forecasting and anomaly detection tasks. They can also work with time series of varying lengths, which is common in practice when irregularly sampled or missing data points are coming from deployed sensors. Lastly, LSTMs can be regularized effectively to prevent overfitting, which is important in time series analysis to ensure that the model generalizes well to unseen data.

TABLE VII
HYPERPARAMETERS SEARCH RANGES FOR MODELS' OPTIMIZATION

Model	Hyperparameter	Range
MLR	no hyperparameters	
SVR	Kernel	[radial basis function]
	C	[1e1, 1e2, 1e3, 1e4]
	Gamma	[1e-1, 1e-2, 1e-3, 1e-4]
RFR	n_estimators	[10, 20, 30, …, 100]
	max_features	[0.1, 0.2, 0.3, …, 1]
	min_sampling_split	[2, 4, 6, …, 20]
MLP	# of dense layers (DS)	[1, 2]
	# of units per DS	[2, 4, 8, 16, 32]
	activation function	[ReLU, tanh, Linear]
	learning rate	[1e-1, 1e-2, 1e-3]
	optimizer	[Adam, SGD]
RNN	# of LSTM layers	[1, 2]
	# of units per LSTM layer	[2, 4, 8, 16, 32]
	bidirectional layer flag	[1, 2]
	# of dense layers (DS)	[1, 2]
	# of units per DS	[2, 4, 8, 16, 32]
	activation function	[ReLU, tanh, Linear]
	learning rate	[1e-1, 1e-2, 1e-3]
	optimizer	[Adam, SGD]
CNN	# of 2D Conv layers	[1, 2]
	# of filters per Conv layer	[10, 20, 30]
	filter size	[2, 3, 4]
	# of dense layers (DS)	[1, 2]
	# of units per DS	[2, 4, 8, 16, 32]
	activation function	[ReLU, tanh, Linear]
	learning rate	[1e-1, 1e-2, 1e-3]
	optimizer	[Adam, SGD]

VI. CONCLUSION AND DISCUSSION

This work contributes to pervasive environmental monitoring by means of autonomous LCSs. Nowadays, the scarce density of certified monitoring stations affects the validity and pervasiveness of air quality measurements. The time series obtained from the few stations may only be sufficient to get to a general overview of the pollution level, but not to sustain counteracting strategies and policies aimed at improving air quality. This leads to a need for more affordable solutions for distributed and continuous monitoring.

One of the main issues limiting the takeover of LCSs is, however, the low reliability of the sensors themselves and the fast expiration of their calibration. Many works have already demonstrated the potential of ML and NNs to cope with such issues. In our study, we focused on a particular calibration method that is carried out directly in the field. This aspect brings the twofold benefit of: 1) speeding up the calibration procedure and 2) taking into account many environmental features that are usually neglected during the common practices. This last feature becomes crucial when we consider moving the LCSs far apart from the reference site, once the calibration is accomplished. It is crucial to clarify that any calibration method needs a sufficient amount of data collected from co-located certified stations in order to be valid, and no less than needed to observe/address seasonal variations. In our study, we aimed to respect this condition while contributing to the mainstream of making LCSs more accurate, but at the same time more sustainable for a pervasive adoption, that is actually hindered by too long-lasting calibration procedures. In Section V-B, we performed a full analysis of how to minimize the training time of an LSTM network in order to achieve a long-lasting calibration with the minimum training effort. We showed that when strong seasonal effects set in, no splitting ratio is immune to calibration degradation, not even the classical 80/20. On the other hand, our analysis suggested that almost all splitting ratios surveyed (except 10/90) recover reasonable correlation values after the low O_3 concentration period occurred at the beginning of October. We have quantified the calibration degradation through the weekly average of R^2. Such a metric is more explanatory than reporting a single R^2 value, independently of the length and the splitting of the dataset. Indeed, by looking at the weekly evolution of the correlation between the corrected LCS and REF time series, we can assess whether the calibration has gone below a threshold value, thus becoming unacceptable. Otherwise, by looking only at the cumulative R^2 value, it could likely be that such patterns are averaged out. We concluded that a 20% training–80% test in an SS splitting scheme can give satisfactory results in terms of calibration durability. Practically, we showed that also small training/test splitting ratios can recover acceptable R^2 performances, while even the biggest splitting ratios are not immune to R^2 disruption when dramatic seasonal effects onset.

Our analysis has been carried out in line with the best practices described in many foundational works (see Section II). This allowed us to identify the LSTM as the best model for calibrating our sensors together with its optimal hyperparameters. The parametrization approach followed for evaluating the performance of different training versus test splitting ratios is also compatible with other relevant studies in the field.

Even if we considered only the case of an O_3LCS, the results obtained here can be extended to many other sensing devices and lay the foundation for an effective and on-the-fly

implementation of ML-based calibration methods for LCSs. This opens to further developments and applications, such as LCSs' aging prediction, drift compensation, and calibration model transfer among similar sensors.

APPENDIX

In Table VII, we report the search ranges wherein the hyperparameters of the models tested in this work have been swept.

ACKNOWLEDGMENT

This work has been carried out in the framework of the FIRST project (Artificial Intelligence for Wireless Sensor Networks). The authors thank the partners which contributed to the FIRST project, namely, APPA Bolzano—Agenzia Provinciale per l'ambiente e la tutela del clima, Bolzano, Italy, and specifically David Simoncello and Luca Verdi, the Digital tech transfer unit of the NOI Techpark, Bolzano, (Partick Ohnewein and Stefano Seppi), the Center for Materials and Microsystems of Fondazione Bruno Kessler, Trento, Italy, (Andrea Gaiardo), and the Computer Science Faculty of the University of Bolzano, Bolzano, (Antonio Liotta), and the Center for Sensing Solutions, Bolzano–IT (Roberto Monsorno). The authors thank the Department of Innovation, Research and University of Autonomous Province of Bolzano for covering the Open Access publication costs.

REFERENCES

[1] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, "Environmental and health impacts of air pollution: A review," *Frontiers Public Health*, vol. 8, p. 14, Feb. 2020, doi: 10.3389/fpubh.2020.00014.

[2] M. Carminati, O. Kanoun, S. L. Ullo, and S. Marcuccio, "Prospects of distributed wireless sensor networks for urban environmental monitoring," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 6, pp. 44–52, Jun. 2019, doi: 10.1109/MAES.2019.2916294.

[3] S. Vardoulakis, N. Gonzalezflesca, B. Fisher, and K. Pericleous, "Spatial variability of air pollution in the vicinity of a permanent monitoring station in central Paris," *Atmos. Environ.*, vol. 39, no. 15, pp. 2725–2736, May 2005, doi: 10.1016/j.atmosenv.2004.05.067.

[4] E. Lagerspetz et al., "MegaSense: Feasibility of low-cost sensors for pollution hot-spot detection," in *Proc. IEEE 17th Int. Conf. Ind. Informat. (INDIN)*, vol. 1, Jul. 2019, pp. 1083–1090, doi: 10.1109/INDIN41052.2019.8971963.

[5] A. Moore, M. Figliozzi, and C. M. Monsere, "Air quality at bus stops," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2270, no. 1, pp. 76–86, Jan. 2012, doi: 10.3141/2270-10.

[6] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards Internet of Things," in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2017, pp. 1–6, doi: 10.1109/CCWC.2017.7868374.

[7] S. Croce and S. Tondini, "Urban microclimate monitoring and modeling through an open-source distributed network of wireless low-cost sensors and numerical simulations," *Eng. Proc.*, vol. 5, no. 18, 2020, doi: 10.3390/ecsa-7-08270.

[8] Sensor. *Community Web Portal*. Accessed: Dec. 13, 2023. [Online]. Available: https://sensor.community/en/

[9] S. Croce and S. Tondini, "Fixed and mobile low-cost sensing approaches for microclimate monitoring in urban areas: A preliminary study in the city of Bolzano (Italy)," *Smart Cities*, vol. 5, no. 1, pp. 54–70, Jan. 2022, doi: 10.3390/smartcities5010004.

[10] S. Zhu, K. Ota, and M. Dong, "Energy-efficient artificial intelligence of things with intelligent edge," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7525–7532, May 2022, doi: 10.1109/JIOT.2022.3143722.

[11] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: A consensus approach for massive IoT networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, May 2020, doi: 10.1109/JIOT.2020.2964162.

[12] J. Lu et al., "A sustainable solution for IoT semantic interoperability: Dataspaces model via distributed approaches," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7228–7242, May 2022, doi: 10.1109/JIOT.2021.3097068.

[13] A. Jarwan, A. Sabbah, and M. Ibnkahla, "Information-oriented traffic management for energy-efficient and loss-resilient IoT systems," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7388–7403, May 2022, doi: 10.1109/JIOT.2021.3132925.

[14] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "R-ARM: Retransmission-assisted resource management in LoRaWAN for the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7347–7361, May 2022, doi: 10.1109/JIOT.2021.3111167.

[15] T. Huang et al., "Adaptive processor frequency adjustment for mobile-edge computing with intermittent energy supply," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7446–7462, May 2022, doi: 10.1109/JIOT.2021.3119866.

[16] M. Silva, A. Riker, J. Torrado, J. Santos, and M. Curado, "Extending energy neutral operation in Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7510–7524, May 2022, doi: 10.1109/JIOT.2021.3133615.

[17] Y. Su, X. Lu, Y. Zhao, L. Huang, and X. Du, "Cooperative communications with relay selection based on deep reinforcement learning in wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 20, pp. 9561–9569, Oct. 2019, doi: 10.1109/JSEN.2019.2925719.

[18] H. Chen, X. Li, and F. Zhao, "A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2763–2774, Apr. 2016, doi: 10.1109/JSEN.2016.2517084.

[19] P. Nowack et al., "Towards low-cost and high-performance air pollution measurements using machine learning calibration techniques," *Atmos. Meas. Tech.*, vol. 14, pp. 5637–5655, Dec. 2020.

[20] T. Veiga et al., "From a low-cost air quality sensor network to decision support services: Steps towards data calibration and service development," *Sensors*, vol. 2, n. 9, p. 3190, May 2021, doi: 10.3390/s21093190.

[21] S. Rajendran et al., "ElectroSense: Open and big spectrum data," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 210–217, Jan. 2018, doi: 10.1109/MCOM.2017.1700200.

[22] E. Esposito, S. De Vito, M. Salvato, V. Bright, R. L. Jones, and O. Popoola, "Dynamic neural network architectures for on field stochastic calibration of indicative low cost air quality sensing systems," *Sens. Actuators B, Chem.*, vol. 231, pp. 701–713, Aug. 2016, doi: 10.1016/j.snb.2016.03.038.

[23] A. Bigi, M. Mueller, S. K. Grange, G. Ghermandi, and C. Hueglin, "Performance of NO, NO_2 low cost sensors and three calibration approaches within a real world application," *Atmos. Meas. Techn.*, vol. 11, no. 6, pp. 3717–3735, Jun. 2018, doi: 10.5194/amt-11-3717-2018.

[24] E. S. Cross et al., "Use of electrochemical sensors for measurement of air pollution: Correcting interference response and validating measurements," *Atmos. Meas. Techn.*, vol. 10, no. 9, pp. 3575–3588, Sep. 2017, doi: 10.5194/amt-10-3575-2017.

[25] N. Masson, R. Piedrahita, and M. Hannigan, "Quantification method for electrolytic sensors in long-term monitoring of ambient air quality," *Sensors*, vol. 15, no. 10, pp. 27283–27302, Oct. 2015, doi: 10.3390/s151027283.

[26] M. Bart et al., "High density ozone monitoring using gas sensitive semi-conductor sensors in the lower Fraser Valley, British Columbia," *Environ. Sci. Technol.*, vol. 48, no. 7, pp. 3970–3977, Apr. 2014, doi: 10.1021/es404610t.

[27] M. I. Mead et al., "The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks," *Atmos. Environ.*, vol. 70, pp. 186–203, May 2013, doi: 10.1016/j.atmosenv.2012.11.060.

[28] N. Masson, R. Piedrahita, and M. Hannigan, "Approach for quantification of metal oxide type semiconductor gas sensors used for ambient air quality monitoring," *Sens. Actuators B, Chem.*, vol. 208, pp. 339–345, Mar. 2015, doi: 10.1016/j.snb.2014.11.032.

[29] W. Jiao et al., "Community air sensor network (CAIRSENSE) project: Evaluation of low-cost sensor performance in a suburban environment in the southeastern United States," *Atmos. Meas. Techn.*, vol. 9, no. 11, pp. 5281–5292, Nov. 2016, doi: 10.5194/amt-9-5281-2016.

[30] R. Papaconstantinou et al., "Field evaluation of low-cost electrochemical air quality gas sensors under extreme temperature and relative humidity conditions," *Atmos. Meas. Techn.*, vol. 16, no. 12, pp. 3313–3329, Jun. 2023, doi: 10.5194/amt-16-3313-2023.

[31] *Evaluation of Measurement Data—Guide to the Expression of Uncertainty in Measurement (GUM:1995)*, document JCGM 100:2008, (GUM 1995 With Minor Corrections), Sep. 2008.

[32] F. Concas et al., "Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis," *ACM Trans. Sensor Netw.*, vol. 17, no. 2, pp. 1–44, May 2021, doi: 10.1145/3446005.

[33] A. R. Shamshiri, M. B. Ghaznavi-Ghoushchi, and A. R. Kariman, "ML-based aging monitoring and lifetime prediction of IoT devices with cost-effective embedded tags for edge and cloud operability," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7433–7445, May 2022, doi: 10.1109/JIOT.2021.3116065.

[34] L. Jia, Z. Zhou, F. Xu, and H. Jin, "Cost-efficient continuous edge learning for artificial intelligence of things," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7325–7337, May 2022, doi: 10.1109/JIOT.2021.3104089.

[35] Z. Wang, C. Hu, D. Zheng, and X. Chen, "Ultralow-power sensing framework for Internet of Things: A smart gas meter as a case," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7533–7544, May 2022, doi: 10.1109/JIOT.2021.3110886.

[36] K. Ahmed and M. Gregory, "Integrating wireless sensor networks with cloud computing," in *Proc. 7th Int. Conf. Mobile Ad-hoc Sensor Netw.*, Dec. 2011, pp. 364–366, doi: 10.1109/MSN.2011.86.

[37] Y. Hashmy, Z. U. Khan, F. Ilyas, R. Hafiz, U. Younis, and T. Tauqeer, "Modular air quality calibration and forecasting method for low-cost sensor nodes," *IEEE Sensors J.*, vol. 23, no. 4, pp. 4193–4203, Feb. 2023, doi: 10.1109/JSEN.2023.3233982.

[38] J. Hofman, M. Nikolaou, S. P. Shantharam, C. Stroobants, S. Weijs, and V. P. La Manna, "Distant calibration of low-cost PM and NO_2 sensors; evidence from multiple sensor testbeds," *Atmos. Pollut. Res.*, vol. 13, no. 1, Jan. 2022, Art. no. 101246, doi: 10.1016/j.apr.2021.101246.

[39] G. Miskell et al., "Reliable data from low cost ozone sensors in a hierarchical network," *Atmos. Environ.*, vol. 214, Oct. 2019, Art. no. 116870, doi: 10.1016/j.atmosenv.2019.116870.

[40] P. Nowack, L. Konstantinovskiy, H. Gardiner, and J. Cant, "Machine learning calibration of low-cost NO_2 and PM_{10} sensors: Non-linear algorithms and their impact on site transferability," *Atmos. Meas. Techn.*, vol. 14, no. 8, pp. 5637–5655, Aug. 2021, doi: 10.5194/amt-14-5637-2021.

[41] P. de Souza et al., "Calibrating networks of low-cost air quality sensors," *Atmos. Meas. Techn.*, vol. 15, no. 21, pp. 6309–6328, Nov. 2022, doi: 10.5194/amt-15-6309-2022.

[42] W. W. Hsieh, *Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels*. Cambridge, U.K.: Cambridge Univ. Press, 2009, doi: 10.1017/CBO9780511627217.

[43] M. A. Zaidan et al., "Intelligent calibration and virtual sensing for integrated low-cost air quality sensors," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13638–13652, Nov. 2020, doi: 10.1109/JSEN.2020.3010316.

[44] N. Zimmerman et al., "A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring," *Atmos. Meas. Techn.*, vol. 11, no. 1, pp. 291–313, Jan. 2018, doi: 10.5194/amt-11-291-2018.

[45] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, "Field calibration of a cluster of low-cost available sensors for air quality monitoring. Part A: Ozone and nitrogen dioxide," *Sens. Actuators B, Chem.*, vol. 215, pp. 249–257, Aug. 2015, doi: 10.1016/j.snb.2015.03.031.

[46] H. Yu et al., "A deep calibration method for low-cost air monitoring sensors with multilevel sequence modeling," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 7167–7179, Sep. 2020, doi: 10.1109/TIM.2020.2978596.

[47] E. M. Considine, C. E. Reid, M. R. Ogletree, and T. Dye, "Improving accuracy of air pollution exposure measurements: Statistical correction of a municipal low-cost airborne particulate matter sensor network," *Environ. Pollut.*, vol. 268, Jan. 2021, Art. no. 115833, doi: 10.1016/j.envpol.2020.115833.

[48] S. S. Patra, R. Ramsisaria, R. Du, T. Wu, and B. E. Boor, "A machine learning field calibration method for improving the performance of low-cost particle sensors," *Building Environ.*, vol. 190, Mar. 2021, Art. no. 107457, doi: 10.1016/j.buildenv.2020.107457.

[49] M. Antonini, A. Gaiardo, and M. Vecchio, "MetaNChemo: A meta-heuristic neural-based framework for chemometric analysis," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106712, doi: 10.1016/j.asoc.2020.106712.

[50] S. De Vito, G. Di Francia, E. Esposito, S. Ferlito, F. Formisano, and E. Massera, "Adaptive machine learning strategies for network calibration of IoT smart air quality monitoring devices," *Pattern Recognit. Lett.*, vol. 136, pp. 264–271, Aug. 2020, doi: 10.1016/j.patrec.2020.04.032.

[51] H. Lee, J. Kang, S. Kim, Y. Im, S. Yoo, and D. Lee, "Long-term evaluation and calibration of low-cost particulate matter (PM) sensor," *Sensors*, vol. 20, no. 13, p. 3617, Jun. 2020, doi: 10.3390/s20133617.

[52] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sens. Actuators B, Chem.*, vol. 129, no. 2, pp. 750–757, Feb. 2008, doi: 10.1016/j.snb.2007.09.060.

[53] M. Kamionka, P. Breuil, and C. Pijolat, "Calibration of a multivariate gas sensing device for atmospheric pollution measurement," *Sens. Actuators B, Chem.*, vol. 118, nos. 1–2, pp. 323–327, Oct. 2006, doi: 10.1016/j.snb.2006.04.058.

[54] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sens. Actuators B, Chem.*, vol. 110, no. 2, pp. 304–311, Oct. 2005, doi: 10.1016/j.snb.2005.02.008.

[55] C. Borrego et al., "Assessment of air quality microsensors versus reference methods: The EuNetAir joint exercise," *Atmos. Environ.*, vol. 147, pp. 246–263, Dec. 2016, doi: 10.1016/j.atmosenv.2016.09.050.

[56] B. Maag, Z. Zhou, and L. Thiele, "A survey on sensor calibration in air pollution monitoring deployments," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4857–4870, Dec. 2018, doi: 10.1109/JIOT.2018.2853660.

[57] J. M. Cordero, R. Borge, and A. Narros, "Using statistical methods to carry out in field calibrations of low cost air quality sensors," *Sens. Actuators B, Chem.*, vol. 267, pp. 245–254, Aug. 2018, doi: 10.1016/j.snb.2018.04.021.

[58] A. Patton et al., "Non-linear probabilistic calibration of low-cost environmental air pollution sensor networks for neighborhood level spatiotemporal exposure assessment," *J. Exposure Sci. Environ. Epidemiol.*, vol. 32, no. 6, pp. 908–916, Nov. 2022, doi: 10.1038/s41370-022-00493-y.

[59] K. Aula, E. Lagerspetz, P. Nurmi, and S. Tarkoma, "Evaluation of low-cost air quality sensor calibration models," *ACM Trans. Sensor Netw.*, vol. 18, no. 4, pp. 1–32, Nov. 2022, doi: 10.1145/3512889.

[60] S. Schmitz et al., "Unravelling a black box: An open-source methodology for the field calibration of small air quality sensors," *Atmos. Meas. Techn.*, vol. 14, no. 11, pp. 7221–7241, Nov. 2021, doi: 10.5194/amt-14-7221-2021.

[61] S. De Vito et al., "Crowdsensing IoT architecture for pervasive air quality and exposome monitoring: Design, development, calibration, and long-term validation," *Sensors*, vol. 21, no. 15, p. 5219, Jul. 2021, doi: 10.3390/s21155219.

[62] P. Han et al., "Calibrations of low-cost air pollution monitoring sensors for CO, NO_2, O_3, and SO_2," *Sensors*, vol. 21, no. 1, p. 256, Jan. 2021, doi: 10.3390/s21010256.

[63] C. Lin, J. Gillespie, M. D. Schuder, W. Duberstein, I. J. Beverland, and M. R. Heal, "Evaluation and calibration of aeroqual series 500 portable gas sensors for accurate measurement of ambient ozone and nitrogen dioxide," *Atmos. Environ.*, vol. 100, pp. 111–116, Jan. 2015, doi: 10.1016/j.atmosenv.2014.11.002.

[64] B. Maag, O. Saukh, D. Hasenfratz, and L. Thiele, "Pre-deployment testing, augmentation and calibration of cross-sensitive sensors," in *Proc. Int. Conf. Embedded Wireless Syst. Netw.*, Feb. 2016, pp. 169–180.

[65] S. Ali, F. Alam, K. M. Arif, and J. Potgieter, "Low-cost CO sensor calibration using one dimensional convolutional neural network," *Sensors*, vol. 23, no. 2, p. 854, Jan. 2023, doi: 10.3390/s23020854.

[66] T. Araújo, L. Silva, A. Aguiar, and A. Moreira, "Calibration assessment of low-cost carbon dioxide sensors using the extremely randomized trees algorithm," *Sensors*, vol. 23, no. 13, p. 6153, Jul. 2023, doi: 10.3390/s23136153.

[67] N. M. Kebonye, "Exploring the novel support points-based split method on a soil dataset," *Measurement*, vol. 186, Dec. 2021, Art. no. 110131, doi: 10.1016/j.measurement.2021.110131.

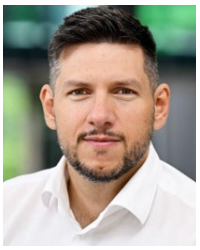
[68] S. Szeghalmy and A. Fazekas, "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning," *Sensors*, vol. 23, no. 4, p. 2333, Feb. 2023, doi: 10.3390/s23042333.

[69] K. Yamamoto, T. Togami, N. Yamaguchi, and S. Ninomiya, "Machine learning-based calibration of low-cost air temperature sensors using environmental data," *Sensors*, vol. 17, no. 6, p. 1290, Jun. 2017, doi: 10.3390/s17061290.

[70] D. Park, G.-W. Yoo, S.-H. Park, and J.-H. Lee, "Assessment and calibration of a low-cost $PM_{2.5}$ sensor using machine learning (HybridLSTM neural network): Feasibility study to build an air quality monitoring system," *Atmosphere*, vol. 12, no. 10, p. 1306, Oct. 2021, doi: 10.3390/atmos12101306.

[71] *European Environment Agency*. Accessed: Dec. 13, 2023. [Online]. Available https://www.eea.europa.eu/en

[72] S. Tondini, "Harmful pollutants and microclimatic parameters from autonomous low-cost sensors deployed in the city center of Bolzano, Italy," Dataset, Eurac Res., Bolzano, Italy, 2022, doi: 10.48784/MYPZ-EV45.
[73] E. Acuña and C. Rodríguez, "An empirical study of the effect of outliers on the misclassification error rate," *IEEE Trans. Knowl. Data Eng.*, 2005. [Online]. Available: https://www.researchgate.net/profile/Edgar-Acuna/publication/239551988_An_empirical_study_of_the_effect_of_outliers_on_the_misclassification_error_rate/links/00b7d534430cad41ce000000/An-empirical-study-of-the-effect-of-outliers-on-the-misclassification-error-rate.pdf
[74] A. Khamis, "The effects of outliers data on neural network performance," *J. Appl. Sci.*, vol. 5, no. 8, pp. 1394–1398, Jul. 2005, doi: 10.3923/JAS.2005.1394.1398.
[75] M. B. Richman, T. B. Trafalis, and I. Adrianto, "Missing data imputation through machine learning algorithms," in *Artificial Intelligence Methods in the Environmental Sciences*, W. E. Haupt, A. Pasini, and C. Marzban, Eds. Dordrecht, The Netherlands: Springer, 2009, pp. 153–169, doi: 10.1007/978-1-4020-9119-3_7.
[76] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, vol. 5, Jan. 2018, Art. no. e3190v2, doi: 10.7287/peerj.preprints.3190v2.
[77] X. Wang et al., "Toward accurate anomaly detection in industrial Internet of Things using hierarchical federated learning," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7110–7119, May 2022, doi: 10.1109/JIOT.2021.3074382.
[78] Y. Liu et al., "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021, doi: 10.1109/JIOT.2020.3011726.
[79] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: An application to sensor data," *Knowl. Inf. Syst.*, vol. 11, no. 2, pp. 137–154, Feb. 2007, doi: 10.1007/s10115-006-0026-6.
[80] F. Harrou, A. Dairi, Y. Sun, and F. Kadri, "Detecting abnormal ozone measurements with a deep learning-based strategy," *IEEE Sensors J.*, vol. 18, no. 17, pp. 7222–7232, Sep. 2018, doi: 10.1109/JSEN.2018.2852001.
[81] H. Liu, "Feature selection," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. New York, NY, USA: Springer, 2011, pp. 402–406, doi: 10.1007/978-0-387-30164-8_306.
[82] X. Ying, "An overview of overfitting and its solutions," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 022022, doi: 10.1088/1742-6596/1168/2/022022.
[83] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, Mar. 2013, doi: 10.1007/s10115-012-0487-8.
[84] F. Ferri, P. Pudil, M. Hatef, and J. Kittler, "Comparative study of techniques for large-scale feature selection," in *Machine Intelligence and Pattern Recognition*, vol. 16. Amsterdam, The Netherlands: Elsevier, 1994, pp. 403–413.
[85] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. Cambridge, MA, USA: MIT Press, 2014.
[86] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines Theories, Concepts, and Applications for Engineers and System Designers*. New York, NY, USA: Apress, 2015, pp. 67–80.
[87] M. Bramer, *Principles of Data Mining*, 3rd ed. London, U.K.: Springer, 2016, doi: 10.1007/978-1-4471-7307-6.
[88] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[89] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, "Efficient densely connected convolutional neural networks," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107610, doi: 10.1016/j.patcog.2020.107610.
[90] C. Chatfield and H. Xing, *The Analysis of Time Series: An Introduction With R* (Chapman and Hall/CRC Textes in Statistical Science). Boca Raton, FL, USA: CRC Press, 2019.
[91] K. Ito and R. Nakano, "Optimizing support vector regression hyperparameters based on cross-validation," in *Proc. Int. Joint Conf. Neural Netw.*, 2003, pp. 2077–2082.
[92] P. Probst, M. N. Wright, and A. Boulesteix, "Hyperparameters and tuning strategies for random forest," *WIREs Data Mining Knowl. Discovery*, vol. 9, no. 3, p. e1301, Jan. 2019, doi: 10.1002/widm.1301.
[93] M. T. M. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: Fundamentals and evolutionary methods," *Natural Comput.*, vol. 17, no. 3, pp. 585–609, Sep. 2018, doi: 10.1007/s11047-018-9685-y.
[94] R. Mahajan and G. Kaur, "Neural networks using genetic algorithms," *Int. J. Comput. Appl.*, vol. 77, no. 14, pp. 6–11, Sep. 2013, doi: 10.5120/13549-1153.
[95] A. Benitez-Hidalgo, A. J. Nebro, J. Garcia-Nieto, I. Oregi, and J. D. Ser, "JMetalPy: A Python framework for multi-objective optimization with metaheuristics," 2019, *arXiv:1903.02915*.
[96] P. Fay, "Photodetectors," in *Encyclopedia of Materials: Science and Technology*. Amsterdam, The Netherlands: Elsevier, 2001.
[97] M. F. Larsen, "Observation platforms," in *Encyclopedia of Atmospheric Sciences*. New York, NY, USA: Academic, 2002, pp. 1449–1454, doi: 10.1016/B0-12-227090-8/00258-X.
[98] F. Tian, S. Yang, and K. Dong, "Circuit and noise analysis of odorant gas sensors in an E-nose," *Sensors*, vol. 5, no. 1, pp. 85–96, Feb. 2005, doi: 10.3390/s5010085.



Stefano Tondini (Member, IEEE) was born in Trento, Italy, in 1987. He received the B.S. degree in physics and the M.S. degree in experimental physics from the University of Trento, Trento, Italy, in 2010 and 2013, respectively, and the Ph.D. degree in physics and nanoscience from the University of Modena and Reggio Emilia, Modena, Italy, in 2017.

In 2015, he has been a Visiting Research Fellow at the Indian Institute of Technology in Kharagpur, Kharagpur, India. In 2017, he joined the Industrial Engineering Department, University of Trento, as a Post-Doctoral Fellow with a focus on secondment to the local entrepreneurial association Confindustria Trento (Italy). From 2018 to 2021, he was with Eurac Research, Bolzano, Italy, as a Senior Researcher, at the Institute for Earth Observation and at the Center for Sensing Solutions, Eurac Research. In 2019, he joined the Afromontane Research Unit at the Free State University, Phuthaditjhaba, South Africa, and at the Institute of Water and Energy Sciences of Pan African University, Tlemcen, Algeria, as a Visiting Researcher. He is currently a Marie Skłodowska-Curie Fellow with the Photonics Integration Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands, in secondment by the Environmental Sensing and Modeling Group, Technische Universität München, Munich, Germany. His current research interests include integrated photonics, software defined networking, Internet-of-Things (IoT) communication protocols, context-aware information systems, machine learning (ML)-based optimization methods, ontology environmental/pollution sensing, IPRs protection, and technology transfer.

Dr. Tondini is an SPIE and IAOA member. AICA and Rotary International has awarded him for the best Ph.D. thesis on Computer Ethics in 2017. He got a top-scored oral presentation at the OFC 2018 in San Diego (US).

Riccardo Scilla was born in Treviso, Italy, in 1997. He received the B.S. degree in informatic engineering from the University of Trieste, Trieste, Italy, in 2019, and the M.S. degree in informatics from the University of Trento, Trento, Italy, in 2021.

His current research interests include machine learning, image recognition, neural networks, time-series forecasting, environmental sensors, open hardware, and open software.

Paolo Casari (Senior Member, IEEE) received the Ph.D. degree in information engineering from the University of Padua, Padua, Italy, in 2008.

He was on leave at the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007, worked on underwater communications and networks. He is an Associate Professor with the University of Trento, Trento, Italy. He is currently the Principal Investigator of the NATO SPS project SAFE-UComm. Previously, he coordinated the NATO SPS project ThreatDetect, and was the Scientific Coordinator of the EU H2020 RECAP and SYMBIOSIS projects. Between 2015 and 2019, he was with the IMDEA Networks Institute, Madrid, Spain, where he led the Ubiquitous Wireless Networks Group. His research interests include many aspects of wireless networked communications, such as channel modeling, protocol design, localization, simulation, and experimental evaluations.

Dr. Casari received two best paper awards. He serves on the editorial boards of IEEE Transactions on Mobile Computing and IEEE Transactions on Wireless Communications, and regularly collaborates with the organizing committee of several international conferences.