Comparative Evaluation of Low-Cost CO2 Sensors for Indoor Air Pollution Monitoring

Rishikesh Bose, Ayu Parmar, Harsha Narla, Sachin Chaudhari
International Institute of Information Technology Hyderabad, India
{rishikesh.bose, ayu.parmar} @research.iiit.ac.in, harsha.narla@students.iiit.ac.in, sachin.chaudhari@iiit.ac.in

2022 IEEE 8th World Forum on Internet of Things (WF-IoT) | 978-1-6654-9153-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/WF-IOT54382.2022.10152133

***Abstract*—In this paper, four low-cost CO2 sensors are evaluated for IoT-based indoor air pollution monitoring. Specifically, CO2 sensors SCD30, Prana Air, MHZ14, and T6713 are evaluated against a standard reference Aeroqual S-500 device. The experiment was carried out in an indoor environment inside one of the labs in IIIT Hyderabad, India. It is shown that calibration is needed for some of these low-cost devices locally even though the sensors may be factory calibrated. For calibration, simple and widely-used machine learning algorithms are employed such as linear regression, least absolute deviation, random forest, support vector regression, and Gaussian regression. The parameters considered to assess the performance of these sensors are coefficient of determination (R^2), coefficient of variability (C_v), and root mean square error (RMSE). After calibration with a reference sensor, it is observed that these low-cost sensors operate well.**

***Index Terms*—IoT, Determination coefficient, Low-cost CO2 sensor, Coefficient of variability, Root mean square error.**

I. Introduction

Carbon dioxide (CO2) is one of the major contributors to indoor air pollution [1]. Exposure to CO2 can produce a variety of health effects, including headaches, dizziness, restlessness, difficulty in breathing, sweating, tiredness, etc [2]. The CO2 level over 800 ppm is hazardous for health and also increases the transmission rate of coronavirus [2]. So, it is important to monitor CO2 levels and take appropriate precautionary actions such as turning on the ventilation system or opening the doors and windows. In addition to pollution monitoring, CO2 monitoring can also be used for interesting applications such as detecting room occupancy [3].

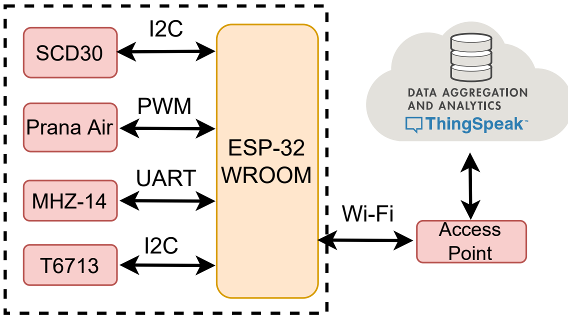
There are several high-end CO2 monitoring systems available in the market such as Amprobe CO2-100 Handheld Carbon Dioxide Meter [4]. However, they are too expensive for indoor air pollution monitoring in smart home and workspace applications. Many low-cost sensors for monitoring CO2 levels are currently available on the global market. These sensors are compact, lightweight and can be deployed in large numbers. As such they are employed in the internet of things (IoT)-based smart city initiatives and several companies have started selling low-cost sensing kits including CO2 for monitoring indoor air pollution. However, one of the biggest disadvantage of the low-cost sensors is reliability. Therefore it is important to assess their performance and suggest ways to improve them so that these low-cost IoT-based indoor air monitoring systems can be adopted and used on scale.

There have been few studies of low-cost CO2 sensors in the literature [5]–[7]. In [5], four low-cost Non-Dispersive Infrared (NDIR) based CO2 sensors were examined namely K30, AN100, S100, and T6615, which were compared with an industry manufactured CO2 analyzer Li-6262 using span calibration for limited gas concentrations (0, 407, 1,110, and 1,810 ppm) for RMSE performance parameter. In [6], NDIR-based MH-Z16 and MOS-based MG-811 were analyzed through exposure to different gas concentrations in an indoor environment. The calibration of the MOS sensors was found to be unstable. In [7], performance evaluation was done for low-cost sensors measuring air temperature, relative humidity, CO2, and PM. For CO2, few low-cost sensors SCD40, SenseAir S8, T6703-5 K, iAQ-Core C, LP8, and ELT T110 were used. The sensors were calibrated with a reference device Li-COR using linear regression.

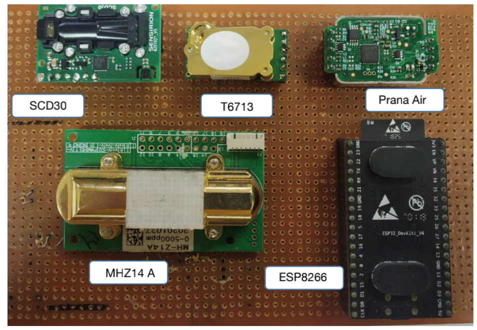
The specific contributions of this paper are

- Four low-cost CO2 sensors, namely SCD30 [8], Prana Air [9], MHZ14 [10] and T6713 [11] are evaluated in this study. These sensors are easily available in India and have not been compared for IoT-based indoor air pollution monitoring till date.
- Three identical test-setups (containing four CO2 sensors each) are used to study the test sensor's behavior. These setups were placed along with a reference sensor Aeroqual S-500 [12] inside a lab in IIIT, Hyderabad, India. Data for each of the 12 sensors (four sensors each in three sets) was collected per minute over 20 days.
- It is shown that these sensors need local calibration even though they are factory calibrated.
- For calibration of these low-cost CO2 sensors, different machine learning algorithms are employed such as linear regression, least absolute deviation regression, random forest, support vector regression, and Gaussian regression.
- The performance evaluation of these sensors and calibration algorithms is carried out in terms of coefficient of determination (R^2), coefficient of variability (C_v) and root mean square error (RMSE).

Unlike [5], [6], which use 2-point span calibration in a ideal laboratory environment, this paper does calibration with co-located measurements with a reference sensor over several days in a practical scenario. Also, unlike [7], where only linear regression is used, this paper evaluates several machine learning algorithms for calibration. Note that the sensors compared in [5]–[7] are different than the ones used in this paper.



(a) Block diagram



(b) Actual view

(c) Aeroqual (Reference device)

Fig. 1: Hardware Description

The structure of the rest of the paper is as follows. In Section II, a brief description of the sensors, nodes, and reference device is given. In Section III, the measurements and the evaluation criteria that were used to compare the test sensors using the data collected during the testing are explained. In Section IV, the regression models that are employed for calibration are discussed. Section V presents performance evaluation parameters. Section VI presents the results, while Section VII concludes the paper.

II. Test Setup

This section offers details about the CO2 sensors, the reference sensor as well as the test setup.

A. Sensor Description

Table I show the specifications of the CO2 sensors used in this paper, i.e., MHZ14 [10], Prana [9], SCD30 [8], and T6713 [11]. They are selected based on their availability, cost, and suitability (accuracy and range) for monitoring CO2 in an indoor environment. All sensors justify the minimum range of CO2 for indoor environment which is 400 ppm. The sensors are low cost ranging from 3400-8610 INR (approximately 40-100 USD) [1]. The initial response time is also adequate which is up to 90 sec. As these sensors' accuracy is comparable, the evaluation has greater significance. The life expectancy of these sensors is 12-15 years. All the mentioned sensors use the NDIR principle. Indicative environmental monitoring is particularly suited for NDIR sensors since they are extremely selective to CO2. When infrared radiation interacts with gas molecules, infrared light is absorbed by the gas molecules at a particular wavelength, causing vibration of the gas molecules, and detects the decrease in transmitted infrared light which is in proportion to gas concentration [5], which allows the sensor to detect CO2 molecules appropriately.

[1]Assuming the conversion rate of 1 USD = 80 INR in July 2022.

TABLE I: Sensor Parameters

Sensor	Range (ppm)	Response	Accuracy	Cost
MHZ14	0-2000	20 s	30 ppm $\pm$ 3%	3396 Rs
Prana	0-10000	15 s	50 ppm $\pm$ 2%	3990 Rs
SCD-30	400-10000	30 s	30 ppm $\pm$ 3%	4800 Rs
T6713	0-5000	90 s	30 ppm $\pm$ 3%	8610 Rs

B. Nodes

Three identical test nodes were created for the experiment. Figs. 1(a) and 1(b) show the block diagram and actual view respectively, of a test node. Each node consists of one unit of SCD30, Prana Air, MHZ14, and T6713. Wi-Fi enabled ESP32 micro-controller module was used to interface these sensors. Sensing interval is one sample per minute. The sensed data is sent to the cloud-based ThingSpeak platform using the Wi-Fi.

C. Reference Instrument

Fig. 1(c) shows the reference sensor Aeroqual S-500 monitor [12] along with the CO2 sensor head [13] for indoor quality monitoring. It is a NDIR sensor-based analyzer used to monitor the amount of CO2 gas in the surrounding air. The reference sensor module is compact, portable, and lightweight. A case study reports a high correlation between this portable monitor and high quality CO2 monitors [13].

III. Measurement

This section briefly explains the data collection as well as data pre-processing used in this work.

A. Data Collection

In this experiment, all the three identical setups containing four CO2 sensors each and the reference instrument were placed inside a lab of IIIT Hyderabad. All devices were powered using a power supply. The main objective was to evaluate the ability of sensors to record the low as well as the high-level concentration in CO2 levels. To adequately record CO2 concentrations, the room's windows were kept closed for the duration of the experiment. The data from each sensor was collected every minute over 20 days in the month of May 2022. Approximately 26,356 data points were collected in total (few data points were lost due network connectivity faliure, ideal number of samples should be 28,800). All data was retrieved and downloaded in CSV format from the ThingSpeak platform for each of the twelve sensors. The data of the reference device was downloaded separately using Aeroqual. The data was processed and analyzed using the Python programming language for 1-minute, 15-minutes, 30-minutes and 1-hour averaged measurements.

B. Data Pre-processing

The data were filtered for removing the outliers using Z-score for outlier detection [14], This score aids in determining how distant a data value is from the mean and whether it is larger or less than the mean. In other words, Z-score indicates how many standard deviations a data point deviates from the mean. A data point is considered to be significantly different from the other data points if its Z-score is more than 3. A data point like that may be an anomaly, the number of outliers is 1.005 percent in our case after cleaning.

IV. Calibration Models

In this paper, linear regression, random forest, least absolute deviation, support vector regression and Gaussian regression algorithms are used which are explained further in detail.

Linear regression [15] is the supervised machine learning model in which the model finds the best fit line between the independent and dependent variable such that

$$\hat{y}_i = m x_i + c, \tag{1}$$

where x_i is the independent variable, $\hat{y}_i$ is dependent variable while m and c are the slope and intercept of the fitted line, respectively. During the training phase, the model is given x_i as input training data and y_i as labels so that we can find the optimal c and m values corresponding to the best line fit. Once trained, the model gives the calibrated value $\hat{y}_i$ based on the value of the input x_i.

Least absolute deviations [15] is a statistical optimal criterion. It is also known as Least absolute errors, Least absolute value, Least absolute residual, sum of absolute deviations, or the L1 norm condition. It tries to find a function that closely approximates a set of data, similar to the least squares method. The approximation function for a collection of (x_i, y_i) data is a simple trend line in two-dimensional Cartesian coordinates. The sum of absolute errors is minimised using this strategy (the sum of the absolute values of the vertical residuals between points generated by the function and corresponding points in the data).

$$\hat{y}_i = m_i + \sum_{k=1}^{K} (m_k x_{ik} + c_i), \tag{2}$$

where is the i^{th} observation on the dependent variable, x_{ik} is the i^{th} observation on the k^{th} explanatory variable, and c_i is a random disturbance for the i^{th} observation.

Random forest regression [15] is a supervised learning algorithm that uses the ensemble learning method for regression. Regression in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This data set consists of observations and features that will be selected randomly during the splitting of nodes.

The Gaussian process regression [15] implements Gaussian processes. The prior mean is taken to be either the mean of the training data or constant and zero. Passing a kernel object specifies the prior's covariance. During the fitting of the Gaussian process regression, the hyper-parameters of the kernel are optimised by maximising the log-marginal-likelihood depending on the passed optimizer. The optimizer can be launched numerous times by setting restart optimizer since the log-marginal-likelihood may contain several local optima. The initial run is always started from the kernel's initial hyper-parameter values, and subsequent runs are started from random hyper-parameter values drawn from the range of permitted values.

Support vector regression [15] is used to forecast discrete values. The support vector machines and support vector regression both operate on the same theory. Finding the optimum fit line is the fundamental tenet of this algorithm. The hyperplane with the most points is the best-fitting line. Unlike other regression models that try to minimize the error between the real and predicted value, it tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line. The model aims to fit the best line within a threshold value, as opposed to other Regression models that aim to reduce the error between the real and projected value. The gap between the hyperplane and boundary line is the threshold value.

V. Performance Evaluation Parameters

The R^2, C_v and RMSE performance parameters are employed in this study. In relation to the reference instrument, the coefficient of determination R^2 examines the capacity to report changes in CO2 levels, which is given by

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2}, \tag{3}$$

where y_i denotes the actual value of reference sensor, $\bar{y}$ denotes the average of y_i, and $\hat{y}_i$ denotes the fitted model's prediction of x_i. Raw sensor output data as well as calibrated values were used to calculate the R^2 values. The ideal R^2 value should be 1 indicating that the observations and the fitted model are same. In practice, however, R^2 should be as close to 1 as much as possible.

Coefficient of variability C_v measures the reproducibility or the variance across three units of the same sensor and is given by

$$C_v = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{\sigma_i}}{\hat{\mu_i}} \times 100\%, \tag{4}$$

where N is the total number of samples, $\hat{\mu_i}$ denotes the sample mean of readings of all three units of a particular sensor at one moment of time and $\hat{\sigma_i}$ is the average standard deviation of all the copies of a sensor. C_v values should be as low as possible, lower C_v values signify low variance across the sensors. The square root of the sum of squares of errors, represented by E_{rms}, is a metric that represents the average of the square root of the sum of squares of errors and is provided by:

$$E_{rms} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(y_i - z_i)^2}, \tag{5}$$

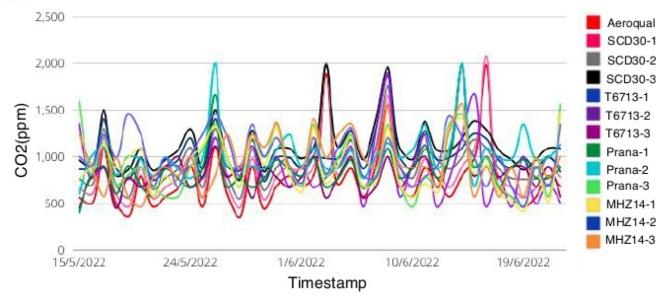
where z represents sensors' value before calibration, predicted value after calibration. E_{rms} values should be as low as possible (closer to zero).

VI. Results

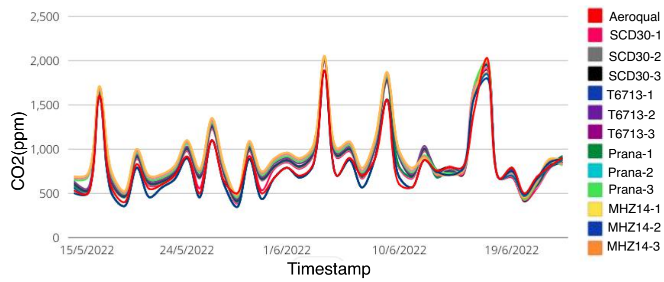
Fig. 2(a) shows the trend using raw data (before calibration) from the twelve sensors (three quantities each of four types) as well as the reference device for one hour average. The measured values are in the range of 400 ppm and 2100 ppm. It can be seen that, although, the sensors are following the trend of the reference in general, there is mismatch at several instances. Fig 2(b) shows the trend of the sensor data after calibration using random forest. It can be seen that the trends are much more in agreement after the calibration with the reference device.

Table II shows the E_{rms} and C_v values for 1-hr averaged samples for sensors in setup-1 before calibration. Although E_{rms} and C_v values for all three setups are in a similar range, only setup-1 values for brevity. T6713 has the best performance followed by SCD30, Prana and MHZ14. It can be also seen that similar to Fig. 2(a), both E_{rms} and C_v values are high indicating mismatch in the trends. For example, E_{rms} values are in the range of 6-29 and C_v in 5-15%. Note that these sensors are factory calibrated as per their specification sheet. This shows that the CO2 sensors considered in this paper, except T6713, have to calibrated locally before deployment. After local calibration, these CO2 sensors operate well as will be shown further.

Figs. 3-6 show the scatter plots for the CO2 values for the sensors MHZ14, SCD30, T6713 and Prana, respectively, with the reference sensor. The scatter plots are plotted for different averaging time intervals (1-min, 15-mins, 30-mins and 1-hour) for both before and after calibration. For after calibration, random forest algorithm is considered. First observation from the scatter plot is that, the raw data has deviations and offsets as compared to the reference sensor data although the offset tend to decrease as averaging interval increases. Second observation is that, the low-cost sensors in general are over-estimating



(a) CO2 vs Time (1-hr avg before calibration)



(b) CO2 vs Time (1-hr avg after calibration)

Fig. 2: CO2 vs Time (Calibrated 1-hr avg using random forest)

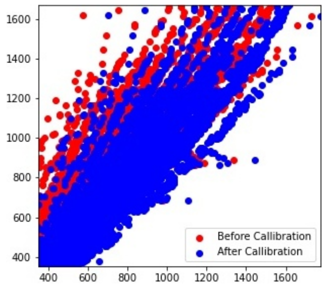
the CO2 values as compared to the reference device. Third observation is that, the sensor's data and reference instrument's data follow a similar trend after calibration, where the results improve as we increase the average time interval starting from 1 minute to 1 hour. Fourth observation is that, the sensor T6713 requires least amount of calibration, whereas sensors like MHZ14, Prana air, and SCD30 requires proper calibration.

TABLE II: E_{rms} and C_v values for 1-hr averaged readings before calibration for sensors in setup-1.

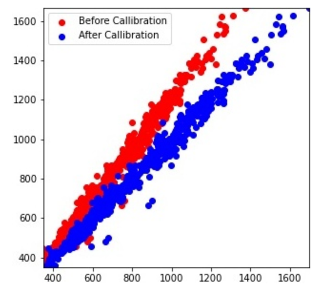
Performance Parameter	**Sensor Name**			
	MHZ14	***Prana Air***	***SCD30***	***T6713***
E_{rms}	29.24	24.68	14.36	6.12
C_v	15.69	12.34	9.62	5.23

Table III contains the average R^2 values of all sensors calculated after performing the calibration using different machine learning algorithms. The coefficient of determination is found to be in the same range using different calibration algorithms. It can be seen that T6713 was giving best values for all the calibration models (0.98). Along with that, for individual calibration models, linear regression gave best results for T6713 for 1-hr average followed by random forest, least absolute deviation, support vector machine, and Gaussian regression.

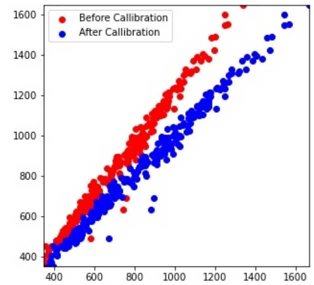
Table IV shows the C_v values for the four sensors after calibration using different algorithms. Only 1-hr averaging



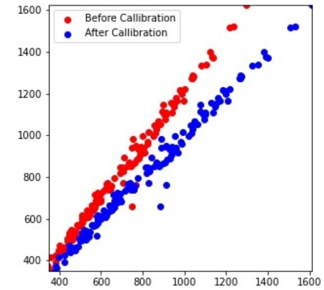
(a) 1-min average



(b) 15-min average

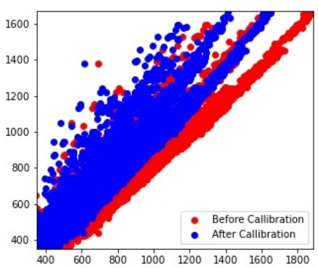


(c) 30-min average

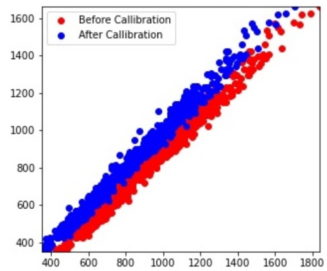


(d) 1-hr average

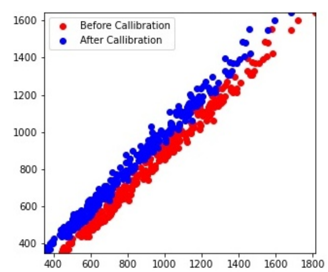
Fig. 3: Scatter plots for MHZ14 vs Aeroqual



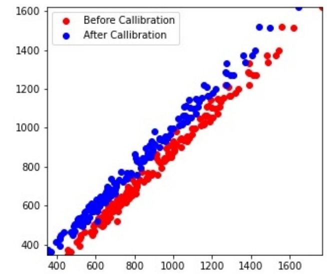
(a) 1-min average



(b) 15-min average

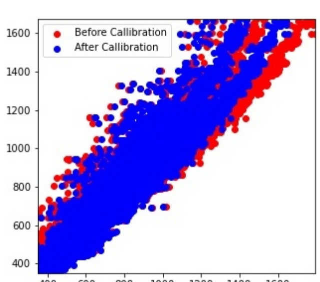


(c) 30-min average

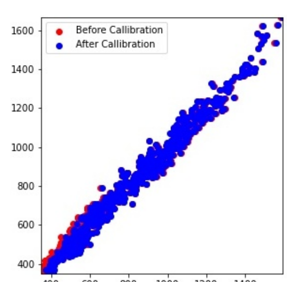


(d) 1-hr average

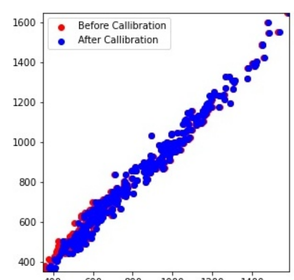
Fig. 4: Scatter plots for SCD30 vs Aeroqual



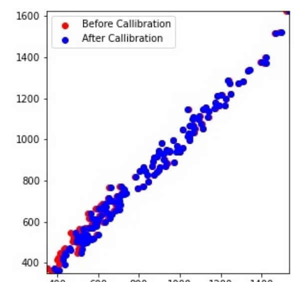
(a) 1-min average



(b) 15-min average

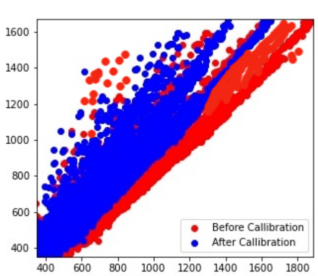


(c) 30-min average

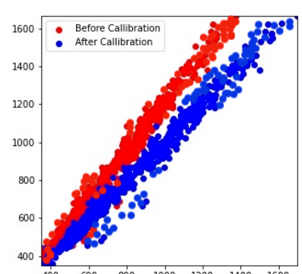


(d) 1-hr average

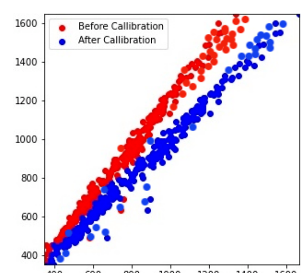
Fig. 5: Scatter plots for T6713 vs Aeroqual



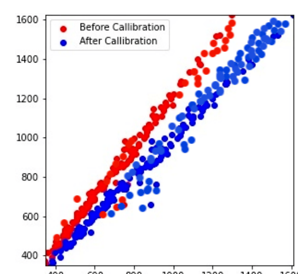
(a) 1-min average



(b) 15-min average



(c) 30-min average



(d) 1-hr average

Fig. 6: Scatter plots for Prana vs Aeroqual

TABLE III: R^2 values using calibration model

Regression Model	Averaging Interval	Sensor Name			
		MHZ14	***Prana***	***SCD30***	***T6713***
Linear regression	1 min	0.78	0.796	0.833	0.846
	30 min	0.829	0.84	0.968	0.978
	1 hr	0.884	0.897	0.974	0.983
Least absolute deviation	1 min	0.79	0.81	0.812	0.846
	30 min	0.843	0.86	0.918	0.924
	1 hr	0.91	0.926	0.957	0.976
Random forest	1 min	0.792	0.803	0.872	0.895
	30 min	0.886	0.87	0.960	0.981
	1 hr	0.906	0.913	**0.977**	**0.984**
Support vector regression	1min	0.77	0.808	0.823	0.86
	30 min	0.863	0.85	0.91	0.939
	1 hr	**0.938**	**0.944**	0.959	0.971
Gaussian regression	1 min	0.78	0.804	0.86	0.86
	30 min	0.849	0.874	0.89	0.936
	1 hr	0.89	0.904	0.959	0.97

TABLE IV: C_v values for 1 hr averaged reading after calibration

Regression method (1-hr)	Sensor Name			
	MH14	***Prana***	***SCD30***	***T6713***
Linear regression	9.69	7.62	5.56	3.22
Least absolute deviation	**8.64**	**6.38**	5.83	3.56
Random forest	8.69	6.62	**5.54**	**3.08**
Suport vector machine	8.89	6.74	5.88	3.16
Gaussian Regression	9.02	6.88	5.78	3.19

TABLE V: E_{rms} values for 1 hr averaged readings after calibration

Regression method (1-hr)	Sensor Name			
	MHZ14	***Prana***	***SCD30***	***T6713***
Linear regression	5.69	5.02	3.56	2.42
Least absolute deviation	**5.44**	5.31	3.63	2.73
Random forest	5.89	**4.23**	3.89	**1.98**
Support vector regression	6.02	5.54	3.62	2.23
Gaussian regression	5.45	5.24	**3.12**	2.26

intervals are considered. It can be observed that there is a good change in C_v values for all sensors after calibration, which dropped down from 15% to 9% for MHZ14, 12% to 6% for prana, 9% to 5% for SCD30, and 5% to 3% for T6713. It is seen that the T6713 sensor has the lowest C_v value for this experiment.

Table V shows the E_{rms} values for 1 hr averaged readings after calibration using different algorithms. It was observed that the values dropped in the range 1-5 from 6-29 after calibration. Also, T6713 sensor requires least amount of calibration as the E_{rms} difference is very minimal which is dropped from 6.12 to 3.16. Along with that, best results for E_{rms} is obtained using random forest algorithm which is 1.98 for T6713, 3.12 for SCD30 using Gaussian regression, 5.02 for prana using linear regression, and 5.44 for MHZ14 using least absolute deviation.

VII. Conclusion

In this paper, four low-cost CO2 sensors, MHZ14, Prana, SCD30 and T6713 are evaluated. It is demonstrated that, except for T6713, calibration is needed locally for the rest of the sensors, even though they may be factory calibrated. T6713 sensor gave the most accurate values after calibration, followed by SCD30, Prana Air and MHZ14. However, the latter three can also be used for CO2 monitoring in indoor environments as they function effectively after calibration. After using linear regression, random forest, least absolute deviation, support vector regression, and Gaussian regression, we were able to achieve a higher determination of coefficient (0.9 and above), lower coefficient of variability (3%) as well as lower root mean square value (1.9). While the performance of all the calibration models is in the same range, the random forest regression provides the best result considering all the parameters. Overall, the low-cost CO2 sensors can be used in IoT-based indoor air pollution monitoring systems, subject to calibration required to ensure reliable output.

Acknowledgment

This research was supported partly by National Geospatial Programme (NGP), India, under grant no. 2073 (2020), PRIF Social Incubator Program (2019) and the Ministry of Electronics and Information Technology (MEITY), Govt. of India under grant no. 3070665 (2020) as part of the Smart City Living Lab project, with no conflict of interests.

References

[1] C. Mandin, "The indoor air quality observatory: a unique project to understand air pollution in our living spaces," *Open edition journal*, 2018.

[2] K. Azuma, N. Kagi, U. Yanagi and H. Osawa, "Effects of low-level inhalation exposure to carbon dioxide in indoor environments: A short review on human health and psychomotor performance," *Environment International*, 2018.

[3] A. Singh, P. Jain, S. Chaudhari, F. Kraemer, "Machine learning-based occupancy estimation using multivariate sensor nodes," *IEEE Globcom workshops*, 2018.

[4] "High end CO2 sensor," https://www.amprobe.com/product/co2-100/.

[5] T. Yasuda, S. Yonemura, and A. Tani, "Comparison of the characteristics of small commercial NDIR CO2 sensor models and development of a portable CO2 measurement device," *MDPI IoT*, Mar, 2016.

[6] T. Araújo, L. Silva, and A. Moreira, "Evaluation of low-cost sensors for weather and carbon dioxide monitoring in internet of things context," *MDPI IoT*, vol. 1, pp. 286–308, 2020.

[7] I. Demanega et al., "Performance assessment of low-cost environmental monitors and single sensors under variable indoor air quality and thermal conditions," *Building and Environment*, vol. 187, p. 107415, 2021.

[8] "Sensirion sensor specifications," accessed 21 March. 2022, https://www.mouser.in/new/sensirion/sensirion.com.

[9] "Prana air sensor specifications," https://www.pranaair.com/air-quality-sensor/carbon-dioxide-co2-sensor/, accessed 21 March. 2022.

[10] "Mhz14 sensor specifications," https://www.co2meter.com/products/MHZ14-co2-sensor-module, accessed 21 March. 2022.

[11] "Telair sensor specifications," accessed 21 March. 2022, https://www.amphenol-sensors.com.

[12] "Aeroqual specifications," https://www.aeroqual.com/product/series-500-portable-indoor-monitor, accessed 21 March. 2022.

[13] A. R. Whitehill, "Evaluation of cairpol and aeroqual air sensors in biomass burning plumes," accessed 21 March. 2022, https://www.mdpi.com/2073-4433/13/6/877/htm.

[14] L. J. Murray R.S., *Statistics*. McGraw-Hill Education, 2012.

[15] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.