Supporting Information for the paper titled "Robust statistical calibration and characterization of portable low-cost air quality monitoring sensors to quantify real-time O_3 and NO_2 concentrations in diverse environments"

Ravi Sahu[1], Ayush Nagal[2], Kuldeep Kumar Dixit[1], Harshavardhan Unnibhavi[3], Srikanth Mantravadi[4], Srijith Nair[4], Yogesh Simmhan[3], Brijesh Mishra[5], Rajesh Zele[5], Ronak Sutaria[6], Purushottam Kar[2], and Sachchida Nand Tripathi[1]

[1]Department of Civil Engineering, Indian Institute of Technology Kanpur, Kanpur, India
[2]Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, India
[3]Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India
[4]Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India
[5]Department of Electrical Engineering, Indian Institute of Technology, Bombay, India
[6]Centre for Urban Science and Engineering, Indian Institute of Technology, Bombay, India

Correspondence: Sachchida Nand Tripath (snt@iitk.ac.in)

Abstract. This document presents supporting material to the main paper, including descriptions of various baseline parametric and non-parametric baseline calibration algorithms, as well as details about the calibration performance offered by various algorithms that are not included in the main paper.

Appendix A: A Recapitulation of Notation and Error Metrics

For sake of convenience to the reader, we revisit the definitions of the error metrics, a discussion on how to interpret the statistical tests, and the notation.

A1 Error Metrics and Statistical Hypothesis Testing

Error Metrics: calibration performance was measured using four popular metrics, mean averaged error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and the coefficient of determination (R^2) (see below). Here n denotes the number of test points for a given dataset and split thereof, the variable t runs over all time-stamps in the testing set, y^t denotes the reference value (either O_3 or NO_2) at the t-th time-stamp, $\hat{y}^t$ denotes the corresponding value predicted by the

calibration model, and $\bar{y}$ denotes the mean reference value i.e. $\bar{y} = \frac{1}{n}\sum_{t=1}^{n} y^t$.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y^t - \hat{y}^t|$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|y^t - \hat{y}^t|}{y^t} \times 100\%$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y^t - \hat{y}^t)^2}$$

$$\text{R}^2 = 1 - \frac{\sum_{t=1}^{n}(y^t - \hat{y}^t)^2}{\sum_{t=1}^{n}(y^t - \bar{y})^2}$$

Statistical Hypothesis Tests: in order to compare the performance of different calibration algorithms on a given dataset (e.g., to find out the best performing algorithm), or compare the performance of the same algorithm on different datasets (e.g., to find out the effect of data characteristics on calibration performance), we performed paired and unpaired two-sample tests, respectively. Our null hypothesis in all such tests proposed that the absolute errors offered by the two algorithms on the same dataset (in case of a paired test) or the same algorithm across different datasets (in case of an unpaired test) were sampled from the same distribution. The test was applied and if the null hypothesis was rejected with sufficient confidence (an α value of 0.05 was used as the standard to reject the null hypotheses), then a winner was simultaneously identified.

Although the Student's t-test is most popularly used in such situations, it essentially assumes that the underlying distributions are normal. However, an application of the Shapiro-Wilk test (Shapiro and Wilk, 1965) rejected the null hypotheses of the errors being normally distributed with high confidence. As a result, we chose the non-parametric Wilcoxon signed-rank test (Wilcoxon, 1945) when comparing two algorithms on the same dataset, and its unpaired variant, the Mann-Whitney U-test (Mann and Whitney, 1947) for comparing the same algorithm on two different datasets. These tests do not make any assumption on the underlying distribution of the errors and are well-suited for our data.

A2 Interpreting the Two-sample Tests

As mentioned earlier, we used the paired Wilcoxon signed ranked test to compare two algorithms on the same dataset. Given that there are 12 datasets and 10 splits for each dataset, for ease of comprehension, we provide globally averaged statistics of wins scored by an algorithm over another. For example, say we wish to compare KNN-D(ML) and NW(ML) as done in Tab C4. We perform the test for each individual dataset and split. For each test, we either get a win for NW(ML) (in which case NW(ML) gets a +1 score and KNN-D(ML) gets 0), or a win for KNN-D(ML) (in which case KNN-D(ML) gets a +1 score and NW(ML) gets 0) or else the null hypothesis is not refuted (in which case both get 0). The average of these scores is then shown. For example, in Tab C4 (left), row 3 column 5 records a value of 0.06 implying that in 6% of these tests, NW(ML) won over KNN-D(ML) in case of O_3 calibration, whereas row 5 column 3 records a value of 0.58 implying that in 58% of the tests, KNN-D(ML) won over NW(ML). In the balance (1 - 0.06 - 0.58 = 0.36) i.e. 36% of the tests, neither algorithm could be declared a winner.

A3 Notation

For every time-stamp t, the vector $\boldsymbol{x}^t \in \mathbb{R}^8$ denotes the 8-dimensional vector of signals recorded by the LCAQ sensors for that time-stamp, namely (RH, T, no2op1, no2op2, oxop1, oxop2, no2diff, oxdiff), while the vector $\boldsymbol{y}^t \in \mathbb{R}^2$ will denote the 2-tuple of the reference values of O_3 and NO_2 for that time step. However, this notation is unnecessarily cumbersome since we will build separate calibration models for O_3 and NO_2. Thus, to simplify the notation, we will instead use $y^t \in \mathbb{R}$ to denote the reference value of the gas being considered (either O_3 or NO_2). The goal of calibration will then be to learn a real valued function $f : \mathbb{R}^8 \to \mathbb{R}$ such that $f(\boldsymbol{x}^t) \approx y^t$ for all time-stamps t (the exact error being measured using metrics such as MAE, MAPE, etc described in Sect A1). Thus, we will learn two functions, say f_{NO_2} and f_{O_3} to calibrate for NO_2 and O_3 concentrations respectively. Since several of our calibration algorithms will involve the use of some statistical estimation or machine learning algorithm, we will let N (resp. n) denote the number of training (resp. testing) points for a given dataset and split thereof. Thus, we will let $\{(\boldsymbol{x}^t, y^t)\}_{t=1}^N$ denote the training set for that dataset and split with $\boldsymbol{x}^t \in \mathbb{R}^8$ and $y^t \in \mathbb{R}$.

Appendix B: Description of Baseline Calibration Algorithms

Below we describe some baseline calibration algorithms used in our study.

B1 Parametric Calibration Models

We first consider parametric calibration models that use an affine function to perform calibration i.e. f is of the form $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ where the *model vector* $\boldsymbol{w} \in \mathbb{R}^8$ and *bias term* $b \in \mathbb{R}$ are inferred from data. We describe several such calibration models below.

B1.1 Alphasense Models

The manufacturers of the Ox-B421 and NO_2-B42F sensors used in the SATVAM ensemble themselves provides four different calibration algorithms which are described below. The four models are meant to reflect different operating conditions for these sensors, based on the ambient temperature range. In the following, $\alpha, \beta, \gamma, \alpha', \beta'$ are constants unique to individual units i.e. two sensors placed in two different SATVAM ensembles would have different values for these constants. These constants are provided by the manufacturer, based on their factory calibration.

$$p_1(a,b) = ((a-\alpha) + 0.6 \cdot (b-\beta))/\gamma$$
$$p_2(a,b) = ((a-\alpha) + \alpha'/\beta' \cdot (b-\beta))/\gamma$$
$$p_3(a,b) = ((a-\alpha) + (b-\beta) - (\alpha'-\beta'))/\gamma$$
$$p_4(a,b) = (a-\alpha)/\gamma$$

The AS1, AS2, AS3 and AS4 models: The above present the four different calibration models and can be directly used to calibrate for $\mathrm{NO_2}$ concentrations as follows. For any $k \in 1 \ldots 4$, the k-th calibration model for $\mathrm{NO_2}$ is proposed to be

$$f^k_{\mathrm{NO_2}} = p_k(\text{no2op1, no2op2})$$

Note that the model disregards RH and T information and uses only the no2op1, no2op2 voltage values to perform calibration for $\mathrm{NO_2}$. Now, it turns out that the $\mathrm{O_3}$ sensor is sensitive to the sum of $\mathrm{NO_2}$ and $\mathrm{O_3}$ concentrations. To account for this, the k-th calibration model for $\mathrm{O_3}$ is proposed to be

$$\begin{aligned} f^k_{\mathrm{O_3}} &= p_k(\text{oxop1, oxop2}) - f^k_{\mathrm{NO_2}}(\text{no2op1, no2op2}) \\ &= p_k(\text{oxop1, oxop2}) - p_k(\text{no2op1, no2op2}) \end{aligned}$$

In our experiments, none of the four Alphasense calibration models offered satisfactory performance. We note that similar equations were recently used by (Chatzidiakou et al., 2019) for calibration of $\mathrm{NO_x}$, $\mathrm{O_3}$ and CO concentrations although the constants in their models are learnt from data from actual deployment rather than factory deployment.

B1.2 Linear Models

Since the Alphasense calibration models are essentially affine functions of the feature vector $\boldsymbol{x} \in \mathbb{R}^8$, we implemented linear regression techniques to fully explore the calibration power of predictive power of affine functions.

Least Squares (LS): The standard least squares formulation seeks to learn a model vector and bias value that minimizes the (squared) RMSE error by solving the following optimization problem over training data:

$$(\boldsymbol{w}^{\mathrm{LS}}, b^{\mathrm{LS}}) = \arg\min_{\substack{\boldsymbol{w}\in\mathbb{R}^8 \\ b\in R}} \sum_{t=1}^{N} ((\boldsymbol{w}^\top \boldsymbol{x}^t + b) - y^t)^2$$

Least Squares on reduced features (LS(MIN)): To assess the effect of the RH and T features as well as the augmented features oxdiff and no2diff, we performed least squares regression on a reduced feature set by withholding the augmented features and RH and T values.

Sparse linear regression (LASSO): The LASSO formulation (Tibshirani, 1996) seeks to learn a sparse model i.e. a model vector $\boldsymbol{w}$ such that one or more coordinates of $\boldsymbol{w}$ are zero. LASSO can be effective at *feature selection* i.e. indicating which of the 8 input features are most relevant for a particular calibration task. To do so, LASSO solves the following regularized optimization problem over training data:

$$(\boldsymbol{w}^{\mathrm{LASSO}}, b^{\mathrm{LASSO}}) = \arg\min_{\substack{\boldsymbol{w}\in\mathbb{R}^8 \\ b\in R}} \lambda \cdot \|\boldsymbol{w}\|_1 + \sum_{t=1}^{N} ((\boldsymbol{w}^\top \boldsymbol{x}^t + b) - y^t)^2,$$

where $\|\boldsymbol{w}\|_1 = \sum_{j=1}^{8} |\boldsymbol{w}_j|$ and $|\cdot|$ denotes the absolute value operator. The *regularization parameter* λ was tuned over a fine grid spanning five orders of magnitude $[0.0001, 0.0002, \ldots, 10, 20, 50]$ using held-out validation.

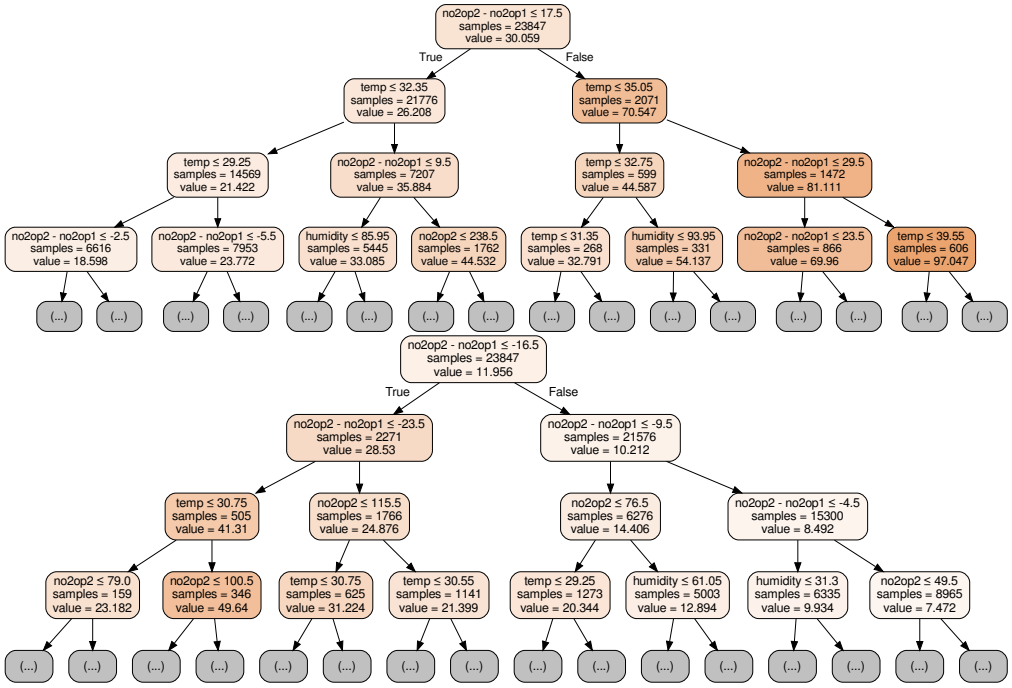

Figure B1. The first three levels of regression trees learnt for O_3 (top) and NO_2 (bottom) calibration for the DD1(Jun) dataset. Each internal node in the tree describes what rule was used to split that node, the number of training samples that reached that node, and the average value of the true (i.e. reference) concentrations of the gas (O_3 or NO_2) within the training samples that reached that node. The shade of color of each node is indicative of the magnitude of the average reference value of training samples at that node. Notice the diversity of the splitting rules used at various nodes in terms of using RH, T as well as electrode potential values (e.g. no2op1, oxop2 etc) to perform the splits.

B2 Nonparametric Calibration Models

We also consider several baseline calibration models based on non-parametric regression techniques, including standard models such as regression trees, and several variants of kernel regression.

B2.1 Regression Trees (RT)

Regression trees are a form of *space partitioning* data structures that recursively subdivide the feature space (in our case $\mathbb{R}^8$) into small regions. Initially, all training samples reside at the *root* node of the tree which represents the entire feature space $\mathbb{R}^8$. Then, a simple rule based on a single feature, for instance whether no2diff = no2op2 - no2op1 ≤ 17.5 or not (see Fig. B1 (top) for this example) is used to *split* this node into two *child* nodes. Thus, training samples at this node that satisfy this rule go to one child and samples that do not satisfy this rule go to the other child. Once the nodes are small enough, i.e. they contain

fewer training samples than a set threshold, the average reference value of samples at that node is used to perform prediction on all (testing) samples that reach that node.

The splitting rules at various nodes are learnt using an exhaustive search to ensure that the child nodes getting created as a result of that rule are as *pure* as possible. In our case, the purity of a node was measured in terms of variance. Specifically, let $\{i_1, \ldots, t_s\} \subseteq [N]$ be the s training samples at a certain node. Then the purity of this node is measured as $\frac{1}{s}\sum_{k=1}^{s}(y^{t_k} - \bar{y})^2$ where $\bar{y} = \frac{1}{s}\sum_{k=1}^{s} y^{t_k}$ is the average reference value of samples at that node.

A standard implementation of a regression tree was used with nodes being asked to be split into two children till the number of training samples at a node fell below a threshold *min_samples*. This threshold was tuned across a fine grid of $[2, 4, 6, 8, 10, 15, 20]$ using held-out validation. Fig. B1 gives examples of actual regression trees learnt on the DD1(Jun) dataset for NO_2 and O_3 calibration.

B2.2 Baseline Algorithms based on $k-$NN Regression Variants

Apart from the Nadaraya-Watson method described in the main paper, we also considered kernel ridge regression and its accelerated version as baseline methods which are described below.

Kernel Ridge Regression (KRR): The KRR algorithm generalizes this technique to learn a parameter value $\{\alpha^t\}_{t=1}^{N}, \alpha^t \in \mathbb{R}$ for every training point. These values are intended to denote the relative importance of various training samples in offering an accurate prediction. Let $\boldsymbol{\alpha} = [\alpha^1, \ldots, \alpha^N] \in \mathbb{R}^N$ denote the vector of these values. Given a feature vector $\boldsymbol{x} \in \mathbb{R}^8$, KRR makes a prediction as follows:

$$f^{\text{KRR}}(\boldsymbol{x}; \boldsymbol{\alpha}) = \sum_{t=1}^{N} \alpha^t \cdot K(\boldsymbol{x}^t, \boldsymbol{x}),$$

These parameters learnt so as to minimize the (regularized) RMSE error on the training set as shown below. Let $G \in \mathbb{R}^{N \times N}$ denote the *Gram* matrix of kernel values among the training points i.e. $G_{ij} = K(\boldsymbol{x}^i, \boldsymbol{x}^j)$.

$$\boldsymbol{\alpha}^{\text{KRR}} = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \lambda \cdot \boldsymbol{\alpha}^\top G \boldsymbol{\alpha} + \sum_{t=1}^{N} (f^{\text{KRR}}(\boldsymbol{x}; \boldsymbol{\alpha}) - y^t)^2,$$

The regularization hyperparameter λ was tuned over a fine grid spanning five orders of magnitude $[0.0001, 0.0002, \ldots, 10, 20, 50]$ using held-out validation. The bandwidth parameter of the Gaussian kernel γ was tuned to the inverse of a certain percentile of the pairwise Euclidean distances between the training feature vectors. This percentile was tuned over the fine grid $[0.1, 0.2, \ldots, 0.9]$ using held-out validation. This range is popularly held as a reasonable range within which a near-optimal bandwidth value can be discovered (Caputo et al., 2002).

Nystroem Method (NYS): Despite its representational power, kernel ridge regression is known to be slow at training and prediction. To speed up prediction times, which must happen in real time, we implemented the Nystroem method (Williams and Seeger, 2001) which is a scalable kernel approximation technique. We omit a detailed description of this method for sake of brevity.

Table C1. Performance of AS3 on data collected by the DD1 and DM2 sensor across the Jun and Oct deployments. All AS models offered extremely poor calibration performance on all datasets. Note the negative R2 values and the extremely large MAE, MAPE and RMSE values across sites and deployments.

	DD1(Jun)		DM2(Oct)	
	O_3	NO_2	O_3	NO_2
MAE	251.8±0.27	107.7±0.11	354.6±2.3	13.6±1.0
RMSE	253.6±0.27	108.2±0.12	356.0±2.3	22.7±2.2
MAPE	1341.9±120.9	1954.6±51.8	8785.4±4607.4	142.4±6.5
R2	-178.7±3.9	-95.5±3.5	-1617.7±104.6	-0.12±0.0

Appendix C: Detailed Calibration Results

Here we present detailed outcomes of the calibration studies comparing algorithms within certain families e.g. Alphasense, linear, non-parametric etc. The main paper only includes a summary of these results.

C1 Alphasense algorithms

We evaluated the four Alphasense algorithms described in Sect. B1.1 on all datasets. Since there is no training required for these models, we directly applied them to the test data for all the splits. All four algorithms exhibit extremely poor performance across all metrics on all datasets, offering extremely high MAE and low R2 values. Two-sample tests confirmed this by declaring all four AS algorithms as losers when compared to any other algorithm (e.g. LS or KNN). This was true for every split of every dataset. However AS3 was the better among the four algorithms and Table C1 presents various error metrics for this variant on two datasets across the two sites and deployments to illustrate the performance level of these algorithms. With the error scales being so huge, violin plots for these algorithms fail to convey useful information and are omitted.

Previous studies (Lewis and Edwards, 2016; Jiao et al., 2016; Simmhan et al., 2019) corroborate this poor performance of the AS calibration algorithms. It was suggested that this is likely due to changing levels of confounding effects at the study sites (Kumar et al., 2015; Mijling et al., 2018; Simmhan et al., 2019; Chatzidiakou et al., 2019). In particular, Chatzidiakou et al. (2019) reported that the electrochemical sensors lose sensitivity at higher temperatures in field. Models that do take into account RH and T as explicit features were found to perform much better. Given the extremely poor performance of these models, we do not consider them in our analyses anymore.

C2 Parametric Models

Among the linear parametric algorithms LS(MIN), LASSO and LS, we found LS to offer the best performance. Tab C2 shows that the paired Wilcoxon test awarded LS a win over the other two algorithms a majority of the time over the 12 datasets and 10 splits of each dataset (see Table C2). A visual inspection of the distribution of absolute errors offered by the three algorithms

(see Fig. C1) confirm that the larger number of features (augmented as well as Rh and T) offered to the LS algorithm do offer it an advantage.

Table C2. Results of the pairwise Wilcoxon signed rank tests on linear parametric models (see Sect. A2 for a key). LS wins over both LS(MIN) and LASSO a majority of the times and is not defeated by any of them more than marginally often. The overall ranking of the algorithms in terms of performance is indicated to be LS > LS(MIN) > LASSO

	O_3				NO_2		
	LS(MIN)	LASSO	LS		LS(MIN)	LASSO	LS
LS(MIN)	0	0.86	0.08	LS(MIN)	0	0.58	0.05
LASSO	0.01	0	0	LASSO	0.23	0	0.01
LS	0.76	0.93	0	LS	0.65	0.77	0

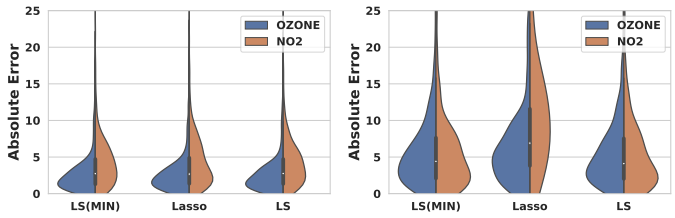

Figure C1. The violin plots on the left and right depict the distribution of absolute errors incurred by various linear parametric calibration models on respectively, the MM5(Jun) and MM5(Oct) datasets. LS offers visibly superior performance on the MM5(Oct) dataset.

C3 Kernel Regression Models

We confirmed that the Nystroem method does indeed offer competitive calibration performance as compared to kernel ridge regression (KRR). In around 47% of the tests, KRR was found to beat the Nystroem method whereas in around 35% of the tests, Nystroem beat KRR. 18% of the tests were inconclusive in declaring a winner. The violin plots for the two algorithms (see Fig. C2) can be used to visually confirm that the algorithms do indeed offer comparable performance. However, as Tab C3 shows, the prediction time offered by the Nystroem method can be more than $4\times$ faster in terms of prediction time than KRR. This highlights the utility of the Nystroem method as an accurate but accelerated approximation for KRR.

C4 *k*-NN and Metric Learning Models

Among the k-NN family of algorithms, the distance weighted k-NN algorithm that uses a learnt metric i.e. KNN-D(ML) was found to offer the best accuracies across all datasets and splits. Table C4 reports the results of the paired Wilcoxon tests comparing all algorithms. In general, KNN-D(ML) was awarded a win over other variants a majority of the time. Some other trends evident from this analysis is that using a learnt metric always helps improve performance (since KNN(ML) wins over

Table C3. Prediction time speedups offered by the Nystroem method over the KRR algorithm on both sites and deployments. Notice that the speedup is generally higher for larger datasets.

			Test time per sample (ms)		
		# Train Samples	KRR	NYS	Speedup
Site D	Jun	23626.5±1299.1	0.39±0.24	0.07±0.042	**4.4**×
	Oct	6682.8±1013.7	0.09±0.08	0.02±0.011	**3.9**×
Site M	Jun	1647.5±229.3	0.14±0.02	0.02±0.007	**5.9**×
	Oct	742.7±173.0	0.009±0.006	0.005±0.002	**1.5**×

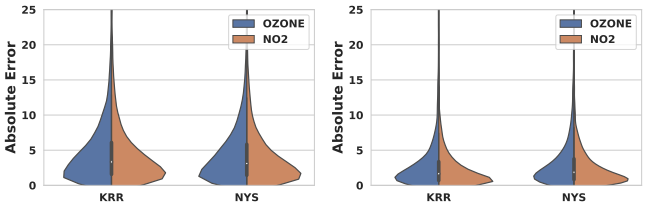

Figure C2. The violin plots on the left and right depict the distribution of absolute errors incurred by KRR and the Nystroem method on respectively, the DD1(Jun) and DD2(Oct) datasets. Both algorithms can be seen to offer comparable performance on both datasets.

KNN at least 69% of the time and KNN-D(ML) wins over KNN-D at least 64% of the time) , as well as that distance-weighing always improves performance (since KNN-D(ML) wins over KNN(ML) at least 78% of the time and KNN-D wins over KNN at least 75% of the time). Tab C5 presents various error metrics for these algorithms on the DD1(Jun) and MM5(Oct) datasets. We avoid presenting a violin plot in this case since the plots are not readily discernible.

Table C4. Results of the pairwise Wilcoxon signed rank tests on k-NN and metric learning models (see Sect. A2 for a key). KNN-D(ML) beats every other algorithm a large fraction of the time and is scarcely ever beaten. The overall ranking of the algorithms is indicated to be KNN-D(ML) > KNN(ML) > KNN-D > NW(ML) > KNN although, as Tab C5 indicates, in terms of error metrics, KNN(ML), KNN-D and NW(ML) are competitive as well.

O_3					
	KNN	KNN-D	NW(ML)	KNN(ML)	KNN-D(ML)
KNN	0	0	0.30	0	0
KNN-D	0.85	0	0.34	0.03	0.01
NW(ML)	0.34	0.28	0	0.18	0.06
KNN(ML)	0.70	0.58	0.56	0	0
KNN-D(ML)	0.81	0.70	0.58	0.87	0

NO_2					
	KNN	KNN-D	NW(ML)	KNN(ML)	KNN-D(ML)
KNN	0	0.01	0.46	0.01	0
KNN-D	0.79	0	0.50	0.03	0
NW(ML)	0.29	0.25	0	0.12	0.03
KNN(ML)	0.76	0.64	0.59	0	0
KNN-D(ML)	0.87	0.76	0.62	0.81	0

Table C5. A comparison of various k-NN and metric learning algorithms on the DD1(Jun) and MM5(Oct) datasets with respect to the MAE and R2 metrics. The best algorithms in terms of mean statistics are highlighted in bold.

O_3				
	DD1(Jun)		MM5(Oct)	
	MAE	R2	MAE	R2
KNN	3.88±0.04	0.909±0.005	2.94±0.21	0.729±0.03
KNN-D	3.82±0.03	0.911±0.004	2.84±0.20	0.744±0.03
NW(ML)	4.23±0.06	0.895±0.005	**2.75±0.22**	**0.751±0.04**
KNN(ML)	3.57±0.05	0.921±0.003	2.87±0.26	0.738±0.04
KNN-D(ML)	**3.52±0.04**	**0.923±0.003**	2.79±0.26	**0.751±0.04**

NO_2				
	DD1(Jun)		MM5(Oct)	
	MAE	R2	MAE	R2
KNN	3.19±0.02	0.757±0.01	3.14±0.19	0.936±0.01
KNN-D	3.13±0.02	0.761±0.01	3.06±0.18	0.940±0.01
NW(ML)	3.49±0.07	0.717±0.02	**2.90±0.27**	**0.943±0.03**
KNN(ML)	2.74±0.06	0.808±0.02	3.02±0.28	0.939±0.03
KNN-D(ML)	**2.67±0.05**	**0.819±0.01**	2.98±0.27	**0.943±0.03**

Competing interests. Author Ronak Sutaria is the CEO of Respirer Living Sciences Pvt. Ltd. which builds and deploys low-cost sensor based air quality monitors with trade-name 'Atmos - Realtime Air Quality' monitoring sensor networks. Ronak Sutaria's involvement was primarily in the development of the air quality sensor monitors and the big data enabled application programming interfaces to access the temporal data but not in the data analysis. Author Brijesh Mishra, subsequent to the work presented in this paper, has joined the Respirer Living Sciences team. The authors declare no other competing interests.

Acknowledgements. This research has been supported under the Research Initiative for Real-time River Water and Air Quality Monitoring program funded by the Department of Science and Technology, Government of India, and Intel® and administered by the Indo-United States Science and Technology Forum (IUSSTF).

References

Caputo, B., Sim, K., Furesjo, F., and Smola, A.: Appearance-based object recognition using SVMs: which kernel should I use?, in: NIPS workshop on Statistical Methods for Computational Experiments in Visual Processing and Computer Vision, 2002.

Chatzidiakou, L., Krause, A., Popoola, O. A., Di Antonio, A., Kellaway, M., Han, Y., Squires, F. A., Wang, T., Zhang, H., Wang, Q., et al.: Characterising low-cost sensors in highly portable platforms to quantify personal exposure in diverse environments, Atmospheric Measurement Techniques, 12, 4643, 2019.

Jiao, W., Hagler, G., Williams, R., Sharpe, R., Brown, R., Garver, D., Judge, R., Caudill, M., Rickard, J., Davis, M., et al.: Community Air Sensor Network (CAIRSENSE) project: evaluation of low-cost sensor performance in a suburban environment in the southeastern United States, Atmospheric Measurement Techniques, 9, 5281–5292, 2016.

Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., Bell, M., Norford, L., and Britter, R.: The rise of low-cost sensing for managing air pollution in cities, Environment International, 75, 199–205, 2015.

Lewis, A. and Edwards, P.: Validate personal air-pollution sensors, Nature, 535, 29–31, 2016.

Mann, H. B. and Whitney, D. R.: On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other, Annals of Mathematical Statistics, 18, 50–60, 1947.

Mijling, B., Jiang, Q., de Jonge, D., and Bocconi, S.: Field calibration of electrochemical NO2 sensors in a citizen science context, Atmospheric Measurement Techniques, 11, 1297–1312, 2018.

Shapiro, S. S. and Wilk, M.: An analysis of variance test for normality (complete samples), Biometrika, 52, 591–611, 1965.

Simmhan, Y., Nair, S., Monga, S., Sahu, R., Dixit, K., Sutaria, R., Mishra, B., Sharma, A., SVR, A., Hegde, M., Zele, R., and Tripathi, S. N.: SATVAM: Toward an IoT Cyber-infrastructure for Low-cost Urban Air Quality Monitoring, in: 15th IEEE International Conference on e-Science (eScience 2019), 2019.

Tibshirani, R.: Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological), 58, 267–288, 1996.

Wilcoxon, F.: Individual Comparisons by Ranking Methods, Biometrics Bulletin, 1, 80–83, 1945.

Williams, C. and Seeger, M.: Using the Nystroem method to speed up kernel machines, in: Neural Information Processing Systems (NIPS), 2001.