Calibrating Low-Cost Sensors by a Two-Phase Learning Approach for Urban Air Quality Measurement

YUXIANG LIN, WEI DONG, and YUAN CHEN, Zhejiang University, China

Urban air quality information, e.g., PM2.5 concentration, is of great importance to both the government and society. Recently, there is a growing interest in developing low-cost sensors, installed on moving vehicles, for fine-grained air quality measurement. However, low-cost mobile sensors typically suffer from low accuracy and thus need careful calibration to preserve a high measurement quality. In this paper, we propose a two-phase data calibration method consisting of a linear part and a nonlinear part. We use MLS (multiple least square) to train the linear part, and use RF (random forest) to train the nonlinear part. We propose an automatic feature selection algorithm based on AIC (Akaike information criterion) for the linear model, which helps avoid overfitting due to the inclusion of inappropriate features. We evaluate our method extensively. Results show that our method outperforms existing approaches, achieving an overall accuracy improvement of 16.4% in terms of PM2.5 levels compared with state-of-the-art approach.

CCS Concepts: • **Human-centered computing** → **Mobile devices**; • **Computer systems organization** → *Embedded and cyber-physical systems*; **Sensor networks**;

Additional Key Words and Phrases: Sensor calibration, Low-cost sensors, Mobile sensor network, Air quality

ACM Reference Format:
Yuxiang Lin, Wei Dong, and Yuan Chen. 2018. Calibrating Low-Cost Sensors by a Two-Phase Learning Approach for Urban Air Quality Measurement. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 18 (March 2018), 18 pages. https://doi.org/10.1145/3191750

1 INTRODUCTION

The recent years have witnessed the severe degradation of air quality in developing countries such as China and other low-income countries. Particulate matter with a 2.5 micrometer diameter, known as PM2.5, can cause cardiopulmonary disease, lung cancer and acute respiratory infection, according to the Journal of Toxicology and Environmental Health [14]. Speaking at the opening of the national people's congress in Beijing on March 2017, the Chinese Premier promises to intensify battle against air pollution. Effective air quality monitoring is a key step to tackle air pollution. The disclosure of fine-grained air quality data allows the public to directly supervise the emissions of air pollutants in their areas.

This work is supported by the National Science Foundation of China (No. 61772465, 61472360), Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Zhejiang Provincial Key Research and Development Program (No. 2017C02044), and the Fundamental Research Funds for the Central Universities (No. 2017FZA5013).
Author's addresses: Yuxiang Lin, Wei Dong, and Yuan Chen, College of Computer Science, Zhejiang University, Hangzhou, China. Emails: {linyx, dongw, cheny}@emnets.org.
Wei Dong is the corresponding author.
Authors' address: Yuxiang Lin; Wei Dong; Yuan Chen, Zhejiang University, Zetong Building, Yuquan Campus, College of Computer Science, Hangzhou, Zhejiang, 310027, China, {linyx,dongw,cheny}@emnets.org.

2474-9567/2018/3-ART18 $15.00
https://doi.org/10.1145/3191750

The traditional approach for measuring air quality relies on stationary stations which are expensive to build and maintain [7]. The high costs limit their dense deployments in cities. The recent advances in embedded technologies and wireless technologies have enabled yet another approach relying on low-cost sensors. These low-cost sensors, deployed on moving vehicles, can provide unprecedented opportunities for spatially fine-grained urban air quality monitoring.

The quality of collected data is very important for many applications. It has been widely reported that low-cost sensors suffer from limited accuracy, high instability, and sensor drift [6, 18, 21]. Hence, low-cost sensors need to be carefully calibrated to preserve a good data quality.

The topic of sensor calibration has been intensively investigated in the past years. The existing studies reveal that (1) Many low-cost sensors show a close to linear dependence between sensor measurements (e.g., raw PM2.5 measurement) and the ground truth phenomenon signal (e.g., the real PM2.5 value) [24]. (2) Besides the real measurements of interest, other sensor readings (e.g., temperature, humidity) could be cross-sensitive to the sensor measurement of interest. Using multi-parameters can significantly improve the calibration accuracy [8, 15]. (3) There exist factors, e.g. noise, which show unknown interaction with the sensor measurement of interest [6].

Most existing calibration methods are based on linear models. However, it cannot model complex interactions among different measurements. While it is possible to directly use machine-learning models such as ANN (Artificial Neural Networks) or RF (Random Forest) for data calibration, such models may lose precision in some cases. In the extreme case of dataset showing perfect linear degree, RF is inherently less accurate than linear models due to the discretization operations in RF.

To address these issues, we propose a general data calibration method based on two-phase learning, assuming the calibration model consisting of a linear part and a nonlinear part. In the first phase, we employ MLS (Multiple Least Square) to train the linear part. Since the inclusion of inappropriate features could result in reduced accuracy, we propose a systematic approach for automatic selection of multiple features. We select the most appropriate subset of features based on AIC (Akaike Information Criterion) [2, 23] which is a general metric considering goodness of fit as well as a penalty which discourages overfitting caused by too many features. A greedy algorithm is devised to select one feature in each step, with the goal to minimize the total AIC. In the second phase, we employ RF to train the residual error of the linear part. RF is an ensemble learning method for regression, and it is operated by constructing a multitude of decision trees at training time. As such it can model complex nonlinear relationship among multiple features.

A very good feature of our two-part model lies in its adaptiveness. For measurements showing a strong linear relationship, the first part model can already reduce the error significantly while the second part plays a less important role in improving the final calibration accuracy. For measurements showing non-linear complex relationships, almost no features will be selected in the first part model and the second part plays a more important role improving the final calibration accuracy.

We implement and evaluate our approach in the Mosaic, a mobile air quality monitoring system [7, 11] in which PM2.5 concentration is the measurement of interest. The low-cost sensors can collect a range of measurements including PM2.5, temperature, humidity, acceleration, angular velocity. We conduct experiments by deploying these low-cost sensors together with the commercially available Dylos sensor (considered as reference) on city buses. Results show that (1) the second phase is important for further reducing the calibration error. Our two-part model achieves a RMSE (Root Mean Square Error) of 14.8 while the linear part achieves a RMSE of 25.2. (2) our method outperforms RF even when RF is carefully optimized (with a RMSE of 15.8). (3) our method outperforms existing approaches in terms of PM2.5 levels, resulting in 6.1%-133% precision improvement compared with state-of-the-art approaches.

The contributions of this paper can be summarized as follows.

- We propose a data calibration method based on two-phase learning. Our method can well capture both the primary linear relationship and the complex unknown nonlinear relationship for low-cost sensors.
- We propose an automatic feature selection algorithm based on AIC for the linear model, which helps avoid overfitting due to the inclusion of inappropriate features.
- We evaluate our method extensively. Results show that our method outperforms existing approaches, resulting in 6.1%-133% precision improvement in terms of PM2.5 levels compared with state-of-the-art approaches.

The rest of this paper is structured as follows. Section 2 introduces the related work. Section 3 introduces the background and the necessary definitions. Section 4 presents our two-phase calibration method in detail. Section 5 describes the evaluation results. Section 6 discuss some important practical issues, and finally, Section 7 concludes this paper and gives future research directions.

2 RELATED WORK

Existing sensor calibration methods can roughly be divided into one-parameter regression and multi-parameter regress. We also introduce sensor data inference and its relevance to sensor calibration.

One-parameter Regression. It aims to determine a calibration function which defines the relationship between the reference signal and *one* most relevant measurement. There is much work devoted to one-parameter regression. Xiang *et al.* [28] propose a sensor random drift model and develop a collaborative calibration technique to automatically compensate for sensor drift error. Saukh *et al.* [24] exploit the linear relationship between raw sensor data and reference data in PM2.5 sensor calibration. They find that the calibration error will accumulate with OLS (Ordinary Least Squares) and propose an improved linear calibration algorithm using GMR (Geometric Mean Regression).

However, the target measurement can usually be affected by multiple factors. For example, PM2.5 concentration can be affected by various environmental factors [15, 17]. One-parameter regression cannot exploit the underlying relationship between these factors and the reference signal, resulting in poor calibration accuracy in practice.

Multi-parameter Regression. It exploits *multiple* relevant features to collaboratively calibrate the measurement of interest. ANN [6, 7, 27] and multiple linear regression are two typical approaches for multi-parameter regression.

Cheng *et al.* [6] propose an ANN based calibration model to calibrate a low-cost sensor (i.e. PPD42NS). In this approach, the widely-used BP (back-propagation) network is chosen and the neural network is capable of well fitting an arbitrary function [7]. The neural network takes the readings of PPD42NS with co-located temperature and humidity readings as inputs and the ground truth value (given by Dylos) as output. However, the ANN-based approach is prone to go to local minima with finite data sets [1, 5], i.e., it tends to fit the trained data well while has poor performance in predicting the new unseen data. To address this problem, Dong *et al.* [7] propose an adaptive calibration method based on both ANN and SVMs (support vector machines). However, such models may lose precision in some cases. In the extreme case of measurements showing perfect linear degree, they are inherently less accurate than linear models since discretization operations in these models.

Maag *et al.* [15] propose an algorithm based on MLS to compute its calibration coefficients for improving the accuracy of low-cost sensors. Their MLS-based calibration algorithm performs well for ozone (O_3) and nitrogen dioxide (NO_2) data calibration. While MLS works well for the gas sensors, it performs poorly for particulate sensors (e.g., PM2.5) because particulate sensor measurements can be affected by multiple environmental factors. Some of these factors exhibits complex non-linear relationship to the

target measurement [8, 15]. To address this issue, Fang *et al.* [8] propose a stepwise regression method which not only considers the linear impacts of multiple parameters but also considers non-linear impacts described by two-way interaction terms. While two-way interactions terms can be useful in modeling one *particular* non-linear relationship, they are incapable of handling more *general* complex relationships. In our approach, we use the RF model to learn the complex relationship between all available features and the residual error. RF can model arbitrary nonlinear relationship among multiple features.

In mobile scenario, sensor calibration can be performed by exploiting rendezvous between the reference sensors and mobile sensors. A calibrated mobile sensor can also be used to calibrate an uncalibrated mobile sensor when they meet, referred to as multihop calibration [16, 24]. In previous work [10], we considered the problem of how to deploy the reference sensors to ensure that all sensors in the network are (k-hop-)calibratable. Efficient deployment algorithms have been proposed. Recently, Maag *et al.* [16] tackle the *bias towards zero* problem, i.e., the error will accumulate over multiple hops with MLS [9]. They propose SCAN, a new constrained least-squares regression method based on MLS to reduce error accumulation. Our two-phase calibration method can also be used in the multihop calibration scenario. Compared with MLS, we mitigate the bias towards zero problem by eliminating the accumulated error with RF in the second phase.

Sensor Data Inference. The difference between calibration and inference lies in whether there exists the measurement of interest (no matter it is accurate or not). Sensor data inference usually utilizes a lot of data sources to infer the target measurement indirectly. There are existing works for inferring urban air quality information [4, 26, 29].

Zheng *et al.* [29] utilize spatial and temporal correlations of air qualities between two areas to estimate the air quality based on a co-training framework *U-Air*. Chen *et al.* [4] propose a spatially fine-grained urban air quality estimation method using ensemble semi-supervised learning with ensemble pruning, based on many kinds of urban data including meteorological data, POIs (points of interests).

Sensor data inference can be useful because it estimates the sensor measurements at any location and time without direct sensor measurement. Our approach can help sensor data inference by feeding the inference algorithm with more accurate measurements.

Summary. Compared with one-parameter regression, our approach utilizes multiple features, hence can significantly improve data calibration accuracy. Existing multi-parameter regression methods are either based on multiple linear regression or machine learning (e.g., ANN, SVM). Linear models assume that different features are independent and cannot model the dependency between features while ANN based models may lose precision in some cases (because of overfitting). Our method addresses these issues by proposing a novel two-phase learning approach which not only considers the primary linear relationship but also captures complex non-linear relationships. Another difference from all existing works is that we carefully address the overfitting problem in the first part model by proposing an *automatic* feature selection algorithm based on AIC.

3 BACKGROUND AND NOTATIONS

This section introduces the necessary background and notations.

3.1 Mosaic

Mosaic is a recent project using mobile sensor nodes equipped on city buses for fine-grained urban air quality measurements. The commercially available low-cost PM2.5 sensors typically suffer from low measurement quality. It is thus an important issue that our calibration method intends to address. In the following paragraphs, we briefly introduce the sensor nodes used in the Mosaic project.

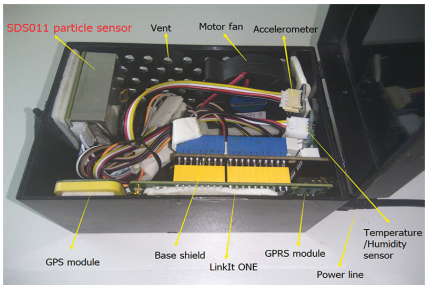


Fig. 1. Mosaic node

Dylos Node. Dylos DC1700 [20] is a commercially available air quality monitor that costs more than 430 dollars. It can measure particle matters with diameter in the range of $0.5\mu m$ to $2.5\mu m$. Besides, Dylos has a fan to provide stable air flow which is important for measurement accuracy. As such, The PM2.5 measurements obtained from Dylos can be considered as a reference. We use Dylos node to calibrate other low-cost sensor nodes (Mosaic nodes) for large-scale deployment in cities.

Mosaic Node. A Mosaic Node is a low-cost air quality sensing node that is specially designed for moving vehicles [7, 11]. A Mosaic node is based on the LinkIt ONE open electronics platform. It consists of various sensors including PM2.5 sensor, temperature and humidity sensor, accelerometer, and GPS (global position system). In addition, an 8GB SD card is used for data storage and a GSM/GPRS module for data communication. Mosaic nodes primarily use car power supply. With a car plug, the car power supply can provide 5V/1A DC.

The PM2.5 sensor is the primary sensor used for air quality measurement. We systematically investigate many commercially available low-cost PM2.5 sensors. Eventually, we have selected SDS011 [25] which costs about 23 dollars. Figure 1 shows a picture for a Mosaic node. The collected measurements are transmitted using GPRS to a cloud server where the measurements are processed and calibrated with our two-phase method.

3.2 Notations

We consider a mobile sensor network scenario with low-cost sensors deployed on moving buses. These low-cost sensors (e.g., Mosaic node) can produce low-precision PM2.5 measurements and other easily fetched measurements such as temperature and humidity. In order to calibrate the low-precision PM2.5 measurements, a high-precision reference sensor (e.g., Dylos) can be deployed together with a low-cost sensor node in the training phase. We assume that the measurement data is periodically sampled.

We introduce the following notations:

- l: we assume the time of training data is divided into l slots. In each slot j, there is a measurement set sampled from the low-cost sensor node and a measurement of interest (i.e. PM2.5) sampled from reference sensor.
- n: we assume the low-cost sensor node can produce n different kinds of features (e.g. temperature, humidity, speed, etc.) including the measurement of interest (i.e. PM2.5).
- $m_i[j]$, $\mathbf{m}_i$: we use $m_i[j]$ to denote the i-th measurement produced by a low-cost sensor node in the j-th time slot. Note that we use i $(1 \leq i \leq n)$ to differentiate different kinds of measurement. Specifically, we use $m_1[j]$ to denote the measurement of interest (PM2.5) and $m_2[j]$ to $m_n[j]$ to denote other measurements, e.g. temperate and humidity. $\mathbf{m}_i$ denotes the vector of all i-th measurements.

- $\mathbf{m}[j]$: it is used to denote a vector of different measurements in the j-th time slot, i.e., $\mathbf{m}[j] = (1, m_1[j], ..., m_n[j])$. Note that the added 1 is used to simplify the expression of the calibration model.
- $\mathbf{M}$: it is the training matrix consisting of l rows and $n+1$ columns. i.e. $\mathbf{M} = [\mathbf{m}[0]; \ldots; \mathbf{m}[l]]$. Each element x_{ij} denotes the j-th measurement in the i-th time slot.
- $\mathbf{M}_S$: Note that not all n measurements are used for calibration: some features have weak linear correlation with the measurement of interest. Including those uncorrelated features may cause the overfitting problem. We use S to denote the feature subset used for calibration. $\mathbf{M}_S$ denotes the matrix selected with features in S and it is a subset of $\mathbf{M}$. i.e. $\mathbf{M}_S = [\mathbf{1}, \mathbf{m}_{i_1}, ..., \mathbf{m}_{i_k}]$ where $i_1, ...i_k \in [1, n]$ and $k \leq n$.
- $m_r[j]$, $\mathbf{m}_r$: we use $m_r[j]$ to denote the measurement of interest (i.e. PM2.5) produced by the reference sensor in the j-th time slot. $\mathbf{m}_r$ denotes the vector of all reference measurements.
- $\hat{m}[j]$, $\hat{\mathbf{m}}$: we use $\hat{m}[i]$ to denote the calibrated measurement of interest in the j-th time slot (for a particular node). $\hat{\mathbf{m}}$ denotes the vector of all calibrated measurements.

A calibration model establishes the relationship from the measurements of low-cost sensors to the calibrated measurement of interest. A good calibration model is able to minimizes the difference between the calibrated measurements $\hat{\mathbf{m}}$ and the reference measurements $\mathbf{m}_r$. We use the multi-parameter calibration model consisting of a linear part and a nonlinear part (we omit the j-th time slot when it is not required to simplify the notation):

$$\begin{aligned} \hat{m} &= linearpart(\mathbf{m}) + nonlinearpart(\mathbf{m}) \\ &= \beta_0 + \beta_1 m_1 + \beta_2 m_2 + \beta_3 m_3 + \cdots + \beta_n m_n + f(\mathbf{m}) \\ &= \mathbf{m}\boldsymbol{\beta} + f(\mathbf{m}) \end{aligned} \tag{1}$$

Note that $\beta_0, \beta_1, \ldots, \beta_n$ are the calibration coefficients consisting of an intercept and n different kinds of features in the linear part. We use $\boldsymbol{\beta}$ to denote the column vector consisting of these coefficients, i.e. $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_n)^T$. $f(\mathbf{m})$ represents an arbitrary nonlinear function with all the features as input. The calibration problem is how to determine the values of $\beta_0, ..., \beta_n$ as well as the function f, given sufficient training data (i.e., $\mathbf{m}$ and m_r pairs). When this calibration model is determined, the raw sensor measurements vector $\mathbf{m}$ can be transformed to a calibrated measurement of interest $\hat{m}$. In order to address this problem, we propose a two-phase learning method: the first phase is used to determine the linear coefficients while the second phase is used to determine the complex nonlinear function.

4 TWO-PHASE CALIBRATION METHOD

4.1 Overview

We propose a two-phase learning method for determining the linear part and nonlinear part of the calibration model. Figure 2 shows an overview of our approach.

In the first phase, the training data ($\mathbf{m}$ and m_r pairs) is fed into our approach. Note that not all n features are used for calibration: some features may have little linear correlation with the measurement of interest. Including those features in the linear model will easily cause the overfitting problem. Hence, we devise a feature subset selection algorithm to select the most appropriate features used in the linear part, denoted as S (and ignore the other features, i.e., their coefficients are set to zero). Next, in order to determine the non-zero coefficients of the linear model, we employ the classical MLS algorithm. After the first phase, the linear model can be accurately determined.

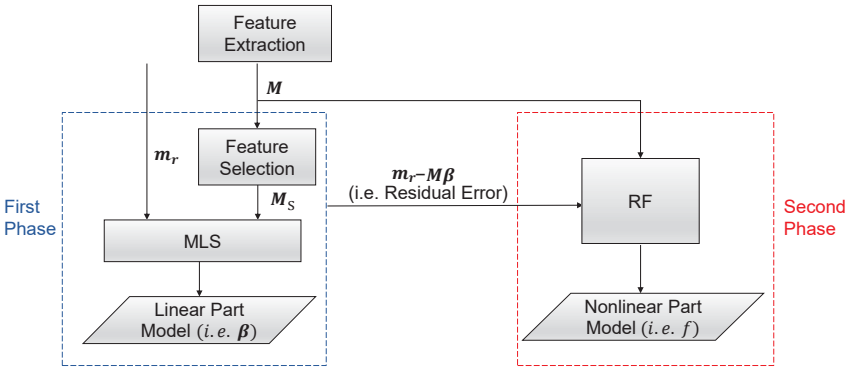


Fig. 2. An overview of the two-phase learning method.

In the second phase, we use RF to learn the residual of the linear model. Both the extracted features and the residual of the linear model, i.e., $\mathbf{e}_{res}$, are fed into our RF construction algorithm. Note that we put all the features into the second phase. This is because the common function of these factors (including factors with strong linear correlation) has complex nonlinear influence on the concentration of PM2.5 [4]. The resulting model is used to approximate the complex nonlinear function. It is worth noting that our RF method can model much more complex relationships compared with existing approach with two-way interaction terms [8]. In addition, RF does not suffer from the overfitting problem compared with decision trees.

After the two phases, we can determine the calibration model consisting both the linear part and nonlinear part. In the following subsections, we first introduce how we extract various features including the measurement of interest (i.e. PM2.5). We then describe the two phases in detail.

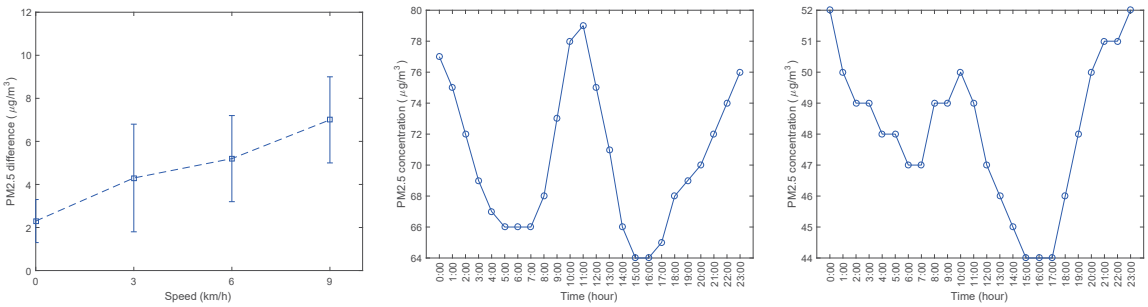
4.2 Feature Extraction

We need to construct and extract features relevant to PM2.5 measurement. These features need to be carefully processed. In our current work, we consider five features including raw PM2.5 measurement, speed, temperature, humidity, and time. In the future, we will include more meteorological measurements, such as NO_2, O_3, NO.

4.2.1 PM2.5 Measurement. There could be extra noises in the measurements of low-cost PM2.5 sensor [6]. To reduce the impact of noise, we employ a moving average method to smooth out short-term fluctuations. We use a central moving average filter [3] which uses data equally spaced on either side of the point in the series where the mean is calculated. The window length l_r is key parameter which determines the tradeoff between smoothness and efficiency. A larger l_r will produce a smoother estimated value of the raw measurements but suffer from higher computation overhead. We set $l_r = 20$ in this paper, which means the time window is fixed at 20 minutes in our work.

4.2.2 Speed. Speed is an important feature for our mobile sensing scenario since it influences the raw PM2.5 measurements.

The underlying reason is explained as follows. Low-cost PM2.5 sensors (e.g., SDS011) typically use a light-scattering method to measure the amount of particles per unit volume. In these sensors, a beam of light from a LED is focused with lens to the sensing point. The particles passing through the sensing point scatter the light which is then sensed by a light sensor. The amount of particles can be inferred from the amount of light since the light sensor can receive more scattered light when there are more particles



(a) (b) (c)

Fig. 3. Impact of speed and time. (a) Impact of speed. (b) PM2.5 concentration in one day from measurements of one month (from January 1, 2017 to February 1, 2017) (c) PM2.5 concentration in one day from measurements of one year (from January 1, 2016 to December 31, 2016).

at the sensing point. One important issue of this light-scattering method is how to stably control the airflow carrying the particles, so that the particle amount can be accurately measured by the readings of the light sensor. While the low-cost sensor has built-in mechanisms to generate airflow (e.g. use resistors to generate heat), the airflow can be significantly impacted by the speed when the sensor is installed on moving buses.

To confirm this, we conduct experiments to see the impact of speed. We place three SDS011 sensors on a rotary table with their distances to the table center being 5cm, 10cm and 15cm, respectively. With the uniform circular motion of rotary table, the speeds of three sensors can be measured as 3km/h, 6km/h and 9km/h, respectively. The Dylos node (generating reference signal) is placed near the rotary table. The sampling intervals of both Dylos node and SDS011 sensors in the rotary table experiments are set as 60s. We repeat the rotary table experiments for 5 times (one hour each time). Figure 3(a) shows the impact of speed on the difference of raw PM2.5 measurement and the reference signal. We see that this difference increases when the speed increases.

We calculate the instant speed from direct measurements from the GPS sensor (containing both location and time information) and acceleration sensor (both sensors exist in the Mosaic node). The speed calculation problem is a classical Kalman filter problem and standard solutions can be employed to solve this problem.

4.2.3 Time. PM2.5 concentration varies during one day. We collected the PM2.5 measurements from an air quality measurement station in Hangzhou to investigate the relationship between time and PM2.5 concentration. Figure 3(b) and Figure 3(c) show the PM2.5 concentration averaged in hours, using the data of one month and one year, respectively. We see that there are clear change patterns during one day. There are two peaks for one day, i.e., at $11am$ and midnight. The PM2.5 concentration is the lowest at $3pm$ since the pollutants spread faster with the increase of temperature.

Adding time as a feature may introduce *temporal bias*, which means the calibration accuracy may decrease when the current change differs from the historical patterns. To mitigate this problem, we can retrain the calibration model periodically, e.g., half a year, to reduce the negative impact. In our current work, incorporating time as a feature indeed helps achieve a higher calibration accuracy since the variation pattern changes gradually in open urban environment. In other circumstances which require real-time warning, we can discard this feature.

Algorithm 1 Multiple linear regression

Input: Feature subset S; Training matrix $\mathbf{M}$; Reference measurements $\mathbf{m}_r$
Output: Calibration coefficients $\boldsymbol{\beta}_S$
1: **procedure** ESTIMATE($S, \mathbf{M}, \mathbf{m}_r$)
2: Generate a new matrix $\mathbf{M}_S$ by selecting the first column and all the other columns $\mathbf{m}_i$ where $i \in S$ from the original matrix $\mathbf{M}$.
3: $\boldsymbol{\beta}_S = (\mathbf{M}_S^T \mathbf{M}_S)^{-1} \mathbf{M}_S^T \mathbf{m}_r$
4: return $\boldsymbol{\beta}_S$

4.3 First Phase

In the first phase, our approach selects the most appropriate features (used in the linear model) and identifies the corresponding linear coefficients. It is important that we select the most appropriate subset from all available features. There is a tradeoff in determining the number of features. Including more available features will improve the calibration accuracy for the training data, but will also likely degrade the overall accuracy because of overfitting. On the other hand, if the number of features is small, the accuracy of the linear part will be unsatisfactory, degrading the overall accuracy if the measurement has close to linear relationship to other cross-sensitive measurements.

The following code snippet shows the procedures for the first phase.

1. $S = select(\{1, 2, ..., n\}, \mathbf{M}, \mathbf{m}_r)$;
2. $\boldsymbol{\beta}_S = estimate(S, \mathbf{M}, \mathbf{m}_r)$;

The procedure *select*() is used to determine the most appropriate feature subset S from all available features. For example, $S = \{1, 3\}$ indicates that the first and third features are used in the linear model. The procedure *estimate*() is used to determine the linear coefficients $\boldsymbol{\beta}$ in the first phase. We first describe the second step (estimate coefficients with MLS) since it is also used in every iteration of the first step (feature selection) until we find the optimal feature subset S.

4.3.1 MLS. We consider the problem of how to estimate the linear coefficients *given* the selected features, i.e., when S is fixed. Before introducing the concrete algorithm, it is necessary to introduce the following notations.

- $\boldsymbol{\beta}_S$. It is the column vector including β_0 and all β_i, $i \in S$. The other coefficients are set to zero. When S equals to the full feature set, i.e., $S = \{1, 2, ..., n\}$, $\boldsymbol{\beta}_S = \boldsymbol{\beta}$.
- $\mathbf{m}_S[j]$. Similarly, it is the vector consisting of the first column and all columns m_i where $i \in S$ in the original vector $\mathbf{m}[j]$.

Algorithm 1 shows how we build the linear model. The procedure *estimate*() is used to determine the coefficients of β_i ($i = 0$ or $i \in S$) given the feature set S and the training data. It uses the MLS regression method which aims to minimize the sum of squared residuals.

With the obtained regression coefficients $\boldsymbol{\beta}_S$, we introduce how to predict the value $\hat{m}_S$ for the linear part, it can be calculated by taking dot products of $\mathbf{m}_S$ and $\boldsymbol{\beta}_S$: $\hat{m}_S[j] = \mathbf{m}_S[j] \cdot \boldsymbol{\beta}_S$.

4.3.2 Feature Selection. If we use all features in the linear model, it is possible that inappropriate features are included and the results of calibration are biased. Therefore we need a systematic approach to select the optimal feature subset S.

While previous approaches used sum of squared error and F-test [8] to measure goodness of model, we employ the AIC (Akaike information criterion) since it is derived rigorously from information theory. AIC is an estimator of the relative quality of statistical models for a given set of data. AIC offers a relative estimate of the information lost when a given model is used to represent the process that generates the

Algorithm 2 A greedy algorithm for feature subset selection

Input: Full feature set $F = \{1, 2, ..., n\}$, Training matrix $\mathbf{M}$, Reference measurements $\mathbf{m}_r$
Output: An appropriate feature subset $S \subseteq F$

procedure SELECT(F,$\mathbf{M}$,$\mathbf{m}_r$)
  $S_i \leftarrow \phi$, $AIC_{\min} \leftarrow \infty$
  for $j = 1, 2, \ldots, n$ **do**
    $sub = -1$
    for $k \in F - S_i$ **do**
      $AIC_{local} = AIC(S_i \cup \{j\}, \mathbf{M}, \mathbf{m}_r)$
      if $AIC_{local} < AIC_{\min}$ **then**
        $AIC_{\min} = AIC_{local}$, $sub = j$
    if $sub! = -1$ **then**
      S_i=$S_i \cup \{sub\}$
    else
      break
  return S

procedure AIC($S, \mathbf{M}, \mathbf{m}_r$)
  $RSS = \sum_{j=1}^{l}(m_r[j] - \hat{m}_S[j])^2$
  $AIC = 2(|S| + 1) + l \cdot \ln(RSS/l)$
  return AIC

data. In addition, it considers both the goodness of fit of the model and the complexity of the model. The goodness of fit is typically measured as the likelihood of the features being corrected based on the training data. The complexity of the model is typically measured as the number of features used in the model. Since more features would result in model overfitting, AIC adds a penalty if more features are used. Since AIC is general metric for any statistical models, it can be applied to our first part model which is a linear model. In our case, it not only considers linearity of features (goodness of fit), but also considers a penalty which discourages overfitting caused by too many features. AIC value roughly indicates the number of features minus the likelihood of the overall model. Therefore, the smaller the AIC value the better the model.

For least squares fitting [2, 23], $AIC = 2k + n\ln(RSS/n) + C$, where k is the number of features used in the model, RSS represents residual sum of squares and C is a constant (can be ignored since it is not important for model comparisons).

We would like to select the feature subset with the smallest AIC value. However, enumerating all possible subset may not always feasible due to exponential computational complexity. To address this issue, we propose a greedy algorithm shown in Algorithm 2 to search for the smallest AIC value with $O(n^2)$ complexity. This greedy algorithm is essentially the step-wise regression method with forward selection, which involves starting with no parameters in the model, testing the addition of each parameter using AIC, adding the parameter (if any) whose inclusion decreases the minimum seen AIC value, and repeating this process until none improves the model in terms of AIC. In reality, we enhance this basic algorithm in two aspects. First, the algorithm also looks for results by backward elimination, i.e., starting with all the parameters in the model, testing the subtraction of each parameter. Second, the algorithm further splits the training data for feature selection and re-evaluation to avoid over-optimistic results (because the data is not necessarily i.i.d.). In the current implementation, the algorithm splits training data (including the training matrix and the corresponding reference signals) into 10 parts. In each iteration i ($1 \leq i \leq 10$),

the algorithm uses 9 parts (except the i-th part) for feature selection and the i-th part to re-evaluate the AIC value with the selected features. The algorithm selects the subset of features with the minimum re-evaluated AIC value.

4.4 Second Phase

In the second phase, our approach tries to use machine learning based approach to learn the complex nonlinear function for low-cost sensors. We first calculate the residual errors of the first phase, i.e. $\mathbf{e}_{\text{res}}$, $\mathbf{e}_{\text{res}} = (e_1, ..., e_l)$ where $e_j = m_r[j] - \hat{m}_S[j]$ ($\hat{m}_S[j]$ is obtained by the first phase). Then we use the RF model to learn the complex relationship between all available features and the residual error.

In our approach, RF is implemented with the toolbox based on scikit-learn framework [12]. There are K iterations for generating K decision trees. In our current design, we set $K = 60$ based on the experimental results. In each iteration, the algorithm randomly samples l training data (the same size as the original data set) with replacement from the original data set. Based on the randomly sampled training data, we employ a decision tree model to construct a decision tree (i.e. the commonly-used classification and regression tree (CART) for RF), denoted as T_i. The value for the nonlinear part $\hat{m}_{RF}$ is simply the mean of the predictions of the individual decision tree:

$$\hat{m}_{RF}[j] = \frac{1}{K} \sum_{i=1}^{K} T_i(\mathbf{m}[j]) \tag{2}$$

With the available prediction of the first phase $\hat{m}_S$ and the second phase $\hat{m}_{RF}$, the final calibrated measurement can be calculated as:

$$\hat{m}[j] = \hat{m}_S[j] + \hat{m}_{RF}[j] \tag{3}$$

5 EVALUATION

In this section, we present a detailed evaluation on our method.

5.1 Experiment Setup and Dataset

We use the low-cost sensor measurements from the Mosaic project [7] deployed in two cites in China, i.e. Hangzhou and Ningbo. In all experiments, we use measurements of the high-precision Dylos nodes equipped on city buses as references (i.e. ground truth). Each Mosaic node collects the raw sensor data to the cloud where further data processing is performed, generating five features including reconstructed PM2.5 measurements, temperature, humidity, speed, and time (in the unit of hour). The sampling intervals of both Dylos and Mosaic node are 60 seconds.

The Hangzhou dataset[1]contains measurements from 8 low-cost sensors installed on buses for 37 days (from December 1, 2014 to December 7, 2014 and from July 1, 2017 to August 1, 2017). The Ningbo dataset contains measurements from 2 low-cost sensors installed road cleaning vehicles for 5 months. We have collected totally 10838 valid measurements in Ningbo and 3360 valid measurements in Hangzhou. The number of measurements is lower than expected due to two reasons. First, the data is collected only during working hours. It is possible that the drivers stopped the cars and powered off the node unexpectedly. Second, the data may contain invalid measurements with missing phenomena data. We use the Ningbo dataset for training and the Hangzhou dataset is only used for validation (due to smaller number of measurements). For the Ningbo dataset, we use the first 70% data (ordered by time) for training and the remaining 30% data for validation. Each evaluation is executed 100 times with a randomly re-sampled RF. In our evaluation, both RMSE and RE (relative error) are used for performance evaluation.

[1]The dataset is publicly available at: http://47.98.46.246/Mosaic/dataset.tar.

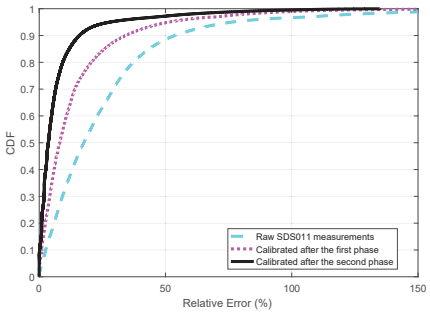


Fig. 4. Relative error of the raw PM2.5 measurements and the calibrated measurements after the first phase and second phase.

Table 1. RMSEs of the raw measurements and the calibrated data after applying our method.

	RMSE
Raw data	57.239
After the first phase	25.169
After the second phase	14.762 ± 0.051

5.2 Overall Performance

RE of PM2.5 Concentration. Figure 4 shows the relative error of the raw PM2.5 measurements and the calibrated measurements after the first phase and second phase. We can see that the raw data has a low measurement quality (compared with the reference signal), i.e., 58% of the data has a relative error larger than 15%, prohibiting its direct use in practical scenarios. Our two-phase method effectively reduces the error: 71% data has an error less than 15% after the first phase, and 90% data has an error less than 15% after the second phase.

RMSE of PM2.5 Concentration. Table 1 shows the RMSEs of the raw measurements and the calibrated data after applying our method. We see that the RMSE is reduced from 57.2 to 25.2 after the first phase, and the RMSE is further reduced to 14.8 after the second phase.

It is worth noting that we do not show 95% confidence intervals for the first phase—MLS (and also Stepwise [8]) since the training set and testing set are fixed for every execution. However, even with the same training set, the trained machine-learning models such as ANN and RF can be different from each execution due to random operations in these models. Hence, we show 95% confidence intervals for our two-phase method (as well as ANN and RF).

Accuracy of PM2.5 Levels. For end users, dividing the range of PM2.5 values into discrete levels may be more useful, as indicated by the official definition of PM2.5 levels [7]. According to the official definition of PM2.5 levels [22], the PM2.5 concentration is divided into 6 discrete levels. The large value of PM2.5 level indicates poor air quality. Since the official definition has levels spanning wide range on some levels, we define PM2.5 levels in the same way as [6] (level 1-4 is the same as the official one; 5 and 6 equal to official level 5; 7 and 8 equal to official level 6).

Table 2 shows the confusion matrix for our method. The overall accuracy is 0.794. As a comparison, the overall accuracy of the raw measurements is only 0.486 while the overall accuracy of Stepwise [8]—the most recent work, on our dataset, is 0.682. The result shows an improvement of 66% over the raw measurement and 16.4% over Stepwise. We have also found that our method consistently improves the precisions of all 8 levels by 6.1%-133%, compared with Stepwise. In addition, the recalls also increase for all 8 levels.

Table 2. Confusion matrix for our method.

Ground Truth	Predictions									
	Level1	Level2	Level3	Level4	Level5	Level6	Level7	Level8		
Level1	**27**	17	0	0	0	0	0	0	0.61	Recall
Level2	16	**254**	58	0	0	0	0	0	0.77	
Level3	0	38	**260**	46	1	0	0	0	0.75	
Level4	0	0	52	**183**	55	0	0	0	0.63	
Level5	0	0	1	53	**347**	53	0	0	0.76	
Level6	0	0	0	0	44	**334**	61	0	0.76	
Level7	0	0	0	0	0	57	**797**	57	0.87	
Level8	0	0	0	0	0	0	61	**379**	0.86	
	0.63	0.82	0.70	0.65	0.78	0.75	0.87	0.87	**0.794**	
	Precision								TOTAL	

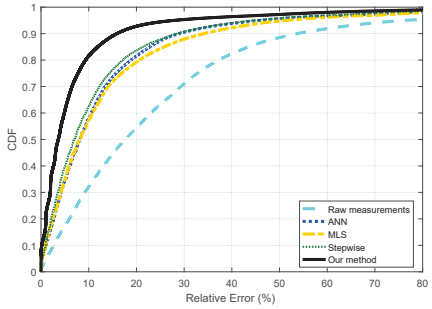


Fig. 5. Comparison in terms of relative error.

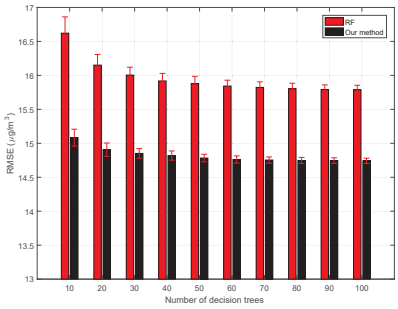


Fig. 6. Impact of the number of decision trees.

Table 3. RMSEs of our two-phase method against other state-of-the-art calibration methods.

Method	Raw data	ANN	MLS	Stepwise	Our method
RMSE	57.239	24.175 ± 0.001	25.142	22.718	14.762 ± 0.051

5.3 Comparative Study

We compare our method with existing calibration methods including ANN [6], MLS [15], Stepwise [8], and RF. ANN, MLS and Stepwise are state-of-the-art approaches used in recent works for air quality measurements. RF has never been used in existing work for calibrating low-cost sensors.

Comparison with ANN, MLS and Stepwise. Figure 5 shows the comparison results in terms of relative error. We see that our two-phase method achieves the best performance. Specially, 82% of the calibrated measurements have relative error less than 10% with our method, while the percentages for other methods are mainly distributed in the range of 56%-64%. Table 3 shows the comparison results in terms of RMSE, which further confirms the superiority of our model.

Comparison with RF. RF is a widely used data model which achieves very good performance in many scenarios. None of the existing works, however, has used this model for PM2.5 calibration. To investigate how well RF can perform in our scenario, we first perform a grid-based optimization using cross-validation on some of the parameters of the RF model. More specifically, we have tested the following parameters: max features per tree (1–5), min sample leaf (1–5) and min sample split (2–6). We use all 5 features and the corresponding reference measurements of the training data in the Ningbo dataset for the grid-search optimization. We have found that RF performs the best when max features per tree equals 4,

Table 4. RMSEs of all the calibration methods in two additional datasets.

Method	ANN	MLS	Stepwise	RF	Our method
RMSEs in abalone	2.1138 ± 0.0001	2.1144	2.0953	2.1067 ± 0.0239	2.0649 ± 0.0178
RMSEs in vinyl	0.2773 ± 0.0001	0.2774	0.2904	0.1925 ± 0.0013	0.1676 ± 0.0032

min sample leaf equals 2 and min sample split equals 4. Therefore, we use the above configuration for constructing the RF model.

We compare RF with our method. The second part RF of our two-phase method is also optimized using a grid-based optimization. We use the 5 features of the training data in the Ningbo dataset and the corresponding residual error after the first phase for the optimization. The best parameters are: max features per tree equals 3, min sample leaf equals 6 and min sample split equals 4. RF itself implements random data selection and random feature selection to model the complex nonlinear relationship and is able to avoid overfitting. Therefore, we have not set an additional dataset in the grid-search optimization to verify whether these parameters are overfit or not. Results show that the RMSEs of RF and our method are 15.844 (with confidence interval of 0.087) and 14.762, respectively.

We have also examined the impacts of decision tree number used in RF. Figure 6 shows the RMSEs by varying the number of decision trees K in RF and our method. We can see that (1) our method performs consistently better than RF with the same number of trees. (2) for both RF and our method, RMSE decreases when K increases. (3) RMSE decreases slowly when K increases beyond a certain value (e.g., 60). In our implementation, we set $K = 60$ to achieve a good tradeoff between accuracy and complexity.

Comparison in Other Datasets. We have employed different calibration approaches in two additional datasets (i.e. the abalone dataset and the vinyl dataset) in MATLAB [13] to further demonstrate the generalizability of our method. The abalone dataset contains 4177 measurements with 8 features while the vinyl dataset contains 68308 measurements with 16 features. These two datasets are provided for generalizing an input-output relationship. We use 70% data for training and the rest 30% data for testing. We employ our two-phase method as well as ANN, MLS, Stepwise and RF in the two datasets. Table 4 shows the RMSEs of above methods. Results show that our two-phase method still performs the best among these methods in both datasets, demonstrating the effectiveness of our method.

5.4 Impact of Different Factors

Impact of Feature Selection. Table 5 shows the feature subsets searched by the greedy algorithm (Algorithm 2) and their corresponding RMSEs for the two phases. We can see that: (1) the final calibration performance is better when the AIC value of a feature subset is smaller. Hence, AIC is an appropriate indicator for feature selection. For example, the feature subset $\{F_{pm2.5}, F_{time}, F_{humi}, F_{speed}\}$ has the smallest AIC, and it achieves the best calibration performance. (2) As expected, $F_{PM2.5}$ is the most important feature by comparing the top five rows. (3) F_{time} and F_{speed} are also important by comparing the 1st row with the 8th and 9th rows: adding F_{time}/F_{speed} reduces RMSE as well as AIC. (4) F_{temp} is irrelevant feature by comparing the 14th row with the 15th row since adding F_{temp} increases RMSE as well as AIC.

For our particular problem with a maximum of five features, the selected feature subset does not reduce AIC by a lot compared with simply using all five features. The automatic feature selection process becomes more important when there are more features (such as SO_2, NO_2, etc.) collected in many existing air quality measurement systems [8, 16]. We believe that the feature selection process will play an important role in our system in the near future when we plan to collect more air quality metrics such as CO_2, air

Table 5. Impact of different feature subsets.

Feature subset	AIC value	RMSEs after the first phase	RMSEs after the second phase
$\mathbf{F_{pm2.5}}$	**54420.5**	**36.4869**	**14.9739 ± 0.0634**
F_{humi}	70929.6	105.9198	15.9762 ± 0.1117
F_{temp}	70929.6	105.9245	15.9899 ± 0.1156
F_{speed}	70929.6	105.9222	15.9939 ± 0.1109
F_{time}	70929.7	105.9123	16.0064 ± 0.1159
$F_{pm2.5} + F_{humi}$	53178.3	33.4801	14.8861 ± 0.0526
$F_{pm2.5} + F_{temp}$	54426.7	36.4658	14.9984 ± 0.0592
$F_{pm2.5} + F_{speed}$	53221.3	33.4016	14.9173 ± 0.0554
$\mathbf{F_{pm2.5} + F_{time}}$	**52969.5**	**33.3643**	**14.8584 ± 0.0592**
$\mathbf{F_{pm2.5} + F_{time} + F_{humi}}$	**51373.8**	**29.6593**	**14.8206 ± 0.0575**
$F_{pm2.5} + F_{time} + F_{temp}$	52974.7	33.3446	14.8810 ± 0.0521
$F_{pm2.5} + F_{time} + F_{speed}$	51407.3	29.6161	14.8673 ± 0.0554
$F_{pm2.5} + F_{time} + F_{humi} + F_{temp}$	51379.3	29.6363	14.8317 ± 0.0621
$\mathbf{F_{pm2.5} + F_{time} + F_{humi} + F_{speed}}$	**49102.8**	**25.1686**	**14.7616 ± 0.0514**
$F_{pm2.5} + F_{time} + F_{humi} + F_{speed} + F_{temp}$	49103.7	25.1416	14.7924 ± 0.0551

pressure, etc. Moreover, the selection order in the greedy algorithm can somehow reflect the linearity of the features.

Our greedy algorithm tradeoffs optimality for efficiency, i.e., it does not guarantee that the subset of selected features is optimal. To see how well it performs in practice, we have investigated five existing datasets in MATLAB [13] (i.e., chemical dataset, abalone dataset, bodyfat dataset, house dataset, vinyl dataset). We compare our greedy algorithm with the brute force approach for enumerating all possible subsets. Results show that our greedy algorithm generates the same best subset as the brute force approach for all these cases. The greedy algorithm runs, however, much faster than the brute force approach. This is especially important for datasets with a large number of features and the brute force approach is computationally infeasible.

Impact of Different Cities. To see whether our model trained using one dataset works well for another dataset, we apply the trained model (trained based on the Ningbo dataset) to the Hangzhou dataset. As depicted in Figure 7, the precision differences of all 8 levels between the Hangzhou dataset and Ningbo dataset are less than 0.06. The average recalls in Hangzhou and Ningbo are 0.75 and 0.76, respectively. The overall calibration accuracy in the Hangzhou dataset is 0.790, which is very close to the accuracy in Ningbo (i.e. 0.794). These results show that the model trained in Ningbo has comparable performance in Hangzhou, implying that our trained model is general and can be directly used for different city environments.

Impact of Different Degrees of Linearity. The measurement data may exhibit different degrees of linearity. Some measurements may have strong linear relationship to the reference signal, some may not. We conduct simulation experiments to show the ability of our method to handle different degrees of linearity. The simulated PM2.5 concentration is $y = \beta_0 + \beta_1 {m'}_1^{\alpha} + \beta_2 {m'}_2^{\alpha} + \cdots + \beta_n {m'}_n^{\alpha} + e$, where ${m'}_1, {m'}_2, ..., {m'}_n$ represent randomly simulated feature values within the measurement range of real features and e is a random noise signal. We generate different traces by controlling a parameter α with $\alpha = 1$ representing perfect linearity.

Figure 8 shows the results. We can see that our method achieves the smallest RMSE in almost all cases. For data showing a strong linear relationship (i.e. α is close to 1), the first part model can already reduce the RMSE significantly while the second part plays a less important role in improving the final calibration accuracy. For data showing non-linear complex relationships (i.e. α is away from 1), almost

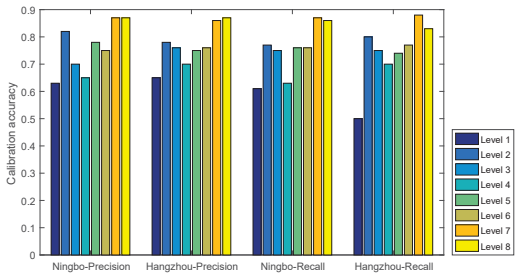


Fig. 7. Performance for different cities.

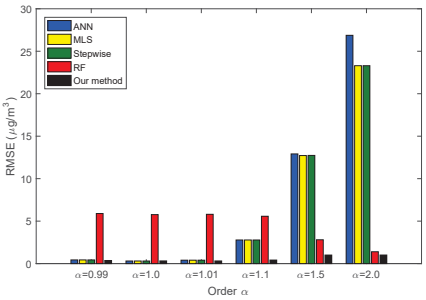


Fig. 8. Comparison of RMSE with different degrees of linearity ($\alpha = 1$ represents perfect linearity).

no features will be selected in the first part model and the second part plays a more important role improving the final calibration accuracy.

6 DISCUSSION

This section presents a discussion on some important issues and future directions of our method.

Feature Transformation. Feature transformation can create new features which could be useful when they have a descriptive power. However, we have found that simple feature transformation cannot achieve further performance improvement. First, feature transformation has no impact on the first part model. To confirm this, we have introduced feature transformations, e.g. two-way interaction terms, quadratic terms, in the linear model. However, our feature selection process avoids selecting these features since these features have little linear relationship with PM2.5 concentration. Second, feature transformation has almost no impact on the second part model. Our evaluation results show that introducing feature transformations do not improve the calibration accuracy. This is because RF is already able to model the complex relationship among all the features including their transformations.

More Measurements and Urban Big Data. The calibration accuracy can be further improved with the help of more measurements and urban big data. First, we can collect more data by a careful redesign of Mosaic node in the future, to include other kinds of data sources such as CO_2, NO, etc. Light is also an important data source. Since low-cost PM2.5 sensors are typically based on a light-scattering method, environmental light is naturally an important feature in calibration. Second, we can embrace urban big data, such as road networks, the intensity of people, traffic condition, and geographical location information. Such features are typically used in existing air quality inference approaches [4, 29]. We believe that these features are also helpful for further improving the data calibration accuracy. Taking this big data, an interesting future work is to extend our method so that it can also infer and predict PM2.5 concentration at any location and at any time.

A practice issue is that the computation overhead will be large when processing a large volume of data. Hopefully, the computation is mainly performed on the cloud side and once the calibration model is trained, it can be used directly for a sufficiently large time period. Moreover, with our careful design (e.g. suitable number of decision trees, greedy algorithm for feature selection, etc.), the computation overhead for model training is still acceptable.

Other Machine Learning Approaches. It is also possible to directly use other machine-learning models for data calibration. We have performed comprehensive experiments for the RF and come to the conclusion that our approach outperforms RF even when RF is carefully optimized. RF needs a large number of trees and large training sets to well handle linear data. We only use RF in the second phase to

learn the complex relationship between all available features and the residual error. The first part linear model can handle the linear data well and thus reduce the noises significantly in the first phase.

Besides RF, a multilayer perceptron can also be used for cross-sensitive calibration and its calibration result reported in [16] has been showed that it has slightly higher drift and noise than linear regression. Investigating other advanced machine learning approaches is one direction of our future work.

Benchmarks. Although we have shown that our method outperforms existing approaches, it is still unclear whether our method comes close to an established benchmark that would allow wide deployment of such low-cost sensors in the field. We have investigated for benchmarks for PM2.5 measurements. Unfortunately, all existing benchmarks are established for stationary measurement stations and have a relatively high accuracy requirement (e.g. relative error<5%). We have found no appropriate benchmarks for low-cost sensors (e.g. <25 US dollars). In our experiments, results show that 82% measurements have less than 10% relative errors and 90% measurements have less than 15% relative error, implying that our approach performs the best compared to existing approaches based on low-cost sensors.

Although there is still gap compared to high-precision stationary measurement stations, we believe that our approach can already be deployed in existing systems for a number of applications. We have deployed a system called Mosaic with our approach in two cities in China in the past three years. Other researchers have also deployed similar systems based on low-cost sensors in Europe [24] or America [19]. For applications such as pollution localization and warning, the absolute accuracy may not be so important. We can infer air pollution based on a large number of low-cost sensors and the relative changes in the measurements.

7 CONCLUSION

In this paper, we propose a two-phase data calibration method consisting of a linear part and a nonlinear part. We use MLS (multiple least square) to train the linear part, and use RF (random forest) to train the nonlinear part. We propose an automatic feature selection algorithm based on AIC (Akaike information criterion) for the linear model, which helps avoid overfitting due to the inclusion of inappropriate features. We evaluate our method extensively. Results show that our method outperforms existing approaches, achieving an overall accuracy improvement of 16.4% in terms of PM2.5 levels compared with state-of-the-art approach.

There remain several limitations in the current work. First, the current dataset contains a limited amount of features. It is thus not possible to fully evaluate advanced techniques such as feature selection and feature transformation. Second, RF also shows good performance in many scenarios. It remains unclear whether there exist more advanced machine learning approaches which outperforms our current approach. Finally, our current approach does not utilize urban big data such as road networks and geographical location information.

These limitations lead to multiple future directions. For example, we would like to redesign our Mosaic nodes to support more measurements such as CO_2, light, etc. An interesting future work is to further explore what other machine learning model could work better in our two phase model. Another interesting future work is to embrace urban big data to further improve our model's capability in calibrating, inferring and predicting PM2.5 concentration.

REFERENCES

[1] C.M. Bishop. 2007. Pattern Recognition and Machine Learning. *Springer* (2007).
[2] H. Bozdogan. 1987. Model selection and Akaike's Information Criterion (AIC): The general theory and its analytical extensions. *Psychometrika* 52, 3 (1987), 345–370.
[3] Central Moving Average. 2017. https://en.wikipedia.org/wiki/Moving_average. (2017).

[4] L. Chen, Y.Y. Cai, Y.F. Ding, M.Q. Lv, C.L. Yuan, and G.C. Chen. 2016. Spatially Fine-grained Urban Air Quality Estimation Using Ensemble Semi-supervised Learning and Pruning. In *Proc. of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.
[5] W.H. Chen, S.H. Hsu, and H.P. Shen. 2005. Application of SVM and ANN for intrusion detection. *Computers & Operations Research* 32, 10 (2005), 2617–2634.
[6] Y. Cheng, X.C. Li, Z.J. Li, S.X. Jiang, Y.L. Li, J. Jia, and X.F. Jiang. 2014. AirCloud: A Cloud-based Air-quality Monitoring System for Everyone. In *Proc. of the 12th ACM Conference on Embedded Networked Sensor Systems*.
[7] W. Dong, G.Y. Guan, Y. Chen, K. Guo, and Y. Gao. 2015. Mosaic: Towards City Scale Sensing with Mobile Sensor Networks. In *Proc. of the 21st IEEE International Conference on Parallel and Distributed Systems*.
[8] X.W. Fang and I. Bate. 2017. Using Multi-parameters for Calibration of Low-cost Sensors in Urban Environment. In *Proc. of the International Conference on Embedded Wireless Systems and Networks*.
[9] C. Frost and Thompson S.G. 2000. Correcting for regression dilution bias: comparison of methods for a single predictor variable. *Journal of the Royal Statistical Society Series A* 163, 2 (2000), 173–189.
[10] K.B. Fu, W. Ren, and W. Dong. 2017. Multihop Calibration for Mobile Sensing: k-hop Calibratability and Reference Sensor Deployment. In *Proc. of IEEE International Conference on Computer Communications*.
[11] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X.J. Liu, J.J. Bu, and C. Chen. 2016. Mosaic: A Low-Cost Mobile Sensing System for Urban Air Quality Monitoring. In *Proc. of IEEE International Conference on Computer Communications*.
[12] Machine Learning in Python tools. 2017. Scikit-learn. http://scikit-learn.org. (2017).
[13] The Mathworks Inc. 2014. Neural Network Toolbox Sample Data Sets for Shallow Networks. (2014).
[14] Journal of Toxicology and Environmental Health. 2017. http://www.tandfonline.com/toc/uteh20/current. (2017).
[15] B. Maag, O. Saukh, D. Hasenfratz, and L. Thiele. 2016. Pre-Deployment Testing, Augmentation and Calibration of Cross-Sensitive Sensors. In *Proc. of the International Conference on Embedded Wireless Systems and Networks*.
[16] B. Maag, Z.M. Zhou, O. Saukh, and L. Thiele. 2017. SCAN: Multi-Hop Calibration for Mobile Sensor Arrays. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), Article 19.
[17] M. Martin, J. Santos, H. Vasquez, and J. Agapito. 1999. Study of the interferences of NO2 and CO in solid state commercial sensors. *Sensors and Actuators B: Chemical* 58, 1 (1999), 469–473.
[18] M. Mead, O. Popoola, G. Stewart, P. Landshoff, M. Calleja, M. Hayes, J. Baldovi, M. McLeod, T. Hodgson, and Dicks J. 2013. The use of electrochemical sensors for monitoring urban air quality in low-cost, highdensity networks. *Atmospheric Environment* 70 (2013), 186–203.
[19] M. I. Mead, O. A. M. Popoola, G. B. Stewart, P. Landshoff, M. Calleja, M. Hayes, J. J. Baldovi, M. W. Mcleod, T. F. Hodgson, and J. Dicks. 2013. The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks. *Atmospheric Environment* 70, 2 (2013), 186–203.
[20] Dylos Air Quality Monitor. 2017. Dylos DC1700. http://www.dylosproducts.com/dc1700.html. (2017).
[21] R. Piedrahita, Y. Xiang, N. Masson, J. Ortega, A. Collier, Y. Jiang, K. Li, R. Dick, Q. Lv, and M. Hannigan. 2014. The next generation of low-cost personal air quality sensors for quantitative exposure monitoring. *Atmospheric Measurement Techniques* 7, 10 (2014), 3325–3336.
[22] PM2.5 Level. 2017. http://www.dwz.cn/cnnRO. (2017).
[23] D. Posada and T.R. Buckley. 2004. Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic Biology* 53, 5 (2004), 793–808.
[24] O. Saukh, D. Hasenfratz, and L. Thiele. 2015. Reducing Multi-hop Calibration Errors in Large-scale Mobile Sensor Networks. In *Proc. of the 14th International Conference on Information Processing in Sensor Networks*.
[25] SDS011 Particle Sensor. 2017. NOVA SDS011. http://www.inovafitness.com/a/minyongchanpin/chuanganqilei/2015/0522/32.html. (2017).
[26] J. Shang, Y. Zheng, and W. Tong. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proc. of the 20th International Conference on Knowledge Discovery and Data Mining*.
[27] L. Spinelle, M. Gerboles, M.G. Villani, M. Aleixandre, and F. Bonavitacola. 2015. Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide. *Sensors and Actuators B: Chemical* 215 (2015), 249–257.
[28] Y. Xiang, L.S. Bai, R. Piedrahita, R.P. Dick, Q. Lv, M. Hannigan, and L. Shang. 2012. Collaborative Calibration and Sensor Placement for Mobile Sensor Networks. In *Proc. of the 11th International Conference on Information Processing in Sensor Networks*.
[29] Y. Zheng, F. Liu, and H.P. Hsieh. 2013. U-Air: when urban air quality inference meets big data. In *Proc. of the 19th International Conference on Knowledge Discovery and Data Mining*.

Received August 2017; revised November 2017; accepted January 2018