Mosaic: Towards City Scale Sensing with Mobile Sensor Networks

Wei Dong, Gaoyang Guan, Yuan Chen, Kai Guo, and Yi Gao
College of Computer Science, Zhejiang University
Emails: {dongw, guangy, cheny, guok, gaoy}@emnets.org

***Abstract*—We introduce Mosaic, a mobile sensor network system for city scale sensing. Currently, we focus on accurate PM2.5 measurements. We design and implement low-cost smart sensors for this purpose. Those sensors are specifically designed for moving vehicles such as buses and taxies. The mobile sensing capability of our system enables a wide coverage of a city with a modest number of sensors. We incorporate a three-layer architecture in Mosaic. Mosaic faces many practical challenges which need to be properly addressed, e.g., reliable node hardware and software design, and accurate PM2.5 measurements. We have built three versions of Mosaic nodes. We present a novel data calibration approach combining traditional Artificial Neural Network (ANN) and Support Vector Machines (SVMs). We launch real-world deployments of Mosaic in two main cities in China—Hangzhou and Ningbo. Our initial efforts show that Mosaic is a feasible approach for city scale sensing and our system can produce highly accurate PM2.5 measurements.**

I. Introduction

Cities around the world are currently under quick transition towards a low carbon environment, high quality of living, and resource efficient economy. Sensing technology is a key building block towards the vision of smart cities in which digital technologies or information and communication technologies (ICT) are employed to reduce costs and resource consumption, and to enhance quality of living.

Over the past years, the quality of air has deteriorated significantly, especially for metropolitan cities in developing countries. For example, the Ministry of Environmental Protection in China started to release air quality reports for 161 main cities in 2014 [1]. According to the reports, the government states that the air quality is not good, and there is a long way to go for air quality improvement. The traditional way for measuring the air quality relies on stationary stations which are expensive to build and maintain (50K-100K dollars) [2]. The high costs limit its dense deployments in cities.

The recent advances in embedded technologies and wireless technologies have enabled a class of smart networked sensors which can provide valuable information with a wide deployment in the cities. For example, GreenOrbs [3] and CitySee [4] are two large-scale sensor network systems consisting of hundreds or thousands of static sensor nodes for measuring carbon emission and absorption in forests and urban areas. Although the cost is low for each individual node ($\sim$ 80 dollars), the overall cost could be unacceptable since a large amount of sensors are required to cover an entire city. For example, the 1,200 nodes are deployed in the CitySee project to cover a around 100km^2 area. For a modest city in China of approximately 16,000km^2 (e.g., Hangzhou), the overall cost is high and the system would be difficult to manage.

In this paper, we introduce Mosaic, a mobile sensor network system for city scale sensing. Currently, we target at PM2.5 measurements. We design and implement low-cost smart sensors for this purpose. Those sensors are specifically designed for moving vehicles such as buses and taxies. The mobile sensing capability of our system enables a wide coverage of a city with a modest number of sensors. We incorporate a three-layer architecture in Mosaic: the sensor network layer consists of mobile PM2.5 sensors with wireless communication capability; the cloud layer reprocesses the data, stores the data, and provides APIs for various applications; the application layer visualizes the data and provides services to end users. We plan to extend our system for sensing more data in the future.

Mosaic faces many practical challenges which need to be properly addressed, e.g.,

- *Reliable node hardware and software design*: how to design node hardware and node software so that they are reliable for long-term deployments in mobile vehicles? How to extend the Mosaic nodes so that they can be easily customized and extended to accommodate future needs.
- *Accurate PM2.5 measurements*: how to address the inherent tradeoff between low cost and high measurement accuracy? Since high-cost sensors are not suitable for dense deployment, we are facing the practical challenge of how to infer highly accurate sensor data with low-cost sensors.

We will introduce our approaches for addressing these challenges in Section III and Section IV, respectively. We launch real-world deployments of Mosaic in two main cities in China—Hangzhou and Ningbo. Our initial efforts show that Mosaic is a feasible approach for city scale sensing, and our system can produce highly accurate PM2.5 measurements with a novel data calibration approach.

Contributions of this paper are summarized as follows:

- We propose a three-layer architecture with low-cost sensors equipped on mobile vehicles for city scale sensing.
- We propose a novel data calibration approach for accurately inferring PM2.5 measurements with lost-cost sensors.
- We have designed and implemented the Mosaic nodes for PM2.5 measurements. We have launched real-world deployment of 10 nodes in Hangzhou and Ningbo, China.

We have also provided a website, an Android application, and a WeChat-based application for visualizing the sensing data.

The rest of this paper is structured as follows. Section II introduces the related work. Section III presents the system architecture. Section IV describes the data calibration approach. Section V introduces the real deployments in Hangzhou and Ningbo, China. Section VI presents the evaluation results. Section VII summarizes the lessons we have learnt, and finally, Section VIII concludes this paper and gives future research directions.

II. Related Work

Urban sensing. Urban sensing is a key building block towards the smart city vision. It attracts much research attention. In [5], the authors introduce CitySense, an urban-scale wireless networking testbed. CitySense consists of about 100 Linux-based embedded PCs outfitted with dual 802.11a/b/g radios and various sensors, mounted on buildings and streetlights across the city of Cambridge. CitySense does not explicitly target for a specific monitoring application. Rather, CitySense serves as a new kind of experimental apparatus for urban-scale distributed systems and networking research efforts. As such, it is unsuitable for fine-grained air quality monitoring which may require a very dense deployment of smart networked sensors.

CitySee [5] is a real-world large-scale sensor network system. The goal of CitySee is to deploy thousands of wireless sensor nodes in an urban area of Wuxi City, China, such that multi-dimensional data including CO_2, temperature, humidity, illumination, location, and etc., could be collected in a real-time manner for further analysis. The initial prototype of the system was deployed on May 30th 2011, which included 100 nodes. In early August 2011, the system reached its largest scale, 1,200 nodes, including 1100 normal nodes and 100 carbon nodes. The deployment covers around 100km^2 urban areas. We think that static deployment of a sensor network is still costly for dense and city scale sensing.

Mobile sensing is a relatively new concept to achieve wide coverage of an area with a modest number of sensors. In essence, node mobility trades off temporal resolution against spatial resolution, enabling a high spatial resolution across large areas without the need for thousands of fixed sensors. Many different mobile sensing applications are developed, including data collection [6, 7], air quality monitoring [8–10], radio spectrum analysis [11], and etc.

Air quality monitoring. Traditionally, pollution measurements are performed using expensive equipment at fixed locations or dedicated mobile equipment laboratories. This is a coarse-grained and expensive approach. In [10] the authors present a vehicular-based mobile approach for measuring fine-grained air quality in real-time. A prototype of sense node is designed to monitor the air quality. Compared to [10], our work realizes the mobile sensing concept a step further. We carefully design the hardware and software so that the node is suitable for deployment in moving vehicles. We address the specific challenge of data calibration with low-cost sensors. In addition, we launched real-world deployments in two main cities in China.

In [8], the authors propose to use a mobile measurement system for air quality measurement. The mobile measurement system consists of ten sensor nodes installed on top of public transport vehicles, which cover a large urban area on a regular schedule. Each sensor node is equipped with a high cost measurement device (MiniDiSC, costs 3K$\sim$5K dollars) to monitor ultrafine particle (UFP) concentrations. It is assumed that the sensing data is highly accurate due to the use of high cost and high precision sensors. In contrast, we use sensor nodes of much lower cost ($\sim$100 dollars), hence we are facing the practical challenge of how to calibrate the low-cost sensors.

Based on the mobile sensing concept of [8], authors in [12] propose three modeling methods to generate high spatio-temporal resolution air pollution maps for urban environments. Since the sensor network coverage is spatially and temporally dynamic, the authors leverage models to estimate the values for the locations and times where the data are not available. Using known sensing data at particular locations to infer unknown data at other locations is complementary to our current work and can be considered as a future work of Mosaic.

AirCloud [2] is a novel client-cloud system for pervasive and personal air-quality monitoring at low cost. At the frontend of AirCloud, two types of Internet-connected PM2.5 sensing nodes, AQM and miniAQM, are created. On the cloud-side, an air-quality analytics engine is designed to calibrate the PM2.5 sensor nodes in real-time. In the current deployment of AirCloud, 12 sensors are deployed at difference locations in a city. A real deployment of AirCloud consists of 10 AQMs, 2 miniAQMs, 3 Dylos (relatively high precision PM2.5 sensor, costs 500$\sim$700 dollars [13]), deployed mainly in static locations. Different from AirCloud, Mosaic equips sensors on moving vehicles to cover a much larger area in a city.

Data calibration, inference, and applications. AirCloud uses the Artificial Neural Network (ANN) model to calibrate the sensor data according to the reference signals produced by a high precision monitor. We find that the ANN calibration method employed in AirCloud yields unsatisfactory results in our mobile environments, motivating us to devise a more accurate inference method.

In [9], the authors propose the concept of multi-hop calibration in mobile sensor networks, i.e., calibrating a sensor using an already calibrated sensor when they meet. The authors exploit the fact that temporally and spatially close measurements of different sensors measuring the same phenomenon are similar. The authors propose a novel multi-hop calibration algorithm using geometric mean regression. Compared with ordinary least squares regression, it significantly reduces error propagation in the network. This work is complementary to our work. We can also use stationary measurement stations (or statically deployed reference nodes) for online calibration, further improving the data measurement accuracy.

In [14], the authors introduce a real system that is deployed in the indoor offices, using the Dylos sensor nodes. This system instantly monitors indoor air quality on different floors of a building. The information can instruct users in

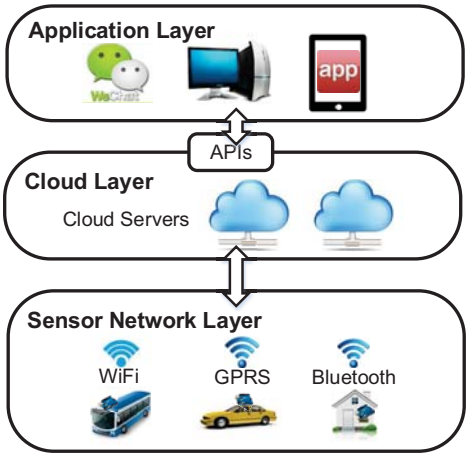


Fig. 1: Three layer architecture in Mosaic

making decision, e.g., finding the right time to work out in the gym or turn on individual air filters in her own office. Different from [14], our Mosaic system targets at outdoor air quality monitoring. In the current stage, we focus on issues of system reliability, data acquisition and calibration. It would be interesting to explore future applications based on our obtained data.

In [15], The authors infer the real-time and fine-grained air quality information throughout a city, based on the (historical and real-time) air quality data reported by existing monitor stations and a variety of data sources we observed in the city, such as meteorology, traffic flow, human mobility, structure of road networks, and point of interests (POIs). The authors propose a semi-supervised learning approach based on a co-training framework that consists of two separated classifiers. Our work is complimentary to their work, the fine-grained data collected by our system can provide valuable information to their framework.

III. System Architecture

A. Three-layer architecture in Mosaic

Our Mosaic system consists of three layers: the sensor network layer, the cloud layer, and the application layer (see Fig. 1).

The sensor network layer. We have developed the Mosaic nodes for collecting various environmental data in the city. We choose the Arduino platform [16] for easy customization and extension. In the current stage, we target at accurate PM2.5 measurements. Both the hardware and software are carefully designed so that the nodes are reliable for long-term deployment in mobile vehicles. Both the sensing data (PM2.5, temperature, and humidity) and the trajectory information (provided by GPS) are transferred to the cloud server by GPRS. In the future, we would extend the Mosaic node for measuring more environmental data. We will introduce the hardware and software of the Mosaic nodes in detail in Section III-B and Section III-C.

The cloud layer. We use the cloud service provided by Aliyun [17]. The raw sensing data is stored and processed. In particular, the raw PM2.5 measurements are calibrated by

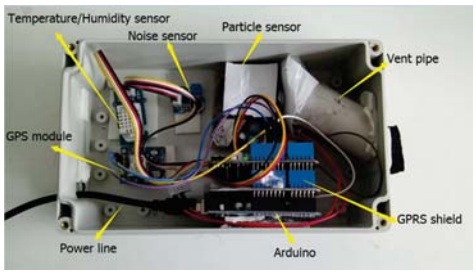


Fig. 2: Mosaic node hardware

novel data calibration method introduced in Section IV. Based on the cloud services, we have provided a set of well-defined APIs for accessing the sensing data. Based on the provided APIs, various applications can be developed.

The application layer. The application layer visualizes the data and provides services to end users by interacting with the cloud layer with the HTTP protocol. Currently, we have developed three different applications—a website, a WeChat based app, and an Android app, based on our well-defined APIs for accessing the sensing data in the cloud layer. We will introduce both the cloud layer and application layer in Section III-D. We will also present how we visualize the PM2.5 data in the evaluation section.

B. Hardware

We have developed three kinds of Mosaic nodes. miniMosaic is designed for PM2.5 monitoring for indoor environments. Each miniMosaic connects to a mobile phone via Bluetooth, providing the PM2.5, temperature and humidity data. Mosaic node v1 and v2 are both designed for outdoor environments with a special consideration for installation on mobile vehicles. Mosaic node v2 differs from Mosaic node v1 in that it incorporates a more powerful and highly integrated microcontroller. Table 1 shows the main components for the three kinds of nodes and Fig. 2 shows the picture of Mosaic node v1. In the following paragraphs, we will briefly introduce the main components incorporated in the Mosaic node.

Main board/microcontroller. We have created miniMosaic and Mosaic node v1 based on Arduino which is an open-source electronics platform [16]. In addition to its open source feature, Arduino has the advantage of easy customization. The microcontroller of the Arduino is Atmel AVR ATmega328P where the suffix P stands for picoPower. The processor can work on a wide range of voltages, from 1.8V to 5.5V. AVR adopts the Harvard architecture which means it has different address spaces for data memory and program memory. The data memory is 2KB and the program memory is 32KB. ATmega328P adopts an advanced RISC architecture with most instructions executing in just one clock cycle. The default clock rate of ATmega328P is 16MHz and can reach 20MHz. We have created Mosaic node v2 based on Linkit ONE since its microcontroller is much more powerful than Arduino and it integrates many peripherals (e.g., GSM/GPRS, WiFi, Bluetooth, GPS, SD card, etc). The microcontroller of Linkit ONE is ARM7 EJ-S^{TM} with 4MB RAM and 16MB ROM. It has a clock rate of 260MHz which is much faster than ATmega328P. Although the cost of Linkit ONE is higher than

Table 1: Main components for three kinds of nodes

Components \ Mosaic node	miniMosaic	Mosaic v1	Mosaic v2
Main board/Microcontroller	Arduino UNO/ATmega328P (16MHz, 2KB RAM, 32 KB Flash)	Arduino UNO/ATmega328P (16MHz, 2KB RAM, 32 KB Flash)	Linkit one/MT2502A ARM7 EJ-S™ (260MHz, 4MB RAM, 16MB Flash)
Power	5V	5V	3.7~4.2V
GSM/GPRS module	N/A	SIM900 (GPRS class 10/8)	MT2502 built-in (GPRS class 12)
WiFi module	N/A	N/A	MediaTek MT5931
Bluetooth module	SLD63030P	N/A	MT2502 built-in
SD card	Seeeduino SD Card Shield V4.0 (SD card 16GB max.)	N/A	MT2502 built-in (SD card 32GB max.)
GPS module	SIM28 (2.5m max. err.)	SIM28 (2.5m max. err.)	MediaTek MT3332
Dust sensor	PPD42NS (Approx. 1μm min.)	PPD42NS (Approx. 1μm min.)	PPD42NS (Approx. 1μm min.)
Temperature/Humidity	DHT11 (range: 20%~90% RH, 0~50 °C)	DHT11 (range: 20%~90% RH, 0~50 °C)	DHT11 (range: 20%~90% RH, 0~50 °C)
Noise sensor	SEN02281P (range: -48~66dB)	SEN02281P (range: -48~66dB)	SEN02281P (range: -48~66dB)

Arduino (∼70 dollars vs. ∼30 dollars), it is much cheaper than the overall cost if we assemble all the peripherals with Arduino.

Power supply. Currently we are using two kinds of power supply: car power supply and a battery. With a car plug, the car power supply can provide 5V/1A DC. We also provide an option to use a battery for power supply.

Communication module. The Mosaic node has several communication modules, e.g., GSM/GPRS, WiFi, and Bluetooth, each for different purposes. GSM/GPRS is mainly used for outdoor environment where the sensors are installed on moving vehicles. GSM/GPRS is suitable if we need real-time data collection and the WiFi communication is not always feasible. Bluetooth is used in miniMosaic for short range communication with smartphones. With the wide deployment of WiFi access points, we believe that the WiFi communication will be widely used in future applications.

Storage module. A Mosaic node has a 8GB SD card for buffering the sensing data. It is also very useful for storing the log information when we develop the node software.

Sensors. Mosaic node has various sensors. We choose PPD42NJ as our particulates sensor as PPD42NJ is of low cost and reports consistent results [2]. Based on the light-scattering method, PPD42NJ is able to measure the count of particulates ($>1\mu$m) per unit volume. Since PM2.5 concentration typically refers to the mass of the particulate matters with diameters less than 2.5μm, calibration and inference are required in order to generate accurate measurement data. In this paper, we use the commonly used Dylos node for the reference for calibration.

A Mosaic node also includes a GPS sensor, a temperature and humidity sensor. These sensor data are used in the calibration process as the concentration of PM2.5 is heavily influenced by meteorology factors according to previous work [2].

C. Node software

The Mosaic node software is written according to the programming paradigm of Arduino and Linkit ONE. There is a `setup()` function for initialization of various device drivers. The application logic is mainly implemented in the `loop()` function which periodically reads various sensor data via interfaces provided by the sensor drivers. Besides sensor drivers, there are also communication drivers, including GSM/GPRS, Bluetooth, and WiFi.

Fig. 3 shows the different software components on the Mosaic nodes. There are three main components which implement the application logic.

The communication module transmits the sensing data via different transmission technologies. For the indoor scenario, we currently use Bluetooth for short range communication with smartphones. For the outdoor scenario (e.g., deployed on moving vehicles), we currently use GSM/GPRS for communication. With the wide deployment of WiFi access points, we expect that WiFi communication will soon be used in the near future.

The remote configuration module is responsible for remotely configuring the working parameters of the software. For example, the data uploading URL address can be dynamically reconfigured via the configuration module. Other examples include the on/off states of various sensors and the data sampling rates.

The logging module is used to buffer the recent sensing data. In the case of transmission failure, the sensing data can still be available by possible retransmissions at a later time.

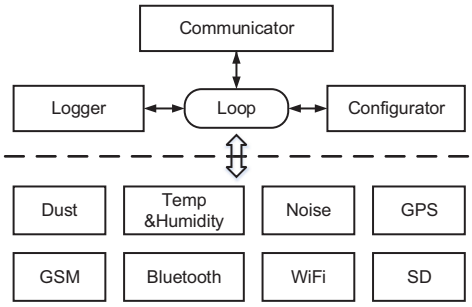


Fig. 3: Mosaic node software

The logging module is also used to record the execution logs which are very useful for program diagnosis.

D. Cloud and apps

We use the cloud services provided by Aliyun, a secure and reliable cloud computing infrastructure in China. We use two single virtual private servers, each consisting of a single-core CPU, 512MB memory, 20GB hard disk and 100Mbps Fast Ethernet. The operating system running on the cloud server is Ubuntu 14.04.3 LTS.

The cloud layer accepts the sensing data, saves the raw data onto the hard disk. The raw PM2.5 sensing data is then calibrated to more accurate measurements using our novel calibration method (see Section IV).

The calibrated PM2.5 data (and other sensing data) are stored in a MySQL database. We have developed some key APIs for upper-layer applications to query the sensing data. The APIs can support third-party developers to build various applications. We have built one website, one Android application, and a WeChat-based application based on these APIs. We will illustrate the basic functionalities of these applications in Section VI-B.

IV. Data Calibration

In this section, we describe how we calibrate measurements produced by our low cost PPD42NJ sensor based on the reference measurements produced by Dylos.

Previous studies reveal that (1) the concentration of PM2.5 is largely influenced by meteorology factors, such as temperature, humidity, etc, (2) the relationship between raw measurements (produced by PPD42NJ) and the reference measurements is complicated and no-linear. Hence we would like to choose learning-based approach with key meteorology factors as inputs.

Artificial Neural Network (ANN) is used as a state-of-the-art data calibration approach in [2]. In this approach, the widely-used back-propagation (BP) network is chosen and the neural network is capable of well fitting an arbitrary function. The neural network takes the readings of PPD42NJ with co-located temperature and humidity readings as inputs and the ground truth value (given by Dylos) as output. We find that the direct use of the ANN-based calibration approach is not suitable for our mobile deployments.

First, the start and stop of buses will introduce significant noises to the PPD42NJ sensors since the airflow has significant impact on the measurement accuracy. We observe that

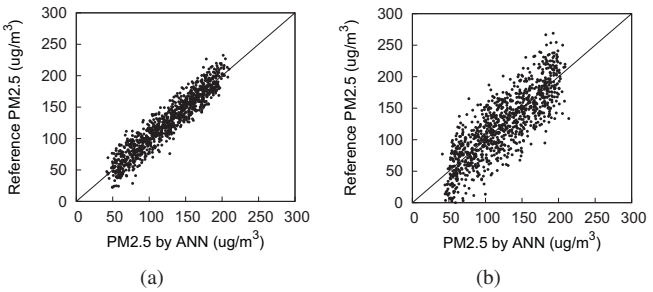


Fig. 4: Two experiments for applying ANN. (a) Experiment I. (b) Experiment II.

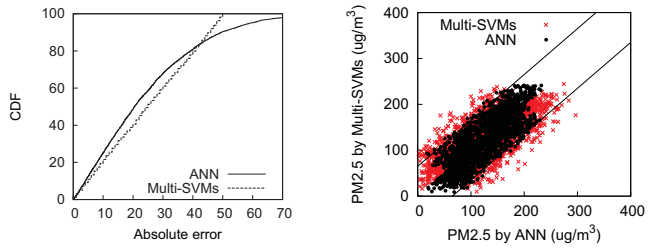
measurements in normal speed produce more consistent results while measurements during the start and stop of buses show much randomness. This is because the moving speed causes different levels of airflow to our sensors without built-in fans.

Second, the ANN-based approach is prone to go to local minima with finite data sets [18, 19], i.e., it tends to fit the trained data well while has poor performance in predicting the new unseen data (i.e., overfitting). To confirm this, we perform ANN-based calibration on two datasets: (1) measurements on the No.16 and No.9 buses during one month; (2) measurements on the No.28 and No.53 buses during one month. The first experiment (Experiment I) is performed on the first dataset: we train the ANN model using 70% data and use it to predict the remaining 30% data. The second experiment (Experiment II) uses two datasets: we train the ANN model using 70% data in the first dataset and use it to predict the remaining data in both datasets. Figs. 4(a)(b) show the scatter plot between the calibrated data (using the trained ANN model) and the reference data (using the Dylos measurements). The second experiment shows that there is a large portion of calibrated data showing a significant deviation from the ground truth with the ANN-based approach.

To address the first problem, we simply filter out the measurement data during the start and stop of buses using GPS data. We believe that the additional use of accelerometer will further improve the detection accuracy and we will investigate this problem in detail in the future.

To address the second problem, we try to employ more suitable learning based approach for calibration. SVM-based model is less prone to overfitting since the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum [19]. We employ Multi-SVMs for our prediction problem. We first discretize the PM2.5 concentration in the range of 0–500. We then use multiple SVMs to classify the raw data into different discrete levels, i.e., the calibrated measurements. We compare Multi-SVMs with ANN for experiment II. Fig. 5(a) shows the CDF of the absolute error. We find that Multi-SVMs outperforms ANN in some cases while ANN outperforms Multi-SVMs in other cases. Hence, the direct use of Multi-SVMs is not suitable.

In order to choose the most suitable approach for a particular raw data point, we investigate each data point in detail. Fig. 5(b) shows the scatter plot with the x-axis showing



(a) (b)

Fig. 5: Comparison of ANN and Multi-SVMs. (a) CDF of absolute error. (b) predicted values of ANN and Multi-SVMs.

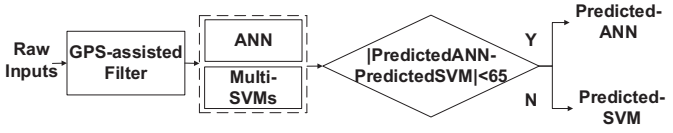


Fig. 6: Workflow of data calibration.

the ANN predicted value (PredictedANN) and the y-axis showing the Multi-SVMs predicted value (PredictedSVM). The dots denote data points for which ANN is more suitable and the crosses denote data points for which Multi-SVMs is more suitable. We can see that up left area and bottom right area ($|\texttt{PredictedANN} - \texttt{PredictedSVM}| \geq 65$), it is clear that Multi-SVMs is better. In the area near the "y=x" line, there are data points for which ANN is better as well as data points for which Multi-SVMs is better. Since there are more data points for which ANN is better, we select ANN in this area ($|\texttt{PredictedANN} - \texttt{PredictedSVM}| < 65$). We thus propose a simple adaptive calibration approach based on the empirical threshold of 65.

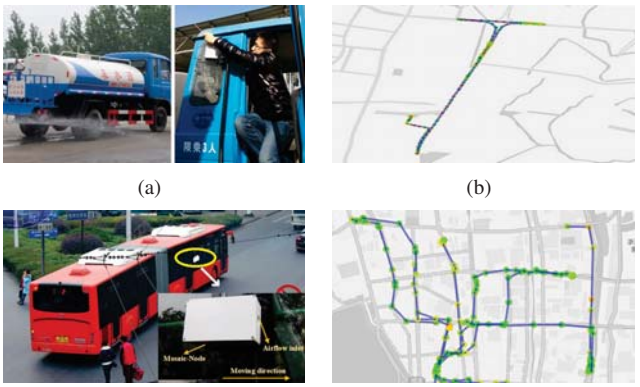
To summarize, Fig. 6 shows the overview of our approach. The raw input data go through two steps. In the first step, our approach filters out noisy data near the start and stop of buses. In the second step, we apply the learning based calibration approach. We first calculate the difference of PredictedANN and PredictableSVM. If the difference is larger than the empirical threshold of 65, PredictedSVM is used as the calibrated measurement. Otherwise, PredictedANN is used as the calibrated measurement.

V. Initial Deployment

We started to deploy our system from December 2014. We deployed two Mosaic nodes v1 on two road cleaning vehicles in the city of Ningbo, China. The road cleaning vehicles follow fixed trajectories every day to clean the roads in the city. We installed the two nodes at the window of the vehicles. Both nodes were installed in a way that the air vent was heading the moving direction of vehicles so that the air-flows can be stable. Each node was powered by vehicles. Each node had a data sampling period of 30 seconds and various sensor data were recorded on SD card. The Mosaic node v1 did not have the wireless communication module. We had to check the nodes regularly to ensure that they were working well. One month later, we retrieved the data manually. During the first

Table 2: Mosaic deployments

#Deployment	1	2	3
Duration	2014.12.13~2015.1.7	2015.2.24~2015.4.3	2015.6.7~now
Node version	Mosaic v1 (no GSM)	Mosaic v1	Mosaic v2
#Node	2	8	2
Vehicle	Cleaning vehicle	Bus	Cleaning vehicle
City	Ningbo, China	Hangzhou, China	Ningbo, China



(a) (b) (c) (d)

Fig. 7: Deployments in Hangzhou and Ningbo. (a)(b) Ningbo. (c)(d) Hangzhou.

deployment, we improved the Mosaic node design. We added the GPRS module so that the sensing data could be transferred to the Internet in a real-time manner.

We launched the second deployment in the city of Hangzhou, China. The deployment consisted of eight Mosaic nodes v1 installed on eight buses for five weeks, covering an area of 2.9×3.1 km^2. We also deployed two Dylos nodes for data calibration. Each node had a data sampling period of 60 seconds.

The third deployment started from June, 2015. We also deployed two Mosaic nodes v1 with GPRS module. In addition, we deployed two Dylos nodes so that we could get the reference measurements for calibration. The Dylos nodes lacked automatic data uploading capability so we had to retrieve the reference measurements every week.

Table 2 summarizes our three real-world deployments and Fig. 7 shows the deployment pictures in Ningbo and Hangzhou. In Fig. 7(b)(d), we show the trajectories of moving vehicles and use different colors to present different pollution levels. Currently, we have implemented Mosaic node v2 with Linkit ONE as the main board. We plan to deploy Mosaic node v2 in the near future.

VI. Evaluation

In this section, we evaluate our Mosaic system. Section VI-A evaluates the accuracy of our data calibration approach, and Section VI-B shows the efficacy of our system by visualizing the collected sensor data.

A. Data Calibration

We compare three data calibration approaches, ANN, Multi-SVMs, and our approach, based on the two datasets described in Section IV. We find that the average absolute errors of

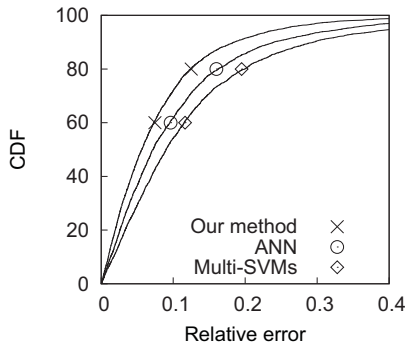


Fig. 8: Relative error comparison

Table 3: Prediction performance of ANN.

Ground Truth	Predictions								Recall
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	
Level 1	**0**	0	1	0	0	0	0	0	
Level 2	40	**195**	86	0	0	0	0	0	**0.61**
Level 3	2	110	**283**	98	3	0	0	0	**0.57**
Level 4	0	3	104	**267**	107	2	0	0	**0.55**
Level 5	0	0	5	107	**420**	117	0	0	**0.65**
Level 6	0	0	0	1	115	**427**	132	0	**0.63**
Level 7	0	0	0	0	2	116	**1003**	111	**0.81**
Level 8	0	0	0	0	0	0	63	**81**	**0.56**
		0.63	**0.59**	**0.56**	**0.65**	**0.65**	**0.84**	**0.42**	
	Precision								

Table 4: Prediction performance of our approach

Ground Truth	Predictions								Recall
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	
Level 1	**0**	0	0	0	0	0	0	0	
Level 2	21	**235**	65	0	0	0	0	0	**0.73**
Level 3	0	82	**344**	69	1	0	0	0	**0.69**
Level 4	0	0	74	**317**	92	0	0	0	**0.66**
Level 5	0	0	3	94	**471**	81	0	0	**0.73**
Level 6	0	0	0	0	79	**499**	97	0	**0.74**
Level 7	0	0	0	0	0	97	**1053**	82	**0.85**
Level 8	0	0	0	0	0	0	50	**94**	**0.65**
		0.74	**0.71**	**0.66**	**0.73**	**0.74**	**0.88**	**0.53**	
	Precision								

ANN and Multi-SVMs are 24.1 and 24.8, respectively. Our approach has an average absolute error of 18, resulting in a 10% improvement compared with ANN-based approach. Fig. 8 gives the relative error distribution of PM2.5 concentration. We can see that our approach outperforms both ANN and Multi-SVMs.

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 [20]. We perform the evaluation on 20,000 data points with 16,000 training data points. Table 3 gives the confusion matrix of ANN, and Table 4 gives the confusion matrix of our approach. We can see that for each level, our approach outperforms ANN in terms of precision and recall. The overall accuracy of ANN-based approach is 0.669 while our approach achieves an overall accuracy of 0.753.

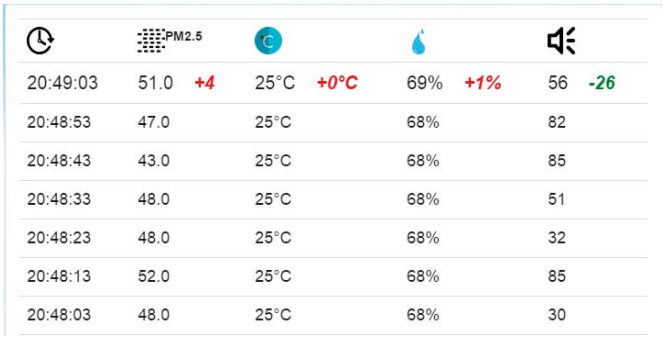


Fig. 9: Real-time data query

Fig. 10: Air quality following the trajectories of moving vehicles

B. Data Visualization and Applications

Based on the cloud services, we have built a website, an Android application, and a WeChat-based application for visualizing the sensor data. Figs. 9–11 show the user interface for the website which includes three main functionalities, i.e., real-time data query, pollution maps, and statistical data. We customize the web pages to be compatible with both PC browsers and mobile browsers.

In the real-time query interface, users can view the air quality data as well as temperature and humidity in a real-time manner (see Fig. 9).

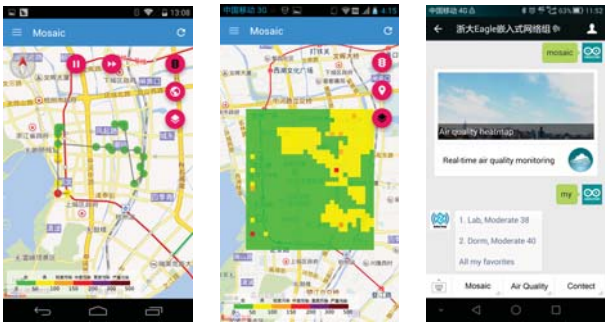
In the pollution maps interface, the air quality data following the trajectories of moving vehicles can be displayed (see Fig. 10). We use different colors to present different pollution levels. This interface can replay how we collect the sensing data. Users can select a city and a date. Users can also control the replay speed. We can also display the heat map of air quality data by existing interpolation methods.

We have also developed an Android application and a WeChat-based application based on our open cloud services (see Fig. 11). We hope that more applications can be developed based on our provided services in the future.

VII. Lessons Learned

In this section, we summarize the lessons we have learnt during the development and deployment of our Mosaic system. We hope that these lessons can provide useful guidance for future mobile sensor network deployments.

String constants waste precious RAM. The compiler compiles string constants onto RAM space which is only 2kB on



(a) (b) (c)

Fig. 11: Android App & WeChat-based App

Arduino UNO. The direct and frequent uses of string constants will thus consume the precious RAM space. Developers can utilize the Arduino keyword PROGMEM which instructs the compiler to place the corresponding string constants on the program memory which is typically much larger.

Pay attention to the shutting down of GPRS module. In the Mosaic node, the GPRS module draws current directly from Arduino without additional 5V adapter. With car power supply, we frequently observe the shutting down of GPRS module due to low power supply. Developers should take this factor into consideration for developing reliable node software.

Discard sampling data during start and stop of buses. The start and stop of buses will cause huge disturbances on the readings of particle sensors. It is better to to filter out the sensing data during that time.

Vibration-absorbing is necessary. Sensing nodes are deployed in a mobile environment with severe vibration. So we have to make sure that the inner structures of each node keep stable. We first tried a screw-based structure. However, the vibration caused by moving vehicle makes many parts of the node very unstable. Therefore, we added adhesive tapes to absorb the vibration on the road.

Pay attention to remote node errors. When sensor nodes are deployed in the wild, there could be many unexpected errors, e.g., no GPRS network connection and abnormal sensing readings. In order to detect such errors in a timely manner, we designed and implemented a text-based alarm system in both the sensor network layer and the cloud layer. By such a mechanism, we will soon be alerted when there are errors.

VIII. Conclusion

In this paper, we introduce Mosaic, a mobile sensor network system for monitoring city scale PM2.5 levels. We design and implement low-cost smart sensors for this purpose. Those sensors are specifically designed for moving vehicles such as buses and taxies. The mobile sensing capability of our system enables a wide coverage of a city with a modest number of sensors. We incorporate a three-layer architecture in Mosaic. We have built three versions of Mosaic nodes. We present a novel data calibration approach combining traditional ANN and SVMs. We have launched real-world deployments of Mosaic in two main cities in China—Hangzhou and Ningbo. Our initial efforts show that Mosaic is a feasible approach for city scale sensing and our system can produce highly accurate PM2.5 measurements. We have also built applications for real-time query of the sensing data and to visualize the sensing data in specified areas in a city.

There are multiple future directions. First, we plan to launch a larger-scale deployment in Hangzhou. Second, we would like to investigate many research challenges involved in this project, e.g., how to deploy the nodes for better coverage.

Acknowledgment

This work is supported by the National Science Foundation of China under Grant No. 61472360, Zhejiang Provincial Platform of IoT Technology under Grant No. 2013E60005, and Zhejiang Commonwealth Project under Grant No. 2015C33077.

References

[1] http://www.gov.cn/zhuanti/2014-12/23/content_2795356.htm.
[2] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "AirCloud: A Cloud-based Air-quality Monitoring System for Everyone," in *Proc. of ACM SenSys*, 2014.
[3] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, L. Mo, W. Dong, Z. Yang, M. Xi, J. Zhao, and X.-Y. Li, "Does wireless sensor network scale? A measurement study on GreenOrbs," in *Proc. of IEEE INFOCOM*, 2011.
[4] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, and J. Wang, "CitySee: not only a wireless sensor network," *IEEE Network*, vol. 27, no. 5, pp. 42–47, 2013.
[5] R. Murty, G. Mainland, I. Rose, A. Chowdhury, A. Gosain, J. Bers, and M. Welsh, "CitySense: An Urban-Scale Wireless Sensor Network and Testbed," in *Proc. of IEEE Conference on Technologies for Homeland Security*, 2008.
[6] R. Sugihara and R. K. Gupta, "Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility," in *Proc. of IEEE DCOSS*, 2008.
[7] R. Sugihara and R. K. Gupta, "Optimizing Energy-Latency Trade-off in Sensor Networks with Controlled Mobility," in *Proc. of IEEE INFOCOM*, 2009.
[8] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the Spatio-Temporal Resolution Limit of Urban Air Pollution Maps," in *Proc. of IEEE PerCom*, 2014.
[9] O. Saukh, D. Hasenfratz, and L. Thiele, "Reducing Multi-hop Calibration Errors in Large-scale Mobile Sensor Networks," in *Proc. of ACM/IEEE IPSN*, 2015.
[10] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time Air Quality Monitoring Through Mobile Sensing in Metropolitan Areas," in *Proc. of ACM Urban Computing*, 2013.
[11] T. Zhang, N. Leng, and S. Banerjee, "A Vehicle-based Measurement Framework for Enhancing Whitespace Spectrum Databases," in *Proc. of ACM MobiCom*, 2014.
[12] A. Marjovi, A. Arfire, and A. Martinoli, "High Resolution Air Pollution Maps in Urban Environments Using Mobile Sensor Networks," in *Proc. of IEEE DCOSS*, 2015.
[13] Dylos, http://www.dylosproducts.com.
[14] X. Chen, Y. Zheng, Y. Chen, Q. Jin, W. Sun, E. Chang, and W.-Y. Ma, "Indoor Air Quality Monitoring System for Smart Buildings," in *Proc. of ACM UbiComp*, 2014.
[15] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring Gas Consumption and Pollution Emission of Vehicles throughout a City," in *Proc. of ACM SIGKDD*, 2014.
[16] Y. Badamasi, "The working principle of an Arduino," in *Proc. of International Conference on Electronics, Computer and Computation*, 2014.
[17] Aliyun, http://www.aliyun.com/product/ecs/?lang=en.
[18] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, no. 10, pp. 2617–2634, 2005.
[19] C. M. Bishop, "Pattern Recognition and Machine Learning," *Springer*, 2007.
[20] PM2.5, http://www.dwz.cn/cnnRO.