

When Sharing Economy Meets IoT: Towards Fine-grained Urban Air Quality Monitoring through Mobile Crowdsensing on Bike-share System

DI WU, TAO XIAO, XUEWEN LIAO, and JIE LUO, Hunan University, China

CHAO WU, Zhejiang University, China

SHIGENG ZHANG, Central South University, China

YONG LI, Tsinghua University, China

YIKE GUO, Imperial College London, United Kingdom

Air pollution is a serious global issue impacting public health and social economy. In particular, exposure to small particulate matter of 2.5 microns or less in diameter (PM_{2.5}) can cause cardiovascular and respiratory diseases, and cancer. Fine-grained urban air quality monitoring is crucial yet difficult to achieve. In this paper, we present the design, implementation, and evaluation of an ambient environment aware system, namely UbiAir, which can support fine-grained urban air quality monitoring through mobile crowdsensing on a bike-sharing system. We have built specific IoT box configured with multiple pollutant sensors and attached on shared bikes to sample micro-scale air quality data in the monitoring space that is split by a scalable grid structure. Both hardware and software data calibration methods are exploited in UbiAir to make the sampled data reliable. Then, we use Bayesian compressive sensing (BCS) as an inference model that leverages the calibrated samples to recover data points without direct measurements and reconstruct an accurate air quality map covering the entire monitoring space. In addition, red envelope based incentive schemes and differential rewarding strategies have been designed in UbiAir, and an adaptive BCS algorithm is proposed to deploy the red envelopes at the most informative positions to facilitate data sampling and inference. We have tested our system on campus with over 100k data measurements collected by 36 students through 18 days. Our real-world experiments show that UbiAir is a light-weight, low-cost, accurate and scalable system for fine-grained air quality monitoring, as compared with other solutions.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computer systems organization** → *Client-server architectures*;

Additional Key Words and Phrases: sharing economy, mobile crowdsensing, Internet of things, air quality monitoring, urban computing

ACM Reference Format:

Di Wu, Tao Xiao, Xuewen Liao, Jie Luo, Chao Wu, Shigeng Zhang, Yong Li, and Yike Guo. 2020. When Sharing Economy Meets IoT: Towards Fine-grained Urban Air Quality Monitoring through Mobile Crowdsensing on Bike-share System. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 61 (June 2020), 26 pages. <https://doi.org/10.1145/3397328>

Author's addresses: D. Wu, School of Design, Hunan University, China; T. Xiao, X. Liao, and J. Luo, Department of Computer Engineering, Hunan University, China; C. Wu, School of Public Affairs, Zhejiang University, China; S. Zhang, School of Computer Science and Engineering, Central South University, China; Y. Li, Department of Electronic Engineering, Tsinghua University, China; Y. Guo, Department of Computing, Imperial College London, United Kingdom. This work is supported by the National Natural Science Foundation of China under Grant No. 61972145, 61932010 and 61772559, the National Key R&D Program of China under Grant No. 2019YFB1405703, and the HuXiang Youth Talent Program under Grant No. 2018RS3040. D. Wu is the corresponding author (Email: dwu@hnu.edu.cn).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2474-9567/2020/6-ART61 \$15.00

<https://doi.org/10.1145/3397328>

1 INTRODUCTION

Ambient (outdoor) air pollution is a major environmental health problem affecting everyone in low, middle, and high-income countries. The World Health Organization estimated in 2016 that outdoor air pollution in both cities and rural areas causes 4.2 million premature deaths worldwide per year [27]. Air pollution occurs when harmful or excessive quantities of substances including gases, particles, and biological molecules from anthropogenic (man-made) or natural sources are released into the atmosphere and dispersed through ambient air. Various locations, activities or other factors are responsible for generating pollutants into our ecosystem. Therefore, efficient and ubiquitous air quality monitoring solutions are greatly needed to effectively infer and control air pollution, especially in highly-populated urban areas.

1.1 Motivations

Although the government-led monitoring is able to measure the pollution near the static stations, it cannot accurately reflect actual air quality people breathe in, which is much more valuable to our health and daily lives. Low-cost stationary sensors and human-carried sensors have been adopted recently to monitor micro-scale urban air quality in client-cloud systems such as AirCloud [4] and Mosaic [11], where ambient pollution data are collected by the mobile client side and regional air quality status are inferred on the cloud server side. However, these exiting solutions are still lacking in terms of coverage and scalability (see Section 2). Therefore, new designs are needed to achieve high coverage and high accuracy while maintaining low cost. Given these considerations, mobile crowdsensing [16, 22, 45], an ubiquitous approach to outsource or share sensing tasks among workers, could be a promising solution to achieve light-weight, fine-grained, and scalable air quality monitoring. With the advent of portable sensing and computing devices (*i.e.*, Dylos and Aeroqual), crowd workers [13, 32] are able to carry small-size air quality monitoring equipment to measure the main pollutants in their surroundings. Nevertheless, crowdsensing air pollution still faces the following two major problems: (1) Commercial off-the-shelf devices usually can only detect one type of air pollutant. To monitor various air pollutants at the same time, crowd workers need to carry additional devices, which is a burden for them. (2) Air quality data at specific locations are usually required for some inference models. These locations may be inconvenient for crowd workers, and sometimes these specific locations may be difficult to access via mobile vehicles.

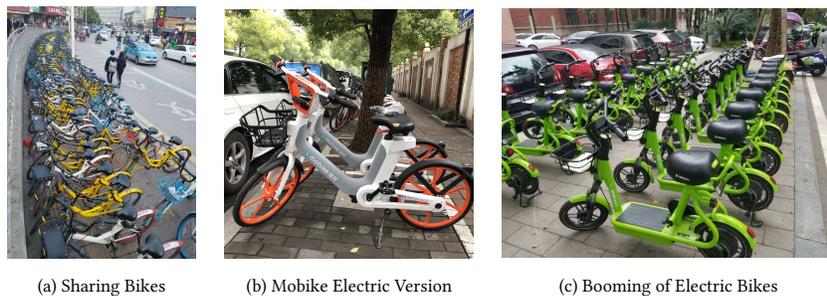


Fig. 1. Bike-sharing and its popularity.

The emergence of sharing economy [29, 34, 39] offers new collaborative forms of consumption, production, finance, and learning to facilitate mobile crowdsensing of urban air quality monitoring. Sharing economy focuses on the sharing of underutilized assets, monetized or not, in ways that improve efficiency, sustainability, and community. With the recent development of various sharing facilities (*e.g.*, sharing bikes, sharing scooters, sharing cars) [12], these mobile platforms provide a natural, scalable, and ubiquitous solution to carry various IoT devices,

and assist people to collect air pollution data at specific locations. For example, bike-sharing, especially dockless bike-sharing, is booming all over the world, and there are more than ten bike-sharing brands (e.g., Mobike, ofo, Hellobike) in China, as shown in Fig. 1 (a). On one hand, as an environment-friendly approach, bike-sharing offers people a convenient way to commute and solves the “last mile” problem. By allowing users to access and park bikes at any valid place, dockless bike-sharing has gained significant popularity. For instance, Mobike has nearly 8.65 million daily active users among 200 million registered users, and operates over 8 million bikes in China and abroad [23]. On the other hand, current bike-sharing is intended for short-term use at a very low price or free. Therefore their business model is usually in doubt and can cause serious financial problems. For example, ofo, a Chinese bike-sharing giant, now suffers from consistently high operation costs and lack of additional funding to expand its business. In this work, we propose to integrate the sharing economy with mobile crowdsensing on IoT platforms, which offers a feasible and sustainable solution to support fine-grained air quality monitoring.

1.2 Challenges and Our Approaches

With respect to the pervasive usage and fast evolution of sharing bikes, as shown by the recently launched electric version and its popularity in Fig. 1 (b) and (c), we present a fine-grained air quality monitoring system, namely *UbiAir*, through mobile crowdsensing on shared bikes. The main challenges related to the system design and implementation are: 1) Unlike the sparse and stationary governmental stations that only provide macro-scale and coarse-grained measurements, *UbiAir* emphasizes on ambient pollution and correspondingly air quality monitoring has to be fine-grained. This implies that geographically-dense measurements are needed on micro-scale size to guarantee high coverage and accuracy, and the measurement network structure can scale to urban scenarios with light weight and low cost. 2) It is normal that data collection errors occur in mobile and distributed sensing. This is especially challenging when a large number of shared bikes are employed during mobile crowdsensing, given that the IoT device carried on each bike is composed of low-cost air pollution sensors with accuracy variations. These errors not only make the dynamic measurements on the client side unable to reflect the true concentration of different air pollutants, but also affect the performance of the air quality inference model on the server side. 3) The operation of efficient air quality monitoring requires specific designs for a concise and practical inference model running on the server side. Ideally, we would like to assign shared bikes to collect air pollution data intensively over the predefined measurement network structure. In reality, there are still a small number of regions that bikes or workers cannot access. Given the spatially distributed data points, how to exploit the collected data to infer the air quality status on the data points without measurement remains an critical issue that the inference model needs to resolve.

UbiAir has been designed and implemented as a holistic system to tackle above challenges through mobile crowdsensing. Specifically, with regard to the convenient usage of dockless shared bikes for short and medium distance trips, *UbiAir* deploys a small-size air quality monitoring box composed of different low-cost pollution sensors on each shared bike. The boxes attached on bikes can be switched on/off by crowd workers (cyclists) to collect air pollutant data through a light-weight mobile application, where user experience has been considered to avoid inquiring the worker’s original trip plan. By doing this, mobile crowdsensing on shared bikes can reach most urban areas due to the broad scope of biking paths from different workers. In addition, we propose a scalable scheme to partition the measurement network into a grid structure, as well as an incentive strategy to motivate workers to collect data at the most informative grid points. To improve measurement accuracy, the collected data can be calibrated separately via both hardware and software, and the mobile application can geotag and upload these calibrated data as sampling information to the server for air quality inference. Since most of the natural environmental data could be approximated by a mixture of Gaussian distributions, *UbiAir* chooses the Gaussian kernel function to implement a Bayesian compressed sensing (BCS) model to infer the sampling data and reconstruct the air quality map with micro-scale and accurate results.

The major contributions of the UbiAir system are summarized as follows:

- A mobile crowdsensing framework to monitor micro-scale air quality on shared bikes capable of detecting different air pollutants, with specific designs on data measurement, data inference, and incentive scheme.
- A scalable and adaptive data collection method with a series of practical processing to achieve high coverage and accuracy, including grid partitioning, data calibration, and differential rewarding strategy.
- An efficient air quality inference model utilizing Bayesian compressive sensing to learn sampling data, and reconstruct fine-grained air quality map covering different regions in an urban area.
- A light-weight and user friendly mobile application to perform data collection, and assist the mobile crowdsensing workers while keep their normal biking behavior intact.
- Extensive real-world evaluation on campus with over 100k air quality data measurements collected by 36 students riding the shared bikes through 18 days, and these students were involved in the mobile crowdsensing process by our incentive scheme.

The rest of this paper is organized as follows: Section 2 introduces the related works. Section 3 presents our system overview. Section 4 explains our data sampling method. Section 5 elaborates our data calibration scheme. Section 6 describes our data analysis model. Section 7 evaluates system performance. Section 8 discusses with users on their experience. Section 9 concludes our paper.

2 RELATED WORKS

Different approaches exist for air quality monitoring. Related works have been proposed to address this issue and can be classified as follows.

2.1 Government-led Measurement Stations

Traditional air quality monitoring ways led by government, such as remote sensing [25] and stationary air quality monitoring stations [44], require enormous space, special resources, and economic investment. For example, a typical stationary station run by the government needs about 200,000 USD for construction and 30,000 USD per year for maintenance [44]. As such, government-led measurement stations are limited in scale. Without a significant number of monitoring sites, it is difficult to obtain fine-grained air quality information from active and populated regions that are geographically distributed in large-scale urban areas, which consists of stationary hotspots (e.g., construction and industrial facilities) and mobile hotspots (e.g., cars and trucks) of air pollution. The macro-scale data measured from the sparse and stationary governmental stations usually cannot reflect the micro-scale air quality status in these regions [11]. The accuracy is also a bottleneck in remote sensing due to weather and other factors. People could be misled by the publicly-available macro-scale and coarse-grained air quality data, without noticing pollutant emissions in their ambient environments, such as small particulate matter of 2.5 microns or less in diameter (PM_{2.5}), which may cause respiratory disease, childhood asthma, cancer, and other health problems.

In comparison with the sparse and stationary governmental stations that only provide low-quality measurements, UbiAir emphasizes on ambient pollution and achieves fine-grained air quality monitoring, where geographically-dense measurements are collected on micro-scale size to guarantee high coverage and accuracy. In addition, the measurement network structure of our UbiAir can scale to urban scenarios with light weight and low cost, which is impossible for government-led measurement stations.

2.2 Ambient Air Quality Monitoring

Some studies have been developed to monitor indoor air quality with special equipment and to monitor the air quality around people with wearable devices. MyPart [35] is a personal and portable particle sensor with a low

cost under 50 USD. It can distinguish and calculate particles of different sizes. W-Air [24] is a personal multi-pollutant monitoring platform for wearables. It adopts a sensor-fusion calibration method to recover high-fidelity ambient pollutant concentrations from the human emissions interference. AirCloud [4] is a client-cloud system for low-cost and stationary air quality monitoring, where two types of Internet-connected particulate matter monitors are designed with a mechanical structure for optimal air flow. AirSense [8] is an intelligent home-based indoor air quality sensing system. It is able to automatically detect pollution events, identify pollution sources, estimate personal exposure to air pollution, and provide reasonable suggestions to help user improve indoor air quality. MAAV [26] is a system to measure air quality, annotate data streams, and visualize real-time PM2.5 levels. It is able to collect data from multiple monitors, annotate the data, and send message when PM2.5 spike is detected. It also displays measurement data and annotations via an interactive tablet interface.

Although these devices or systems can monitor the air quality around the equipment, these monitoring devices are not placed on mobile vehicles. Using low-cost stationary sensors or human-carried sensors suffers the lack of coverage and scalability problem, therefore cannot achieve large-scale fine-grained air quality monitoring. Our UbiAir adopts the widely deployed shared bikes available and close to users as the mobile platform to mount monitoring devices. Furthermore, these shared bikes can collaboratively complete large-scale monitoring through distributed crowdsensing, with accuracy guarantee by grid partitioning, data calibration, and differential rewarding strategy.

2.3 Fine-grained Air Quality Monitoring

Recently, various studies have been attempting to get fine-grained air pollution data by low-cost sensors. Two sensing models were designed to obtain fine-grained air quality data in [5], where one model was for public transportation infrastructure with fixed and reliable routes along high-capacity corridors, and the other model relies on personal sensing devices in cars. Mosaic [11] is a low-cost urban air quality monitoring system based on mobile sensing. The Mosaic-Nodes was built with a constructive airflow disturbance design and GPS-assisted filtering based on an adjusted airflow structure. In these studies, the sensing devices are carried by city buses or deployed in cars. However, due to the geographical constraints (*e.g.* residential area, narrow road, mountains, lakes), these public vehicles can only measure a limited number of locations of the entire monitoring space. On the other hand, public vehicles are driven on a fixed route on the road, still suffering the lack of coverage and resulting in limited data collection. To solve these problems, we propose a mobile crowdsensing framework in UbiAir that is capable of assigning shared bikes to collect air pollution data intensively over the predefined measurement network structure, with specific designs on data measurement and incentive scheme.

To avoid exhaustive measurements over the entire monitoring area, some studies used a data inference model to approximate the value of unmeasured area. The inference model used in [44] is based on a few public air quality stations, meteorological data, road networks, taxi trajectories, and point of interests. Third-Eye [20] was developed as a mobile application that can utilize mobile phones to monitor PM2.5 with high-quality. It also designed a inference model that is based on web crawling and large-scale data set of the outdoor images taken by mobile phone. Although these multi-source data based inference models are able to effectively solve above problems that only limited number of measurements can be collected in the entire monitoring space, different types of data besides air pollutant data are needed in these models and some types of these data are not easy to collect without governmental resources. In reality, the operation of efficient air quality monitoring requires specific designs for a concise and practical inference model running on the server side. Therefore, an efficient air quality inference model has been deployed in our UbiAir utilizing Bayesian compressive sensing to merely learn sampling air pollutant data, and reconstruct fine-grained air quality map covering different regions in an urban area.

2.4 Air Quality Monitoring Using Bikes

As the route of public transportation (e.g. bus, taxi, metro) is fixed, bikes as a more agile mobile sensing platform have been used for ambient data construction. BikeNet [6] was proposed for cyclist experience mapping leveraging opportunistic sensor networking principles and techniques. The system is focused on quantifying cyclist performance, but cyclist-environmental information are collected on the trajectory to infer the performance. The emerging urban infrastructure have been studied in [10] to identify shared bicycling behaviors across stations and show how these behaviors relate to temporal and spatial dynamics of a city. These have led several works to address air-quality monitoring using bikes as the mobile platform, such as Canarin II [1], CyclAir [31], and Aeroflex [7]. These works focused on environmental data rather than behavior data, and all have designed their own prototype bikes with the similar implementation approach. That is, individual bikes equipped with air quality measurement devices collect ambient data and transfer it to the back-end database to provide the public with air quality information on the tracks that the bicycles pass by. Air pollution monitoring using public bicycle infrastructure was addressed in [21] as well by mounting the sensing box on a public bike. However, this kind of docked shared bicycle needs to be returned to the cycling station, and the riding trajectory is usually very limited, which cannot effectively cover large monitoring area.

The existing air quality monitoring works using bikes can only monitor the data either on the individual bike's trajectory or limited area by several docked public bikes. On contrast, our UbiAir system can cover entire monitoring area using mobile crowdsensing combined with the sharing economy paradigm on dockless shared bikes. Moreover, due to the lack of design on incentive mechanism, existing works cannot enroll and maintain enough number of bikes and volunteers to consistently contribute sufficient data for long-term air quality monitoring. The red envelope incentive mechanism proposed in our work can solve this problem to a certain extent. Additionally, except the work mentioned in BikeNet [6], almost all the existing works with mounted sensor box on bikes have not addressed the sensor calibration issue along the bicycling trajectory, which may results in data collection deviation from the actual values and provisioning the incorrect air quality information to the public. On contrast, our UbiAir system specifically tackles this issue by specific designs on software and hardware calibrations.

3 SYSTEM OVERVIEW

UbiAir is designed within a mobile crowdsensing framework and implemented as a novel client-server system following the sharing economy paradigm. As shown in Figure 2, the framework of our system consists of three major modules: data sampling, data calibration, and data analysis. These modules are interconnected and dependent on each other to form an organic system.

3.1 Data Sampling

Our main objectives in data sampling module are to incentivize users to be involved in the mobile crowdsensing process and collect the ambient air quality data at or near the desired locations by the on-demand assignments from our measurement network structure.

- **Grid Partitioning.** UbiAir deploys measurement network with scalable grid structure where masses of users having different trajectories ride their bikes across the grids and corresponding data sampling is performed at the grid points. The system server splits the urban areas that need to be monitored into dense grid cells, and the size of grid cells can be adjusted adaptively upon the demands on measurement accuracy. The grid partitioning method can satisfy the requirements of our air quality inference model on acquiring spatially distributed data as well.
- **Red Envelope Incentive.** We have designed an incentive scheme that mimics the *Red Envelope*, a monetary gift which is given during holidays or special occasions in Chinese and other East Asian and Southeast

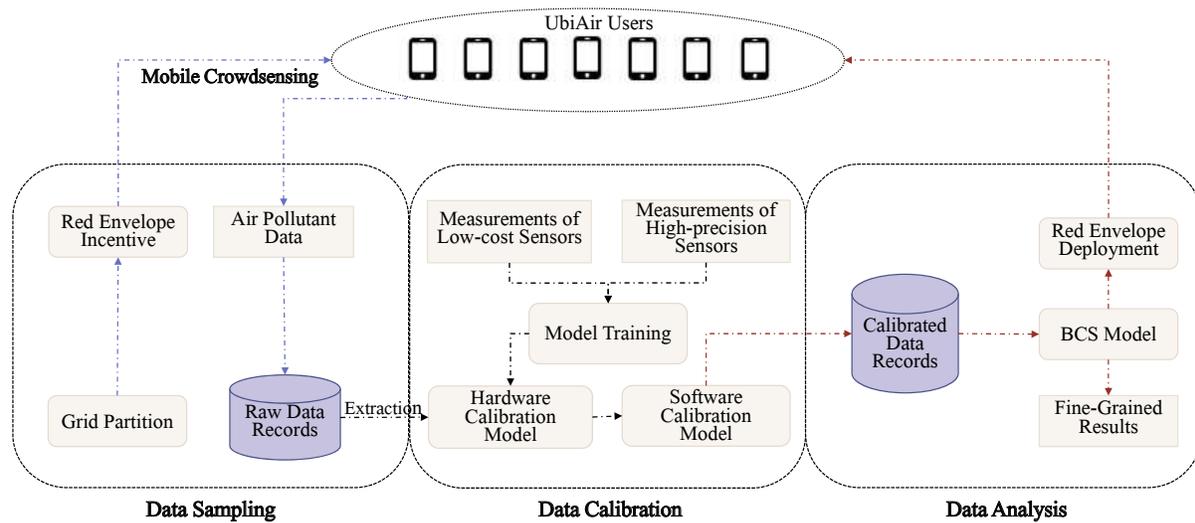


Fig. 2. System overview of UbiAir.

Asian societies. Our scheme encourages users to grab red envelopes distributed over specific grid points, accompanying with tasks to collect air quality data at these positions. The bonus amount of each red envelope is determined by a differential rewarding strategy, through evaluating the information gain and the difficulty of data collection at corresponding position. Generally, more data collected with high-coverage biking trip, more rewards grabbed with incremental red envelopes.

3.2 Data Calibration

The air pollutant data crowdsensed from the data sampling module are recorded as raw data in UbiAir client on sharing bikes. These raw data are classified with their pollution types and labeled with their geographical information, and will be uploaded to the backend server via the UbiAir mobile application. The data calibration module extracts the key information and converts the raw data into more accurate data through our hardware calibration model and software calibration model, sequentially.

- **Hardware Calibration.** Because of the inherent errors among different sensors, the UbiAir server first runs training process over the measurements of low-cost sensors and the measurements of high-precision sensors, and derived a hardware calibration model through learning their errors. The trained model can be used to improve the accuracy of the collected data and make the calibrated data close to the ground truth, from the hardware aspect.
- **Software Calibration.** The air quality data cannot be always collected at the expected sampling positions (grid points) exactly, some of which could be sensed and recorded at nearby places passed by the sharing bikes. We tune sensor's sampling frequency and choose redundant sampling data around the grid point without measurement. These samples along the biking trajectory are used to estimate the concentration of air pollutant at specific grid point statistically. By doing this, UbiAir can get data from all the desired grid points, which have been well calibrated to be ready for our air quality inference model.

3.3 Data Analysis

The calibrated data at grid points are extracted from its corresponding database and used as the input to the Bayesian compressive sensing (BCS), which is deployed as our inference model to reconstruct the air quality map in urban areas.

- **BCS Model.** The BCS technology provides a method to recover data from a number of measurements much less than that required by the Nyquist-Shannon theorem [40]. Given the grid structure for data collection, when the amount of sampling data is sufficient to meet the basic requirement of BCS model, this inference model can get fine-grained results by recovering the air quality data of entire monitoring area with high precision. Otherwise, the model will generate missing grid points that need further measurements.
- **Red Envelope Deployment.** The data at different positions has different effects on the recovery result of the BCS model. We want to collect data at positions that minimize the recovery error. UbiAir proposes an adaptive BCS method to assess the information gain of these positions and rank them accordingly. Once specific grid points are selected, users are expected to be dispatched to collect data at these positions. Correspondingly, the red envelopes will be deployed on these important positions to reward users that finish these tasks.

More details about the design and implementation of above modules and their operations will be explained in following sections.

4 DATA SAMPLING

To design a fine-grained air quality aware system, the first thing we have to do is to get the data in a reasonable way with tradeoff between practical deployment and effective performance. Here we propose a grid partitioning method to sample air pollutant data on-demand. Furthermore, a red envelope incentive scheme is designed to encourage users to participate in mobile crowdsensing, and user experience has been considered to avoid interfering their biking plan.

4.1 Grid Partitioning

Considering the sharing bike users usually have different destinations and various biking paths, it is difficult to determine whether the user will pass the sampling position we are interested in. Given this situation, we have designed an effective way to sample air pollutant data following a feasible grid network structure.

Specifically we split the space to be monitored into uniform grid cells by horizontal and vertical lines. In general, the density of the grid cells is determined by the spatial precision of the air pollutant data we want to reconstruct. The advantages of splitting the space to be monitored into uniform grid cells are: 1) achieving on-demand data collection, where users are preferred to collect data at our specified grid points (sampling positions); 2) making the collected data more evenly distributed in space, not concentrated in a small piece of area; 3) facilitating the data recovery of our inference model, where the Gaussian basis matrix $B_{n \times n}$ in the adaptive BCS algorithm can be easily processed, as explained in Section 6.2.

The data sampling process using grid partitioning is illustrated by an example in Figure 3, where the monitoring space is split to form a grid structure map with resolution of 9×9 grid cells. Initially, some red envelopes are generated and deployed at specific grid points selected by our adaptive BCS method, as shown in Figure 3 (a). The crowd workers can collect data and get reward from the red envelopes in two ways: one is that the user takes the initiative to intentionally ride to the red envelope position and finish data collection task, following the black solid arrow; the other is that the user's biking path just passes through a red envelope position and the data is collected along the biking trajectory automatically, as depicted by the dashed line. Figure 3 (b) presents the situation after different users have collected the data and grabbed corresponding red envelopes. Then the original red envelope marks are replaced by mission completion marks. When the data collection at all the red

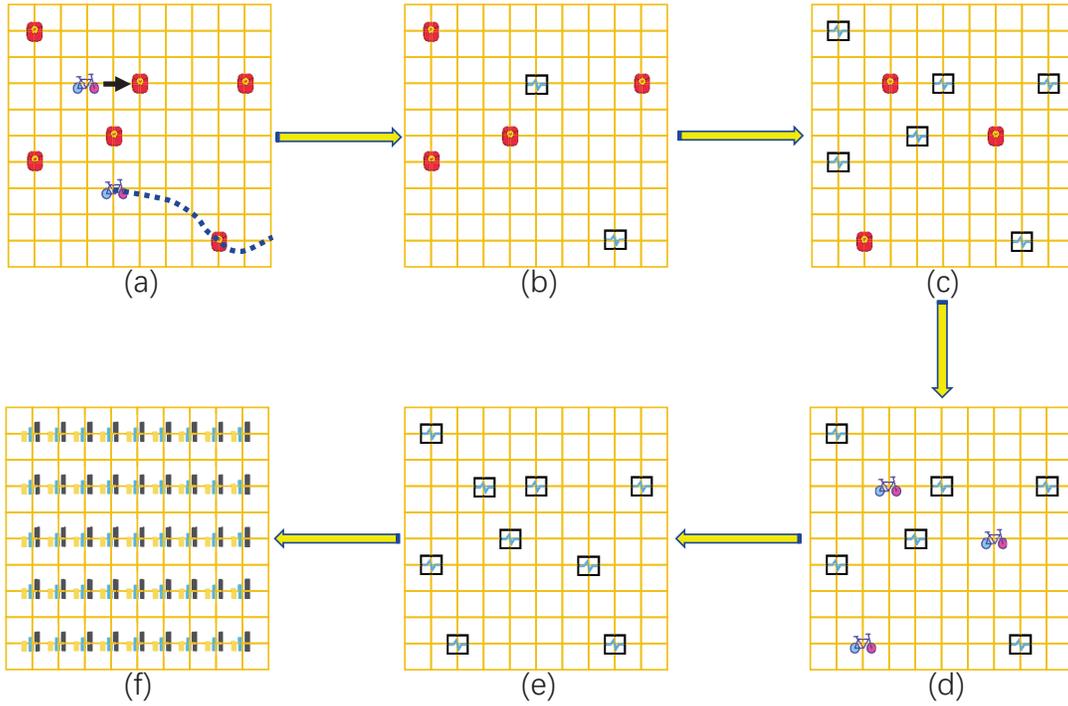


Fig. 3. Grid partitioning and data sampling.

envelope positions is completed, a new round of red envelope distribution is generated in Figure 3 (c) to place rewards at other grid points requiring data sampling, and corresponding data collection is performed in Figure 3 (d). UbiAir repeats this red envelope generation and data collection process, as shown in Figure 3 (c) and (d), until meets the requirement of its BCS model on acquiring the expected amount of data, as depicted in Figure 3 (e). Then, these sampling data collected at partial grid points will be used as input data for the BCS model to reconstruct the air quality data covering the entire grid map, as illustrated in Figure 3 (f).

4.2 Red Envelope Incentive

Considering some users may be reluctant to participate in the data sampling process, we encourage users in a paid way with very small amount of bonus. The virtual red envelopes developed in WeChat, a multi-purpose messaging, social media and mobile payment app developed by Tencent, has proved that it is an effective way to draw instant attention and keep user's adhesiveness [14]. This inspired us to adopt the idea of virtual/digital red envelopes in our incentive scheme, but with specifically new design on distribution and rewarding for mobile crowdsensing, to motivate users to collect data at crucial grid points for air quality sampling.

The red envelopes are generated and deployed at specific grid points where their information gain are ranked in the top k . The value of k can be customized by the UbiAir system. The calculation of the information gain of grid points by our BCS model is detailed in Section 6.2. After red envelopes are distributed over the prior grid points, the amount of bonus of the red envelope $R_{i,j}$ at grid point with coordinates (i, j) is modeled in Eq. (1):

$$R_{i,j} = \delta I_{i,j} + \epsilon T_{i,j}^2 \quad (1)$$

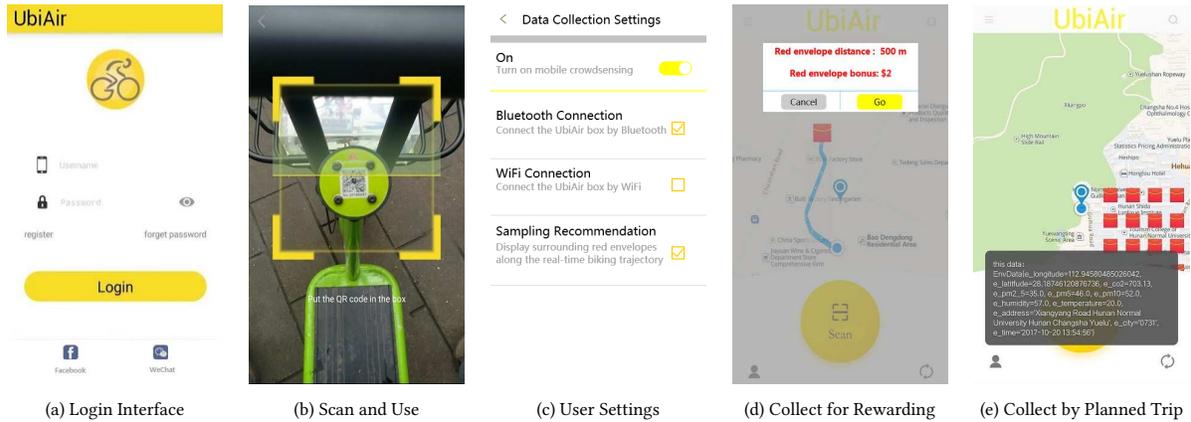


Fig. 4. The UI of UbiAir mobile application.

where $I_{i,j}$ is the information gain of deploying the red envelope at coordinates (i, j) . $T_{i,j}$ is the duration that the red envelope has been retained at coordinates (i, j) since its generation. δ and ϵ are constant parameters, and their value is set according to the amount of money we can afford.

Intuitively, if the data at some grid point positions have not been collected after some time, it tends to be difficult to collect these data. To guarantee the timeliness of finishing data collection over the whole grid map, UbiAir should reasonably increase the bonus amount of these special red envelope to match their difficulty levels. Within the tolerant amount of bonus, we set a bonus threshold U . When $R_{i,j} > U$, we consider the position of red envelope too remote or unreachable in space (such as a red envelope in a lake) and remove this red envelope from the grid map. After that, UbiAir system will generate a new red envelope at the $(k + 1)$ th grid point where its information gain is ranked $(k + 1)$ th among all the grid points. According to this rule, we can finally collect data from k grid points.

4.3 User Experience

We have implemented the UbiAir mobile application including the above-mentioned designed functionalities on the mobile Android platform. The backend service is running on Alibaba Cloud with the server implemented in Python/MySQL. The mapping, navigation and location-based services used in UbiAir are provided by AutoNavi, which is a major provider of map information in China. UbiAir is committed to creating a light-weight and user friendly mobile application that performs efficient data collection, and assist workers being involved in the mobile crowdsensing while keep their normal biking behavior intact. The operation process of crowd workers has been carefully designed in UbiAir and simplified as much as possible to deliver a great user experience.

Figure 4 illustrates the user interface (UI) of UbiAir displaying the main functions of the system, and its usability design. Figure 4 (a) shows the login page of the system. Users can choose to log in to UbiAir with their authorized identification; they also can log in with accounts from other popular social networks (e.g. WeChat, Facebook). After authorization, UbiAir guides the users to unlock their dockless sharing bikes that carry the UbiAir box composed of various air pollutant sensors. As shown in Figure 4 (b), this step can be easily done by scanning the quick response (QR) code within the suggested scanning area provided the rear camera of mobile phone. Subsequently, UbiAir automatically pops up the UI of data collection settings to encourage the users to be involved in its mobile crowdsensing, as illustrated in Figure 4 (c). The users can opt to turn on/off the UbiAir box for mobile crowdsensing during their biking trip. Once data collection is enabled, the users need to choose a

wireless communication method to connect their mobile phone with the UbiAir box (e.g. Bluetooth, WiFi), so that the collected data can be temporarily stored in the phone during the biking trip and later uploaded to the server.

In addition, UbiAir present an option in Figure 4 (c) to recommend the data sampling positions to users by displaying surrounding red envelopes along the real-time biking trajectory. Once this option is selected, the distance and bonus of an nearby red envelope will pop up and navigate the users to collect corresponding data at the specific grid point according to their current location, as shown in Figure 4 (d). By doing this, the users are considered to be more active in data collection with intention to grab the red envelopes through adjusting their trip plan. Otherwise, for the users that would like to join the mobile crowdsensing but hope their original biking plan intact, UbiAir also presents a rewarding solution for them without affecting their biking experience, as shown in Figure 4 (e). By turning on data collection in Figure 4 (c), the user will get a view of the map showing the positions of all the red envelopes in Figure 4 (e), so that the biking path to destination can be planned beforehand, with inclination to grab some red envelopes. Once the UbiAir box attached on sharing bikes passes the position of a red envelope and samples its data, corresponding bonus will automatically deposit into the user's account. We show the details of uploaded data information in Figure 4 (e), but in reality these information will be hidden from the user. Note that in both Figure 4 (d) and (e), there is an scan button for users to reconnect their mobile phone with the UbiAir box once the previous wireless communication is interfered and disconnected.

5 DATA CALIBRATION

The emerging of cheaper, smaller and more portable sensors makes large-scale, ubiquitous monitoring of air quality possible. However, many low-cost sensors lack sufficient accuracy on data measurement, resulting in bias on hardware-dependent sampling precision and low stability vulnerable to the dynamic changes of environmental conditions. Thus, data calibration from both hardware and software aspects is essential to the UbiAir box composed of low-cost sensors.

5.1 Hardware Calibration

Hardware calibration is performed to address the low precision issue and tune the sampling bias of low-cost sensors before their deployment in real world. The two-phase learning [18] as a typical approach to calibrate low-cost sensors is adopted and extended in UbiAir for urban air quality measurement, where the measurements of low-cost sensors are trained to obtain corresponding model parameters, which are used to enable the calibration measurement \hat{c} close to the reference measurement c_r (true value).

To obtain a well trained model, the relationship between the measurements of low-cost sensors and the expected calibration measurement is decomposed into a linear part and a non-linear part. The linear part is trained in the first phase by multiple least square (MLS) method and the non-linear part is trained in the second phase by random forest (RF) approach, respectively. During the two-phase learning process, overfitting may happen due to the inclusion of inappropriate features before training the model. To avoid this, we use an automatic feature selection algorithm based on the Akaike information criterion (AIC) [28] to select proper features during the linear training process. The final calibration measurement can be expressed as Eq. 2.

$$\hat{c} = \text{linearpart}(c) + \text{nonlinearpart}(c) = \beta_0 + \beta_1 c_1 + \beta_2 c_2 + \beta_3 c_3 + \dots + \beta_u c_u + f(c) \quad (2)$$

where $\beta_0, \beta_1, \dots, \beta_u$ are calibration coefficients consisting of an intercept β_0 and u different kinds of features selected by the automatic feature selection. A vector c is composed of raw sensor measurements.

In the first phase (linear part) of Eq. 2, the least squares regression model is used to determine the values of $\beta_0, \beta_1, \dots, \beta_u$. In the second phase (nonlinear part), we first calculate the residual error, that is the difference between the calibration measurement of the first phase and the reference measurement. Then we use the RF approach to learn the complex relationship between all available features and the residual error, which is

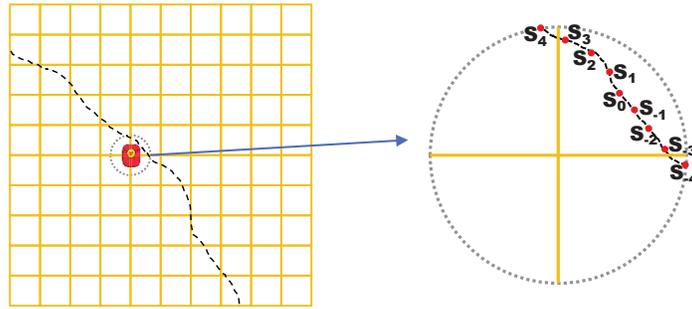


Fig. 5. Software calibration.

represented by $f(c)$ in Eq. 2. Through these operations, a well trained model for hardware calibration can be derived via dynamically adjusting the calibration measurement and approaching the reference measurement.

5.2 Software Calibration

The air quality data cannot be always directly collected at the expected sampling positions (grid points) during data collection process, due to geographical constraints, or crowd workers having no willingness to reach these positions. We divide these expected grid points without direct measurements into emission source points and non-emission source points. The emission source point refers to position where certain air pollutants are released, such as industrial factories. The non-emissions source point does not generate pollution but may be impacted by the dispersed air pollutants emitted from other places. The software calibration is proposed to approximate the measurement of air quality data at these specific grid points where the sharing bikes have a high probability to pass around with certain distance along their trajectories.

Here, we introduce a prior knowledge that most environmental signals (such as gases, particles, and biological substances) follow a molecular diffusion process. That is, the amplitude of the signal is smoothly attenuated from source to tail, which has the same property as the Gaussian distribution. This characteristic can be used to determine whether the position is the source of emissions. In general, if there is no emission source at a grid point, the concentration of sampled air pollutants near the grid point does not change much. Otherwise, the concentration of sampled air pollutants near the emission source gradually decreases as the distance from the source increases. In our software calibration, for the grid point with emission source, we estimate its air quality status with a series of data sampled at different positions around it by Gaussian averaging, where the sampling position closer to the emission source has a higher weight. For the grid point with non-emission source, data sampled at different positions around it are not much different, so we estimate its air quality status by arithmetic averaging.

As shown in Figure 5, we determine a sampling circle centered at a specific grid point without direct measurement. When the sharing bike carrying a UbiAir box with pre-tuned sampling frequency enters the circle, a series of data sampled at different positions along the biking trajectory in this circle can be used to estimate the concentration of air pollutants at the specific grid point statistically. We set the sampling position closest to the grid point as S_0 . As for a sharing bike, from its entering point to its leaving point of the sampling circle, its UbiAir box run data sampling at stable interval according to the pre-tuned sampling frequency, resulting in a series of sampling positions. We evenly selected a number of sampling positions before and after S_0 , labeled as S_{-4} to S_{-1} , and S_1 to S_4 , respectively. For the 9 sampling positions, we obtain 9 sampling data correspondingly as a data set $S = (s_{-4}, s_{-3}, \dots, s_0, \dots, s_3, s_4)$. Based on the data set, the Gaussian averaging for emission source point X_{ESP}

and the arithmetic averaging for non-emission source point X_{NESP} are operated as follows, respectively:

$$X_{ESP} = \frac{\sum_i (G_i * s_i)}{\sum_i G_i}, \quad X_{NESP} = \frac{\sum_i s_i}{|S|} \quad (3)$$

where G_i is a Gaussian function which could be described as $G_i = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2}{2\sigma^2}}$. σ^2 is a constant parameter which can be set according to the practical requirements. s_i is the sampling data from the data set S obtained within the sampling circle. $|S|$ represents the number of sampling data in the data set S .

6 DATA ANALYSIS

After above data calibration operations, UbiAir can get sufficient sampling data from the desired grid points, which have been well calibrated to be ready for data analysis on the fine-grained air quality data reconstruction, and the calculation of information gain at grid point. We illustrate the key approaches in this section.

6.1 Data Reconstruction Using BCS

Given all the pollutant data collected at the grid points of our monitoring region, these values can be aggregated into an n by 1 vector X_r . X_r represents the instant pollutant distribution in this region. The goal of air quality monitoring is to recover X_r with a limited number of measurements by the sensors of UbiAir box. The reconstruction of the pollutant data X_r is then solving a linear regression problem as:

$$Y_{m \times 1} = \Phi X_r + e_n = \Theta_{m \times n} w_{n \times 1} + e_n \quad (4)$$

where $Y_{m \times 1}$ is an m by 1 vector ($m \ll n$) that stands for the sensor measurements, $\Theta = \Phi B$ is the projection matrix, B is a fixed Basis matrix, Φ is the sampling matrix, w is the sparse weights to be estimated, and e_n are the zero-mean Gaussian distributed noises in the measurements. The measurement/sampling matrix Φ represents the positions of red envelopes. Each row in the sampling matrix Φ is exactly a unit vector with only one non-zero element in it. In this way, ΦX_r is an m by 1 vector composed of the collected values of the sensors at the red envelope positions.

It is reasonable to assume that in our grid partitioning most environmental data are sparse under Gaussian Kernel basis B . Compressive sensing provides us a method for estimating sparse solutions to underdetermined linear regression [2, 3]. Its algorithm reconstructs the environmental data as $X = Bw$. The Gaussian basis matrix is defined as: $B = [\Psi(X_{r1}) \Psi(X_{r2}) \cdots \Psi(X_{rn})]^T$, where $\Psi(X_{ri}) = [K(X_{ri}, X_{r1}) \cdots K(X_{ri}, X_{rn})]$ and $K(X_{ri}, X_{rj})$ is the predefined Gaussian Kernel function. We have $K(X_{ri}, X_{rj}) = \exp\left\{-\eta_1(X_{ri1} - X_{rj1})^2 - \eta_2(X_{ri2} - X_{rj2})^2\right\}$, where η_1 and η_2 are hyper-parameters of the kernel function, and the coordinates of X_{ri} is (X_{ri1}, X_{ri2}) .

Specifically, Bayesian compressive sensing (BCS) [15] is used as an inference model in UbiAir to estimate the sparse vector $w_{n \times 1}$ in Eq. 4, in which Bayesian models are applied to maximize the posterior probability of $w_{n \times 1}$. Generally speaking, our BCS recovery algorithm combines hierarchical sparseness priors for $w_{n \times 1}$ and e_n with relevance vector machine (RVM) based BCS inversion [36] to estimate $w_{n \times 1}$. Given $Y_{m \times 1}$ and $\Theta_{m \times n}$, we estimate α and σ_0^2 that are the hyper-parameters in Gaussian priors for $w_{n \times 1}$ and e_n as described in [15] by maximizing $P(w|y, \alpha, \sigma_0^2)$, and then the sparse vector w can be determined. Moreover, our BCS recovery algorithm employs a fast sparse Bayesian learning algorithm to improve the computational speed. The detailed processes in this fast algorithm can be referred in [37].

6.2 Adaptive BCS

BCS inference model provides the posterior density function for $w_{n \times 1}$ instead of a point estimate of w . This property enables us to adaptively estimate the optimal next projection to be added into the measurement matrix,

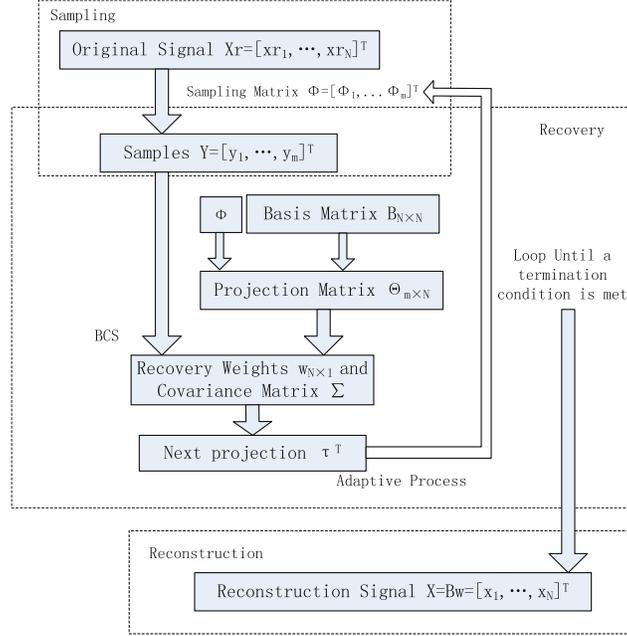


Fig. 6. Work flow of the adaptive BCS.

which facilitates data sampling selectively at the most informative positions rather than randomly. Here we propose an adaptive BCS approach in UbiAir to deploy red envelopes at the most informative positions to attract bike-sharing users to collect data.

6.2.1 Selecting Projections Adaptively. The sparse weights vector w is actually a multivariate Gaussian distribution with the mean μ and covariance matrix Ω [17]. A projection matrix Θ can be designed to minimize the differential entropy $h(X) = -\int P(X) \log P(X) dX$ for the reconstructed data $X = Bw$ [15]. To deploy a new red envelope is equivalent to adding a new row on the projection matrix. If we add a new projection τ on Θ , where τ^T is a new row, and we want to minimize the $h(X)$, it has been proven in [15] that the goal is equivalent to maximizing the $\tau^T \Omega \tau$ as:

$$\tau^T \Omega \tau = \tau^T \text{Covariance}(w) \tau \cong \text{Variance}(Y) \quad (5)$$

where the τ^T to be added into Θ represents the most informative measurement. $\tau^T \Omega \tau$ is equivalent to a measure of the “information gain” at grid point in our scenario.

Given the air quality monitoring problem shown in Eq. (4), the projection matrix $\Theta = \phi B$ is actually choosing rows from basis matrix B and we aim to choose the optimal row from B one by one to build the Θ and minimize $\tau^T \Omega \tau$. In this case, τ^T is a row in B . The measure of how informative τ^T is can be then described as follow:

$$\text{next_score}(i) = \tau^T \Omega \tau = a^T B \Omega B a = B_a^T \Omega B_a \quad (6)$$

where a^T is a 1 by n unit vector in which the i -th element is one and B_a^T is the i -th row of the basis matrix B .

Accordingly, our UbiAir uses the measurements at the positions of deployed red envelopes as the feedbacks to help guide the deployment of the next red envelope. Moreover, the adaptive BCS approach greedily takes the current estimated variances as the criterion to optimizing red envelope deployments.

6.2.2 *The Computational Model.* There are three phases in our environmental monitoring model, which are sampling, recovery and reconstruction, as shown by the work flow in Figure 6. The three phases are detailed as follows:

- 1) Sampling phase: UbiAir samples the air quality data with the sampling matrix, in which red envelope deployment information is contained. An initial red envelope deployment consisted of few random red envelope positions is generated to start the work flow shown in Figure 6.
- 2) Recovery phase: UbiAir estimates the sparse vector w and its covariance matrix with the BCS technique. Then the system calculates the $next_{scores}$ for all the rows in B that have not been in Θ so far, and revise the sampling matrix by adding k rows to Θ that corresponds to the top k $next_{score}$. The k rows indicate the next top k red envelope positions. UbiAir runs the sampling recovery loop until termination condition is met.
- 3) Reconstruction phase: The air quality data in the monitoring region can be reconstructed via a simple matrix multiplication $X = Bw$.

Note that in the adaptive BCS, the red envelope deployment phase and sampling phase are implemented simultaneously. That is, the adaptive BCS algorithm uses the data that has been sampled to find k new red envelope deployment positions. After obtaining these deployment positions, UbiAir incentivizes users to collect new data at these positions until enough sampling data are obtained to reconstruct the fine-grained results for air quality monitoring using our BCS inference model.

7 PERFORMANCE EVALUATION

We have conducted comprehensive experiments to evaluate the proposed UbiAir system. First, we evaluate the recovery accuracy of our system with the calibrated sampling data. Second, we evaluate the impact of three external factors on system performance; that is grid cell's density, sampling error and sampling time. Finally, we evaluate the effectiveness of the data calibration method and red envelope deployment method used in our system. In our experiments, the parameters η_1 and η_2 used in the BCS model are empirically set to 20, and the initial number of sampling positions is set to 30.

7.1 Data Sampling and Recovery

The UbiAir box as the sampling device in our system can be configured with six types of sensors to collect six different air pollutants at maximum, namely PM2.5 sensor, CO₂ sensor, PM5 sensor, PM10 sensor, temperature sensor and humidity sensor. The price of UbiAir box is around 20 USD. We have built and customized the UbiAir box that is easy to carry by the sharing bikes. These boxes can be charged either by a portable battery or the embedded power system of electric bikes, as shown in Figure 7. Note that the data collected by our UbiAir box are transmitted to participants' mobile phones for caching during the biking trip and later uploading to the server, so that the storage and communication costs in our UbiAir box can be reduced greatly to maintain at an acceptable level for large-scale deployment as a sharing IoT device.

7.1.1 *Participant Bootstrapping and Recruitment.* We verified our UbiAir system initially at our university, and the initial mobile crowdsensing participants involved in our system are mainly students on campus. Figure 7 (d) shows the map of experimental area, marked with the static monitoring stations and the grid. We started with a small number of students by explaining the red envelope incentive to them, as described in Section 4.2, and they would love to try the UbiAir application to obtain corresponding rewards. Later, we showed these initial participants the performance difference between the fine-grained monitoring by mobile crowdsensing and the coarse-grained monitoring by traditional static stations. For example, Figure 7 (e) shows the concentration of PM2.5 recovered by our UbiAir system and the concentration of PM2.5 measured by the nearest static monitoring station over ten consecutive days from the same monitoring area, respectively. We observe there is a certain performance deviation between the two methods, though their trend are basically the same. The measurements of

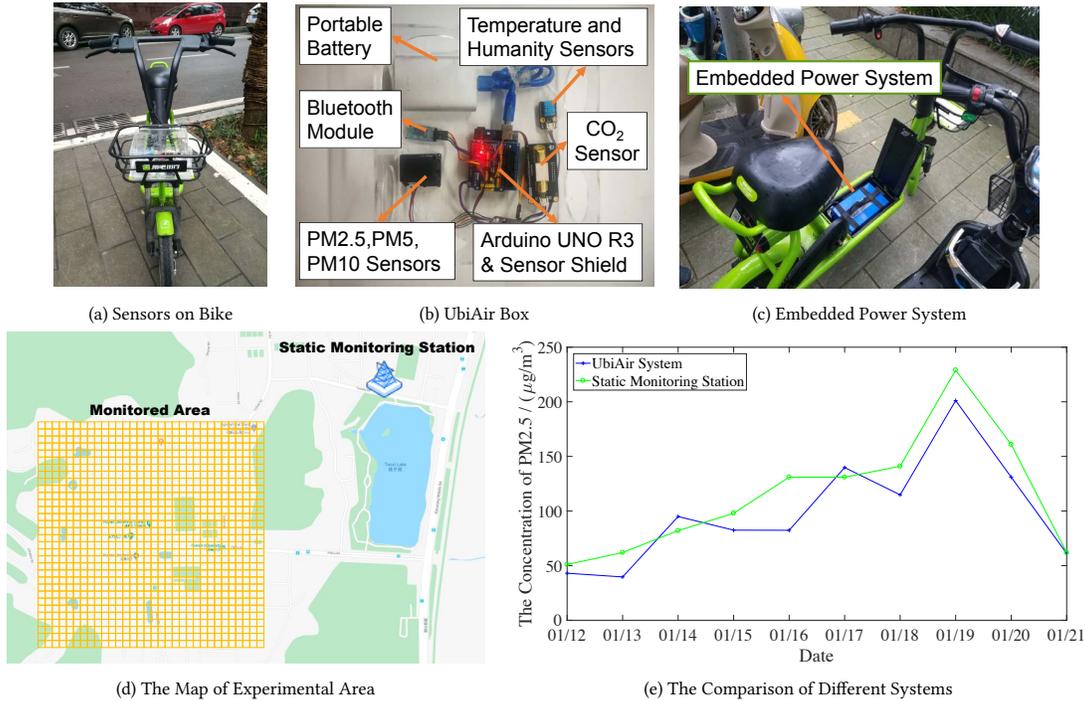


Fig. 7. The demo of sharing bikes carrying UbiAir box to collect air pollutant data.

UbiAir obviously fluctuate more than the static station along with the time; it shows that our platform is reliable to reflect the micro changes in ambient air pollution. Considering the air quality data measured directly by the static station can only use the data value collected at a fix position to represent the general air quality situation in its vicinity, the air quality conditions far from the static station may deviate from the coarse-grained measurement by the static station. On contrary, UbiAir system can use sampling data at grid points to reconstruct the distribution of air pollution concentration in details, therefore presenting a fine-grained air quality information.

After we showed these visual results to our initial participants, they agreed with the importance of micro-scale and fine-grained air quality monitoring, and keep using UbiAir as an ambient air quality alert. Also, these participants spread the positive feedbacks of our UbiAir to their friends, and encourages them to share and compare the rewards with each other. In addition, through the differential incentive schemes using red envelopes to improve user experience during mobile crowdsensing, as described in Section 4.3, UbiAir system is able to motivate users for participating and recruit sufficient crowd works to collect the expected sampling data at specific grid points with timeliness guarantee, which will be uploaded to the backend server with their geotags for data calibration and fine-grained air quality recovery.

7.1.2 Evaluation Results. In this set of experiments, we recruited 36 crowd workers by our red envelope incentive scheme to sample different air pollutant data over an urban area that has been split as a grid map to be monitored. The resolution of the grid map is 32×32 , which contains 1024 grid cells and the side length of each grid cell is around 30 meters. Figure 8 has presented both the sampling maps and reconstruction maps of four different air pollutants (CO_2 , $\text{PM}_{2.5}$, PM_5 , and PM_{10}). The sampling map shows the distribution of the sampling

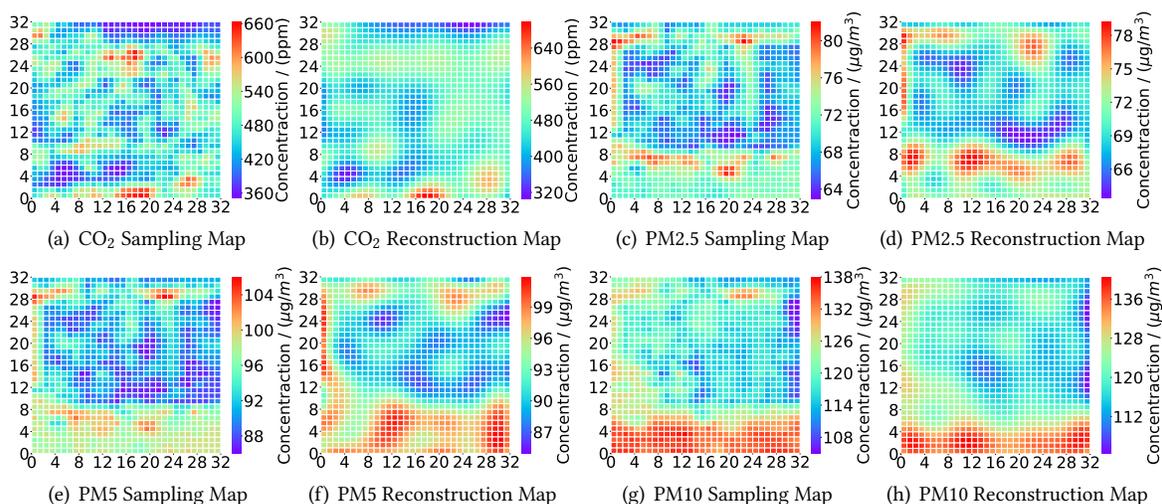


Fig. 8. Data sampling and recovery of different air pollutants.

data collected from all the grid points over the measurement map. The reconstruction map shows the fine-grained results using adaptive BCS approach in the same monitoring area, where a small number of measurements (100 sampling data) from the sampling map at specific positions are selected as the input to the BCS model for data reconstruction.

As we can see from all the figures depicted by two dimensional (2D) heatmaps, the data reconstruction map presents smoother color transition than the data sampling map, and the grid squares at the same position (*i.e.*, the same row and the same column) present similar colors among these heatmaps. The areas with high pollutant concentrations are marked with various red color and the deep red square indicates serious pollution level. This shows that the data construction can present fine-grained and stable results, because our adaptive BCS can select sampling data at the most informative positions and our data calibration scheme can improve the quality of sampling data for the BCS inference model. We calculate the reconstruction error with $\frac{\|X_r - X\|_2}{\|X\|_2}$, and the symbol meanings have been defined in Section 6. Our results show that the data reconstruction errors of CO₂, PM_{2.5}, PM₅, and PM₁₀ are 9.29%, 3.32%, 2.62% and 2.37%, respectively. These data reconstruction errors might change according to their collection time and pollution concentration, but all with an averaging performance below 10%. This further justifies that the UbiAir system can obtain satisfactory recovery accuracy.

7.2 Impact of External Factors

The density of grid cells, the data sampling time and the data sampling error are three important external factors that might influence the system performance. Here, we investigate these relevant issues using PM₅ as the air pollutant data to be collected and analyzed.

7.2.1 Density of Grid Cells. The recovery accuracy varies with the density of grid cells. Given the number of sampling positions is constant, when the density of grid cells is low, the recovery is more accurate because we get the information at most of these grid cells. Conversely, if the density of grid cells is high, the number of sampling positions is comparatively sparse to provide enough information to recovery the air pollutant data of the monitored area. The density of grid cells can be adjusted by the size of each grid cell. We choose a same region and divide it into grid cells by two density strategies, with resolution in the grid map of 25×25 grid cells

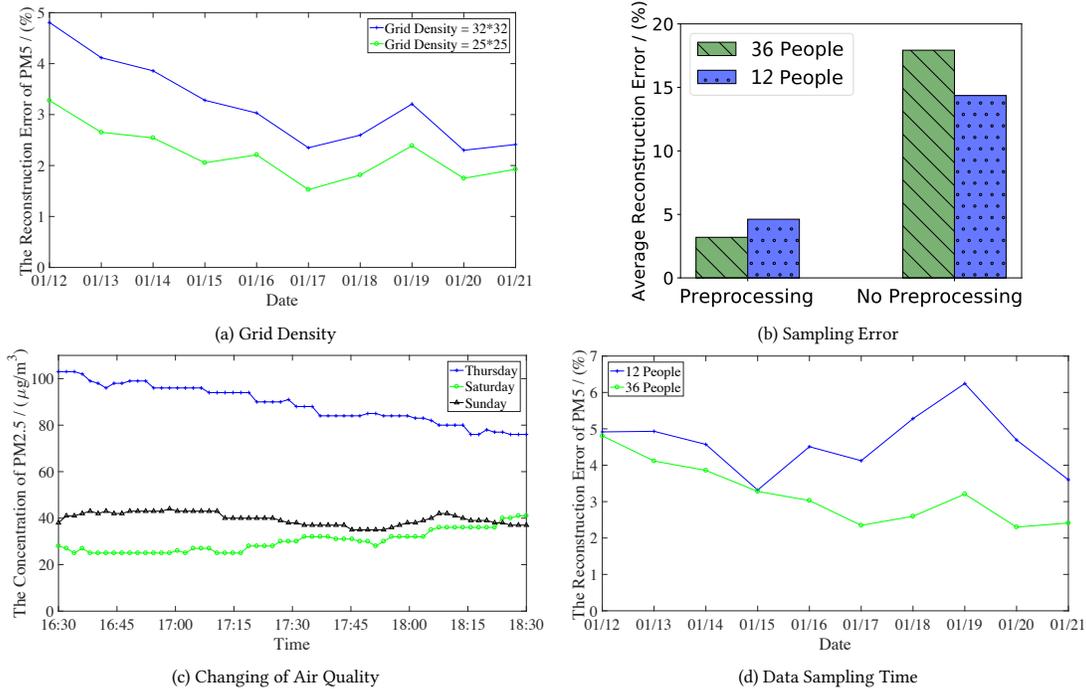


Fig. 9. The impact of external factors on reconstruction error.

and 32×32 grid cells, respectively. Then we recover PM5 as the target air pollutant data of the whole grid map with the calibrated measurements of 100 sampling positions for the two different densities.

The recovery accuracy for each case over ten consecutive days is shown in Figure 9 (a). We have observed from the results that the recovery accuracy of grid cells with low density are higher than that of high density. This confirms our intuition. Furthermore, Figure 12 shows that when the density of grid cell is constant, there are more sampling positions and the recovery effect is better. This also gives us some guidance on choosing a reasonable density to achieve expected recovery accuracy during mobile crowdsensing. When increasing the density of the grid cell, the number of sampling positions should be increased accordingly to ensure a certain degree of recovery accuracy, although this will increase the complexity of computation of our system. How to balance the complexity of the algorithm with the accuracy of recovery is a problem worth studying in the future.

7.2.2 Data Sampling Error. In this experiment, we have compared the average recovery accuracy of the uncalibrated PM5 data with the calibrated PM5 data for ten consecutive days. The comparison results are shown in Figure 9 (d). As can be seen from this figure, when the data is not calibrated, the recovery result that employing the group of 36 workers is worse than the recovery result with the group of 12 workers. This may not in line with our intuition that the sampling time of 36 people should be less than that of 12 people, so the recovery performance should be better. However, after data calibration, the opposite result occurs, and the result that sampling with 36 workers is better than sampling with 12 workers on recovery accuracy is what we expect by introducing data calibration into our system.

Considering that the number of sensors used by 36 workers is more than the number of sensors used by 12 workers, the sampling error caused by redundant sensors with various hardware precision is an external factor

that has a large impact on recovery accuracy. In addition, the impact of uncalibrated data from low-cost sensors usually has a greater influence on the recovery accuracy than the sampling time. Therefore the uncalibrated data from 36 workers result in lower recovery accuracy than that from 12 workers. After we refine the two sampling data sets with both hardware calibration and software calibration, the recovery accuracy of the 12 workers becomes worse than that of the 36 workers. The improved recovery accuracy verifies the effectiveness of our data calibration algorithms. It also confirms the impact of sampling time on recovery accuracy, that is, the shorter the sampling time is, the higher the recovery accuracy can be achieved.

7.2.3 Data Sampling Time. It is worth setting the time/delay/latency of the measurement for data sampling, and this can be analyzed on how fast air quality changes, and consequently, how soon each measurement expires. Since the changing trend of PM_{2.5} measurements is more volatile than PM₅, we depict the concentration values of the calibrated PM_{2.5} data collected per minute at the same location with our UbiAir sensor box on three different days from 4.30 pm to 6.30 pm, as shown in Figure 9 (c). In general, the changing trend of PM_{2.5} concentration is different. According to Figure 9 (c), the PM_{2.5} concentration is decreasing on Thursday, increasing on Saturday, and relatively stable on Sunday, along with the time evolving. In addition, the variation range of PM_{2.5} concentration is different over the three days as well, with the smallest range on Sunday (nearly $9 \mu\text{g}/\text{m}^3$), and the largest range on Thursday (nearly $27 \mu\text{g}/\text{m}^3$). Therefore, from our observation, it is difficult to predict the changing trend and changing rate of PM_{2.5} concentration, due to the fact that it is affected by many factors, such as weather, factory emissions, and vehicle exhaust emissions. Consequently, during UbiAir sampling period, we need to constantly monitor the changes of air pollutants at the same location, and the pollutant value collected by the UbiAir box sensor is uploaded to the back-end server for hardware calibration. Then, the new value will be compared with the previous value sampled at the same location and stored in the back-end database. Once the difference is too large to exceed the predefined threshold which could be customized during data sampling, then the previous measurement data expires and will be replaced by the new value. Furthermore, if there is no new pollutant value collected for a location since recording its previous data sampling over a comparably long time, the database needs to set all the previous measurement data over the grid map expired and re-generate red-envelopes for the new round of grid points to run data sampling again.

After ensuring the valid duration of data measurement, we would like to investigate how the system performance varies when the sample time changes. It is obvious that the sampling time (duration) to finish all the data collection tasks required by our BCS model for data construction changes with different number of workers involved during mobile crowdsensing in the same monitoring area. The higher the number of workers, the shorter the data sampling time. We first choose two groups of workers in our experiments, one group with 12 workers and the other group with 36 workers. Then we let these workers sample the air pollutant data at the same time of the day and in the same area; the area is divided into 32×32 grid cells, as described above. After that, 100 sampling positions are selected by the adaptive BCS method. The air pollutant data of the entire monitoring area are reconstructed with the calibrated measurements at the 100 sampling positions.

We compared the PM₅ data reconstruction errors of the two groups in Figure 9 (d); it shows the effects of different sampling times on the recovery accuracy. To evaluate the general performance, we use the sampling data collected over ten consecutive days, and these sampling data have been calibrated before running data reconstruction. We observe that the reconstruction error of the sampling data finished in a shorter time with averaging duration around 10 minutes (36 workers involved) is smaller than the data sampling over a longer period of time with averaging duration around 20 minutes (12 workers involved). In other words, the shorter the sampling time, the higher the recovery accuracy is. These results in Figure 9 (d) reflect the fact that the concentration of air pollutants (such as PM₅) in positions close by is usually correlated, and this correlation will become weaker as the duration of air pollutant data collection increases. That is, even if the distance between the sampling positions is not too far, as long as the sampling time takes too long, the correlation of the data

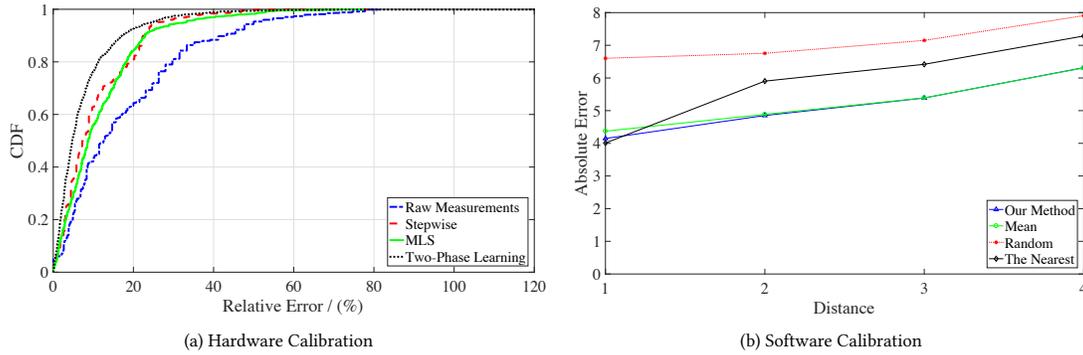


Fig. 10. The comparison of different calibration methods for PM5 estimation.

collected by the workers is not strong. This characteristic will cause interference to our inference model, making its data reconstruction error larger. Therefore, we need to reduce the time cost in the data sampling phase. The most effective way to reduce the sampling time is to increase the number of workers in the mobile crowdsensing. This further verifies the necessity of designing a feasible incentive scheme such as our red envelope strategies in data sampling process. The red envelope incentive we propose can reduce the sampling time to a certain extent, because its rewarding strategies can differentiate the information gain and the difficulty of data collection at sampling positions. Furthermore, our UbiAir can detect red envelope deployment positions inaccessible or geographically difficult to reach, and new red envelopes can be generated by the adaptive BCS at new locations where the workers are possible to collect the corresponding data and finish their data collection tasks in short time.

7.3 Effectiveness of Data Calibration

The low-cost sensors in the UbiAir box may have bias on hardware-dependent sampling precision during data measurement, so we have proposed the two-phase learning based hardware calibration method in Section 5.1. In addition, the Gaussian processing based software calibration is proposed to obtain the data on unreachable locations in Section 5.2. Both calibrations methods are used to provide reliable and accurate data measurements for sustainable air quality monitoring in our UbiAir system, and their performance in comparison with other methods are evaluated as follows.

7.3.1 Hardware Calibration. To evaluate the performance of our two-phase learning based hardware calibration, we compare it with two state-of-the-art methods, namely MLS and Stepwise [9] used in recent works for the calibration of air quality measurements as introduced in Section 5. A real-world PM5 data set collected by our UbiAir system from one sunny work day, one rainy work day, and one sunny weekend day respectively in the same monitoring area is used to present comprehensive conditions for the calibration evaluation. In our experiment, we use one high-precision sensor as the reference node to provide ground truth PM5 value and additional three environmental features (*i.e.*, CO₂, temperature, and humidity) to train the model. We also deploy three UbiAir boxes with low-cost sensors to collect raw PM5 measurements along the same trajectories as the high-precision sensor and run calibration after the measurements. Totally, our data set contains 4873 measurements from both the high-precision sensor and low-cost sensor, where 70% of the data are used for training and the rest data are used for testing.

Compared with the MLS and Stepwise methods that usually present poor performance for the nonlinear part, our two-phase learning based hardware calibration can not only fit the linear part well in the first phase, but also provide robust performance for the nonlinear part due to its adoption of the random forest in the second phase. Figure 10 (a) shows the comparison results in terms of the relative error of PM5 estimation. The raw measurements are depicted to present a baseline. We observe that our two-phase learning based calibration achieves the best performance, while MLS and Stepwise methods perform similarly. Specifically, according to the CDF (Cumulative Distribution Function), 76% of the calibrated measurements have relative error less than 10% using our two-phase learning method in UbiAir, while the percentages for MLS method is 56% and the percentages for Stepwise method is 63%. In addition, we calculate the root mean square error (RMSE) for these methods, and derive that the two-phase learning still yields the best result with a value of 7.906. Comparatively, the RMSE is 10.746 for MLS, 11.280 for Stepwise, and 16.537 for the raw measurement, respectively, which further confirms the superiority of our two-phase learning method on effective hardware calibration.

7.3.2 Software Calibration. The software calibration is proposed to approximate the measurement of air quality data at specific grid points where the sharing bikes have passed around with certain distance, based on their data collection at a series of sampling positions along the biking trajectories. To evaluate its performance, we choose a grid point with known air quality as the reference point, and then use four different trajectories passing around the grid point to estimate its air quality separately within the sampling circle, where nine sampling points on each trajectory are used to run the estimation once so that we have four estimation results for the reference point. The distance from the four trajectories to the reference point are different. That is, the smaller the ID of the trajectory, the closer the trajectory is to the reference point. We repeat this evaluation process to estimate the air quality for 342 grid points, and obtain the average performance as shown in Figure 10 (b). Our UbiAir estimation method using Gaussian processing of the nine sampling data from each trajectory is compared with another three methods, namely Mean (averaging the nine sampling data), Random (randomly selection from the nine sampling data), and The Nearest (sampling data closest to the reference point).

Figure 10 (b) presents the average performance of the absolute errors of PM5 estimations from the 342 reference point. As we can see from the results, when the distance from the reference point increases, the absolute error increases, indicating that it is not accurate to sample on a trajectory that is too far away such as the trajectory 4. That is why we set the sampling circle to exclude the trajectory that is distant from the emission source point, and only the sampling data from the trajectory inside the circle can be used to estimate the air quality. In addition, when the trajectory is close to the reference point such as the trajectory 1, the absolute error of the nearest method is the smallest as $4.0 \mu\text{g}/\text{m}^3$. This is because when the sampled position is close enough to the reference point, the value of this position will be almost equal to the reference value, and introducing the other 8 sampling data may cause estimation deviation from the reference value. For this case, the performance of our UbiAir is second only to the nearest method but still good. Furthermore, as the distance increases, UbiAir method starts to perform better than the nearest method. Though the average method also achieves good performance, when the trajectory is closer to the reference point, it is not as good as UbiAir. Overall, we conclude that our software calibration method in UbiAir is robust enough to various conditions and can present better performance than other methods.

7.4 Adaptive BCS based Red Envelope Deployment

In order to evaluate the efficiency of our proposed adaptive BCS algorithm on data recovery, we compare the adaptive BCS with other three methods on selecting sampling positions and deploying red envelopes, namely random deployment, open-loop entropy and open-loop mutual information. The random deployment method selects the data sampling positions randomly; this process is easy to understand. The open-loop entropy and open-loop mutual information methods are specifically introduced as follows.

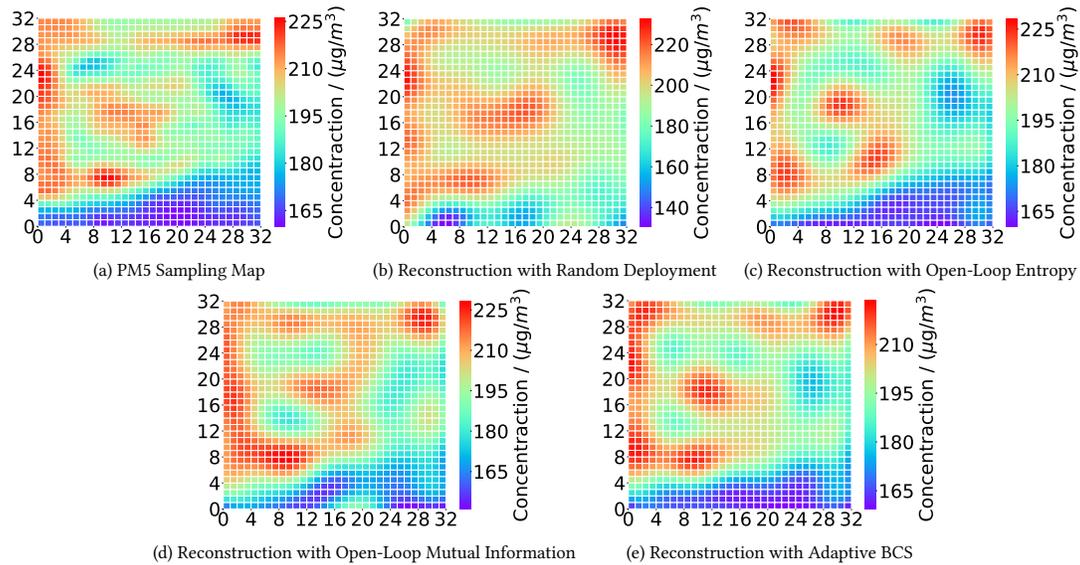


Fig. 11. The comparison of different deployments on data reconstruction map.

7.4.1 Open-Loop Entropy Approach. “Entropy” is a widely used measure of the uncertainty in the information theory and many other areas. In this open-loop entropy approach, the entropies of all possible red envelope positions are calculated and we deploy the next red envelope to the position with the highest entropy. The entropy in this approach is described in [17]. If we denote y as one of all possible red envelope positions and A as the set of the positions that have been sensed, as the distance between y and A increases so does the entropy value. Thus, far apart positions tend to give high entropies.

7.4.2 Open-Loop Mutual Information Approach. The open-loop entropy criterion tends to deploy the red envelope along the boundary of the monitored region, resulting in a sensor carried by the sharing bike on the boundary cannot detect the data out of the region and may waste sensed information. This problem was noticed and can be solved by that the mutual information (MI) criterion [30]. Based on this approach, we could maximize the mutual information criterion to estimate the optimal red envelope positions.

7.4.3 Evaluation Results. In our evaluations for these algorithms, the comparison experiments are performed with PM5 as the pollutant data over a grid map with 32×32 grid cells. The BCS inference model as explained in Section 6.1 is used in all the methods for data reconstruction, but their red envelope deployment strategies are different. Figure 11 (a) shows the distribution map of the sampling data collected at all the grid points. The performance on data reconstruction with the random red envelope deployment method, the open-loop entropy method, the open-loop mutual information method and the adaptive BCS method are presented in Figure 11 (b), (c), (d) and (e), respectively, for comparison. As can be seen from Figure 11, with the same number of selected measurements (100 calibrated data), the reconstructed data over the grid map with adaptive BCS approach is closer to the original sampling data collected from all the grid points than the other three deployment methods. The reconstruction error is 4.12% for the random red envelope deployment, 4.00% for the open-loop entropy, 3.16% for the open-loop mutual information, and 2.35% for the adaptive BCS, respectively. The performance of our approach is significantly lower than that of the other three approaches because the adaptive BCS enables us to

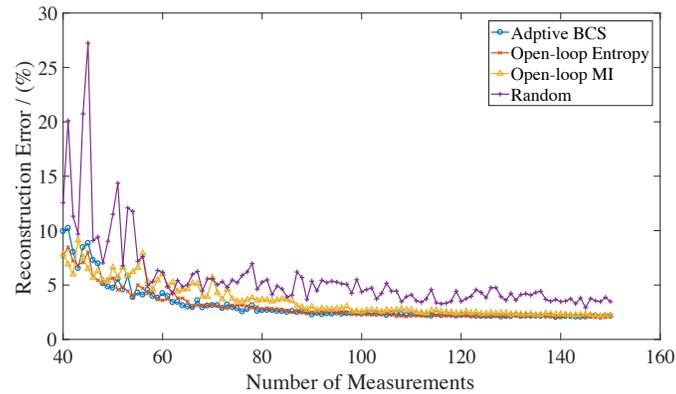


Fig. 12. The comparison of different deployments on data reconstruction error.

adaptively estimate the optimal next projection to be added into the measurement matrix, which makes it possible to sample data selectively at the most informative positions rather than others. Therefore, our evaluation verifies that the adaptive BCS algorithm can effectively improve the data recovery accuracy of air quality monitoring.

Furthermore, we studied the relationship between the number of model's input data and the reconstruction error, by comparing adaptive BCS with the other three methods. It can be clearly seen from the Figure 12 that no matter which red envelope deployment algorithm is used, the reconstruction error decreases as the number of the input data increases. But after the number of the input data reaches a certain level, the reconstruction error does not decrease significantly. In addition, the effects of different algorithms vary on reconstruction errors. Specifically, the curve of the random deployment algorithm is highly fluctuated along with the increased number of input data, because the input data selected by the algorithm each time is random (*i.e.*, the position of the red envelope deployment is random). The randomness makes the performance of BCS inference model unstable on red envelope deployment. Compared with the randomly deployment, the open-loop entropy and open-loop mutual information (MI) algorithms will calculate the information gain of the red envelope deployment position and carefully select the position that brings better benefits to the BCS inference model, so as to reduce the reconstruction error of the BCS inference model. The selection of the red envelope positions by the Open-loop MI or the Open-loop Entropy greatly improves the performance of the BCS inference model. Also, we can see that the Open-loop Entropy is better than Open-loop MI on reducing the reconstruction error, because the Open-MI tends to deploy red envelope along boundaries very well. This will result in very little data collected at the non-boundaries, affecting the performance of the BCS inference model. In addition, we can observe that the reconstruction error of our adaptive BCS converges to almost the same value as the reconstruction error of the Open-loop Entropy, so the adaptive BCS can also bring similar performance enhancements to the BCS inference model as the Open-loop Entropy. Meanwhile, the complexity of the adaptive BCS is lower than that of the Open-loop Entropy, therefore it tends to be the best algorithm for red envelope deployment among these methods.

8 USER FEEDBACKS AND FUTURE WORKS

We have held discussions with some of the test users of UbiAir system. Specifically, we first inquire what they learn from this work that they didn't know before. Their answers can be categorized as four aspects: 1) During the data collection process, they are aware that the ambient air quality might change frequently due to the spatial-temporal complexity, which was not well-known to some people used to trust the publicly-available

macro-scale and coarse-grained air quality data. So they agree with the importance of micro-scale and fine-grained air quality monitoring, and would love to see applications such as our UbiAir. 2) Most of our test users didn't believe the low-cost sensors can accurately estimate the concentration of different air pollutants, but are finally convinced by their experience with the UbiAir box that fine-grained air quality monitoring can be achieved through the coupling effects between sensor calibration and data reconstruction. 3) The implementation and deployment of UbiAir system has inspired our users that the sharing economy and the deployment of IoT can be naturally coexisted to support many sustainable applications, due to their common characteristics, e.g. low-cost, large-scale and replaceable. 4) Surprisingly, many users tell us that since the red envelope incentive of our UbiAir application encourages them to share and compare the rewards with each other, their awareness of environmental protection has been built or strengthened during this process as well. Therefore, they are happy to see interesting and practical incentive schemes with positive impacts to keep human in the loop.

In addition, we further discuss with these users about what we can do to further improve the performance and user experience of our UbiAir system. The feedbacks are collected and summarized as follows. We also give our solutions as future works.

- In practical operation of data collection, the generated sampling positions with red envelopes might be located in areas that cannot be reached or easily accessed by sharing bikes (e.g. indoor area, caves and tunnels, lakes and rivers). To guarantee the overall performance of the UbiAir system, we could design more advanced estimation method based on the proposed software calibration [43]. Considering current software calibration is inherently an offline operation run on the UbiAir server side and need the client side to geotag the data with geographical information such as GPS. For the cases without positioning signals, we could extend the estimation to be an online operation with self-calibration based on the real-time sampling data along the biking trajectory [19]. In addition, other shared IoT platforms could be deployed and used to collect data for those areas. For example, we could deploy a small number of public sensors fixed on these areas as an open testbed, customize the UbiAir box on existing shared electric scooters to get more convenient access, and use flying drones attached with sensors for near-surface monitoring.
- The data collected by our UbiAir box are transmitted to mobile phone and cached during the biking trip. After the trip finished, the UbiAir mobile application can upload the cached data to the server. During this process, the battery power of mobile phone is consumed during the biking trip. Also, some users are reluctant to spend their data plan to upload data [42]. To solve these issues, we could compensate users by crediting their usage of mobile phone based on the mileage of their biking trip participating in the data collection. The users also could choose to upload the cached data to the server only when WiFi is detected and connected nearby [41]. In addition, in order to obtain timely air pollutant data for fine-grained monitoring, UbiAir users could be encouraged by rewarding if they use data plan to immediately upload collected data through cellular traffic.
- User privacy is a critical problem that we need to pay attention to during the usage of UbiAir system [38]. Some users may not want us to get their private information, such as biking trip details and identity information. To solve these issues, we could use the blockchain technology to block sensitive information on user side and only keep the air pollutant data information [33]. For example, the users could use their own key pairs to upload data without their identity information, and smart contract could be designed to ensure the user's privacy will not be leaked during rewarding process. In addition, the collected data could also be encrypted and traded over blockchain platform as a sustainable business model to run UbiAir, where the inherently distributed features of blockchain could be used to differentiate the sensitive levels of various data and separate the storage and transaction of these data.

9 CONCLUSIONS

UbiAir has been designed and implemented as a mobile crowdsensing system for fine-grained air quality monitoring. The system can greatly reduce the number of sensors and resource consumption by utilizing the mobility of sharing bikes attached with customized IoT sensing devices. It only needs to sample a small number of air pollutant data by cyclists as the crowd workers, and the air pollutant data of the entire monitoring area can be recovered with high spatial resolution. UbiAir exploits grid-based region partitioning method, a red envelope incentive method, and a data calibration method to improve the recovery accuracy. Furthermore, an adaptive BCS model is proposed to select grid positions for red envelope deployment during data sampling so that the final reconstruction error of air quality data can be greatly reduced. Therefore, UbiAir is a light-weight, low-cost, accurate and scalable system, presenting promising performance and usability on combining mobile crowdsensing with sharing economy for fine-grained air quality monitoring.

REFERENCES

- [1] Davide Aguiari, Giovanni Delnevo, Lorenzo Monti, Vittorio Ghini, Silvia Mirri, Paola Salomoni, Giovanni Pau, Marcus Im, Rita Tse, Mongkol Ekpanyapong, and Roberto Battistini. 2018. Canarin II: Designing a smart e-bike eco-system. In *IEEE CCNC*. 1–6.
- [2] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. 2010. Model-based compressive sensing. *IEEE Transactions on Information Theory* 56, 4 (2010), 1982–2001.
- [3] Emmanuel J. Candès, Nathaniel Braun, and Michael B. Wakin. 2007. Sparse Signal and Image Recovery from Compressive Samples. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Washington, DC, USA, April 12-16, 2007*. 976–979.
- [4] Yun Cheng, Xiucheng Li, Zhijun Li, Shouxu Jiang, Yilong Li, Ji Jia, and Xiaofan Jiang. 2014. AirCloud: a cloud-based air-quality monitoring system for everyone. In *ACM Conference on Embedded Network Sensor Systems (SenSys), Memphis, Tennessee, USA, November 3-6, 2014*. 251–265.
- [5] Srinivas Devarakonda, Parveen Sevusu, Hongzhang Liu, Ruilin Liu, Liviu Iftode, and Badri Nath. 2013. Real-time air quality monitoring through mobile sensing in metropolitan areas. In *ACM KDD Workshop on Urban Computing (UrbComp)*. ACM, 15:1–15:8.
- [6] Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. 2009. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks* 6, 1 (2009), 6:1–6:39.
- [7] Bart Elen, Jan Peters, Martine Van Poppel, Nico Bleux, Jan Theunis, Matteo Reggente, and Arnout Standaert. 2013. The Aeroflex: A Bicycle for Mobile Air Quality Measurements. *Sensors* 13, 1 (2013), 221–240.
- [8] Biyi Fang, Qiumin Xu, Taiwoo Park, and Mi Zhang. 2016. AirSense: an intelligent home-based sensing system for indoor air quality analytics. In *ACM UbiComp*. ACM, 109–119.
- [9] Xinwei Fang and Iain Bate. 2017. Using Multi-parameters for Calibration of Low-cost Sensors in Urban Environment. In *International Conference on Embedded Wireless Systems and Networks (EWSN)*. 1–11.
- [10] Jon Froehlich, Joachim Neumann, and Nuria Oliver. 2009. Sensing and Predicting the Pulse of the City through Shared Bicycling. In *IJCAI*. 1420–1426.
- [11] Yi Gao, Wei Dong, Kai Guo, Xue Liu, Yuan Chen, Xiaojin Liu, Jiajun Bu, and Chun Chen. 2016. Mosaic: A low-cost mobile sensing system for urban air quality monitoring.. In *IEEE INFOCOM*. 1–9.
- [12] Radhika Garg and Christopher Moreno. 2019. Understanding Motivators, Constraints, and Practices of Sharing Internet of Things. *ACM IMWUT (UbiComp)* 3, 2 (2019), 44:1–44:21.
- [13] Bin Guo, Yi Ouyang, Cheng Zhang, Jiafan Zhang, Zhiwen Yu, Di Wu, and Yu Wang. 2017. CrowdStory: Fine-Grained Event Storyline Generation by Fusion of Multi-Modal Crowdsourced Data. *ACM IMWUT (UbiComp)* 1, 3 (2017), 55:1–55:19.
- [14] Tencent Inc. 2020. WeChat. <https://www.wechat.com/>.
- [15] Shihao Ji, Ya Xue, Lawrence Carin, et al. 2008. Bayesian compressive sensing. *IEEE Transactions on Signal Processing* 56, 6 (2008), 2346–2356.
- [16] Keunseo Kim, Hengameh Zabihi, Heeyoung Kim, and Uichin Lee. 2017. TrailSense: A Crowdsensing System for Detecting Risky Mountain Trail Segments with Walking Pattern Analysis. *ACM IMWUT (UbiComp)* 1(3) (2017), 65:1–65:31.
- [17] Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9, Feb (2008), 235–284.
- [18] Yuxiang Lin, Wei Dong, and Yuan Chen. 2018. Calibrating Low-Cost Sensors by a Two-Phase Learning Approach for Urban Air Quality Measurement. *ACM IMWUT (UbiComp)* 2, 1 (2018), 18:1–18:18.
- [19] Kaikai Liu, Di Wu, and Xiaolin Li. 2016. Enhancing Smartphone Indoor Localization via Opportunistic Sensing. In *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9.

- [20] Liang Liu, Wu Liu, Yu Zheng, Huadong Ma, and Cheng Zhang. 2018. Third-Eye: A Mobilephone-Enabled Crowdsensing System for Air Quality Monitoring. *ACM IMWUT (UbiComp)* 2, 1 (2018), 20:1–20:26.
- [21] Xiaofeng Liu, Chaosheng Xiang, Bin Li, and Aimin Jiang. 2015. Collaborative Bicycle Sensing for Air Pollution on Roadway. In *IEEE UIC-ATC-ScalCom*. 316–319.
- [22] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang. 2016. TaskMe: multi-task allocation in mobile crowd sensing. In *ACM UbiComp*. 403–414.
- [23] Beijing Mobike Technology Co. Ltd. 2020. Mobike. <https://www.mobike.com/>.
- [24] Balz Maag, Zimu Zhou, and Lothar Thiele. 2018. W-Air: Enabling Personal Air Pollution Monitoring on Wearables. *ACM IMWUT (UbiComp)* 2, 1 (2018), 24:1–24:25.
- [25] Randall V Martin. 2008. Satellite remote sensing of surface air quality. *Atmospheric Environment* 42, 34 (2008), 7823–7843.
- [26] Jimmy Moore, Pascal Goffin, Miriah Meyer, Philip Lundrigan, Neal Patwari, Katherine Sward, and Jason Wiese. 2018. Managing In-home Environments through Sensing, Annotating, and Visualizing Air Quality Data. *ACM IMWUT (UbiComp)* 2, 3 (2018), 128:1–128:28.
- [27] World Health Organization. 2018. Ambient (outdoor) air quality and health. [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health).
- [28] David Posada and Thomas 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.
- [29] Blaine A. Price, Avelie Stuart, Gul Calikli, Ciaran McCormick, Vikram Mehta, Luke Hutton, Arosha K. Bandara, Mark Levine, and Bashar Nuseibeh. 2017. Logging you, Logging me: A Replicable Study of Privacy and Sharing Behaviour in Groups of Visual Lifeloggers. *ACM IMWUT (UbiComp)* 1, 2 (2017), 22:1–22:18.
- [30] Naren Ramakrishnan, Chris Bailey-Kellogg, Satish Tadepalli, and Varun N Pandey. 2005. Gaussian processes for active data mining of spatial aggregates. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 427–438.
- [31] Eike Schneiders and Mikael B. Skov. 2019. CyclAir: A Bike Mounted Prototype for Real-Time Visualization of CO2 Levels While Cycling. In *IFIP INTERACT*. 678–687.
- [32] Fengrui Shi, Zhijin Qin, Di Wu, and Julie A. McCann. 2018. Effective Truth Discovery and Fair Reward Distribution for Mobile Crowdsensing. *Pervasive and Mobile Computing* 51 (2018), 88–103.
- [33] Fengrui Shi, Zhijin Qin, Di Wu, and Julie A. McCann. 2018. MPCSToken: Smart Contract Enabled Fault-Tolerant Incentivisation for Mobile P2P Crowd Services. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*. 961–971.
- [34] Fengrui Shi, Di Wu, Dmitri I. Arkhipov, Qiang Liu, Amelia C. Regan, and Julie A. McCann. 2019. ParkCrowd: Reliable Crowdsensing for Aggregation and Dissemination of Parking Space Information. *IEEE Transactions on Intelligent Transportation Systems* 20(11) (2019), 4032–4044.
- [35] Rundong Tian, Christine Dierk, Christopher Myers, and Eric Paulos. 2016. MyPart: Personal, Portable, Accurate, Airborne Particle Counting. In *ACM CHI*. ACM, 1338–1348.
- [36] Michael E Tipping. 2001. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research* 1, Jun (2001), 211–244.
- [37] Michael E. Tipping and Anita C. Faul. 2003. Fast Marginal Likelihood Maximisation for Sparse Bayesian Models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003*.
- [38] Zhen Tu, Runtong Li, Yong Li, Gang Wang, Di Wu, Pan Hui, Li Su, and Depeng Jin. 2018. Your Apps Give You Away: Distinguishing Mobile Users by Their App Usage Fingerprints. *ACM IMWUT (UbiComp)* 2, 3 (2018), 138:1–138:23.
- [39] Shuai Wang, Tian He, Desheng Zhang, Yuanchao Shu, Yunhuai Liu, Yu Gu, Cong Liu, Haengju Lee, and Sang Hyuk Son. 2018. BRAVO: Improving the Rebalancing Operation in Bike Sharing with Rebalancing Range Prediction. *ACM IMWUT (UbiComp)* 2, 1 (2018), 44:1–44:22.
- [40] Chao Wu, Di Wu, Shulin Yan, and Yike Guo. 2013. Sensor Deployment in Bayesian Compressive Sensing Based Environmental Monitoring. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 37–51.
- [41] Di Wu, Dmitri I. Arkhipov, Thomas Przepiorka, Yong Li, and Qiang Liu. 2017. From Intermittent to Ubiquitous: Enhancing Mobile Access to Online Social Networks with Opportunistic Optimization. *ACM IMWUT (UbiComp)* 1, 3 (2017), 114:1–114:32.
- [42] Di Wu, Dmitri I. Arkhipov, Thomas Przepiorka, Qiang Liu, Julie A. McCann, and Amelia C. Regan. 2017. DeepOpp: Context-aware Mobile Access to Social Media Content on Underground Metro Systems. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*. 1219–1229.
- [43] Yuan Zhang, Lichun Bao, Max Welling, Shih-Hsien Yang, and Di Wu. 2010. Localization Algorithms for Wireless Sensor Retrieval. *Comput. J.* 53, 10 (2010), 1594–1605.
- [44] Yu Zheng, Furu Liu, and Hsun-Ping Hsieh. 2013. U-Air: When urban air quality inference meets big data. In *ACM KDD*. ACM, 1436–1444.
- [45] Tongqing Zhou, Zhiping Cai, Bin Xiao, Leye Wang, Ming Xu, and Yueyue Chen. 2018. Location Privacy-Preserving Data Recovery for Mobile Crowdsensing. *ACM IMWUT (UbiComp)* 2, 3 (2018), 151:1–151:23.