

CurrentSense: A novel approach for fault and drift detection in environmental IoT sensors

Sumukh Marathe, Akshay Nambi, Nishant Shrivastava, Manohar Swaminathan, Ronak Sutaria*
Microsoft Research India, *Respirer Living Sciences Pvt. Ltd

ABSTRACT

Sensor data quality plays a fundamental role in increasing the adoption of IoT devices for environmental data collection. Due to the nature of the deployment, i.e., in-the-wild and in harsh environments, coupled with limitations of low-cost components, sensors are prone to failures. A significant fraction of faults result from drift and catastrophic faults in sensors' sensing components leading to serious data inaccuracies. However, it is challenging to detect faults by analyzing just the sensor data as a faulty sensor data can mimic non-faulty data and an anomalous sensor reading need not represent a faulty data. Existing data-centric approaches rely on additional contextual information or sensor redundancy to detect such faults. This paper presents a systematic approach to detect faults and drifts, by devising a novel sensor fingerprint called CurrentSense. CurrentSense captures the electrical characteristics of the hardware components in a sensor, with working, drifted, and faulty sensors having distinct fingerprints. This fingerprint is used to determine the sensors' health, and compensate for drift or diagnose catastrophic faults without any contextual information. The CurrentSense approach is non-intrusive, and can be applied to a wide variety of environmental sensors. We show the working of the proposed approach with the help of air pollution sensors. We perform an extensive evaluation in both controlled setup and real-world deployments with 51 sensors across multiple cities for 8 months period. Our approach outperforms existing anomaly detectors and can detect and isolate faults with an F_1 score of 98% and compensate for sensor drift errors by 86%.

1 INTRODUCTION

The proliferation of the Internet of Things (IoT) devices has led to the deployment of billions of sensors in various domains to sense and monitor the environment [24]. Applications of IoT in environmental monitoring range from air pollution monitoring, water quality monitoring, extreme weather sensing, endangered species protection, to commercial farming and many more. Since these applications rely on the fidelity of the sensed data for making decisions, it is *fundamental to determine the quality of the sensor data*, by detecting whether the sensor is working, faulty, or drifted.

A vast majority of IoT devices deployed today utilize compact digital sensors owing to their reliability and ease of installation. While these digital sensors have paved the way for large-scale IoT deployments, it is quite challenging to determine the accuracy, and the fidelity of the data, just by analyzing the digital data stream. Especially, given the nature of the IoT deployments in environmental monitoring, i.e., in-the-wild and in harsh conditions, IoT sensors are prone to failures [31]. A majority of these faults are due to drift and catastrophic faults in sensors' sensing components [34]. Typically, when a sensor fails, either due to malfunctioning of a few components (leading to catastrophic faults) or wear and tear (leading to drift), they do not just stop sending data, but continue

to transmit faulty or dirty data [16, 18]. Given the uncontrolled environment in which these sensors are deployed, it is impossible to assess the accuracy of the data without additional contextual information or sensor redundancy [31].

In this paper, we present a novel approach called CurrentSense to detect catastrophic faults and drift in digital sensors, going beyond traditional data-centric approaches. CurrentSense is based on the following simple idea: Every electrical or electro-mechanical sensor draws current from the IoT device for its operation. By sampling the current drawn by the sensor we can derive a unique electrical characteristic fingerprint that differs between working, faulty, and drifted sensors. Our key observation based on theoretical and extensive experimental evidence is that when any sensor component accumulates damage that causes a fault or drift, the damage also changes other physical properties of the sensor, which affects its current consumption. Thus, by monitoring the current consumption of the sensor without any additional details of the sensor, we can now accurately derive the status of the sensor.

Current signature analysis has played a key role in fault diagnosis over the past decade mostly in household appliances (such as HVAC) or industry equipment's (such as induction motors) [22]. The focus has been limited towards detecting abnormalities in power consumption patterns of an appliance. This limitation is due to the dynamic change in consumption patterns either due to the complexity of the appliance being monitored (e.g., multiple states in a washing machine cycle) or usage variations due to the presence of users (turning on/off appliances at any point). Our work is inspired and build upon the recent works in energy monitoring area [25, 33, 35]. Our key novelty is the application of current monitoring for accurate fault detection and isolation in low-cost environmental IoT sensors. Further, CurrentSense approach can accurately detect and quantify drifts in these sensors caused due to aging or wear and tear of components. This paper presents a holistic view of current signature analysis to systematically detect and isolate faults/drifts in large-scale environmental IoT deployments.

CurrentSense fingerprint has the following key characteristics: (i) *Distinct* for a working, drifted, and faulty sensor; (ii) *Quantifies the amount of drift*; (iii) *Independent of the measured phenomena*; (iv) *Non-intrusive with no or minimal hardware modification* and can be automatically measured using simple lightweight software APIs. In a typical operation, a baseline fingerprint is first determined before deployment to characterize a working sensor. The fingerprint is then measured periodically in a device deployed in the field and compared against the baseline. We have devised an edge algorithm that runs locally on the IoT device and evaluates the fingerprints to determine if the sensor is working, faulty or drifted.

The CurrentSense approach can be applied to a range of digital environmental sensors to monitor sensor health and improve data reliability (see Section 7). However, to effectively showcase the working of CurrentSense approach we use Air pollution $PM_{2.5}$

sensors in the remainder of the paper. The choice of $PM_{2.5}$ sensors is informed by the severity of the sensor reliability problem that exists today in air pollution monitoring [32] and the impact CurrentSense can have to advance the sensor data quality leading to dependable policy decisions. Air pollution is a major concern worldwide, with an estimated 7 million deaths every year [36] and $PM_{2.5}$ is a major factor contributing to this mortality [38]. Furthermore, air pollution is known to be a complex phenomenon with spatio-temporal variations requiring fine-grained, hyper-local monitoring to identify potential sources and curb pollution levels. With advances in sensing technologies, recent efforts have employed low-cost sensors for fine-grained $PM_{2.5}$ monitoring at scale [19, 21]. These sensors are compact, portable, and typically cost between \$30 to \$100 [9, 10]. There are numerous ongoing efforts that deploy 100s of IoT devices, which use these low-cost $PM_{2.5}$ sensors at scale, from academic projects [2, 8], smart city initiatives [13], to commercial projects by companies such as AirCasting [1] and Clarity [4]. However, recent works report several challenges in sensor reliability with prevalent sensor faults and drifts in these deployments (see Section 2.2), leading to significant data inaccuracies, causing a butterfly effect from polluting the data lake to making bad policy decisions [21, 27].

The CurrentSense approach is validated with a popular $PM_{2.5}$ sensor from Plantower [10]. We have implemented the proposed techniques in a real-world deployment of 51 IoT devices, each equipped with a low-cost $PM_{2.5}$ sensor, deployed across multiple cities for over 8 months. To show the efficacy of CurrentSense, we performed extensive evaluation in both controlled setup and real-world deployments. Further, we show that CurrentSense approach can be easily applied to other $PM_{2.5}$ sensors from different manufacturers, *namely*, Sensirion [9], and Honeywell [5]. Finally, in Section 7 we also show how CurrentSense can be used to detect faults in other important environmental sensors such as CO_2 , multi-gas and temperature sensors. To the best of our knowledge, this is the first work that addresses sensor faults and drift in low-cost environmental sensors at scale¹ without using contextual information and sensor redundancy. We also note that in combination with data-centric approaches (see Section 5.5) this can be an even more powerful fault diagnosis system. The key contributions are:

- We present a novel CurrentSense fingerprint, to systematically detect and isolate faults. Further, we show that the proposed approach can be applied to a range of environmental sensors.
- CurrentSense fingerprint can be used to detect if the sensor is drifted and also, quantify the amount of drift. This can then be used to either automatically compensate for the drift or intelligently trigger a calibration procedure.
- We demonstrate the efficacy of CurrentSense to detect and localize faults with a deployment of 51 IoT devices equipped with $PM_{2.5}$ sensors across multiple cities for over 8 months period.

2 BACKGROUND AND $PM_{2.5}$ SENSOR FAULTS

We now describe the working of a low-cost $PM_{2.5}$ sensor and then present numerous data faults observed in $PM_{2.5}$ sensors along with limitations and challenges in the state-of-the-art approaches.

¹We plan to open source the CurrentSense library to the community towards fault and drift detection in a wide range of sensors including $PM_{2.5}$ sensors.

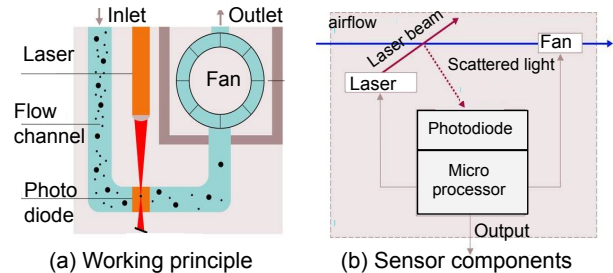


Figure 1: Working principle of $PM_{2.5}$ sensor and its components [9].

2.1 Low-cost $PM_{2.5}$ sensor and its working

A typical low-cost $PM_{2.5}$ sensor is based on the optical light scattering principle. Figure 1 shows the working principle and the block diagram of $PM_{2.5}$ sensor, which has four main components, namely, FAN, LED, photodiode, and an internal microcontroller. First, a controlled airflow is created inside the sensor by means of a small DC FAN. The particles travel from inlet to outlet via a chamber with the help of the airflow and pass through a light source, usually a laser beam from the LED, causing light scattering. The scattered light is then detected by the photodiode and converted to a mass concentration output [5, 9, 10]. The process of estimating the number of particles present in each size, i.e., PM 1, PM 2.5, PM 4, and PM 10 is called binning. The estimated number of particles is converted to mass concentration in $\mu g/m^3$. Since this conversion is based on some pre-defined average particle density and airflow values, *the low-cost sensors can only provide a good estimation of PM concentration rather than accurate readings as obtained in traditional expensive sensors* [29]. Majority of the air pollution monitors deployed at scale today use one of the following three popular sensors, *viz.*, Plantower [10], Sensirion [9], and Honeywell [5]. Thus, we use the above three sensors in our experimentation.

2.2 Understanding $PM_{2.5}$ Sensor data faults

Sensed data that is inconsistent with the measured phenomenon’s true value is referred to as data fault [18, 31].

Catastrophic faults: While $PM_{2.5}$ sensor manufacturers claim 8-10 years of operating lifetime [5, 9, 10], numerous works have reported that these low-cost sensors can breakdown within 2-3 months of operation [20, 27]. In fact, in our own deployment we observed sensor failures under just *five weeks* of deployment.

Sensor faults due to failure in microcontroller or battery can be easily identified as the sensor stops responding to commands or stops sending data. However, faults and drift in $PM_{2.5}$ sensors’ internal components, i.e., FAN and LED, are the most common causes of sensor faults resulting in generating garbage or dirty data, which are hard to detect as we describe next.

Case-1: Mimicking data- Figure 2(a) represents a scenario where a faulty sensor data mimics a working sensor data. Both sensors were deployed in different outdoor locations and it can be seen that the data from both the sensors are in similar range. However, upon manual inspection, Sensor 11 had a FAN fault, which resulted in non-uniform airflow leading to inconsistent PM reading, which is hard to distinguish without sensor redundancy.

Case-2: Anomalous data- Figure 2(b) shows $PM_{2.5}$ data from sensors 17 and 15 with few regular spikes. Sensor 17 was deployed close to the road and the spikes in the data correspond to vehicular traffic.

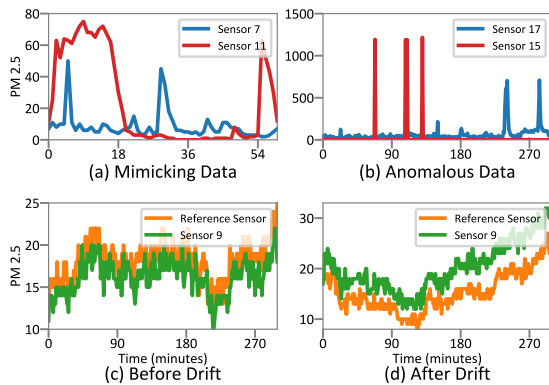


Figure 2: Sensor data faults in $PM_{2.5}$ sensor.

However, Sensor 15 had a FAN fault, which resulted in non-uniform airflow and also a temporary accumulation of particles leading to inconsistent and occasional peaks. Given the hyper-local variations in air pollution, the spikes in the data could either represent valid data or faulty data and it is non-trivial to resolve.

Sensor drift: Low-cost $PM_{2.5}$ sensors are *estimation-based sensors* and hence *sensor calibration* is necessitated where a low-cost sensor measurement is transformed in a way that closely matches the measurement from a high-end reference sensor, usually done by co-locating [20, 32]. However, upon deployment, this calibration may not remain valid, mainly due to wear and tear, i.e., change in FAN’s speed or LED’s intensity, leading to sensor drift and gradual change in PM data, which can go unnoticed for long periods.

Figure 2(c) shows the PM data for Sensor 9 and a reference sensor (a new low-cost sensor) both placed at the same location (co-located). Data from Sensor 9 is in agreement with the reference sensor, indicating no drift. Sensor 9 was then deployed in the field for 2 months and then we co-located it again with the reference sensor. Figure 2(d) shows the offset in PM data between Sensor 9 and reference sensor, due to sensor drift. Upon manual inspection, we found that the LED’s light intensity in Sensor 9 was changed (due to wear and tear) as compared to when it was first deployed, leading to incorrect $PM_{2.5}$ estimation. In all the above scenarios, we identified sensor faults by manual inspection and as discussed it is impossible to detect such faults by just analyzing sensor data.

2.3 Related work

State of the art techniques that aim to address the above challenges can be broadly classified into two: data-centric and system-centric.

Data-centric efforts. Prevalent research efforts have largely focused on data-centric approaches (rule-based or anomaly detection), where historical data (ranging from days to years) of the sensor is analyzed and a fault is identified if the data is out of bounds of the expected behavior [31]. However, such an approach has inherent limitations [34] namely, (i) a faulty sensor data can mimic non-faulty data and (ii) an anomalous sensor reading need not represent a faulty data, especially in the context of air pollution with hyper-local variations.

(i) *Fault detection in air pollution sensors:* Several long-term deployments have shown that data quality is a major concern and advocate developing sophisticated techniques for sensor fault detection [32, 34]. Chen *et al.* [19] employ statistical techniques with

additional contextual data to detect anomalies, i.e., temporal anomaly or spatial or combination, based on data collected from near-by devices and its own historical data. However, temporal and spatial dependency lead to high false positives and negatives, due to local variations in $PM_{2.5}$ data [23]. A sensor redundancy approach is presented in [17], which includes two $PM_{2.5}$ sensors in the device, and spikes or abnormalities in the data are validated by checking the presence across both the sensors. This approach is not feasible due to the cost and battery requirements, especially in deployments with 100s-1000s of sensors. Recent works [26, 37] model $PM_{2.5}$ data to estimate or predict sensor readings using spatio-temporal correlations with adjacent sensors. Any deviation between the sensed and predicated data is categorized as fault. These approaches can detect sudden changes in the data, but give inaccurate predictions in case of gradual, seasonal, and other environmental changes.

(ii) *Sensor drift detection:* Many researchers report that $PM_{2.5}$ sensors start exhibiting drift within a few months of deployment due to wear and tear, aging, and semiconductor impurity effects [20, 32]. Periodic calibration of the sensor is a common technique employed to eliminate drift, where a deployed sensor is co-located with a high-end sensor periodically. However, this process is cumbersome, expensive, and requires the sensor to be brought back for co-location [20]. Recent works [30, 37] present a blind calibration technique using learning algorithms to calibrate sensors based on the assumption that data from nearby sensors should be highly correlated. However, such techniques perform poorly when applied to $PM_{2.5}$ as nearby sensors can have differing levels of pollution due to various emission sources leading to hyper-local variations.

These sensor data-driven approaches rely on significant historical sensor data (ranging from few hours, days to years) to model the sensed data behavior. In contrast, CurrentSense *relies on just a few hardware signatures that can be collected within a short period of time (few minutes) to learn the fault conditions.* Further, the data-driven approaches rely on additional contextual information such as data from near-by sensors, deployment location (roof-tops/on-road), and weather information to determine faults/anomalies. The CurrentSense *fingerprints are agnostic to the environment in which the sensors are deployed and hence do not require any contextual information to detect faults* as described in Section 3.1.

System-centric efforts. Sensor faults can occur due to various reasons, from improper installation to environmental factors such as dust, to sensing component failures [31, 34]. Recent work on sensor fault detection [18] has shown that, fall-curve – sensor’s voltage response when the power is turned off, can be used to characterize sensor faults in analog sensors. However, this approach has several limitations, namely, (i) it works only for analog sensors where a sensor’s output voltage can be measured directly and hence cannot be applied to compact digital sensors, (ii) fall-curve is designed to only detect faults, and cannot be used to detect and measure sensor drift, and (iii) fall-curve requires the sensor to be powered down to determine its status, which may not be feasible in safety-critical applications. This paper focuses on *developing a novel approach towards detecting faults and drifts in digital environmental sensors.* CurrentSense relies on the fact that any fault/drift in sensor components induces change in the current drawn by the sensor, which can be easily discerned. We show CurrentSense’s efficacy through extensive controlled experiments and real-world deployment.

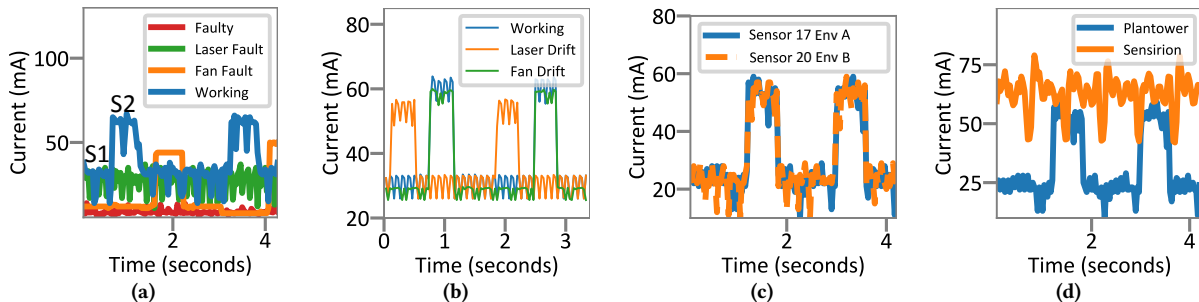


Figure 3: CurrentSense characteristics: (a) Characteristic-1 (working vs. faulty), (b) Characteristic-2 (quantify drift), (c) Characteristic-3 (independent of measured phenomena), and (d) Characteristic-4 (different sensor type).

Current signature analysis for fault diagnosis. Current signature analysis is a popular technique used for fault diagnosis in various systems. For example, motor current signature analysis (MCSA) is a well-known approach to detect faults in rotor-bar, bearings, etc., of a three-phase induction motor drive [22]. The approach essentially relies on high resolution sampling of supply current to the motor. HVACs is another equipment where current signature has played a key role in fault diagnosis. Fine-grained measured to detect abnormalities or faults in HVAC components [35]. Recently, several intrusive [25] and non-intrusive [33] load monitoring approaches have explored anomaly/fault detection in electrical appliances. SocketWatch [25] is a plug attached to an appliance such as refrigerator, washing machine and TV, in a household to learn the behavioral model of an appliance, by measuring its active and reactive power consumption patterns. It detects appliance malfunctions by observing any deviations from these patterns. RIMOR [33] is a non-intrusive approach that predicts the energy consumption of a household using historical energy data and contextual information, and flags an anomaly when the actual consumption deviates significantly from the predicted consumption.

Majority of the above research efforts focus on just detecting anomalies by measuring significant deviation in current consumption patterns of household or industrial appliances. To the best of our knowledge, this is the first work which applies current monitoring for accurate fault detection and isolation in low-cost environmental sensors for root-cause analysis. Unlike household appliances, whose usage depends on the occupancy of users, leading to dynamic change in current profiles, low-cost sensors have a fixed schedule of operation resulting in almost static energy consumption profile. Further, we go a step beyond fault detection, where CurrentSense is also used for detecting and compensating for the drift observed in such sensors due to natural aging of components.

3 CURRENTSENSE AND ITS WORKING

CurrentSense approach samples the current drawn by the sensor to derive a unique electrical characteristic fingerprint. The intuition here is that when a sensor goes faulty its hardware characteristics vary leading to change in its current consumption. CurrentSense needs to monitor only the overall aggregated consumption of the sensor to determine its status. Hence it is non-intrusive, does not require opening the sensor to monitor the signals, and requires no hardware modification. *The current consumption of the sensor is typically dictated by the electronics and the program schedule running in the microcontroller of the sensor.* These programs are pre-loaded by

the manufacturers, which include a fixed set of actions that always remain the same. Figure 3(a) shows the CurrentSense fingerprint sampled at 30Hz along with the different states of a working Plantower $PM_{2.5}$ sensor (shown in blue) [10]. Upon powering up the sensor, the FAN runs continuously at a pre-defined speed to maintain a constant airflow (state S1 with 35mA consumption). This is followed by a periodic trigger of LED every 2.5 seconds (state S2 with 55mA consumption) to estimate $PM_{2.5}$. This current consumption profile acts as the “fingerprint” for a working sensor.

3.1 CurrentSense characteristics

We now describe the key characteristics of CurrentSense.

Characteristic-1: Distinguishable between working and faulty sensor. Faults/drift in the sensor components results from physical degradation of the sensor, which also change its current consumption. Figure 3(a) shows the distinct current consumption profiles for a working, FAN fault, LED fault, and completely faulty $PM_{2.5}$ sensor. We can clearly see the distinguishable current signatures between different faults. Further, CurrentSense can also localize the faults to a FAN or an LED, due to their unique current characteristics.

Characteristic-2: Detect and quantify drift. As sensor components experience drift, its current consumption also varies. For instance, when an LED drifts it reduces its radiant intensity, and when FAN drifts it reduces its RPM, both resulting in decreased current consumption. Figure 3(b) shows the current consumption of a working, FAN drifted, and LED drifted sensors. While the overall fingerprint looks similar for all the three sensors, the current consumption of drifted sensors deviates as compared to a working one, which can be used to detect and quantify the drift.

Characteristic-3: Independent of measured phenomena. Since $PM_{2.5}$ sensors are based on optical light scattering principles, CurrentSense fingerprint is independent of the measured phenomena and the environment. Hence, the fingerprint remains the same despite changes in pollution levels, temperature and humidity, unless there is a fault/degradation in sensor components. Figure 3(c) shows identical CurrentSense fingerprints for Sensor 17 and 20 deployed in two different environments. Sensor 17, Env A was deployed in a dusty and high temperature environment (PM levels around 200, temperature = 55°C) and Sensor 20, Env B was deployed in a relatively low pollution and low temperature environment (PM levels around 10, temperature = 15°C).

Characteristic-4: Distinct for each manufacturer. CurrentSense fingerprints from the same sensor manufacturer have very little variations even across different batches. Thus, fingerprint once collected for a manufacturer should be sufficient to distinguish

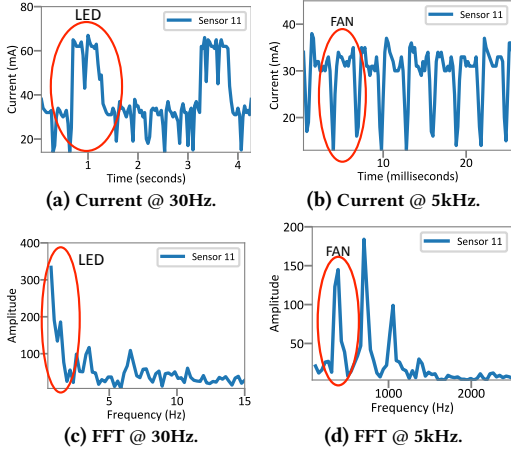


Figure 4: CurrentSense and FFT features at 30Hz and 5kHz.

between a working and faulty sensor. However, *fingerprint can vary from one manufacturer to another due to the internal program schedule on the controller and sensor components*. For instance, in a Sensirion sensor, both the LED and FAN are switched “on” at a different frequency as compared to Plantower as shown in Figure 3(d). Thus one has to collect the fingerprint per manufacturer and can re-use it for different sensor instances from the same manufacturer.

3.2 Extracting features from CurrentSense

To reliably use CurrentSense, we transform the time domain current data to frequency domain by extracting Fast Fourier Transform (FFT) features. A $PM_{2.5}$ sensor includes a set of electro-mechanical components each operating at different frequencies. For instance, in a Plantower sensor, the LED switching on/off frequency is 0.4Hz (i.e., every 2.5s) and the FAN typically rotates at a speed of 200 Hz (12000 RPM) (derived from the datasheet [10]). Thus, CurrentSense is sampled at two rates, viz., 30Hz and 5kHz, to capture all the variations and monitor both low-frequency (LED) and high-frequency components (FAN) of the sensor. We sample current at two frequencies since we are constrained by the window length of the current sample. While the exact sampling rates need to be fine-tuned based on the time domain and subsequent frequency domain resolution required, the general intuition for frequency selection is that we want to monitor the signals present at the operating frequency along with higher order harmonic frequencies of the component. Thus, at each of the sampling rates, we collect 128 data points, where the number of data points collected is optimised so as to include a minimum number of cycles of the periodic signals observed and also consume as less memory as possible.

To derive features from CurrentSense fingerprint, we extract 64 point FFT features from both the 30Hz and 5kHz current signals. We selected 64 point FFT features mainly due to, (i) memory limitation on a resource-constrained device and hence lower length is preferable, and (ii) length lower than 64 has overlapping frequency components, thus limiting the usefulness of the FFT features. Thus a CurrentSense fingerprint is a tuple of length 128:

$$\langle \text{fft}_1, \dots, \text{fft}_{64}, \text{fft}_1^1, \dots, \text{fft}_{64}^1 \rangle, \quad (1)$$

where, $\text{fft}_1, \dots, \text{fft}_{64}$ and $\text{fft}_1^1, \dots, \text{fft}_{64}^1$ represents FFT features from 30Hz and 5kHz CurrentSense signal, respectively.

Figure 4(a) and (b) show the current consumption of a working $PM_{2.5}$ sensor sampled at 30Hz and 5kHz. The highlighted region in

Figure 4(a) and (b) clearly indicates the change in current consumption due to the LED and FAN ON, respectively. Figure 4(c) and (d) shows the corresponding FFT features extracted from the 30Hz and 5kHz current signal with the highlighted regions showing the dominant frequencies when FAN and LED is switched ON.

3.3 Edge Algorithms for Fault Detection

We present two edge algorithms that is trained on features derived from CurrentSense fingerprints to classify the sensor status.

3.3.1 Binary classification algorithm. We perform a binary classification to detect if the sensor is working or faulty by *collecting training data only from working (non-faulty) sensors*. The key idea is that if the measured fingerprint diverges from the training data collected from a set of working sensors, then there is a fault.

Model training: We collected 1000 samples of CurrentSense fingerprints from 5 working sensors and extracted feature vectors as described in Equation 1. We then calculate the mean and standard deviation for each of the 128 FFT features to derive a feature dictionary. This is then loaded onto the device for classification. We note that the process of collecting the fingerprints is carried out in a lab environment and *takes just few minutes for each sensor*.

Model testing and accuracy: For a test fingerprint, we first compute the z-score or the standardized score for each raw FFT feature [12]. We then compute the Euclidean distance between the feature vector comprising of calculated z-scores and zero vector. If this distance is within a certain threshold (3 standard deviation (STD)) we classify the sensor as working, otherwise faulty.

We used 400 samples from 2 working and 2 faulty sensors (with even distribution) and if the distance between test features and the dictionary is greater than 3 STD, we classify it as faulty. The overall F_1 score for the binary classification was 97% with high precision and recall for both classes. This is mainly due to the distinct features present in the working CurrentSense fingerprints.

Discussion: The key benefit of this approach is that the user has to just collect a few samples of CurrentSense from only working sensors. *Thus reducing the time to build the model and also making the approach simple and easy to deploy at scale*. However, such an approach can only detect if there is a fault or not, but cannot localize or isolate the fault to individual components.

3.3.2 Multi-class classification algorithm. This approach isolates the fault and classifies the sensor status to either, working, FAN fault, LED fault, or completely faulty. To this end, we collect fingerprints for each of the above four classes during training.

Model training: We collected 1000 CurrentSense fingerprints across 5 sensors, namely, 2 working sensors and 1 sensor each with FAN, LED, and completely faulty. Similar to the binary classification method, the entire process of collecting data is carried out in a lab environment and takes just few minutes for each sensor. We then build a classifier model using the above training data. We use – Bonsai [28], a state-of-the-art classification model that is optimized to run on resource-constrained devices such as Arduino. Bonsai is a novel tree based algorithm, which maintains high prediction accuracy while minimizing model size. The training requires the following input parameters, viz., number of features (128), number of classes (4), number of train samples, projection dimension (28), and depth of the Bonsai tree (3). The trained model is then loaded onto the device.

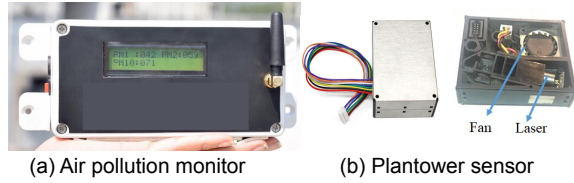


Figure 5: Air pollution devices and $PM_{2.5}$ sensors used in our deployment.

Model testing and accuracy: The test fingerprint collected is evaluated with the trained model locally to derive the sensor status. To test the accuracy of the trained model, we used around 700 CurrentSense samples from 7 sensors (100 samples from each sensor), which includes 2 working, 2 FAN fault, 2 LED fault and 1 completely faulty sensor. The overall test accuracy was 99% with high precision and high recall across all four classes. Thus showing the efficacy of the model to accurately detect and localize faults.

Discussion: This approach *can detect and localize faults (to either FAN or LED fault)* in $PM_{2.5}$ sensors. However, it requires collecting training data for each class of faults *a priori*. In Section 5 we present the efficacy of the above models to accurately detect faults.

4 EXPERIMENTAL SETUP

4.1 Deployment details

We are working with a company (anonymized for double blind), who have deployed over 200+ pollution devices across 20 cities over the past two years. In this paper, for evaluating CurrentSense approach, we have selected and re-deployed 51 devices from the 200+ devices for over 8 months across multiple cities. These 51 devices use a Plantower sensor and constitute our real-world deployment.

Figure 5 shows the compact air pollution device used in our deployment, which includes a $PM_{2.5}$ sensor (Plantower 7003 [10]), STM32 microcontroller, 4000 mAh rechargeable battery, a GPS and 4G modules for location information and data transmission. The 51 devices are deployed in various static locations such as building roof-tops, beside the road, and at road intersections. The device measures PM values continuously and sends out an average PM value every one minute to a centralized server through 4G. We manually inspected the sensors and ensured all the sensors are working by co-locating them in one location before the deployment.

4.2 Ground truth and deployment stats

To collect ground truth information on the status of the sensor, we manually inspected all the sensors every week (the only way to collect reliable ground truth due to hyper-local variations in PM data). Specifically, we went to the deployed location with a reference sensor (new low-cost sensor, verified regularly against high-end sensor) and compared the PM data between the reference and deployed sensor. If the PM data is in agreement, then the deployed sensor is labeled “working”. If the PM data variation is $>5\%$ of the reference sensor, then the sensor is labeled as “drifted” and if the variation in PM data is significantly high we classify them as “faulty”. In the case of faulty, we further inspect the FAN and LED components to isolate the faulty component. Furthermore, if we detect a faulty/drifted sensor, we *do not* replace them, thus allowing us to test the efficacy of the proposed CurrentSense approach.

Deployment stats. At the end of 8 months, among 51 sensors, 33 sensors were working, 9 were drifted, 4 has FAN fault, 3 has LED

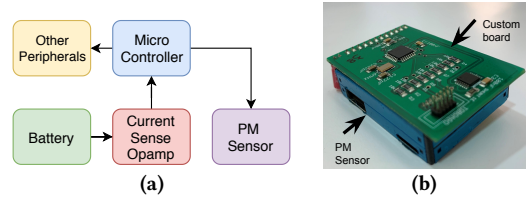


Figure 6: CurrentSense measurement: (a) Block diagram - scenario 1, (b) Plantower $PM_{2.5}$ sensor retrofitted with custom board - scenario 2.

fault and 2 are completely faulty. Thus around 35% (18/51) of the sensors experienced faults/drift in our deployment.

4.3 How to measure CurrentSense?

We now describe in detail how to measure CurrentSense fingerprints. A typical IoT device generally has a current monitoring circuit (i.e., current sense amplifiers), to monitor the current consumption of the battery-powered low-cost devices. Current sense amplifiers (or current shunt monitors) are differential amplifiers that monitor the current flow by measuring the voltage drop across a sense element, typically a shunt resistor [11]. These are low-cost ($< \$1$), precise (error $< 1\%$), ultra-low power consumption ($260\mu A$) and can support currents from 10s of μA to 10s of A. We now describe two scenarios to extract CurrentSense fingerprint:

Scenario-1: Device already has a current sense amplifier.

Our pollution monitoring device shown in Figure 5(a) already includes a Texas Instruments INA4180 current sense amplifier [11]. Hence, in this case, we have a lightweight software code (API) running on the microcontroller to sample the current signature. Figure 6(a) shows the block diagram of the electrical system with the current sense amplifier. Since the amplifier is present before the microcontroller, it measures the current consumption of the entire device, which includes a compact $PM_{2.5}$ sensor along with other peripherals such as GPS and 4G modules. Since we monitor the status of the sensor only periodically (i.e., every 1 minute or 1 hour or 1 day), we need to sample CurrentSense only for a short duration. Thus, when we extract the fingerprint using our APIs, we ensure to programmatically turn off the high-power consuming peripherals such as GPS and 4G, thus restricting the measurement to only the current consumption of the $PM_{2.5}$ sensor. All the experimental results in this paper are derived using this method. The lightweight API to extract CurrentSense can run on any microcontroller.

Scenario-2: Device has no current sense amplifier. In cases where the pollution device does not include a current amplifier, we have designed a low-cost compact board (costing $< \$5$) that can be easily retrofitted to the existing $PM_{2.5}$ sensors. Figure 6(b) shows the retrofitted compact board with $PM_{2.5}$ sensor, which includes a current sense amplifier and some minimal electronics. This board now enables extraction of CurrentSense and can work with any $PM_{2.5}$ sensor. Note that, there is no modification required except the sensor connector for its usage with other manufacturers. Since the amplifier is now directly connected to the $PM_{2.5}$ sensor, the device measures the current consumption of the sensor directly irrespective of the peripherals present in the air pollution device.

5 FAULT DETECTION AND ISOLATION

We now present results from controlled experiments and real-world deployment to showcase the ability of CurrentSense to detect and isolate faults in digital $PM_{2.5}$ sensors.

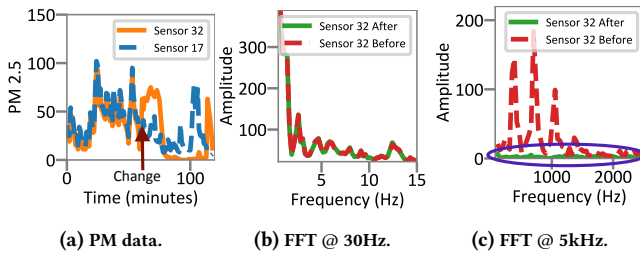


Figure 7: Fault detection and isolation - Fan (controlled).

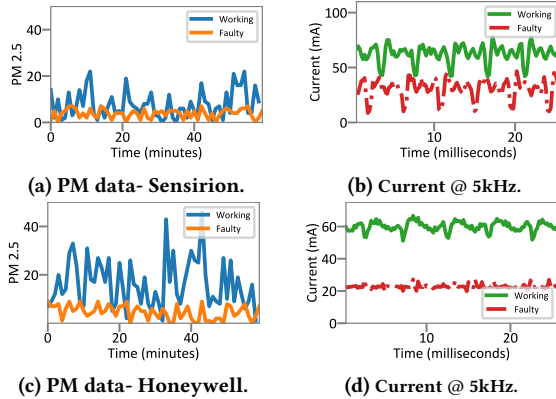


Figure 8: Fault detection- (a), (b) LED fault - Sensirion and (c), (d) Completely faulty - Honeywell.

5.1 Controlled experiments

Figure 7(a) shows PM data of two $PM_{2.5}$ sensors at the same location. We see that the PM data of both the sensors, match accurately till time = 50 minutes. At time = 50 minutes, we manually injected a fault, specifically, we replaced the “working” FAN of Sensor 32 with a “faulty” FAN. At this point, we can see that the PM data of Sensor 32 diverges significantly compared to Sensor 17, but is still in the similar PM data range. *Without contextual information or redundant sensors, it is challenging to detect such faults.* We now show that by comparing CurrentSense fingerprint of Sensor 32 before and after fault injection, we can easily detect and isolate the faulty component. Figure 7(b) and (c) shows the FFT features extracted from 30Hz and 5kHz current signal. We can see that the FFT features at 30Hz before and after the change are identical, indicating no LED fault. However, the FFT features at 5kHz corresponding to the FAN are significantly different before and after the change, indicating a FAN fault. *Thus, we can accurately detect and isolate faults by analyzing CurrentSense fingerprints.* Similar experiments were also conducted to validate LED and complete sensor fault.

5.2 Applicability of CurrentSense on different $PM_{2.5}$ sensor manufacturers

We now show the usage of CurrentSense on other popular $PM_{2.5}$ sensors such as Sensirion SPS30 [9] and Honeywell HPMA115S0 [5]. Note that, the entire fault detection approach remains the same as earlier. The only difference is that the CurrentSense fingerprint and its FFT features for a working and faulty sensor vary for each manufacturer, thus necessitating training a new model per sensor manufacturer. Figure 8(a), (c) shows the PM data from two Sensirion and two Honeywell sensors, respectively. While the data from both the sensors are in the same range, one of them is faulty. Figure 8(b),

Table 1: Fault classification using supervised algorithm. Classes: Working (W), FAN fault (FF), LED fault (LF), Completely faulty (CF).

(a) F_1 score					(b) Confusion matrix					
Metrics	Working				Laser fault	Completely faulty	W	FF	LF	CF
	Working	Fan fault	Laser fault	Completely faulty						
Precision (%)	100	96	95	97			1.00	0.00	0.00	0.00
Recall (%)	97	100	100	100			0.04	0.96	0.00	0.00
F_1 score (%)	99	98	97	98			0.05	0.00	0.95	0.00
							0.03	0.00	0.00	0.97

(d) shows the CurrentSense fingerprint at 5kHz sampling rate. It can be seen that the current consumption of the faulty sensor (LED fault in Sensirion and complete fault in Honeywell) is quite distinctive compared to a working sensor, indicating a fault. *Thus, CurrentSense can be used seamlessly to detect and isolate faults across $PM_{2.5}$ sensor manufacturers.*

5.3 Real-world deployment results

Fault evaluation. In Section 3.3.2 we showed that the multi-class classification has 99% accuracy in detecting the sensor status using the training data from controlled experiments. Before deployment, we first load this trained model for fault detection across all 51 IoT devices. Since the models are optimized to run on resource-constrained devices the evaluation is performed locally. The deployed device periodically measures the CurrentSense fingerprint (every 1 minute) and evaluates it with the trained model to derive sensor status. The classification result (sensor status) along with the fingerprint was sent to the centralized server for analysis.

Since ground truth was collected once every week, we verified the classification results from all the 51 sensors once every week. Specifically, we sub-sampled 10 CurrentSense fingerprints once every week for 8 month period across all 51 sensors, resulting in 17340 CurrentSense fingerprints (10 samples x 34 weeks x 51 sensors). Table 1(a) shows the precision, recall and F_1 score obtained using the trained supervised model on all 17K fingerprints. The model has high scores for both precision and recall showing that the classifier is returning accurate results with very low false positives and false negatives across all classes as also shown in the confusion matrix in Table 1(b). For instance, in the case of working sensors, the model accurately predicted a working sensor as a working with a precision of 100% with no false positives or negatives. Similarly, a FAN fault predicted as FAN fault has a precision of 96% with 4% false positives and no false negatives. *Thus, a model trained with data collected in the lab can still accurately detect and isolate faults in real-world with an overall F_1 score of 98% across all classes, owing to the robustness of CurrentSense fingerprints derived for each fault.*

Binary classification model accuracy. We also evaluated our binary classification model (Section 3.3.1), which classifies the sensor status to either working or faulty. For the 17340 fingerprints collected in real-world, the binary classification model has an F_1 score of 96.5%, with working class having precision and recall of 99% and 96% and faulty class having precision and recall of 94% and 98%. *Thus, even a simple binary classification model can accurately classify if the sensor is working or faulty with our fingerprints.*

Isolation and recovery from sensor faults. Even though $PM_{2.5}$ sensors are compact, their internal components are all modular (see Figure 5(b)). The individual components such as FAN and LED can be replaced easily, upon accurately detecting the faulty component. CurrentSense’s ability to isolate faulty components thus assists in identifying the component that needs to be replaced as compared

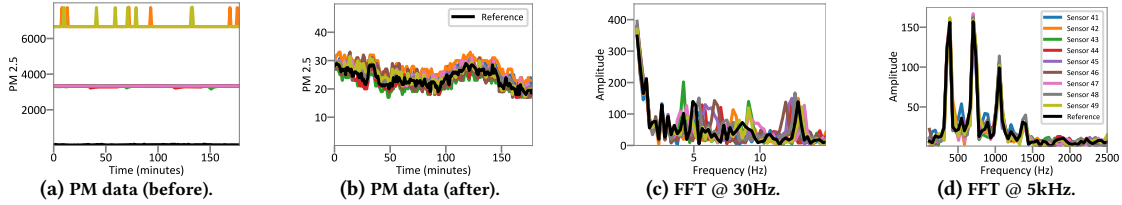


Figure 9: Stuck-at fault detection using data-centric and CurrentSense.

to discarding the entire sensor. This approach is more economical than replacing the sensor in large deployments, especially given that these can break within a few weeks of deployment [20].

5.4 Comparison with data-centric algorithms

We selected a state of the art anomaly detector for $PM_{2.5}$ sensors, ADF [19] as our baseline. ADF divides the continuous sensor data into a time slice and identifies three types of anomalies:

(i) **Spatial Anomaly** is detected if measurement data is far greater (or lower) than the mean of its neighboring high-end reference sensor in the same time slice. We selected one month of data from 8 devices in our deployment, which were within 1km radius of an expensive high-end reference sensor. We divide the time-series data into fixed 30 minute intervals and calculate the mean and standard deviation in each time slice. Any deviation greater than 2 standard deviations from the reference sensor was detected as anomaly. Such an approach has average F_1 score of 77.8% with 69.6% Precision and 98.9% Recall. The poor results are mainly due to hyper-local variations in the pollution levels, even for sensors within 1km radius of reference.

(ii) **Temporal Anomaly** is detected when there is any significant deviation in sensor expected behavior as compared to its historical data. We analyzed 1 month of historical sensor data for each of the 8 devices to derive mean and standard deviation of the data. We then evaluated it over a 1 month period, where any deviation from the expected behavior (2 STD) was reported as an anomaly. The average F_1 score is 67.2% with precision of 50.7% and recall of 99.3%. Distribution of particle matters is generally non-stationary due to numerous emission sources and environmental factors, resulting in unpredictable temporal variability.

(iii) **Spatio-temporal Anomaly** is detected if the measurement data is identified as both spatial and temporal anomaly in the same time slice. For the one month evaluation data, the average F_1 score is 33.0% with 60.2% Precision and 22.7% Recall. Note that, in all the above cases we manually verified the sensor status everyday for ground truth along with reference sensor.

To summarize, data-centric approaches relies on historical data (e.g., 1 month prior sensor data) or dense deployment for comparison with reference sensor and yet perform poorly (high false positives and negatives) due to hyper-local, spatio-temporal variations. In contrast, the CurrentSense approach does not rely on historical sensor data (just few fingerprints collected in a lab setting within few minutes) and uses novel fingerprints to detect faults. Further, the fingerprints are independent of the environment and measured phenomena. In the above evaluation, CurrentSense was able to detect faults with an F_1 score of 98.5% with 97.4% precision and 99.8% recall.

5.5 CurrentSense with data-centric algorithms

Since $PM_{2.5}$ sensors are deployed in outdoors and exposed to a high concentration of particulate matter, it is quite common for the

sensor to go bad giving either continuous very high or very low PM data (“Stuck at” fault [31]). Traditional data-centric approaches can detect these faults easily by analyzing the sensor data, however, it is not clear what the problem is? The company deploying 200+ sensors across numerous cities encountered such faults at least once every 3 weeks and due to the ambiguity in the cause of failure, they just discard and replace the sensor with a new sensor. At the end of our deployment, upon manual inspection, 18% of the sensors (i.e., 9/51) were identified with *Stuck at fault* (PM data with continuous high or low data). Figure 9(a) shows very high/low PM data measured by these 9 sensors along with a working sensor at the same location. We validated this with a temporal anomaly detector [19] described previously. While the anomaly detector identified all the 9 sensors to be faulty, the root-cause for the fault is unclear. In such cases, CurrentSense can be used to deduce the reason for the fault.

Figure 9(c) and (d) show FFT features of CurrentSense at 30Hz and 5kHz for the working sensor (in black) and the 9 dusty sensors. It can be seen that all the 9 dusty sensors have similar FFT features compared to a working sensor. Further, our Bonsai model when evaluated against these features identified all the sensors as working. Hence by deduction, we narrow the fault to environmental factors, specifically, dust accumulation inside the air chamber due to prolonged exposure to particle matters. To verify this, we performed a cleaning procedure on all the 9 sensors by manually blowing air into the sensor to remove any dust particles. Figure 9(b) shows the data after cleaning for all the 9 sensors along with the working sensor. We can clearly see that across all the sensors the PM data now closely matches the working sensor, thus recovering from the dust fault. Thus, to address faults due to environmental factors we employ a two-step process, where a data-centric algorithm first detects an anomaly, and then CurrentSense is used to eliminate false positives and localize the fault accurately.

5.6 Performance profiling

We now present the time taken to extract FFT features and inference time for binary and multi-class classification models. Our pollution device uses a resource-constrained microcontroller, i.e, STM32, on which it takes just 38ms to extract 128-FFT features from the current signal. Further, the latency and memory required for binary model is 5ms and 8KB, and for multi-class it is 13.7ms and 14KB, respectively. Further, CurrentSense approach increases the energy consumption by just 0.012%, when the sensor status is derived every 1 minute, this includes CurrentSense measurement, feature extraction, and classification. Thus, CurrentSense approach can run on any resource-constrained microcontrollers in real-time with very low memory and power footprint requirement.

6 DETECTING AND MEASURING DRIFT

Sensor drift occurs when one or more of the sensor components change its properties, due to wear and tear or degradation. In $PM_{2.5}$

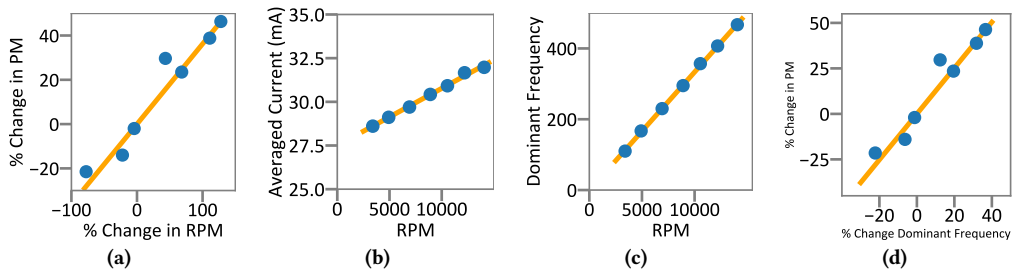


Figure 10: FAN drift results, (a): % change in RPM vs. % change in PM, (b): RPM vs. avg. current, (c): RPM vs. dominant frequency, (d): % change in dominant frequency vs. % change in PM.

sensors, it could be a change in FAN’s RPM and/or LED’s intensity, both leading to drift in PM data. Since sensor drift is gradual over time, it can go *unnoticed for long periods* and is quite challenging to detect without co-location with reference sensor. Typically, a sensor is identified as drifted when the PM data difference between the deployed and the reference sensor is $>5\%$ [32]. Through extensive experiments, we show how CurrentSense can be used to detect and quantify sensor drift without any sensor redundancy.

6.1 FAN Drift

We now present results to validate the following key observation, i.e., *Change in FAN’s RPM leads to change in both, PM data measured and CurrentSense fingerprint*. Thus, by monitoring the current drawn we could detect drift in the FAN’s operation.

Theory of operation. The FAN in $PM_{2.5}$ sensors is a brushless DC motor (BLDC) driven by direct current and Hall effect sensor. Speed of a BLDC motor is directly proportional to the voltage applied to the rotor [3]. Hall effect sensors act as “brushes” in a BLDC motor and are used to control the motor polarity. Motor polarity switches twice in one fan rotation in case of a unipolar BLDC motor. *This switching of polarity is reflected in the current consumption, thus giving a current signal with frequency twice that of the FAN speed.* Thus, the FAN speed is directly correlated with the frequency of the current signal, and any drift in speed is also reflected in the current drawn. Note that, since we do not have access to the internals of the compact $PM_{2.5}$ digital sensor, it is not feasible to obtain the FAN speed directly and hence we use the current signal to determine the FAN speed.

Setup. To study the effect of FAN drift, we manually varied the voltage applied to the FAN, which changes the FAN’s RPM.

1: Change in RPM \implies Change in airflow rate and PM data.

The FAN component is used to ensure there is a constant airflow for particles to flow through. Since the sensor assumes constant airflow to estimate PM data, any change in FAN’s RPM will also change airflow rate/particles entering the chamber, leading to a change in PM data. To verify this, we co-located a reference sensor (a new low-cost sensor) along with the FAN speed controlled sensor. If there is no drift in both the sensors, the difference in RPM between a reference sensor and a controlled sensor (% change in RPM) should be 0 and so should be the PM difference reported by both the sensors. Figure 10(a) shows that as the % change in RPM increases between the sensors, the % change in PM data also increases. The zero-intercept line shows that there is a strong correlation between change in RPM to change in PM data.

2: Change in RPM \implies Change in current consumption.

Since the FAN in $PM_{2.5}$ sensor is a BLDC motor, the speed of a

BLDC motor is directly proportional to the voltage applied [3]. Thus, as FAN’s RPM increases, the current drawn by the FAN also increases as shown in Figure 10(b).

3: Change in RPM \implies Change in dominant frequency.

Figure 10(c) shows the correlation between the RPM and dominant frequency from the 5kHz FFT signal. We can clearly see as RPM increases, the dominant frequency component of the 5kHz signal also increases linearly, with the best fit line having an r^2 of 0.99. Thus by monitoring the change in dominant frequency, we can detect a change in FAN’s speed, which represents drift in the FAN.

4: Change in dominant frequency \implies Change in PM.

Figure 10(d) shows the % change in PM data between a reference and a FAN speed controlled sensor for varying % change in dominant frequency. We can see that as % change in dominant frequency increases (indicating change in FAN’s RPM), the % change in PM data also increases with a strong correlation.

Thus, by measuring the % change in dominant frequency we can quantify the change induced in PM values, which can then be used to compensate for the drift. For plantower sensors from our experiments we see that 20% change in dominant frequency induces 25% change in PM values (as seen in Figure 10(d)).

6.2 LED Drift

Similarly, we now validate that, *Change in LED’s radiant intensity leads to change in both, PM data and CurrentSense fingerprint*.

Theory of operation. Light scattering sensors typically require a narrow and focused light beam, and hence high power LEDs are often used. Over time, LED’s accumulate damage, primarily through hot carrier injection, which reduces their intensity. For instance, an LED’s intensity can reduce by 15 - 25% after just 1500 hours of operation [6]. Further, LED’s intensity is directly proportional to the current drawn and hence any fault/drift in an LED will lead to change in its intensity and the current drawn.

Setup. To study LED drift, we manually varied the voltage applied to the LED, which changes the emitted radiant intensity [6].

1: Change in LED intensity \implies Change in PM data.

As LED’s emitted light intensity decreases, the light scattered due to particles also decreases inside the chamber, thus resulting in lower PM data being measured. To verify this, we co-located a reference sensor along with the LED intensity controlled sensor. Figure 11(a) shows that as the % change in LED intensity (compared with a reference sensor) increases, the % change in PM values between the reference and controlled sensor also increases. The best fit line shows the strong correlation between LED intensity and PM values.

2: Change in LED intensity \implies Change in current consumption.

LED’s intensity is directly correlated with the current consumption.

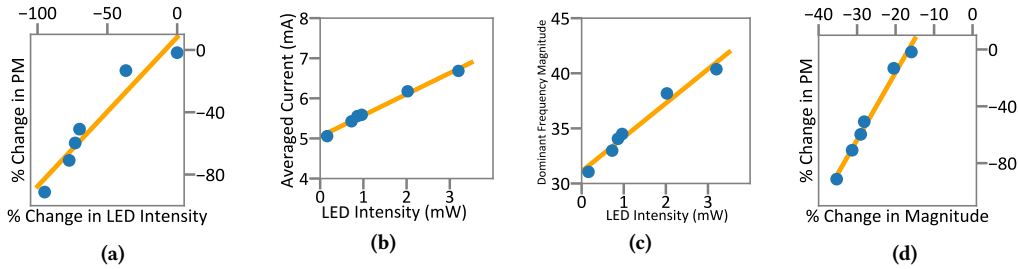


Figure 11: LED drift results, (a): % change in LED intensity vs. % change in PM, (b): LED intensity vs. avg current, (c): LED intensity vs. dominant frequency magnitude, (d) % change in dominant frequency magnitude vs. % change in PM.

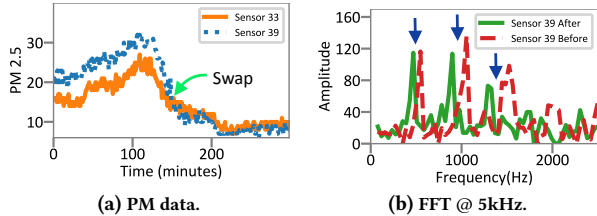


Figure 12: Sensor drift detection - Fan (controlled)

Figure 11(b) shows that as the LED intensity increases the average current consumption of the LED also increases, with the best fit line having an r^2 of 0.99.

3: *Change in LED intensity* \implies *change in FFT dominant frequency's magnitude*.

The LED in $PM_{2.5}$ sensor triggers every 2.5 seconds and hence the operating frequency is around 0.4Hz. Thus, the magnitude of the dominant frequency in 0.4Hz bin represents the radiant intensity emitted by the LED. Figure 11(c) shows change in LED's light intensity also changes the magnitude of the dominant frequency from 30Hz signal. Thus, change in radiant intensity can be detected by measuring the change in magnitude of the dominant frequency.

4: *Change in magnitude of dominant frequency* \implies *Change in PM*. Figure 11(d) shows the % change in PM data between a reference sensor and a LED intensity controlled sensor for varying % change in magnitudes of the dominant frequency. As the % change in magnitude increases (i.e. higher change in light intensity) the % change in PM also increases with a strong correlation indicated by the best fit line with r^2 of 0.98.

Thus, by measuring the % change in magnitude of the dominant frequency, we can detect and measure the change in LED's intensity and PM values. Specifically, a 10% change in dominant frequency's magnitude induces a 35% change in PM data (as seen in Figure 11(d)).

6.3 Controlled experiments

We now show FAN drift detection from our controlled experiments. We measured the FAN RPM (as described in section 6.1) for two sensors in our deployment. Sensor 39 with 5300 RPM has higher RPM (23% increase) than Sensor 33 with 4300 RPM.

Figure 12(a) shows the PM data measured by both the sensors at the same location. It can be seen that till time = 149 minutes the data from both the sensors had a similar trend but with a fixed offset (Sensor 39 reading higher values due to increased FAN speed). At time = 150 minutes, we interchanged the fan component from Sensor 33 to 39. At this point, we can clearly see the sudden change in PM data for both the sensors. The change in PM data is mainly due to the change in FAN's speed resulting in change in air flow rate. Note that, during this experiment, there was no change in any

other sensor components or in the environment. Thus showing the impact of FAN drift on PM data. Figure 12(b) shows the FFT features at 5kHz current signal before and after FAN change for Sensor 39. The highlighted region shows the change in the dominant frequency before and after the change. Thus, by monitoring the change in dominant frequency we can detect FAN drift. Similar behavior was also observed from LED drift experiments.

6.4 Real-world deployment results

Each IoT device measures the CurrentSense fingerprint during deployment and evaluates it with the trained model to derive sensor status. If the sensor status is *faulty*, the system detects the faulty component and stops the evaluation. However, if the status of the sensor is *working*, an additional check is performed to detect if the sensor has *drifted* or *not*. This is because the drifted sensors still have a similar fingerprint as compared to that of a working sensor.

Drift evaluation. Before deployment, the CurrentSense approach automatically records and store the fingerprint along with the dominant frequency components at 30Hz and 5kHz in the EEPROM, indicating a working, non-drifted signature. Upon classification of the sensor as *working* based on the trained model, the device then compares the % change in dominant frequency between the stored fingerprint and measured fingerprint to detect *sensor drift*. Further, the % change in dominant frequency is used to estimate the appropriate compensation for the PM data as described in Section 6.5. In our deployment we found 9 sensors to be drifted (18% of the sensors) and upon manual inspection of components, we found all to be due to FAN drift.

Figure 13(a) shows the % change in dominant frequency at 5kHz signal (as compared to the per sensor fingerprint recorded before the deployment) against % change in PM (as compared with the reference sensor) for all sensors at the end of 8 months. After eliminating the 9 faulty sensors, we can see three clusters in the figure, the cluster to the left corner represents the 33 working sensors with very low change in dominant frequency and PM error, indicating *no drift*. The cluster in the middle shows 6 sensors that are *moderately drifted* with change in both dominant frequency and PM. Finally, the cluster in the right shows 3 sensors that are *significantly drifted*, i.e., around 40% change in dominant frequency and around 30% change in PM values. We can see that there are no outliers in the plot, i.e., when there is a change in dominant frequency there is always a corresponding change in the PM data, indicating that the CurrentSense approach accurately detected all the drifted sensors.

Figure 13(b) shows PM data of Sensor 4 in agreement as compared with the reference sensor on Day 1, with very low PM RMSE (error = 0.1). At the end of the deployment (after 8 months), CurrentSense

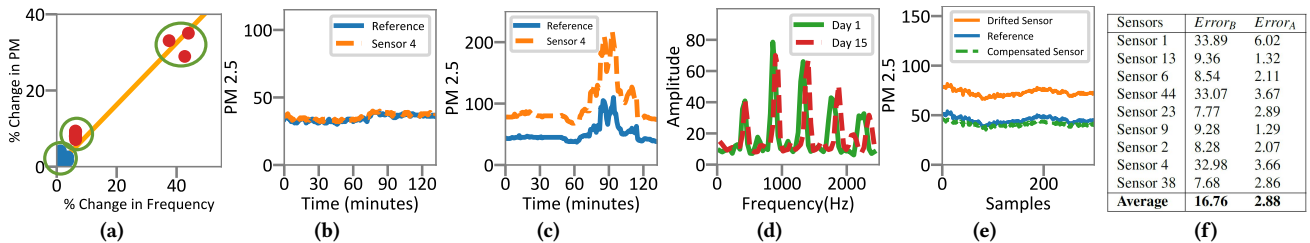


Figure 13: Sensor drift results from our deployment. (a): % change in dominant frequency at 5kHz vs. % change in PM data across all sensors, (b): PM data of Sensor 4 and reference sensor [Day 1 of deployment], (c): PM data of Sensor 4 and reference sensor [Last day of deployment], (d): FFT at 5kHz for Sensor 4 on day 1 and 15, (e): Sensor 4 PM data before and after compensation along with reference data, (f): PM RMSE error before and after drift compensation.

approach identified Sensor 4 to be drifted. Figure 13(c) shows PM data of sensor 4 has drifted significantly with a PM error of 32.9 compared to the reference sensor. Upon manual inspection, we found that the FAN’s speed has changed by 36% inducing drift in PM data. Hitherto, such sensor drifts would go unnoticed unless there is sensor redundancy, which can now be accurately detected using the proposed CurrentSense approach.

6.5 Drift compensation results

As described earlier, there is a linear relationship between % change in PM data and % change in dominant frequency observed, in case of FAN drift and % change in magnitude of dominant frequency, in case of LED drift as shown in Figure 10(d) and 11(d), respectively. We use this mapping for drift compensation, where we perform a simple linear regression to derive PM change based on the measured % change in dominant frequency. Note that, the compensation is applied only if the % change in dominant frequency is within an acceptable threshold. If the % change is higher than the threshold then the sensor is brought back to the lab for co-location.

Figure 13(d) shows the change in dominant frequency at 5kHz signal for Sensor 4 on day 1 and day 15, clearly indicating FAN drift. In order to compensate for the drift, we compute the % change in dominant frequency (in this case it was 20% change as compared to day 1) and then apply appropriate compensation. Figure 13(e) shows PM data from the reference sensor, Sensor 4 (drifted at day 15) and the compensated Sensor 4 PM data based on CurrentSense. We can clearly see the mismatch in PM data between the drifted and the reference sensor, with a PM error of 32.9. Upon applying the compensation based on the % change in dominant frequency, PM data of the drifted sensor closely matches the reference sensor and the PM error before and after drift compensation decreased from 32.9 to 3.66. Thus validating the efficacy of CurrentSense approach towards drift detection and compensation.

Figure 13(f) shows the PM error before ($Error_B$) and after ($Error_A$) applying the drift compensation for all the 9 drifted sensors. We can see that the RMSE error between the reference sensor and the drifted sensor after applying the compensation reduces drastically across all the sensors. Overall, the CurrentSense approach was able to reduce the average PM error across all drifted sensors by 86%, thus addressing the $PM_{2.5}$ data inaccuracy challenges.

7 APPLICABILITY OF CURRENTSENSE TO OTHER SENSOR TYPES

We now show the applicability of CurrentSense fingerprints to detect faults in three popular environmental sensors, namely, CO_2 , multi-gas and a digital temperature sensor.

7.1 Fault detection in CO_2 sensors

We selected a popular digital CO_2 sensor [14], which has three main components, an LED, photodiode and an internal microcontroller. The key idea is that there are regions of the IR spectrum, where specific gases have a higher absorption rate, for example, the wavelength for CO_2 is $4.26\mu m$, and O_3 is $9.0\mu m$. Thus by monitoring the light scattering at particular wavelengths we can estimate the gas concentration. Figure 14(a) shows the CO_2 concentration in parts per million (ppm) for a working and faulty CO_2 sensor. We can see that it is non-trivial to distinguish working/faulty sensor data without contextual information. We consider CO_2 sensor as a blackbox sensor and Figure 14(b) shows the overall current consumption for the CO_2 sensor. CurrentSense fingerprint accurately captures the current consumption of a CO_2 sensor (periodic turn ON of the LED component) and is distinguishable for a working, drifted (LED drift) and a faulty CO_2 sensor. Thus, showcasing the efficacy of CurrentSense approach to detect faults and drifts.

7.2 Fault detection in multi-gas sensors

We now show the working of CurrentSense on another blackbox sensor, namely, a multi-gas sensor. Multi-gas sensor is a MEMS based sensor that combines multiple metal-oxide sensing elements to measure a total VOC signal. We use a digital Multi-gas sensor [7], where the sensing principle is based on a heated film of metal-oxide nanoparticles. The adsorbed oxygen on the metal-oxide reacts with the target gas and thereby releasing electrons. This results in a change of the electrical resistance that is measured by the sensor. The current consumption is mostly driven by the heater controller to maintain the required temperature on the film and additional electronics for signal processing. Figure 14(c) shows the CurrentSense fingerprint of a working and faulty multi-gas sensor. Here the fault was due to malfunction of heater controller and can be observed in the change in current consumption.

7.3 Fault detection in temperature sensors

Finally, we evaluate CurrentSense approach on a commonly used digital temperature sensor – DS18B20 [15]. This sensor embodies the most basic digital sensor, which uses only one data line along with ground, to transmit temperature data to the device. At power-up, the sensor is in the IDLE state and when the IoT device issues command of temperature conversion, the sensor goes into ACTIVE state and converts the temperature measured by the analog transducer, and writes it to the buffer. The sensor then goes back to its IDLE state and the IoT device can now read the data in the buffer. Figure 14(d) shows the temperature data of a working and a faulty sensor. For both the sensors, the temperature conversion command is triggered to put the sensor in ACTIVE state after every

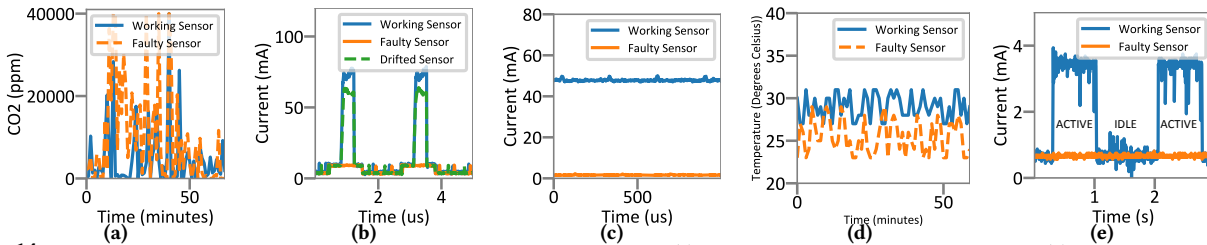


Figure 14: Applicability of CurrentSense fingerprints to other environmental sensors. (a): CO₂ data for working and faulty sensors, (b): CurrentSense fingerprint of working, drifted and faulty CO₂ sensors, (c): CurrentSense of working and faulty multi-gas sensor, (d): temperature data for working and faulty sensors, (e): CurrentSense of working and faulty temperature sensor.

1 second of IDLE time. We can see that the temperature data from both the sensors are in valid range and non-trivial to distinguish the faulty sensor. The faulty sensor in this case had a malfunctioning transducer due to water/heat damage. In Figure 14(e) we can clearly see that the current consumption of the working sensor in ACTIVE state is distinct from that of the faulty sensor. Thus we can detect faults in black box sensors by just analyzing the CurrentSense fingerprints. We are currently working with partners to integrate CurrentSense into other sensors and deploy them at scale.

8 CONCLUSION

This paper presents a systematic approach to detect faults and drifts in environmental IoT sensors, by devising a novel sensor fingerprint, called CurrentSense. CurrentSense can be used to effectively detect faults and drift that could not be diagnosed before without additional contextual information or sensor redundancy. We presented extensive experimental results from both controlled experiments and real-world deployment to showcase the efficacy of CurrentSense. The proposed approach significantly outperforms the state of the art anomaly detection algorithms. Our binary and multi-class classification models can run on resource-constrained IoT devices and has an accuracy of 98% and 96.5% in detecting faults. Further, CurrentSense approach was able to reduce the average PM error across all drifted sensors in our deployment by 86%. Finally, we also show that CurrentSense fingerprint is non-intrusive, and can be applied to a wide range of sensors to monitor sensor health and improve sensor data quality.

REFERENCES

- [1] 2019. AirCasting. <http://aircasting.org>.
- [2] 2019. Array of Things. <https://arrayofthings.github.io>.
- [3] 2019. BLDC Motor Control with Hall Effect Sensors Using the 9S08MP. <https://www.nxp.com/docs/en/application-note/AN4058.pdf>.
- [4] 2019. Clarity. <http://joinclarity.io>.
- [5] 2019. Honeywell HPM Series Particulate Matter Sensor HPM115S0. <https://sensing.honeywell.com/HPMA115S0-XXX-particulate-matter-sensors>.
- [6] 2019. IR LED lifetime tests. <https://scitec.uk.com/irleds/lifetime-tests>.
- [7] 2019. Multi-Gas, Humidity and Temperature Module SVM30. <https://www.sensirion.com/en/environmental-sensors/>.
- [8] 2019. OpenSense, ETH Zurich. <http://www.opensense.ethz.ch/>.
- [9] 2019. Particulate Matter Sensor SPS30. <https://www.sensirion.com/en/environmental-sensors/particulate-matter-sensors-pm25/>.
- [10] 2019. The Plantower PMS5003 and PMS7003 Air Quality Sensor. <http://www.plantower.com/en/content/?110.html>.
- [11] 2019. Texas Instruments INA4180 Quad-channel, 26V current sense amplifier for cost-sensitive systems. <http://www.ti.com/product/INA4180>.
- [12] 2019. Z-score. https://en.wikipedia.org/wiki/Standard_score.
- [13] 2020. AirBox. <http://pm2.5.taipei/>.
- [14] 2020. CO2 and RHT Sensor Module. <https://www.sensirion.com/en/environmental-sensors/carbon-dioxide-sensors/carbon-dioxide-sensors-co2/>.
- [15] 2020. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [16] 2020. How data quality makes IoT projects more profitable. <https://www.ionow.com/2019/11/28/100178-data-quality-makes-iot-projects-profitable/>.

- [17] F MJ Bulot, S J Johnston, P J Basford, G L Foster, A KR Morris, S J Cox, and M Loxham. 2019. Long-term field comparison of multiple low-cost particulate matter sensors in an outdoor urban environment. *Scientific reports* (2019).
- [18] Tusher Chakraborty, Akshay Nambi, Ranveer Chandra, Rahul Sharma, Manohar Swaminathan, Zerina Kapetanovic, and Jonathan Appavoo. 2018. Fall-curve: A novel primitive for IoT Fault Detection and Isolation. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM SenSys, 95–107.
- [19] Ling-Jyh Chen, Yao-Hua Ho, Hsin-Hung Hsieh, Shih-Ting Huang, Hu-Cheng Lee, and Sachit Mahajan. 2017. ADF: An anomaly detection framework for large-scale PM2.5 sensing systems. *IEEE Internet of Things Journal* 5, 2 (2017), 559–570.
- [20] Yun Cheng, Xiaoxi He, Zimu Zhou, and Lothar Thiele. 2019. ICT: In-field Calibration Transfer for Air Quality Sensor Deployments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 6.
- [21] 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 *Proceedings of the 12th ACM SenSys*. ACM, 251–265.
- [22] M. El Hachemi Benbouzid. 2000. A review of induction motors signature analysis as a medium for faults detection. *IEEE Trans. on Ind. Electronics* 47, 5 (2000).
- [23] Xinwei Fang and Iain Bate. [n.d.]. Issues of using wireless sensor network to monitor urban air quality. In *Proceedings of the First ACM Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems*.
- [24] Paula Fraga-Lamas, Tiago Fernández-Caramés, Manuel Suárez-Albela, Luis Castedo, and Miguel González-López. 2016. A review on internet of things for defense and public safety. *Sensors* 16, 10 (2016), 1644.
- [25] T. Ganu, D. Rahayu, D. P. Seetharam, R. Kunnath, A. P. Kumar, V. Arya, S. A. Husain, and S. Kalyanaraman. 2014. SocketWatch: An autonomous appliance monitoring system. In *2014 IEEE PerCom*. 38–43.
- [26] Mohamed-Faouzi Harkat, Majdi Mansouri, Mohamed Nounou, and Hazem Nounou. 2018. Enhanced data validation strategy of air quality monitoring network. *Environmental research* 160 (2018), 183–194.
- [27] Milena J, A Bartonova, and et al. 2015. On the use of small and cheaper sensors and devices for indicative citizen-based monitoring of respirable particulate matter. *Environmental Pollution* 206 (2015), 696–704.
- [28] Ashish Kumar, Saurabh Goyal, and Manik Varma. 2017. Resource-efficient machine learning in 2 KB RAM for the internet of things. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1935–1944.
- [29] Edward S Macias and Rudolf B Husar. 1976. Atmospheric particulate mass measurement with beta attenuation mass monitor. *Environmental Science & Technology* 10, 9 (1976), 904–907.
- [30] M Mueller, J Meyer, and C Hueglin. 2017. Design of an ozone and nitrogen dioxide sensor unit and its long-term operation within a sensor network in the city of Zurich. *Atmospheric Measurement Techniques* 10, 10 (2017), 3783–3799.
- [31] Kevin Ni, Nithya Ramanathan, Mohamed Nabil Hajj Chehad, Laura Balzano, Sheela Nair, Sadaf Zahedi, Eddie Kohler, Greg Pottie, Mark Hansen, and Mani Srivastava. 2009. Sensor network data fault types. *ACM TOSN* 5, 3 (2009), 25.
- [32] Aakash C Rai, Prashant Kumar, Francesco Pilla, Carlo Ratti, Ansar Yasar, and David Rickerby. 2017. End-user perspective of low-cost sensors for outdoor air pollution monitoring. *Science of The Total Environment* 607 (2017), 691–705.
- [33] Haroon Rashid, Nipun Batra, and Pushpendra Singh. 2018. Rimor: Towards identifying anomalous appliances in buildings. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys)*. 33–42.
- [34] A B Sharma, L Golubchik, and R Govindan. 2010. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM TOSN* 6, 3 (2010), 23.
- [35] S. R. Shaw, L. K. Norford, D. Luo, and S. B. Leeb. 2002. Detection and Diagnosis of HVAC Faults via Electrical Load Monitoring. *HVAC&R Research* 8, 1 (2002).
- [36] WHO. 2019. 7 million premature deaths annually linked to air pollution. <https://www.who.int/mediacentre/news/releases/2014/air-pollution/en/>.
- [37] Jiawen Wu and Guanghui Li. 2019. Drift Calibration Using Constrained Extreme Learning Machine and Kalman Filter in Clustered Wireless Sensor Networks. *IEEE Access* 8 (2019), 13078–13085.
- [38] Yu-Fei Xing, Yue-Hua Xu, Min-Hua Shi, and Yi-Xin Lian. 2016. The impact of PM2.5 on the human respiratory system. *Journal of thoracic disease* 8, 1 (2016).