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

Sumukh Marathe, Akshay Nambi, Nishant Shrivastava, Manohar Swaminathan, Ronak Sutaria
13 ZB
Produced by IoT Devices in 2019

Determining Sensor Data Quality is an Imperative
Determining Sensor Data Quality is an Imperative

- Working
- Faulty
- Drifted
The Challenge

Typically, A Sensor Keeps Sending Data After It Fails
Every electrical sensor draws current from the IoT device.

Damage to a sensor affects its current consumption.

We can derive an electrical fingerprint that differs between Working, Faulty and Drifted sensors.
CurrentSense

1. Distinct for a working, drifted, and faulty sensor

2. Quantifies the amount of drift

3. Independent of the measured phenomena

4. Non-intrusive with no or minimal hardware modification
Agenda

2. Background and PM$_{2.5}$ Sensor Faults
3. CurrentSense and its Working
4. Experimental Setup
5. Fault Detection and Isolation
6. Detecting and Measuring Drift
7. Applicability of CurrentSense to other Sensor Types
Agenda

2. Background and PM$_{2.5}$ Sensor Faults

3. CurrentSense and its Working

4. Experimental Setup

5. Fault Detection and Isolation

6. Detecting and Measuring Drift

7. Applicability of CurrentSense to other Sensor Types
Agenda

2. Background and PM$_{2.5}$ Sensor Faults

3. CurrentSense and its Working

4. Experimental Setup

5. Fault Detection and Isolation

6. Detecting and Measuring Drift

7. Applicability of CurrentSense to other Sensor Types
What is PM$_{2.5}$?
**PM$_{2.5}$**
Particulate Matter two and one half microns or less in width

$30 - $100

Frequent Data Faults
Low-cost PM$_{2.5}$ Sensor and its working

1. Fan Creates Controlled Airflow
2. Particles travel from inlet to outlet, passing through light source
3. Light scatters as it hits the particles
4. Scattered light is detected by photo diode and converted to a mass concentration output
What is a Data Fault?
Catastrophic Faults

ie: Fan stops spinning

Case 1: Mimicking Data

The faulty sensor mimics working sensor data.

Case 2: Anomalous Data

The faulty sensor reports anomalous data.
Sensor Drift

ie: LED Light intensity changes

Low cost PM$_{2.5}$ require calibration to estimate correctly

After deployment, this calibration may not remain valid as sensors wear.

This loss of calibration is very difficult to detect.
Related Work

Data-centric efforts

System-centric efforts

Current Signature Analysis
Related Work: Data-centric efforts

Data of the sensor is analyzed and a fault is identified if the data is out of bounds of the expected behavior.

- A faulty sensor can mimic non-faulty data
- An anomalous sensor reading need not represent faulty data

Fault Detection in Air Pollution Sensors
- Use the sensor’s placement in time/space to detect anomalies
- Use redundant sensors
- Compare sensor readings to some predicted value
Related Work: System-centric efforts

Use the sensor’s voltage response when being turned off to characterize sensor fault.

- Works only for analog sensors where a sensor’s output voltage can be measured directly.
- Fall-curve is designed to only detect faults, and cannot be used to detect and measure sensor drift.
- Fall-curve requires the sensor to be powered down to determine its status.
Related Work: Current Signature Analysis

There are other domains in which current signature analysis has been used to detect faults.

Examples
- Motor Current Signature Analysis (MCSA)
- HVAC.
- SocketWatch.
CurrentSense performs current monitoring
For fault detection and isolation in low-cost IoT sensors
Agenda

2. Background and PM$_{2.5}$ Sensor Faults

3. CurrentSense and its Working

4. Experimental Setup

5. Fault Detection and Isolation

6. Detecting and Measuring Drift

7. Applicability of CurrentSense to other Sensor Types
Agenda

2. Background and PM$_{2.5}$ Sensor Faults
3. CurrentSense and its Working
4. Experimental Setup
5. Fault Detection and Isolation
6. Detecting and Measuring Drift
7. Applicability of CurrentSense to other Sensor Types
Deployment details

51

Devices

8

Months

7

Days between Inspections
Deployment details

33 Working
9 Faulty
9 Drifted

51 Devices
Agenda

2. Background and PM$_{2.5}$ Sensor Faults

3. CurrentSense and its Working

4. Experimental Setup

5. Fault Detection and Isolation

6. Detecting and Measuring Drift

7. Applicability of CurrentSense to other Sensor Types
Agenda

2. Background and PM$_{2.5}$ Sensor Faults
3. CurrentSense and its Working
4. Experimental Setup
5. Fault Detection and Isolation
6. Detecting and Measuring Drift
7. Applicability of CurrentSense to other Sensor Types
Controlled Experiments

Fan fault injected at $T = 50$

CurrentSense Features change Dramatically at 5kHz

CurrentSense Features do not change at 30Hz

PM data

FFT @ 5kHz

FFT @ 30Hz
Conclusion: We can accurately detect and isolate faults by analyzing CurrentSense fingerprints.
Real-world deployment results

1 Measurement Per Minute

10 Fingerprints Per Week Subsampled
since ground truth was taken weekly

\[ 10 \times 34 \times 51 = 17340 \]

Fingerprints \times Weeks \times Devices = \text{Total Fingerprints}
### Real-world deployment results

<table>
<thead>
<tr>
<th></th>
<th>Working</th>
<th>Fan Fault</th>
<th>LED Fault</th>
<th>Complete Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Fan Fault</td>
<td>0.04</td>
<td>0.96</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LED Fault</td>
<td>0.05</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td>Complete Fault</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.97</td>
</tr>
</tbody>
</table>

- **% Precision**: 97.4%
- **% Recall**: 99.8%
- **% F1**: 98.5%
# Real-world deployment results

<table>
<thead>
<tr>
<th></th>
<th>Working</th>
<th>Fan Fault</th>
<th>LED Fault</th>
<th>Complete Fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Fan Fault</td>
<td>0.04</td>
<td>0.96</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>LED Fault</td>
<td>0.05</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td>Complete Fault</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Conclusion: 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.
Comparison with data-centric algorithms
CurrentSense VS ADF

An Anomaly Detection Framework for Large-Scale PM$_{2.5}$ Sensing Systems
Comparison with data-centric algorithms

Spacial Anomaly
Hyper-local variations in the pollution levels

F₁ = 77.8%

Temporal Anomaly
Distribution of particle matters is generally non-stationary

F₁ = 67.2%

Spatio-temporal Anomaly

F₁ = 33.0%
Discussion
- **Flexible.** Applies to a wide variety of sensors.

- **Rigorously Tested.** Example of thorough experimentation.

- **Relevant.** This could feasibly be rolled out in the near future.

- **Limited.** Cannot detect faults due to environmental factors

- **Costly.** Current amplifiers are expensive relative to the cost of pollution sensors
Any Questions?
What benefits/challenges would there be if a device manufacturer wanted to ship devices with CurrentSense already loaded?
What other applications are there for this “electrical fingerprint”? 
In what contexts is drift correction appropriate? Are there any it is not appropriate in?
Are there any digital sensors this approach would not work well for?