Location Monitoring Framework for Citizen Science Sensors

Problem

Community-driven air quality sensors provide a unique insight into current AQI readings that government-maintained air quality monitoring stations cannot.

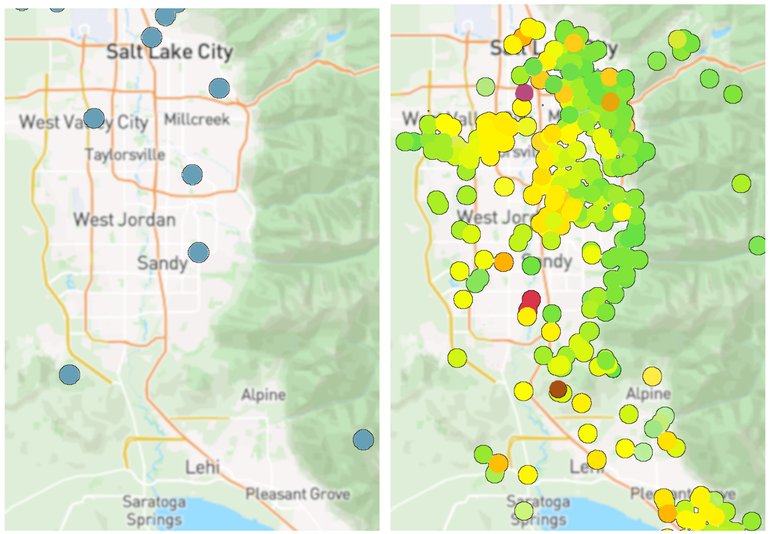
Due to the large amount of these nodes and the large geographic region they cover, these inexpensive sensors provide a highly granular, real-time perspective on the current state of air-quality.

Any willing participant with one of these commodity sensors can install the sensor outside their home/business and share raw measurements to a public service that represents their readings on a map.

Citizen science sensors are made to be inexpensive, the majority of them **do not have any hardware such as a GPS module to determine their location in real time.**

Any forgetful person, careless user, or malicious actor can register their air quality sensor to whatever location on the map without any verification.

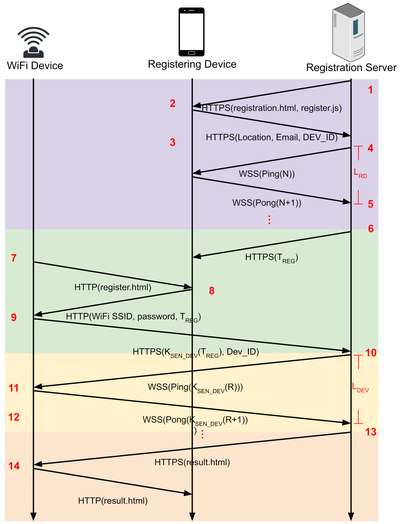
Such oversights in location verification of deployed air quality sensors can **misrepresent data that many rely on to take the appropriate health precautions.**



Number of government-maintained air quality stations (left) and commodity air quality sensors deployed by citizen scientists (right).

Methodology

To ensure that a commodity air quality sensor is actually in the location where it is registered, we need to ensure the validity of two things: **the location of the sensor during registration and the fact that the sensor has not moved after being installed.**



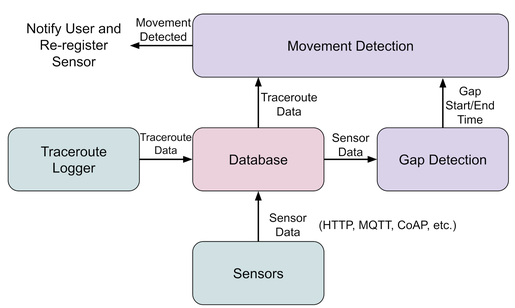
Communications between sensor, smartphone, and server during registration.

Relocation Detection
Since we cannot modify the sensor with extra hardware to detect changes in physical movement, **we analyze the sensor's network patterns to determine whether it has moved or not.**

(1) Air quality data is uploaded to a central database and the server logs the network traffic into a rolling log.

(2) If the beginning of a gap in data is detected, the sensor is placed into a challenged state until the gap of the data ends.

(3) Our algorithm compares previous network patterns with the current ones. If they differ within a certain range, the sensor has moved.



Movement detection algorithm being triggered by gaps in data

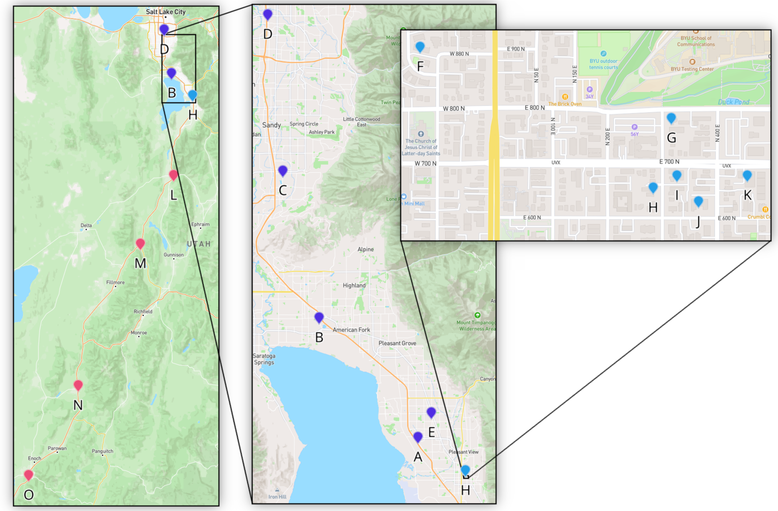
Determining the Sensor Location
Our framework does the following during the registration process:

(1) the service where the sensor is registered asks for geolocation data provided by the registering device (i.e GPS from a smartphone)

(2) the registration server measures the latency between itself and the registering smartphone

(3) the sensor is given access to WiFi and reports its latency to the server

(4) the server compares the latency between the registering device and sensor and determines whether or not they are in the same location.



Regions where the change of location detection framework was tested starting from a rural area (left), an interstate area (center), and a intra-city area (right).

Results

To test our relocation detection we logged network patterns for 15 different nodes for a 20 week collection period.

	L	M	N	O
L	99.1	0	0	0
M	0	99.9	0	0
N	0	0	99.9	23.9
O	0	0	1.8	98.0

(a) Rural Area

	A	B	C	D	E
A	99.9	0	0	0	14.6
B	0	96.39	0	0	0
C	0	0	99.7	0	0
D	0	0	0	98.0	0
E	11.3	0	0	0	99.8

(b) Inter-City

	F	G	H	I	J	K
F	99.9	0	66.6	0	0	66.6
G	0	96.3	0	0	0	0
H	66.6	0	99.9	0	0	66.6
I	0	0	0	94.9	0	0
J	0	0	0	0	99.9	0
K	66.6	0	66.6	0	0	99.9

(c) Intra-City

Across all geographic regions, we observed that there was **a strong correlation between the network patterns from the same node**. As seen on the diagonals in each comparison table for each region, the system could identify at least 94.9% of each node's own network patterns.

We note that the ability to identify movement is consistent across all regions.

Acknowledgements

A special thank you to Milt and Heidi Shipp. Without their generous donation this work would not have been possible.