

# Enhancing Mobile Apps To Use Sensor Hubs Without Programmer Effort

Nishad Gothoskar

ngothosk@andrew.cmu.edu

---

## 1: Summary

---

Sensors are an important component in the future of technology. Heart rate sensors, GPS, accelerometer, gyroscope, etc. Their continuous sensing allows our devices to gather immense amounts of information and adapt, learn, and provide useful data accordingly. But with this continuous sensing comes a lot of energy consumption. But contrary to popular belief, this energy drain isn't just from keeping the sensors on, but for keeping the processor running when in fact it could be in idle. That is what the sensor hub does. It allows the energy-demanding processor to go to idle while it runs and deals with the sensors. The major problem this paper addresses is that it is hard to properly use and program with the sensor hub.

MobileHub works by learning how an app uses a sensor and rewriting the application to properly use the sensor hub. This allows us to form a "new" APK that is more energy efficient but works just the same as the original application. So this "learning" process involves tracking the flow of sensor data and what changes it makes in the program state and the program itself. This way, TaintDroid can learn what sensor actions and events trigger events and changes in the application and therefore, the sensor hub can just learn to detect this and send signals to the application accordingly.

Then comes the buffering step. MobileHub learns which sensor actions cause events immediately and will send those events immediately because otherwise the application won't react in time and won't function the way it was supposed to. On the other hand, all other sensor data that is captured can be buffered because whether it is transmitted now or later, it won't make a difference in the application state (hopefully).

But since this is machine learning, you can't have a deterministic solution. So the results in some apps were nothing or even detrimental. But, in the apps where the process did work correctly, there were power improvements of 70 to 80%

---

## 2: Strengths

---

- Why leave the processor on when you can have a "dedicated" sensor hub that is meant for running the sensors?
- Energy proportionality we can design the hardware for what it'll be used for

---

## 3: Weaknesses

---

- Learning is deterministic so learning may not pick up certain subtleties on how the sensor data is used.
- Might make "wrong" changes to an application and it won't function properly

---

**4: Future Directions**

---

- Make smarter decisions on when it's better to just leave the app alone because too many changes are being made and the runtime of the app is being effected