Sylvia Jin

Microbug PCB

This little programmable “bug” can travel in a straight line, go in circles, and follow a line! All packed in a ~1” x 1.5” board, powered by a single 1.5V button cell battery.

How does it work?

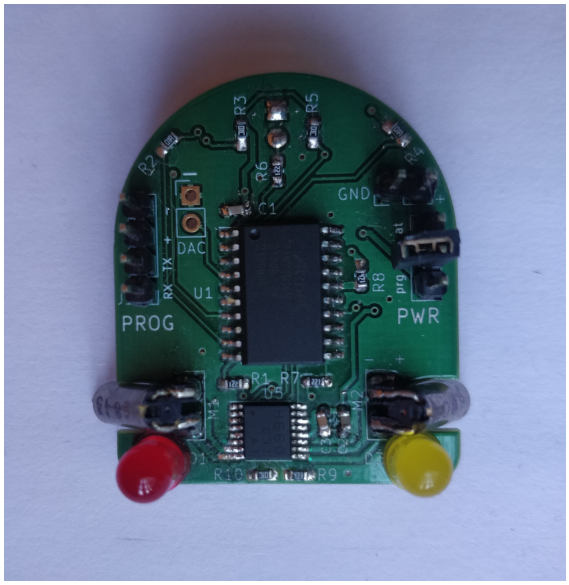
At its heart, the bug has an ATTiny1616 microcontroller that takes in programmed input and reads from 2 reflectance sensors, controlling a motor driver hooked up to 2 tiny motors, acting as the “wheels” for this bug. Because it’s programmable, it’s capable of:

- Going in a straight line: alternate turning on the left/right motor
- Circles: only turn on 1 motor
- Follow a line: programming either the left or right motor to turn on when the corresponding reflectance sensor detects something too bright. This requires finding out what “too bright” means through experimentation.

We switch between program mode and battery-powered mode by shorting a jumper between ground and either the positive battery terminal, or a programming pin connected to the TX/RX of the microcontroller.

Video + more pictures

Here it is in action! https://drive.google.com/file/d/1u6ReDlqAdUyclzlYbZF9ifZn9OFh6d3S/view



Top, with microcontroller and 2 LEDs

Bottom, with battery holder and reflectance sensors

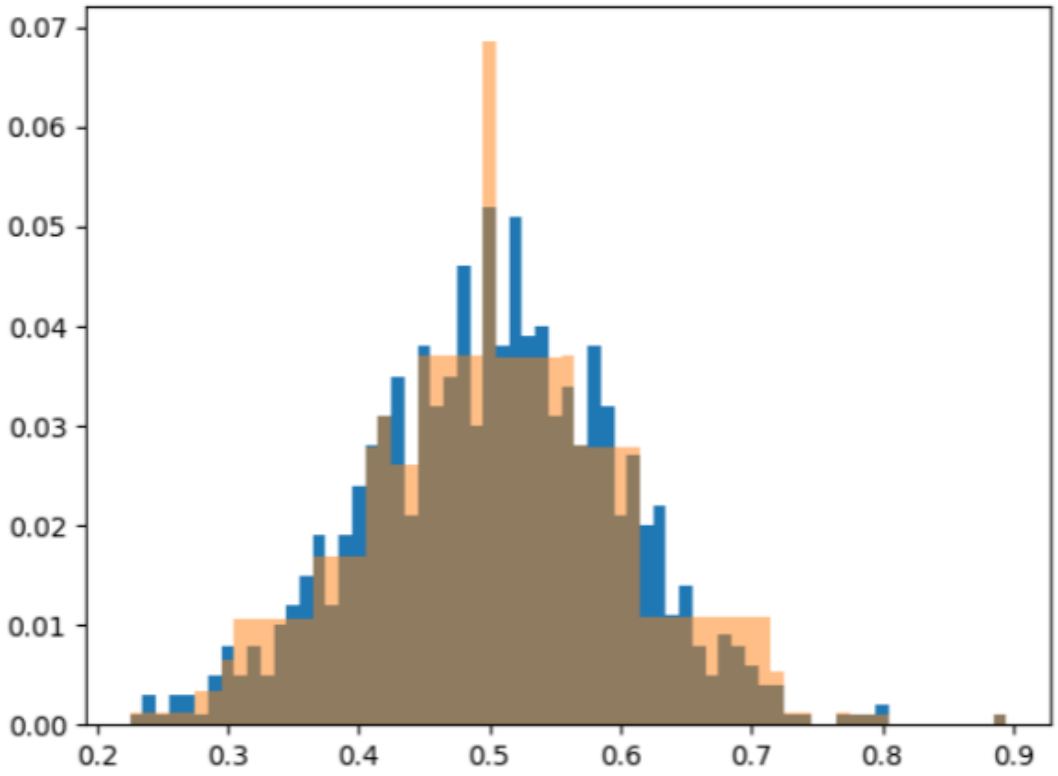
Sylvia Jin

Differential Privacy Implementation

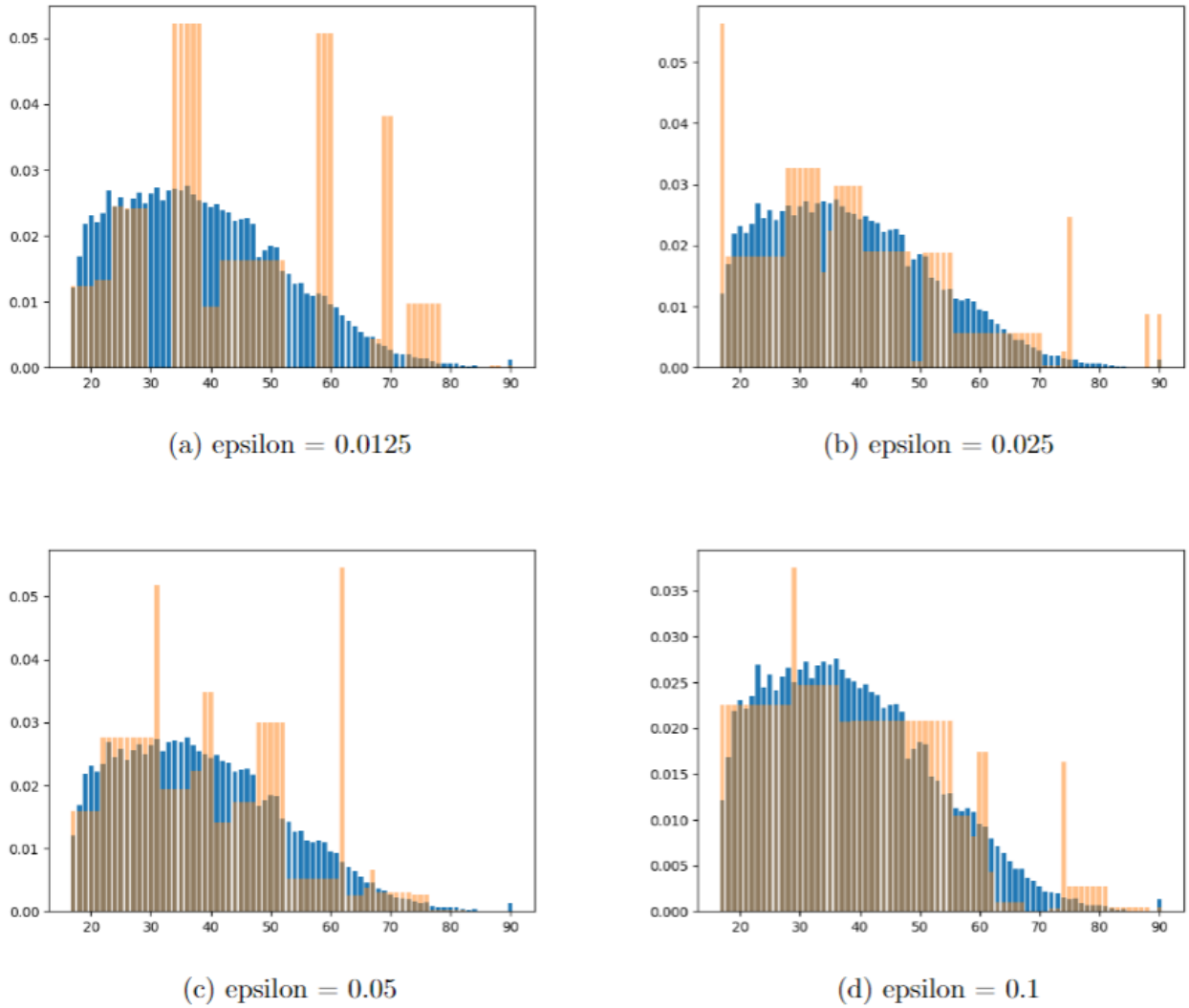
Suppose you had access to a database of medical records, where the contents had sensitive, potentially personally-identifying information. Anonymizing the data by just scrambling names is insufficient, since many “linkage attacks" make it possible to de-anonymize names given other biographical information. For example, suppose you knew a friend was hospitalized in summer 2021 for about a week, and you know their birthday and gender. Searching the database based on the other characteristics you know would be enough to look them up, since the chances of 2 people having the same birthday AND being hospitalized at the same time is very very low.

One solution to this is differential privacy, which adds some appropriately-scaled noise to any queries you make to your database. The essence of a differentially private data mechanism is to allow collection of aggregate statistics from databases, but NOT allow information that de-anonymizes any individual or small set of people. Mathematically, it means that removal of *any one input record* from any possible input dataset changes the probability the randomized computation produces *any possible* output by at most a factor of exp(ε).

We added noise according to the Laplace distribution. In the figure below, the blue samples are from a true normal distribution, while the orange overlay shows the differentially-private results, using ε=1.



We also used this on a real dataset (UCI's dataset of adults' ages) with varying values of ε. As shown, using smaller ε increases privacy but gives less accurate results. Using $\varepsilon \geq 0.05$ seems to be a sufficient compromise.



(a) epsilon = 0.0125

(b) epsilon = 0.025

(c) epsilon = 0.05

(d) epsilon = 0.1

Figure 3: Distributions with increasing epsilon values

Bigger organizations, such as the US Census, Google, and Microsoft have already put this into practice. The added noise has an average of 0, so it doesn't significantly affect statistical accuracy but still protects privacy, proving useful in machine learning algorithms that take in huge amounts of data.