

# Internet of Things

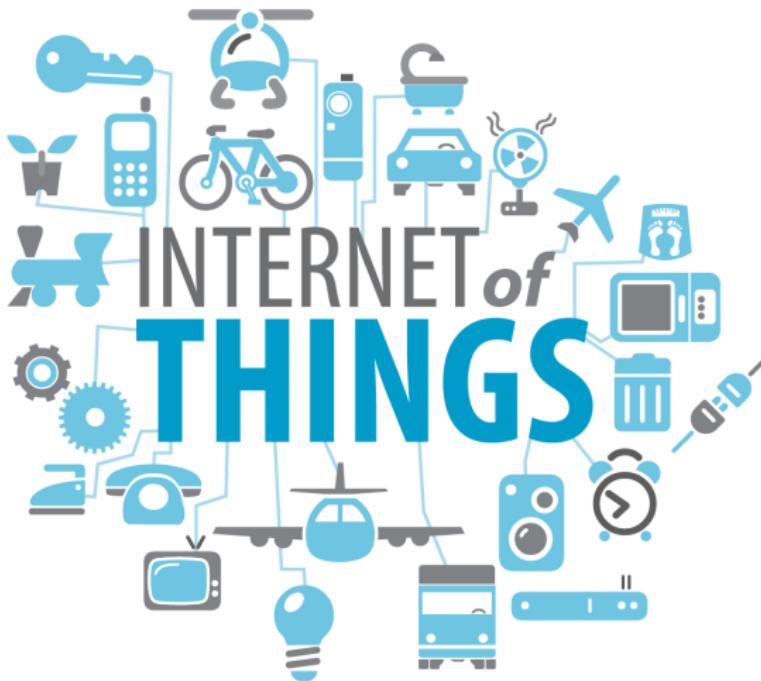
## Data Acquisition to Machine Learning.

*Ciira wa Maina*  
*cwamaina.dekut@gmail.com*

12th November, 2018

# The Internet of Things

- ▶ Connecting things over the internet
- ▶ We can measure physical states and use this information to guide actions



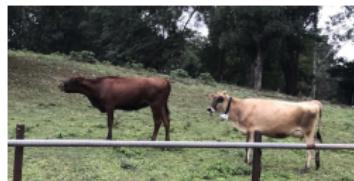
## Internet of Things

*Adding digital inputs/outputs to physical things - Jan Jongboom*



# Use Cases Closer Home

- ▶ Agriculture, Environmental Monitoring, ...



# Getting the Data

- ▶ Before data scientists can work their magic on data, it must be acquired
- ▶ Sensor systems provide a rich data source
- ▶ But data acquisition is not always easy...



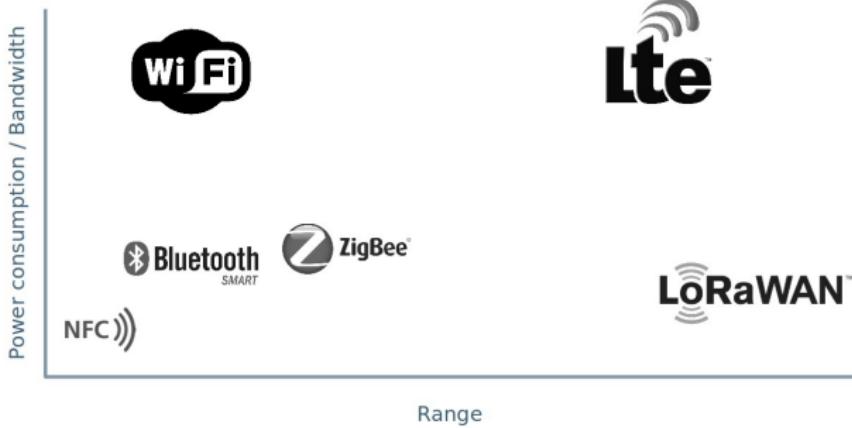
# Getting the Data

- ▶ It may also require armed escort!

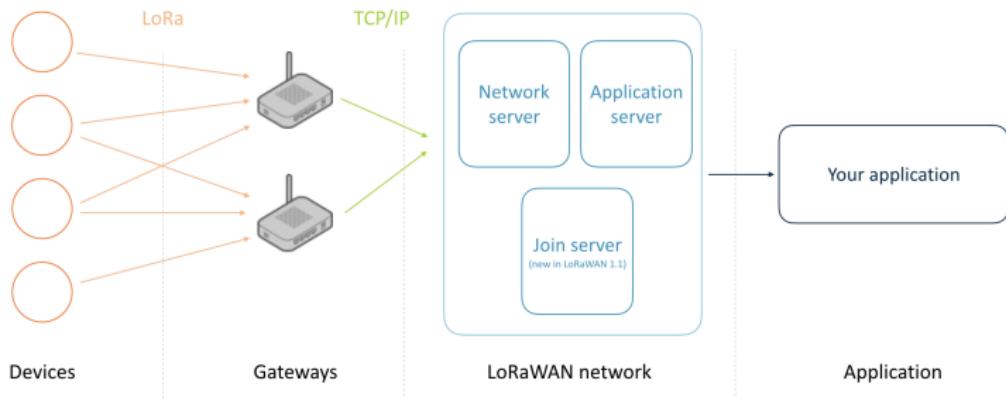


# LoRa

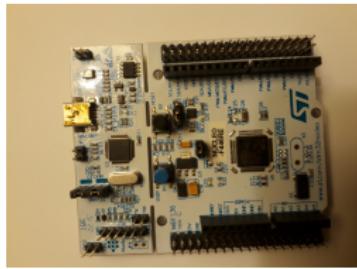
- ▶ Low power, long range network
- ▶ Ideal for low bandwidth situations such as sending sensor data



# LoRa Networks



# Devices



# Mbed OS

- ▶ A free, open-source operating system for embedded devices
- ▶ Ideal for IoT
- ▶ Allows development of applications in C/C++

**arm  
MBED**

# Mbed OS

- ▶ Mbed OS implements
  - ▶ Radio drivers
  - ▶ Networking
  - ▶ Hardware drivers
  - ▶ Storage
- ▶ Allows rapid prototyping

# Mbed OS Code Development

- ▶ Online compiler <https://os.mbed.com/compiler>
- ▶ Offline CLI
- ▶ Try things out on the online simulator  
<https://labs.mbed.com/simulator>

Arm Mbed OS simulator

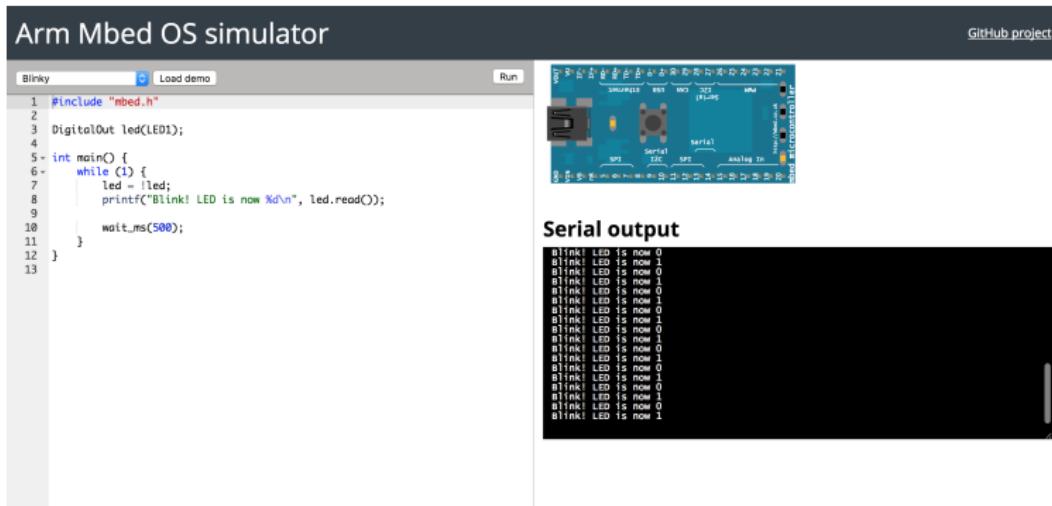
GitHub project

Blinky Load demo Run

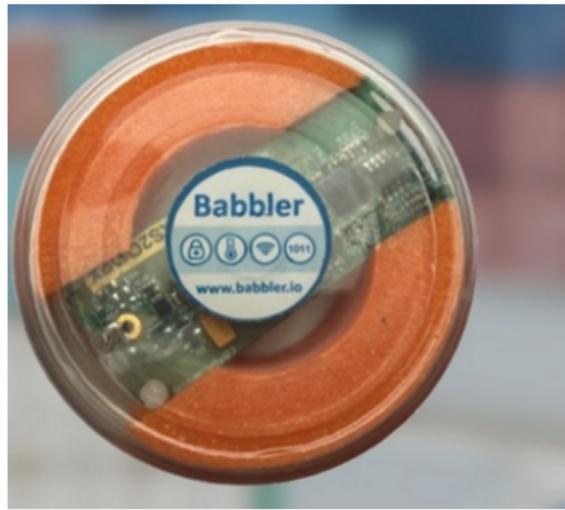
```
1 #include "mbed.h"
2
3 DigitalOut led(LED1);
4
5 int main() {
6     while(1) {
7         led = !led;
8         printf("Blink! LED is now %d\n", led.read());
9
10        wait_ms(500);
11    }
12 }
```

Serial output

```
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
Blink! LED is now 0
Blink! LED is now 1
```



- ▶ Babbler: A device to monitor cargo on transit



# Built with Mbed

- ▶ A device to monitor tilting of power poles



# Gateways



# Gateways

- ▶ Gateway placement is important! The higher the better



# Gateways

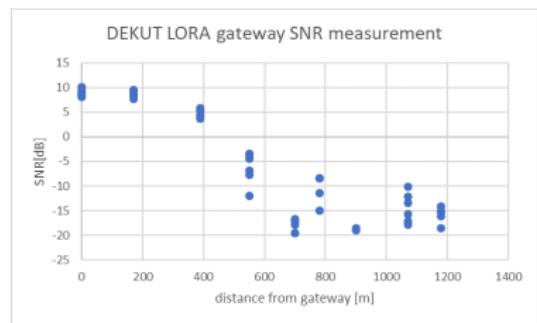
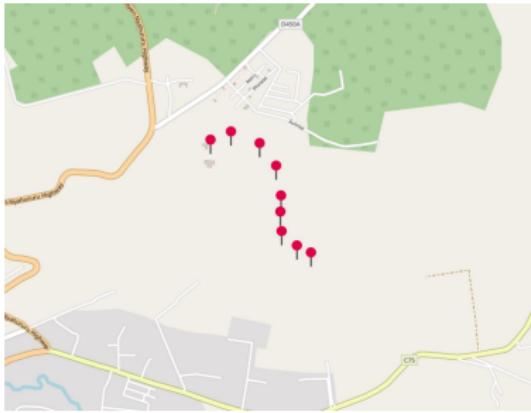
- ▶ Jared and Stephen taking the gateway to new heights.



# Gateways



# Gateways



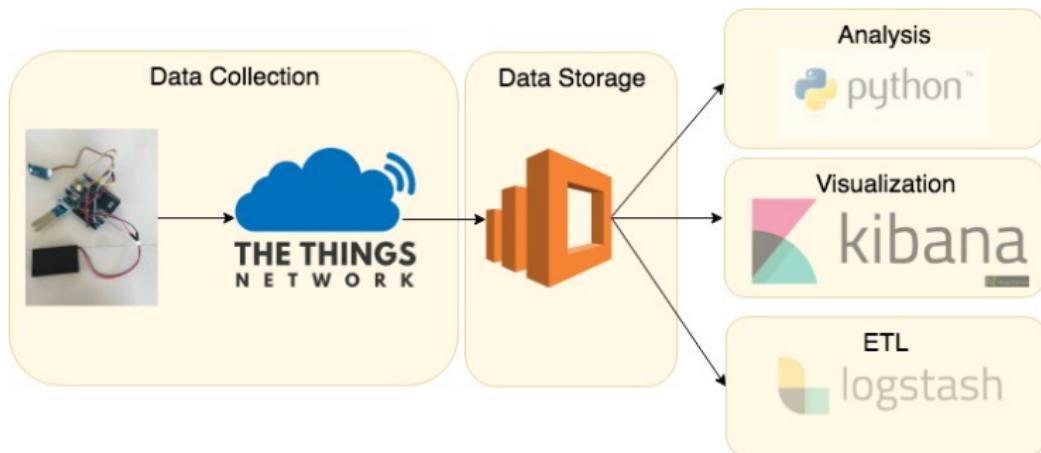
## Network Servers

- ▶ Servers that understand the LoRa protocol
- ▶ Companies offer this as a service



# Applications

- ▶ The network server forwards data to database
- ▶ Database could be from any cloud provider or local host
- ▶ Applications query data and use it to guide decisions



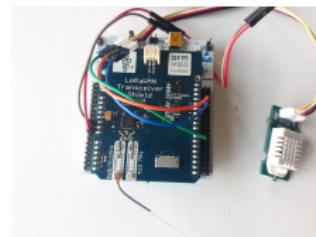
# Demo Application

- ▶ Temperature and humidity measurement
- ▶ Smart office motion detection
- ▶ Data transmitted to the Things Network
- ▶ Data stored on Amazon Web Services or Cayenne



# Sense the Environment

```
56 // Send a message over LoRaWAN
57 static void send_message() {
58     CayenneLPP payload(50);
59     int attempt = 0;
60
61     float temperature = 0.0f;
62     float humidity = 0.0f;
63     int error_code;
64
65     while (attempt++ < SENSOR_READ_ATTEMPTS) {
66         error_code = temperature_humidity_sensor.readData();
67         if (error_code != ERROR_NONE) {
68             printf("Error = %d\n", error_code);
69             wait_ms(SENSOR_WAIT_TIME_MS);
70             continue;
71         } else {
72             temperature = temperature_humidity_sensor.ReadTemperature(CELCIUS);
73             humidity = temperature_humidity_sensor.ReadHumidity();
74             break;
75         }
76     }
}
```



# Set Up Devices on The Things Network

The screenshot shows the The Things Network Console homepage. At the top, there's a navigation bar with the logo "THE THINGS NETWORK CONSOLE COMMUNITY EDITION", "Applications", "Gateways", "Support", and a user profile icon. Below the header, a greeting message says "Hi, cmaina!" with a small profile icon. A welcome message reads: "Welcome to The Things Network Console. This is where the magic happens. Here you can work with your data. Register applications, devices and gateways, manage your integrations, collaborators and settings." Two main sections are displayed: "APPLICATIONS" (represented by a stack of three blue rounded rectangles) and "GATEWAYS" (represented by a blue rounded rectangle with a circular antenna icon). The bottom of the page features a footer with navigation icons for back, forward, search, and other console functions.

# Set Up Devices on The Things Network

The screenshot shows the The Things Network Console interface. At the top, there's a navigation bar with links for Applications, Gateways, Support, and a user icon. Below the navigation is a breadcrumb trail: Applications > IoT-Demo-AWS > Devices > IoT-Demo-AWS-01. The main content area is titled "DEVICE OVERVIEW". It displays the following information:

- Application ID:** IoT-Demo-AWS
- Device ID:** IoT-Demo-AWS-01
- Activation Method:** ABP
- Device EUI:** 88 Bf 3F 01 9C Fa 0F A7
- Application EUI:** 70 03 05 7E 00 01 3F 95
- Device Address:** 28 01 18 C9
- Network Session Key:** (hex values: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00)
- App Session Key:** (hex values: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00)
- Status:** 13 hours ago
- Frames up:** 0 ([reset frame counters](#))
- Frames down:** 0

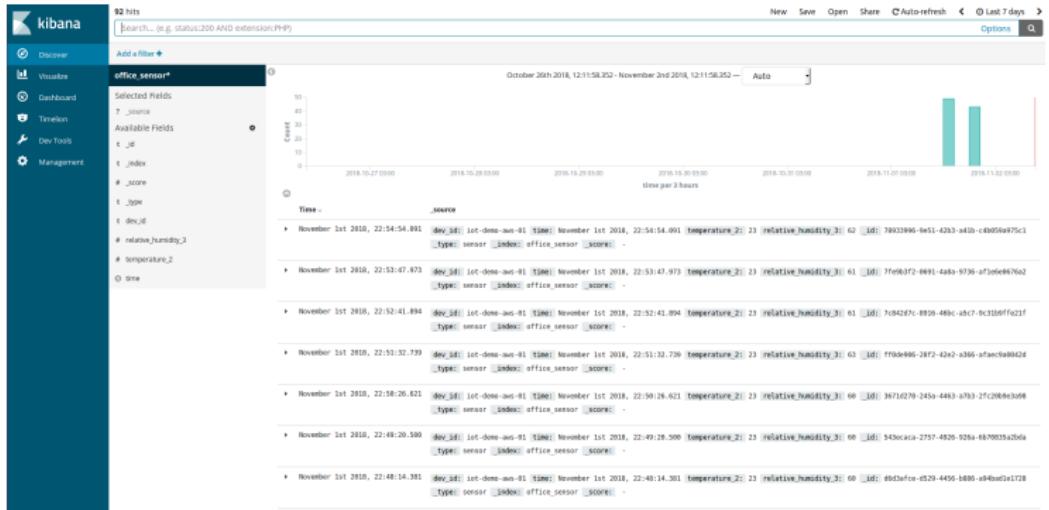
# Amazon Web Services Integration

The screenshot shows the AWS IoT Things console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account-related options. Below the navigation is a breadcrumb trail: 'Things > iot-demo-aws-01'. The main area has a dark header with the thing name 'iot-demo-aws-01' and a 'Actions' dropdown. On the left, a sidebar lists 'Details', 'Security', 'Groups', 'Shadow', 'Interact', 'Activity', 'Jobs', and 'Violations'. The 'Details' section is expanded, showing the 'Thing ARN' field which contains the value 'arn:aws:iot:eu-west-2:163325955848:thing/iot-demo-aws-01'. Below this is a 'Type' section with a search bar containing 'lorawan'. Under 'Attributes', there are four entries: 'app\_id' with value 'iot-demo-aws', 'dev\_ns' with value '008F9FD19CFADFA7', 'dev\_id' with value 'iot-demo-aws-01', and 'app\_ns' with value '7085D57ED0013F95'. The bottom of the page features standard AWS navigation icons.

# Amazon Web Services Integration - Database

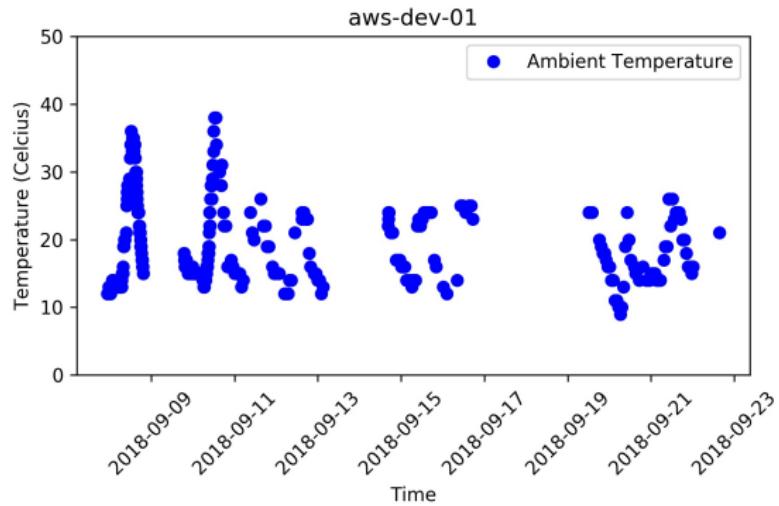
The screenshot shows the AWS Elasticsearch service interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups'. Below it, a sidebar lists 'Dashboard', 'My domains' (with 'iot-demo' selected), and 'Reserved instances'. The main content area is titled 'iot-demo' and includes tabs for 'Configure cluster', 'Modify access policy', 'Manage tags', 'Delete domain', and 'Upgrade domain'. A sub-menu for 'Overview' is open, showing details like 'Domain status: Active', 'Elasticsearch version: 6.3', 'Endpoint: https://search-iot-demo-gh44qpw6nqjllaz7wolqg.eu-west-2.es.amazonaws.com', 'Domain ARN: arn:aws:es:eu-west-2:163325165848:domain/iot-demo', and 'Kibana: https://search-iot-demo-gh44qpw6nqjllaz7wolqg.eu-west-2.es.amazonaws.com/\_plugin/kibana/'. The bottom of the screen features standard browser navigation icons.

# Amazon Web Services Integration - Database



# Act on Data

- ▶ Visualization
- ▶ Monitor variables and guide actions



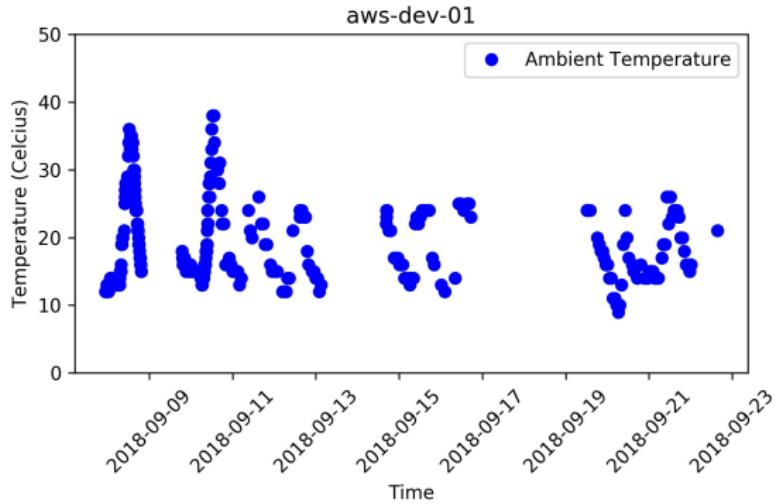
# Machine Learning Example

- ▶ Temperature data from the coffee farm at DeKUT
- ▶ Temperature influences susceptibility to fungal disease
- ▶ Current monitoring is manual



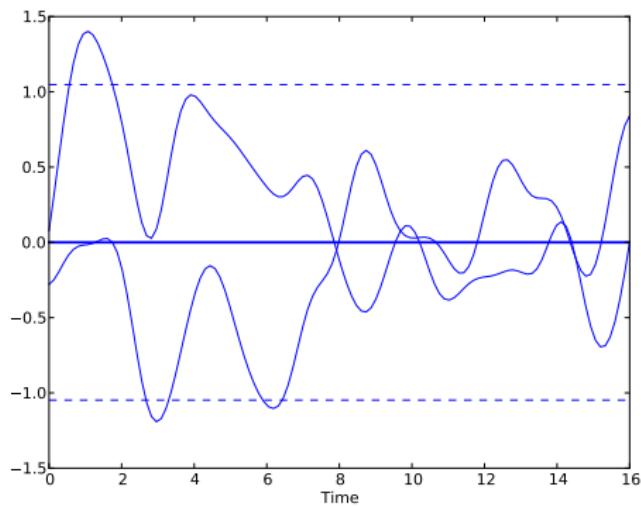
# Machine Learning Example

- ▶ Gaussian Process regression to fill missing values



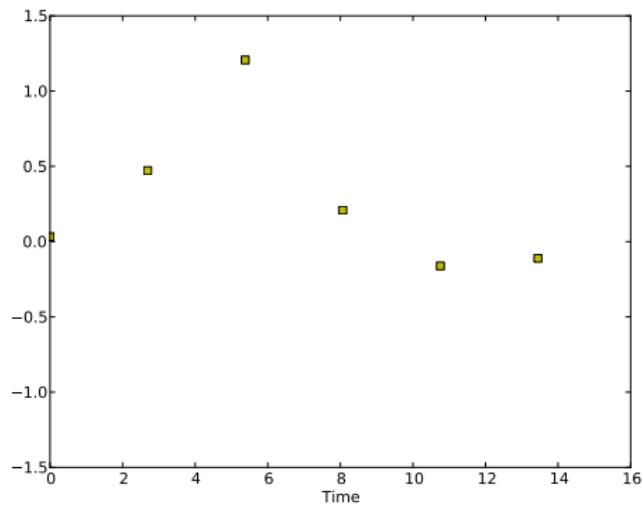
# Gaussian Processes

- ▶ A Gaussian process (GP) is a distribution over the space of functions.
- ▶ This distribution is completely specified by a mean function  $m(t)$  and a covariance function  $k(t, t')$ .
- ▶  $f(t) \sim \mathcal{GP}(m(t), k(t, t'))$



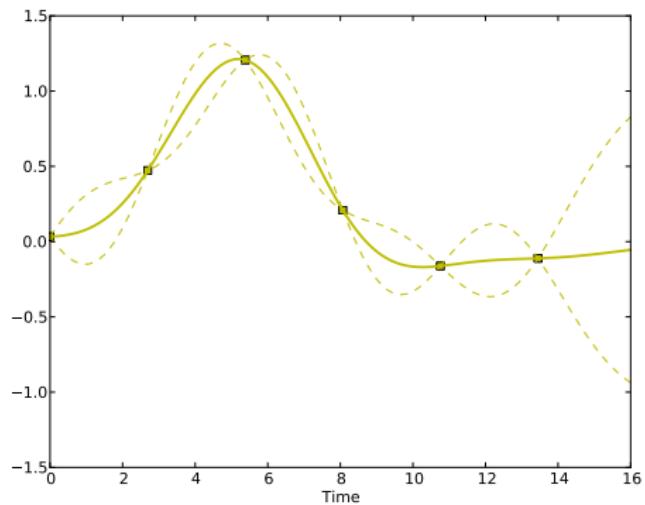
# Gaussian Processes cont

- ▶ Given observations we would like to infer an interpolant



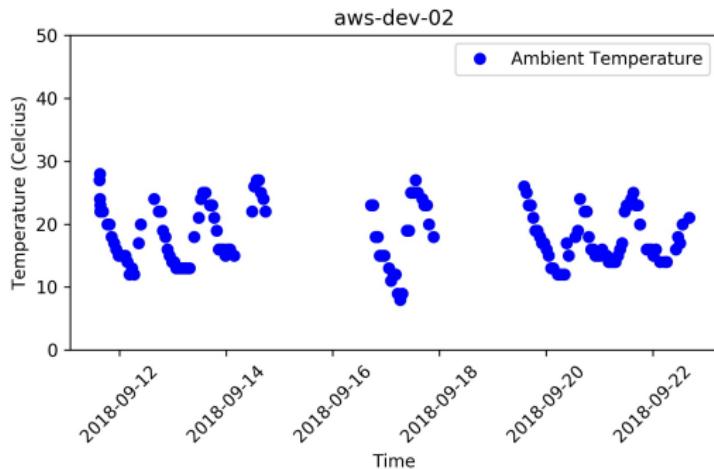
# Gaussian Processes cont.

- ▶ Posterior distribution of functions given observations



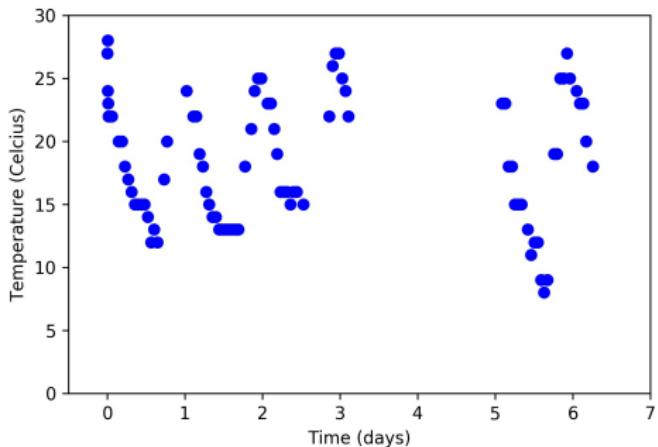
## Gaussian Processes cont.

- ▶ Original temperature data collected at the DeKUT coffee farm between 11th and 22nd September 2018.



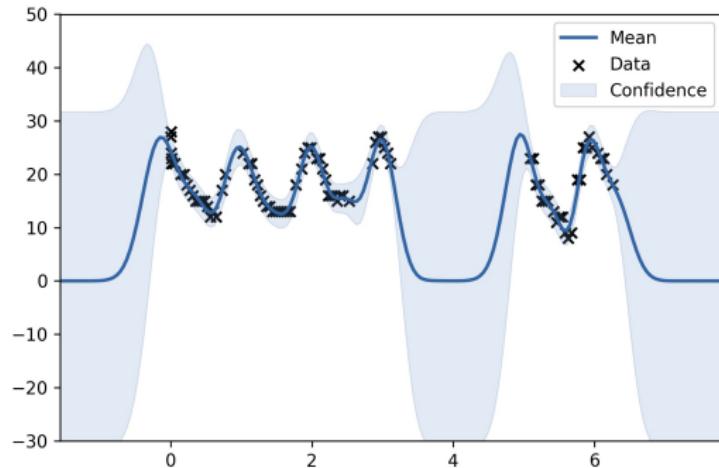
## Gaussian Processes cont.

- ▶ Same data with time axis normalized in terms of days (first 7 days)
- ▶ Missing data from the 3rd and 4th day



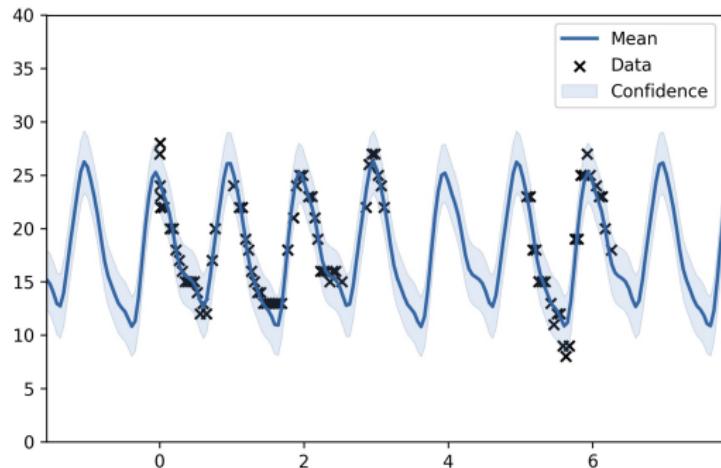
## Gaussian Processes cont.

- ▶ Gaussian process fit with radial basis function kernel
- ▶ Fit using GPy from SheffieldML  
([sheffieldml.github.io/GPy/](https://sheffieldml.github.io/GPy/)).



# Gaussian Processes cont.

- ▶ Gaussian process fit with periodic kernel



# Conclusion

- ▶ Data acquisition is an important step in data science
- ▶ LoRa is ideal for IoT applications requiring low power and long range
- ▶ Rapid prototyping is achievable for proof-of-concept
- ▶ Finding the ideal use cases is important

## Hands on work

- ▶ This repo describes the process of programming the Nucleo boards.

<https://github.com/ciiram/dsa-abuja-mbed-demo>

- ▶ This repo reproduces the analysis of the coffee data using Gaussian

processes.<https://github.com/ciiram/dsa-abuja-demo>

*Thank You*