

# Internet of Things Data Acquisition to Machine Learning.

*Ciira wa Maina*  
*[cwamaina.dekut@gmail.com](mailto:cwamaina.dekut@gmail.com)*

12th November, 2018



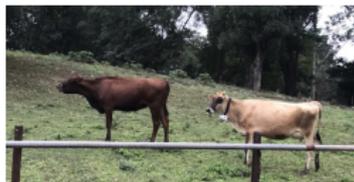
## Internet of Things

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



# Use Cases Closer Home

- ▶ Agriculture, Environmental Monitoring, ...

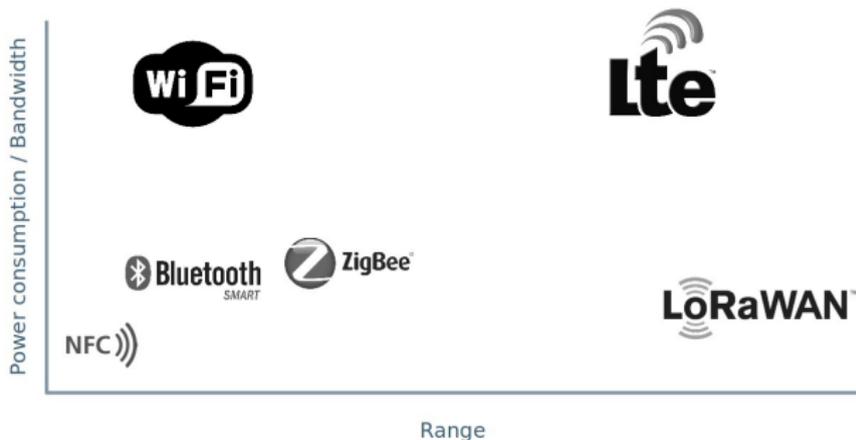




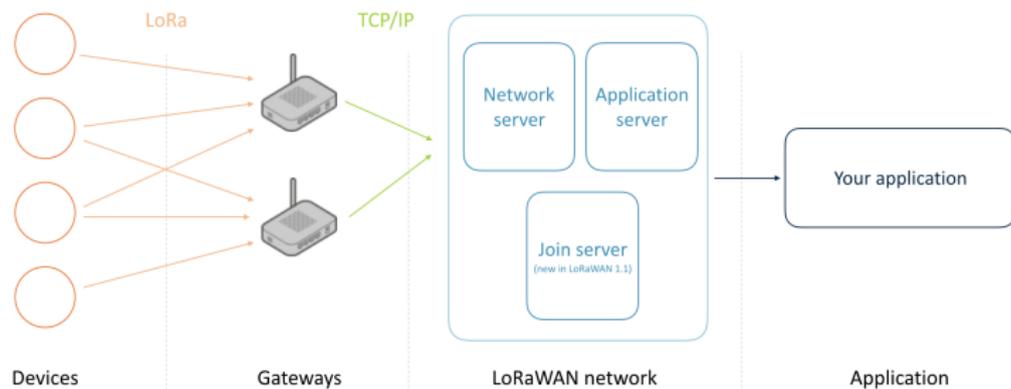


# LoRa

- ▶ 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

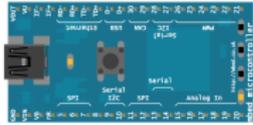
- ▶ Mbed OS implements
  - ▶ Radio drivers
  - ▶ Networking
  - ▶ Hardware drivers
  - ▶ Storage
- ▶ Allows rapid prototyping

# 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

```
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 }
13 }
```



**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
```

- ▶ Babbler: A device to monitor cargo on transit



# Built with Mbed

- ▶ A device to monitor tilting of power poles



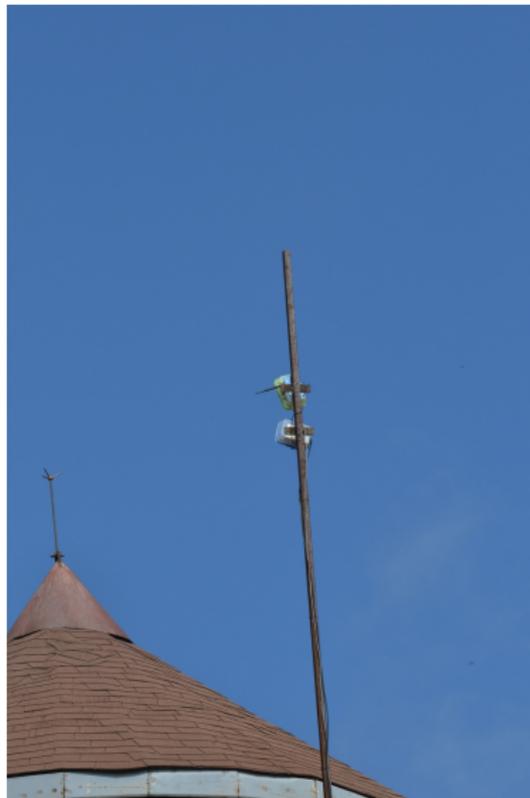
# Gateways



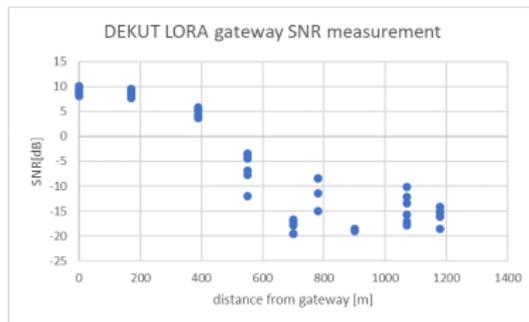
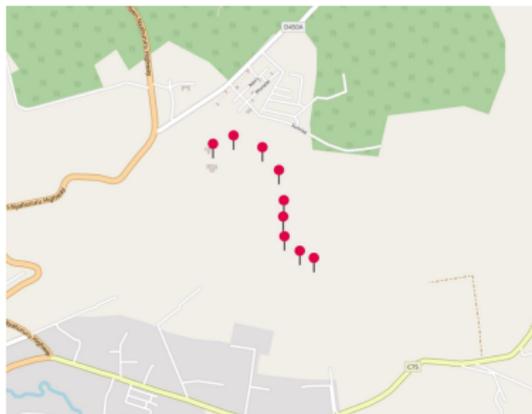




# Gateways

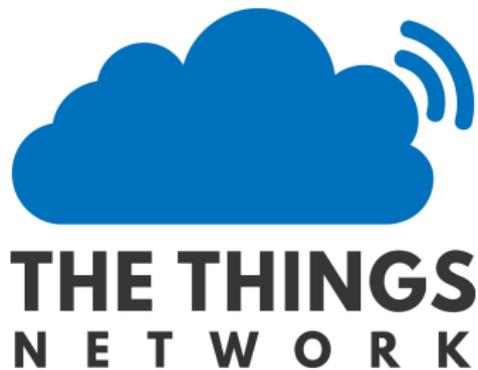


# Gateways



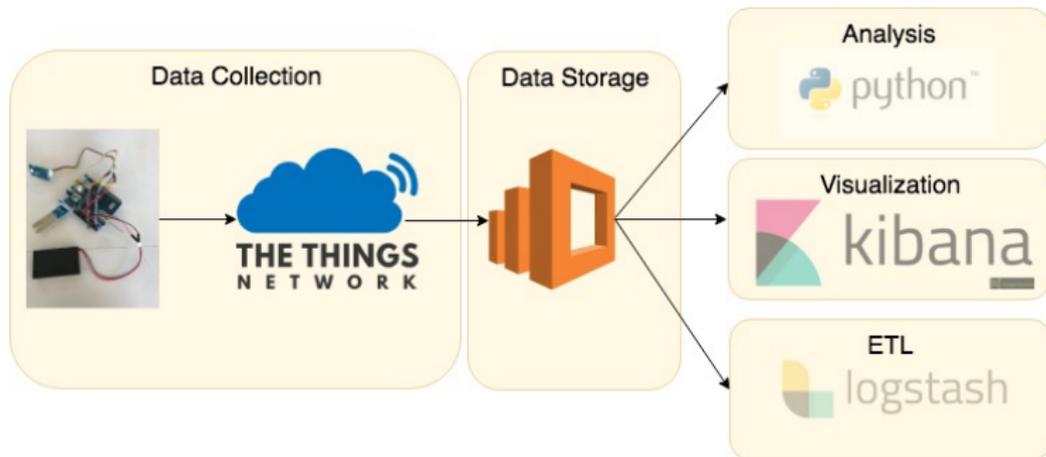
# Network Servers

- ▶ 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



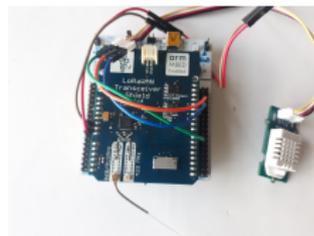
# Demo Application

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

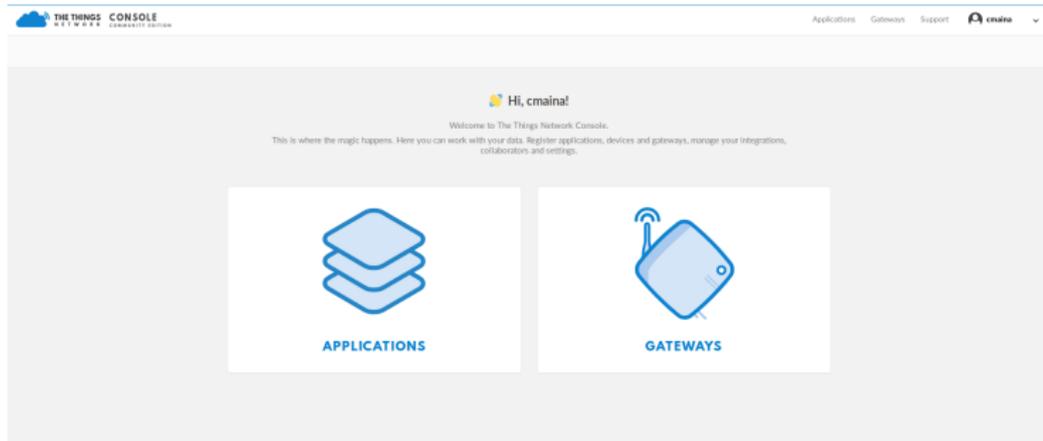


# Sense the Environment

```
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     }
```

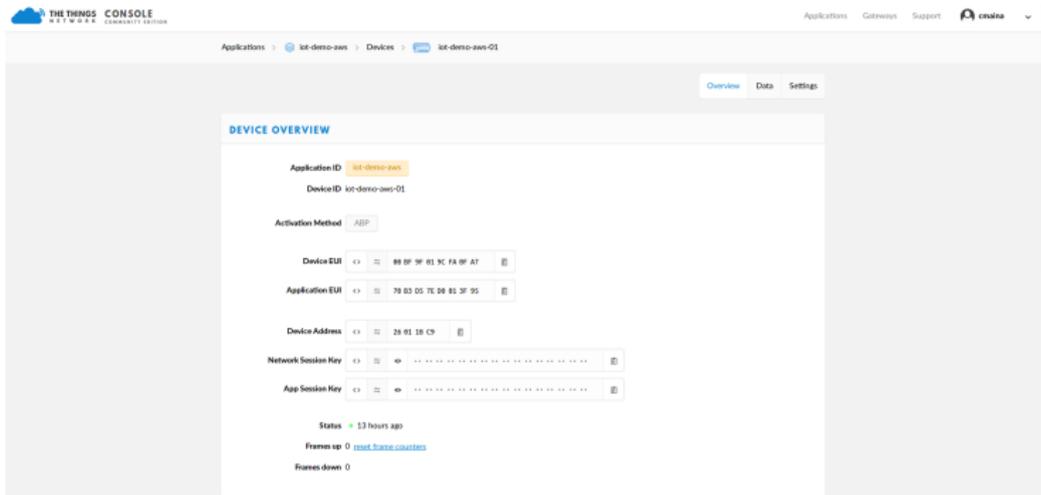


# Set Up Devices on The Things Network



The screenshot shows the 'THE THINGS NETWORK CONSOLE' interface. The top navigation bar includes 'Applications', 'Gateways', 'Support', and a user profile for 'cmains'. The main content area features a welcome message: 'Hi, cmains! 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.' Below this are two prominent buttons: 'APPLICATIONS' with a stack of three blue squares icon, and 'GATEWAYS' with a blue square icon representing a gateway device with an antenna and a small circle.

# Set Up Devices on The Things Network



The screenshot displays the 'DEVICE OVERVIEW' page in The Things Console. The breadcrumb navigation shows 'Applications > kit-demo-aws > Devices > kit-demo-aws-01'. The page has tabs for 'Overview', 'Data', and 'Settings'. The main content area shows the following details:

- Application ID:** kit-demo-aws
- Device ID:** kit-demo-aws-01
- Activation Method:** ABP
- Device EUI:** 88 BF 9F 81 9C FA 0F A7
- Application EUI:** 78 03 05 7E 08 81 3F 95
- Device Address:** 29 01 18 C9
- Network Session Key:** [Redacted]
- App Session Key:** [Redacted]
- Status:** 13 hours ago
- Frames up:** 0 [reset frame counters](#)
- Frames down:** 0

# Amazon Web Services Integration

The screenshot displays the AWS IoT console interface. At the top, the navigation bar includes the AWS logo, 'Services', and 'Resource Groups'. The user is logged in as 'Administrator @ 1633-2396-5...' in the 'London' region. The main content area shows the details for a specific IoT Thing named 'lot-demo-aws-01'.

**Thing Details:**

- Thing ARN:** `arn:aws:iot:eu-west-2:163325965848:thing/lot-demo-aws-01`
- Type:** `larawan`
- Attributes:**

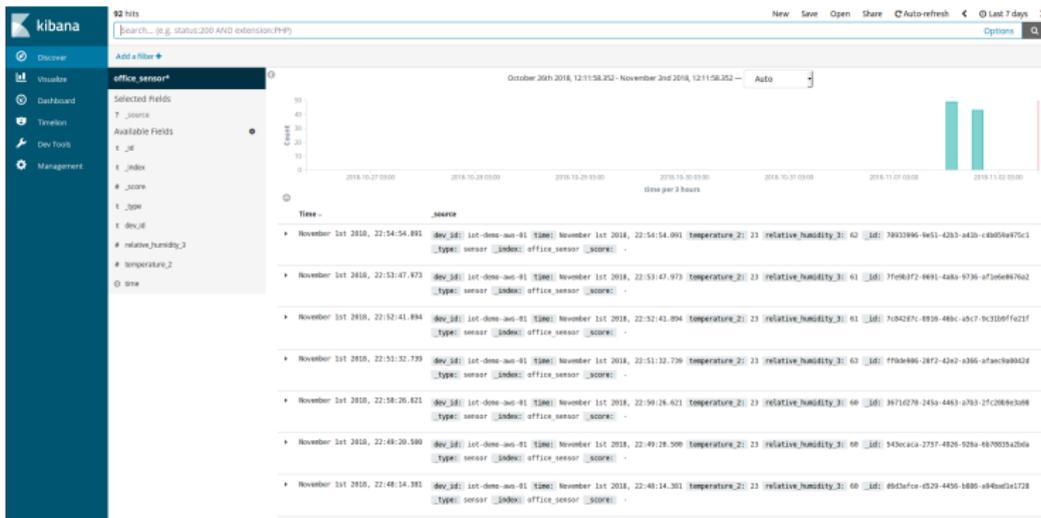
Attribute key	Value
app_id	lot-demo-aws
dev_id	00B5F019CFADKA7
dev_id	lot-demo-aws-01
app_id	7063D57ED0013F93

# Amazon Web Services Integration - Database

The screenshot displays the AWS IAM console interface for an IoT Demo domain. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information: 'Administrator @ 1635-2596-5...', 'London', and 'Support'. The left sidebar shows a 'Dashboard' with 'My domains' and 'Reserved instances'. The main content area is titled 'iot-demo' and features several action buttons: 'Configure cluster', 'Modify access policy', 'Manage tags', 'Delete domains', and 'Upgrade domain'. Below these buttons are tabs for 'Overview', 'Cluster health', 'Indices', 'Monitoring', 'Logs', and 'Upgrade History'. The 'Overview' tab is active, showing the following details:

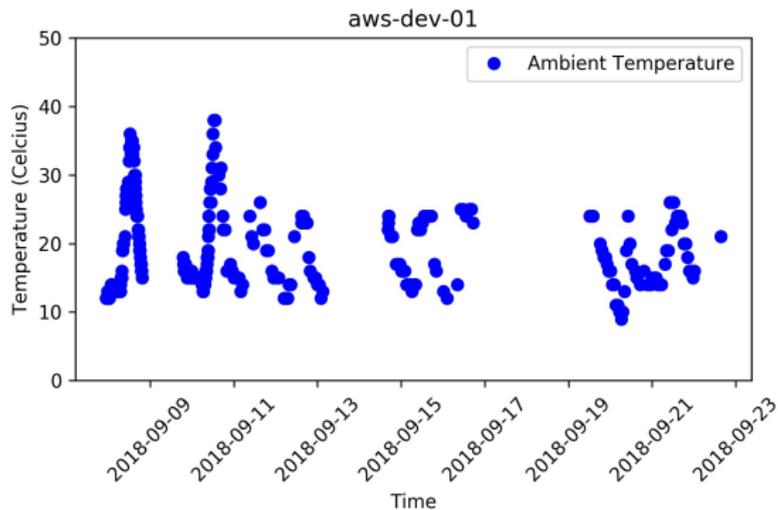
- Domain status:** Active
- Elasticsearch version:** 6.3
- Endpoint:** <https://search-iot-demo-gfrn4npwdfnsp3lauz7wsk0q.eu-west-2.es.amazonaws.com>
- Domain ARN:** [arn:aws:es:eu-west-2:163525965848:domain/iot-demo](#)
- Kibana:** [https://search-iot-demo-gfrn4npwdfnsp3lauz7wsk0q.eu-west-2.es.amazonaws.com/\\_plugin/kibana/](https://search-iot-demo-gfrn4npwdfnsp3lauz7wsk0q.eu-west-2.es.amazonaws.com/_plugin/kibana/)

# Amazon Web Services Integration - Database



# Act on Data

- ▶ Visualization
- ▶ Monitor variables and guide actions



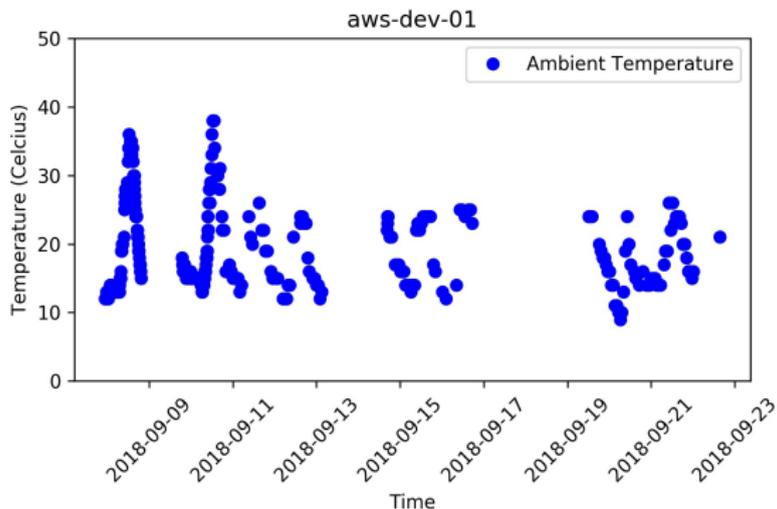
# Machine Learning Example

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



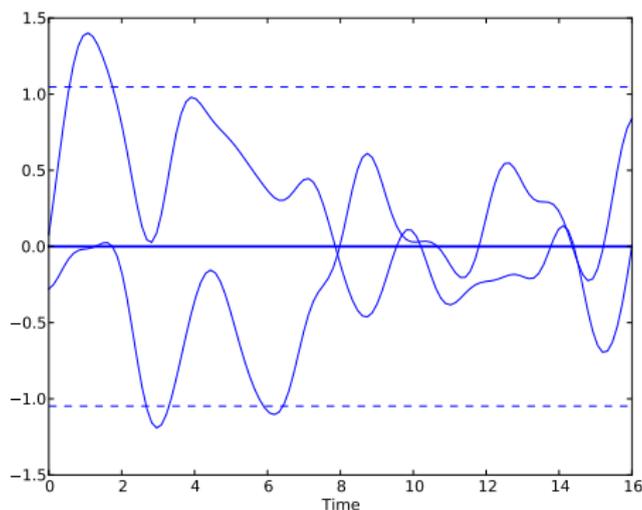
# Machine Learning Example

- ▶ Gaussian Process regression to fill missing values



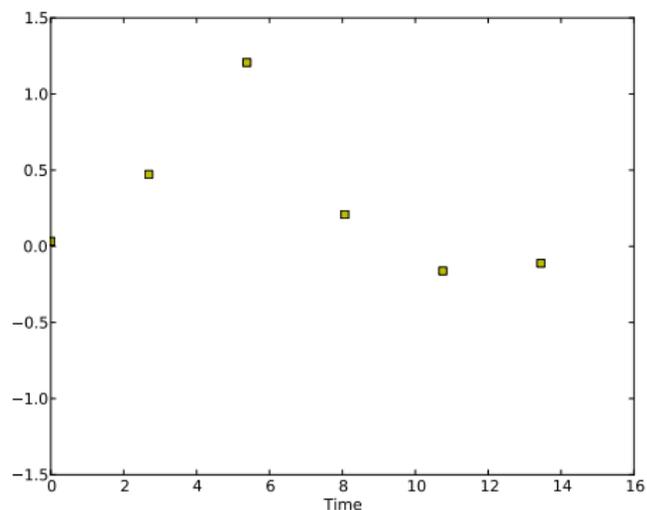
# Gaussian Processes

- ▶ 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  $k(t, t')$ .
- ▶  $f(t) \sim \mathcal{GP}(m(t), k(t, t'))$



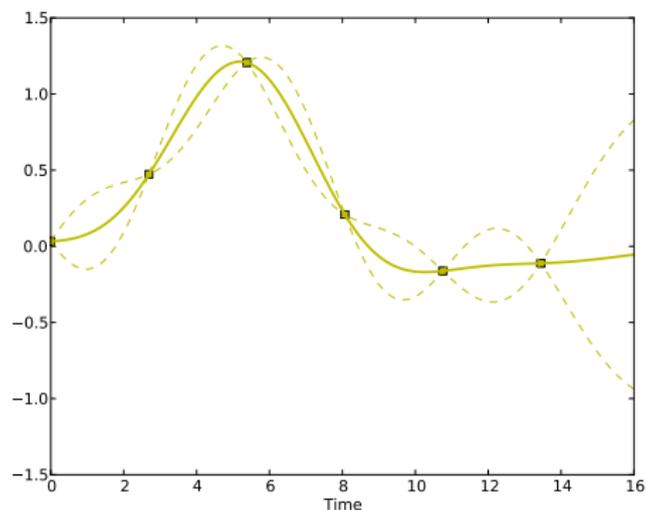
# Gaussian Processes cont

- ▶ Given observations we would like to infer an interpolant



## Gaussian Processes cont.

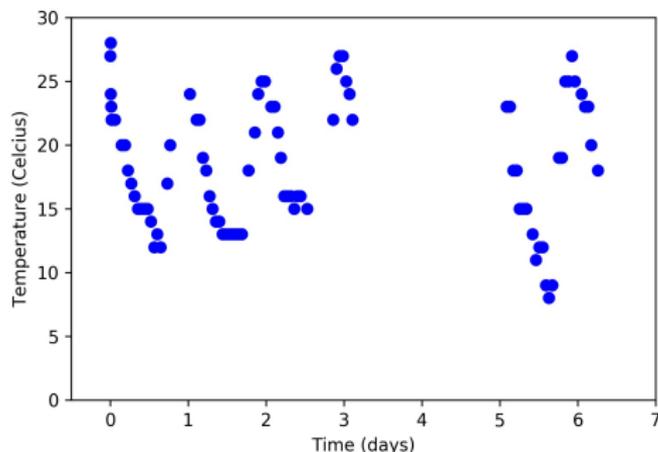
- ▶ Posterior distribution of functions given observations





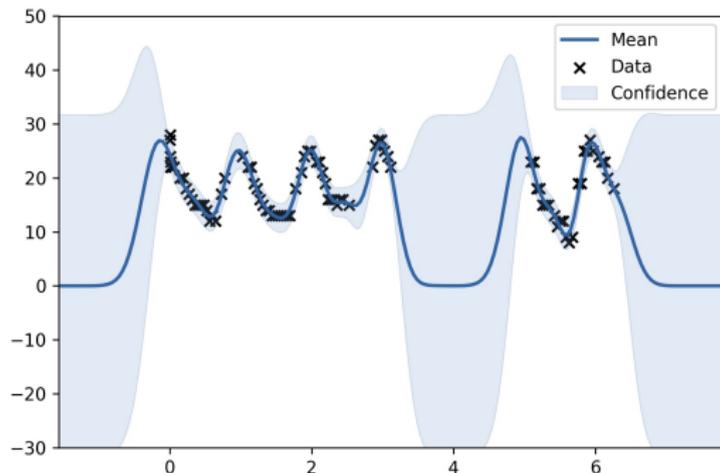
## Gaussian Processes cont.

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



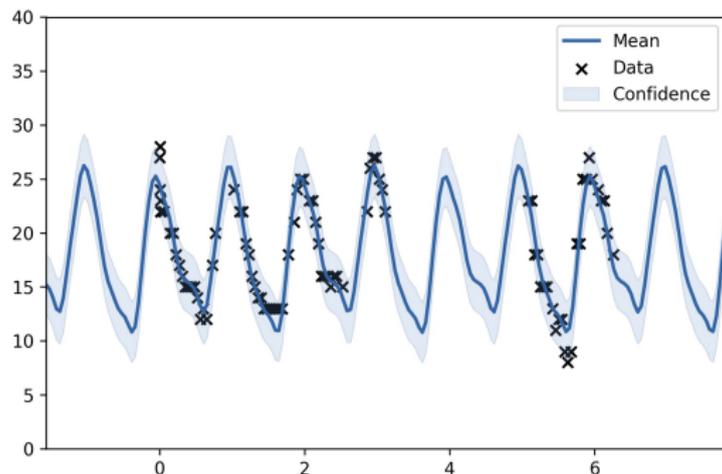
## Gaussian Processes cont.

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