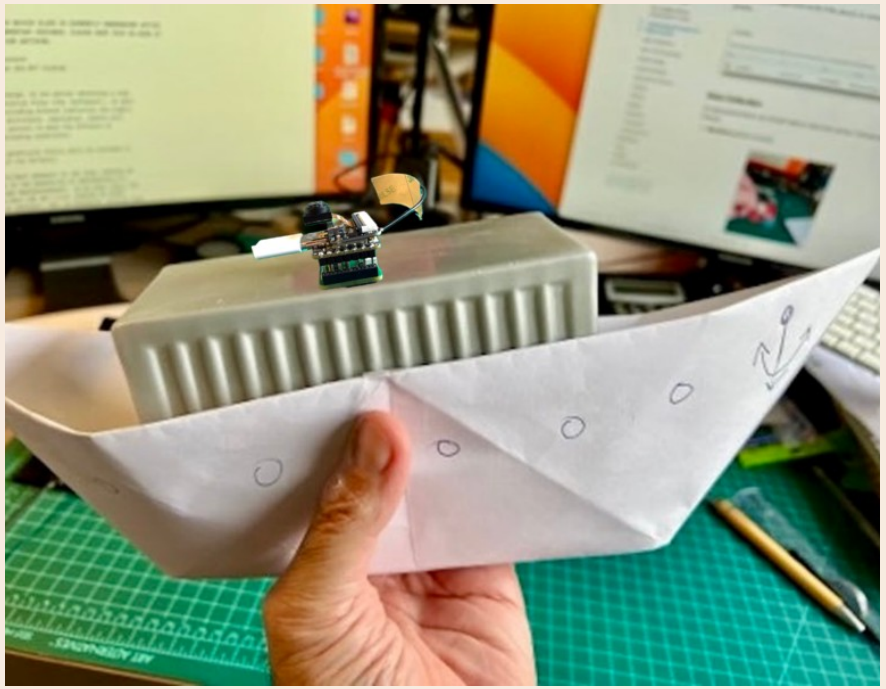
CHAPTER 5 · TINYML ON SEEED

Motion and Anomalies



An accelerometer, two seconds of motion, and a model that knows when it has never seen this before

IMU · SPECTRAL FEATURES · K-MEANS

52 slides · 1 lab · XIAOML Kit

AIEng4D

WHERE WE ARE · PART I

01

From sound to motion

Sixteen thousand numbers per second became 50. The signal is slow, so the hard part moves from compute to data.

LEARNING OBJECTIVES

What you will be able to do

01

Read the IMU

The LSM6DS3TR-C on the kit, its axes, and acceleration in m/s² at 50 Hz.

02

Collect motion data

Stream the accelerometer to Edge Impulse Studio with the CLI data forwarder.

03

Turn motion into features

Two seconds of data become 63 spectral features, for a small dense network.

04

Detect what is new

Add K-means anomaly detection, deploy both models, and act on the result.

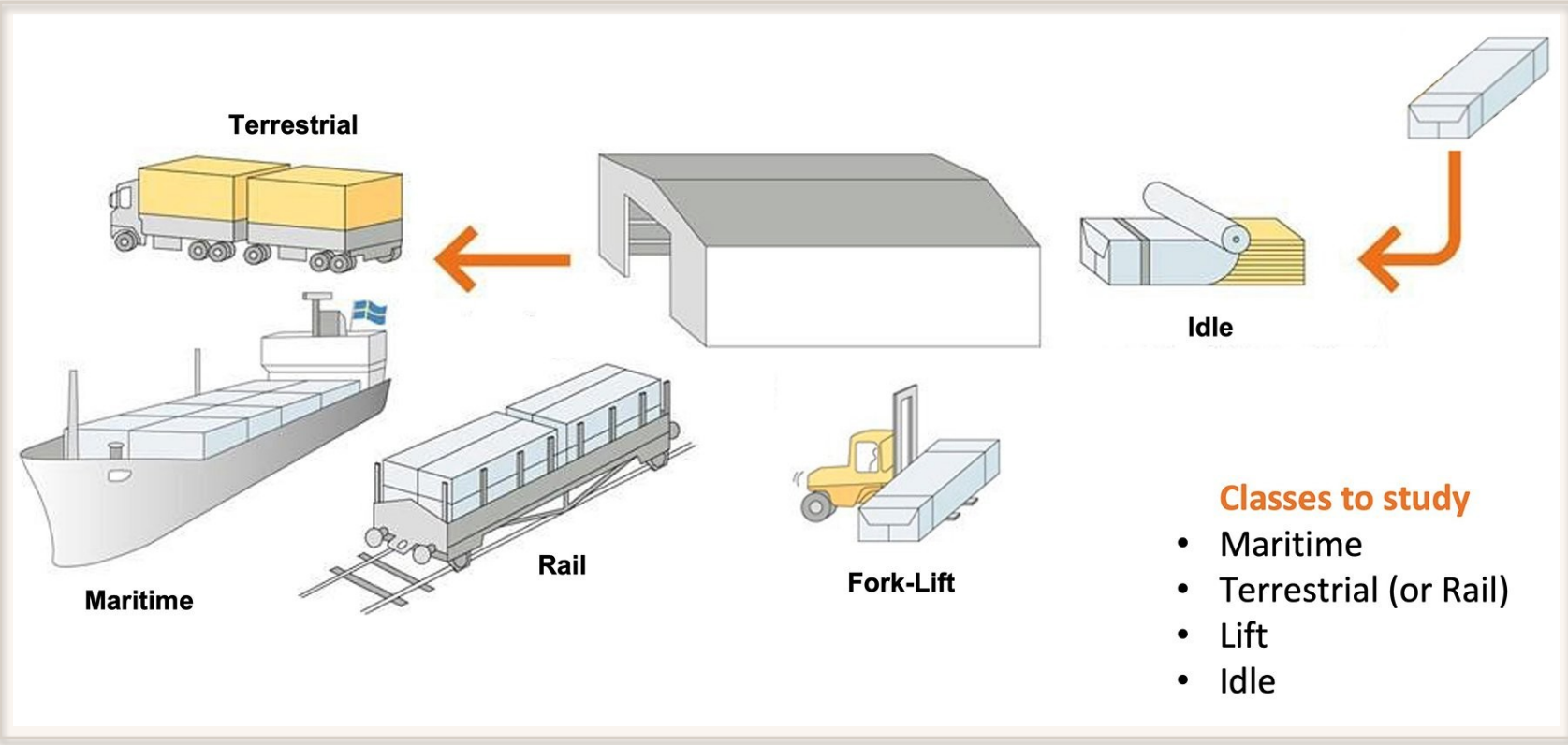
THE PROBLEM

Millions of containers move every day by ship, truck, and train. What happens to them on the way?

A sensor inside the package can tell the phase of the trip, and flag the moment something goes wrong.

THE PHASES OF A TRIP

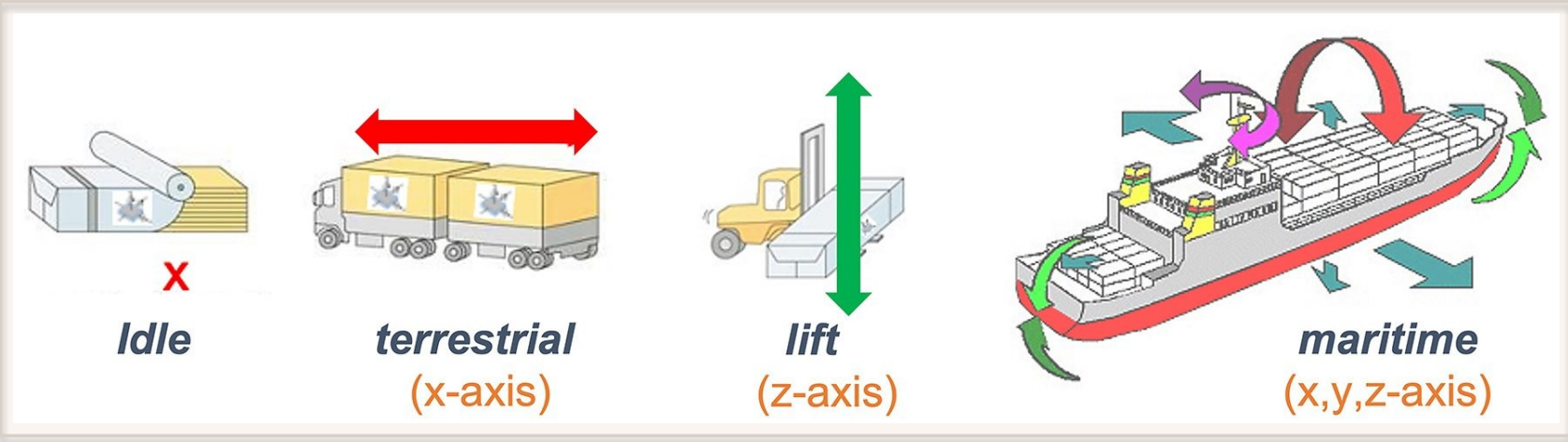
Four classes to study



1. Maritime — pallets on a boat
2. Terrestrial — pallets on a truck or a train
3. Lift — pallets moved by a forklift
4. Idle — pallets stored in a warehouse

WHAT EACH CLASS LOOKS LIKE

Each phase moves along different axes



1. Idle — almost no movement
2. Terrestrial — mostly horizontal, back and forth
3. Lift — mostly vertical, up and down
4. Maritime — waves on all three axes

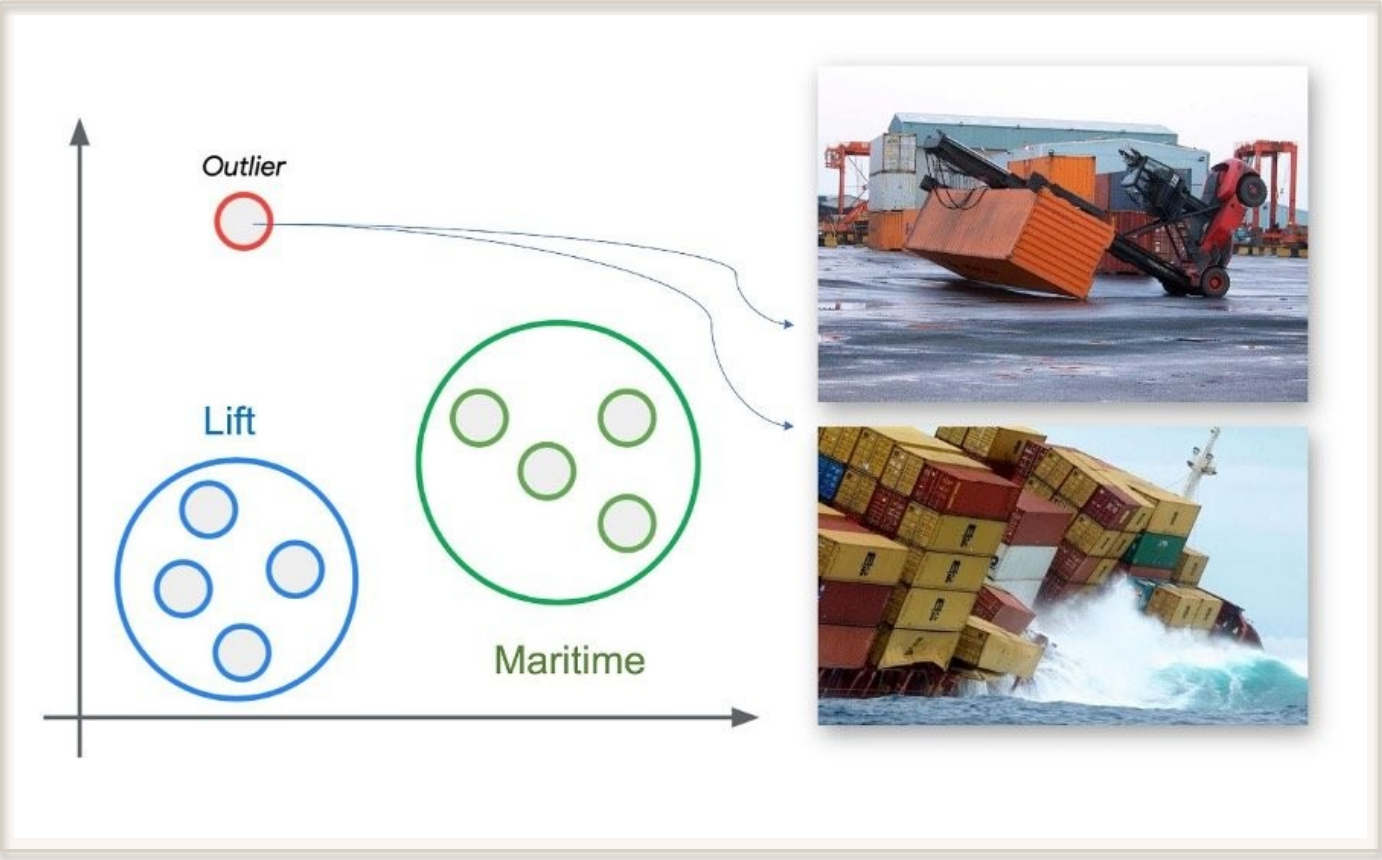
TWO QUESTIONS, TWO MODELS

Which of the motions I know is this? And — is this like anything I have ever seen?

A classifier answers the first. Only an anomaly detector can answer the second.

ANOMALY DETECTION

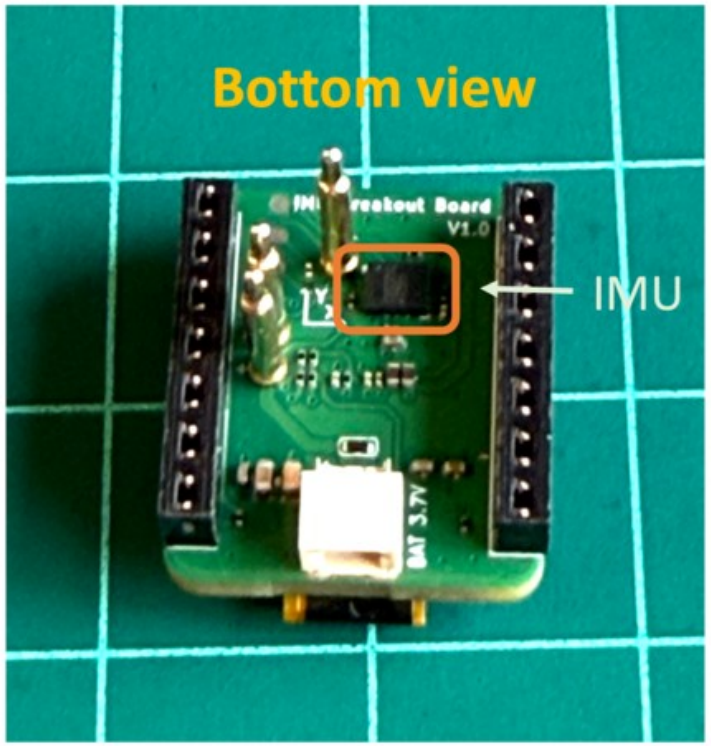
Known motions form clusters; an anomaly falls outside

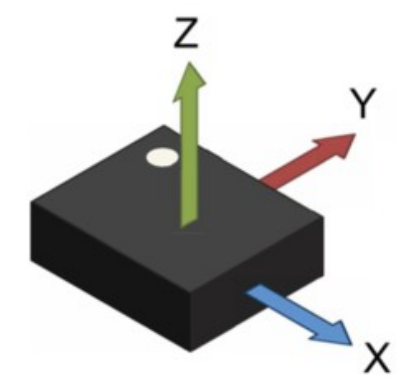


1. Each known class forms a cluster of points in feature space
2. A new sample far from every cluster is an outlier
3. A container falling off a ship, or one lying upside down

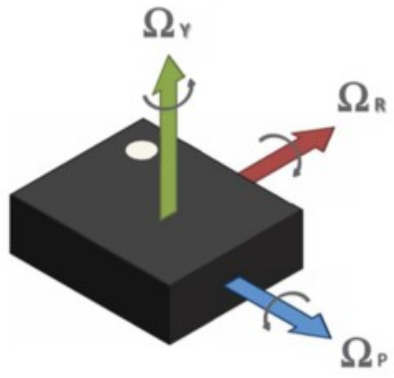
THE SENSOR

A 6-axis IMU, already on the expansion board





Direction of detectable acceleration (top view)



Direction of detectable angular rate (top view)

1. LSM6DS3TR-C — 3-axis accelerometer and 3-axis gyroscope
2. On the bottom of the expansion board, on I2C at address 0x6A
3. X to the right, Y forward, Z up out of the board
4. This lab uses only the accelerometer

LSM6DS3TR-C

What the IMU gives us

01

Accelerometer

±2, ±4, ±8, or ±16 g. We use ±2 g, the default.

02

Gyroscope

±125 up to ±2000 degrees per second. Default ±250 dps.

03

16-bit, I2C

A 16-bit converter, read over I2C at address 0x6A.

04

The library

Seeed Arduino LSM6DS3 by Seeed Studio. Not the Arduino_LSM6DS3 library, which is for other boards.

FIRST TEST

Read six values, ten times per second

XIAOML Kit IMU Test (excerpt)

```
#include <LSM6DS3.h>
#include <Wire.h>

LSM6DS3 myIMU(I2C_MODE, 0x6A);    // IMU on I2C

void setup() {
  Serial.begin(115200);
  if (myIMU.begin() != 0) {
    Serial.println("ERROR: IMU initialization failed!");
  }
}

void loop() {
  float ax = myIMU.readFloatAccelX();   // g
  float ay = myIMU.readFloatAccelY();
  float az = myIMU.readFloatAccelZ();
  float gx = myIMU.readFloatGyroX();    // degrees/s
  // ... print the six values
  delay(100);                           // 10 Hz
}
```

— Flat on the table, Z reads about +1.0 g — that is gravity.

— X and Y stay near 0 g, and the gyroscope near 0 °/s.

— Move the kit, and watch the values change.

02

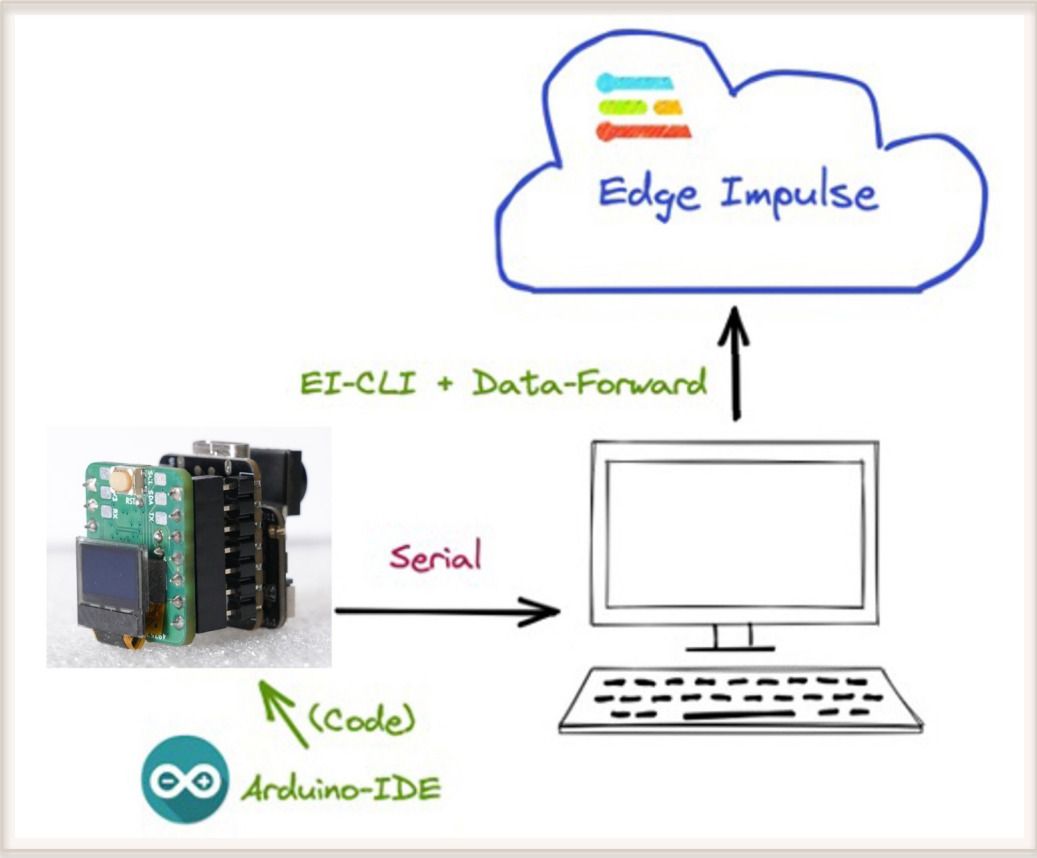
MOTION DATA

Motion becomes numbers

Stream the accelerometer to Edge Impulse Studio, and move the kit the way a package would move.

THE DATA FORWARDER

Serial to the computer, and the CLI sends it to the Studio



1. The Arduino sketch prints the accelerometer values on the serial port
2. The Edge Impulse CLI data forwarder reads the serial port
3. It sends the samples to your project in the Studio

DATA COLLECTION SKETCH

50 Hz, three axes, in m/s², separated by tabs

motion data collection (excerpt)

```
#define FREQUENCY_HZ   50
#define INTERVAL_MS    (1000 / (FREQUENCY_HZ + 1))

void loop() {
  if (millis() > last_interval_ms + INTERVAL_MS) {
    last_interval_ms = millis();

    float ax = myIMU.readFloatAccelX();
    float ay = myIMU.readFloatAccelY();
    float az = myIMU.readFloatAccelZ();

    Serial.print(ax * 9.81);  Serial.print("\t");
    Serial.print(ay * 9.81);  Serial.print("\t");
    Serial.println(az * 9.81);       // m/s²
  }
}
```

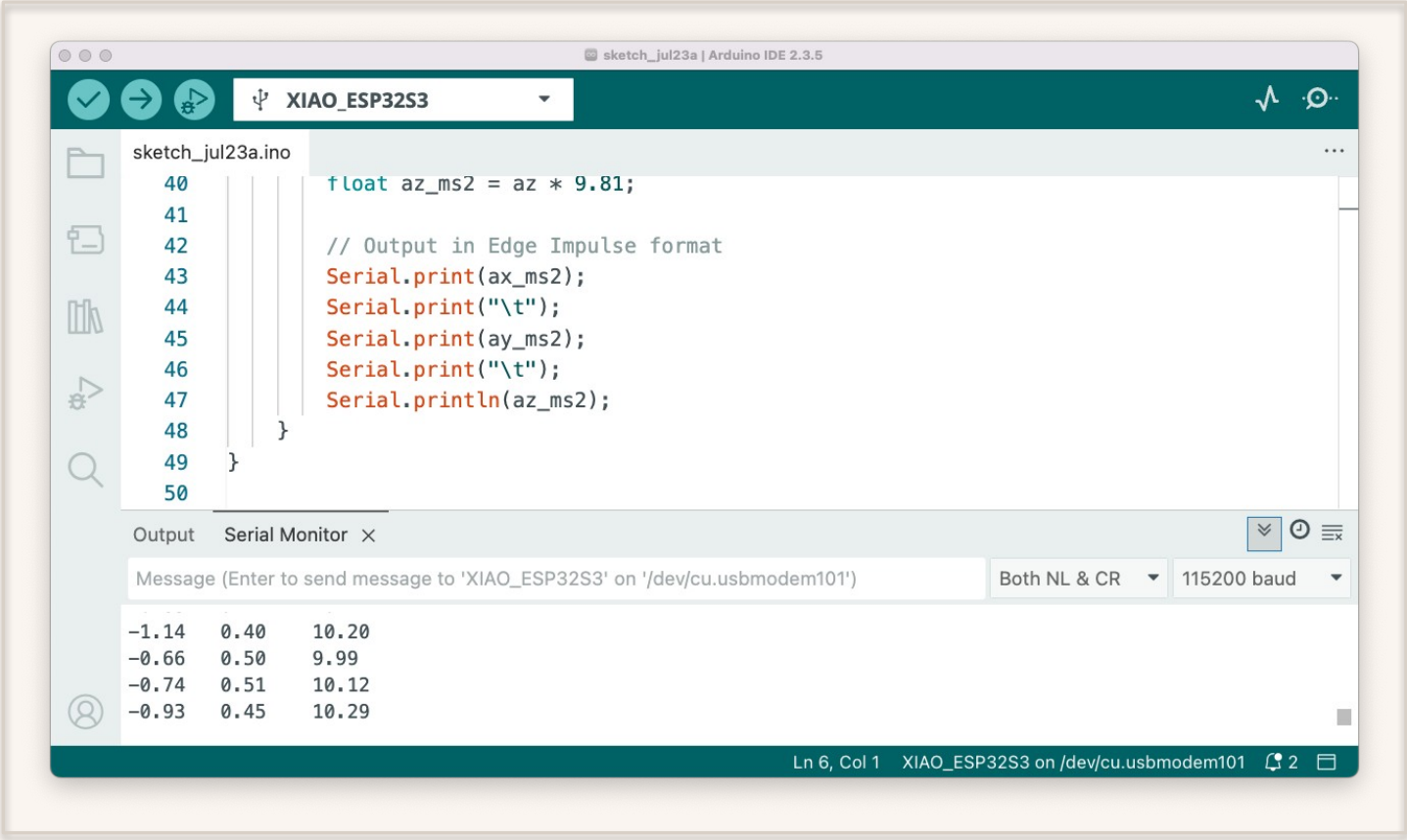
— One line per sample — the forwarder counts the lines to find the frequency.

— g × 9.81 gives m/s², so gravity reads about 9.8 on Z.

— Keep the sketch running, and close the Serial Monitor.

SERIAL MONITOR

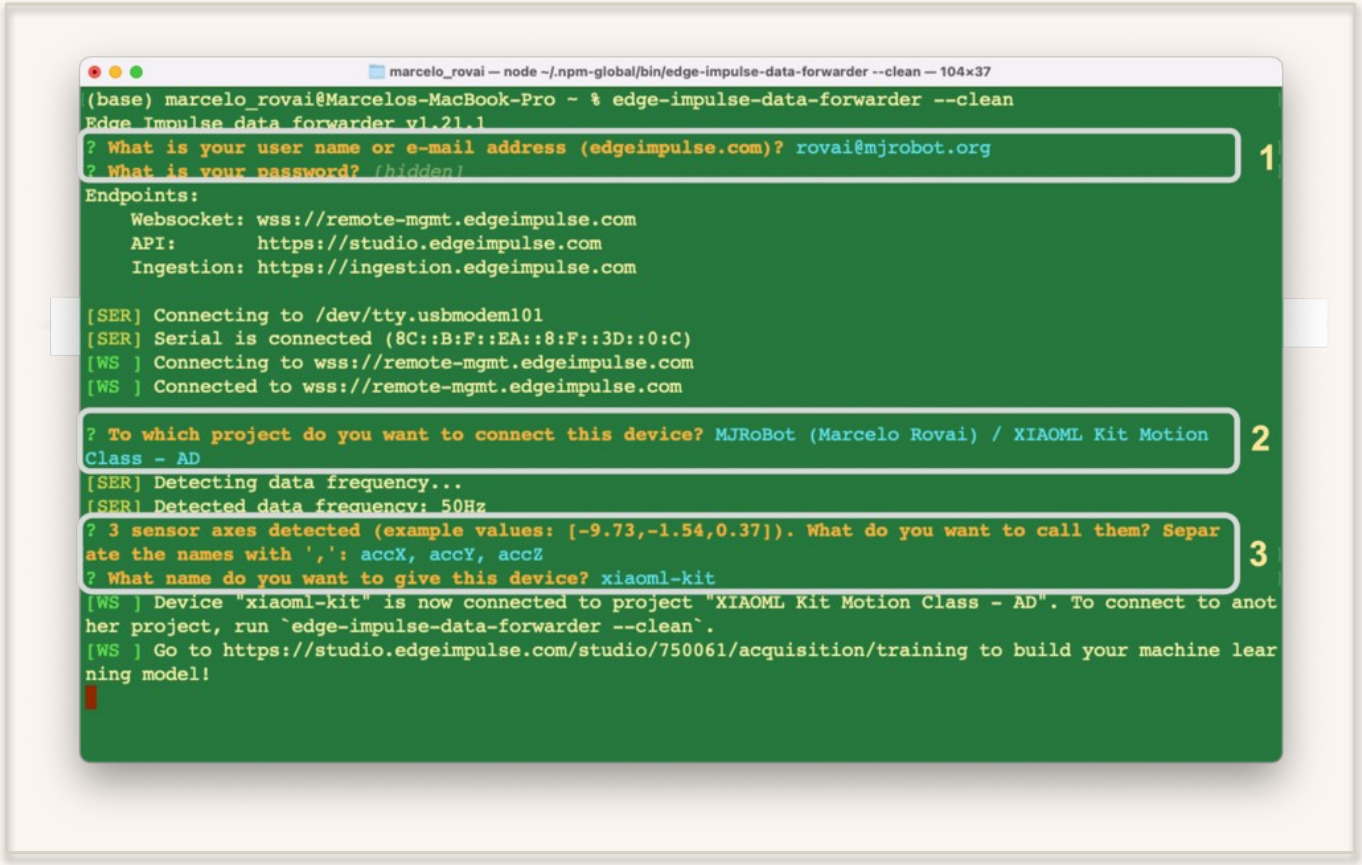
Three columns, and gravity on Z



1. accX, accY, accZ in m/s², one line per sample
2. Z near 9.8 m/s² with the kit flat — gravity
3. Then turn the Serial Monitor off, or the forwarder cannot open the port

EDGE-IMPULSE-DATA-FORWARDER --CLEAN

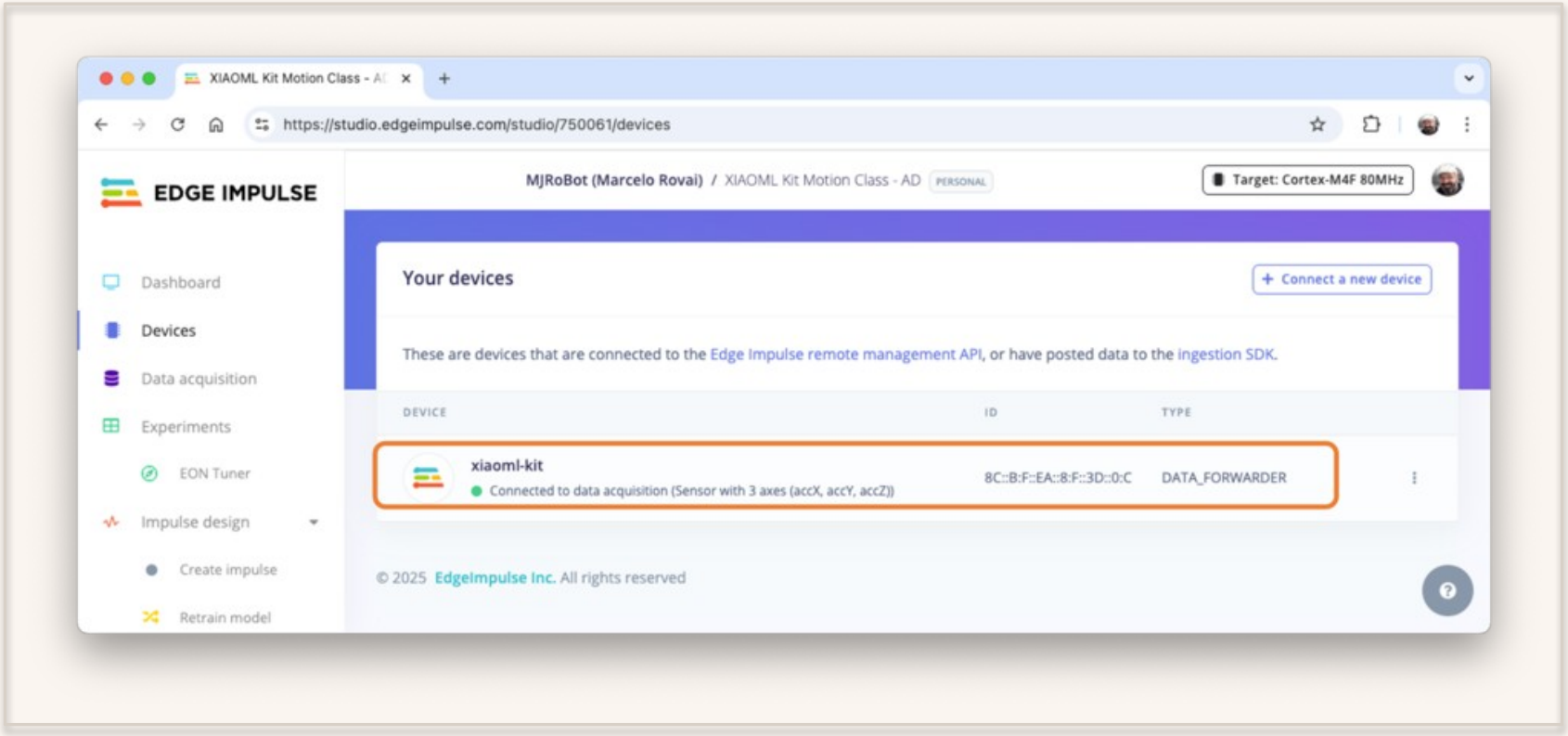
Log in, choose the project, and name the axes



1. Your Edge Impulse e-mail and password
2. The project to connect to
3. The forwarder detects 3 axes at 50 Hz — name them accX, accY, accZ
4. Give the device a name, for example xiaoml-kit

DEVICES

A green dot means the kit is connected



1. Devices tab of the Studio project
2. xiaoml-kit, connected through the DATA_FORWARDER
3. Sensor with 3 axes: accX, accY, accZ

MOVEMENT SIMULATION

Your hand plays the ship, the truck, and the forklift

01

Maritime

Wave-like, undulating motion on all three axes, with gentle rolling and pitching.

02

Terrestrial

Horizontal, in straight lines, left to right and back, with small shakes and bumps.

03

Lift

Mostly vertical — up, pause, down — with a little horizontal positioning.

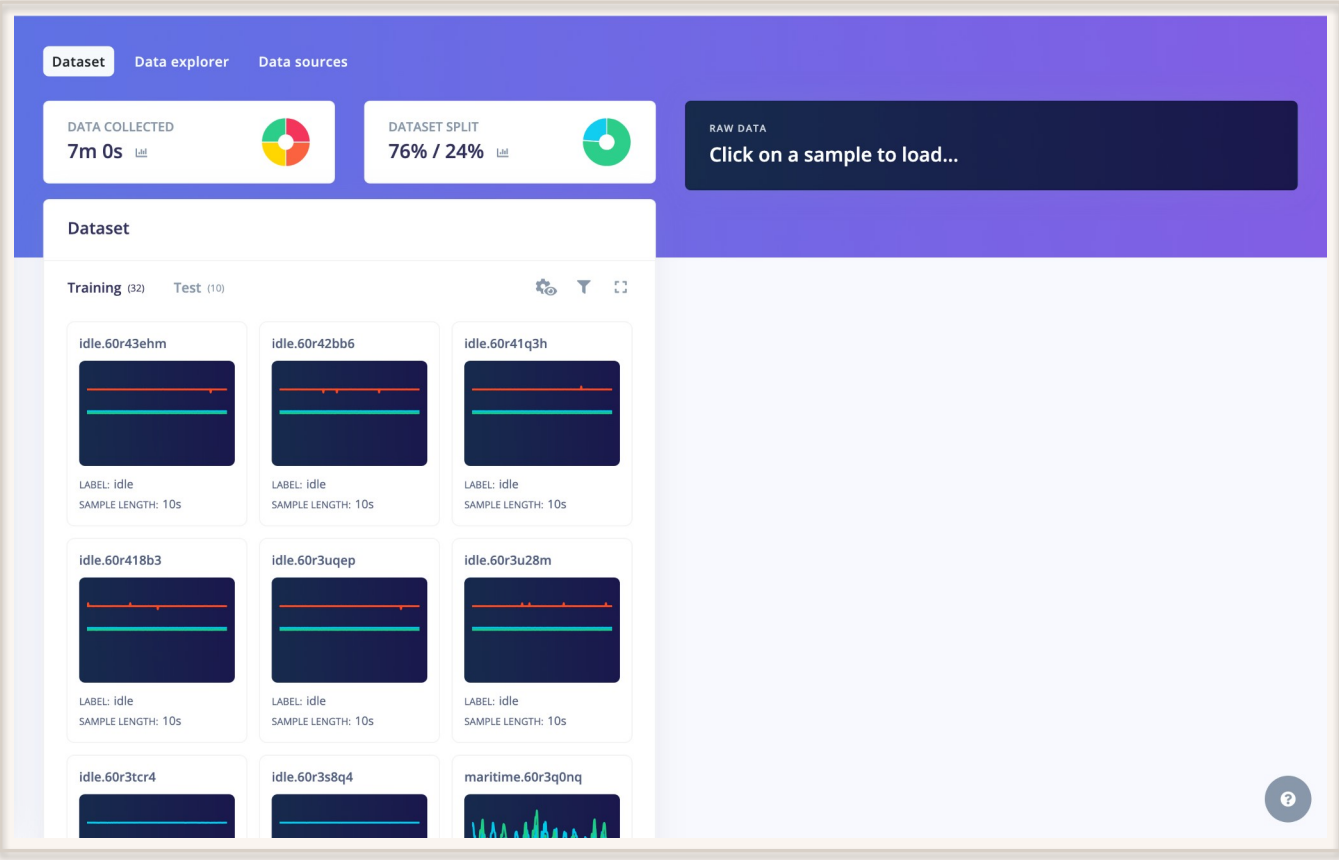
04

Idle

On a stable surface. Minimal movement, only room vibrations and sensor noise.

DATA ACQUISITION

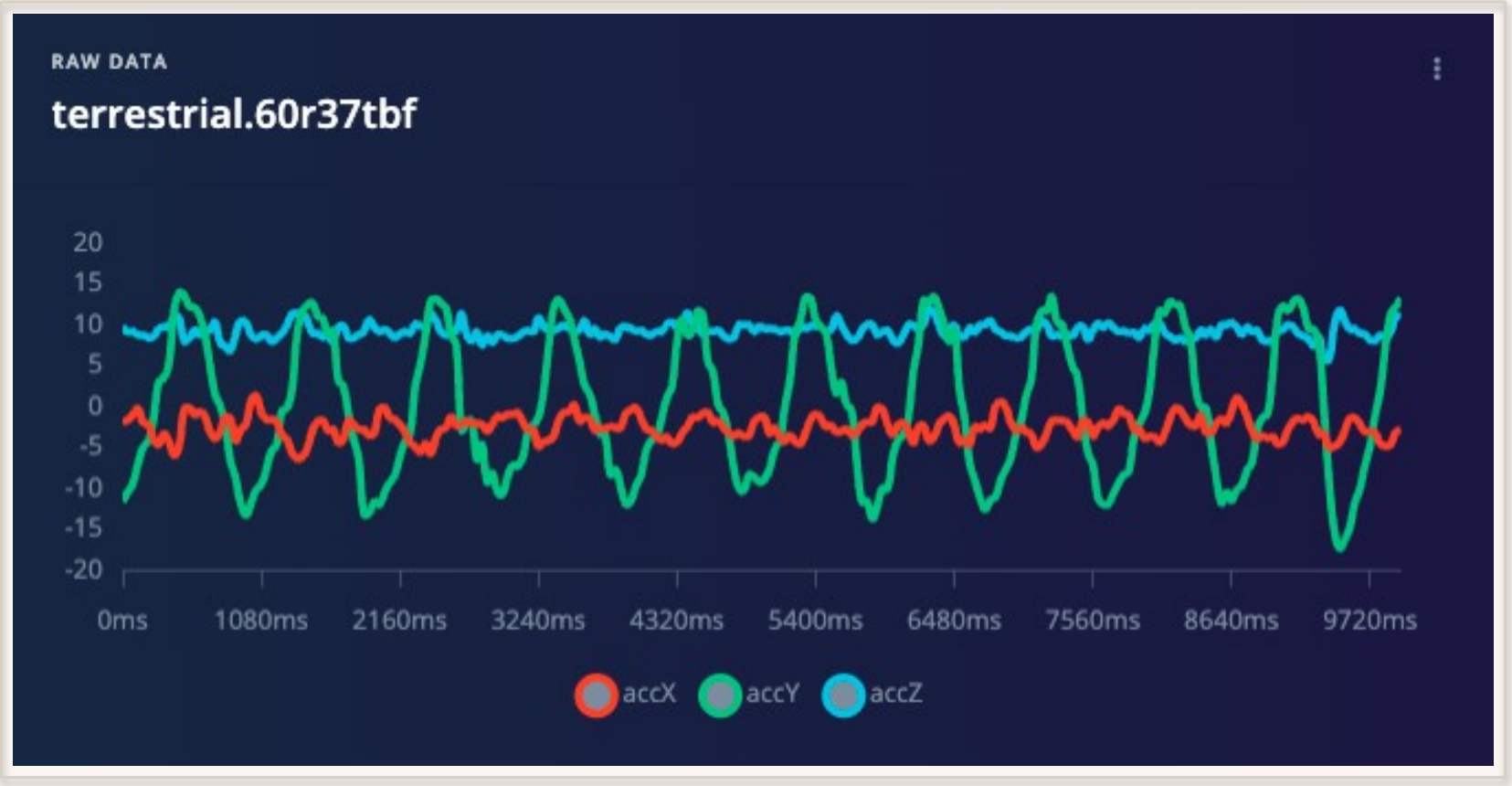
Ten-second samples, about two minutes per class



1. Device xiaoml-kit, 3 axes, 50 Hz, sample length 10,000 ms
2. Type the label, press Start sampling, and move the kit
3. 7 min of data in this project — 32 training and 10 test samples
4. Move two samples per class to the test set, or use the automatic split

ONE TERRESTRIAL SAMPLE

A wave on Y, and gravity on Z



1. accY oscillates — the left-right movement
2. accX stays around zero
3. accZ stays around 9.8 m/s², because of gravity

03

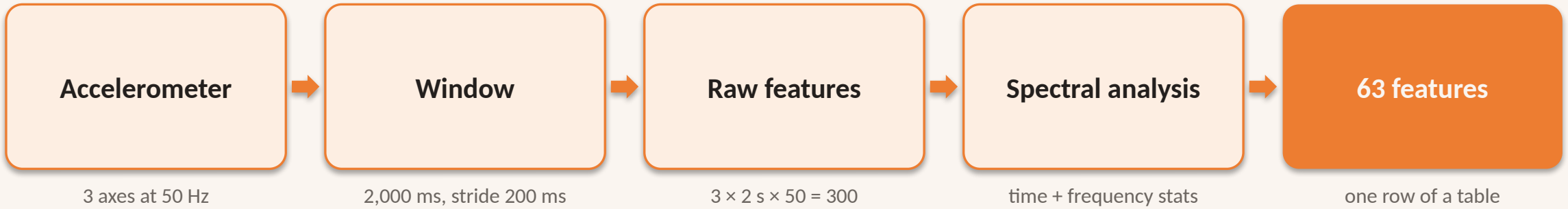
EDGE IMPULSE STUDIO

Two seconds become 63 numbers

A sliding window, spectral features, a small dense network, and K-means on the side.

THE IMPULSE

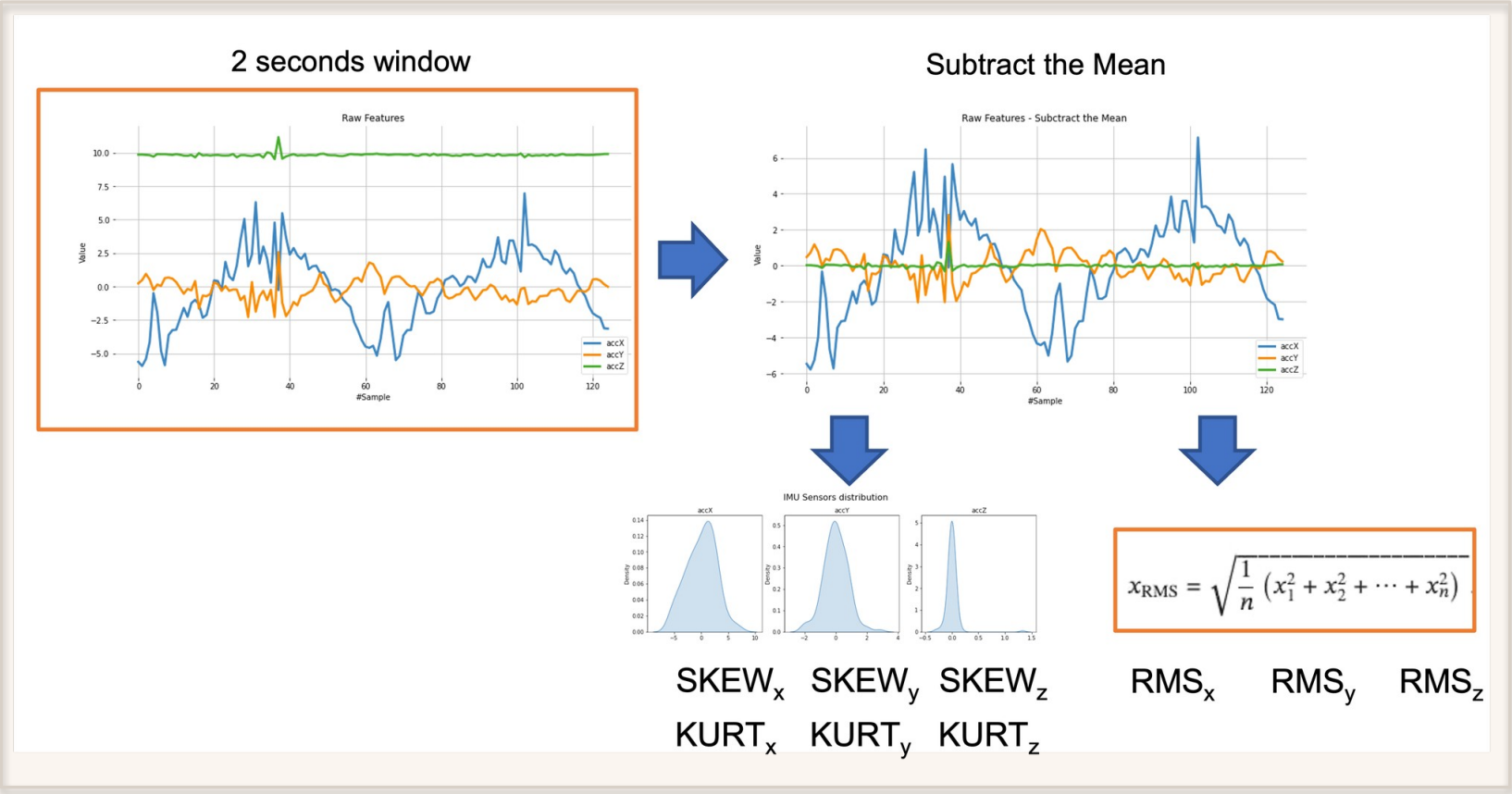
From a time series to tabular data



The same 63 features feed two models — the classifier and K-means.

TIME DOMAIN

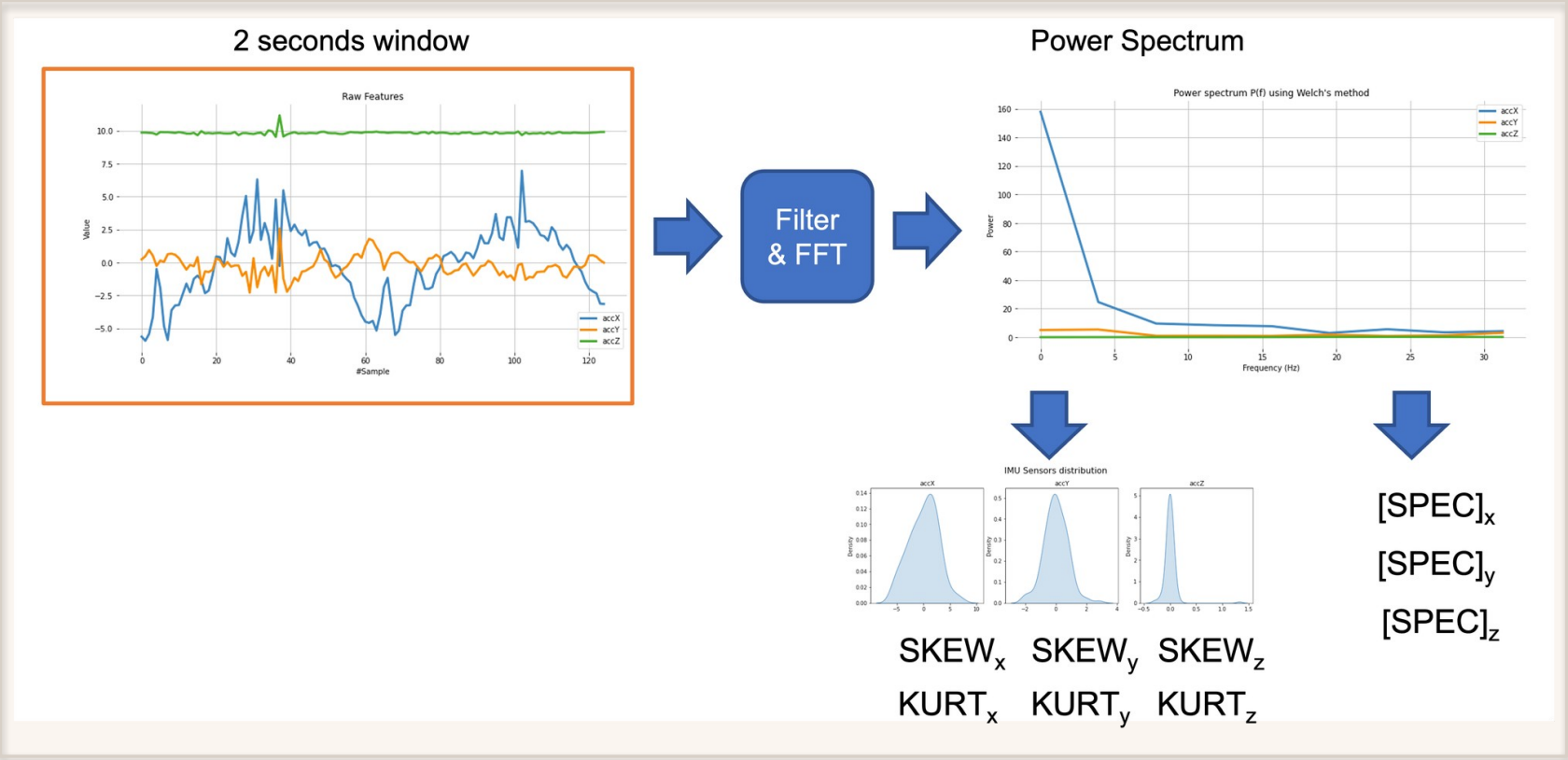
Per axis — RMS, skewness, and kurtosis



1. Take the 2-second window and subtract the mean
2. RMS — how strong the motion is
3. Skewness and kurtosis — the shape of the distribution of values

FREQUENCY DOMAIN

Per axis — the power spectrum, its skewness and kurtosis



1. Filter, then an FFT of each axis
2. Spectral power in frequency bands — how fast the motion repeats
3. Skewness and kurtosis of the spectrum

COUNTING THE FEATURES

21 features per axis, 63 in total

3 time statistics

RMS, skewness, and kurtosis of the window.

16 spectral bands

A 32-point FFT at 50 Hz gives 16 bands of about 1.56 Hz, from 0.78 to 25 Hz.

2 spectral statistics

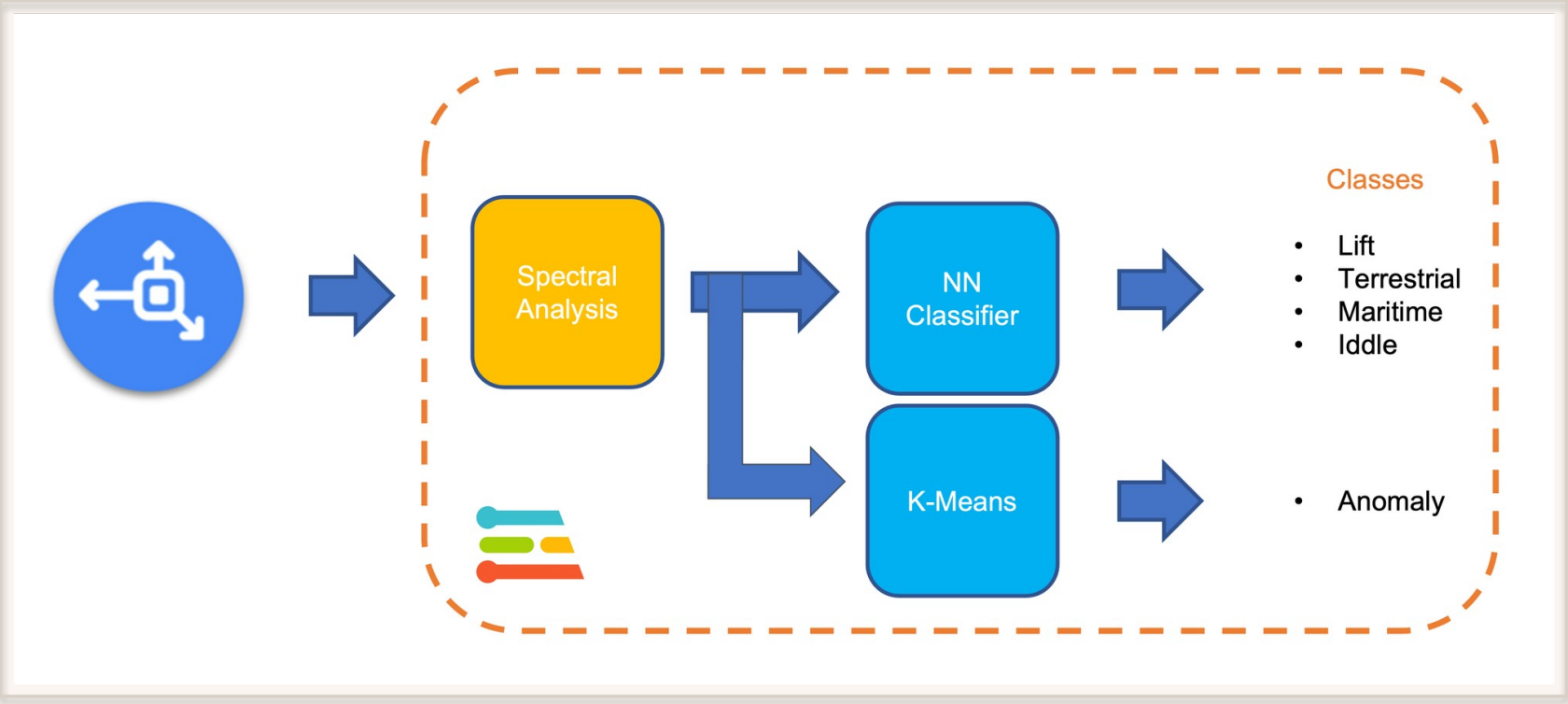
Skewness and kurtosis of the spectrum.

× 3 axes = 63

300 raw values in, 63 features out — about five times fewer.

TWO LEARNING BLOCKS

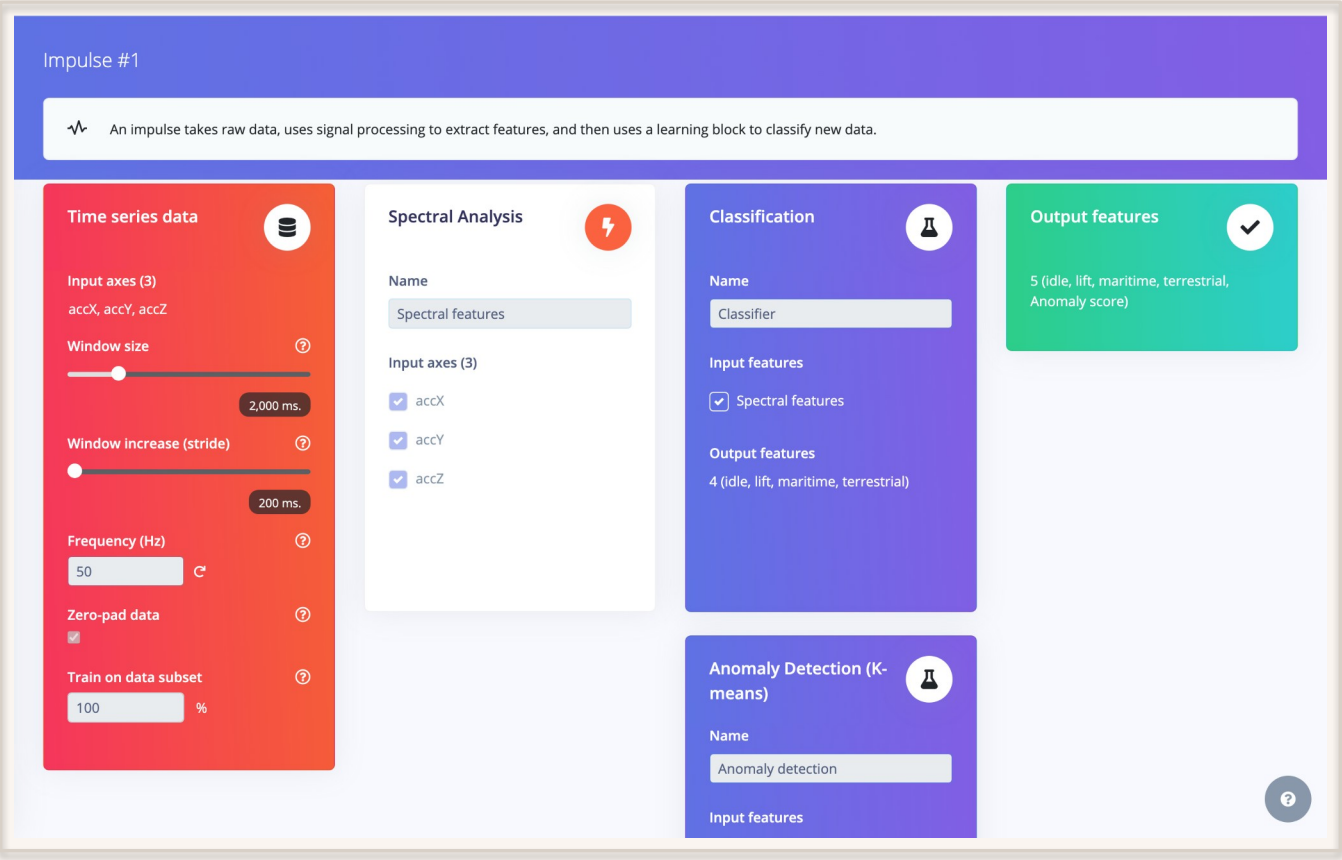
One set of features, a classifier and an anomaly detector



1. Spectral analysis produces the features
2. The NN classifier gives lift, terrestrial, maritime, or idle
3. K-means gives an anomaly score

CREATE IMPULSE

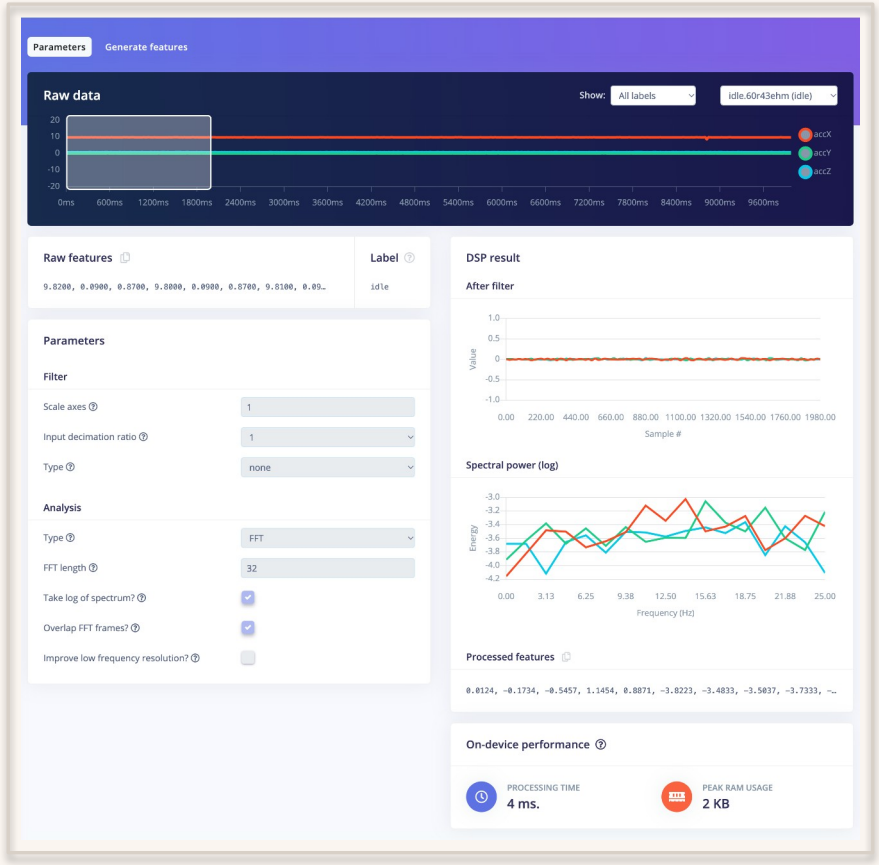
Time series, Spectral Analysis, Classification, and K-means



1. Window 2,000 ms, stride 200 ms, 50 Hz, zero-pad on
2. Spectral Analysis on accX, accY, accZ
3. Classification — 4 classes, in alphabetical order
4. Anomaly Detection (K-means) — the 5th output: the anomaly score

SPECTRAL FEATURES

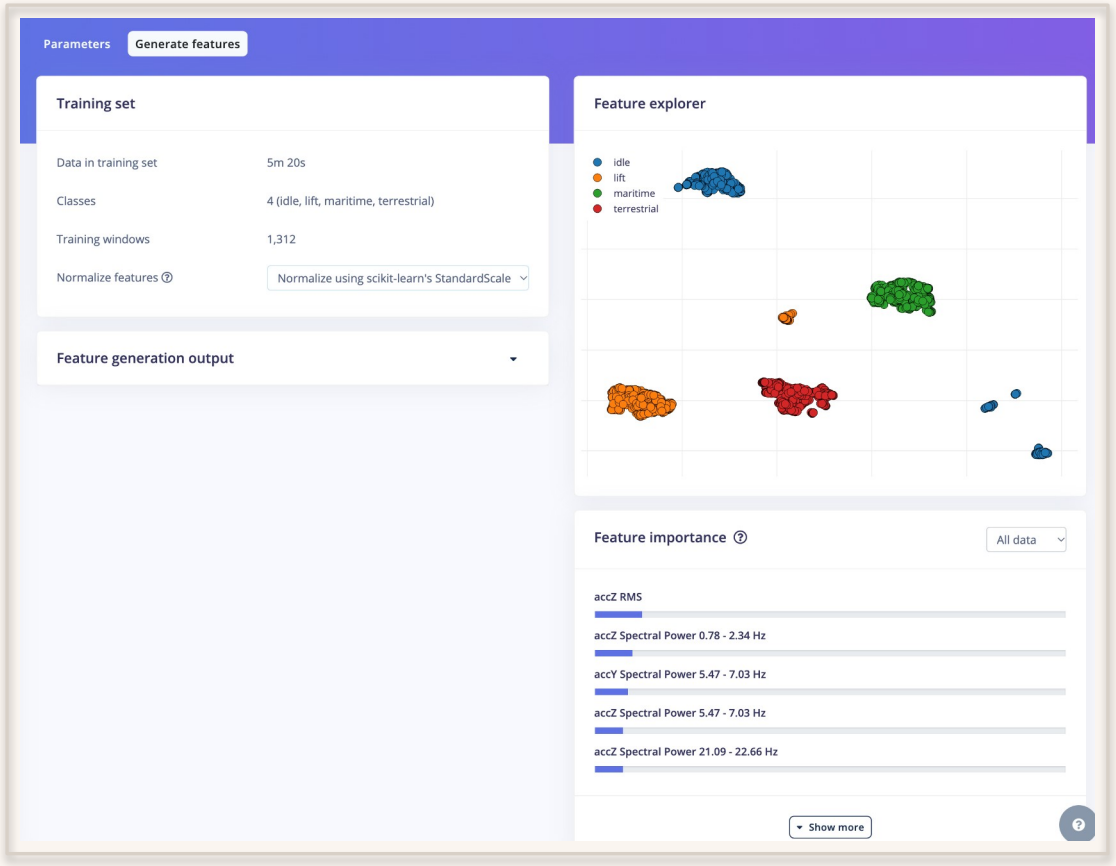
FFT of 32 points, and 4 ms on the device



1. The raw window, highlighted on the 10-second sample
2. Filter none, FFT length 32, log of the spectrum, overlapping frames
3. DSP result — the filtered signal and its spectral power
4. On-device estimate: 4 ms and 2 KB of RAM

GENERATE FEATURES

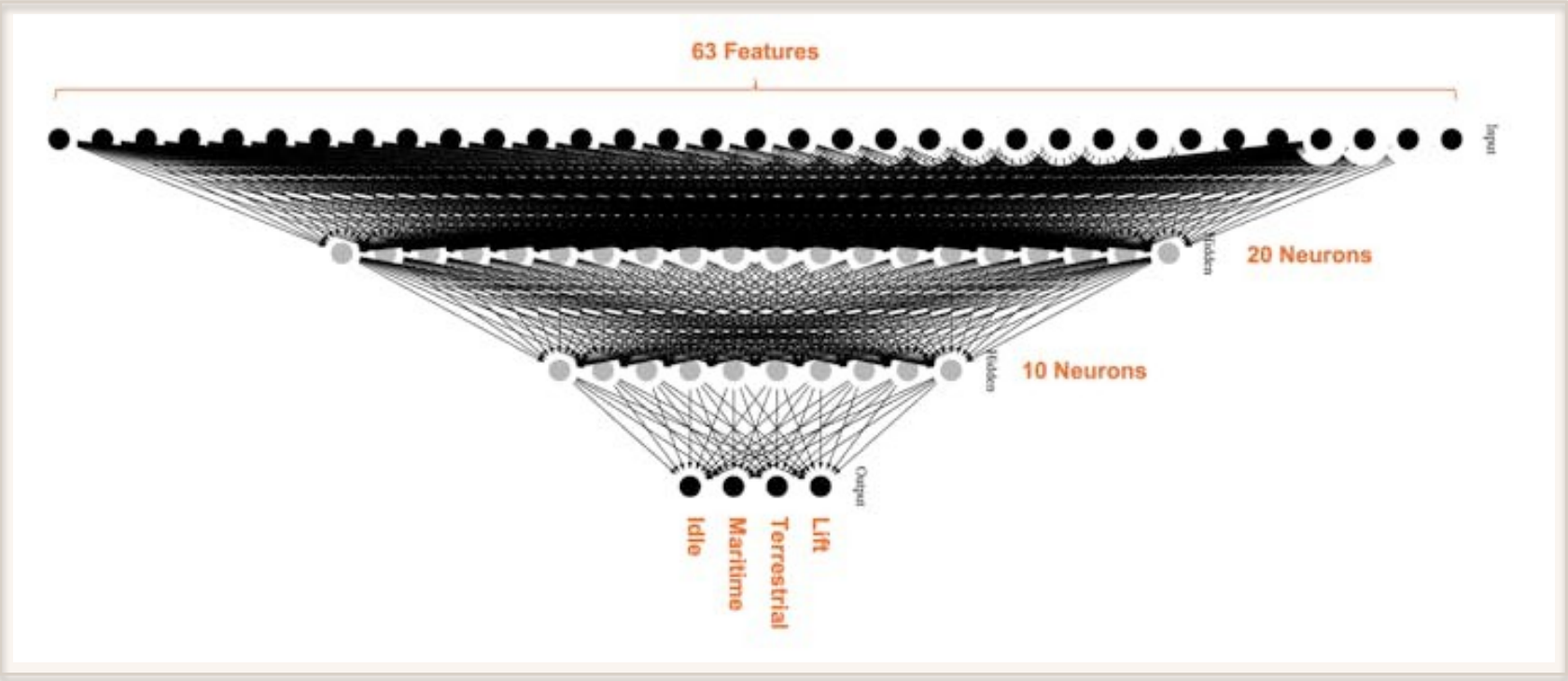
1,312 windows, and four well-separated clusters



1. 5 min 20 s of training data become 1,312 windows of 63 features
2. Normalized, and projected to 2D with UMAP
3. Four clear clusters — the classifier should do well
4. Most important — accZ RMS, accZ 0.78–2.34 Hz, accY 5.47–7.03 Hz

MODEL DESIGN

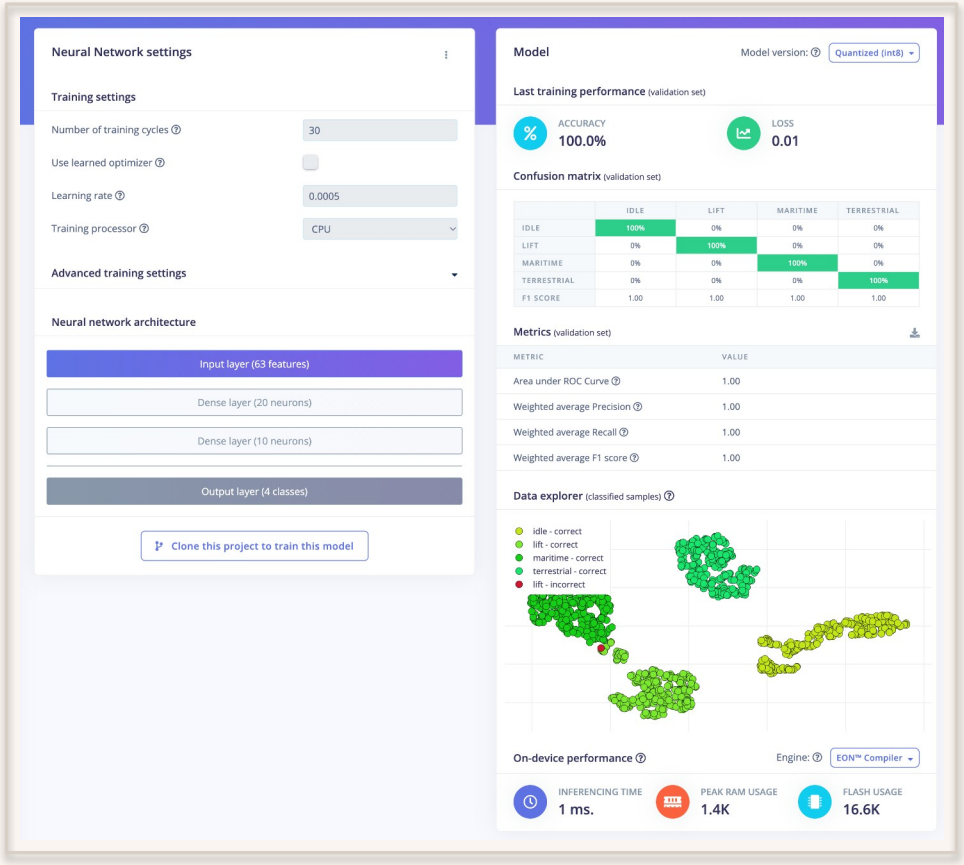
A dense network — 63, 20, 10, 4



1. 63 inputs, one per feature
2. Two hidden layers, 20 and 10 neurons
3. 4 outputs — idle, lift, maritime, terrestrial
4. About 1,500 weights — no convolution needed

TRAINING

30 epochs, and 100 % on validation



1. 30 training cycles, learning rate 0.0005, 20 % for validation
2. Input 63 features → dense 20 → dense 10 → 4 classes
3. Validation accuracy 100 %, loss 0.01 — int8 model
4. On-device estimate: 1 ms, 1.4 KB of RAM, 16.6 KB of flash

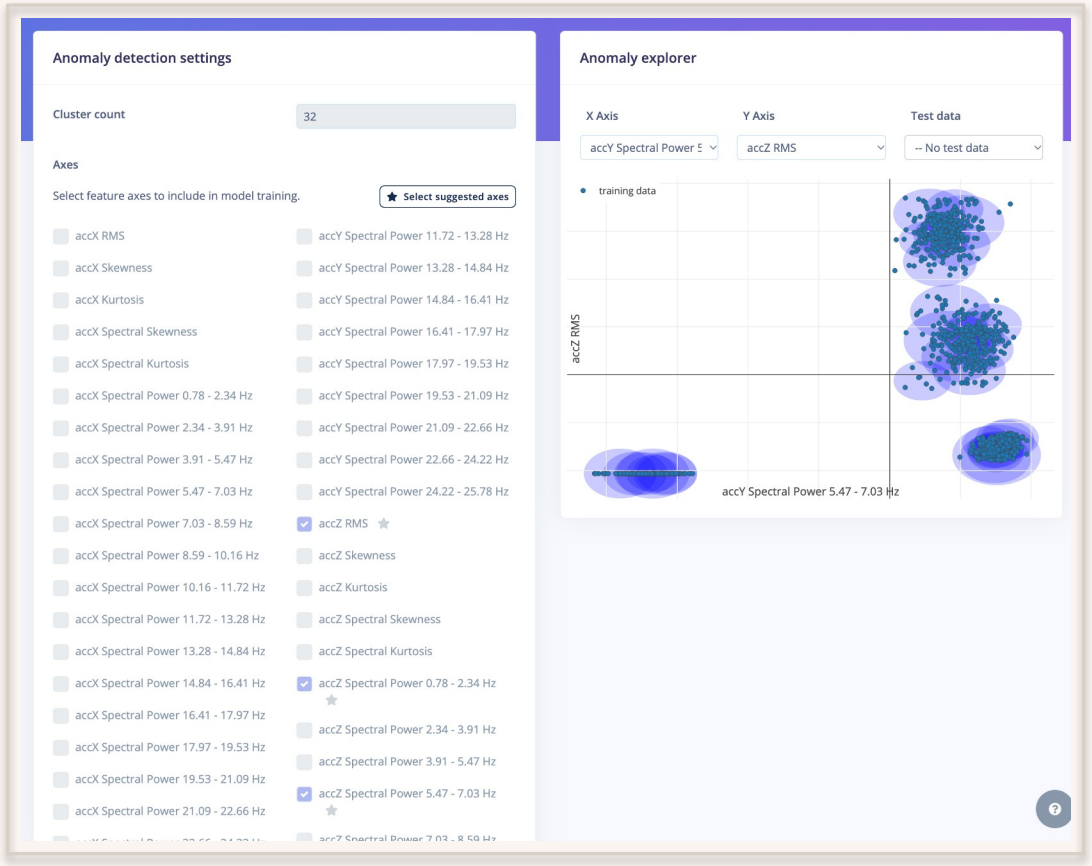
READ 100 % WITH CARE

100 % means these four motions are easy to tell apart. It does not mean the model knows the world.

One person, one hand, one table. A real container on a real truck will look different — and anything outside the four classes still gets one of their names.

ANOMALY DETECTION (K-MEANS)

32 clusters, on the most important features



1. Cluster count 32, as suggested by the Studio
2. Select suggested axes — accZ RMS, accZ 0.78–2.34 Hz, accY 5.47–7.03 Hz
3. Each blue circle is one cluster of normal training data

ANOMALY EXPLORER · TEST DATA

Normal data falls inside the clusters, and scores below zero



1. Orange — a maritime test sample, on accY 5.47–7.03 Hz × accZ RMS
2. It falls inside the maritime clusters
3. Anomaly score: min −0.16, max 0.07, average −0.09
4. Small or negative scores mean normal

MODEL TESTING

99.4 % on test data — and the misses are flagged, not mislabeled

ACTUAL ↓ / PREDICTED →	IDLE	LIFT	MARITIME	TERRESTRIAL	ANOMALY	UNCERTAIN
Idle	100 %	0 %	0 %	0 %	0 %	0 %
Lift	0 %	100 %	0 %	0 %	0 %	0 %
Maritime	0 %	0 %	98.8 %	0 %	1.2 %	0 %
Terrestrial	0 %	0 %	0 %	98.8 %	1.2 %	0 %
F1 score	1.00	1.00	0.99	0.99		

Public project, int8, 99.39 %. The few windows that missed were not called another class — the anomaly score flagged them.

LIVE CLASSIFICATION

Test with the kit, still connected to the Studio



1. Live classification records 10 s from the kit through the forwarder
2. Every window gets four scores and an anomaly score
3. The model runs in the Studio, not on the board yet

04

DEPLOY AND ACT

Motion on the board

An Arduino library, a few edits for the kit's IMU, two answers per window, and the OLED.

DEPLOYMENT

Arduino library, int8 — 5 ms estimated for the XIAOML Kit

estimate for XIAOML Kit.

Quantized (int8)

Selected ✓

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	4 ms.	1 ms.	**5 ms.**
RAM	1.7K	3.1K	**3.1K**
FLASH	-	36.4K	**-**
ACCURACY			**99.39%**

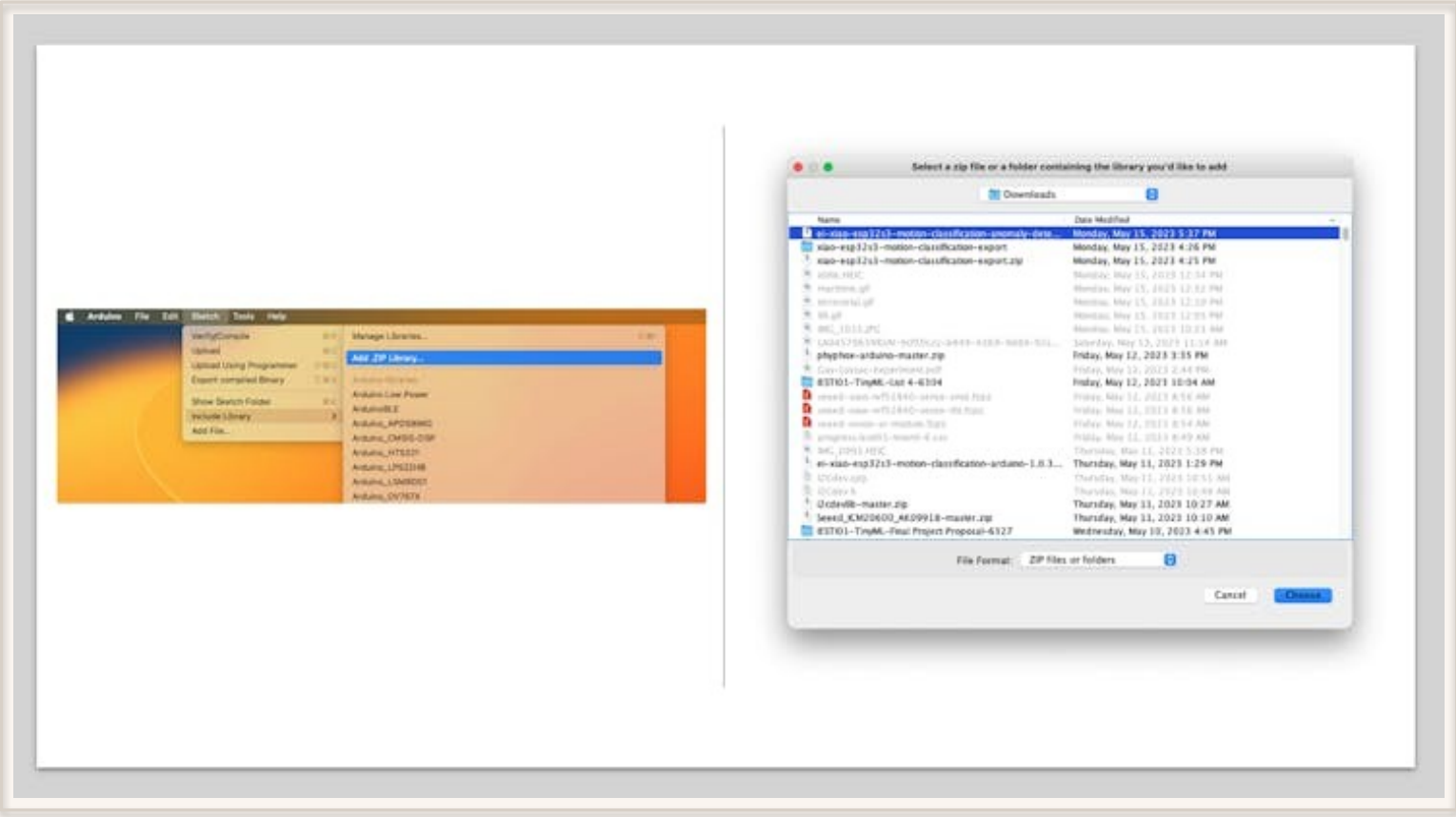
Unoptimized (float32)

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	4 ms.	3 ms.	**7 ms.**
RAM	1.7K	3.3K	**3.3K**
FLASH	-	40.4K	**-**
ACCURACY			**99.39%**

1. Search for and select Arduino library, then Quantized (int8) and Build
2. int8: spectral features 4 ms + classifier 1 ms = 5 ms
3. 3.1 KB of RAM and 36.4 KB of flash
4. float32: 7 ms — the same 99.39 % accuracy

THE LIBRARY

Add the .ZIP library to the Arduino IDE



1. Sketch › Include Library › Add .ZIP Library
2. File › Examples › XIAOML_Kit_Motion_Class_-_AD_inferencing
3. Start from nano_ble_sense_accelerometer — a different board, a few changes

THE EDITS

From the Nano 33 BLE Sense IMU to the kit's

`motion_class_ad_inference.ino` — the changed parts

```
// 1. includes: the LSM6DS3 library instead of Arduino_LSM9DS1
#include <XIAOML_Kit_Motion_Class_-_AD_inferencing.h>
#include <LSM6DS3.h>
#include <Wire.h>

// 2. the IMU object and the range, now in m/s²
LSM6DS3 myIMU(I2C_MODE, 0x6A);
#define CONVERT_G_TO_MS2    9.81f
#define MAX_ACCEPTED_RANGE  2.0f * CONVERT_G_TO_MS2

// 3. setup(): myIMU.begin() instead of IMU.begin()

// 4. loop(): read the kit's IMU, convert first, then clip
float x = myIMU.readFloatAccelX();
buffer[i + 0] = x * CONVERT_G_TO_MS2;
if (fabs(buffer[i + j]) > MAX_ACCEPTED_RANGE)
  buffer[i + j] = copysign(MAX_ACCEPTED_RANGE, buffer[i + j]);
```

— The original clipped in g; here the buffer is in m/s², so convert before the range test.

— ±2 g, the default range of the IMU, is 19.62 m/s².

— The complete sketch is on the lab's GitHub.

THE LOOP

Sample two seconds, classify, then check the anomaly score

motion_class_ad_inference.ino — loop() (excerpt)

```
for (int i = 0; i < EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE; i += 3) {
  uint64_t next_tick = micros() + (EI_CLASSIFIER_INTERVAL_MS * 1000);
  // read x, y, z  ->  buffer[i], buffer[i+1], buffer[i+2]
  delayMicroseconds(next_tick - micros());       // keep 50 Hz
}

numpy::signal_from_buffer(inference_buffer,
                          EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE, &signal);
run_classifier(&signal, &result, debug_nn);

if (max_confidence > 0.6)                        // classifier
  ei_printf("PREDICTION: %s\n", predicted_class.c_str());
else
  ei_printf("UNCERTAIN\n");

if (result.anomaly > 0.5)                        // K-means
  ei_printf("ANOMALY DETECTED! Score: %.3f\n", result.anomaly);
```

— 300 values — 100 readings of 3 axes, one every 20 ms.

— One call to run_classifier runs the features, the network, and K-means.

— Two thresholds, two questions — 0.6 for the class, 0.5 for the anomaly.

SERIAL MONITOR · MARITIME

11 ms to prepare, less than 1 ms to classify

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 11 ms, Classification: 0 ms, Anomaly: 0 ms):
    idle: 0.00000
    lift: 0.01953
    maritime: 0.97266
    terrestrial: 0.00391
Anomaly score: -0.028

PREDICTION: maritime (97.3% confidence)
```

DSP

11 ms

NN + K-MEANS

< 1 ms

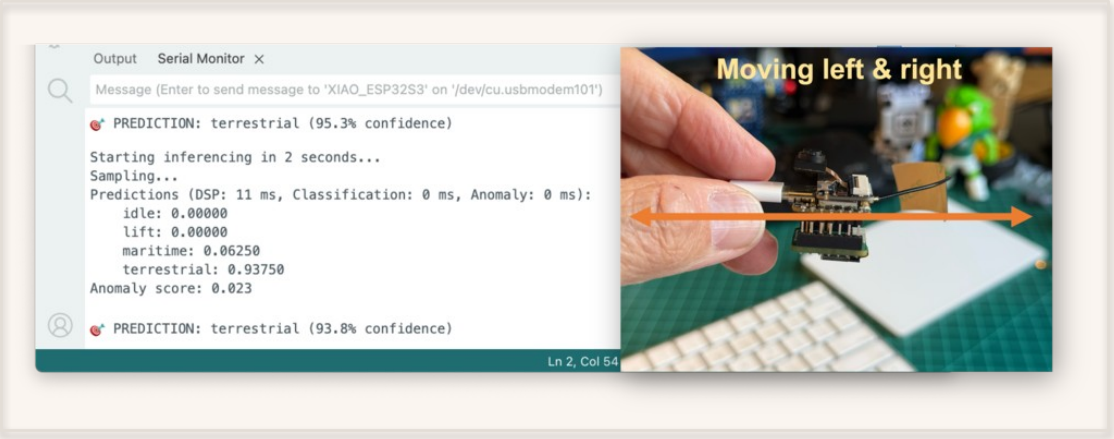
WINDOW

2,000 ms

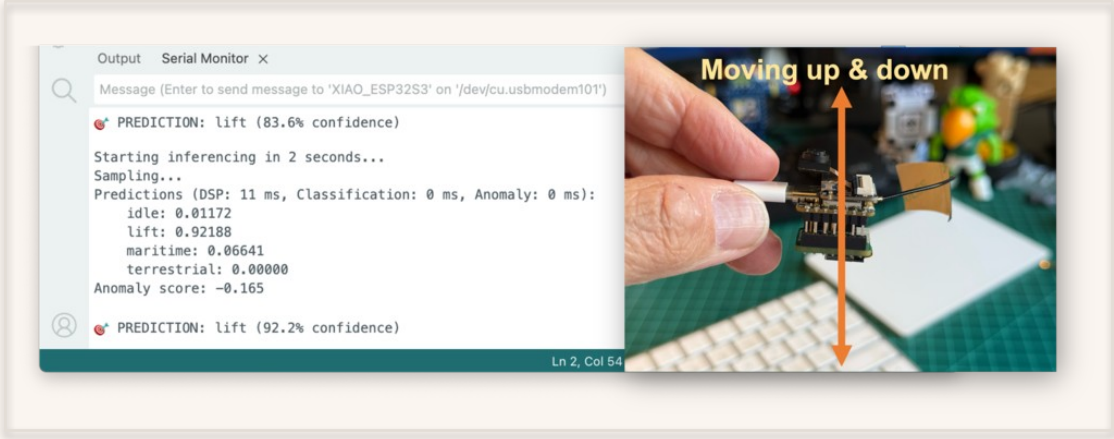
THE OTHER CLASSES

Back and forth is terrestrial; up and down is lift

Terrestrial



Lift



Lift: 92.2 % confidence, anomaly score −0.165. Normal motion keeps the anomaly score below zero.

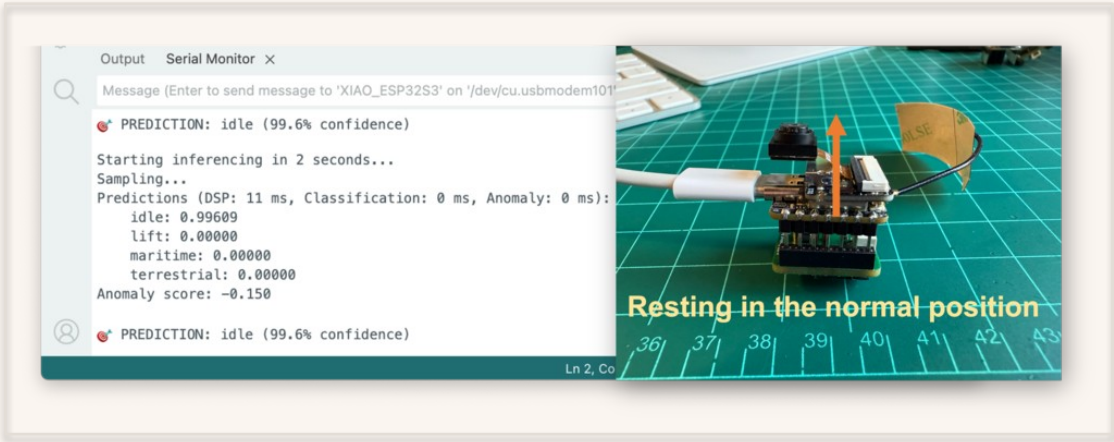
AND THE LAST TWO

Waves are maritime; resting flat is idle

Maritime



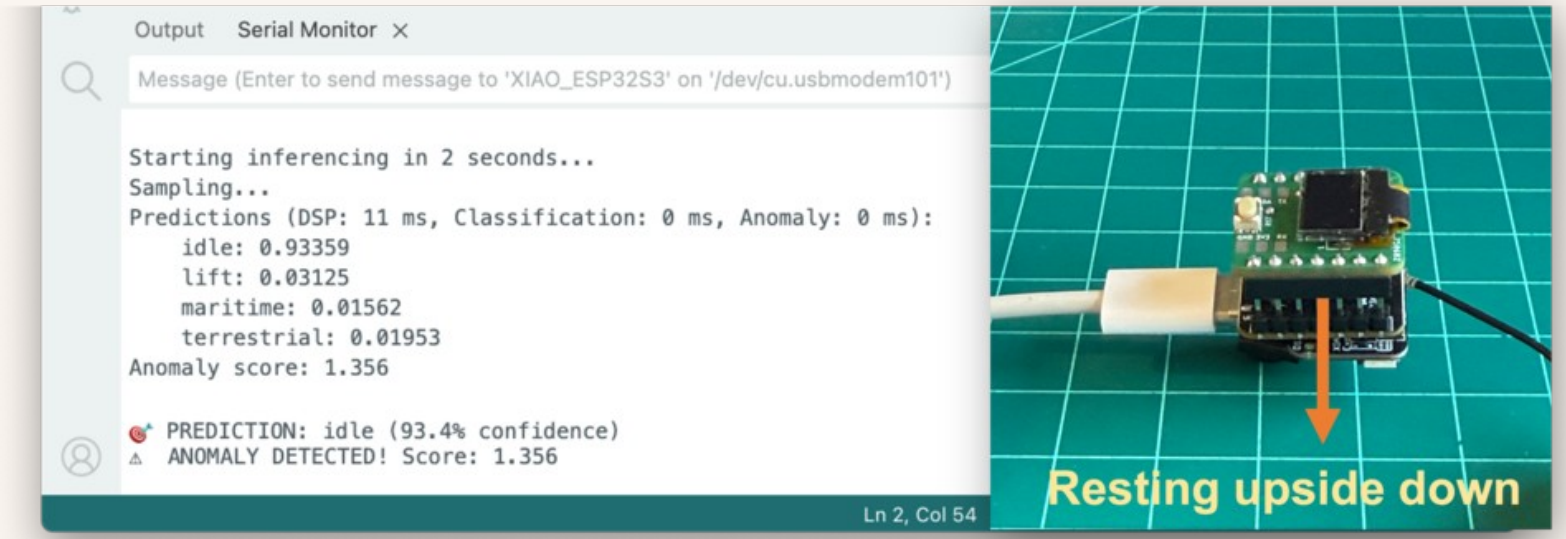
Idle



Maritime: 97.3 %, anomaly score −0.028. Idle, resting in the normal position: 99.6 %, anomaly score −0.150.

NOW, AN ANOMALY

Upside down — "idle" with 93 %, and an anomaly score of 1.356



1. The kit rests upside down — no motion, but gravity now points the other way
2. The classifier says idle, 93.4 % — it can only pick a class it knows
3. K-means scores 1.356, far above 0.5 — ANOMALY DETECTED

THE LESSON OF THIS CHAPTER

A classifier always answers with a class it knows. Only an anomaly detector can say "I have never seen this."

High confidence is not the same as being right. In the field, keep both answers.

THE BUDGET

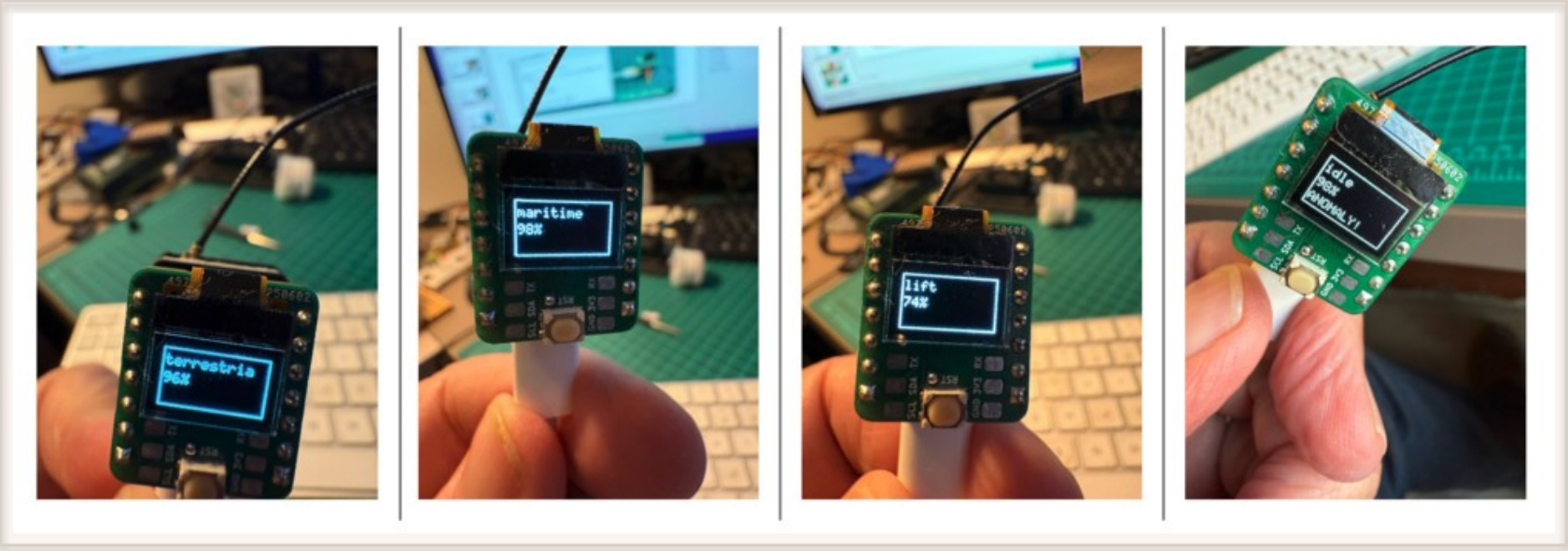
In motion, the clock is the bill

CONFIGURATION	INPUT	PRE-PROCESS	MODEL	OUTPUT
Ch. 5 motion, XIAO (book)	2 s, 3 axes	11 ms	< 1 ms	4 scores + anomaly
Ch. 4 KWS, XIAO (book)	1 s audio	515 ms	3 ms	4 scores
Ch. 3 FOMO, XIAO	96^2, gray	4 ms	143 ms	centroids
Ch. 2 classifier, XIAO (measured)	96^2, RGB	4 ms	199 ms	1 label

Same board in every row. For motion, the board computes for about 11 ms and waits 2,000 ms for the window to fill.

POST-PROCESSING · XIAOML KIT OLED

The motion class on the display, no computer needed



1. Terrestrial, maritime, lift, and idle, each on the 0.42" OLED
2. Powered by a battery, a power bank, or any 5 V supply
3. The OLED sketch is on the lab's GitHub

BEYOND CONTAINERS

The same pipeline works for any vibration or motion

01

Asset tracking

Logistics monitoring — what happened to the cargo, and when.

02

Predictive maintenance

A motor that starts to vibrate differently is an anomaly before it is a failure.

03

Human activity

Walking, running, falling — on a wrist or in a phone.

04

Vehicles and cities

Equipment monitoring and IoT sensor networks.

SUMMARY

Three things to remember

01 Features do the heavy lifting

300 raw values become 63 spectral features in 11 ms, so a 63-20-10-4 network is enough.

02 A classifier always picks a class

Upside down, it said idle with 93 %. K-means, trained only on normal data, scored 1.356 and flagged it.

03 In motion, the clock is the bill

The board computes for about 11 ms and waits 2 s for the window. To react faster, shorten the window.

CHECK YOURSELF

The kit lies upside down. The board prints "PREDICTION: idle (93.4%)". Which output tells you something is wrong?

A The idle confidence — anything above 0.6 is reliable

B The anomaly score — 1.356, above the 0.5 threshold

C The DSP time — 11 ms means the sensor failed

The classifier can only choose among the four classes it was trained on, so its confidence says nothing about new situations. K-means measures the distance to normal data.

The interactive deck gives per-answer feedback.

Questions?

That closes Part I — vision, sound, and motion, on one small board.

