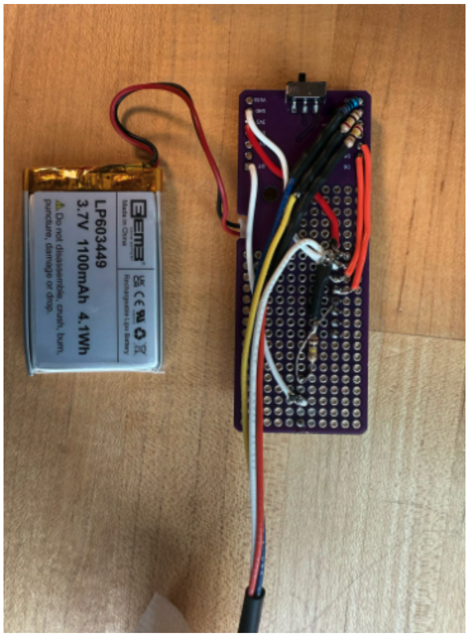
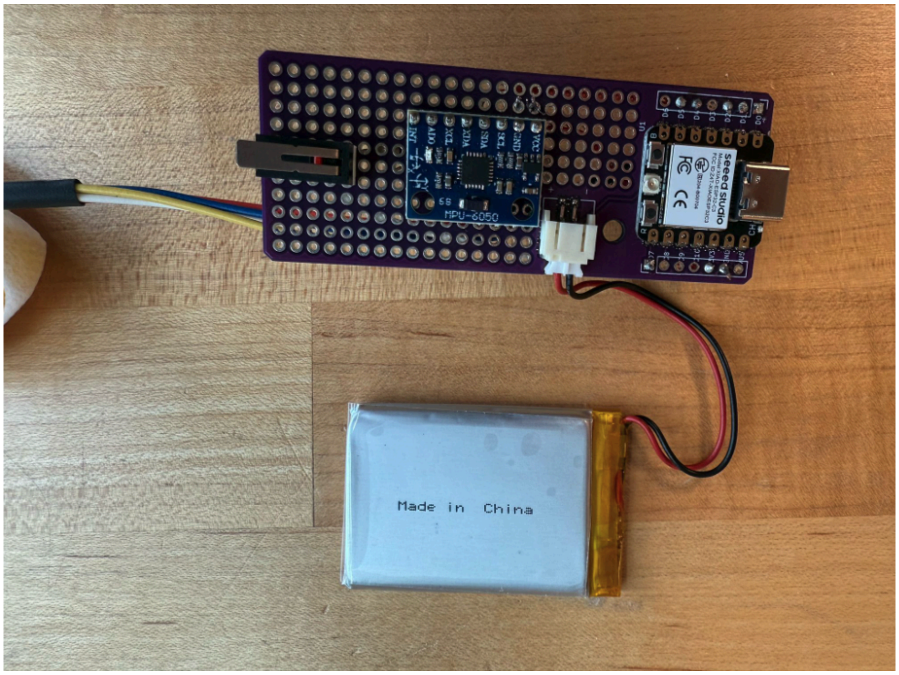
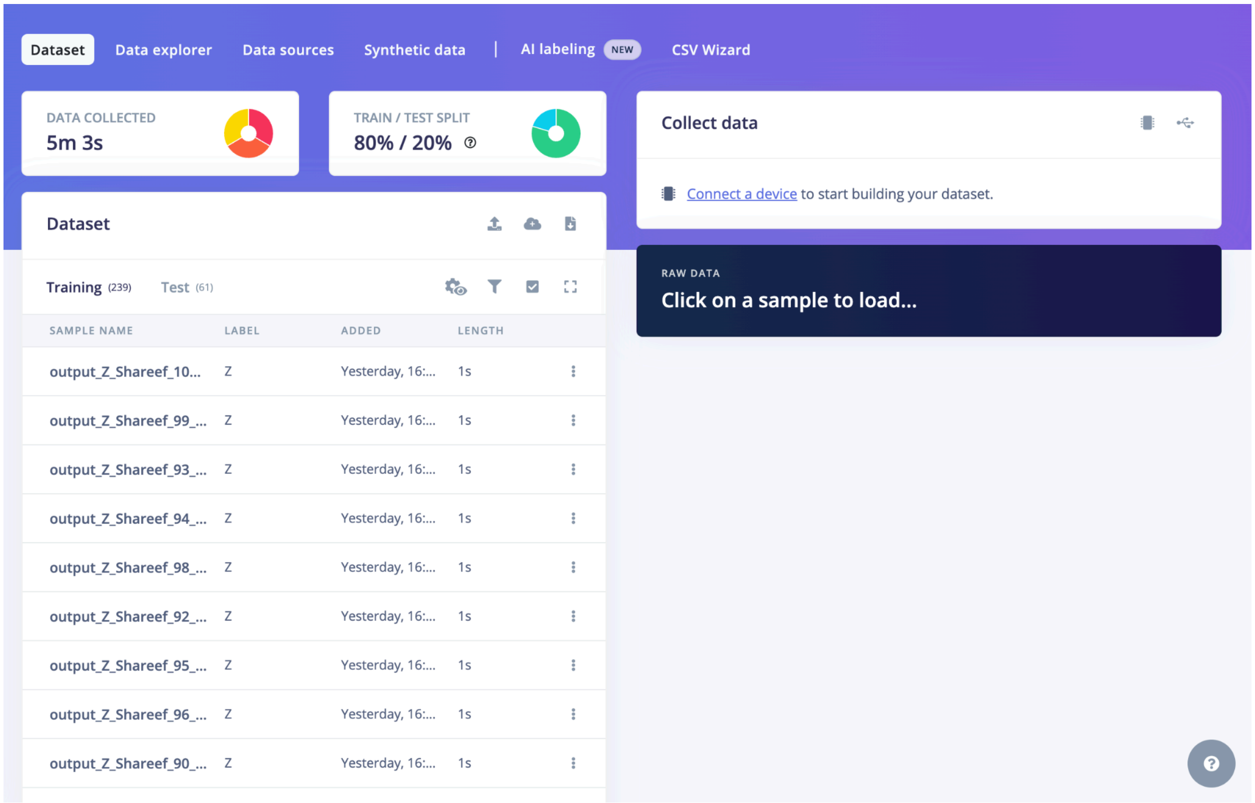
Pictures of hardware setup and connections


LP603449
3.7V 1100mAh 4.1Wh
Do not disassemble, crush, burn, puncture, damage or drop.
Made in China
Rechargeable Lipo Battery



Data collection process and results

Data



DATA COLLECTED
5m 3s

TRAIN / TEST SPLIT
80% / 20%

Collect data

Connect a device to start building your dataset.

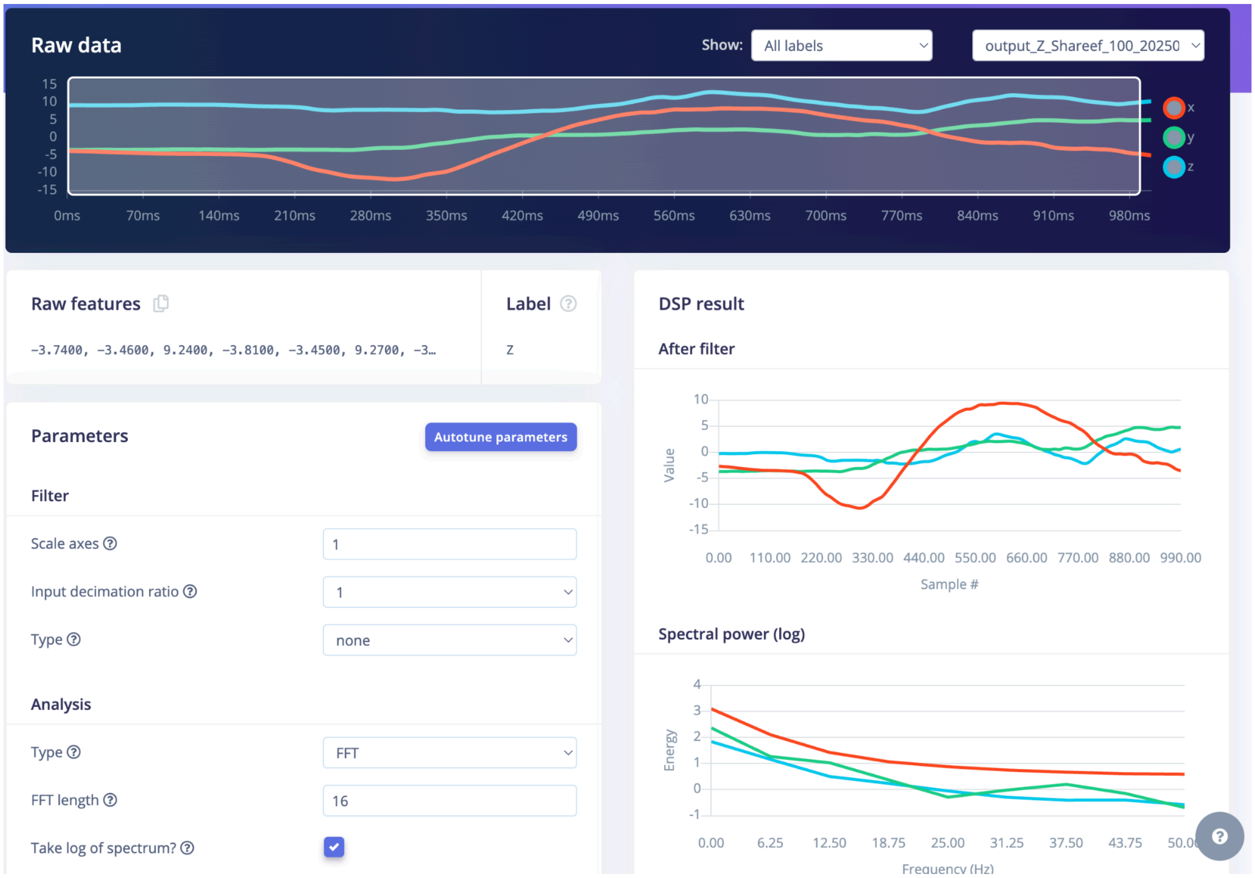
Dataset

Training (239) Test (61)

SAMPLE NAME	LABEL	ADDED	LENGTH
output_Z_Shareef_10...	Z	Yesterday, 16:...	1s
output_Z_Shareef_99_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_93_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_94_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_98_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_92_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_95_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_96_...	Z	Yesterday, 16:...	1s
output_Z_Shareef_90_...	Z	Yesterday, 16:...	1s

RAW DATA

Click on a sample to load...



Raw features

```
-3.7400, -3.4600, 9.2400, -3.8100, -3.4500, 9.2700, -3…
```

Label

z

Parameters

Filter

Scale axes ⓘ: 1

Input decimation ratio ⓘ: 1

Type ⓘ: none

Analysis

Type ⓘ: FFT

FFT length ⓘ: 16

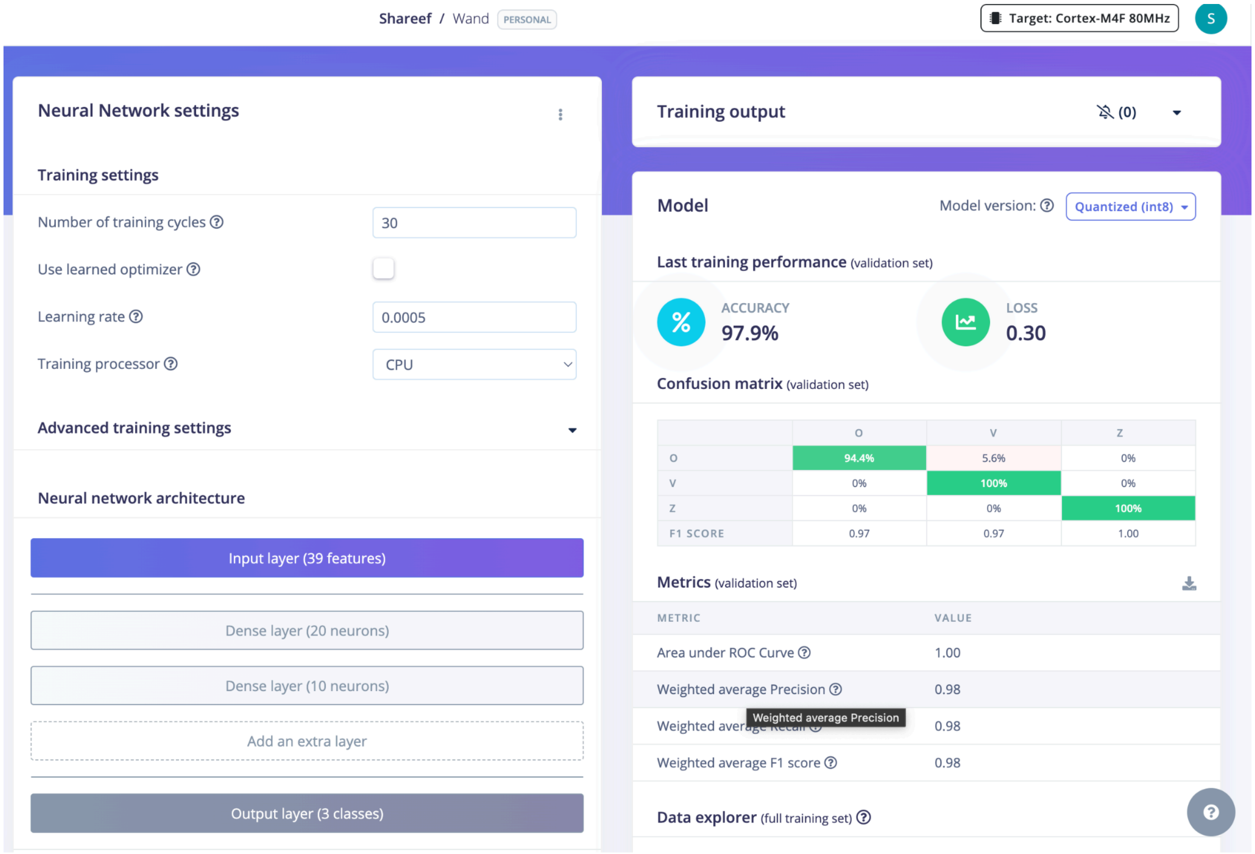
Take log of spectrum? ⓘ ☑

DSP result

After filter

Spectral power (log)

Edge Impulse model architecture and optimization
3 Layers: Input (300) → Dense (100) → Output (3 Classes)



Shareef / Wand PERSONAL

Target: Cortex-M4F 80MHz

S

Neural Network settings

Training settings

Number of training cycles ⓘ	30
Use learned optimizer ⓘ	□
Learning rate ⓘ	0.0005
Training processor ⓘ	CPU

Advanced training settings

Neural network architecture

Input layer (39 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (3 classes)

Training output

(0)

Model

Model version: ⓘ Quantized (int8)

Last training performance (validation set)

ACCURACY
97.9%

LOSS
0.30

Confusion matrix (validation set)

	O	V	Z
O	94.4%	5.6%	0%
V	0%	100%	0%
Z	0%	0%	100%
F1 SCORE	0.97	0.97	1.00

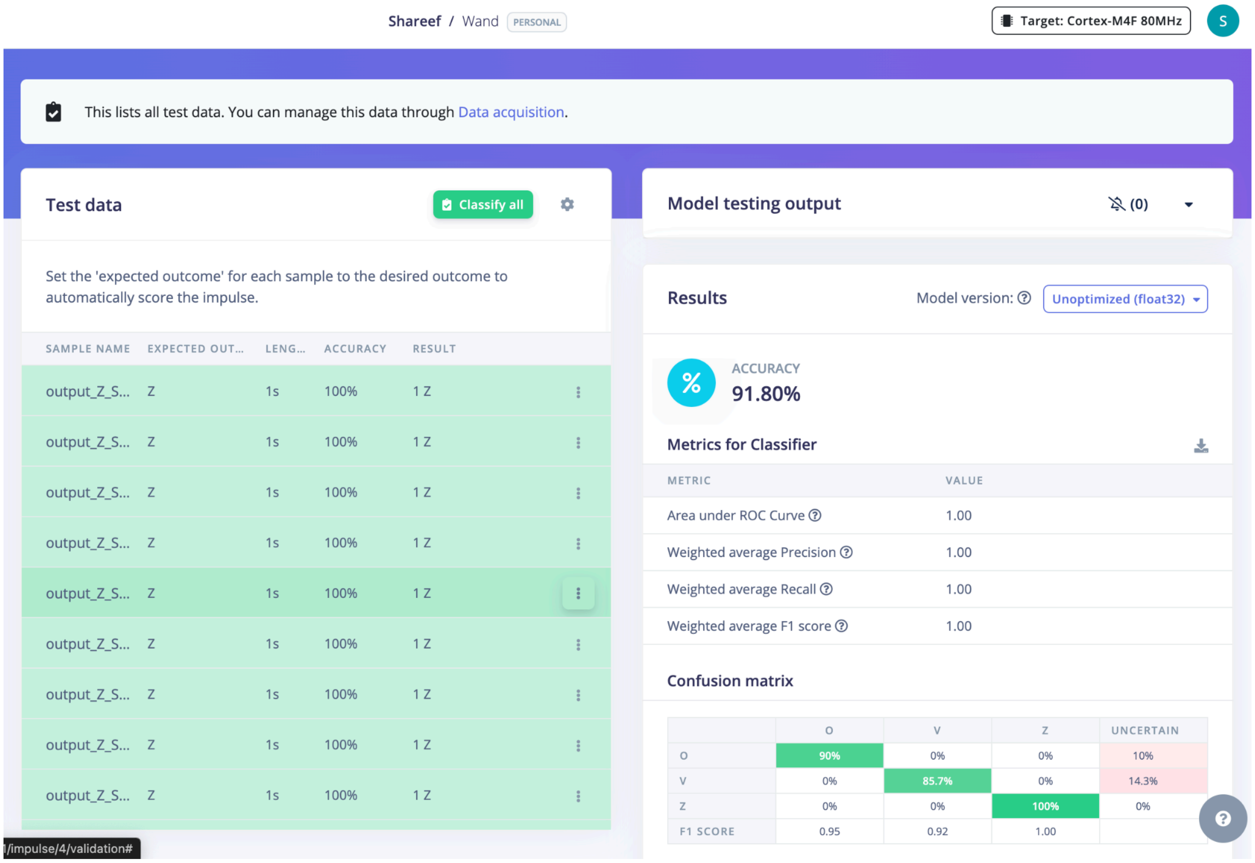
Metrics (validation set)

METRIC	VALUE
Area under ROC Curve ⓘ	1.00
Weighted average Precision ⓘ	0.98
Weighted average Recall ⓘ	0.98
Weighted average F1 score ⓘ	0.98

Weighted average Precision

Data explorer (full training set) ⓘ

Performance analysis and metrics



Shareef / Wand PERSONAL

Target: Cortex-M4F 80MHz

This lists all test data. You can manage this data through Data acquisition.

Test data

Classify all

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUT...	LENG...	ACCURACY	RESULT
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z
output_Z_S...	Z	1s	100%	1 Z

/impulse/4/validation#

Model testing output

(0)

Results

Model version: Unoptimized (float32)

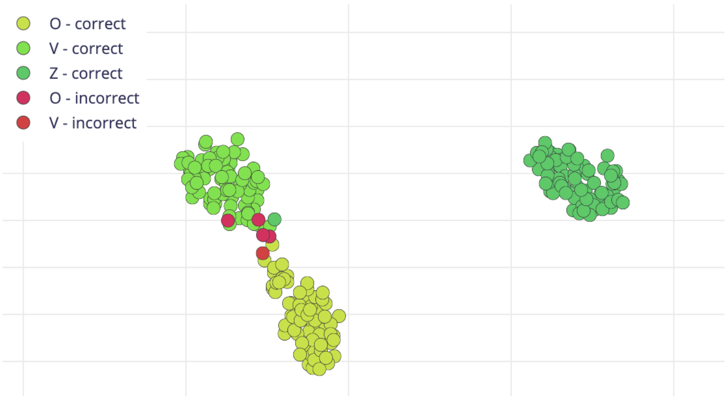
ACCURACY 91.80%

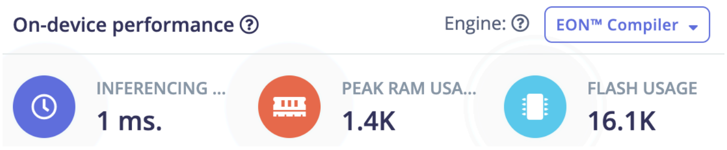
Metrics for Classifier

METRIC	VALUE
Area under ROC Curve	1.00
Weighted average Precision	1.00
Weighted average Recall	1.00
Weighted average F1 score	1.00

Confusion matrix

	O	V	Z	UNCERTAIN
O	90%	0%	0%	10%
V	0%	85.7%	0%	14.3%
Z	0%	0%	100%	0%
F1 SCORE	0.95	0.92	1.00	





On-device performance

Engine: EON™ Compiler

INFERENCING ... 1 ms.

PEAK RAM USA... 1.4K

FLASH USAGE 16.1K

Demo video link
https://github.com/shareefjasim/TECHIN515_Sp25/tree/main/lab4

Discussion Questions

Q1: Why should you use training data collected by multiple students?

Using diverse data improves generalization. A model trained only on one person may overfit to that person's motion patterns, making it less reliable when used by others. Multiple contributors simulate real-world variability, enhancing robustness.

Q2: Effect of window size in Edge Impulse?

Larger window sizes capture longer movements, but may reduce responsiveness and increase model complexity. Smaller windows may miss slow gestures. Our window size (~1s) balanced accuracy and inference time, capturing complete gestures while fitting MCU constraints.

Q3: Two strategies to improve model performance?

1. **Data Augmentation** – Apply jitter, scaling, or slight time warps to increase robustness.

2. **Model Pruning** – Reduce unnecessary weights/layers to improve performance on-device while maintaining accuracy.

Q4: Challenges Faced & Solutions

- ESP32 port detection issues: resolved by manually specifying port.

- Data inconsistency due to gesture speed: fixed by enforcing 1s window + standard motion.

- LED pin conflicts: ensured unique GPIOs with series resistors.

- False predictions: improved model with balanced dataset and edge smoothing.