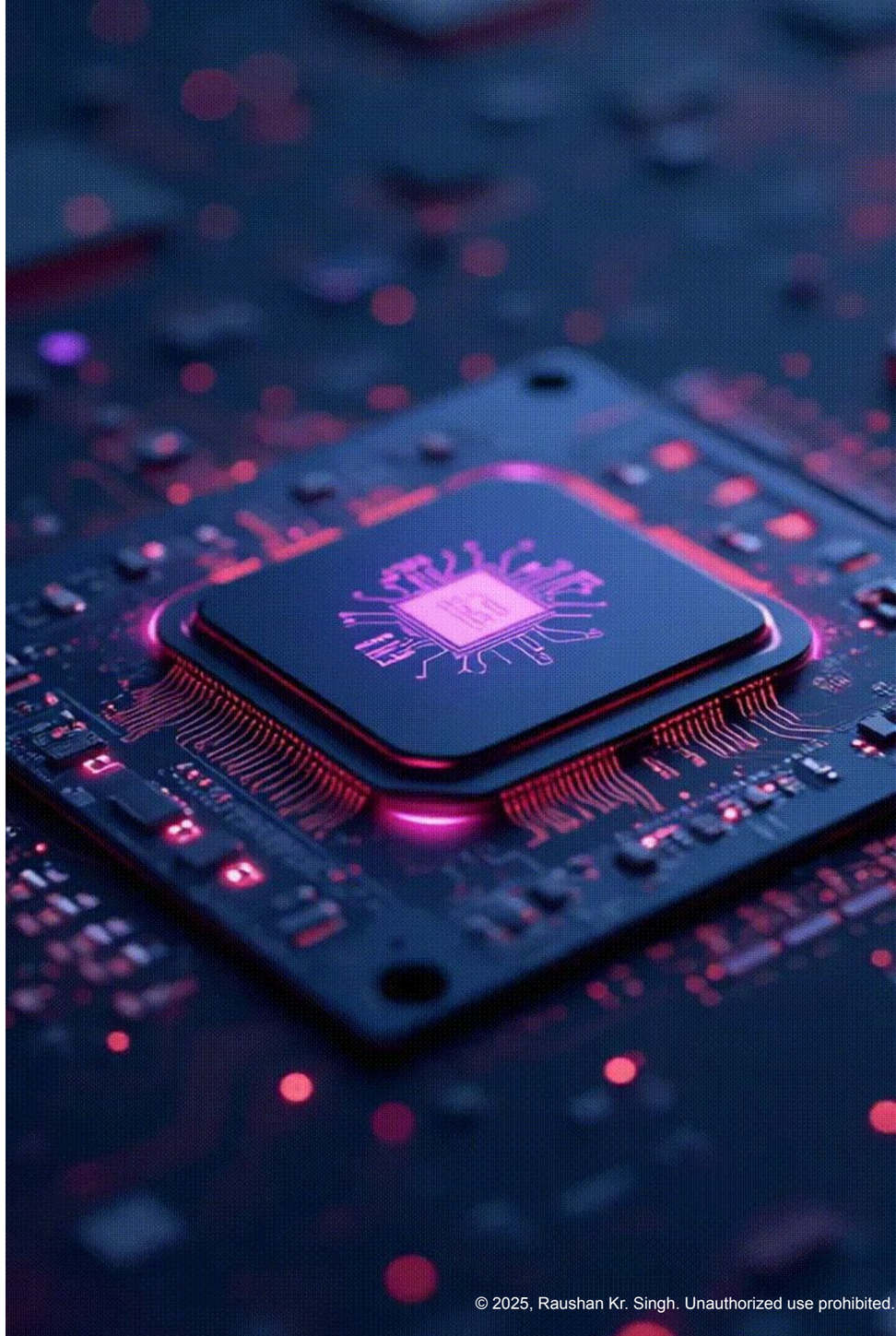
TinyML : Edge AI Bootcamp: Building Intelligent IoT with TinyML (PART A & B)

Raushan Kr. Singh

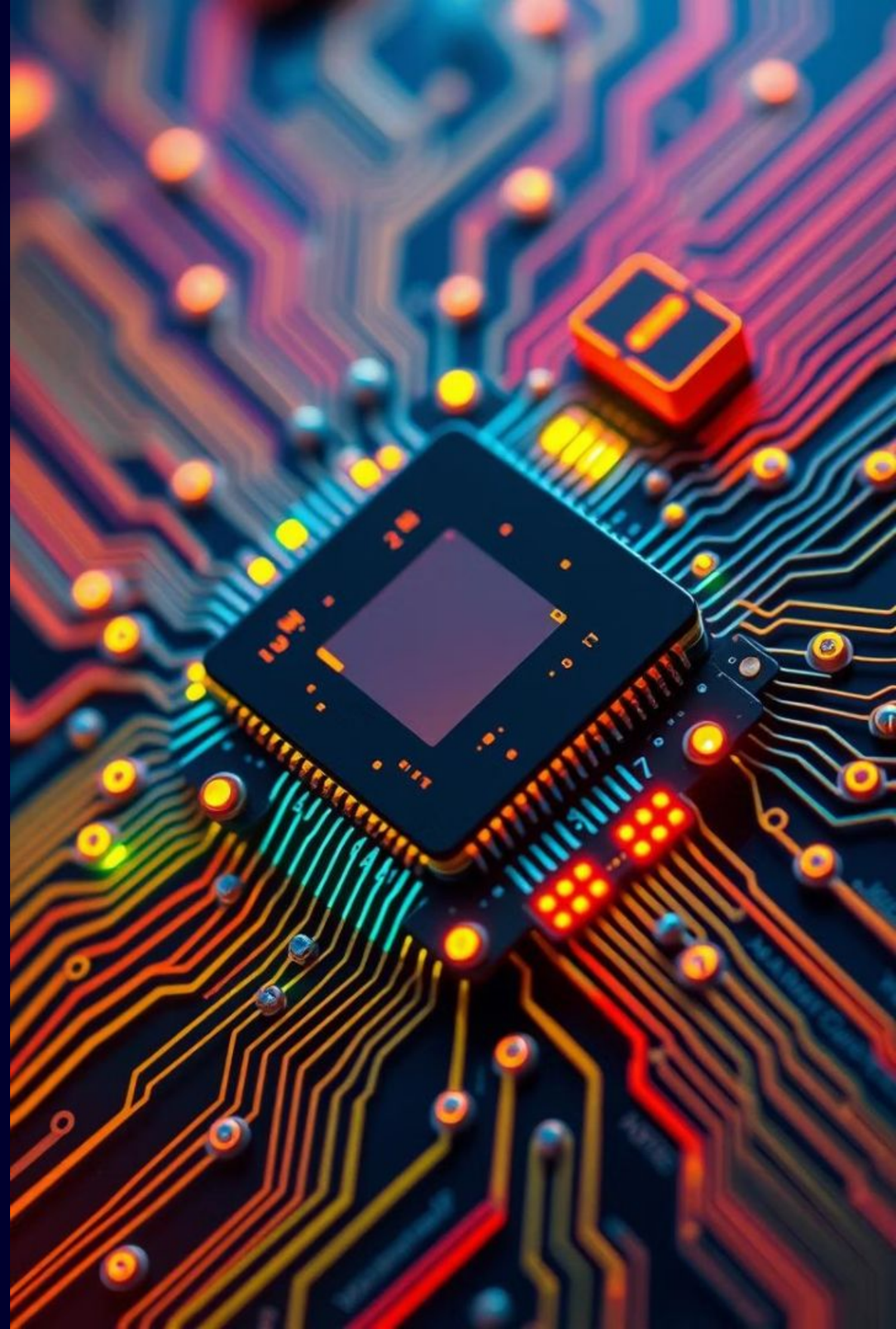
CEO, Fulectronix Technologies

IIT Ropar



Edge Impulse: Powering TinyML at the Edge

Build, train, and deploy machine learning models directly on edge devices



Edge Impulse: The Edge AI Platform

End-to-End Development

From data collection to model deployment, Edge Impulse offers a complete development pipeline.

Device Compatibility

It targets microcontrollers, sensors, and other edge devices, with over 40 sensor integrations.

Real-World Applications

The platform powers diverse applications like predictive maintenance and human sensing.

Why Use Edge Impulse?

Edge Impulse simplifies complex machine learning development, making it accessible to a wider range of developers.

Simplified ML Development

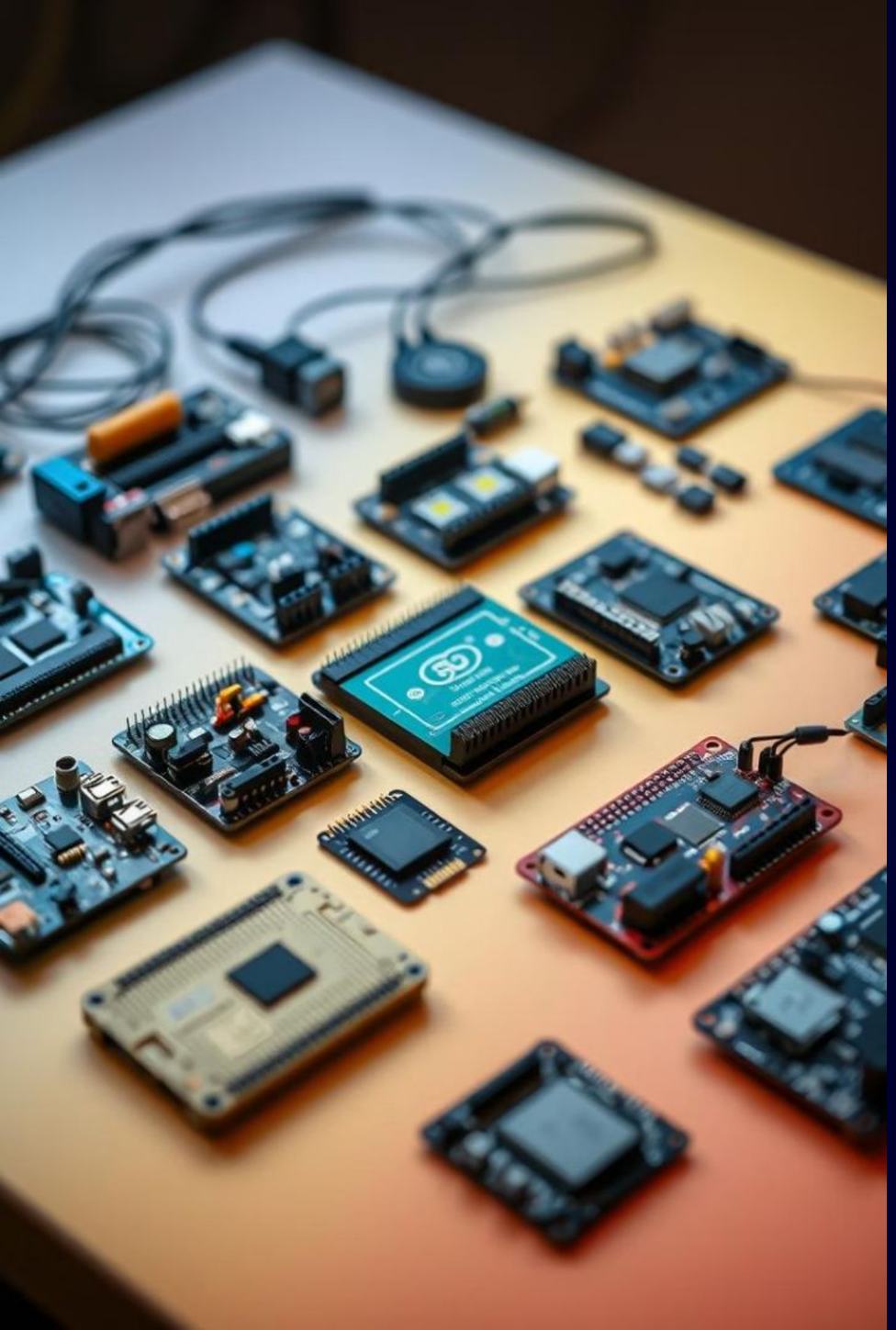
Access powerful no-code/low-code ML tools.

- Visual interface for model building.
- Pre-built machine learning blocks.

Flexible Code Generation

Generate optimized code in Python, C++, and Javascript.

- Supports various programming languages.
- Seamless integration into existing projects.



Edge Impulse Benefits

Edge Impulse offers a range of benefits for efficient and effective edge ML deployment.

Real-time Data

Collect data directly from devices in real time.

Broad Compatibility

Works with Arduino, ESP32, Raspberry Pi, and more.

Optimized Performance

Models are lightweight, fast, and function offline.

Reduced Model Size

Achieve up to an 80% reduction in model size.

Edge Impulse: Streamlined ML Development

Edge Impulse offers specialized tools for each stage of the machine learning development process.

Data Collection

Utilize the integrated data forwarder tool.

Feature Extraction

Access various signal processing blocks.

Model Training

Leverage AutoML, TensorFlow Lite, and CoreML.

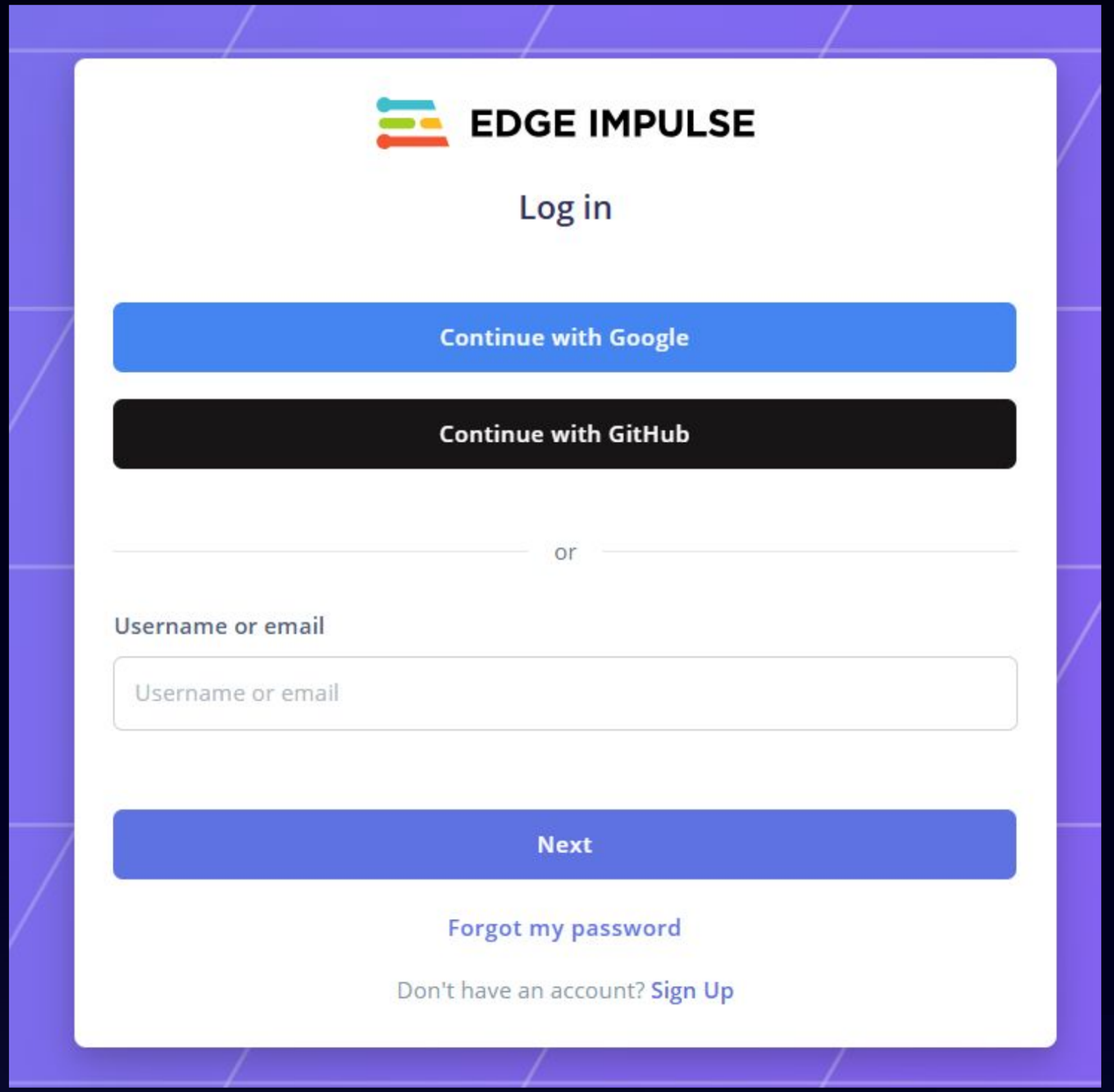
Deployment

One-click deployment to various target devices.

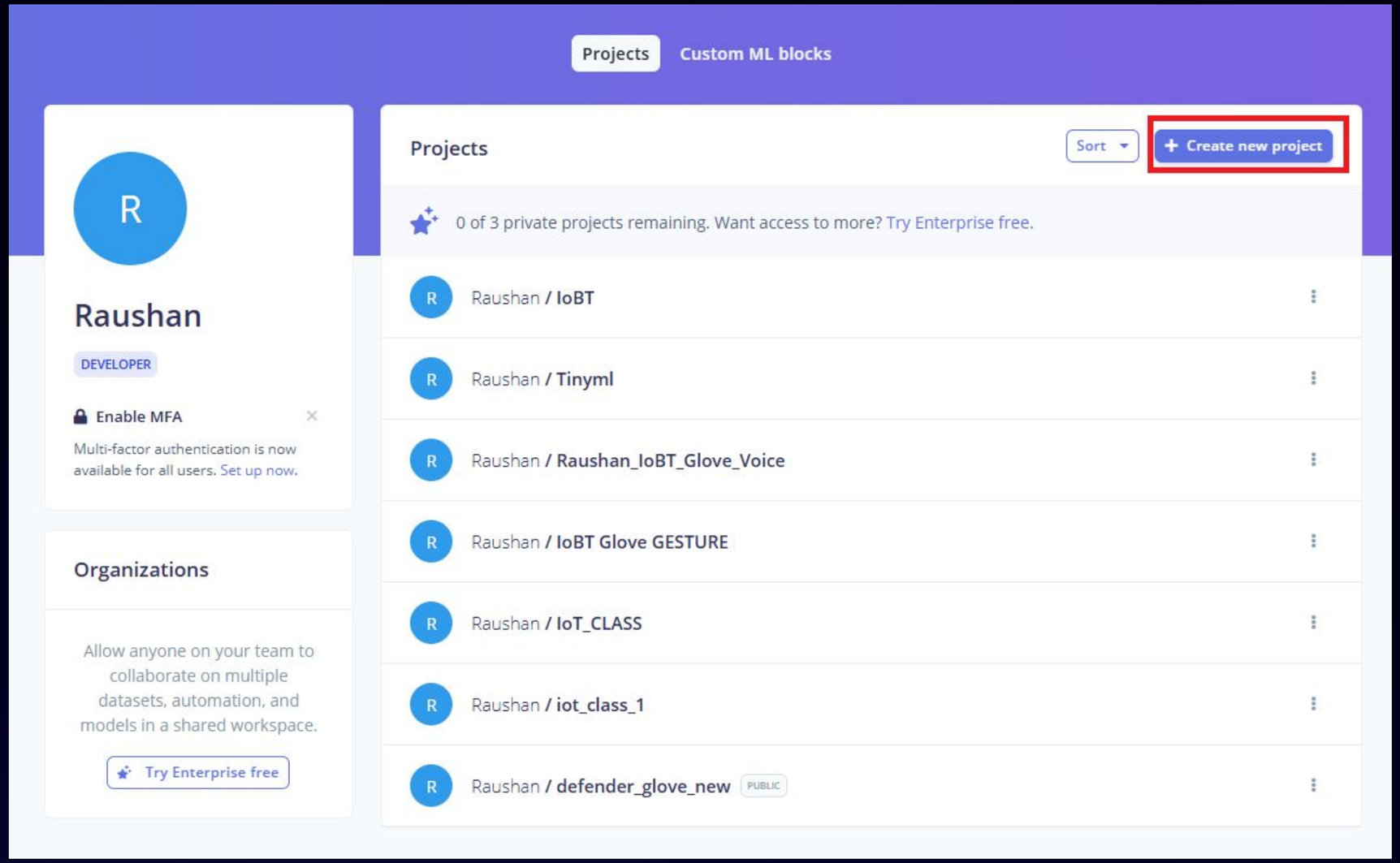
EDGE IMPULSE

Simulation

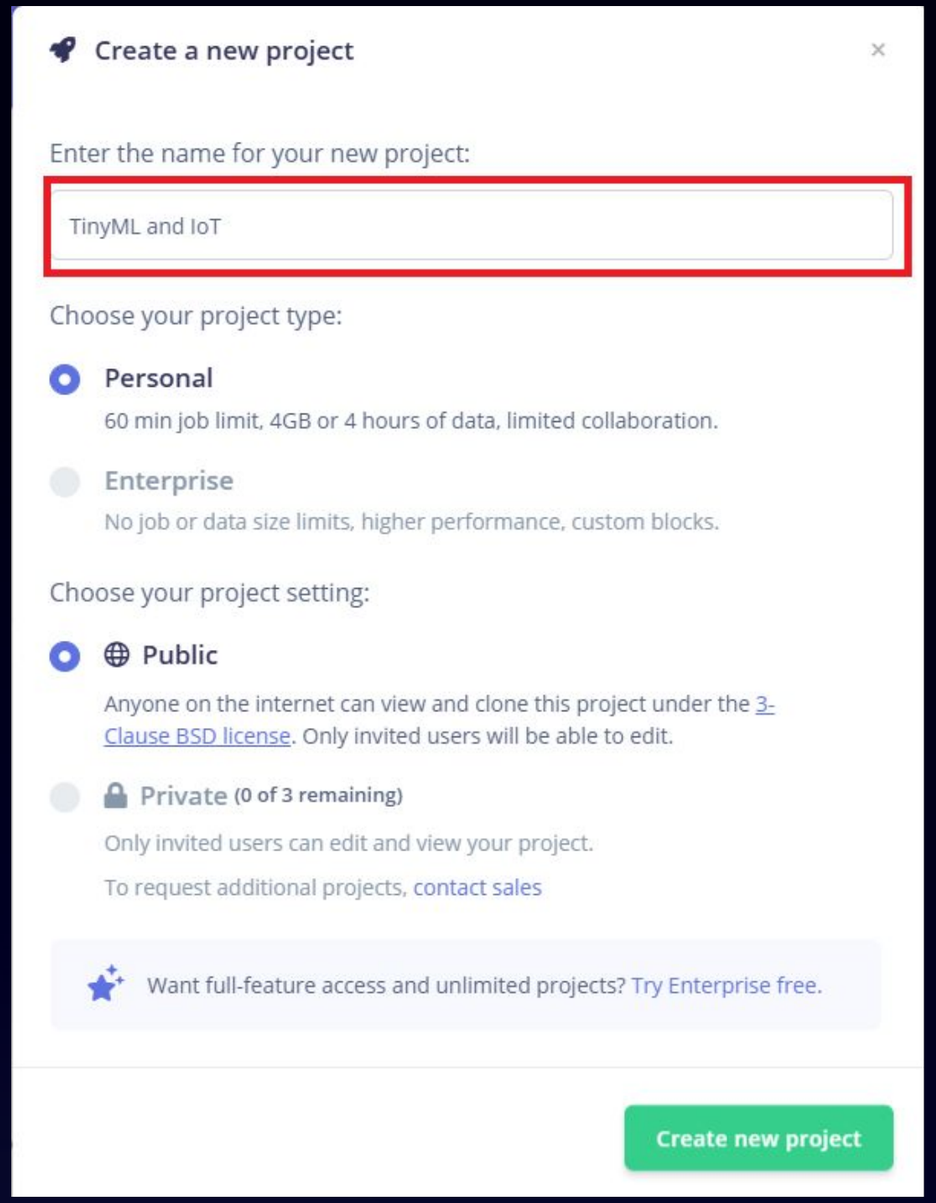
Step 1: Create account on Edge Impulse Studio & Login



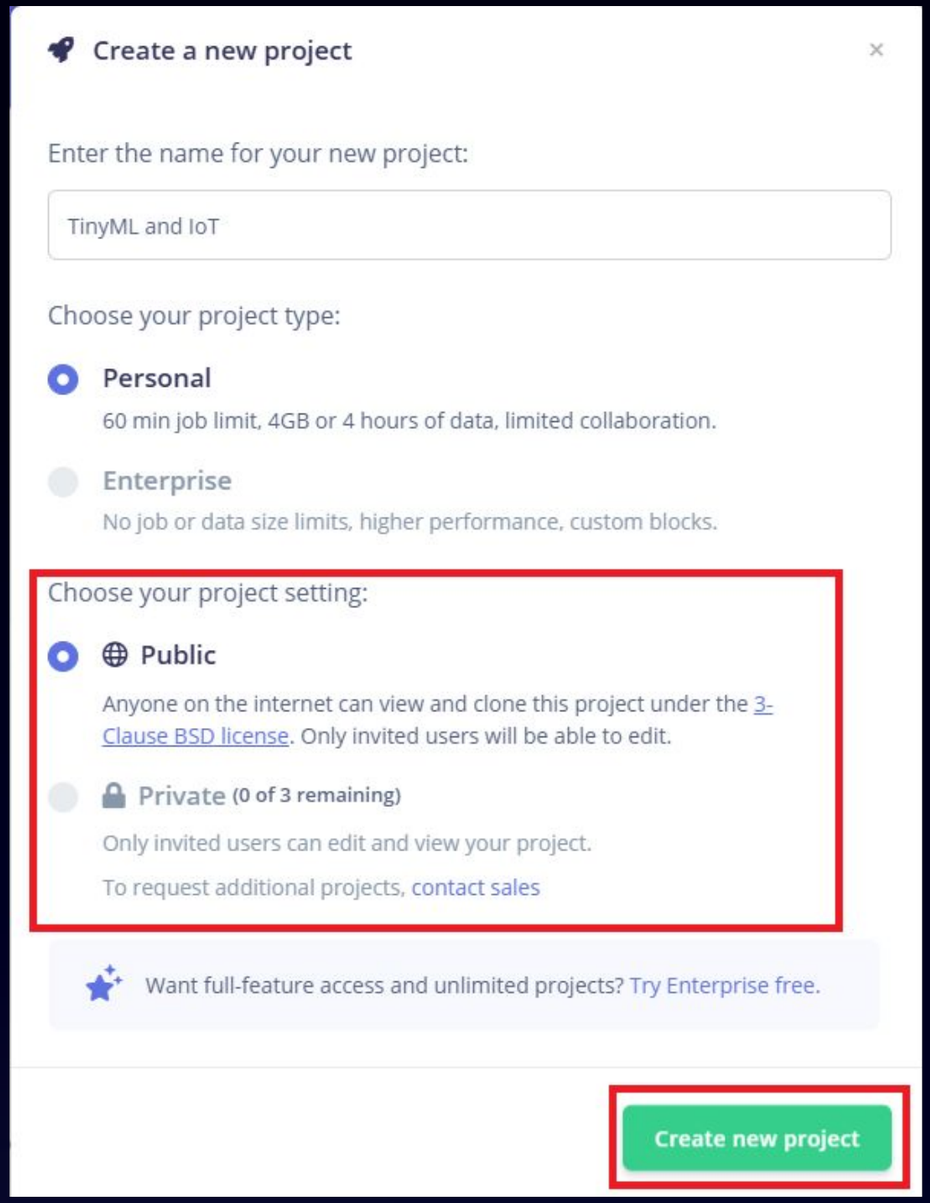
Step 2: Create a new project



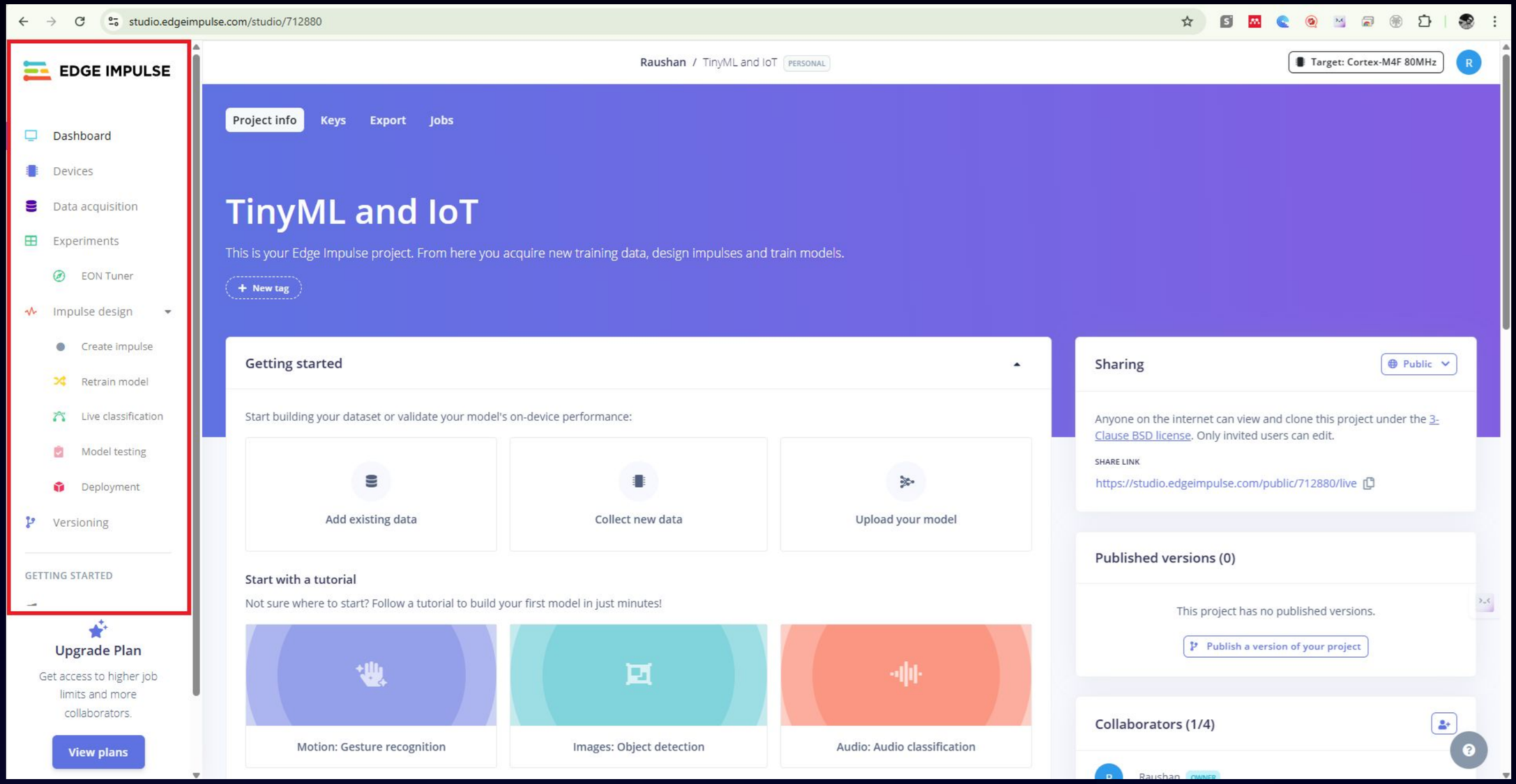
Step 3: Enter Project Name and Select Setting



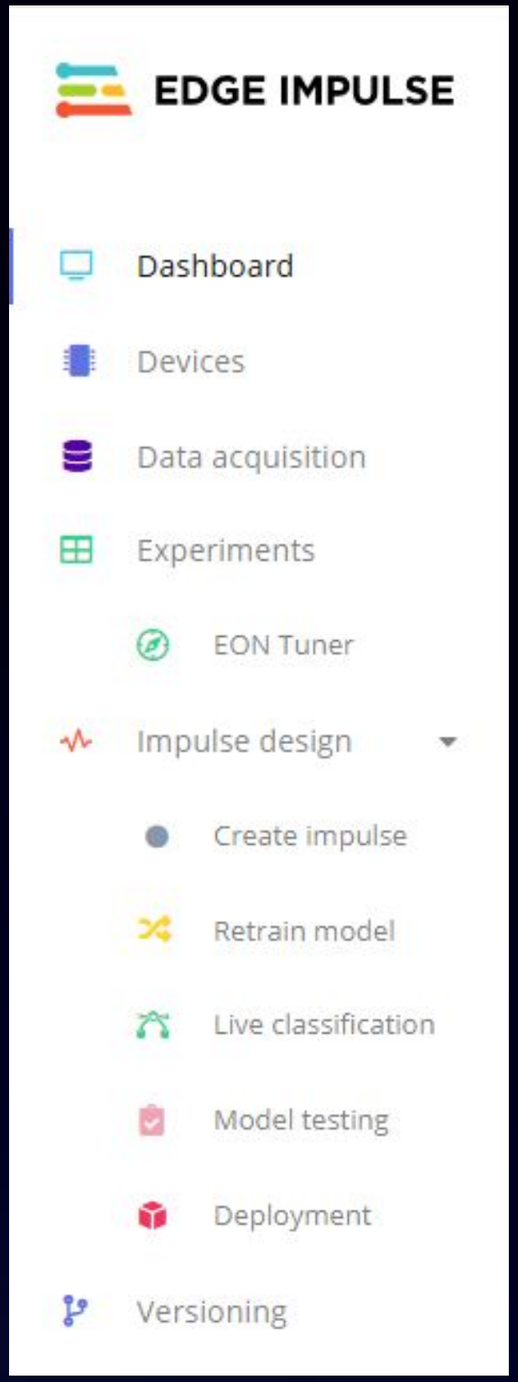
Step 4: Enter Project Name and Select Setting



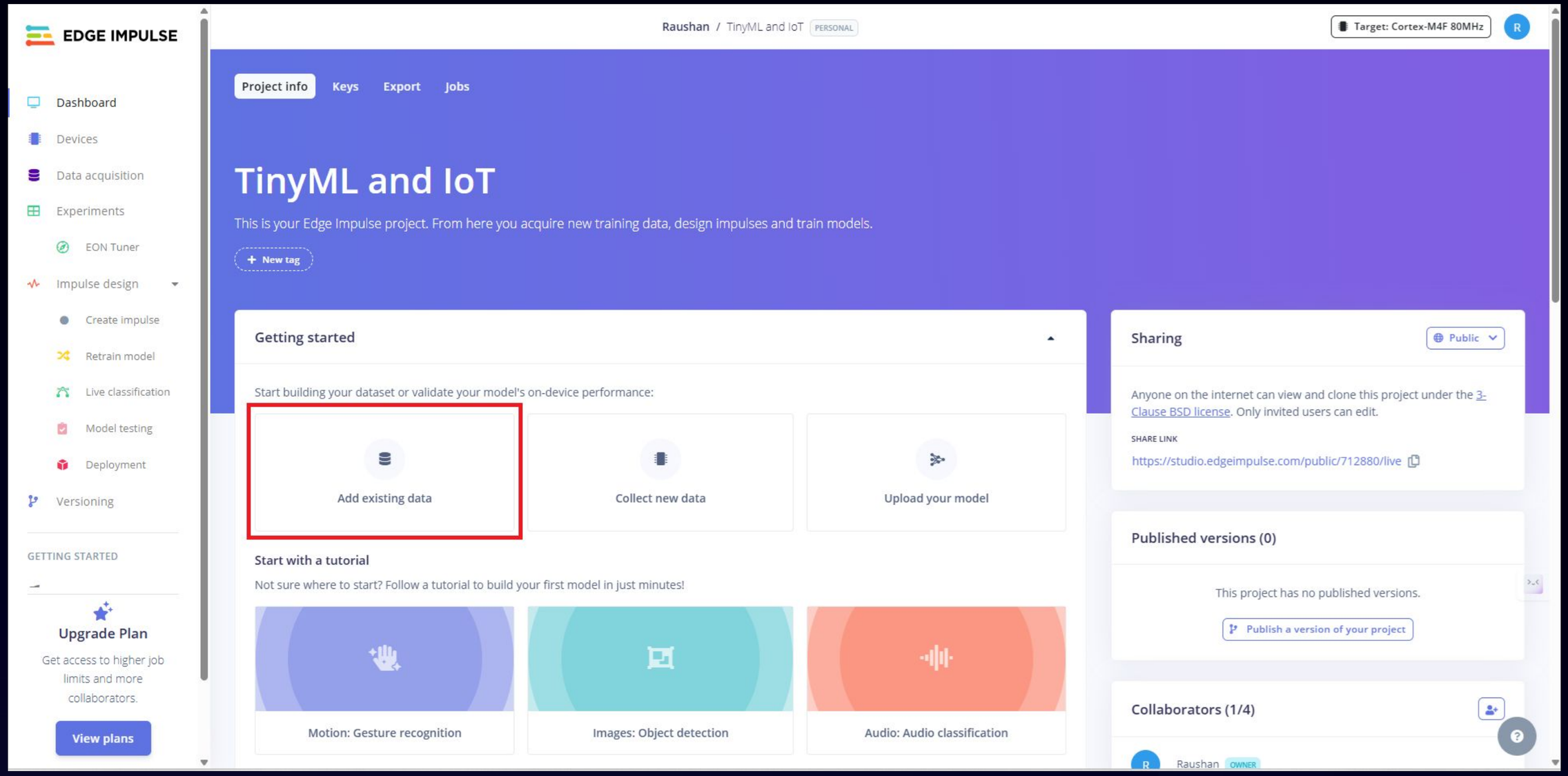
Step 5: Getting Started



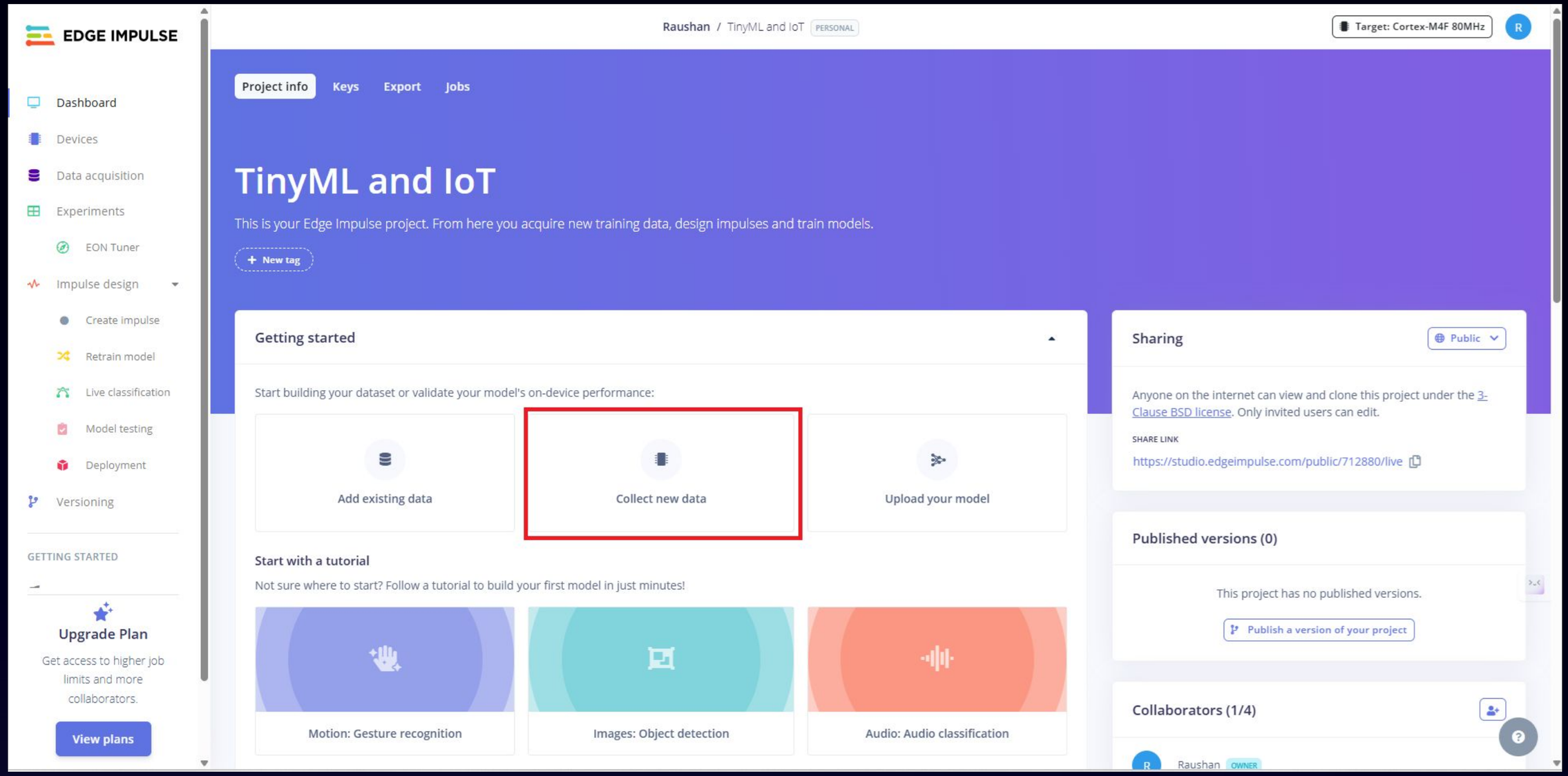
Step 6: Getting Started



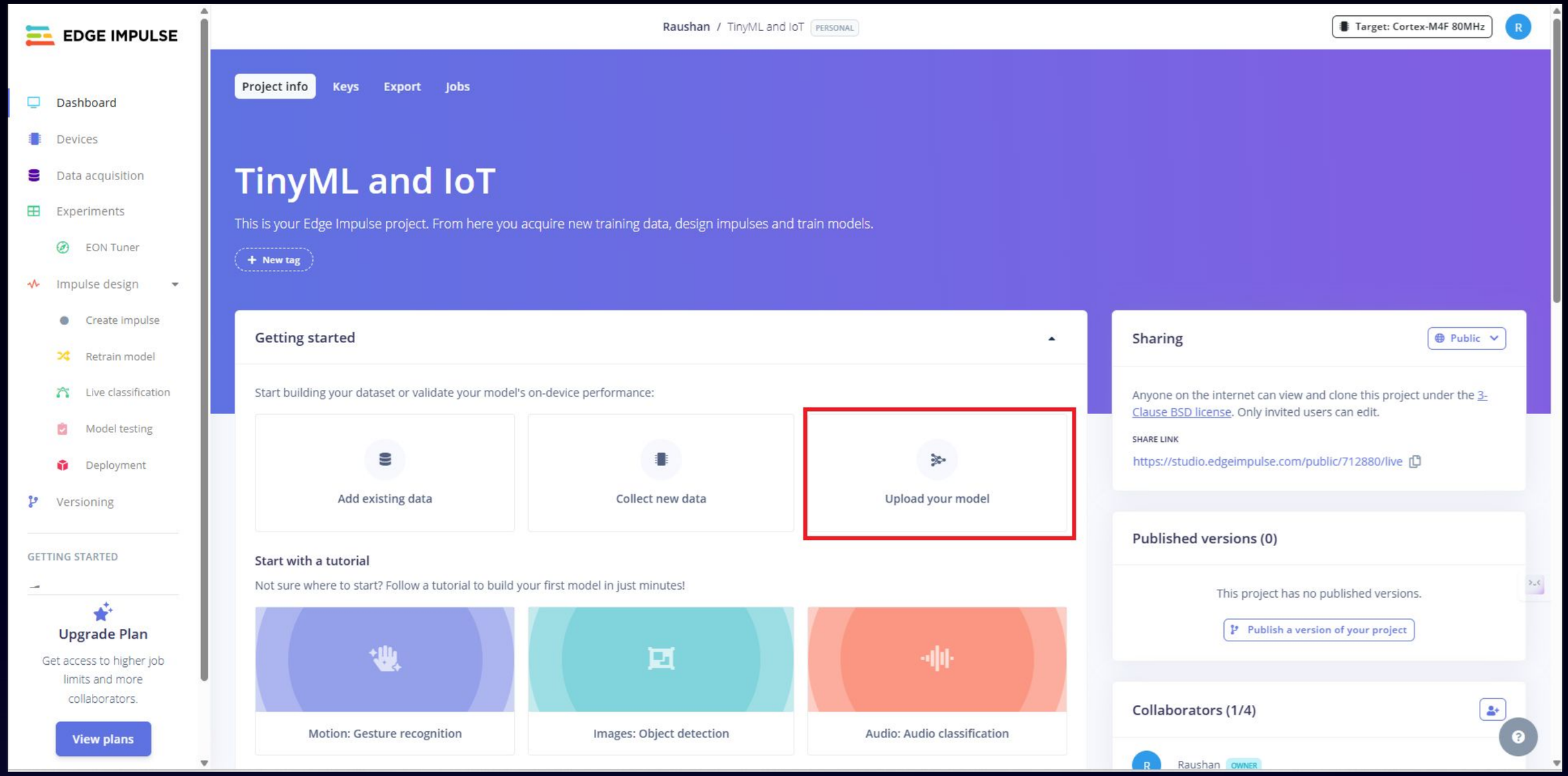
Step 7: Getting Started



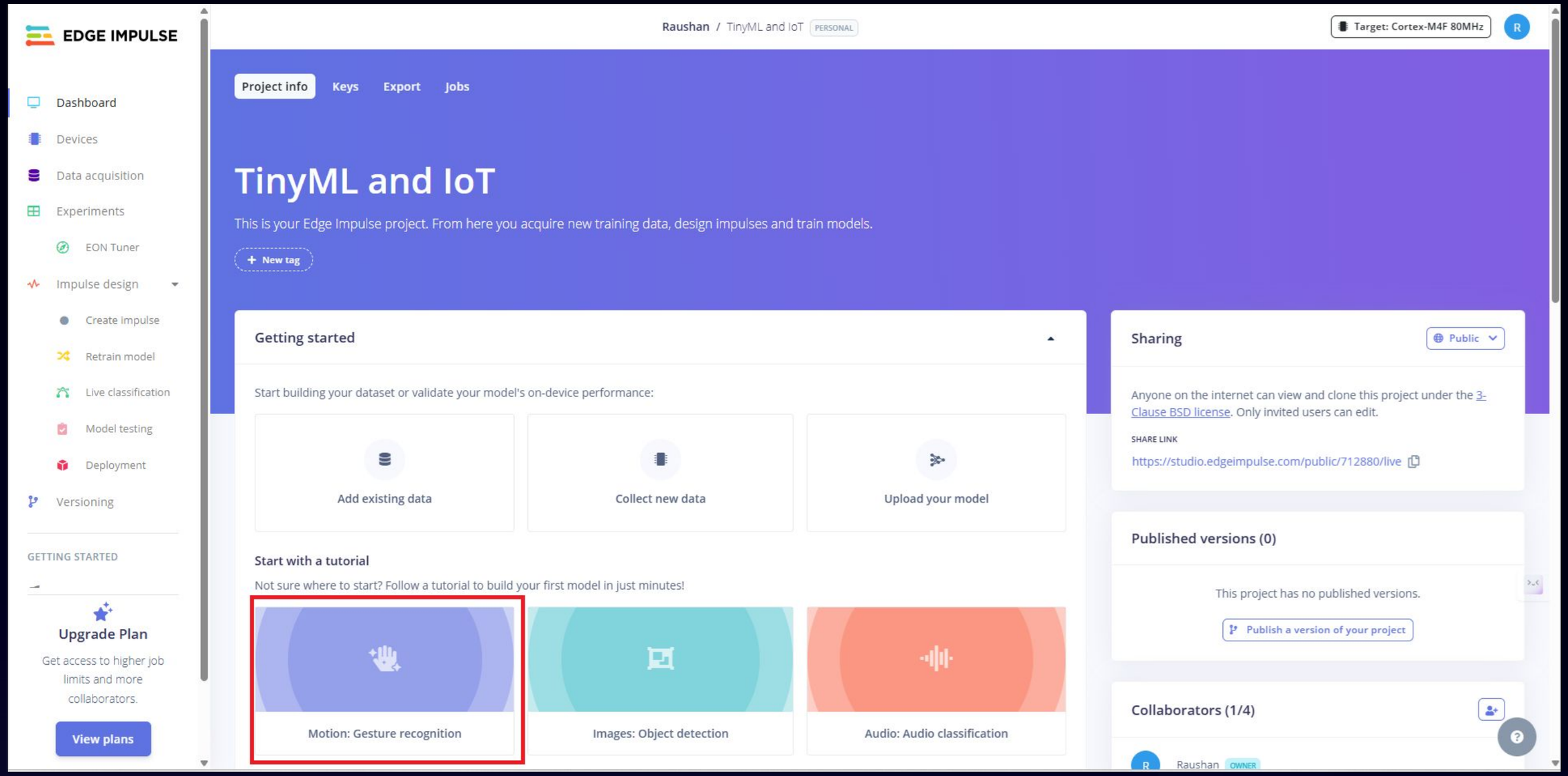
Step 8: Getting Started



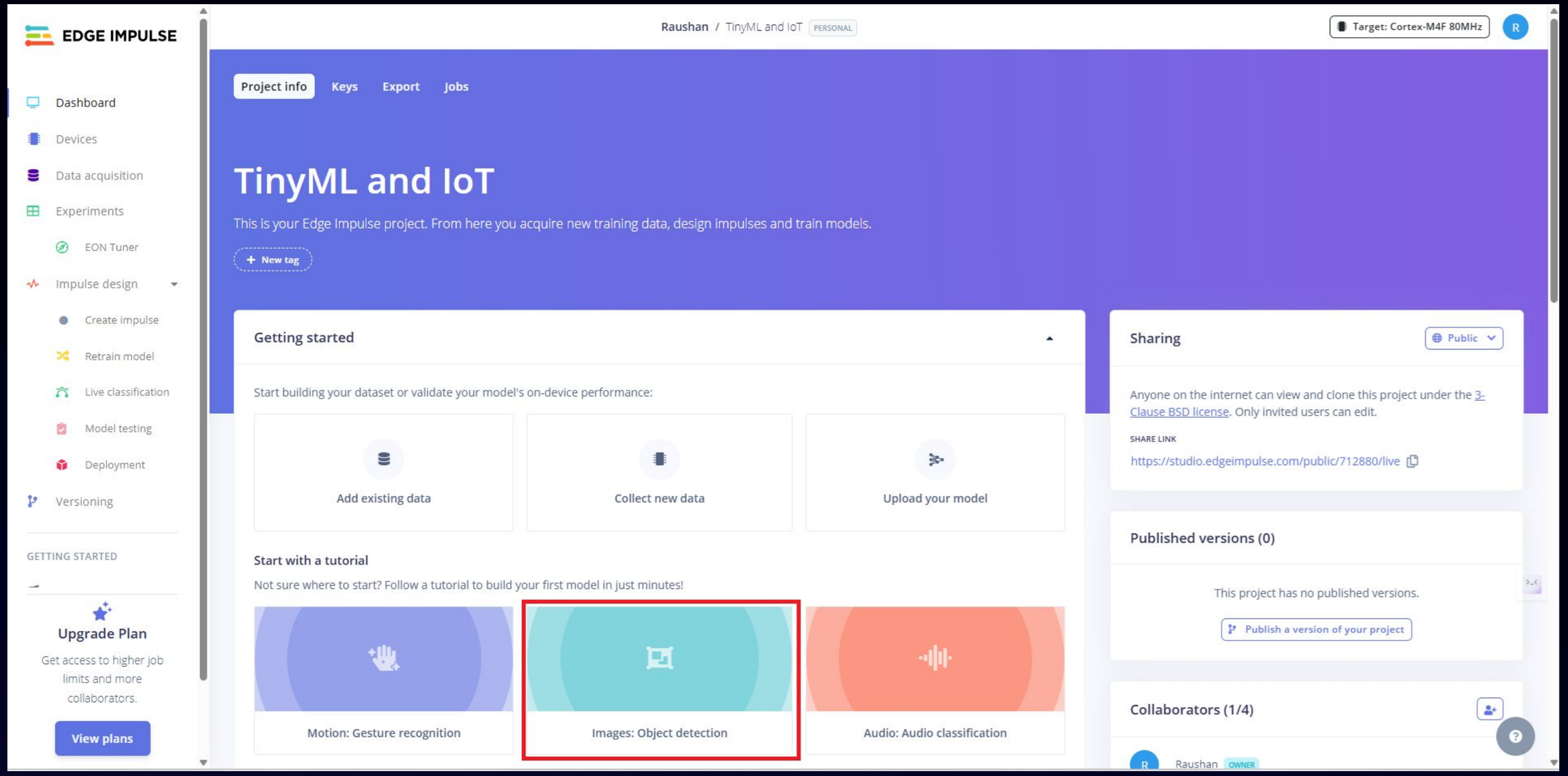
Step 9: Getting Started



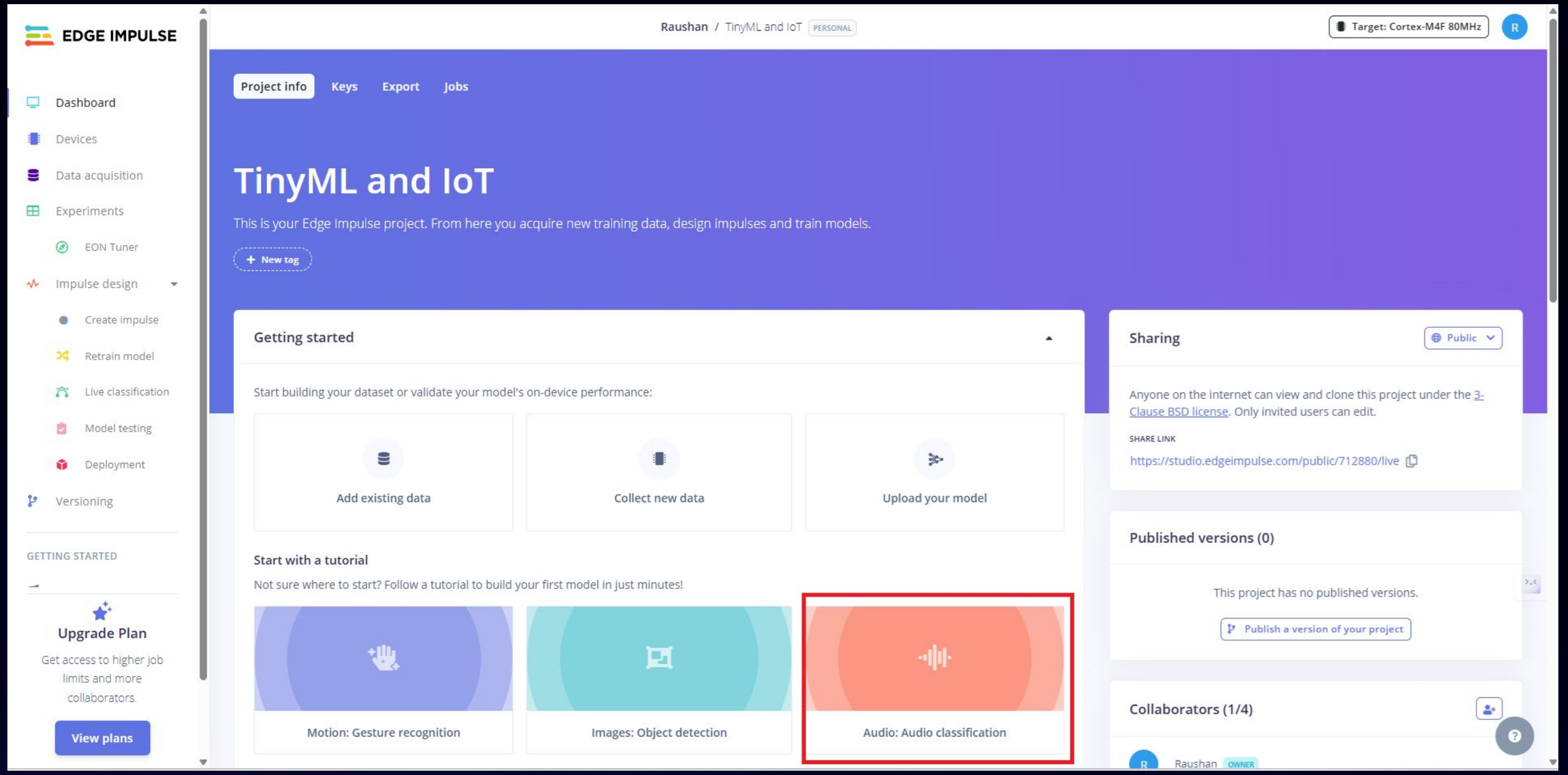
Step 10: Getting Started



Step 11: Getting Started



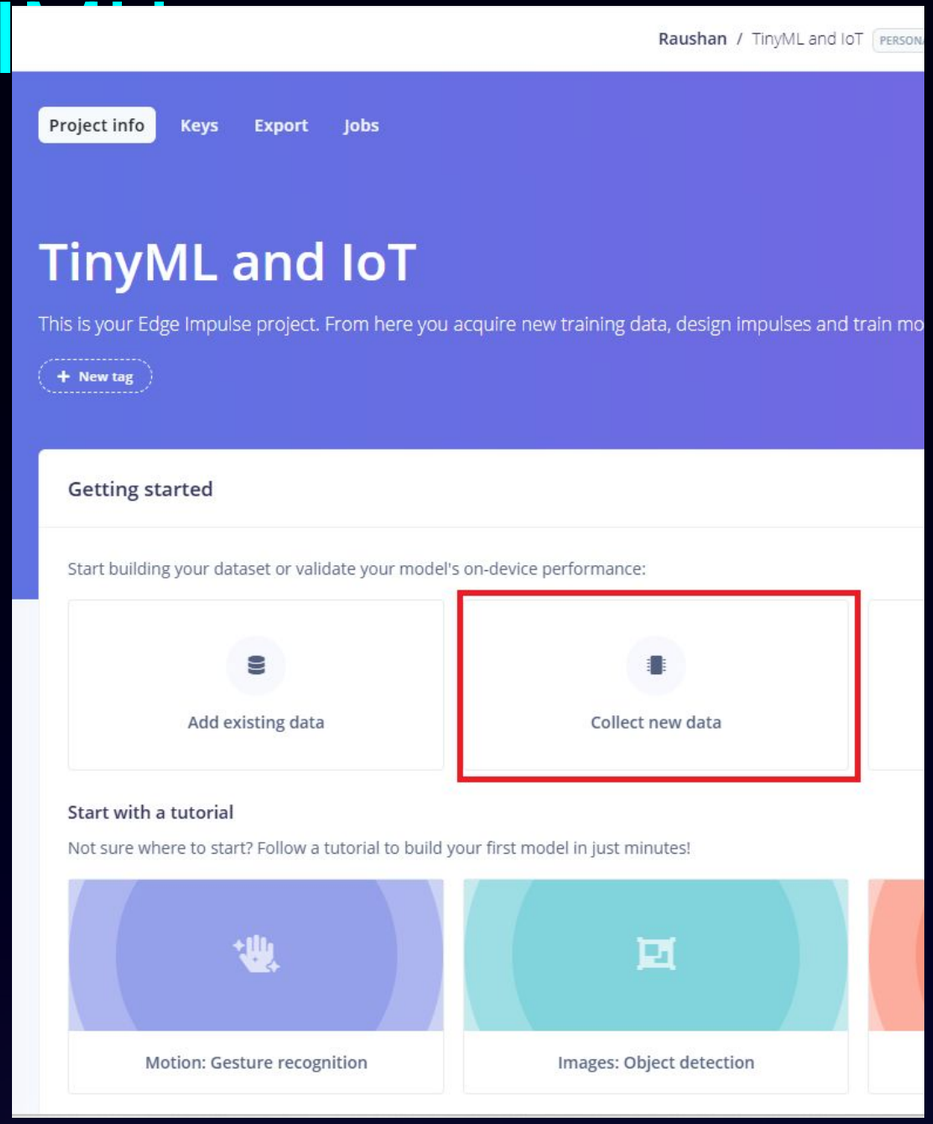
Step 12: Getting Started

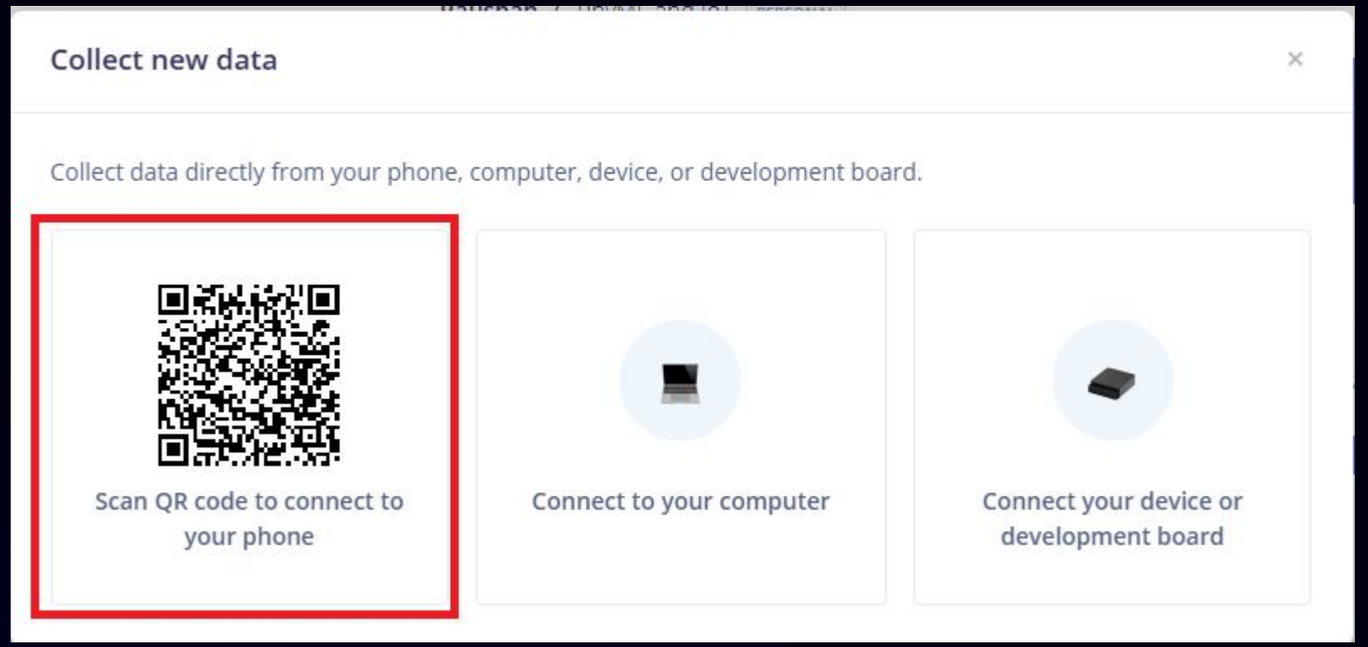


EDGE IMPULSE

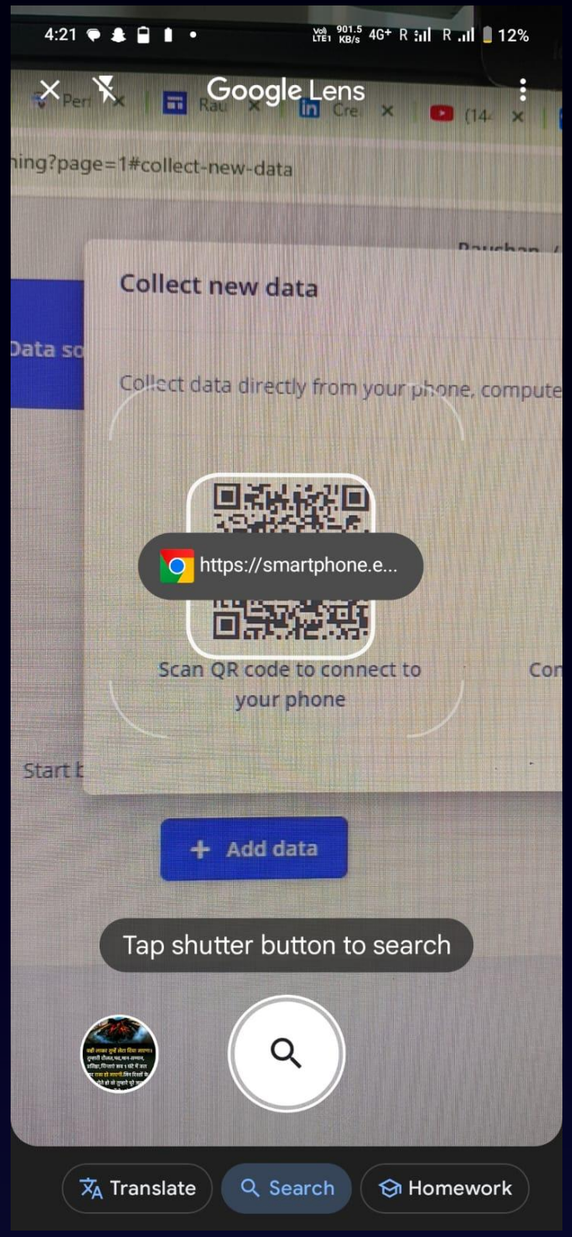
Data Collection

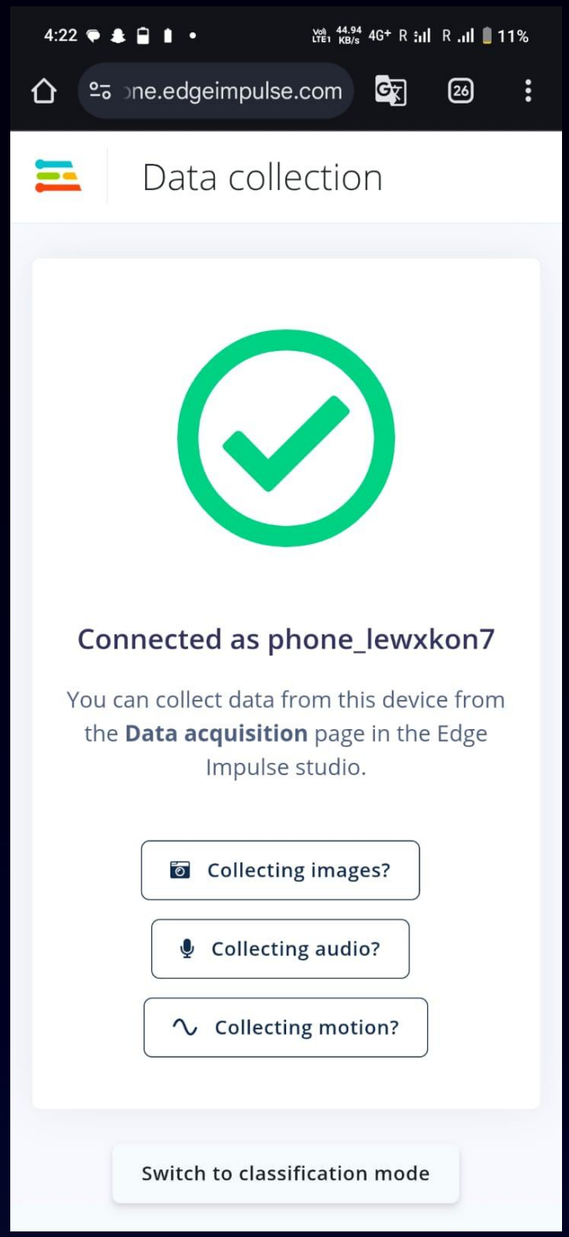
Step 13: Collect new data and Scan QR to use Phone



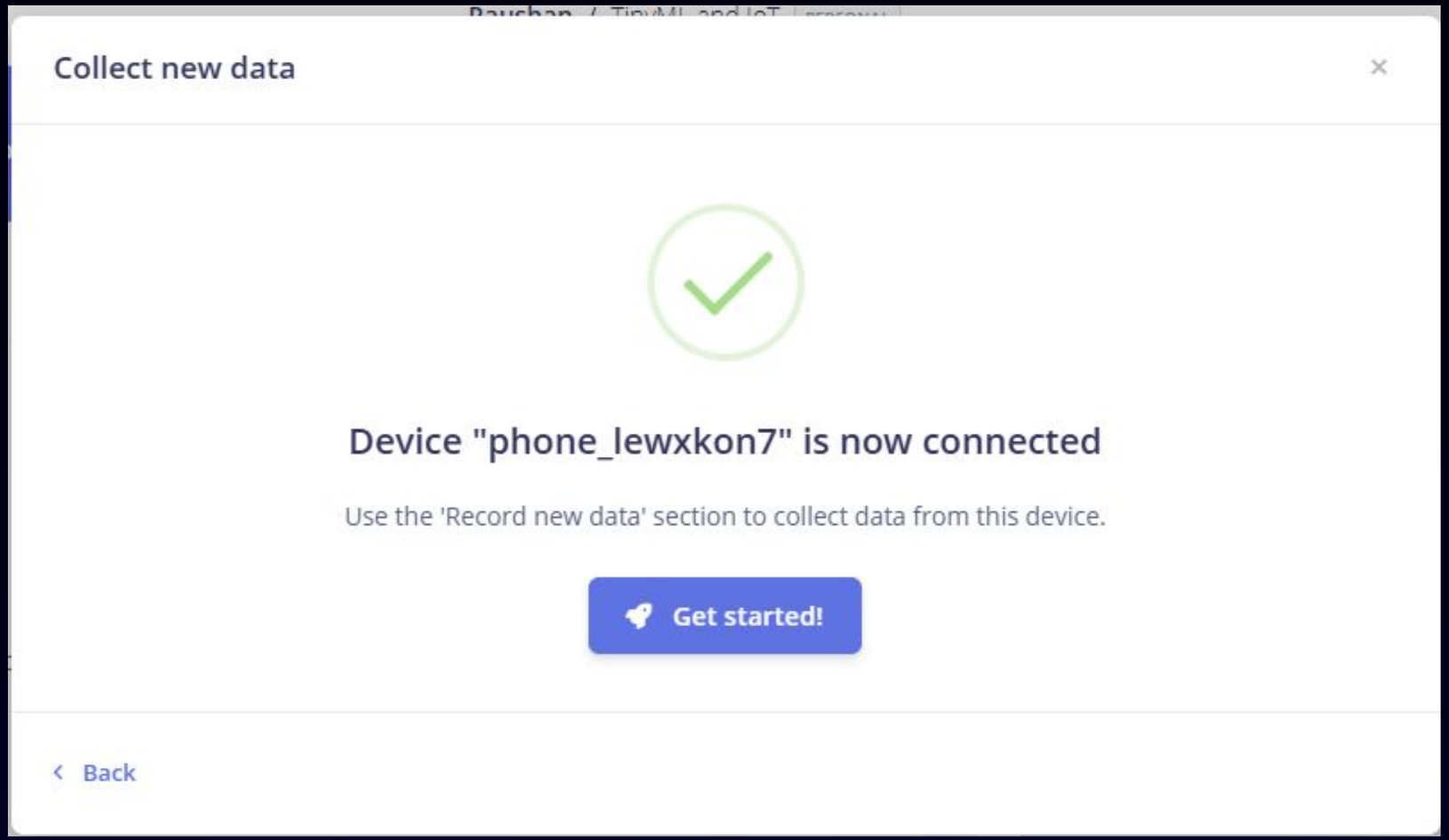


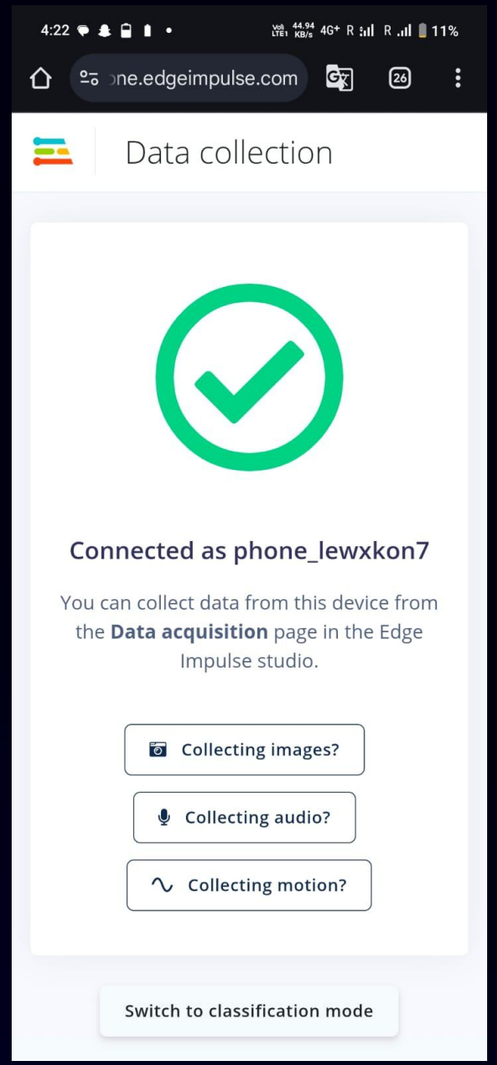
Step 14: Phone Screen: Scan QR in Google Lens and Cont.





Step 15: Create account on Edge Impulse Studio & Login

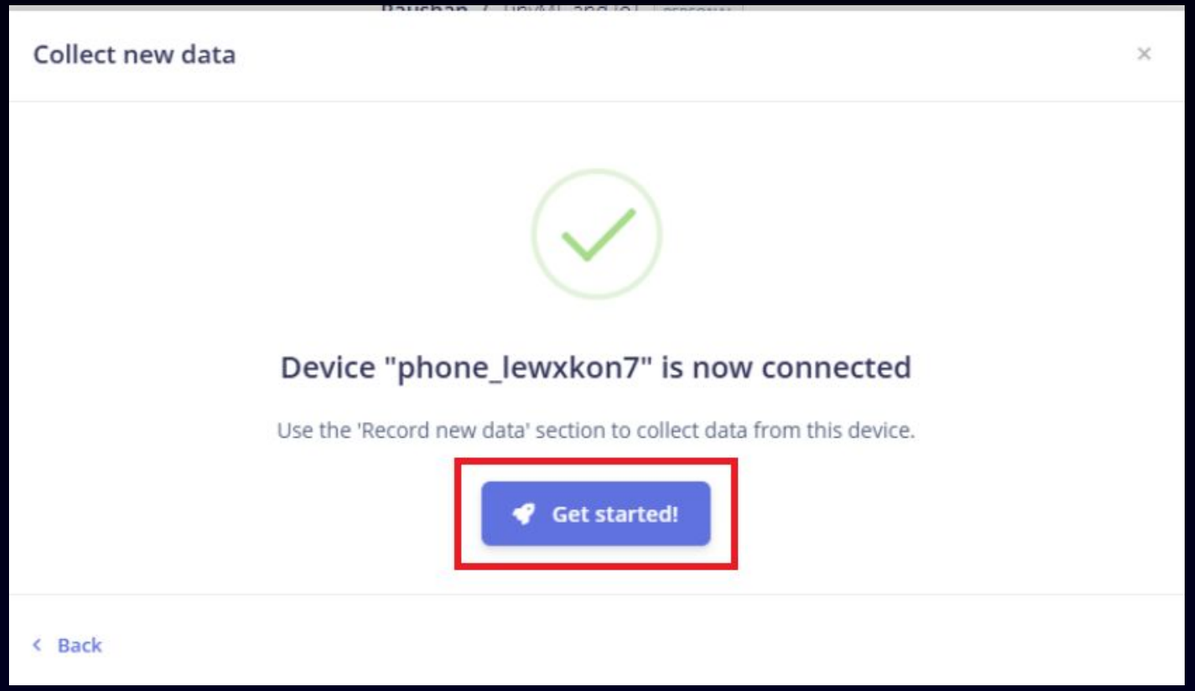




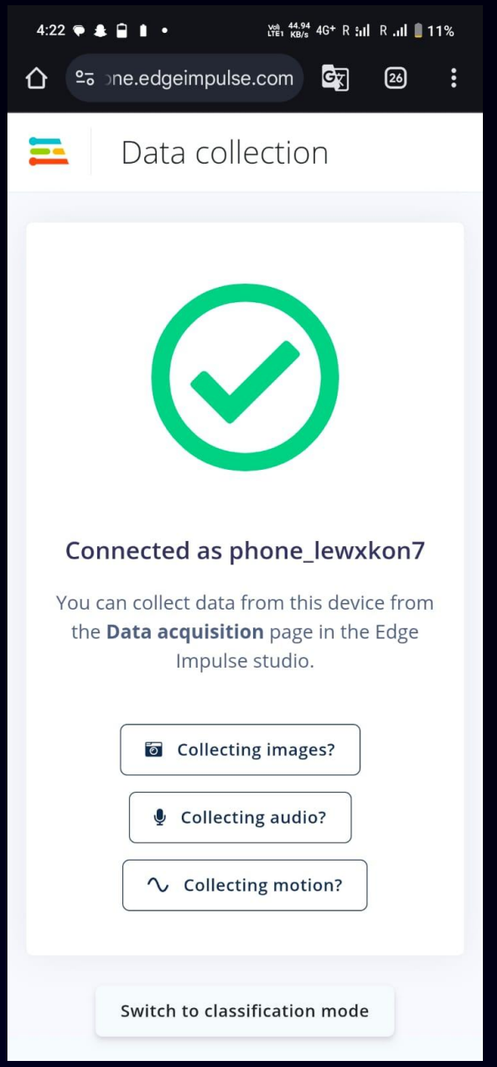
Computer Screen

Phone Screen

Step 16: Create account on Edge Impulse Studio & Login

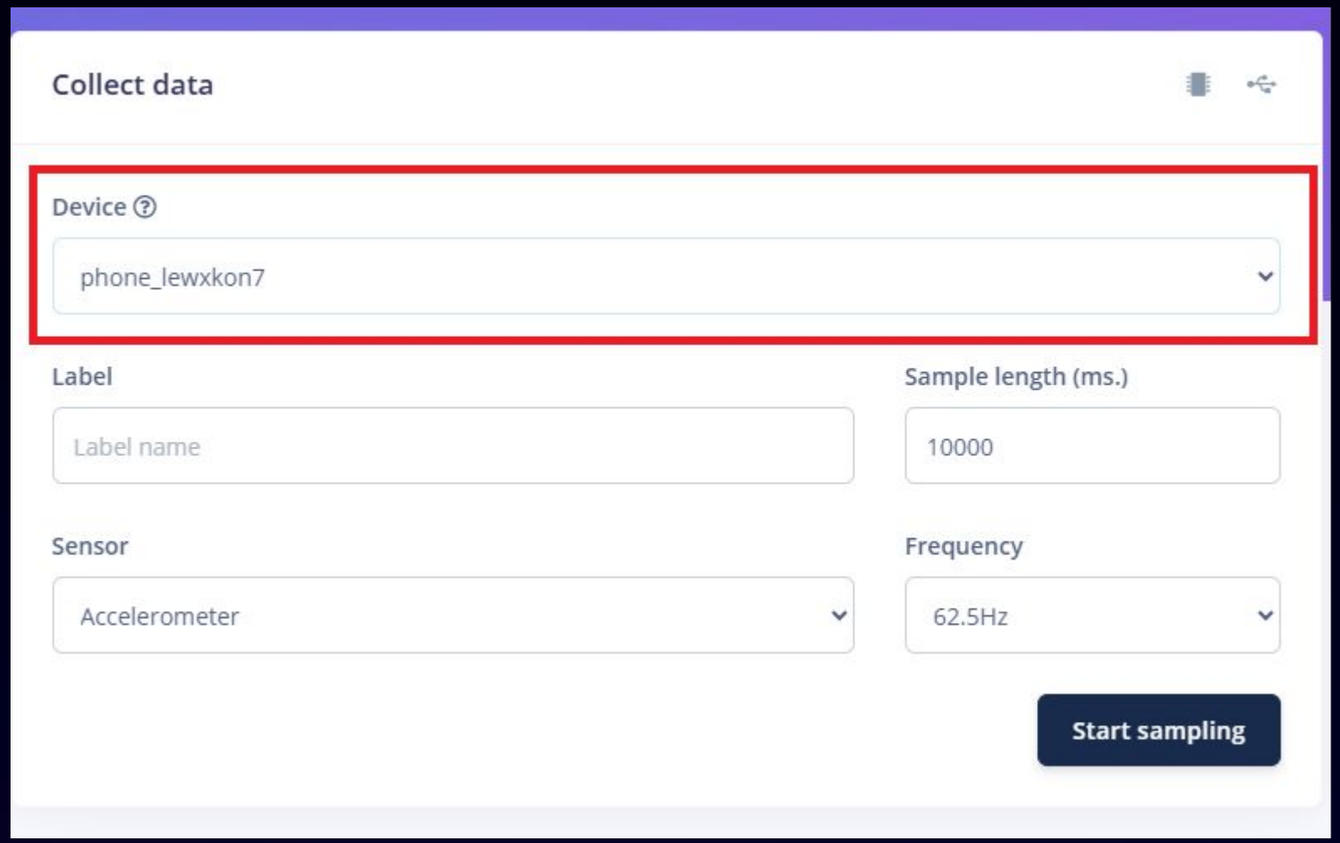


Computer Screen

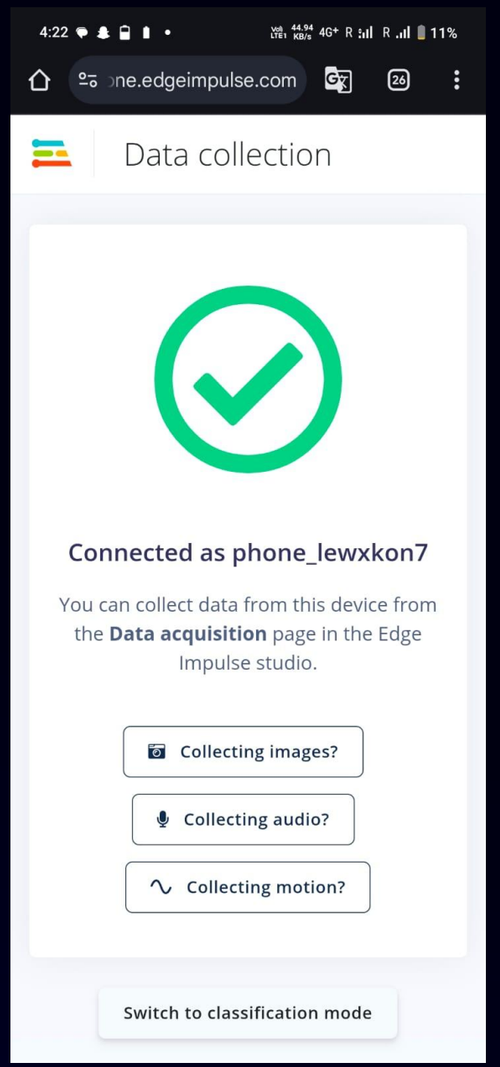


Phone Screen

Step 17: Verify both the phone and PC Screen



Computer Screen

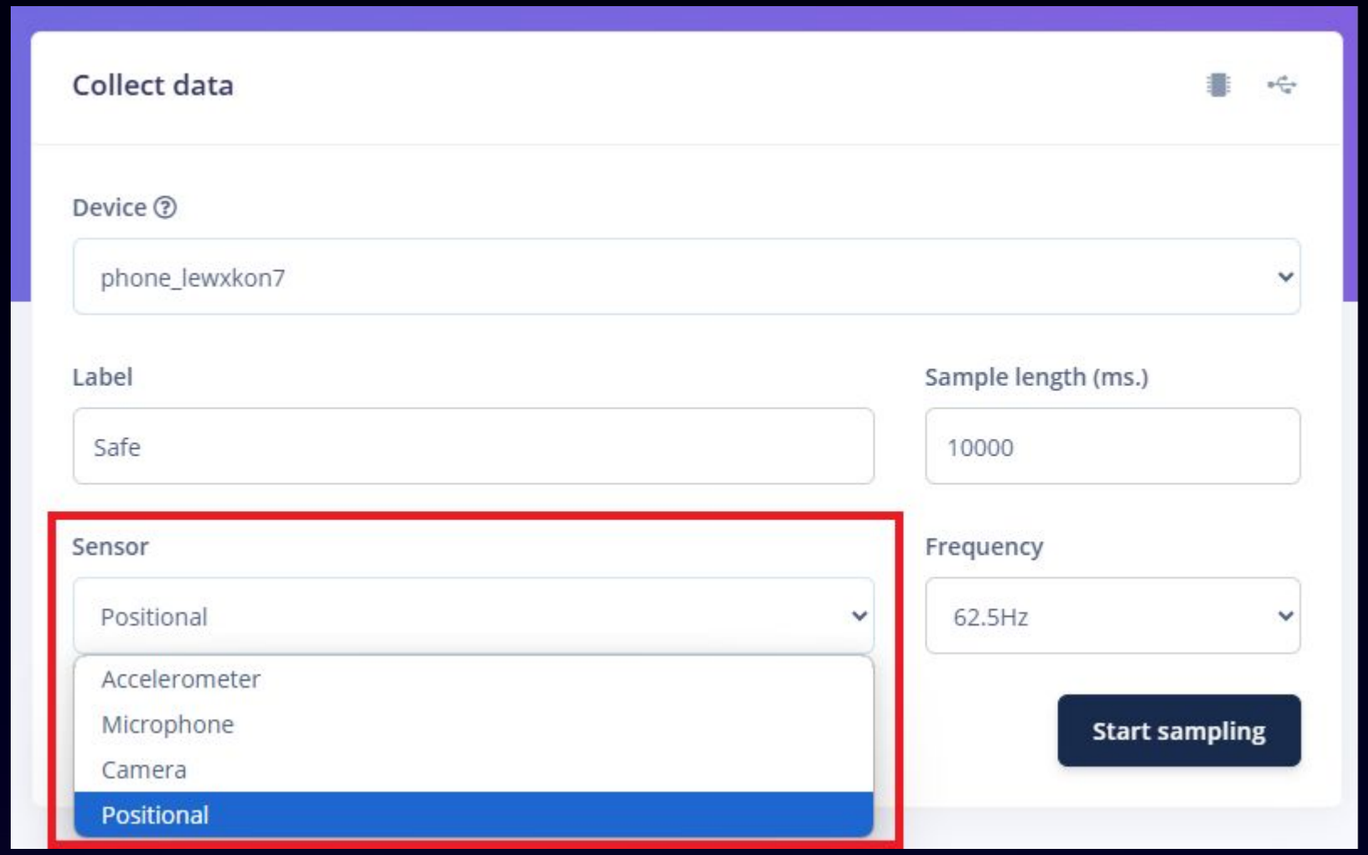


Phone Screen

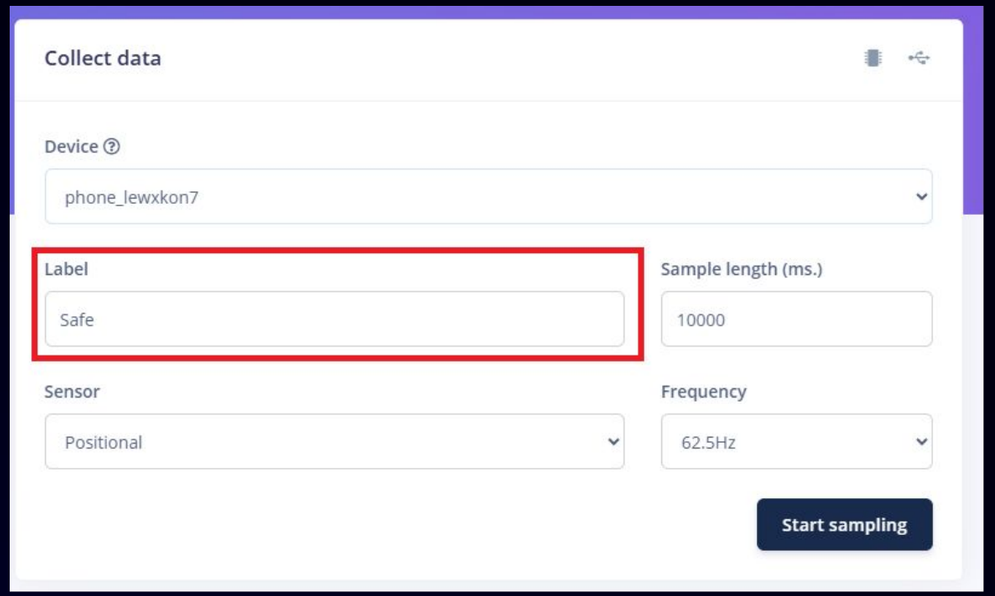
EDGE IMPULSE

Experiment 1: Fall Detection

Step 18: Verify both the phone and PC Screen

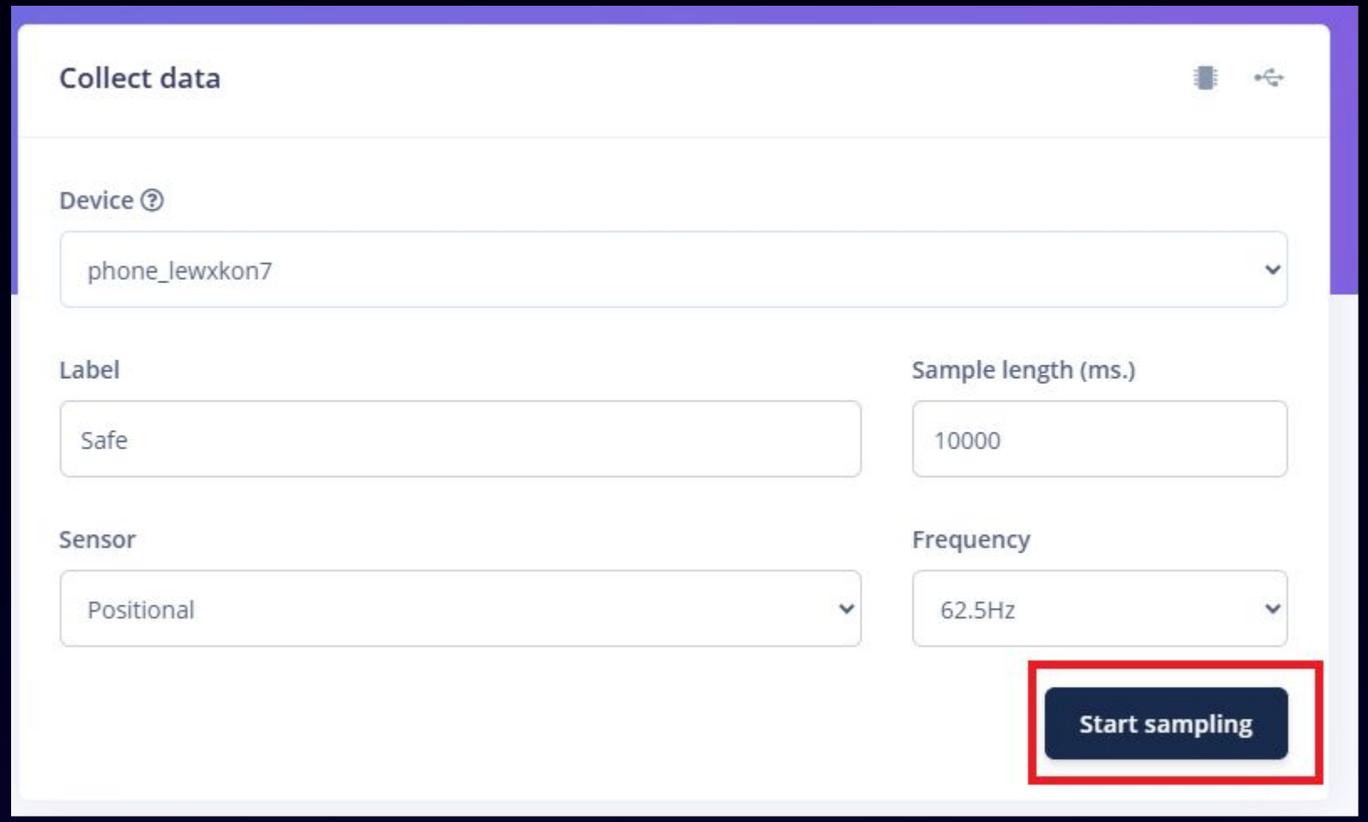


Select Sensor

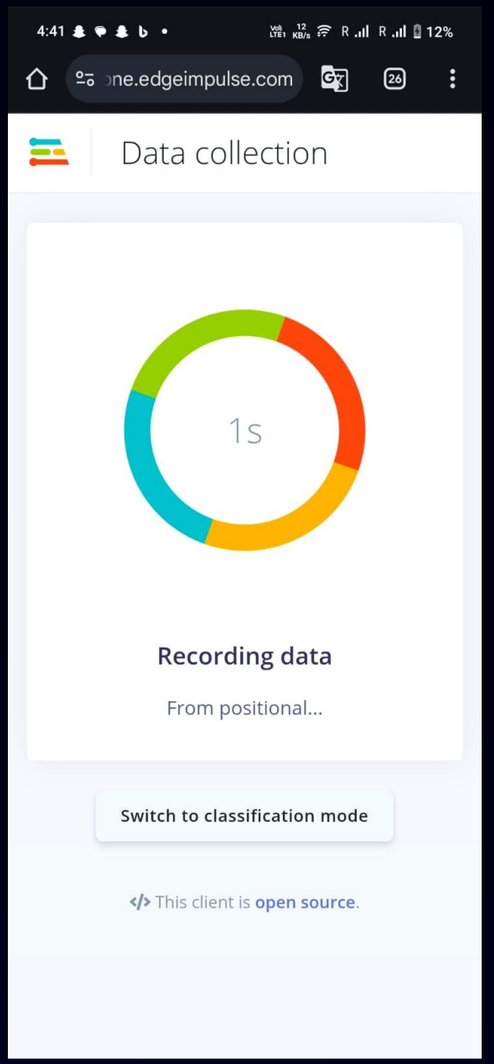


Label the Sample

Step 19: Verify both the phone and PC Screen

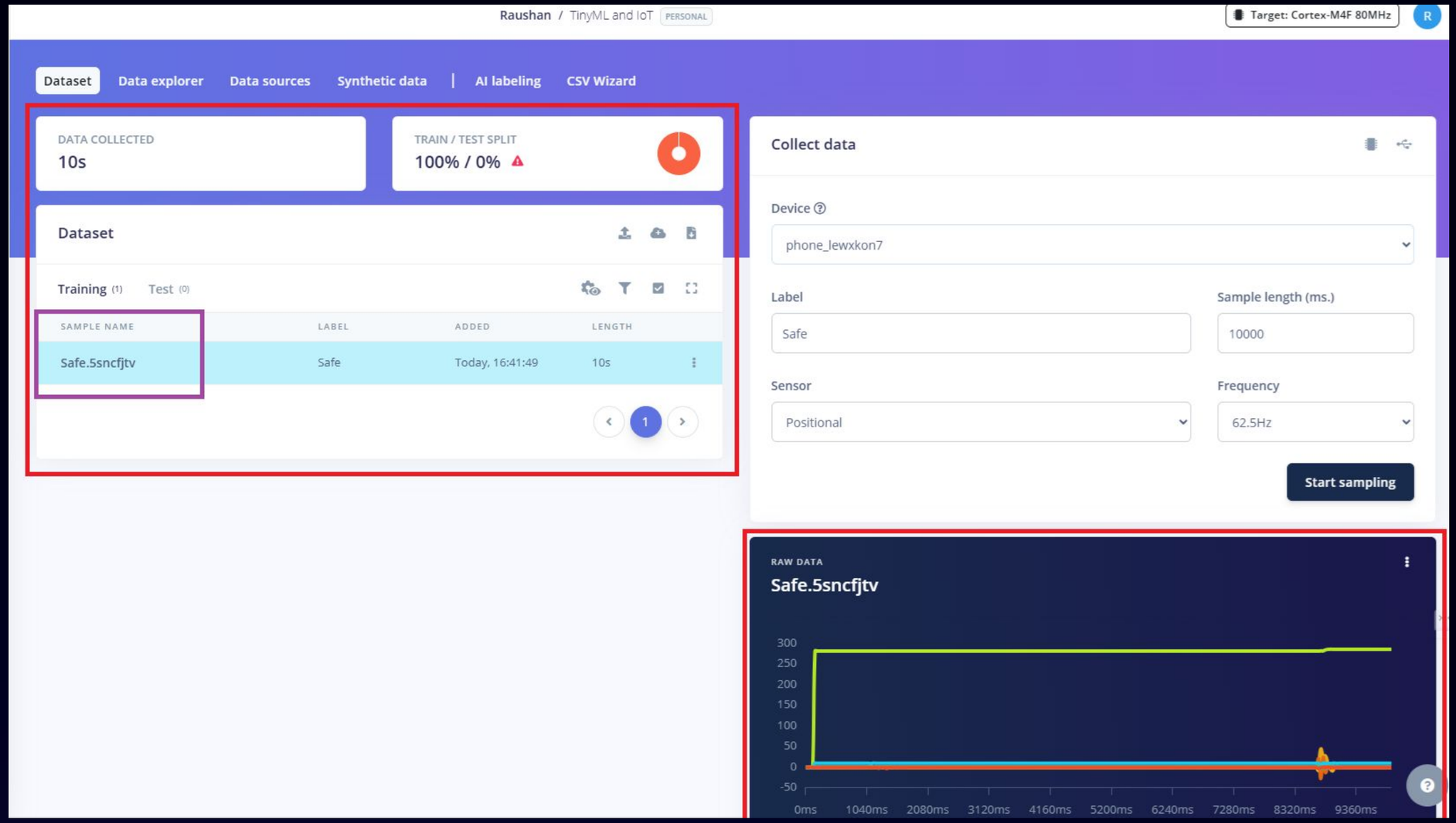


PC: Start Sampling

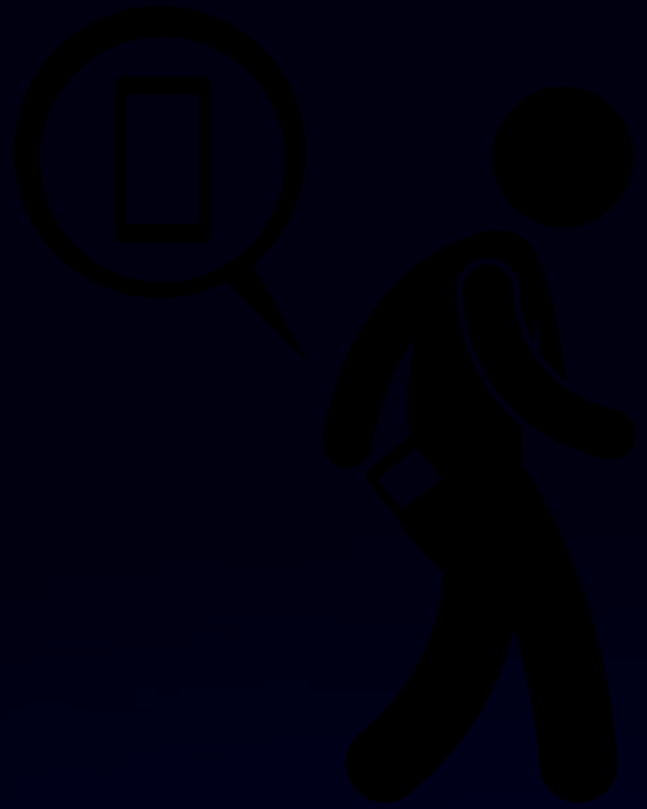


Phone: Record data

Step 20: Verify both the phone and PC Screen



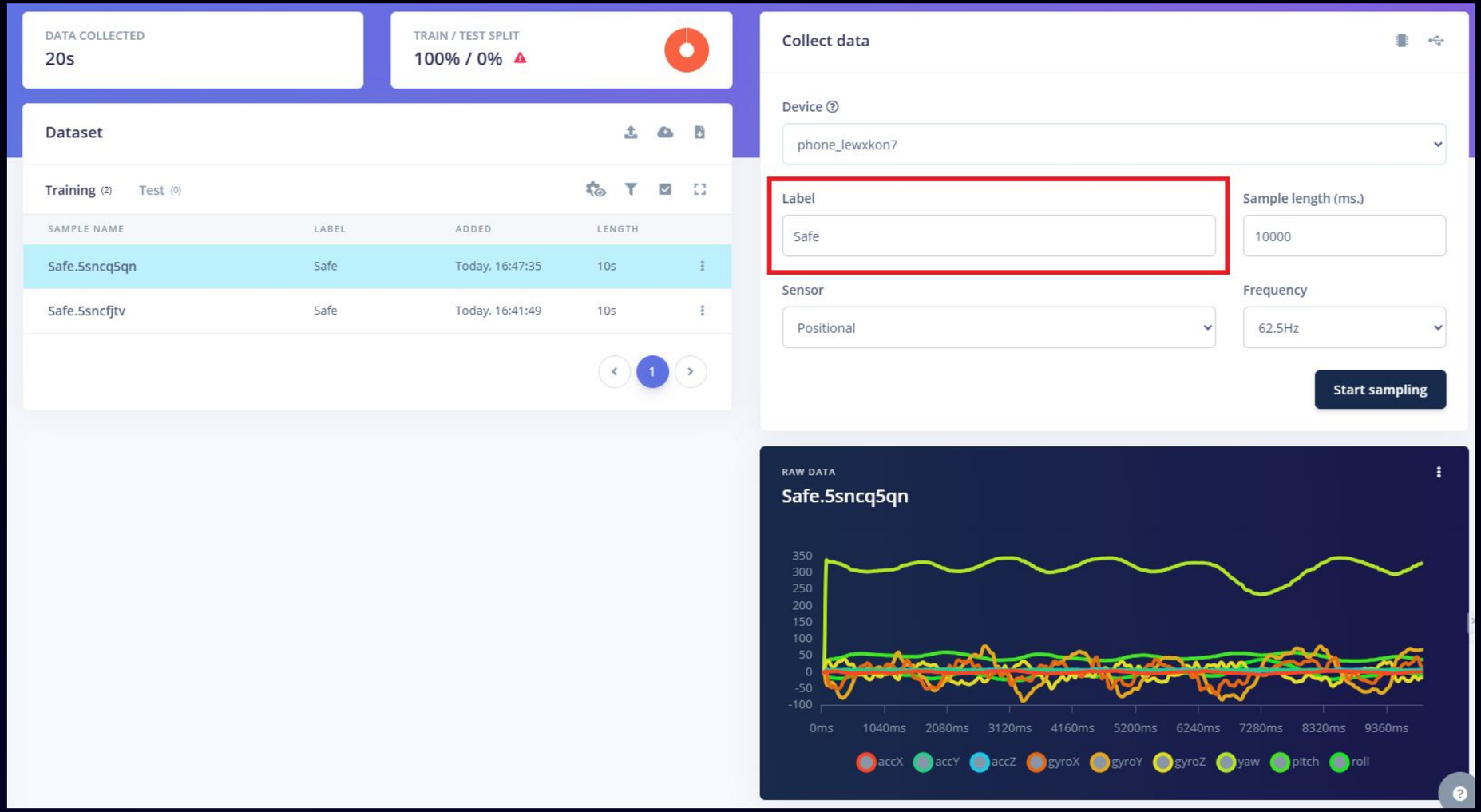
Data collection and Visualization



EDGE IMPULSE

Labelling: Safe

Step 21: Keep phone in pocket and do normal work



Change Label to “Safe”

Step 22: Keep doing that for around 30-40 samples

Dataset | Data explorer | Data sources | Synthetic data | AI labeling | CSV Wizard

DATA COLLECTED
3m 20s

TRAIN / TEST SPLIT
100% / 0%

Dataset

Training (20) Test (0)

SAMPLE NAME	LABEL	ADDED	LENGTH
Safe.5sndifjm	Safe	Today, 17:00:52	10s
Safe.5sndhvr0	Safe	Today, 17:00:36	10s
Safe.5sndhe01	Safe	Today, 17:00:17	10s
Safe.5sndh06c	Safe	Today, 17:00:03	10s
Safe.5sndgb3a	Safe	Today, 16:59:42	10s
Safe.5sndfm6q	Safe	Today, 16:59:20	10s
Safe.5sndf84t	Safe	Today, 16:59:06	10s
Safe.5sndeqfe	Safe	Today, 16:58:52	10s
Safe.5snddu3q	Safe	Today, 16:58:23	10s
Safe.5snddfu6	Safe	Today, 16:58:08	10s
Safe.5sndcoav	Safe	Today, 16:57:44	10s

Collect data

Device
phone_lewxkon7

Label
Safe

Sample length (ms.)
10000

Sensor
Positional

Frequency
62.5Hz

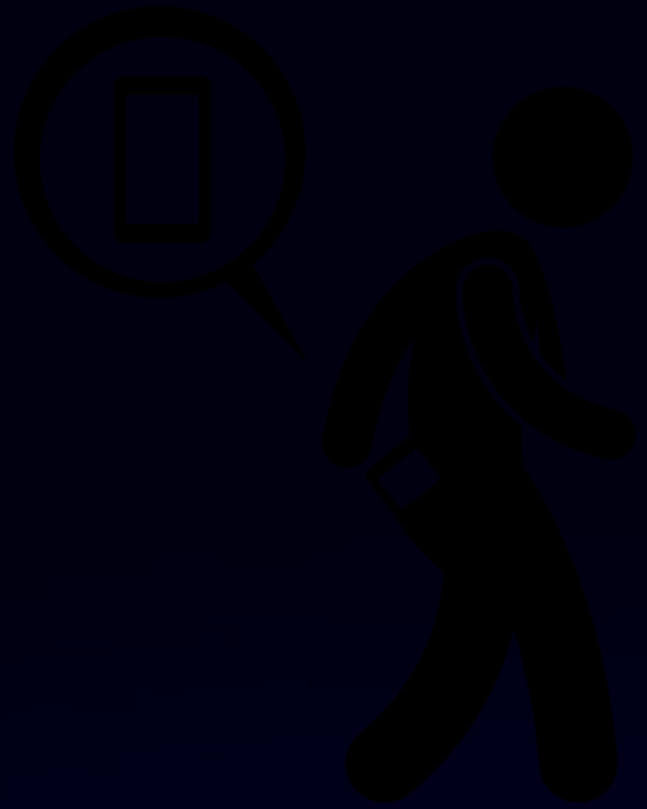
Start sampling

RAW DATA
Safe.5sndifjm

400
300
200
100
0
-100
-200

0ms 1040ms 2080ms 3120ms 4160ms 5200ms 6240ms 7280ms 8320ms 9360ms

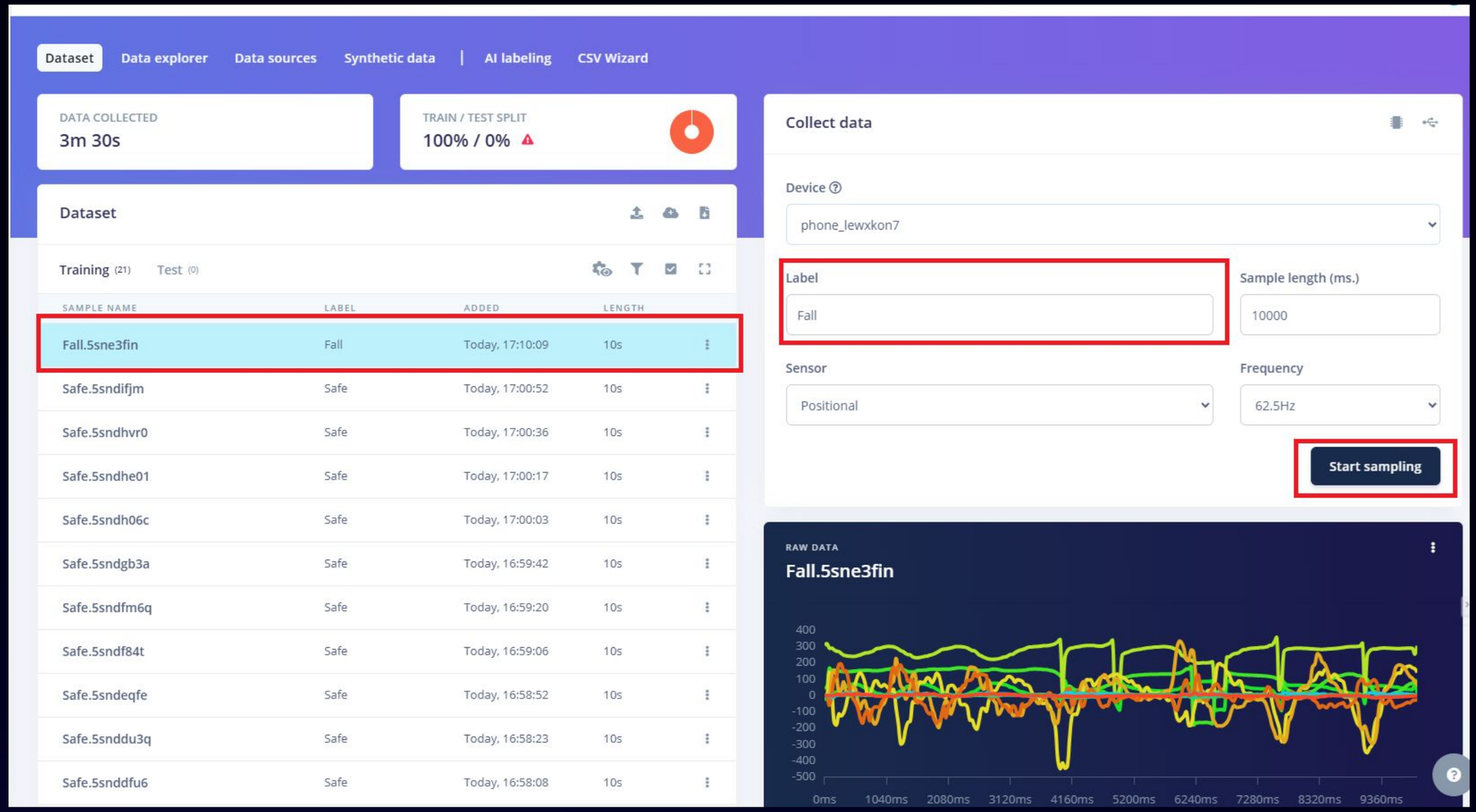
Data collection and Visualization



EDGE IMPULSE

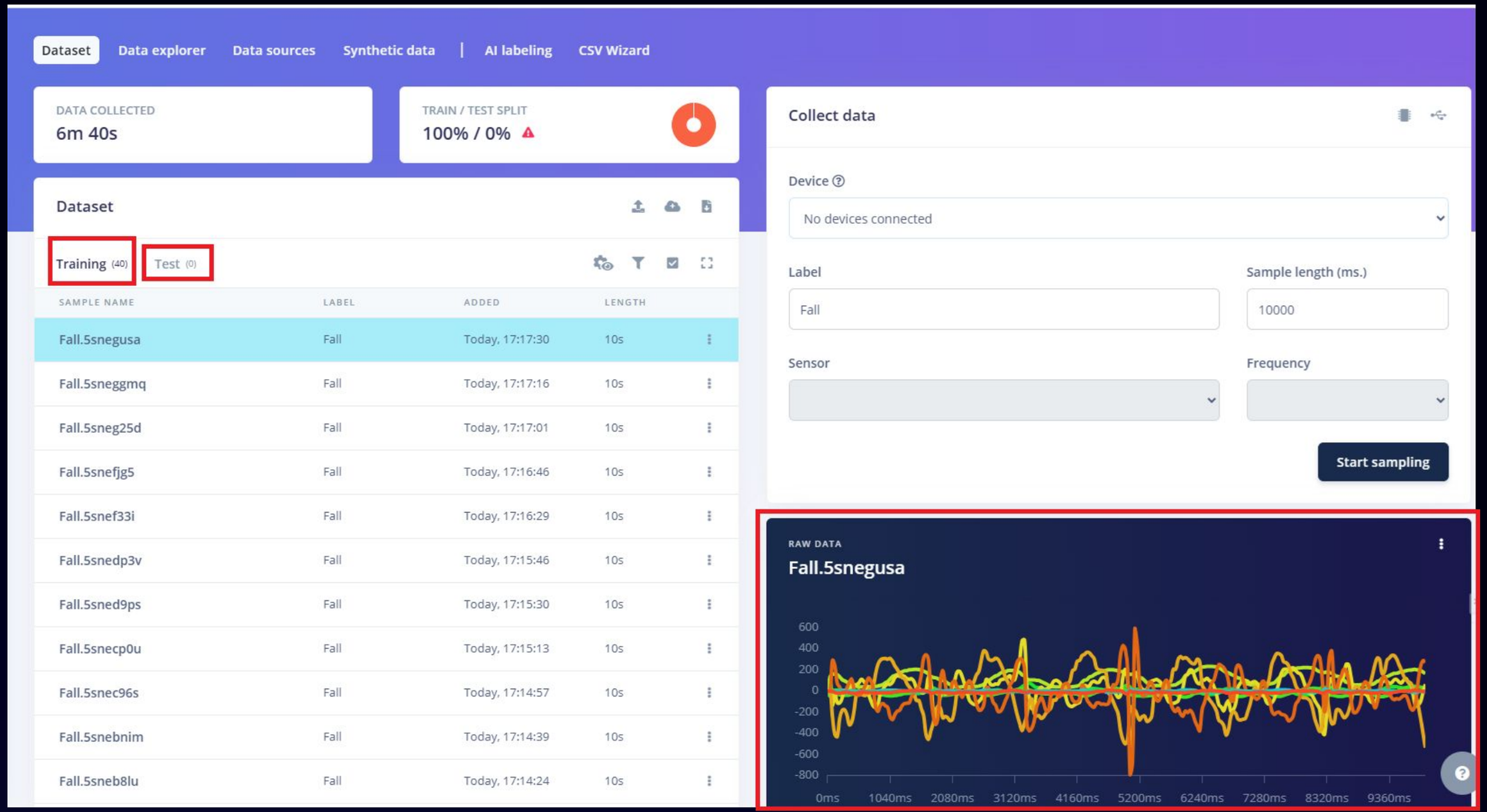
Labelling: Fall

Step 23: Keep collecting fall data.



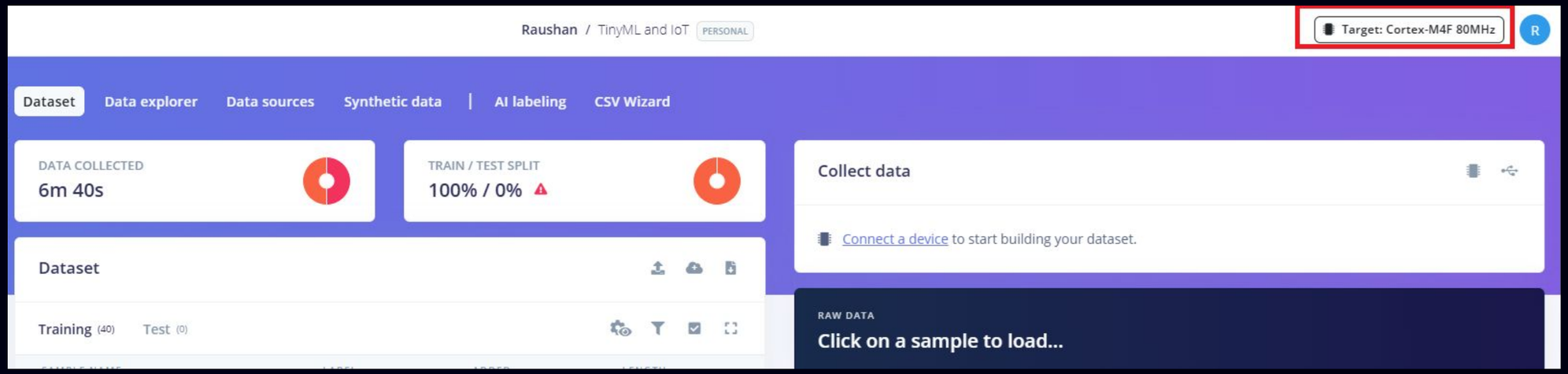
Change Label to “Fall”

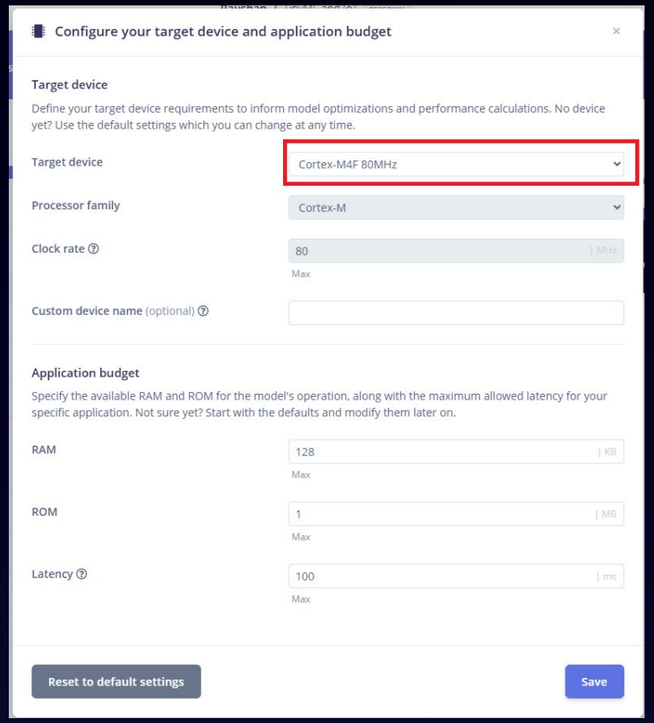
Step 24: Keep doing that for around 30-40 samples

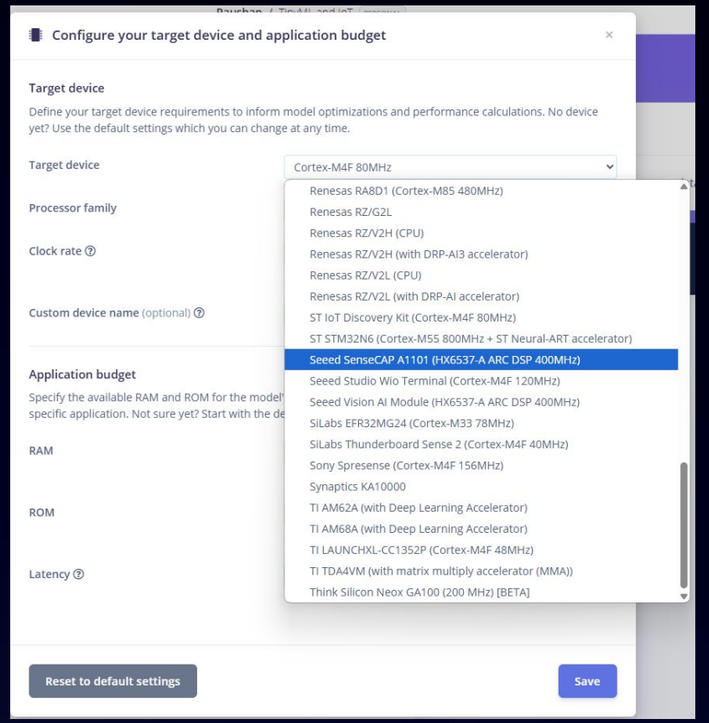


Change Label to “Fall”

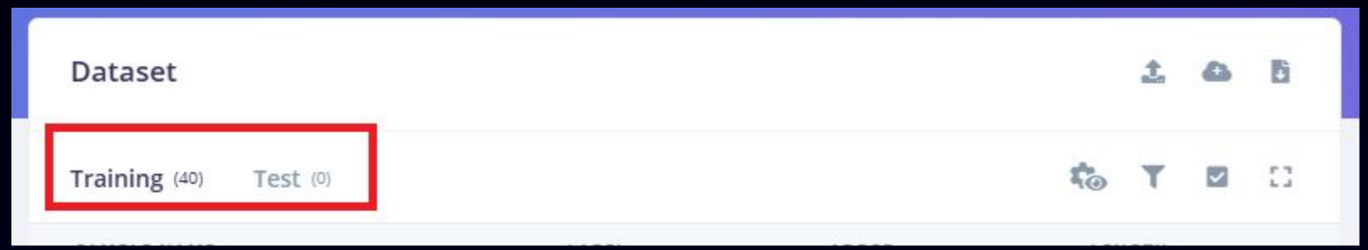
Step 25: Select the Model deployment tool

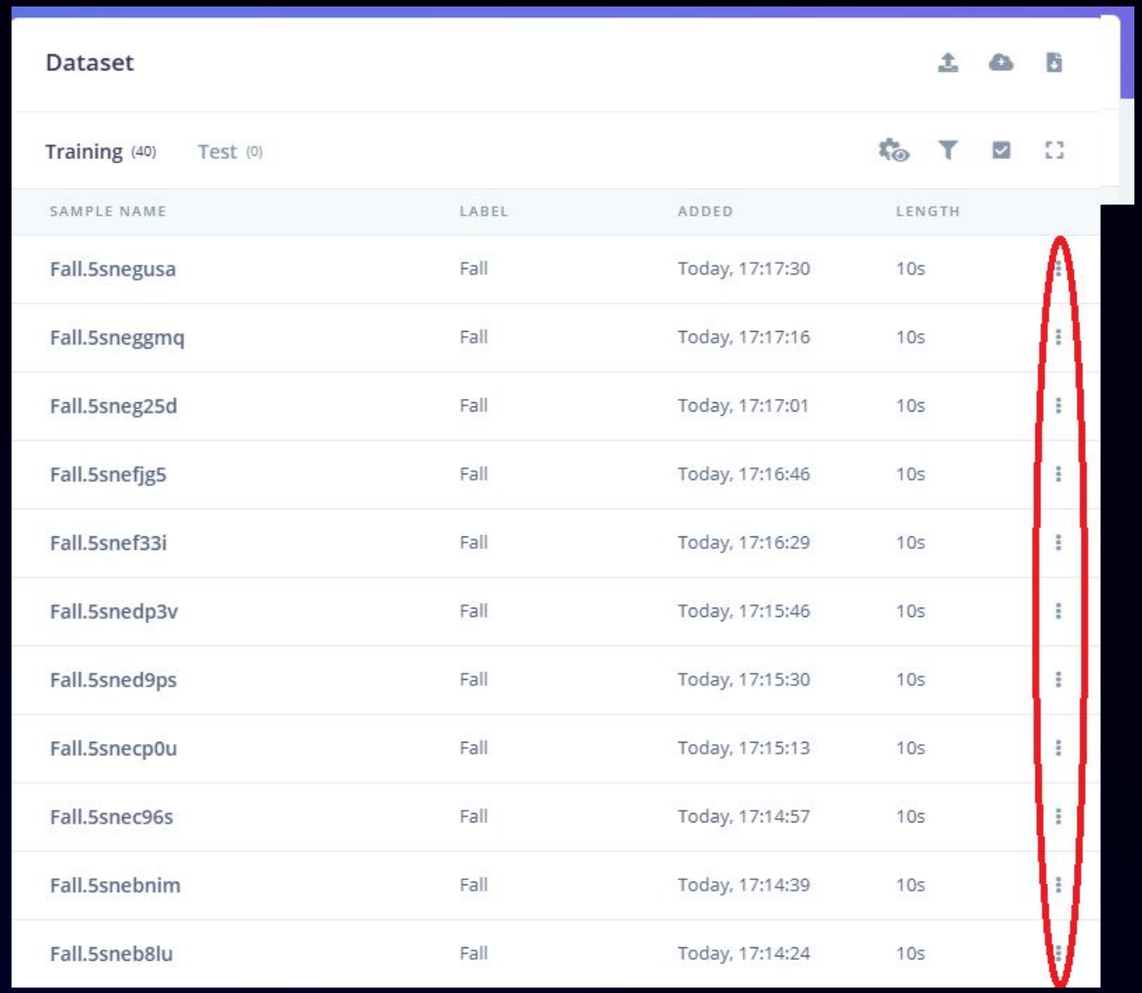




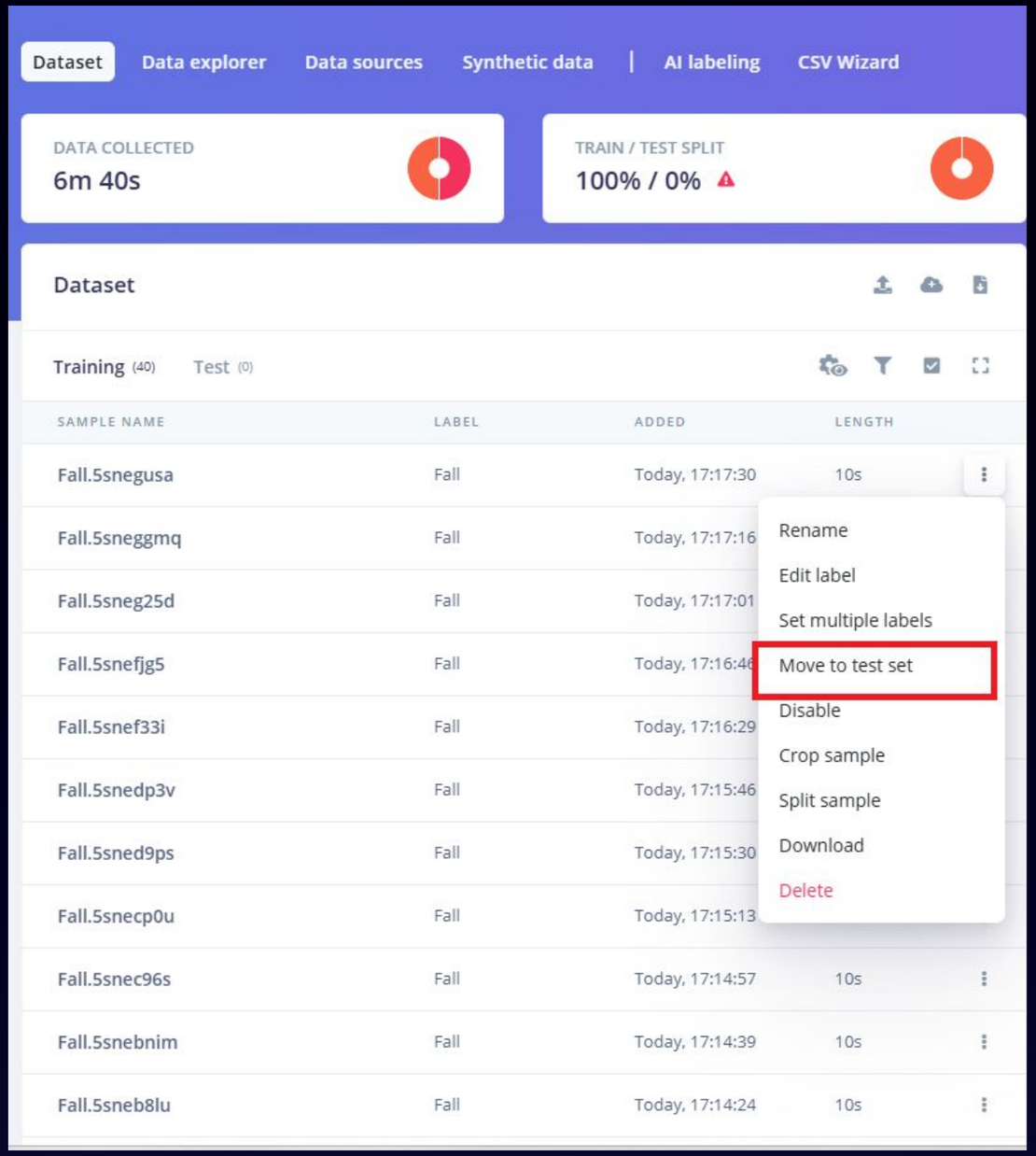


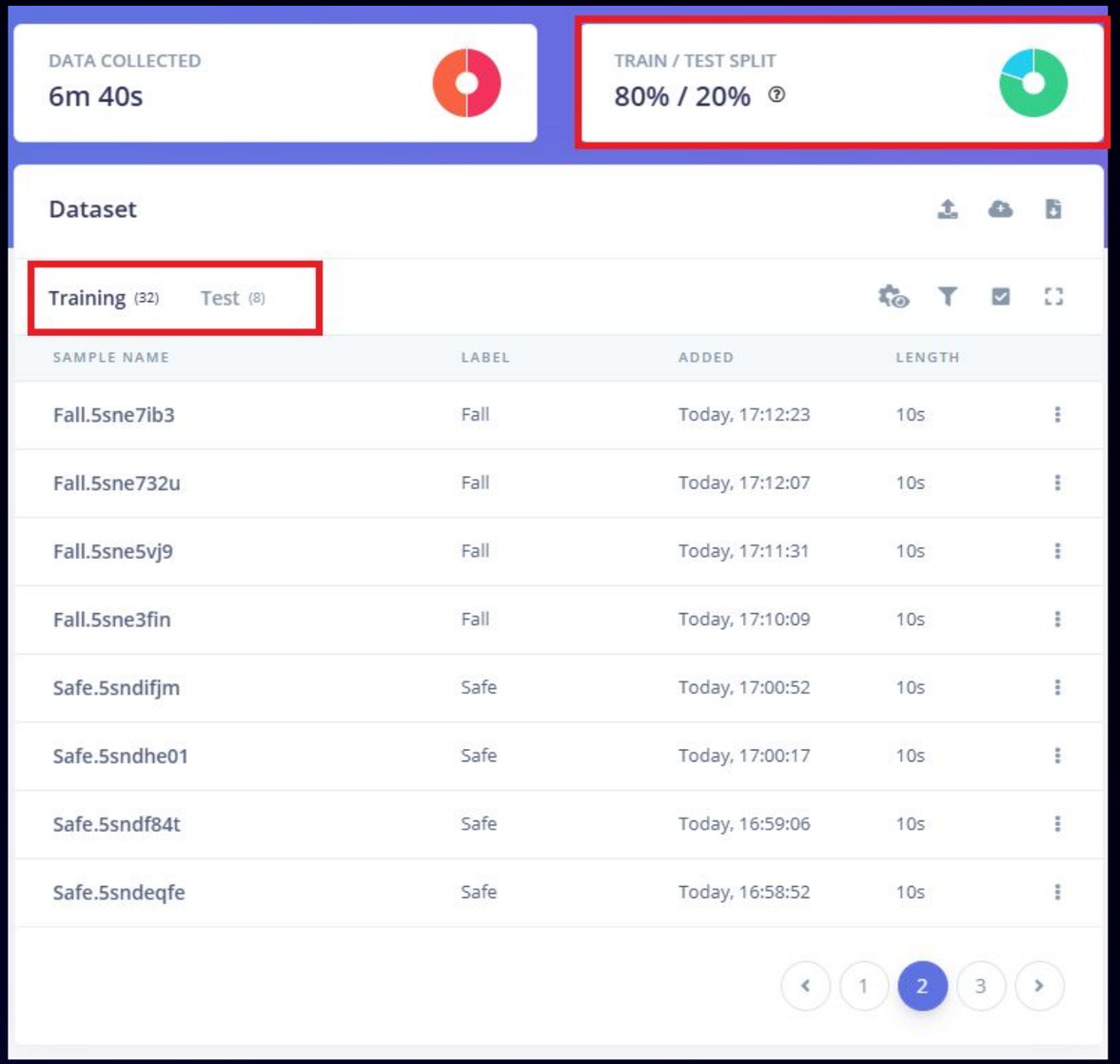
Step 26: Divide Training and Test sample to 100/20 %



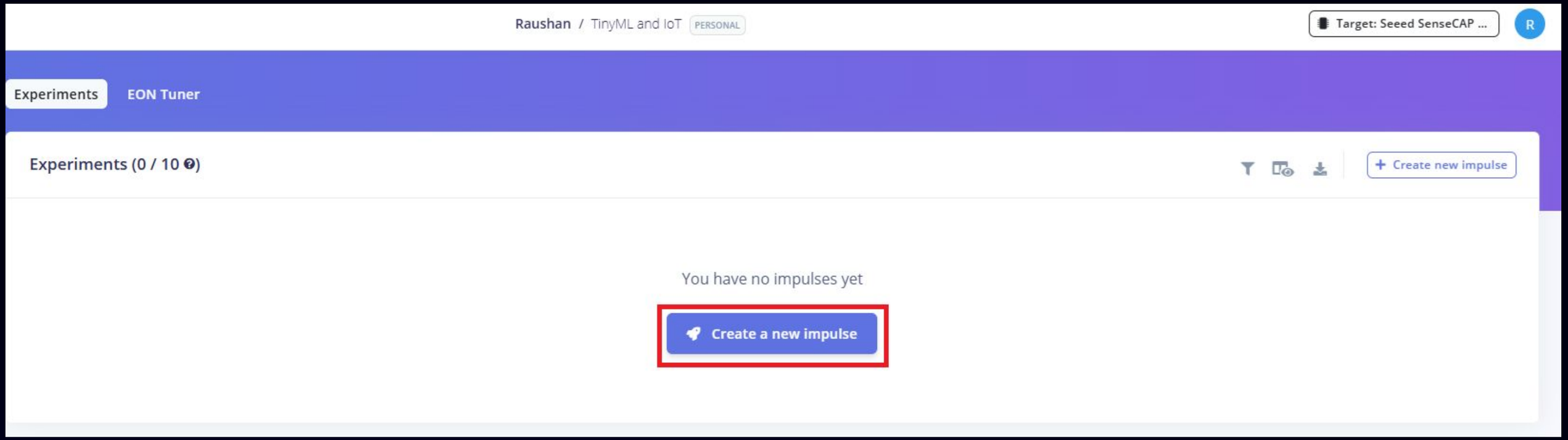


Step 27: Move 4 samples from safe & fall each to test

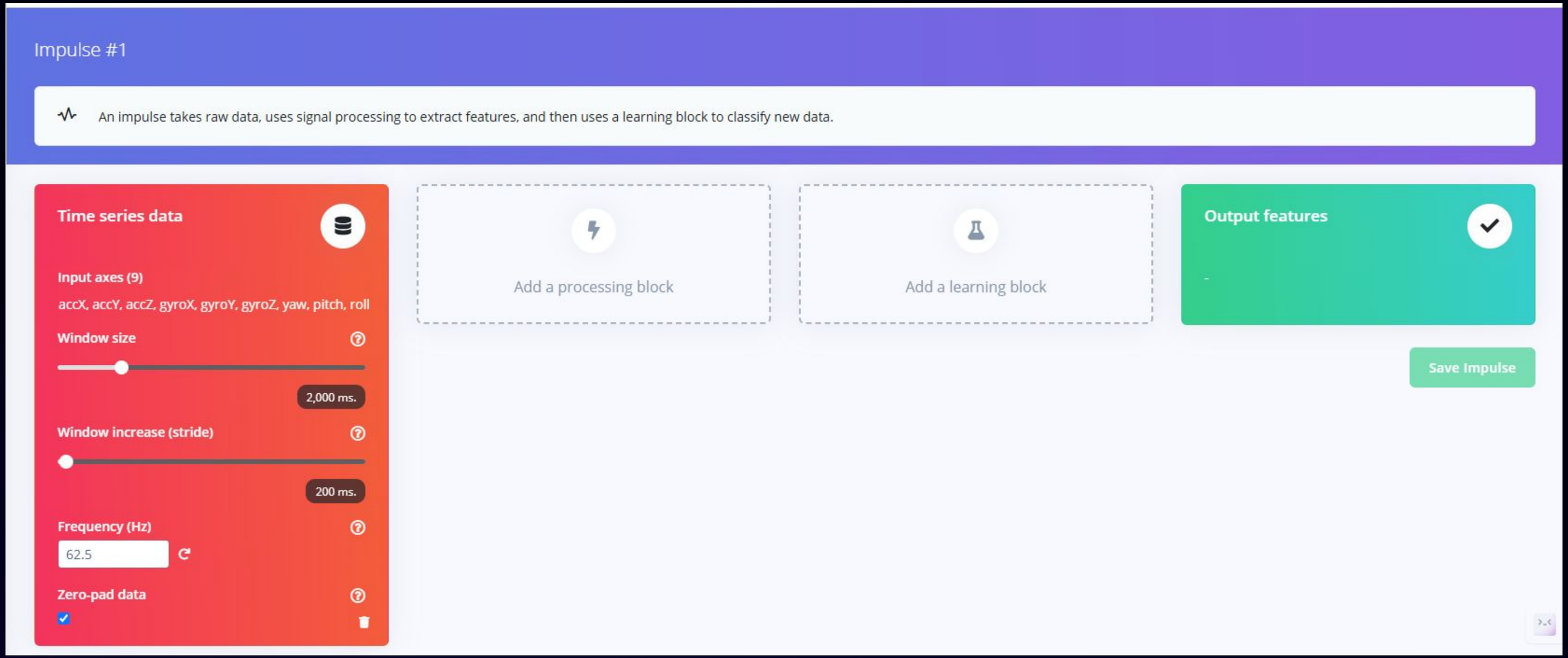




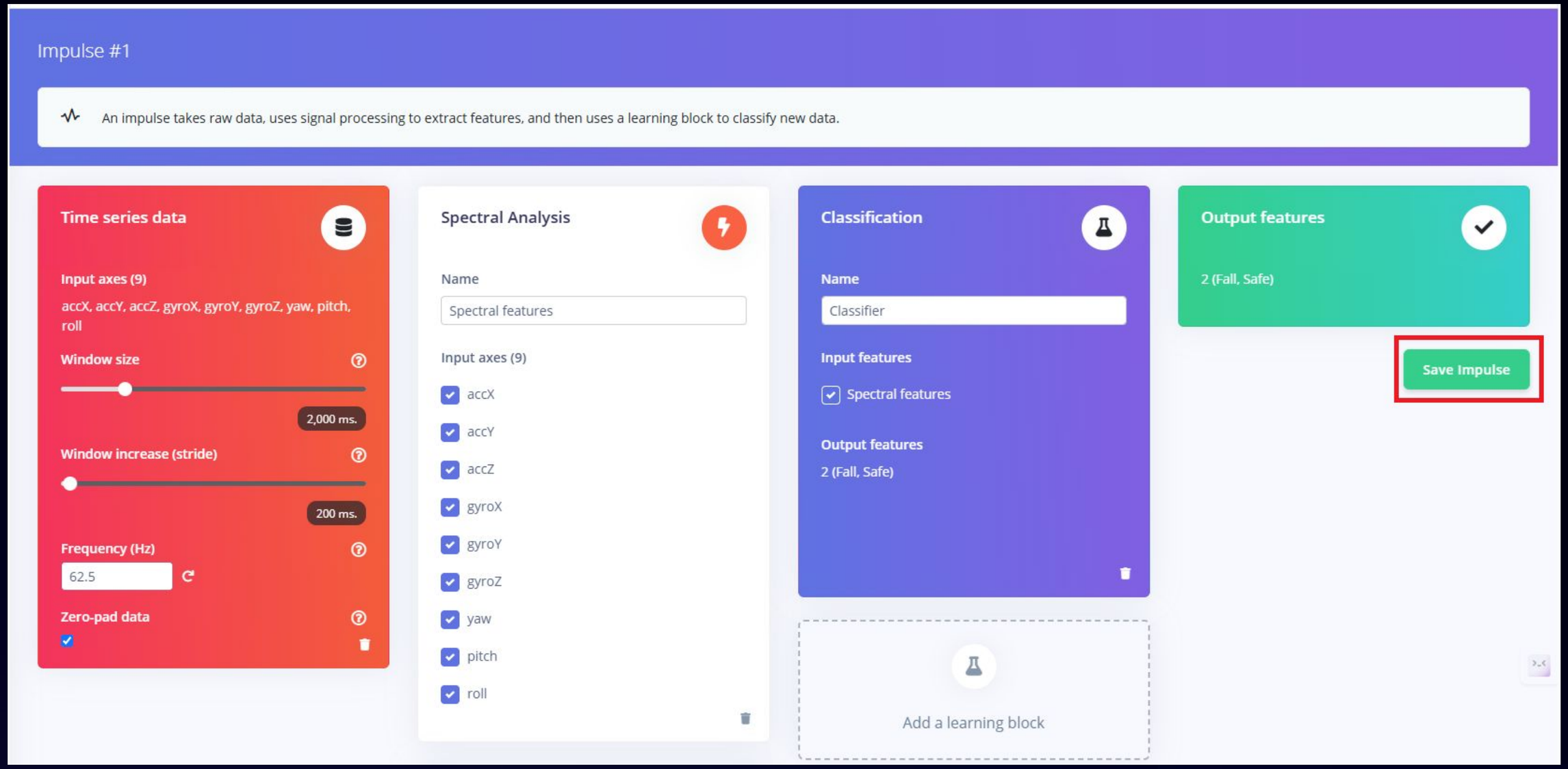
Step 28: Create impulse



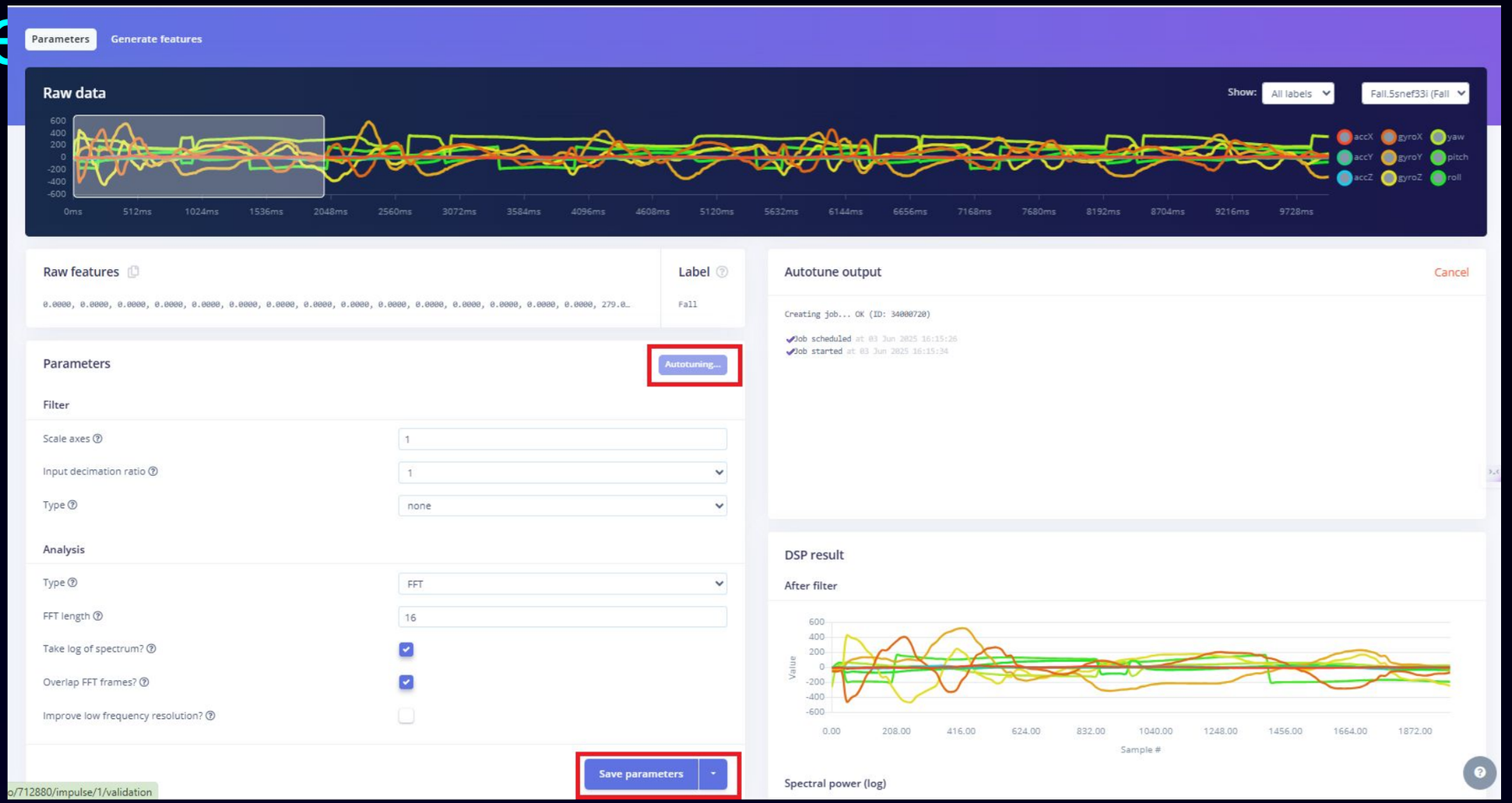
Step 29: Create impulse: Add processing and Learning



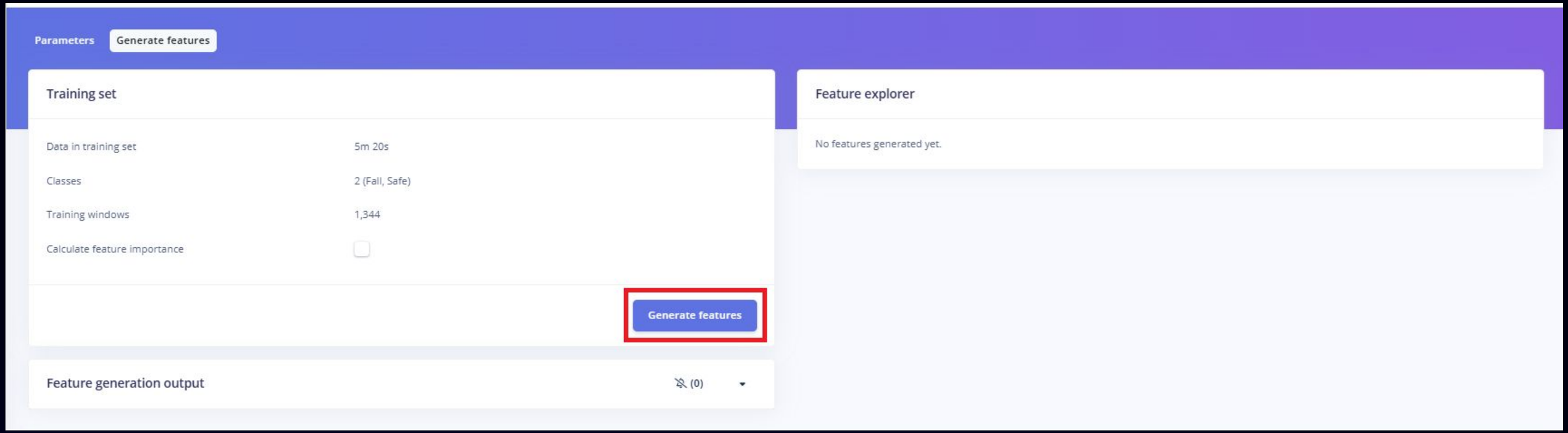
Step 30: Create impulse: Add processing and Learning



Step 31: Spectral Features: Autotune and Save Parame



Step 32: Generate Feature



Step 33: Generate Feature

Parameters | Generate features

Training set

Data in training set	5m 20s
Classes	2 (Fall, Safe)
Training windows	1,344
Calculate feature importance	☑

Generate features

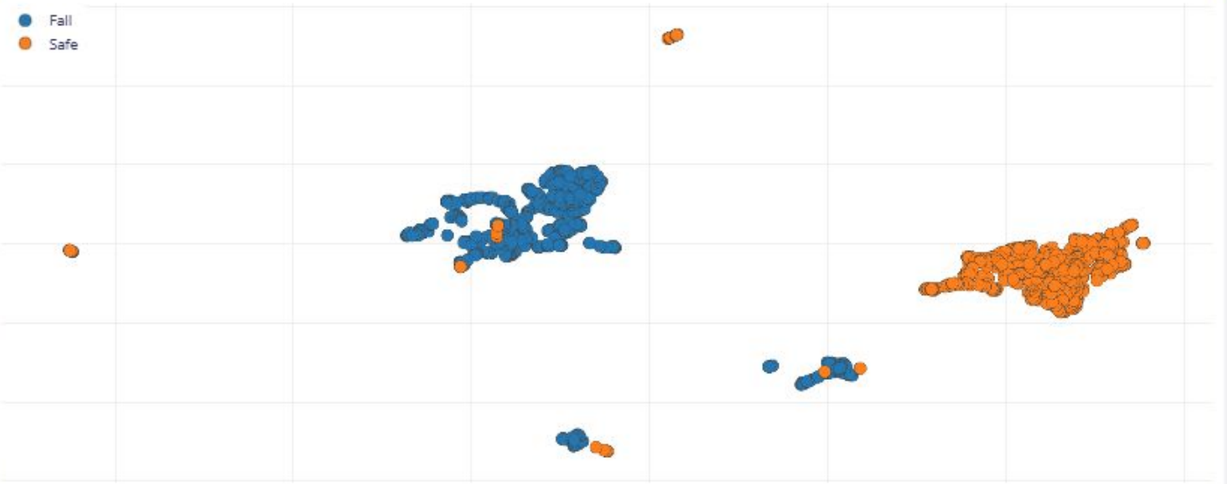
Feature generation output

(0)

```
[ 785/1344] Creating features...
[ 842/1344] Creating features...
[ 898/1344] Creating features...
[ 955/1344] Creating features...
[1011/1344] Creating features...
[1068/1344] Creating features...
[1124/1344] Creating features...
[1181/1344] Creating features...
[1237/1344] Creating features...
[1294/1344] Creating features...
[1344/1344] Creating features...
Created features

Job completed (success)
```

Feature explorer

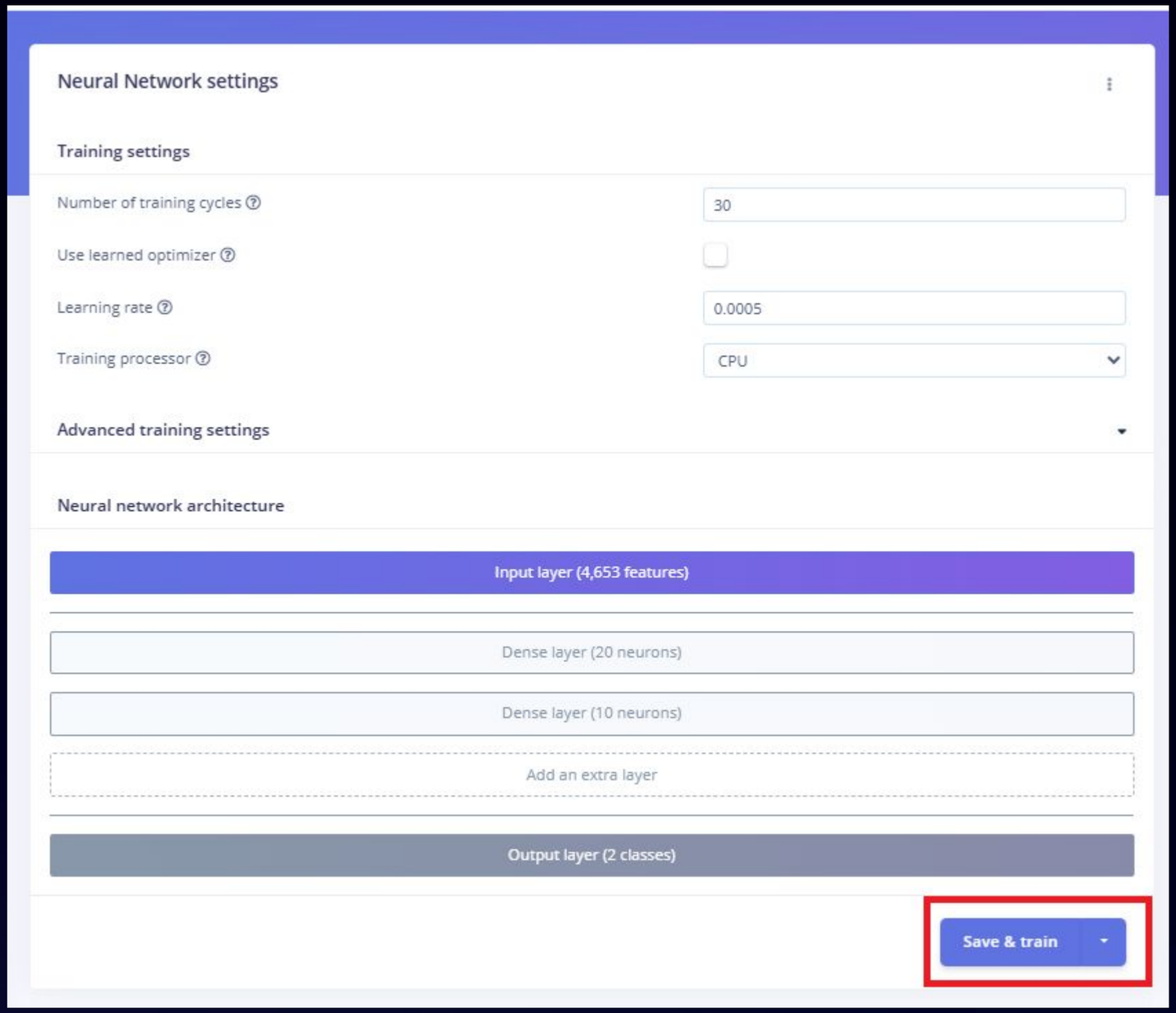


On-device performance

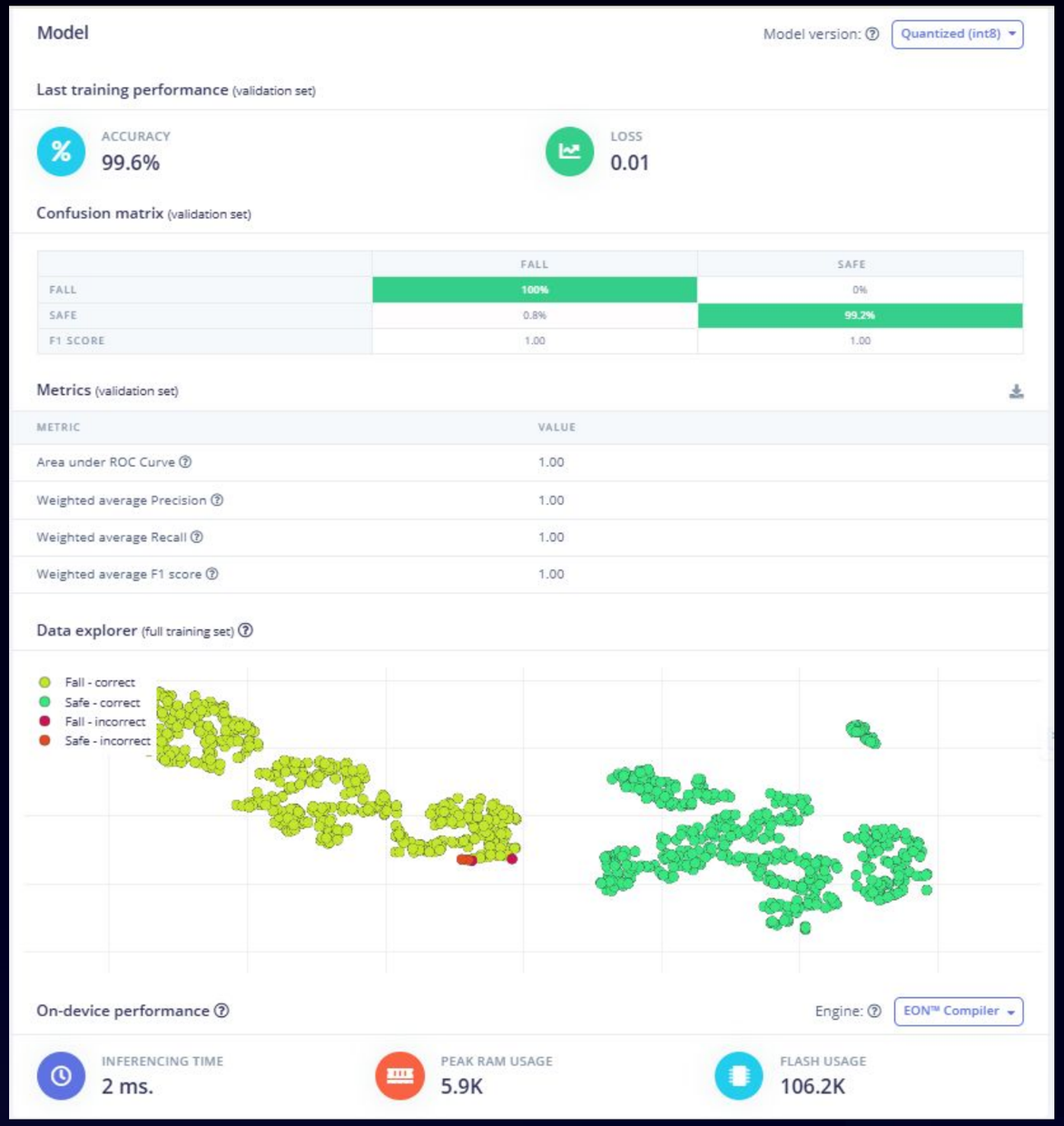
PROCESSING TIME
33 ms.

PEAK RAM USAGE
5 KB

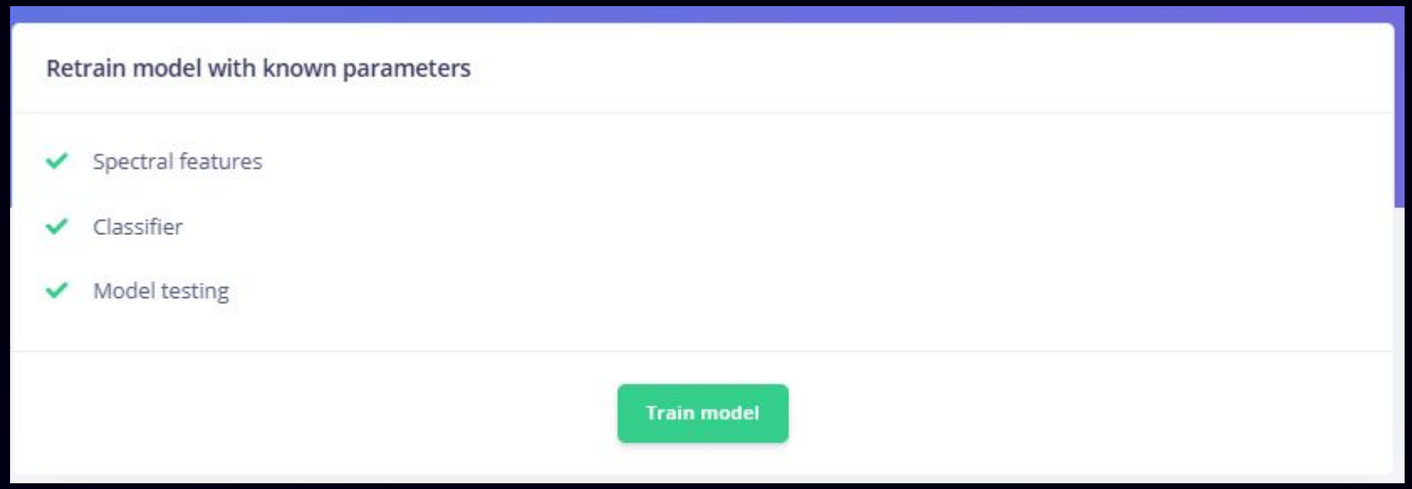
Step 34: Classifier

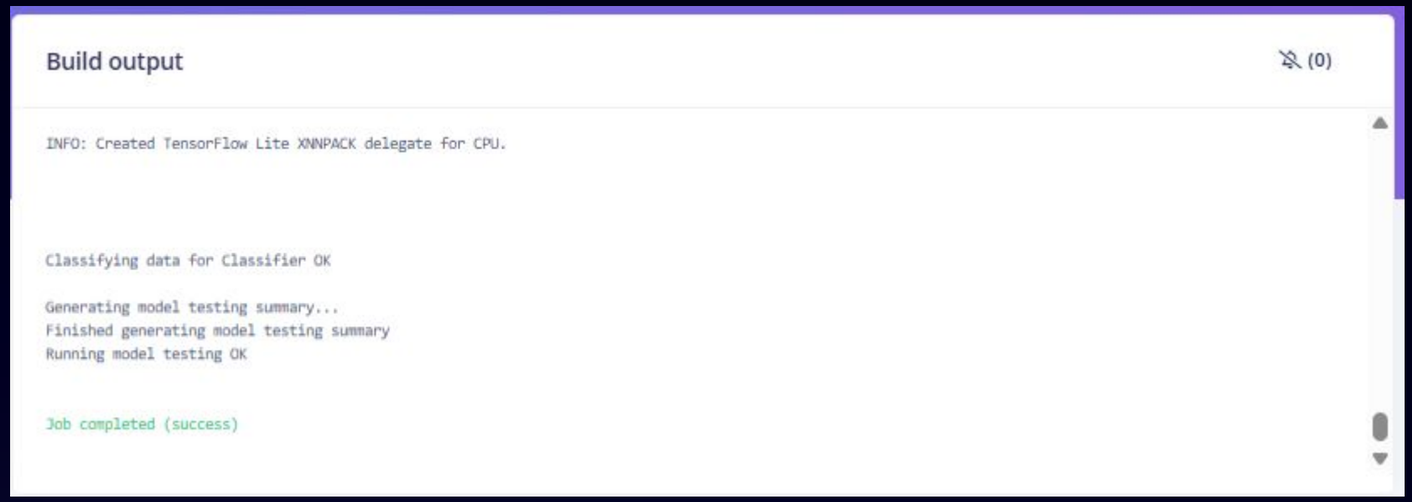


Step 35: Model

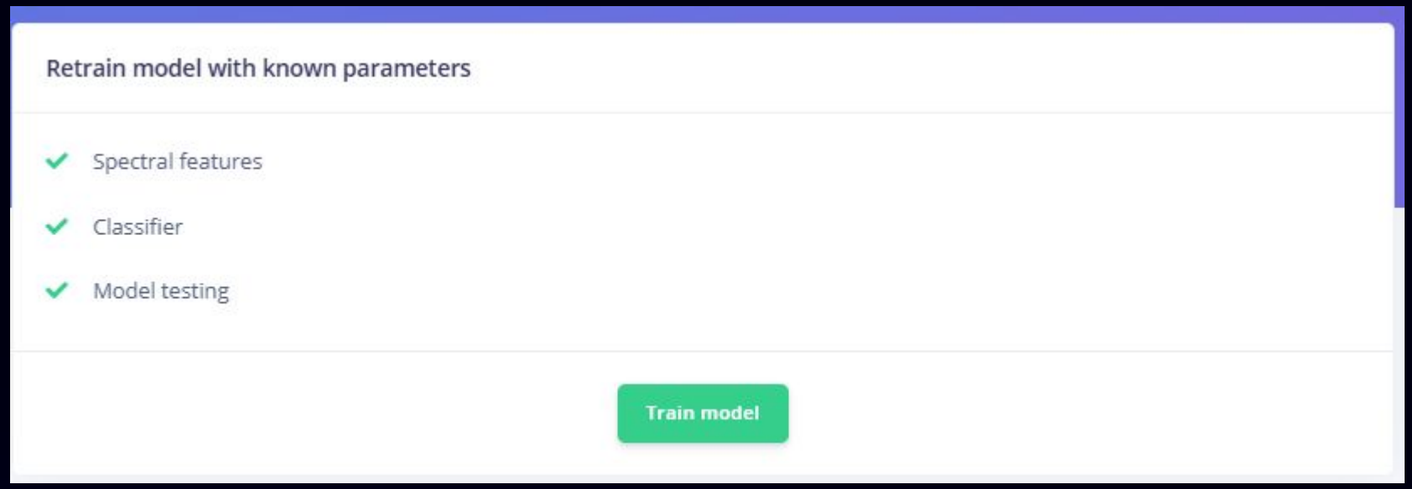


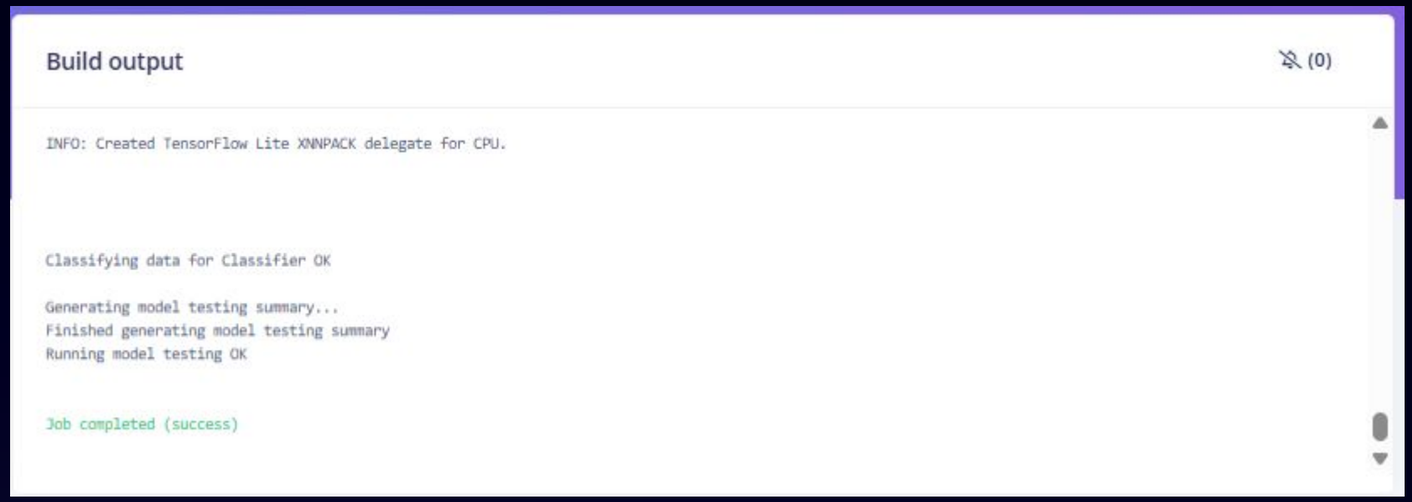
Step 36: Retrain Model with known parameters



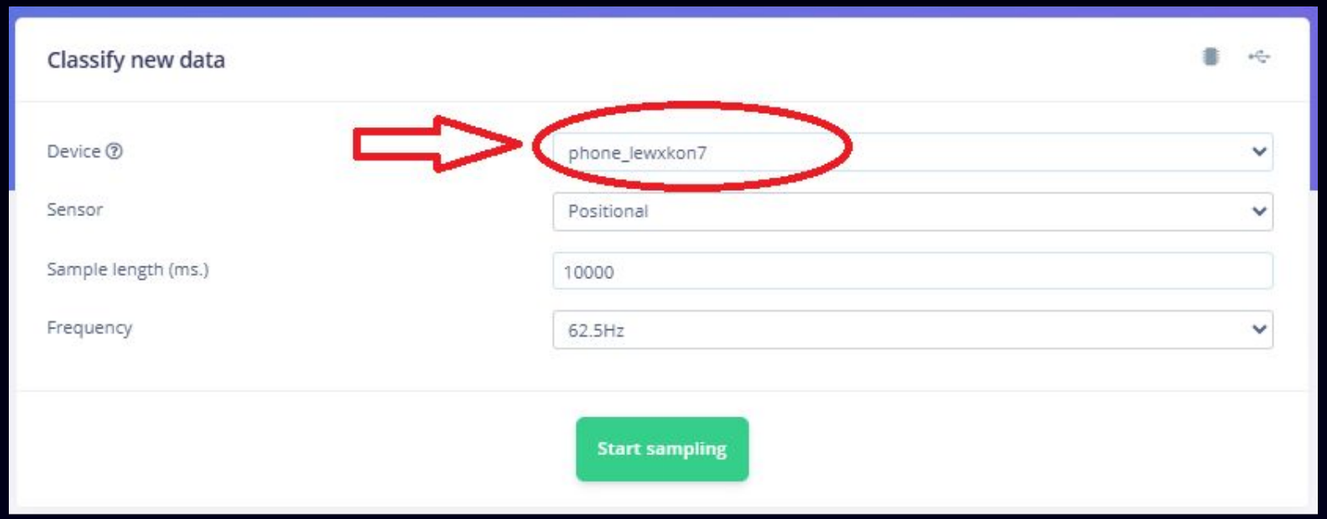


Step 37: Retrain Model with known parameters



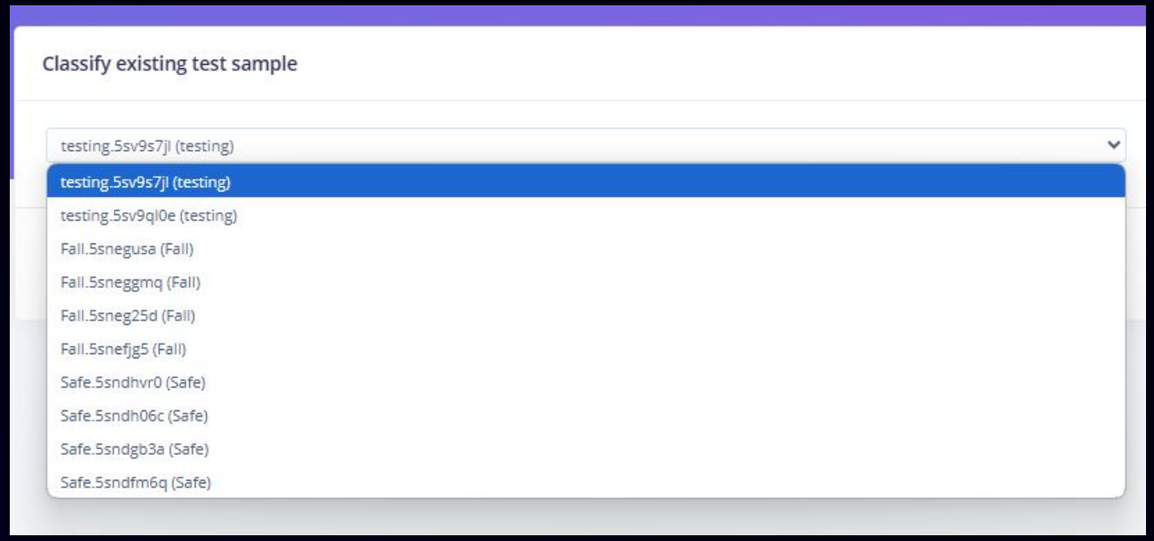


Step 38: Live Classification



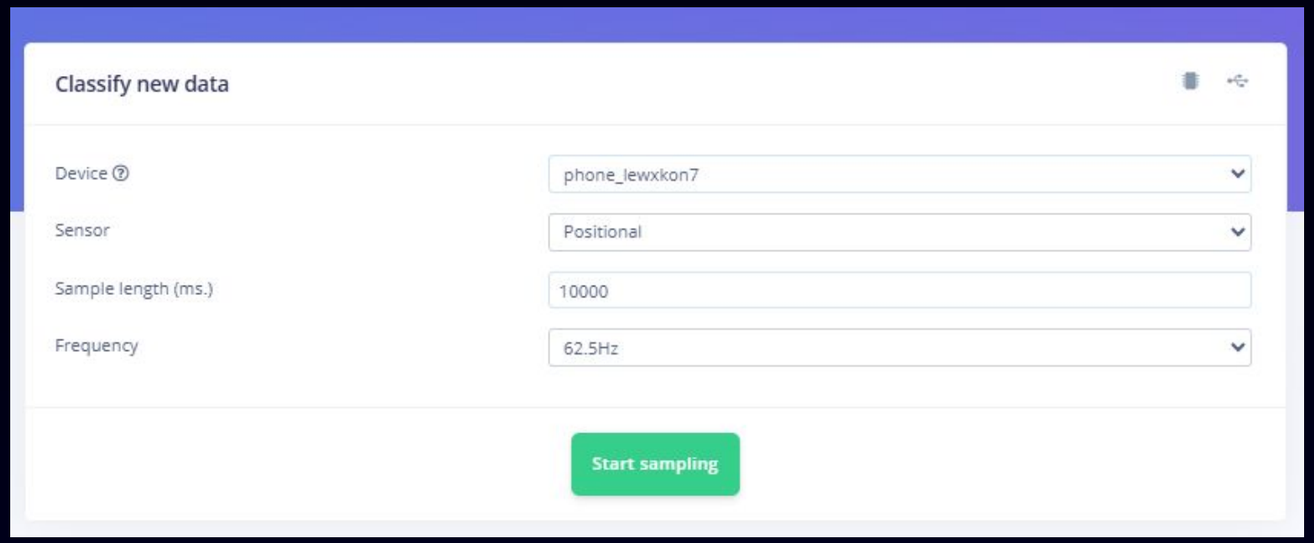
Phone Should be connected

Live from Phone

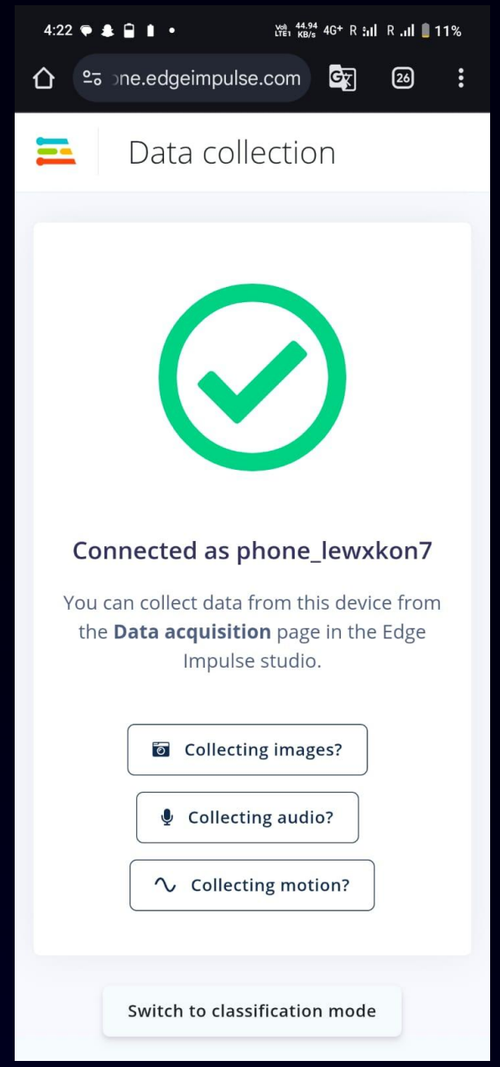


From existing test samples

Step 39: From Phone

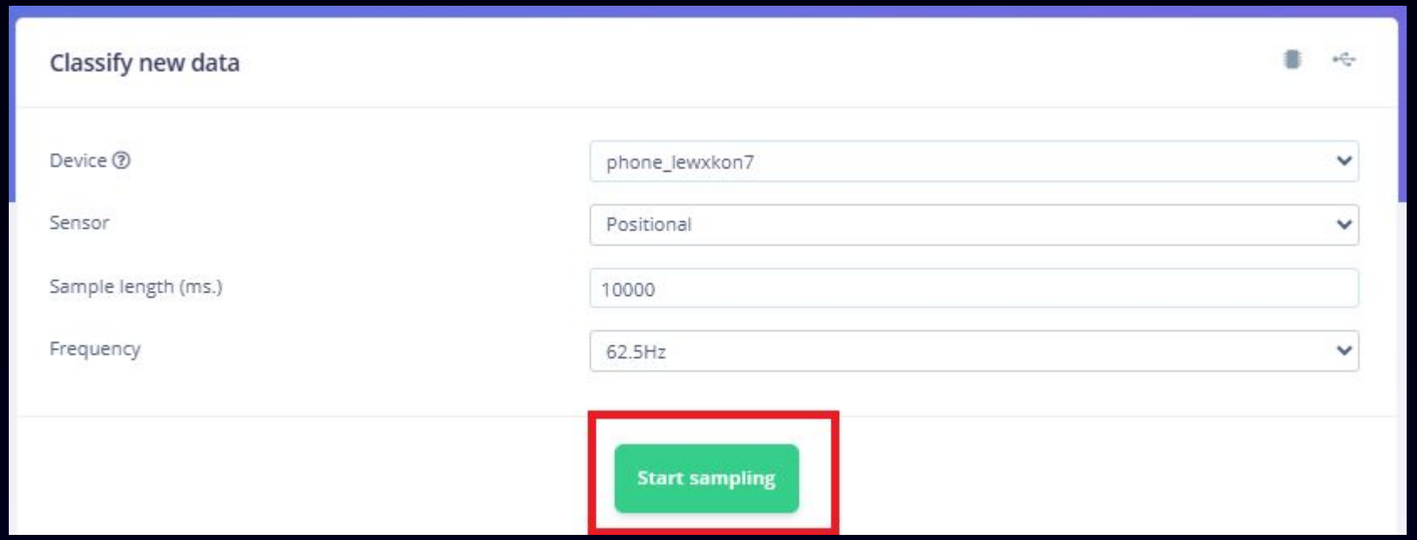


Live from Phone

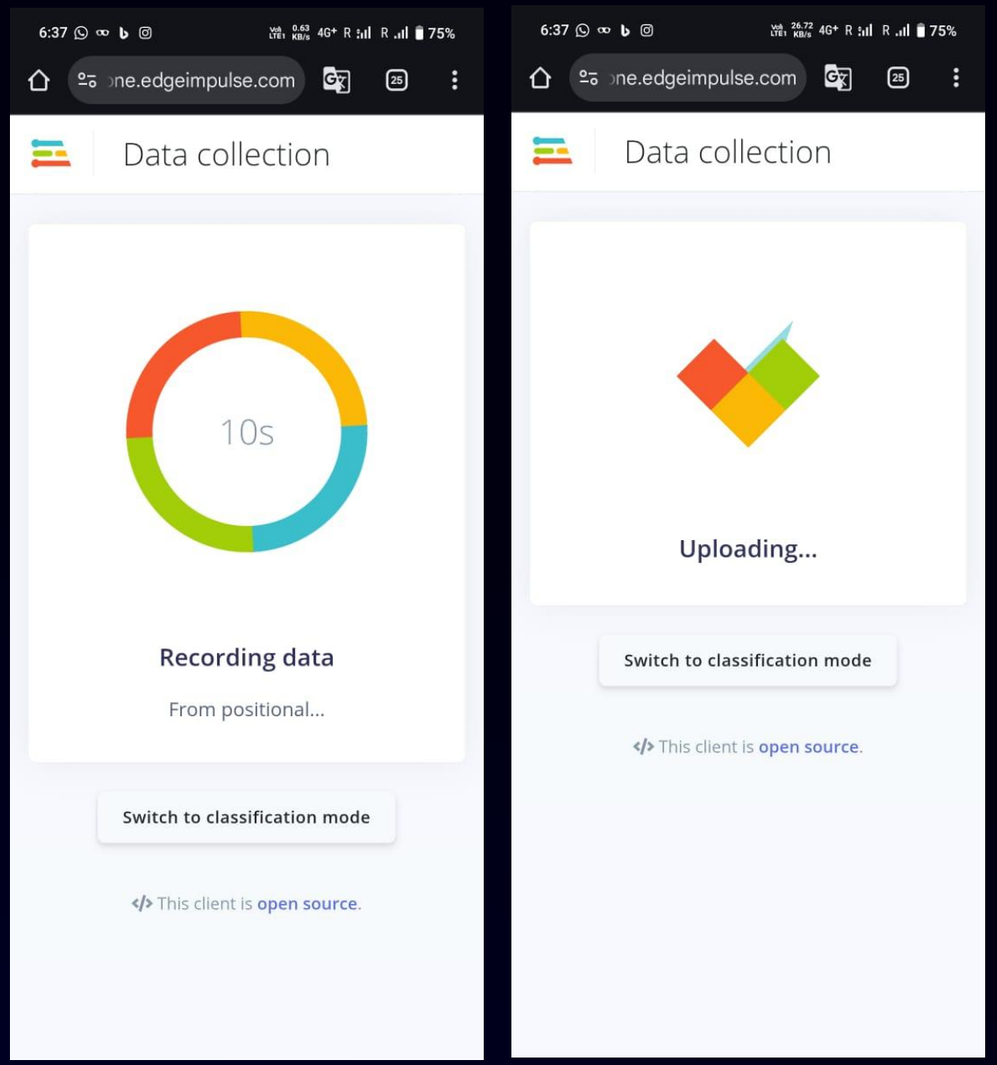


Phone Screen

Step 40: From Phone



Click start sampling



Phone Screen

Step 41: Detailed Result

Detailed result

Show only unknowns

TIMESTAMP	FALL	SAFE
0	0	1.00
192	0	1.00
384	0	1.00
576	0.09	0.91
768	0.14	0.86
960	0.20	0.80
1,152	0.12	0.88
1,344	0.24	0.76
1,536	0.25	0.75
1,728	0.24	0.76
1,920	0.21	0.79

Step 42: Model Testing

Test data

Classify all

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

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT
testing.5svaabof	testing	10s		
testing.5sv9s7jl	testing	10s		
testing.5sv9ql0e	testing	10s		
Safe.5sndgb3a	Safe	10s	100%	42 Safe
Safe.5sndfm6q	Safe	10s	100%	42 Safe
Safe.5sndh06c	Safe	10s	100%	42 Safe
Safe.5sndhvr0	Safe	10s	98%	41 Safe, 1 uncertain
Fall.5snefjg5	Fall	10s	100%	42 Fall
Fall.5sneg25d	Fall	10s	100%	42 Fall
Fall.5sneggmq	Fall	10s	100%	42 Fall
Fall.5snegusa	Fall	10s	100%	42 Fall

Model testing output

(0)

Results

Model version: Unoptimized (float32)

ACCURACY
99.70%

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

	FALL	SAFE	UNCERTAIN
FALL	100%	0%	0%
SAFE	0%	99.4%	0.6%
F1 SCORE	1.00	1.00	

Feature explorer

Step 43: Deployment

Configure your deployment

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Search deployment options

DEFAULT DEPLOYMENT

C++ library

A portable C++ library with no external dependencies, which can be compiled with any modern C++ compiler.

MODEL OPTIMIZATIONS

Model optimizations can increase on-device performance but may reduce accuracy.

EON™ Compiler

Same accuracy, 30% less RAM, 16% less ROM.

Quantized (int8)

Selected ✔

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	33 ms.	2 ms.	**35 ms.**
RAM	5.4K	5.9K	**5.9K**
FLASH	-	106.2K	-
ACCURACY			-

Unoptimized (float32)

Select

	SPECTRAL FEATURES	CLASSIFIER	TOTAL
LATENCY	33 ms.	3 ms.	**36 ms.**
RAM	5.4K	19.5K	**19.5K**
FLASH	-	376.1K	-
ACCURACY			**99.70%**

To compare model accuracy, run model testing for all available optimizations.

Run model testing

Estimate for Seeed SenseCAP A1101 (HX6537-A ARC DSP 400MHz) - Change target

Build

Run this model

Scan QR code or launch in browser to test your prototype

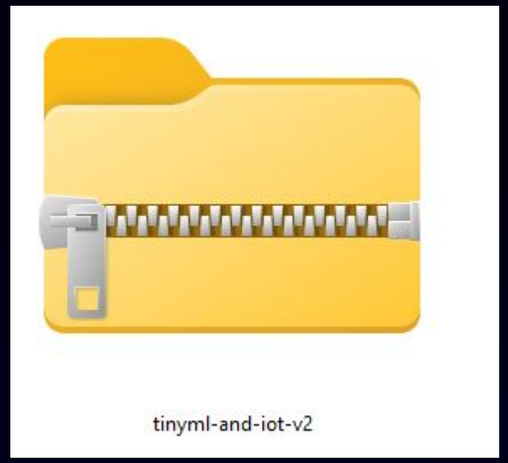
Launch in browser

Step 44: Deployment

Sampling...

	FALL	SAFE
7	0.00	1.00
6	0.00	1.00
5	0.00	1.00
4	0.00	1.00
3	0.00	1.00
2	0.00	1.00
1	0.00	1.00

Step 45: Final Library



Thank You for Your Attention