SUBJECT- CHS

PROJECT REPORT

TINYML USING EDGE IMPULSE

TOPIC- MOTION DETECTION MODEL

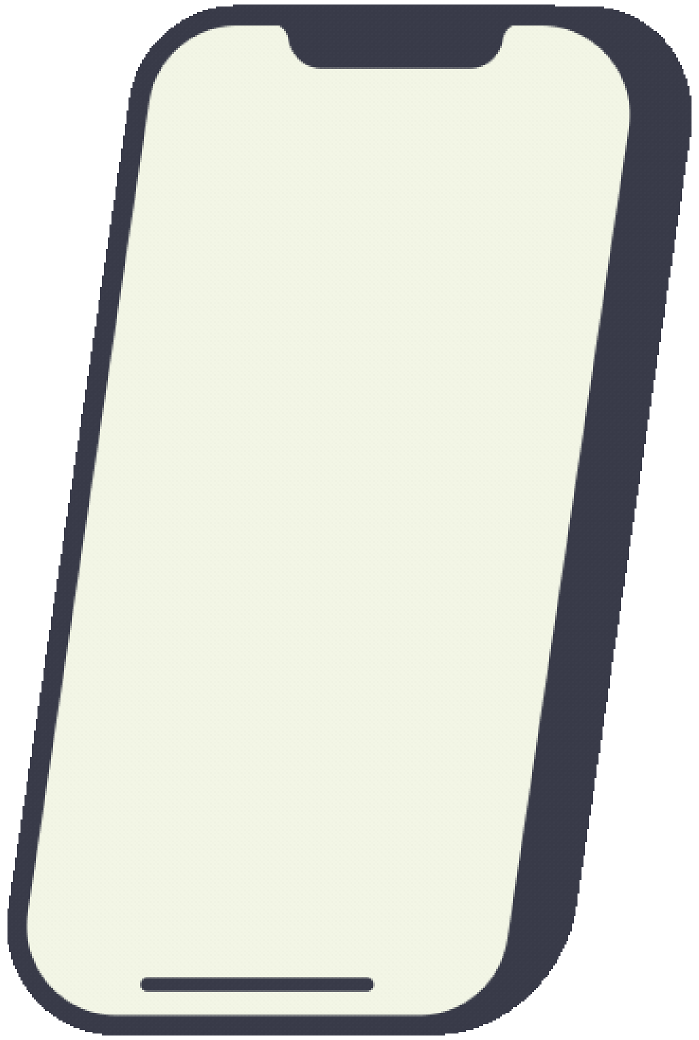
SUBMITTED BY:
AYUSH CHAUHAN-2021UCS2103
SWATI SHARMA-2021UCS2158
DEEPANSHU SINGHAL-2021UCS2159

Get Started

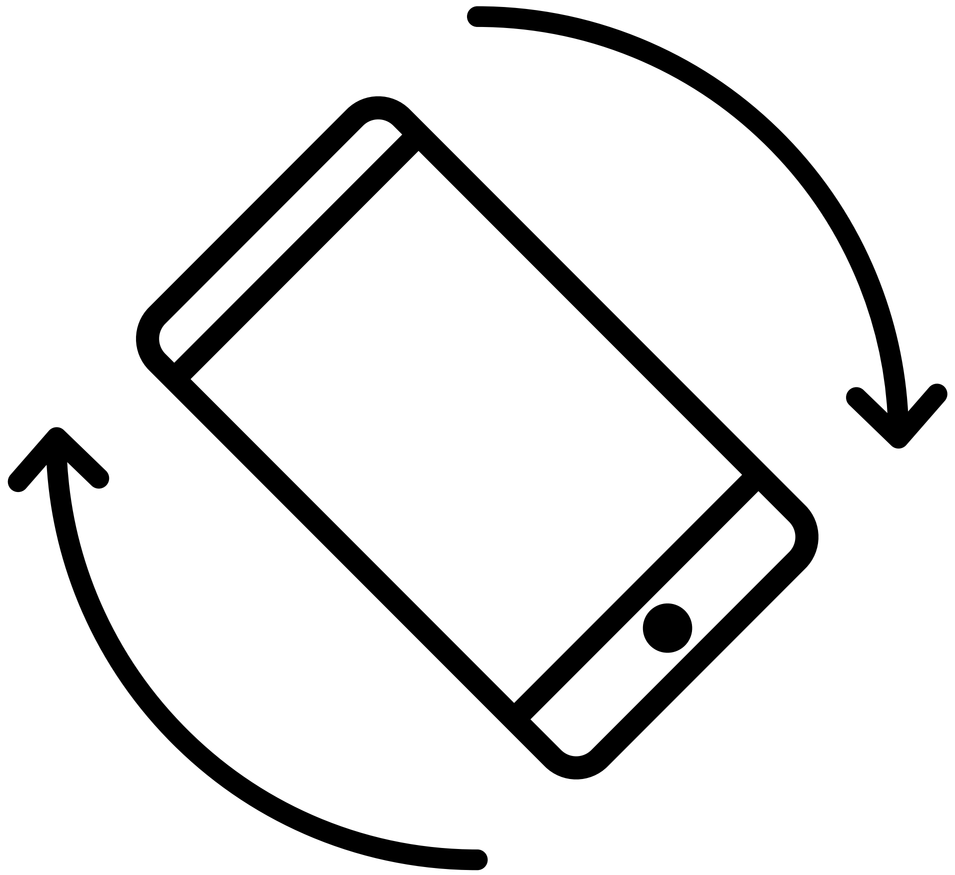
Objective:

The objective is to enhance our motion detection model to accurately identify and track diverse movements, including left-right, up-down, and waving motions. This expansion aims to showcase the model's versatility, making it suitable for a broader range of applications such as surveillance, robotics, and gesture recognition.

OverView



Wave



Left-Right

Up-Down

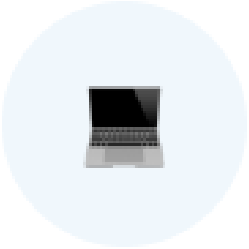
DATA COLLECTION

COLLECTION OF DATA THROUGH VARIOUS DEVICES

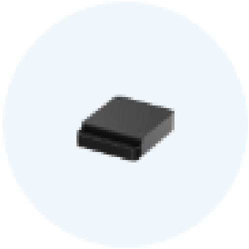
Collection of data by using phone , computer aur any other development board like arduino.
Training Data and Testing data are divided for different purpose to make the classification of the model .

- Gather labeled motion data for training your model. This could include accelerometer or gyroscope data from your chosen sensor.
- Preprocess the data to extract relevant features and normalize values.

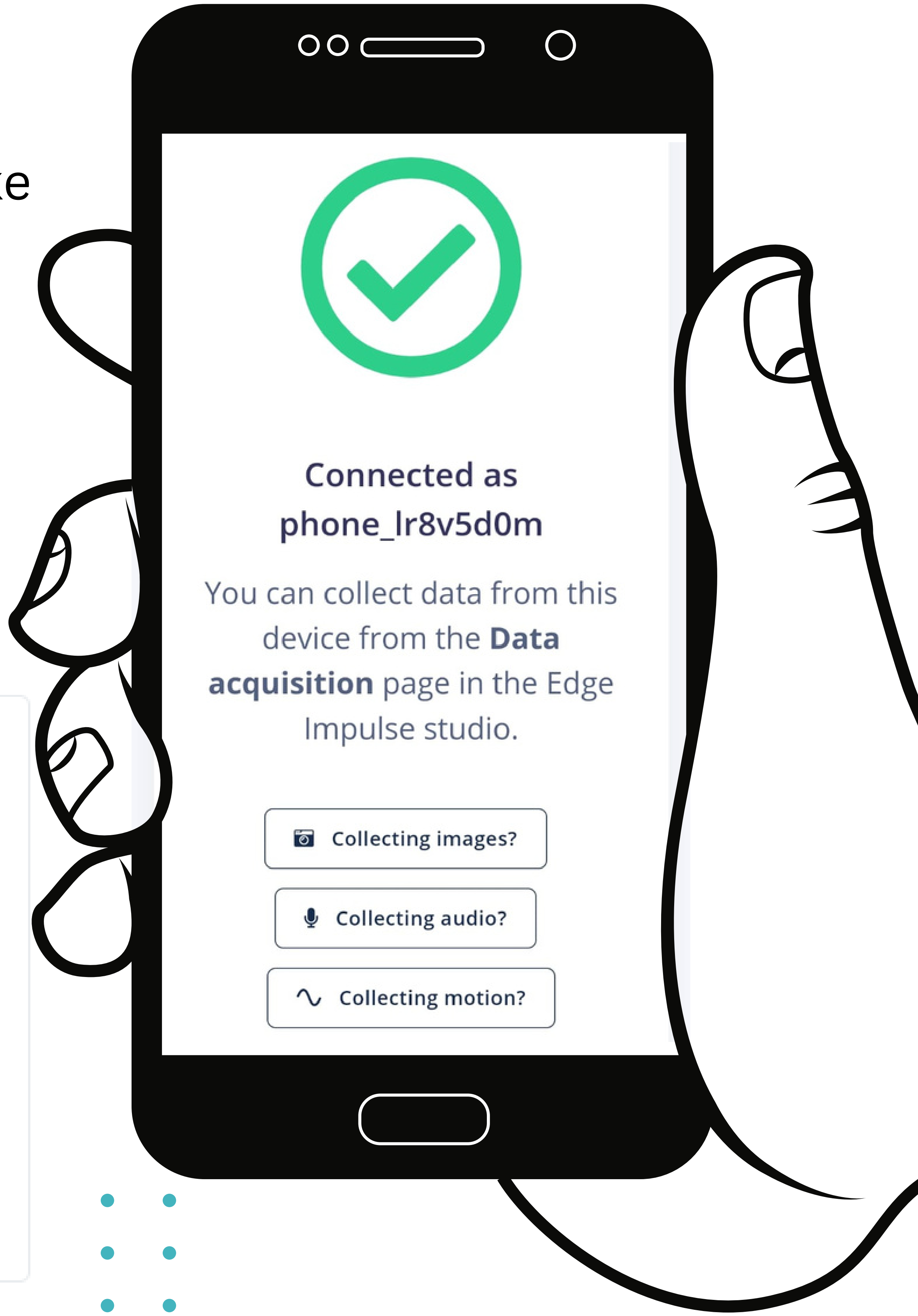
Scan QR code to connect to your phone



Connect to your computer



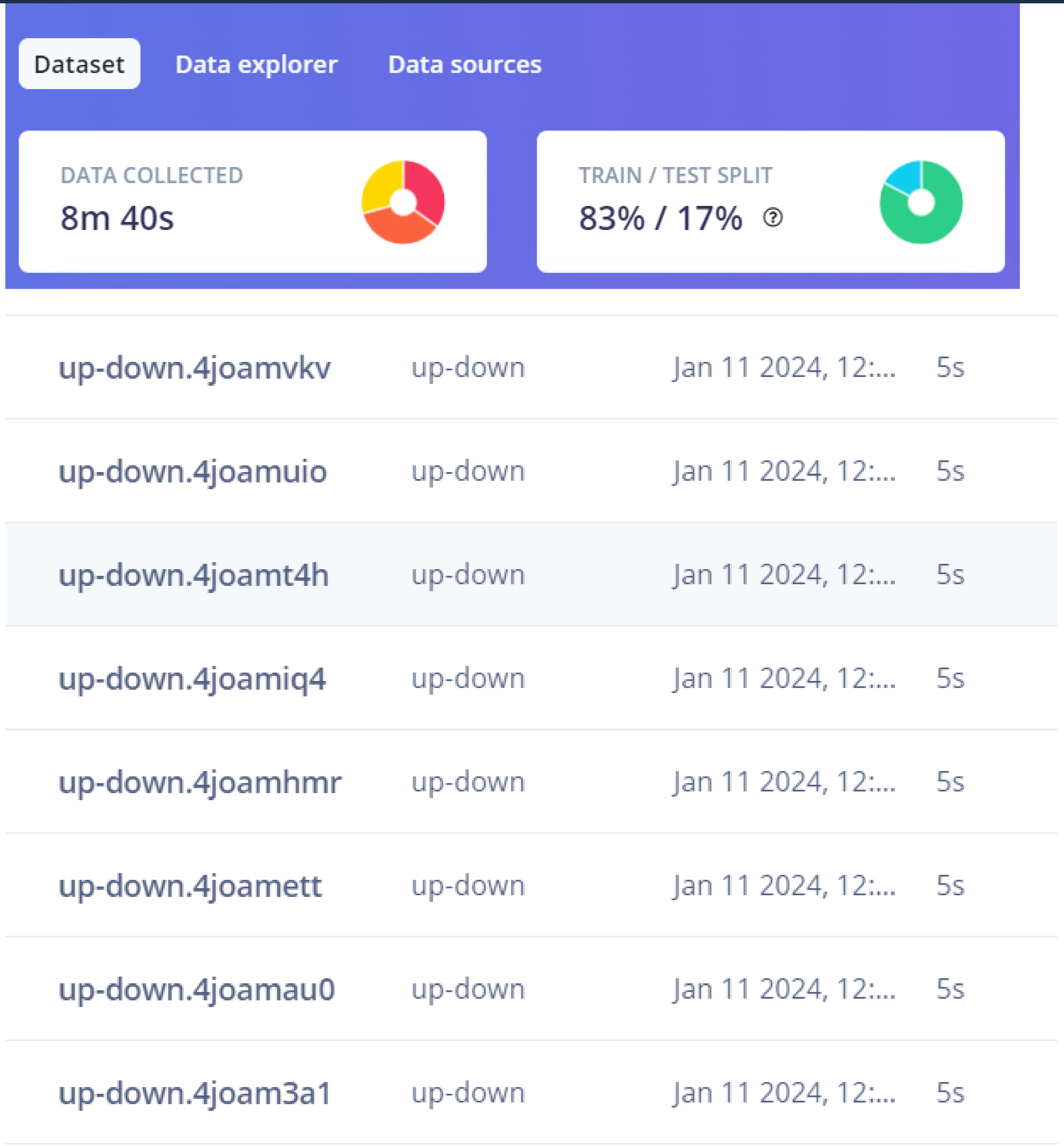
Connect your device or development board



DATA COLLECTION

TRAINING DATA

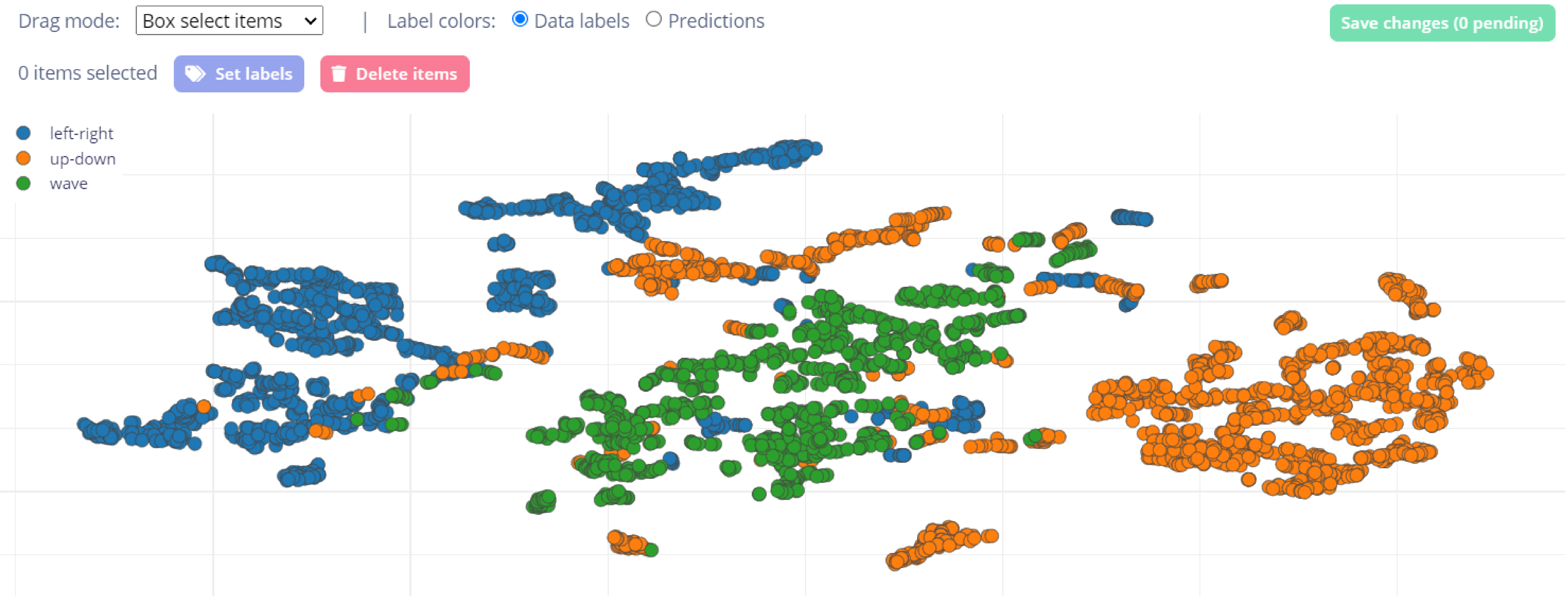
- Train the selected model using the preprocessed motion data. This may involve using a machine learning an online platform Edge Impulse.
- Optimizing the model for deployment on a microcontroller by considering factors like quantization and pruning.



DATA ACQUISATION

DATA EXPLORER

The data explorer shows a complete view of all data in your project. You can clear labels through the menu on the right, and inspect or change labels by clicking on individual data items.



DATA ACQUISATION

DATA EXPLORER

Algorithm Used

K-means

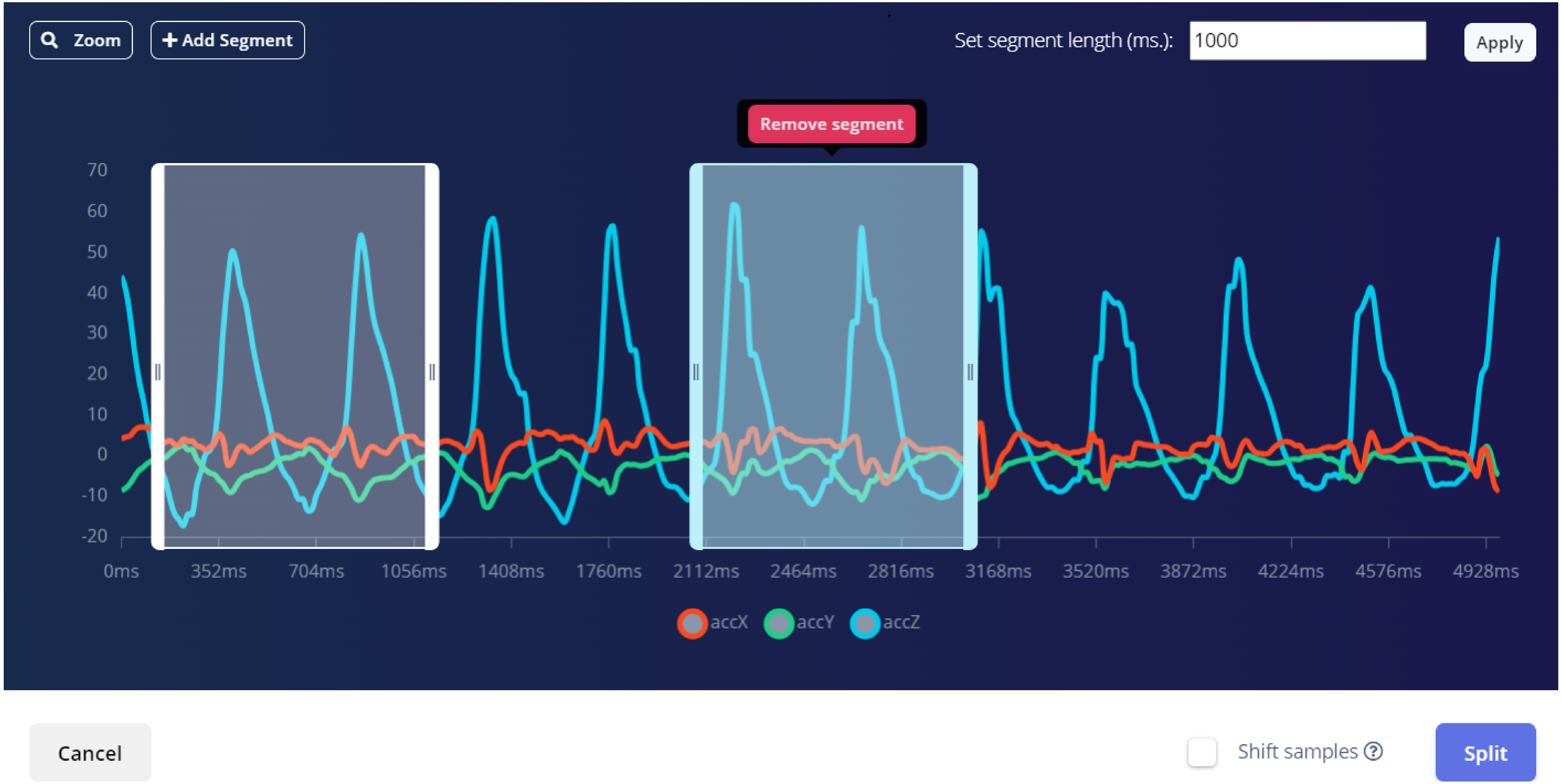
K-Means assigns data points to 'k' clusters iteratively, optimizing centroids to minimize intra-cluster variance. It's an unsupervised algorithm widely used for data grouping and segmentation.

Naive Bayes

A probabilistic classification algorithm assuming feature independence. It calculates class probabilities and assigns the most likely class to new data. Simple yet effective, it's commonly used for text classification and spam filtering.

DATA ACQUISATION

SPLITTING OF RAW DATA



NETWORK ARCHITECTURE

Neural Network settings

Training settings

Number of training cycles ⓘ

Advanced training settings

Validation set size ⓘ	20 %
Split train/validation set on metadata key ⓘ	
Batch size ⓘ	32
Auto-weight classes ⓘ	☐
Profile int8 model ⓘ	☑

Neural Network

Neural network architecture

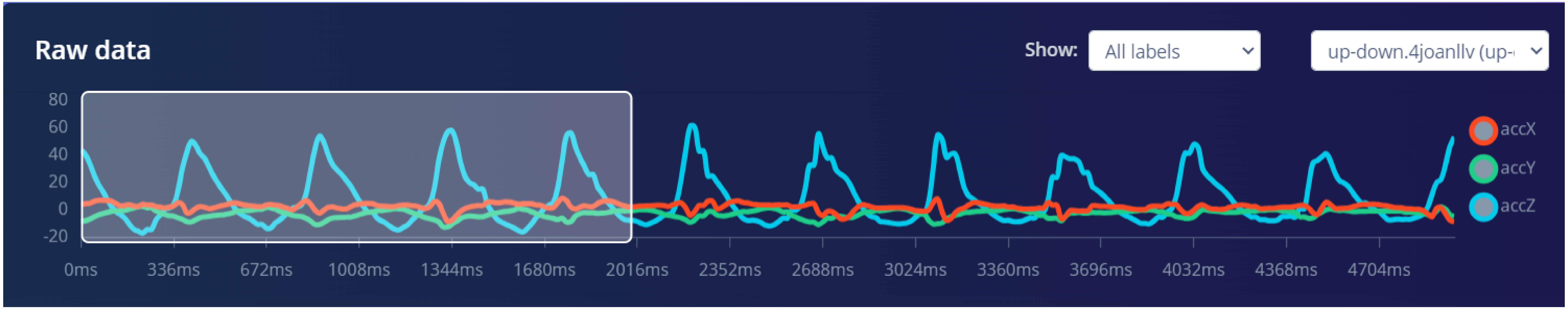
Input layer (39 features)

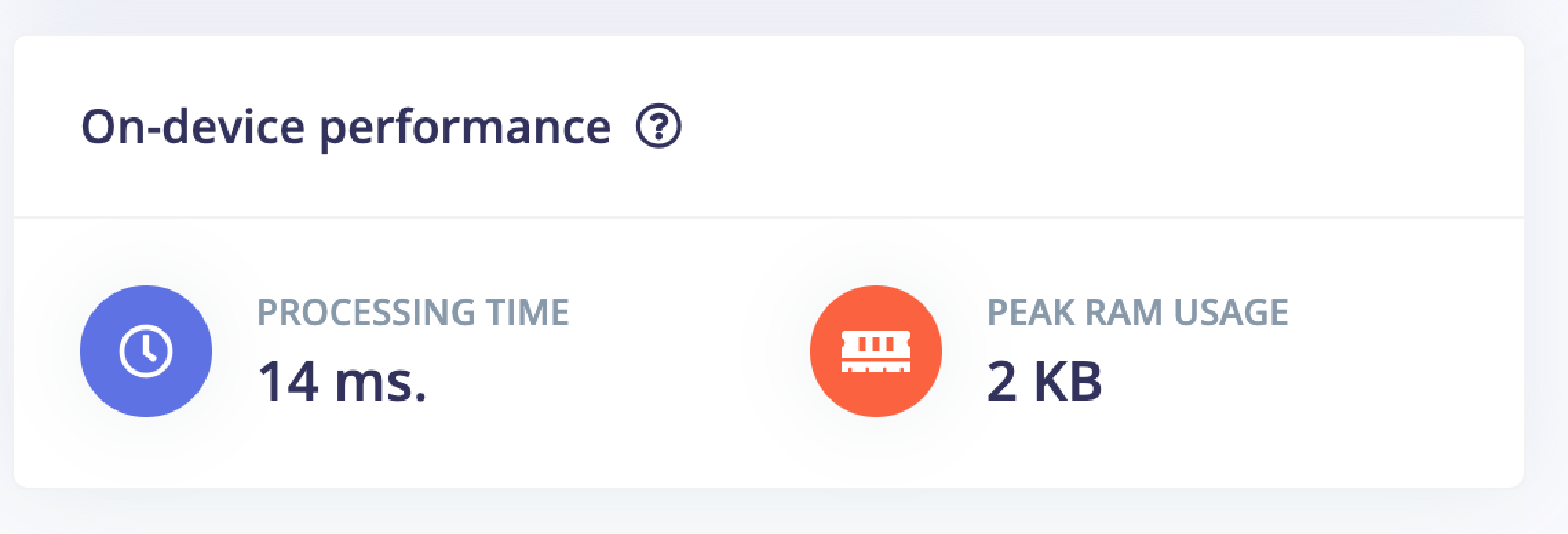
Dense layer (20 neurons)

Dense layer (10 neurons)

Output layer (3 classes)

SPECTRAL FEATURES

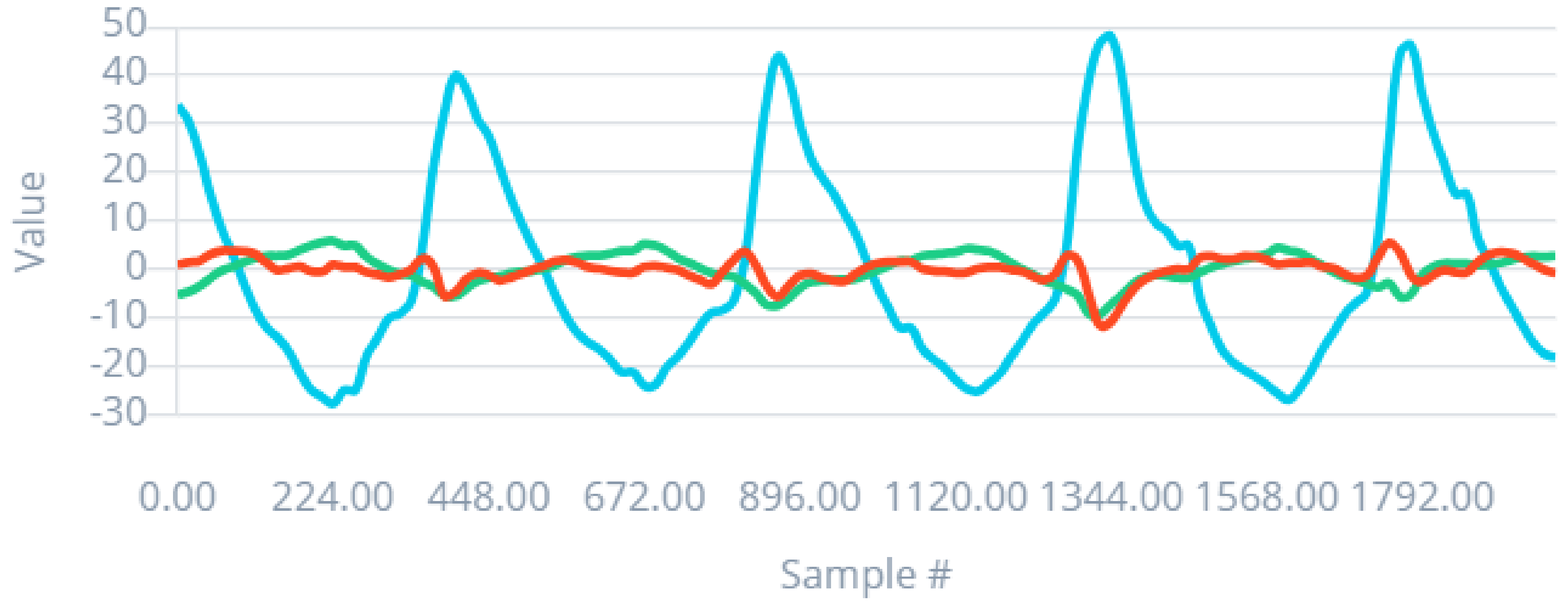




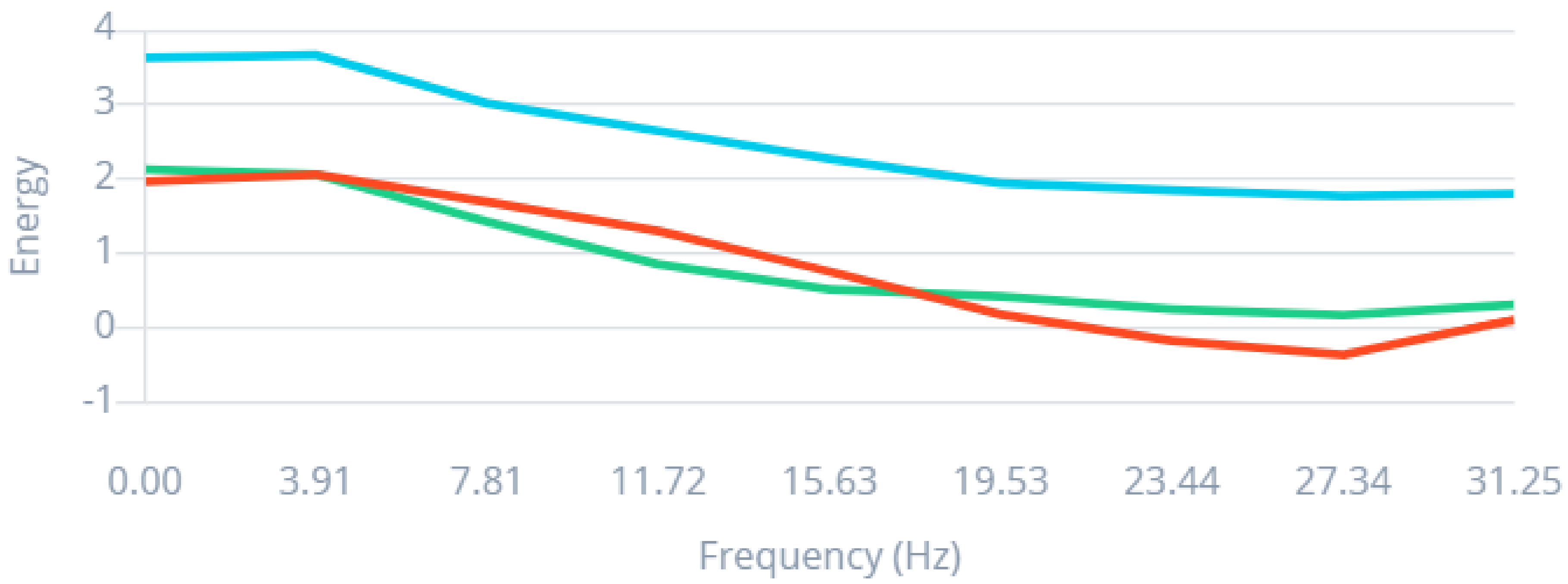
SPECTRAL FEATURES

DSP result

After filter



Spectral power (log)



ANOMALY DETECTION

Anomaly explorer (3,307 samples)

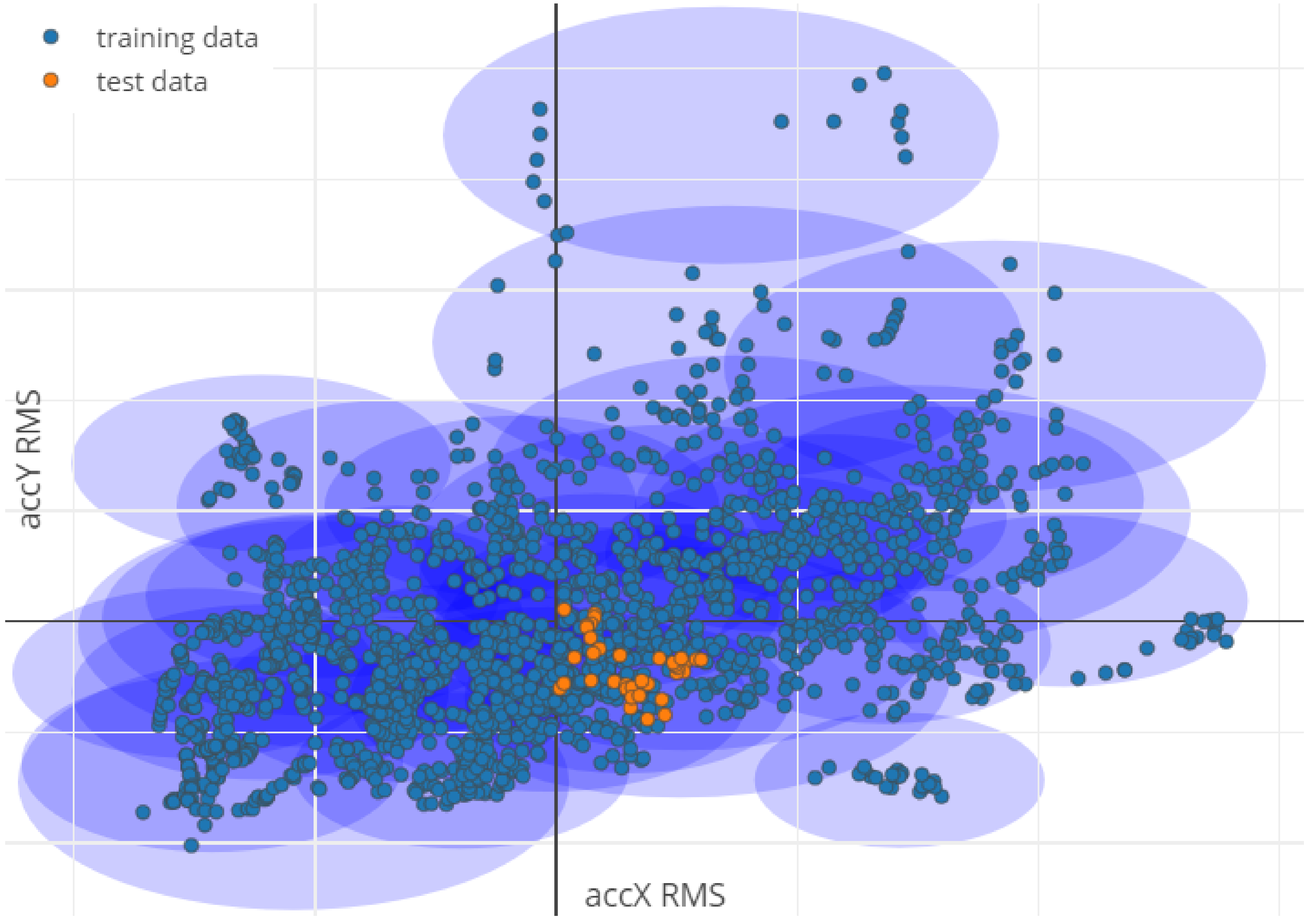
X Axis	Y Axis	Test data
accX RMS	accY RMS	wave.4joado61

Anomaly score

`min: -0.6355, max: -0.2020, avg: -0.4242`

Average axis distance

`accX RMS: 0.2434, accY RMS: 0.2851, accZ RMS: 0.2980`



CLASSIFIER

Model

Model version: ? Quantized (int8)

Last training performance (validation set)

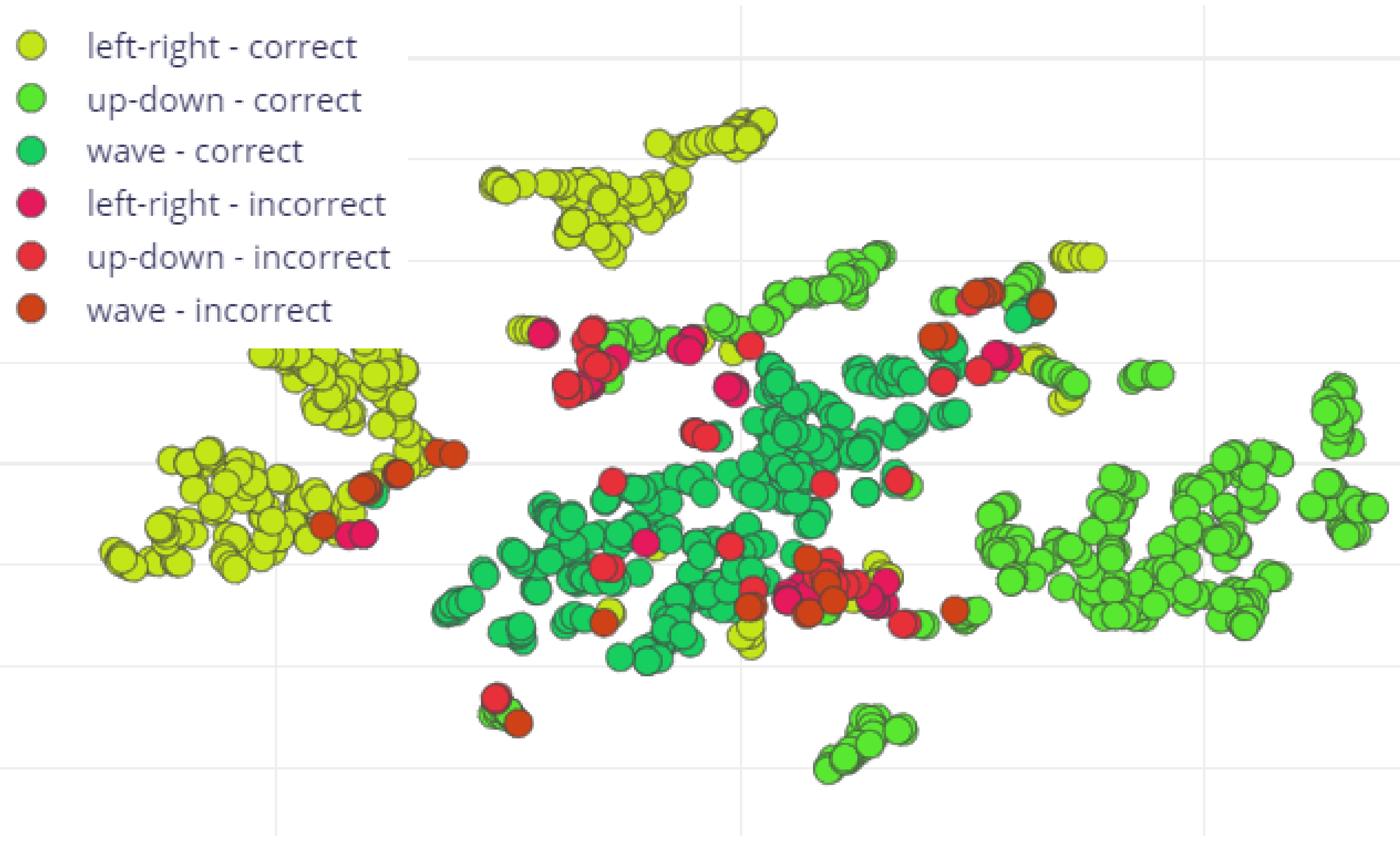
ACCURACY
94.6%

LOSS
0.17

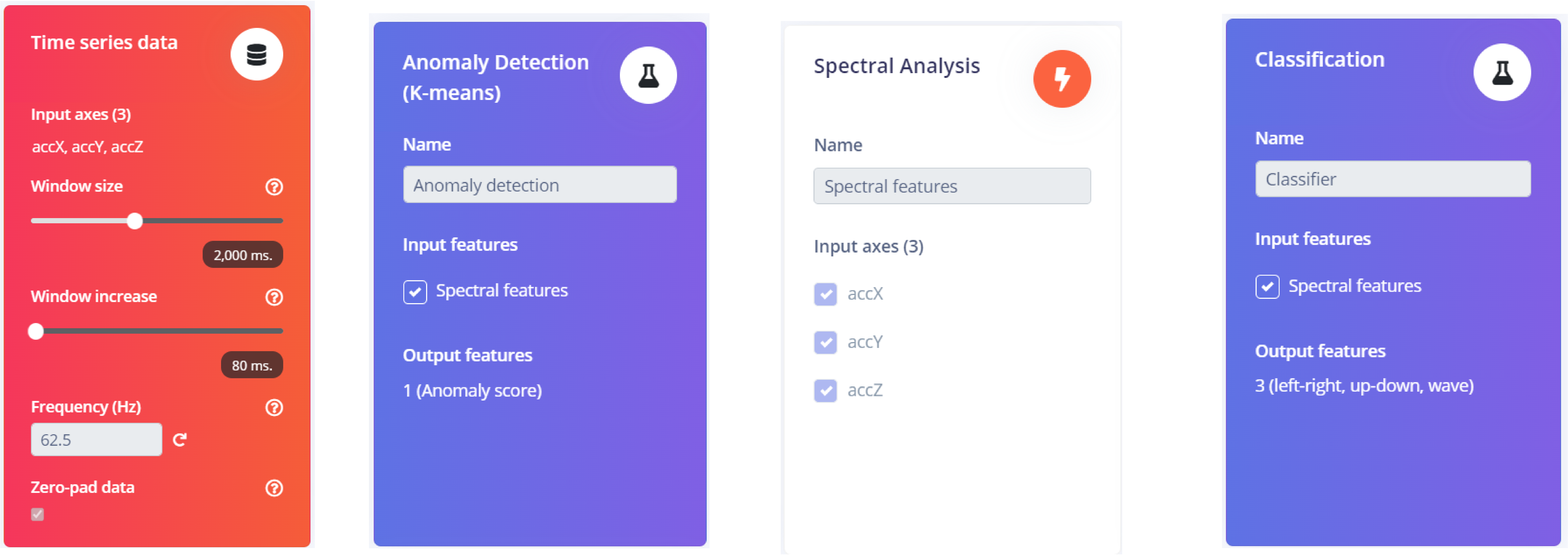
Confusion matrix (validation set)

	LEFT-RIGHT	UP-DOWN	WAVE
LEFT-RIGHT	94.7%	2.9%	2.4%
UP-DOWN	1.6%	93.8%	4.5%
WAVE	2.9%	1.5%	95.6%
F1 SCORE	0.95	0.95	0.94

Data explorer (full training set) ?



IMPULSE DESIGN



About Edge Impulse

A robust platform for embedded machine learning, Edge Impulse streamlines the development and deployment of TinyML models on edge devices. Its user-friendly interface supports diverse applications, making it a go-to solution for efficient implementation and integration of machine learning in resource-constrained environments.

Future Scope

1. Multi-Class Motion Detection:
    - Expand the model to handle multiple types of motions simultaneously, providing a more comprehensive solution for diverse scenarios.
2. Fine-Tuning for Specific Environments:
    - Investigate the adaptability of the model to specific environments or conditions by fine-tuning parameters for optimal performance.
3. Real-Time Implementation:
    - Explore real-time implementation possibilities, enabling the model to process and respond to motion in live streaming scenarios.
4. Integration with Edge Devices:
    - Extend the model's deployment capabilities by optimizing it for integration with edge devices, enhancing its suitability for edge computing applications.
5. Enhanced Object Recognition:
    - Integrate object recognition capabilities to identify and track specific objects within the detected motion, adding depth to the model's functionality.

Future Scope

6. **Dynamic Thresholding:**
    - Implement dynamic thresholding techniques to adaptively adjust sensitivity, ensuring robust performance across varying motion intensities.
7. **Gesture Recognition:**
    - Investigate the feasibility of incorporating gesture recognition, allowing the model to interpret and respond to specific human gestures within the detected motions.
8. **User Interface Enhancement:**
    - Develop a user-friendly interface to visualize motion outputs, facilitating easier interpretation and interaction with the model's prediction
9. **Optimization for Low-Power Devices:**
    - Optimize the model for low-power consumption, enabling deployment on battery-operated or resource-constrained devices without compromising performance.