IESTI01 - TinyML

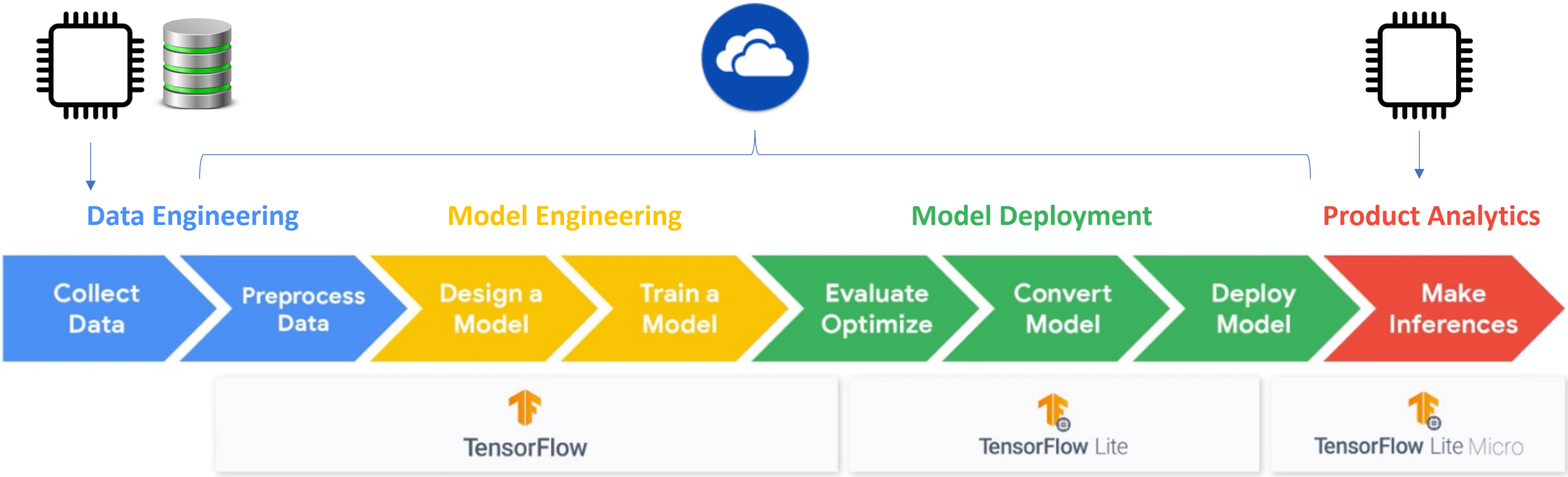
Introduction to
Edge Impulse Studio

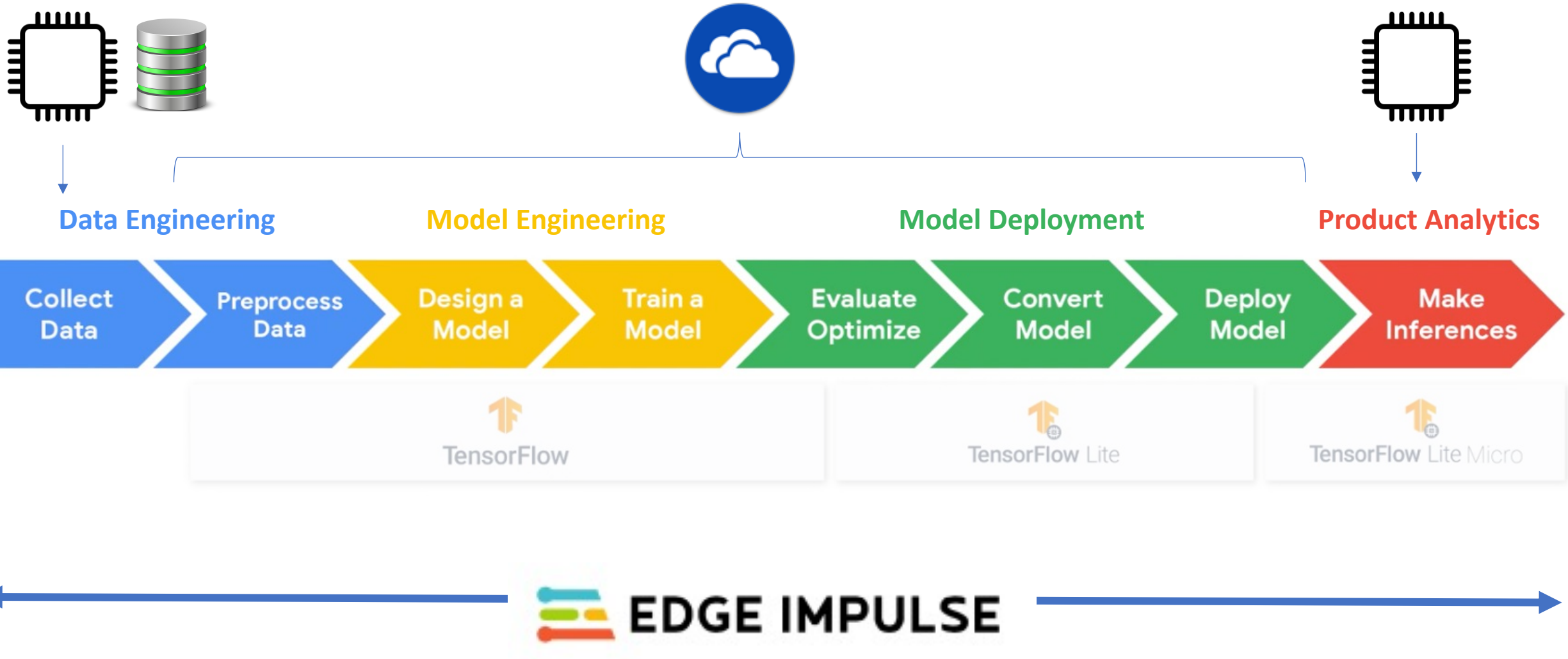
Prof. Marcelo Rovai

June 16th, 2021

ML Project Workflow

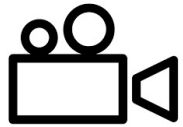
with Edge Impulse Studio

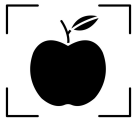

Data Engineering
Model Engineering
Model Deployment
Product Analytics
Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro


Data Engineering
Model Engineering
Model Deployment
Product Analytics
Collect Data
Preprocess Data
Design a Model
Train a Model
Evaluate Optimize
Convert Model
Deploy Model
Make Inferences
TensorFlow
TensorFlow Lite
TensorFlow Lite Micro
EDGE IMPULSE

EdgeML

TinyML





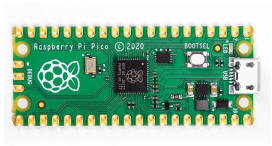
Anomaly Detection
Sensor Classification
20 KB

KeyWord Spotting
Audio Classification
50 KB

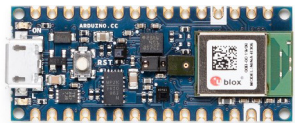
Image
Classification
250 KB+

Object
Detection
Complex Voice
Processing
1 MB+

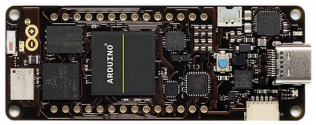
Video
Classification
2 MB+



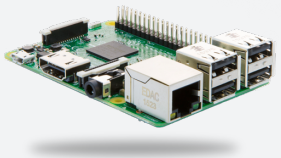
Rpi-Pico
(Cortex-M0+)



Arduino Nano
(Cortex-M4)



Arduino Pro
(Cortex-M7)



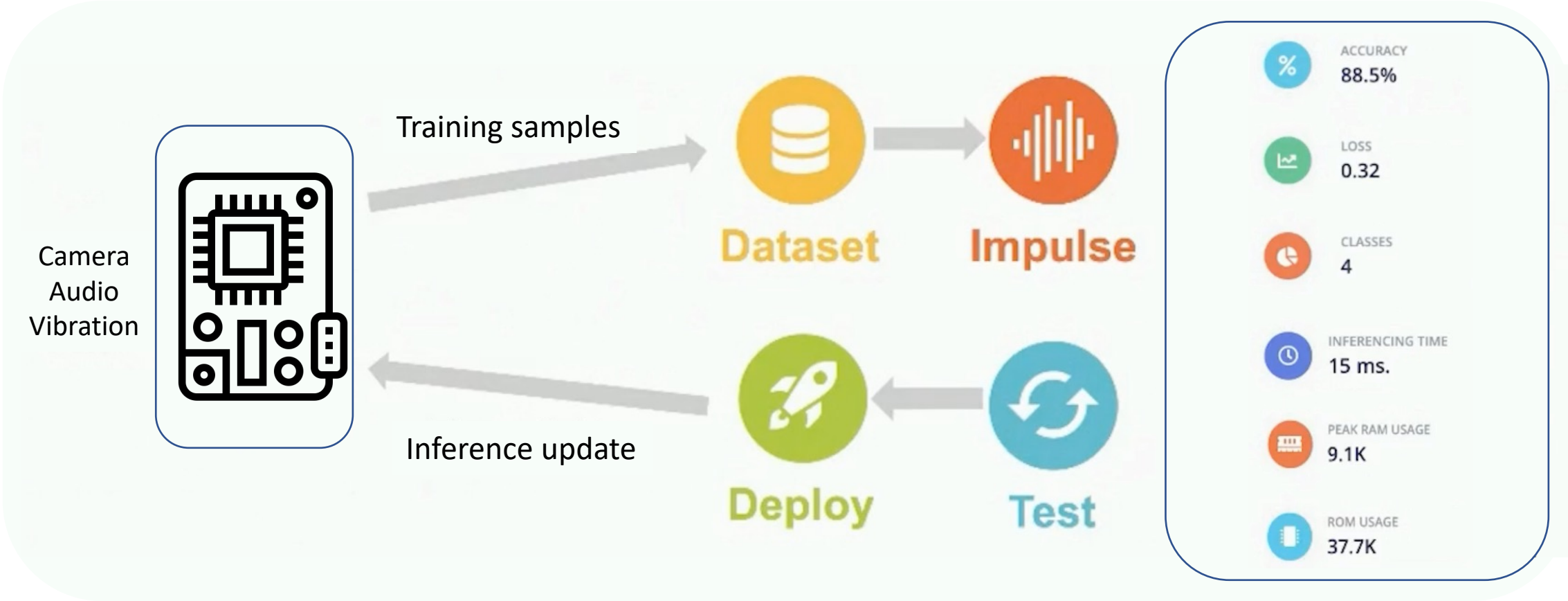
Raspberry Pi
(Cortex-A)

Jetson Nano
(Cortex-A + GPU)

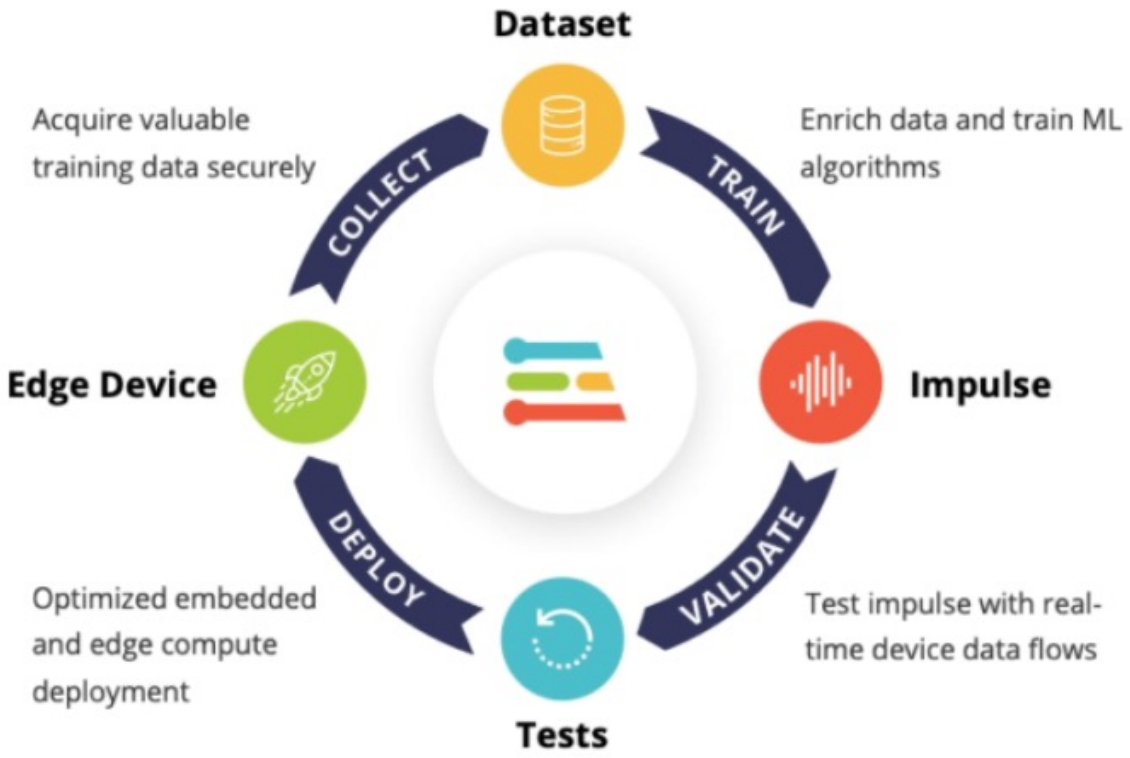
Sourse: Edge Impulse

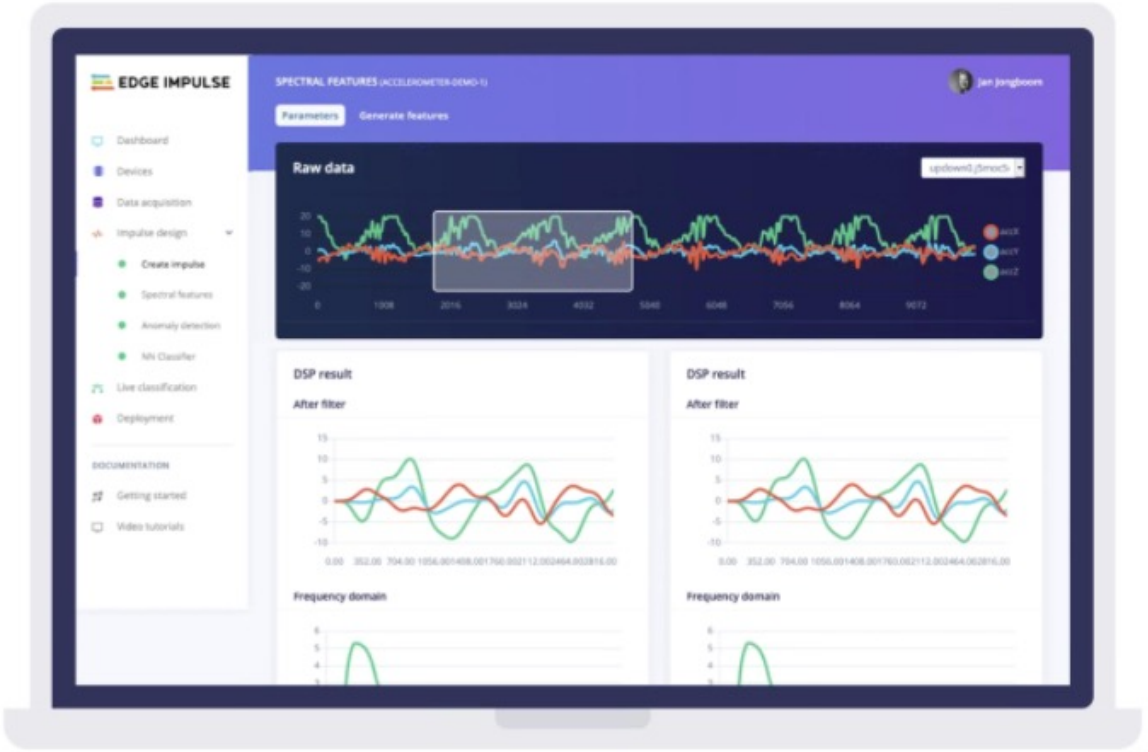
Data-driven engineering

Data-driven engineering

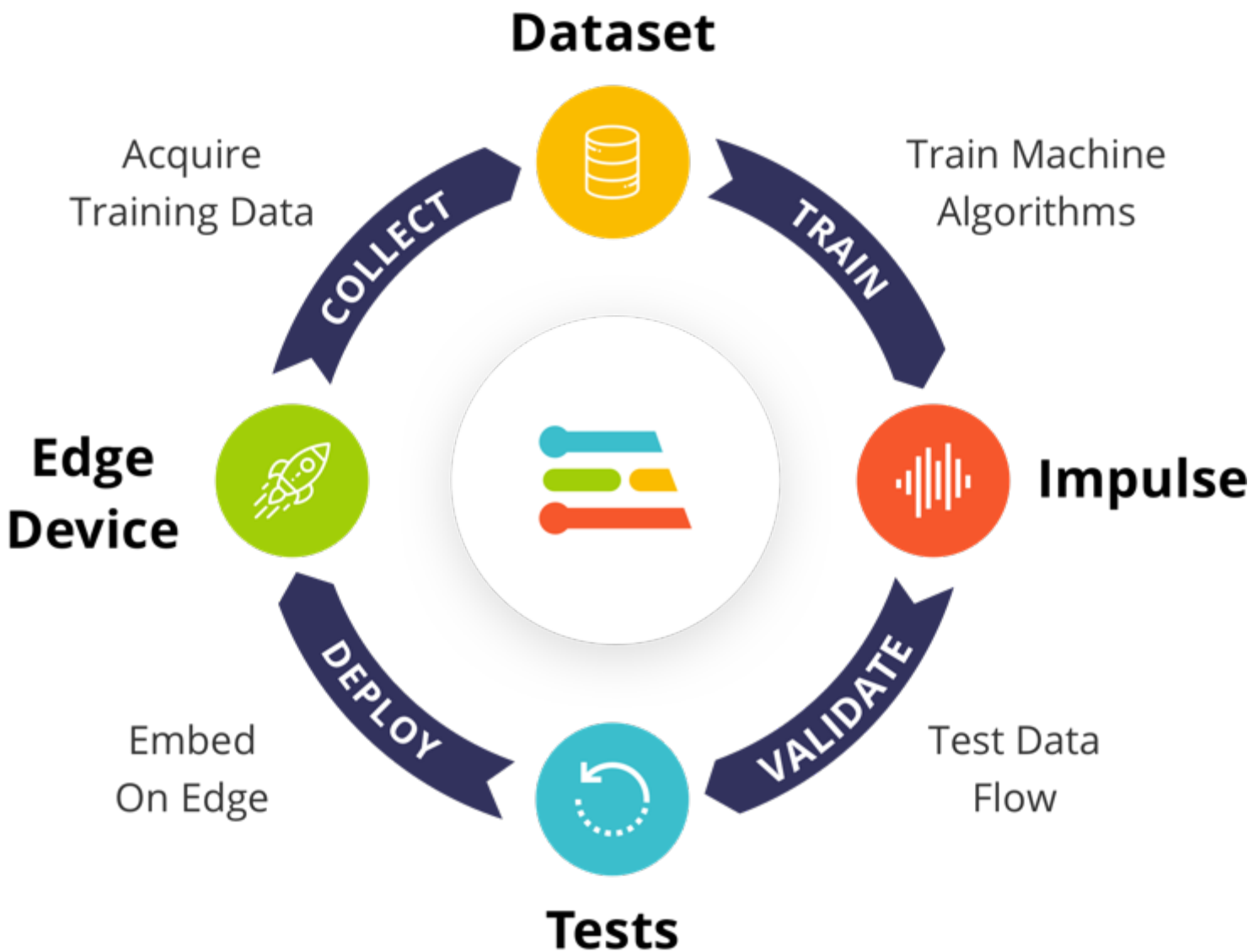


EI Studio - Embedded ML platform





Learn more at http://edgeimpulse.com

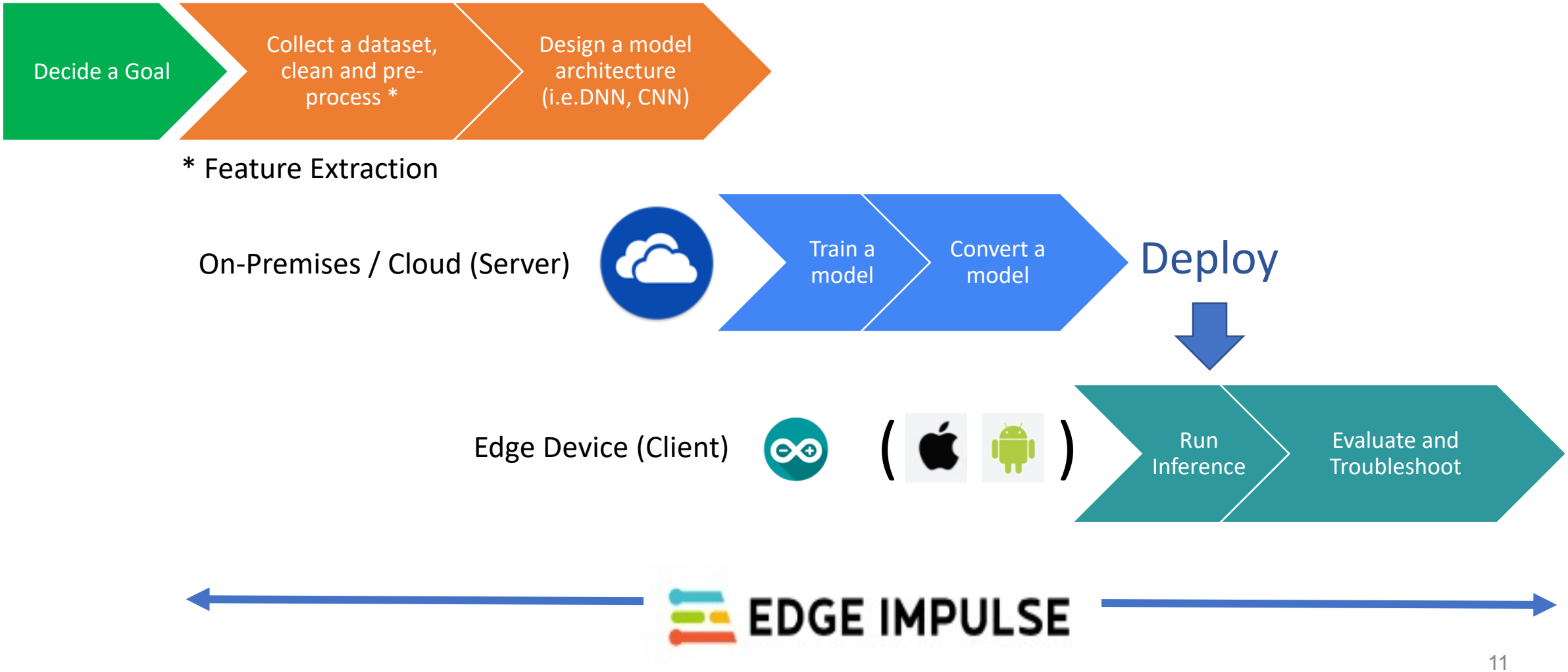


- Signal Processing
- Anomaly Detection
- Classification

Feature Extraction

with Edge Impulse Studio

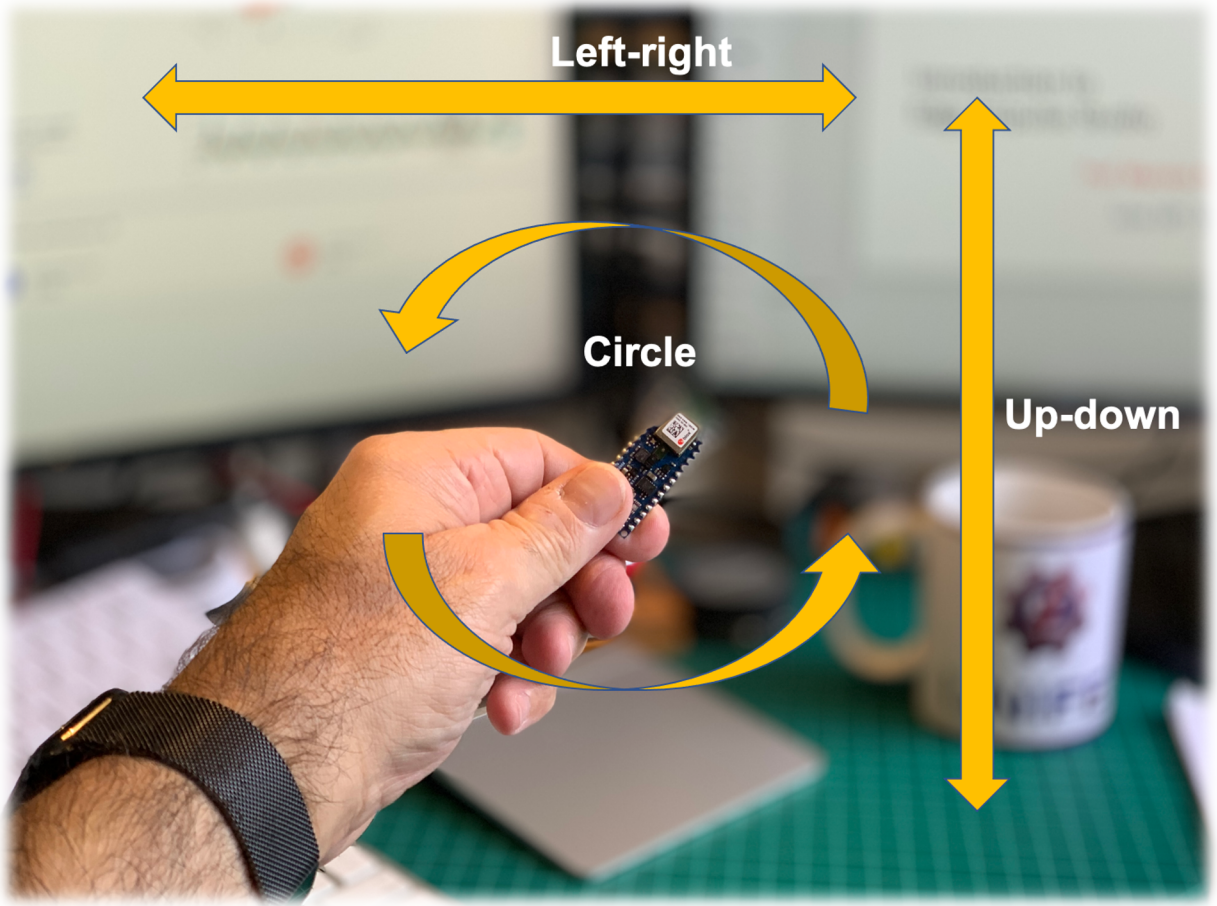
Deep Learning Workflow



Gesture Classification Project

Decide a Goal

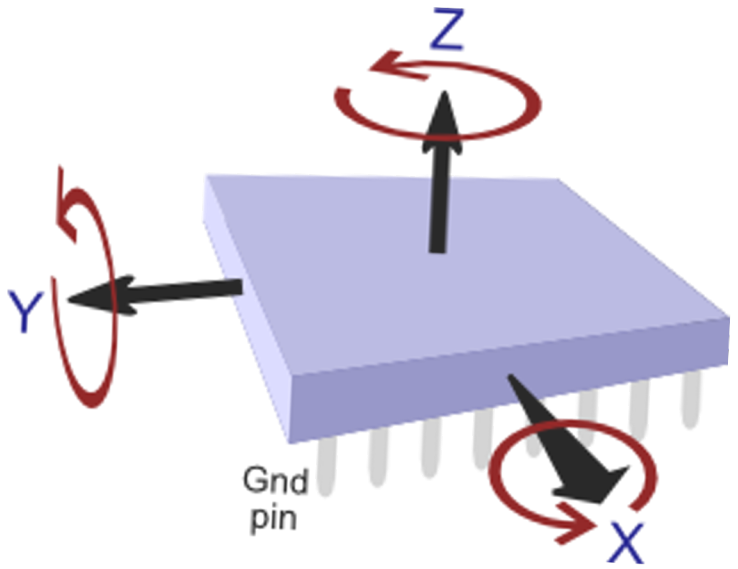
- Manual gestures:
  - Up-Down
  - Left-Right
  - Circle
  - Idle



Gesture Classification Project

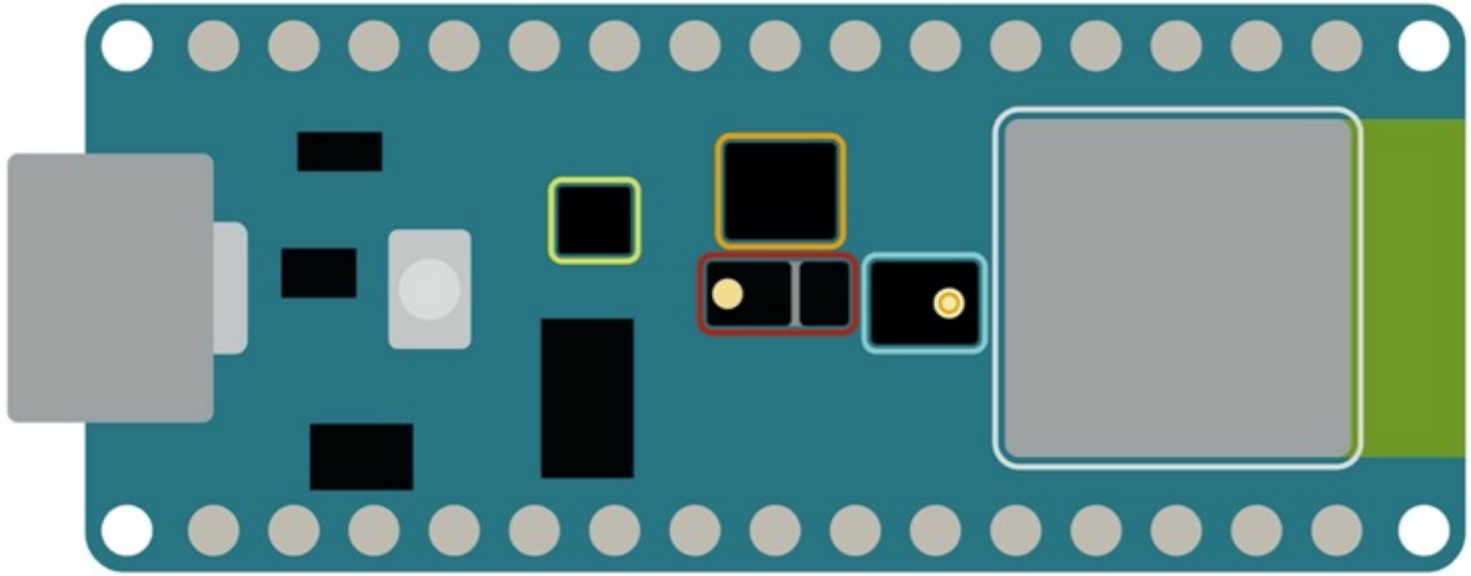
Decide a Goal

- Manual gestures (Classes) as:
  - Up-Down
  - Left-Right
  - Circle
  - Idle



- Data: collect & test using accelerometer as sensor
- Deploy in Arduino Sense

Nano 33 BLE SENSE



- Color, brightness, proximity and gesture sensor
- Digital microphone
- Motion, vibration and orientation sensor
- Temperature, humidity and pressure sensor
- Arm Cortex-M4 microcontroller and BLE module

Gesture Classification Project

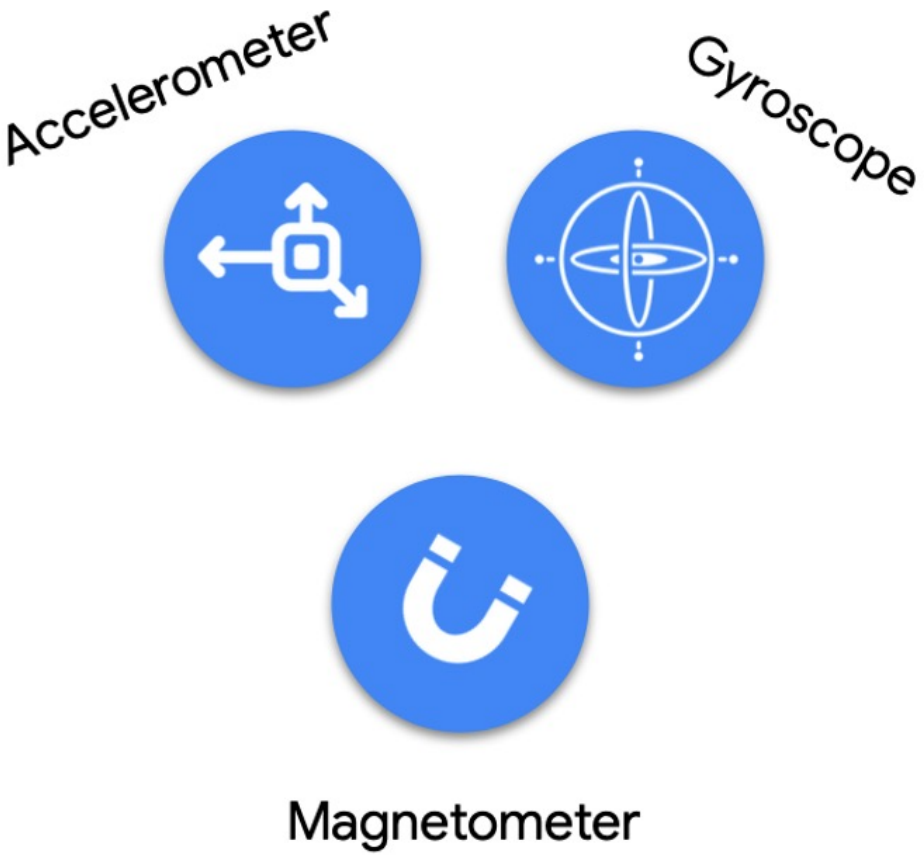
Decide a Goal

- Manual gestures (Classes) as:
  - Up-Down
  - Left-Right
  - Circle
  - Idle

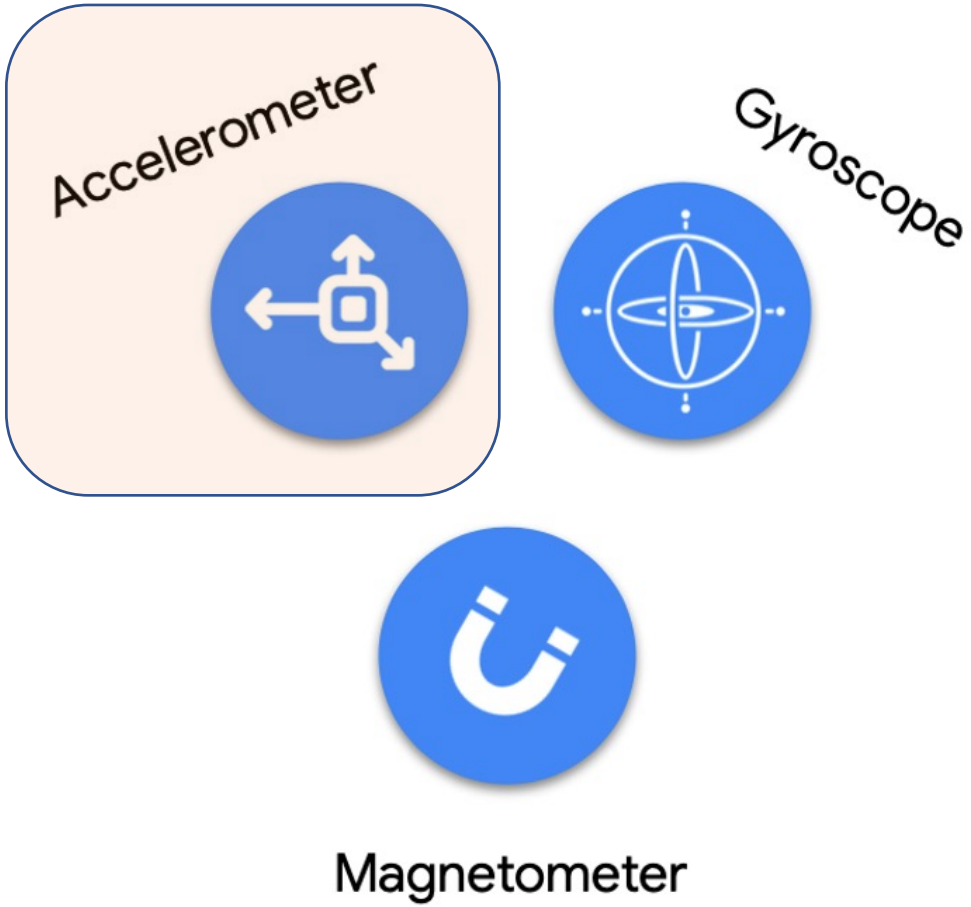
- Data: collect & test using accelerometer as sensor

➔ (Test model using Smart Phone and Edge Impulse)

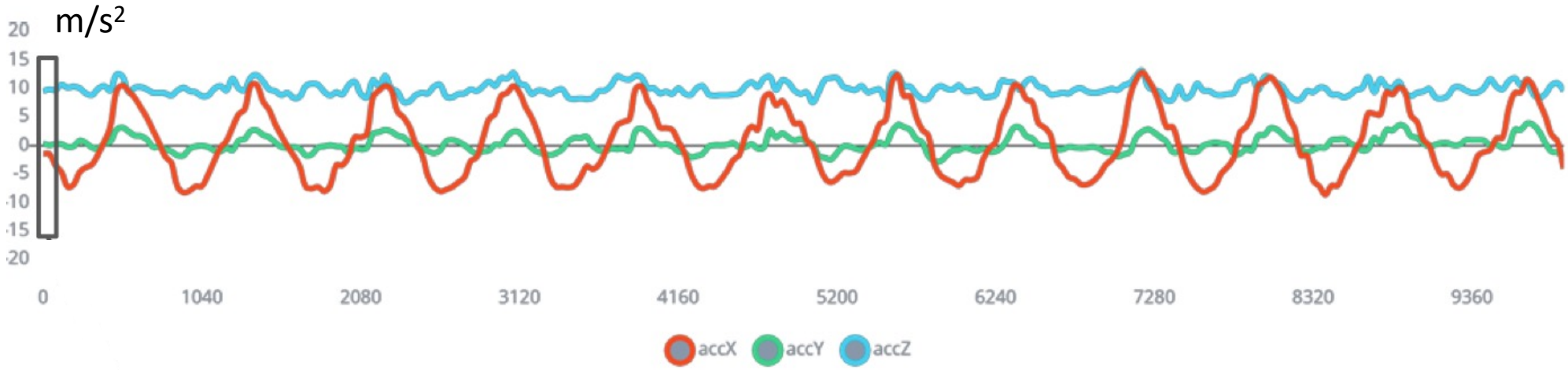
Inertial Measurement Unit (**IMU**) - 9-Axis

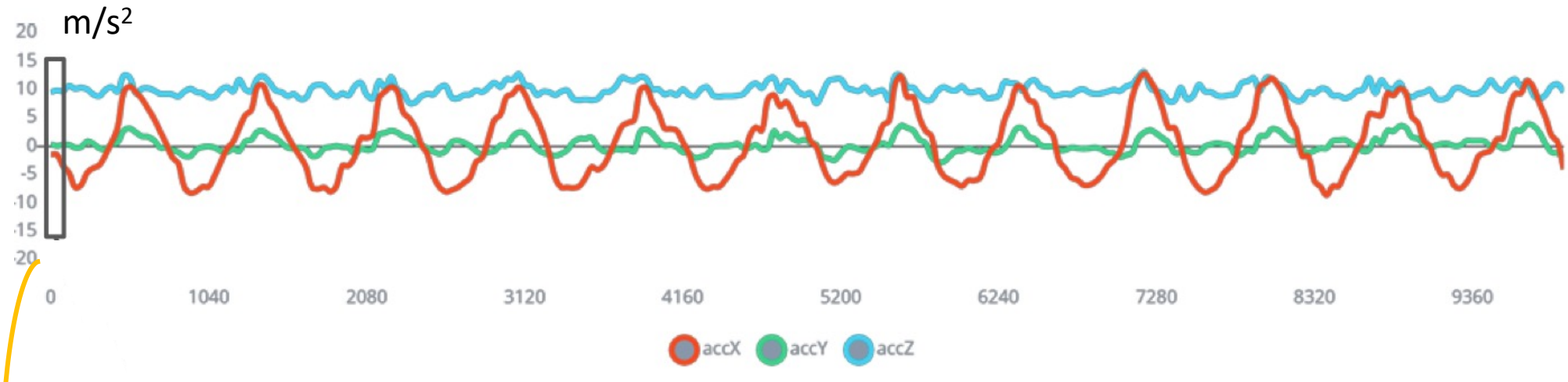


Inertial Measurement Unit (**IMU**)



Example: 10 seconds of data, capured with a sample rate: 62.5 Hz





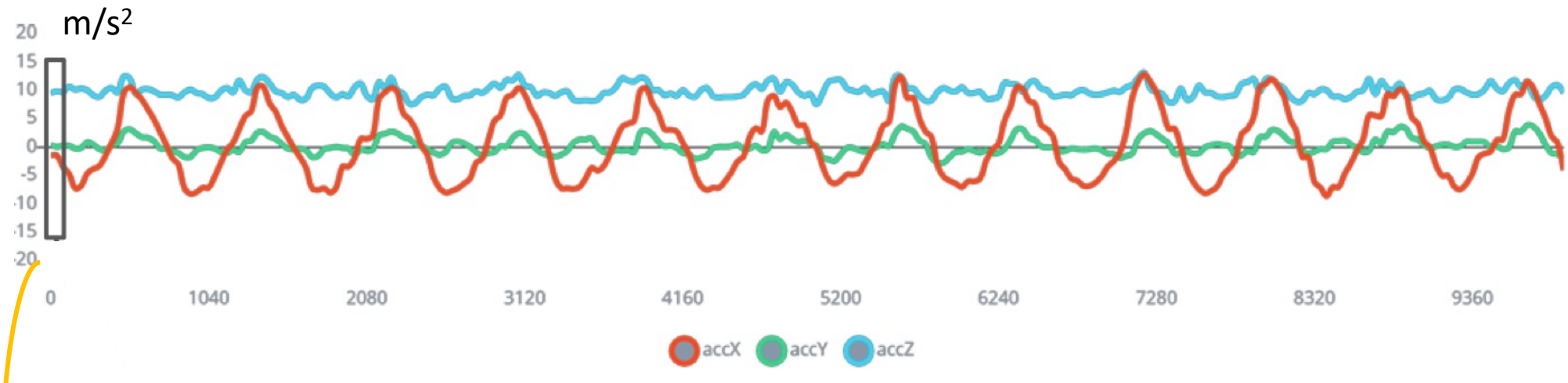
Raw Features

- accX
- accY
- accZ

Model

Classes

- Left-Right
- Up-Down
- Circle
- Iddle



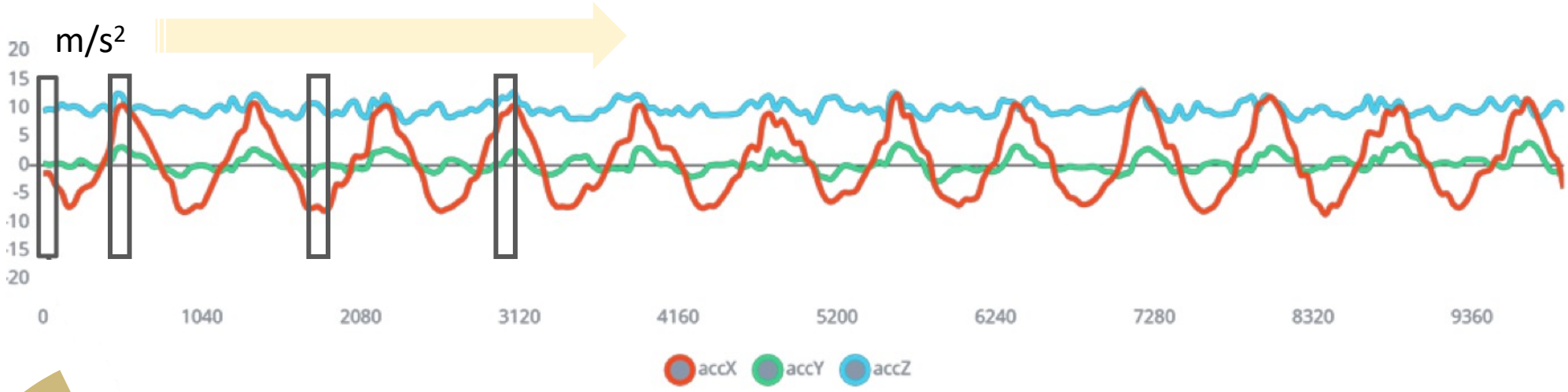
Raw Features

[-0.22, 0.01, 9.81]

Model

Classes

[1 0 0 0]



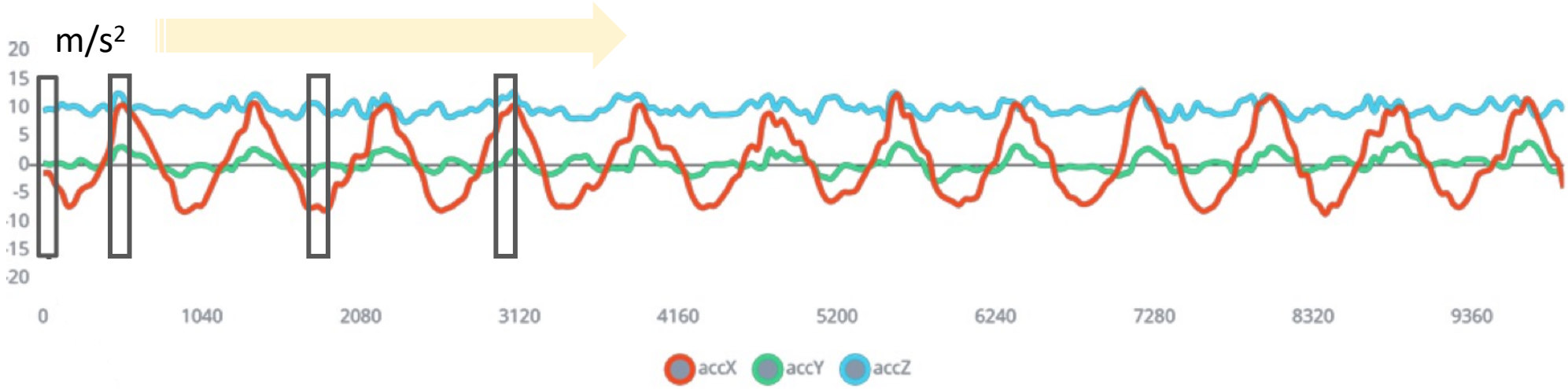
Raw Features

$$\begin{bmatrix} -0.22, 0.01, 9.81 \\ 9.74, 0.24, 9.95 \\ -7.22, 0.01, 9.83 \\ \dots \\ 8.72, 0.31, 9.89 \end{bmatrix}$$

Model

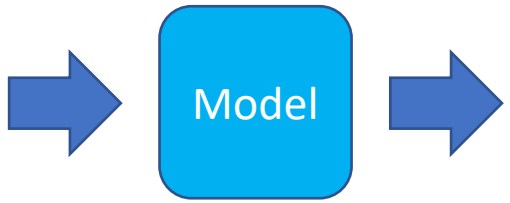
Classes

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Raw Features

$$\begin{bmatrix} -0.22, 0.01, 9.81 \\ 9.74, 0.24, 9.95 \\ -7.22, 0.01, 9.83 \\ \dots \\ 8.72, 0.31, 9.89 \end{bmatrix}$$

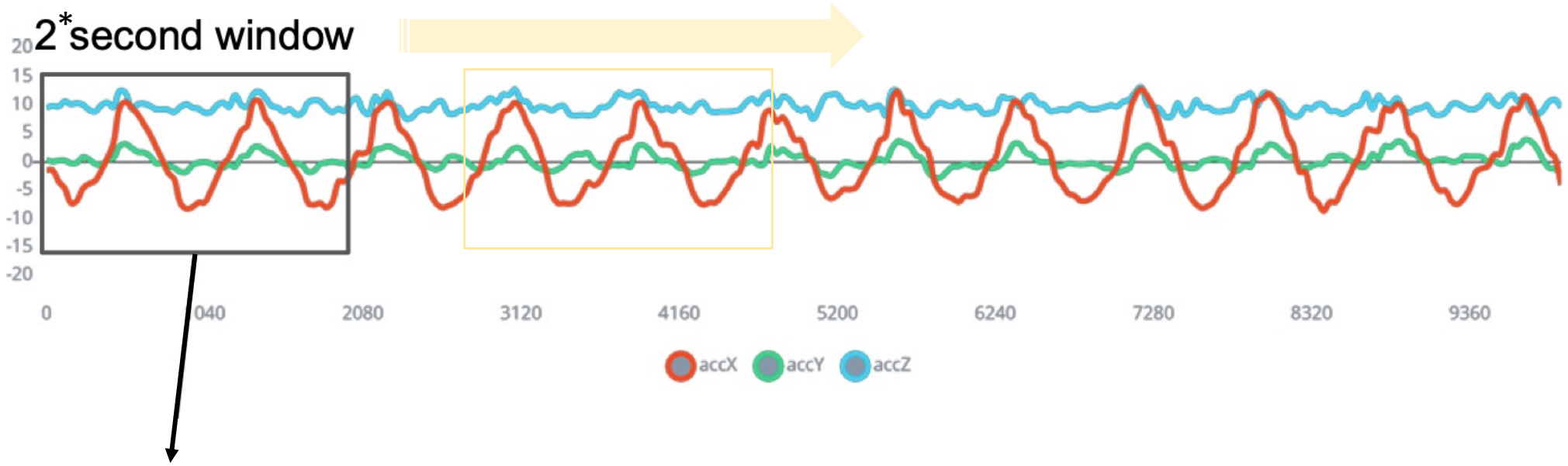


Classes

$$\begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix}$$

Problem!

✓Time-Series Data

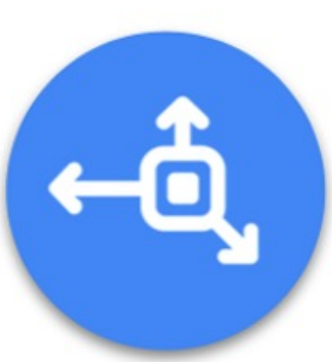


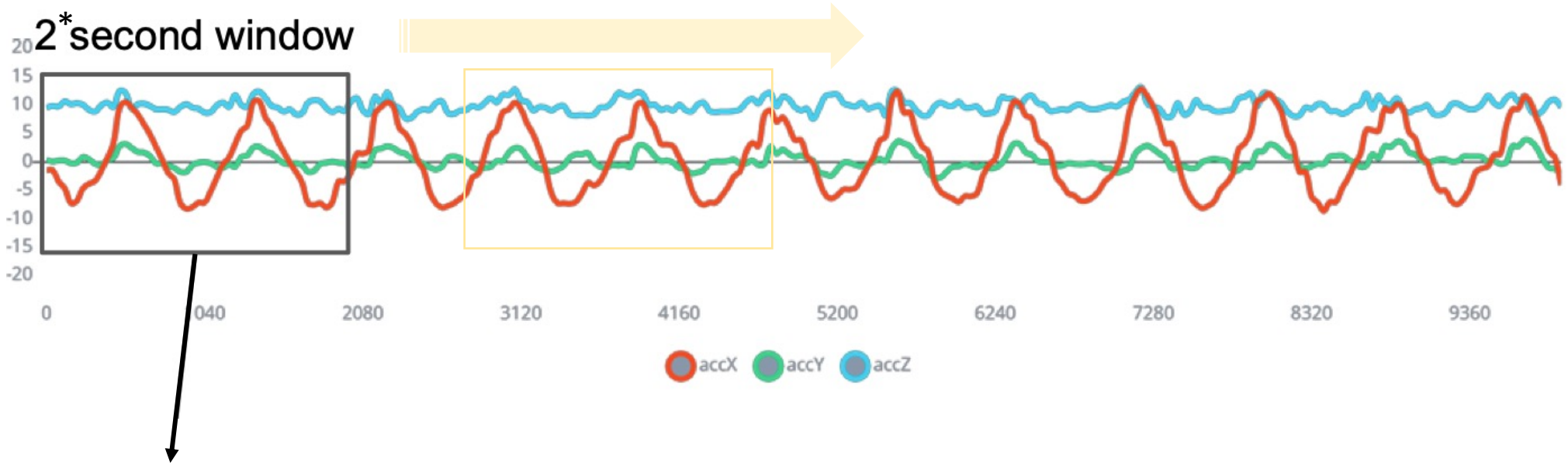
Raw Features as a window

- 125** samples for each axis
- 375 total features

* 2 seconds is needed to capture 1 or 2 cycles of movement

** 2 seconds at sample rate of 62.5 Hz ➔ 125 samples





Raw Features as a window

- 125** samples for each axis
- 375 total features

Automatic Feature Extraction using DL

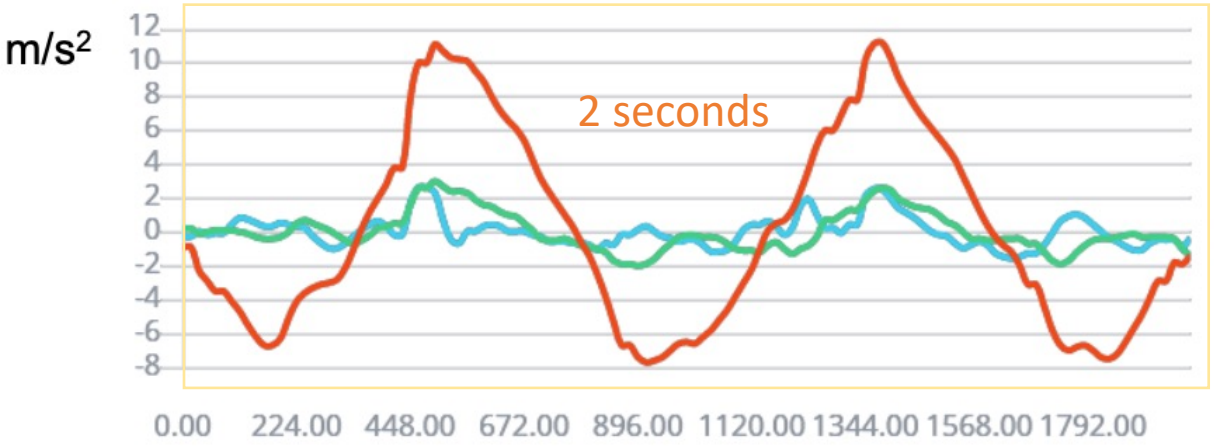
- Computational complexity ➔ More Memory
- Lots of training data

Problem!

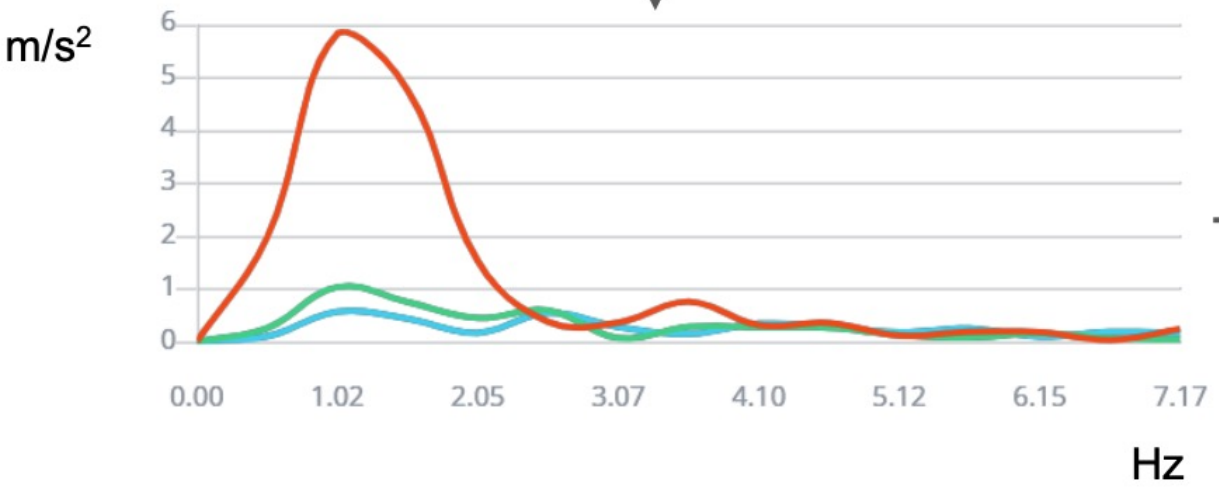
* 2 seconds is needed to capture 1 or 2 cycles of movement

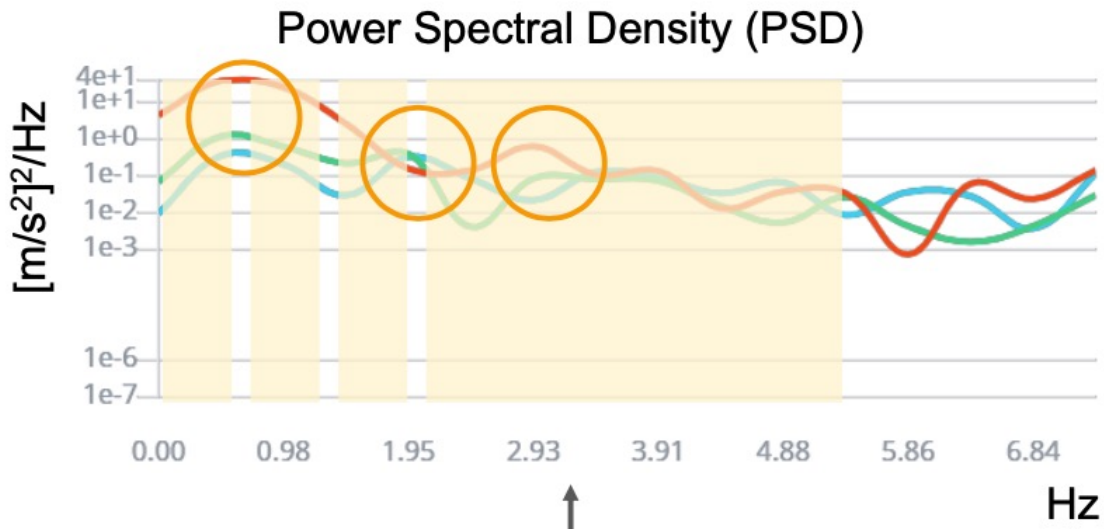
** 2 seconds at sample rate of 62.5 Hz ➔ 125 samples

Manual Feature Extraction



Feature extraction: FFT

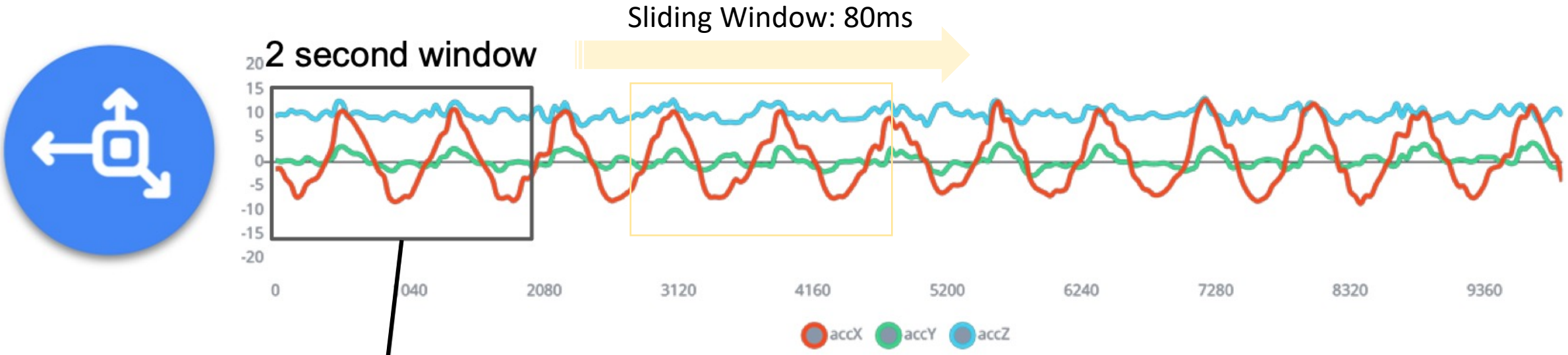




Feature extraction: PSD

4 bins
3 peaks

https://blog.endaq.com/why-the-power-spectral-density-psd-is-the-gold-standard-of-vibration-analysis



375 Raw Features

- Raw Data from sensor

Manual Feature Extraction

Spectral Analysis

33 Features

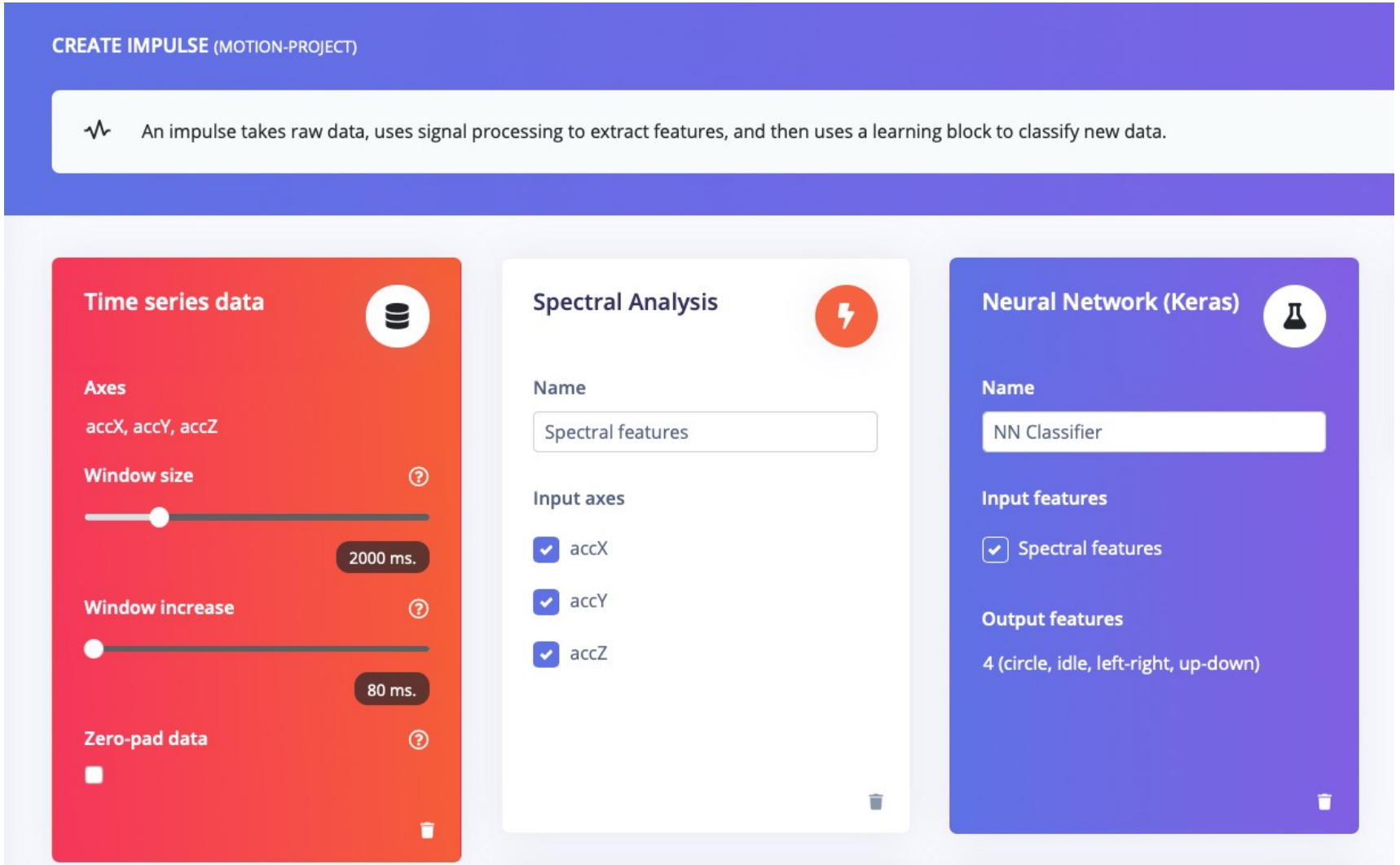
- RMS
- PSD

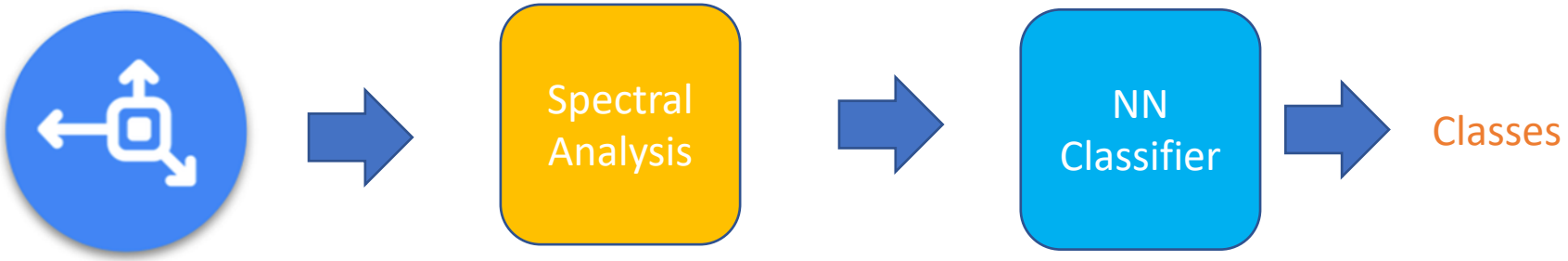
NN Classifier

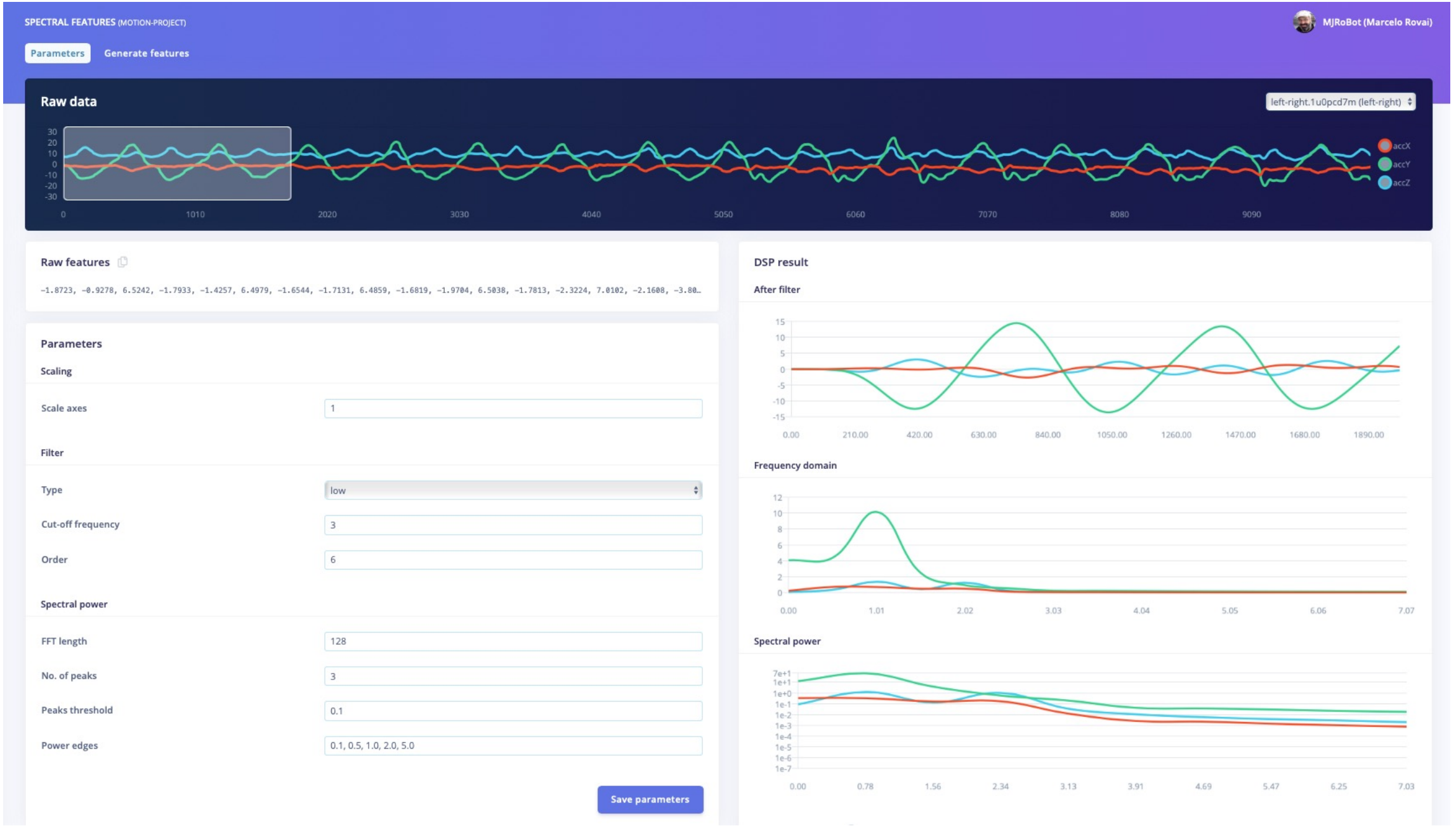
Classes

- Left-Right
- Up-Down
- Circle
- Iddle

An **impulse** takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data


CREATE IMPULSE (MOTION-PROJECT)
An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.
Time series data
Axes
accX, accY, accZ
Window size
2000 ms.
Window increase
80 ms.
Zero-pad data
Spectral Analysis
Name
Spectral features
Input axes
accX
accY
accZ
Neural Network (Keras)
Name
NN Classifier
Input features
Spectral features
Output features
4 (circle, idle, left-right, up-down)


Spectral Analysis
NN Classifier
Classes


SPECTRAL FEATURES (MOTION-PROJECT)
MJRoBot (Marcelo Rovai)
Parameters
Generate features
Raw data
left-right.1u0pcd7m (left-right)
accX
accY
accZ
Raw features
-1.8723, -0.9278, 6.5242, -1.7933, -1.4257, 6.4979, -1.6544, -1.7131, 6.4859, -1.6819, -1.9704, 6.5038, -1.7813, -2.3224, 7.0102, -2.1608, -3.80…
Parameters
Scaling
Scale axes
1
Filter
Type
low
Cut-off frequency
3
Order
6
Spectral power
FFT length
128
No. of peaks
3
Peaks threshold
0.1
Power edges
0.1, 0.5, 1.0, 2.0, 5.0
Save parameters
DSP result
After filter
Frequency domain
Spectral power

SPECTRAL FEATURES (MOTION-PROJECT)

MJRoBot (Marcelo Rovai)

Parameters | Generate features

Training set

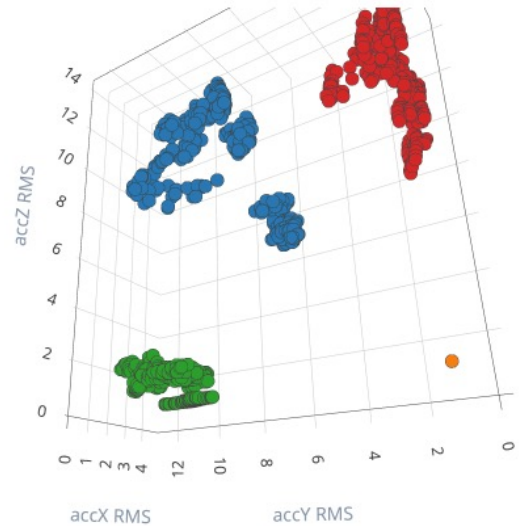
Data in training set	2m 40s
Classes	4 (circle, idle, left-right, up-down)
Window length	2000 ms.
Window increase	80 ms.
Training windows	1,616

Generate features

Feature explorer (1,616 samples)

X Axis: accX RMS | Y Axis: accY RMS | Z Axis: accZ RMS

- circle
- idle
- left-right
- up-down

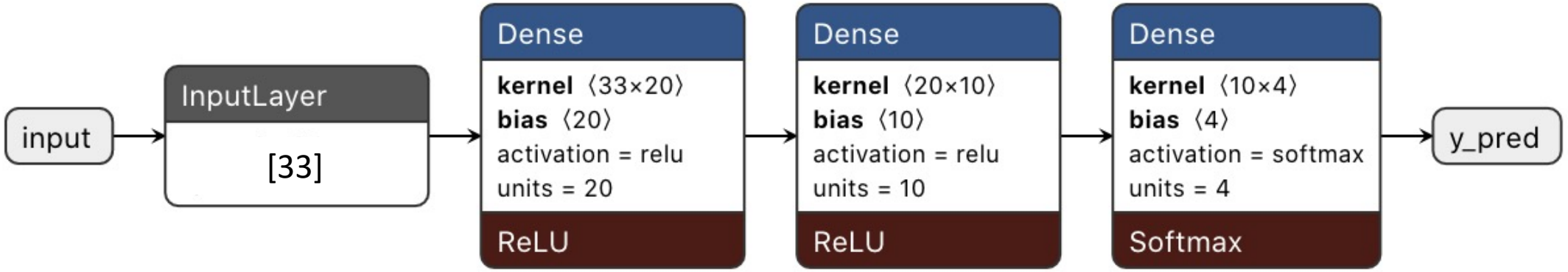


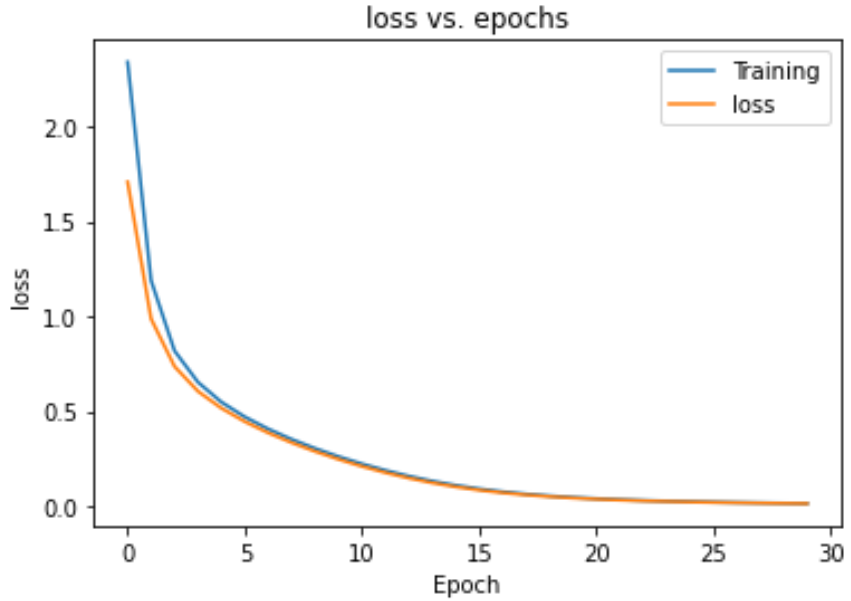
On-device performance

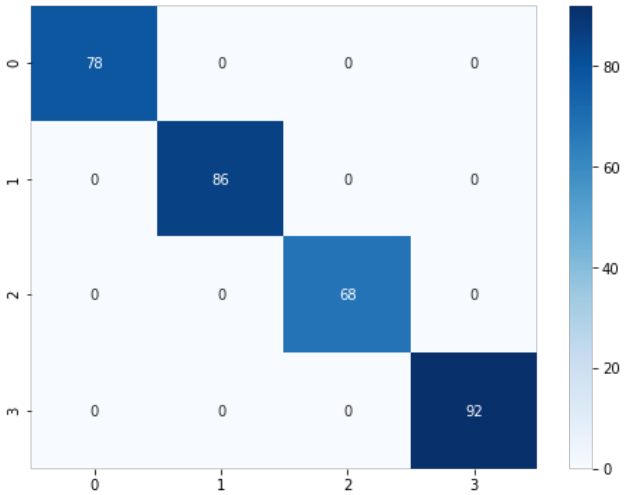
PROCESSING TIME
9 ms.

PEAK RAM USAGE
5 KB

Create and Train a DNN model







NN CLASSIFIER (MOTION-PROJECT)

#1 Click to set a description for this version

MJRoBot (Marcelo Rovai)

Neural Network settings

Training settings

Number of training cycles: 30

Learning rate: 0.0005

Minimum confidence rating: 0.60

Neural network architecture

Input layer (33 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (4 features)

Start training

Training output

Model

Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY
100.0%

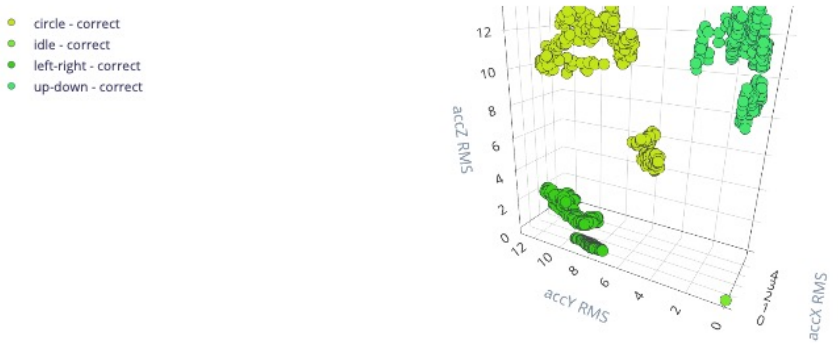
LOSS
0.01

Confusion matrix (validation set)

	CIRCLE	IDLE	LEFT-RIGHT	UP-DOWN
CIRCLE	100%	0%	0%	0%
IDLE	0%	100%	0%	0%
LEFT-RIGHT	0%	0%	100%	0%
UP-DOWN	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00

Feature explorer (full training set)

accX RMS | accY RMS | accZ RMS



On-device performance

INFERENCING TIME
1 ms.

PEAK RAM USAGE
1.5K

ROM USAGE
15.3K

Reading Material

Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning (Coursera)
- Text Book: "TinyML" by Pete Warden, Daniel Situnayake

I want to thank Shawn Hymel and Edge Impulse, Laurence Moroney from Google, Harvard professor Vijay Janapa Reddi, Ph.D. student Brian Plancher and their staff for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the TinyML4D, an initiative to make TinyML education available to everyone globally.

Thanks

And stay safe!