CEC Continuing Education Center

DesignNews

Machine Learning in Microcontrollers

DAY 4 : Training a Model for the STM32

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

THE SPEAKER

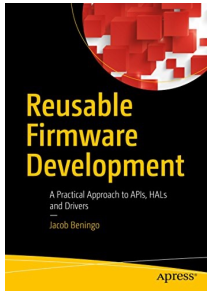
Jacob Beningo

Visit 'Lecturer Profile'

Beningo Embedded Group - President

Focus: Embedded Software Consulting and Training

Specializes in creating and promoting embedded software **excellence** in businesses around the world.

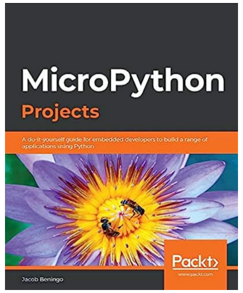


Reusable Firmware Development

A Practical Approach to APIs, HALs and Drivers

Jacob Beningo

Apress

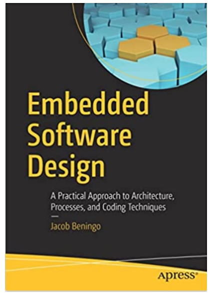


MicroPython Projects

A do-it-yourself guide for embedded developers to build a range of applications using Python

Packt> www.packt.com

Jacob Beningo



Embedded Software Design

A Practical Approach to Architecture, Processes, and Coding Techniques

Jacob Beningo

Apress

Blogs for:

- DesignNews.com
- Embedded.com
- EmbeddedRelated.com
- MLRelated.com

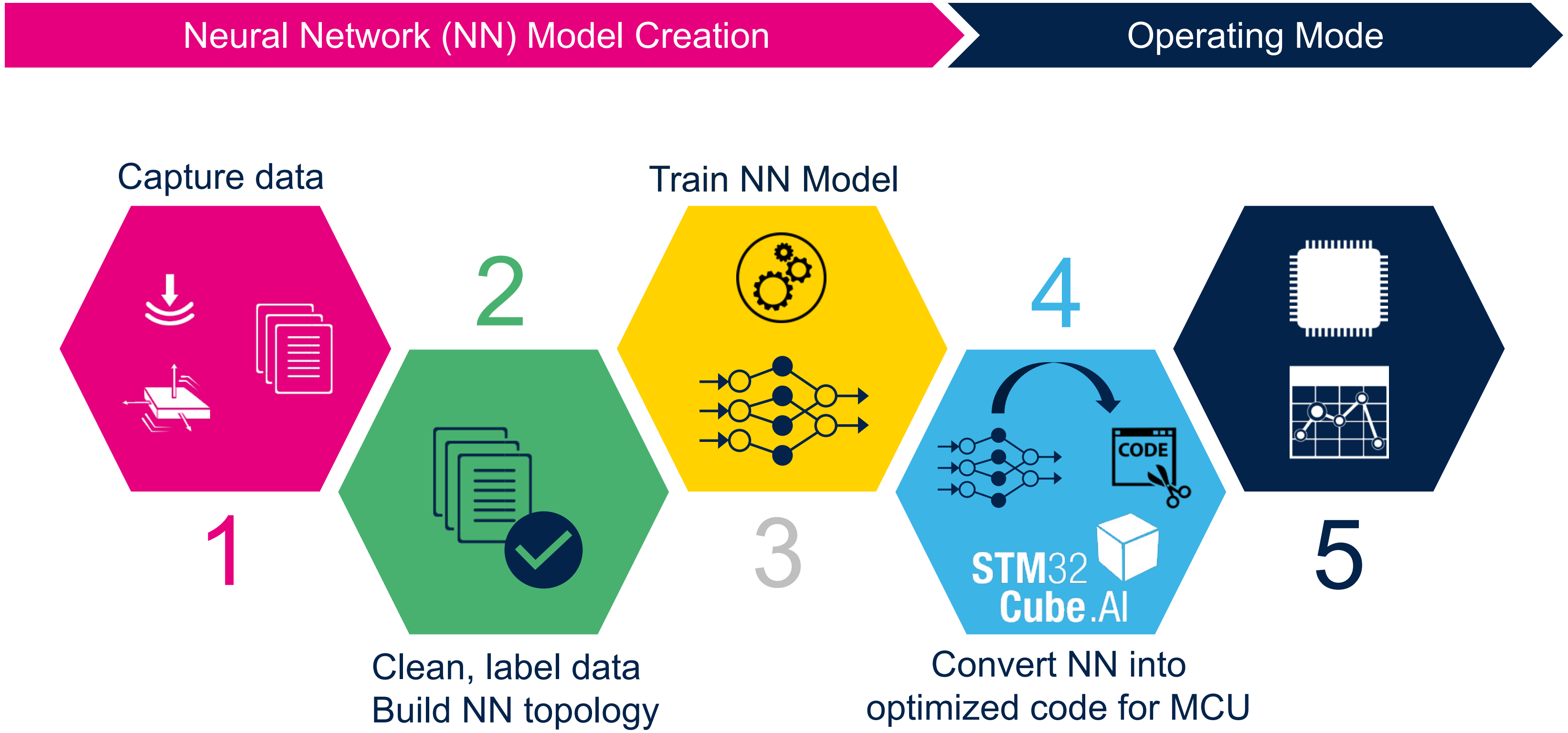
Visit www.beningo.com to learn more …

Course Sessions

- AI and ML for Microcontrollers
- Writing Embedded Software with ChatGPT and Open.AI
- Tools for Machine Learning in Microcontrollers
- **Training a Model for the STM32**
- Deploying Machine Learning Models

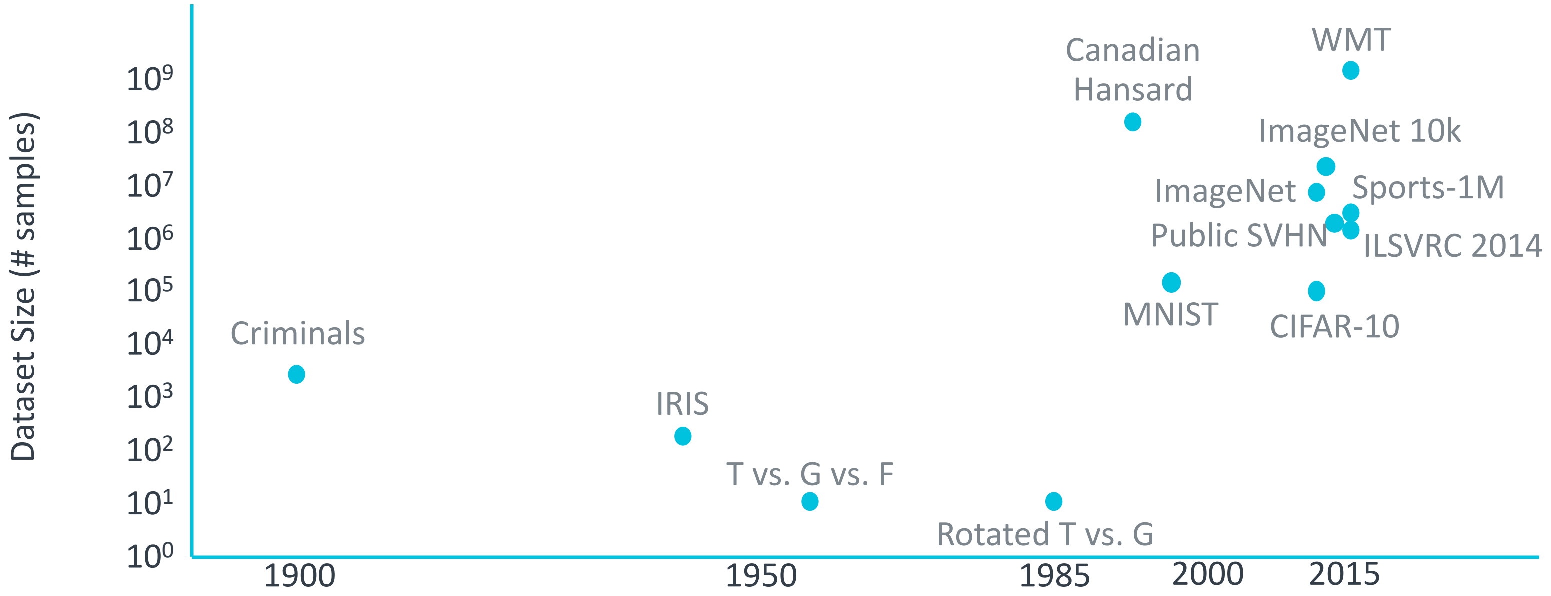
1 Capturing, Cleaning, and Labeling Data

Capturing, Cleaning, and Labeling Data



Capturing, Cleaning, and Labeling Data

1) Online Data Sets
2) Generate it
3) Collect it
4) Buy it



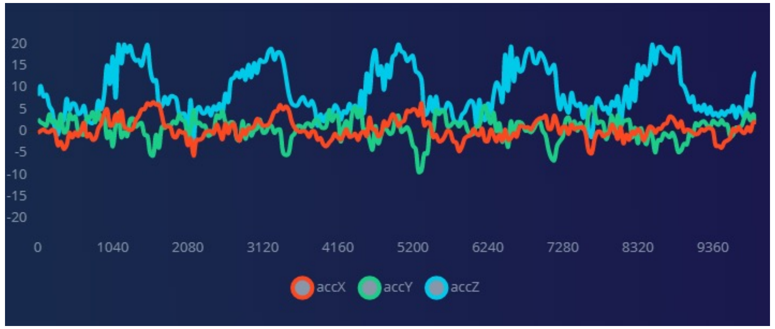
Collecting it is the most interesting … (and the most work)

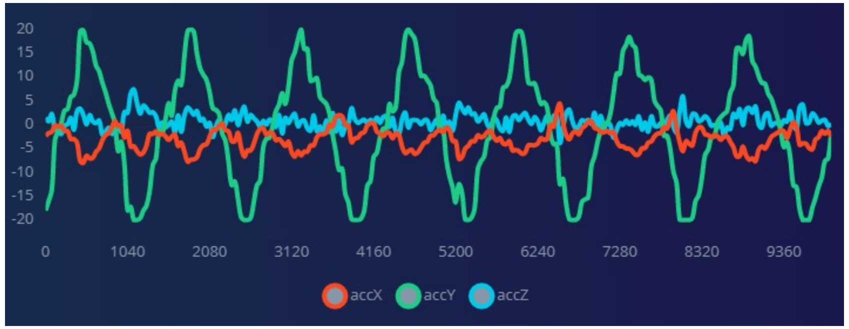
Example Application – Gesture Classification

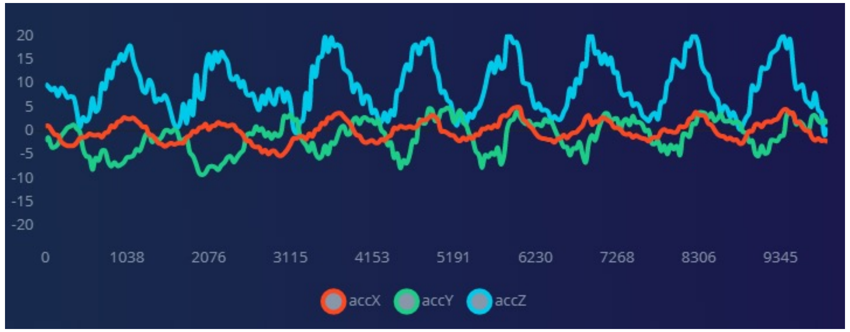
Label 1 – Up and Down

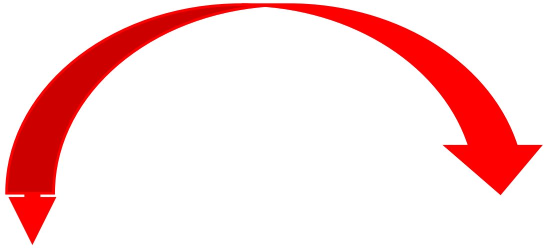
Label 2 – Wave

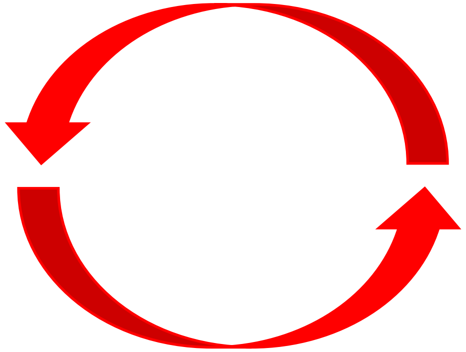
Label 3 – Circle











How do you plan to get the data for your own applications?

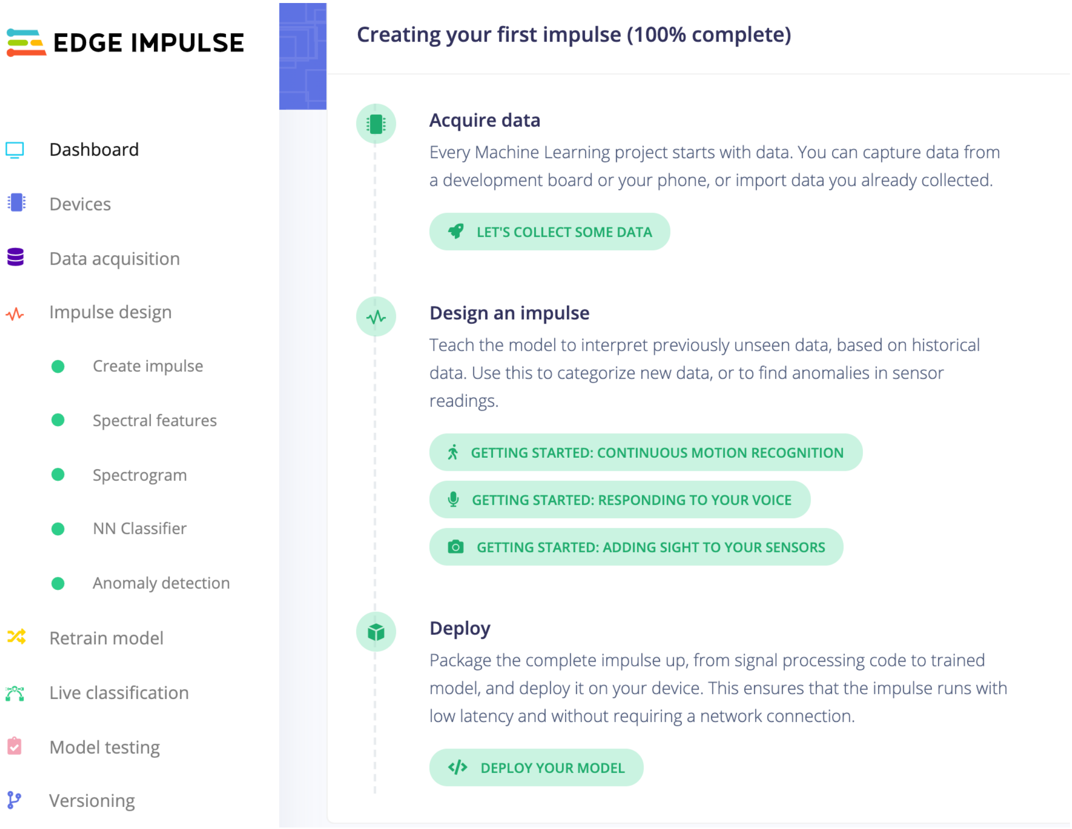
- Online Data Sets
- Generate it (through simulation)
- Collect it (experimentation)
- Buy it

2 Collecting Data with Edge Impulse

Edge Impulse

Edge Impulse was designed for software developers, engineers and domain experts to solve real problems using machine learning on edge devices without a PhD in machine learning.

www.edgeimpulse.com



Creating your first impulse (100% complete)

Acquire data

Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.

LET'S COLLECT SOME DATA

Design an impulse

Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.

GETTING STARTED: CONTINUOUS MOTION RECOGNITION

GETTING STARTED: RESPONDING TO YOUR VOICE

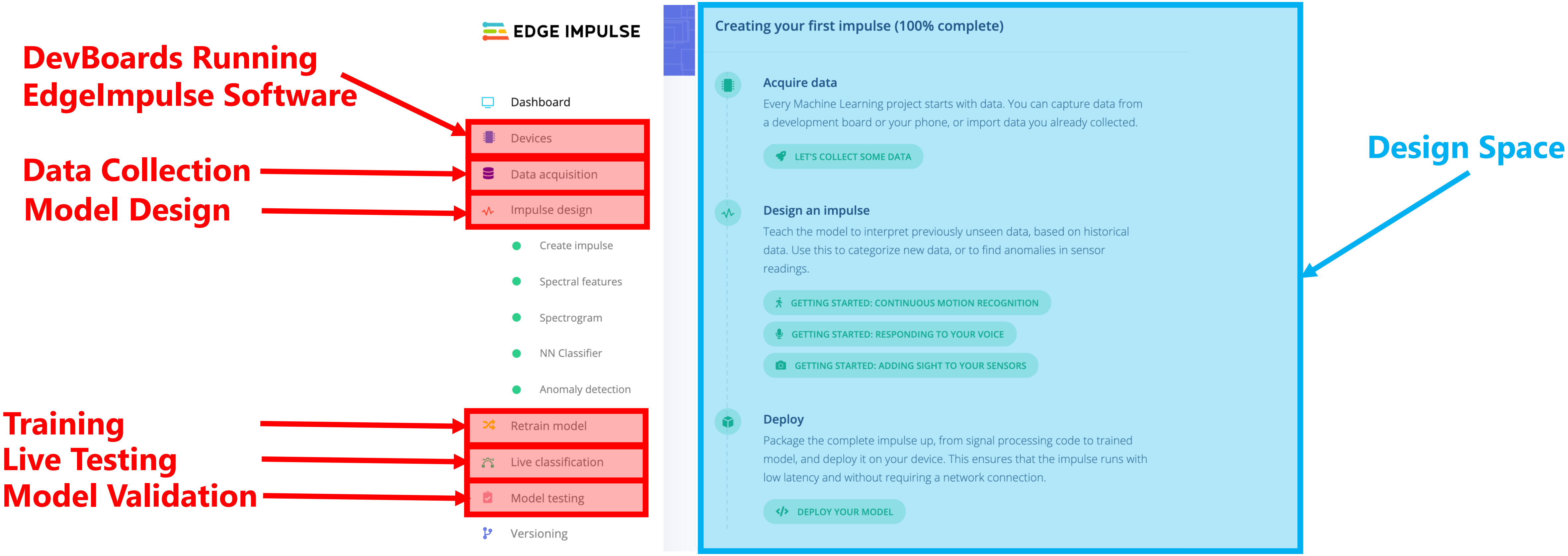
GETTING STARTED: ADDING SIGHT TO YOUR SENSORS

Deploy

Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with low latency and without requiring a network connection.

DEPLOY YOUR MODEL

Edge Impulse



DevBoards Running EdgeImpulse Software

Data Collection

Model Design

Training

Live Testing

Model Validation

Design Space

EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
  - Spectral features
  - Spectrogram
  - NN Classifier
  - Anomaly detection
- Retrain model
- Live classification
- Model testing
- Versioning

Creating your first impulse (100% complete)

Acquire data

Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.

LET'S COLLECT SOME DATA

Design an impulse

Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.

GETTING STARTED: CONTINUOUS MOTION RECOGNITION

GETTING STARTED: RESPONDING TO YOUR VOICE

GETTING STARTED: ADDING SIGHT TO YOUR SENSORS

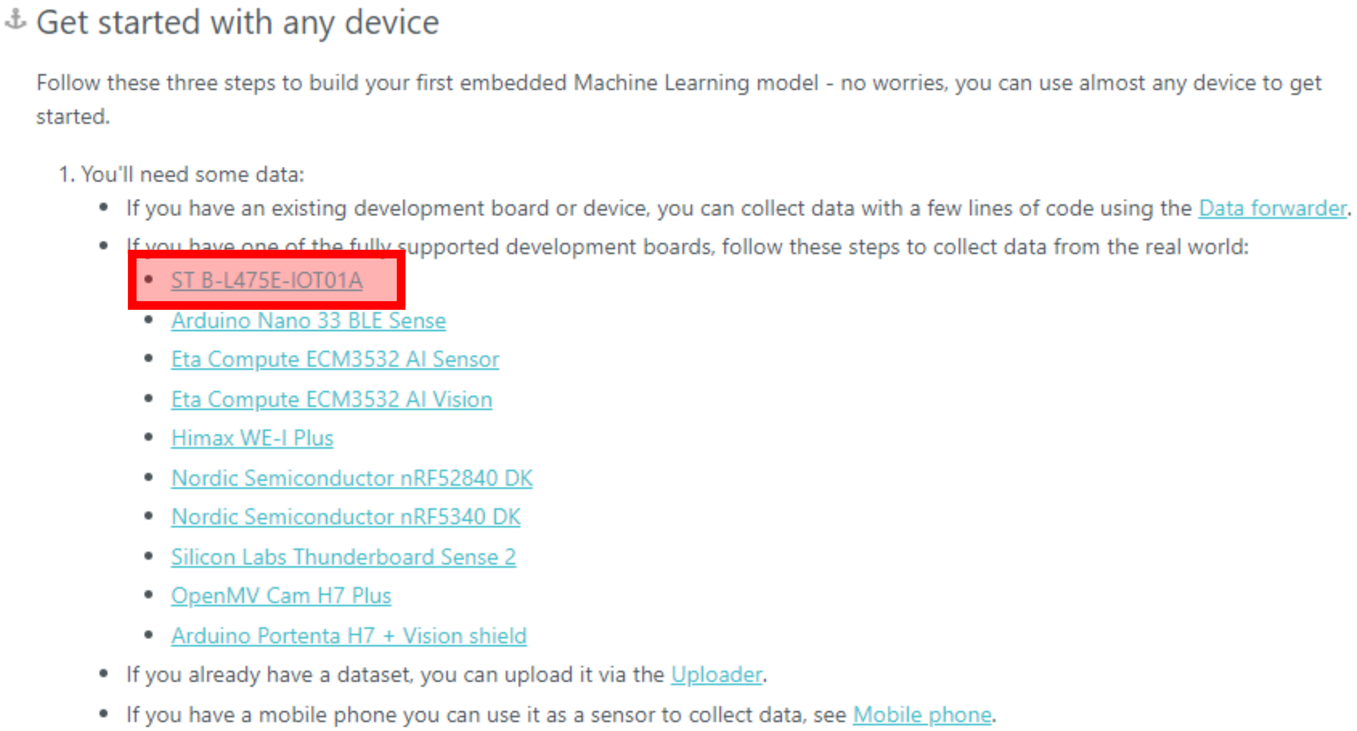
Deploy

Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with low latency and without requiring a network connection.

DEPLOY YOUR MODEL

Edge Impulse – Board Setup

Visit https://docs.edgeimpulse.com/docs

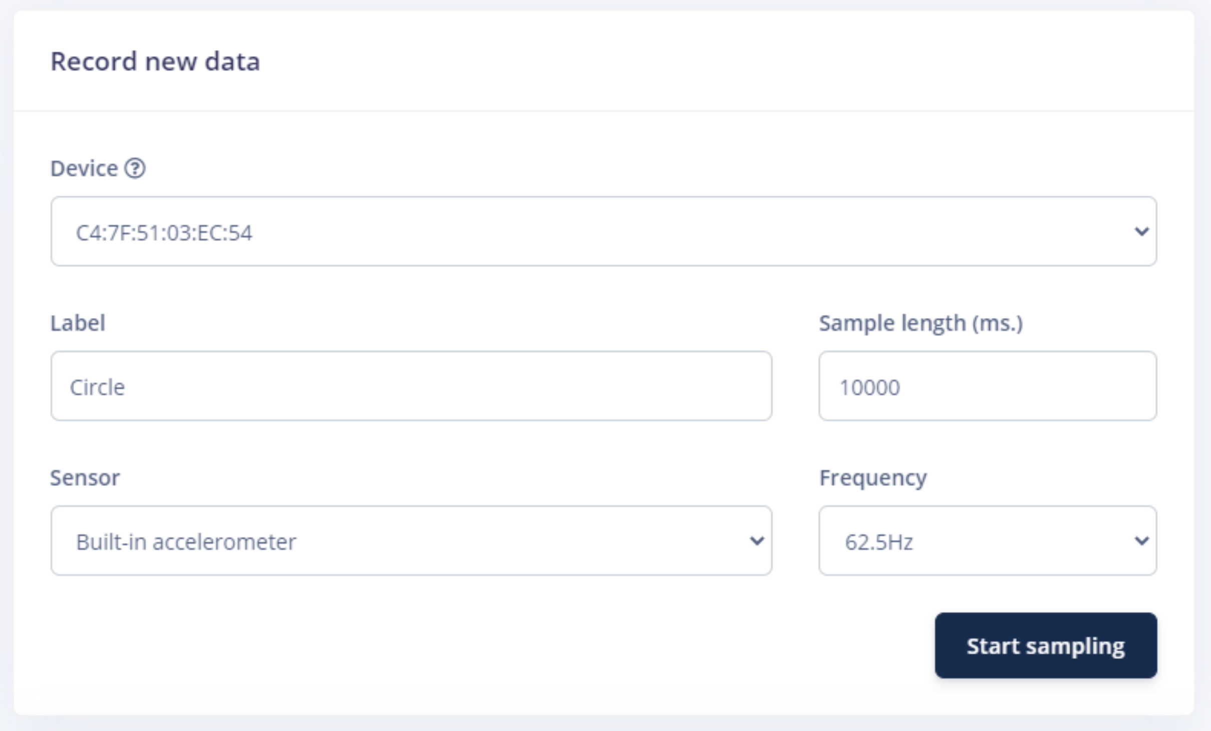


Get started with any device

Follow these three steps to build your first embedded Machine Learning model - no worries, you can use almost any device to get started.

1. You'll need some data:
   - If you have an existing development board or device, you can collect data with a few lines of code using the Data forwarder.
   - If you have one of the fully supported development boards, follow these steps to collect data from the real world:
     - ST B-L475E-IOT01A
     - Arduino Nano 33 BLE Sense
     - Eta Compute ECM3532 AI Sensor
     - Eta Compute ECM3532 AI Vision
     - Himax WE-I Plus
     - Nordic Semiconductor nRF52840 DK
     - Nordic Semiconductor nRF5340 DK
     - Silicon Labs Thunderboard Sense 2
     - OpenMV Cam H7 Plus
     - Arduino Portenta H7 + Vision shield
   - If you already have a dataset, you can upload it via the Uploader.
   - If you have a mobile phone you can use it as a sensor to collect data, see Mobile phone.

Capturing and Labeling Data



Record new data

Device ⓘ

C4:7F:51:03:EC:54

Label

Circle

Sample length (ms.)

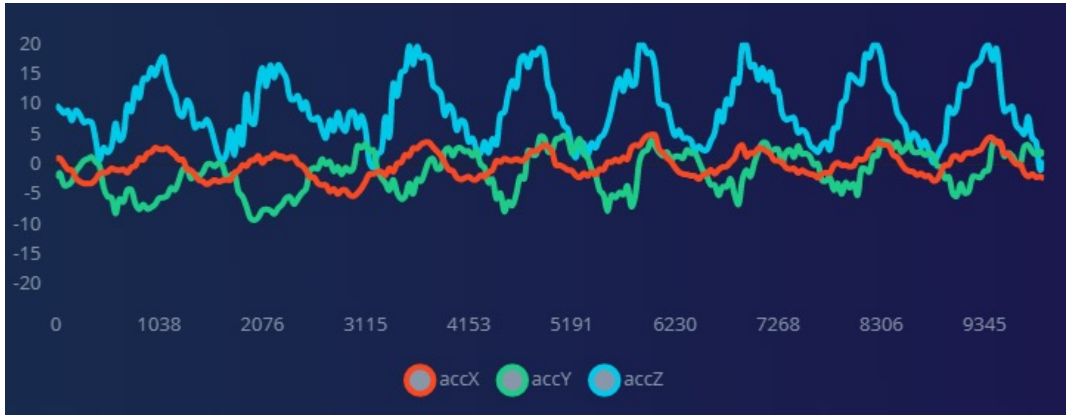
10000

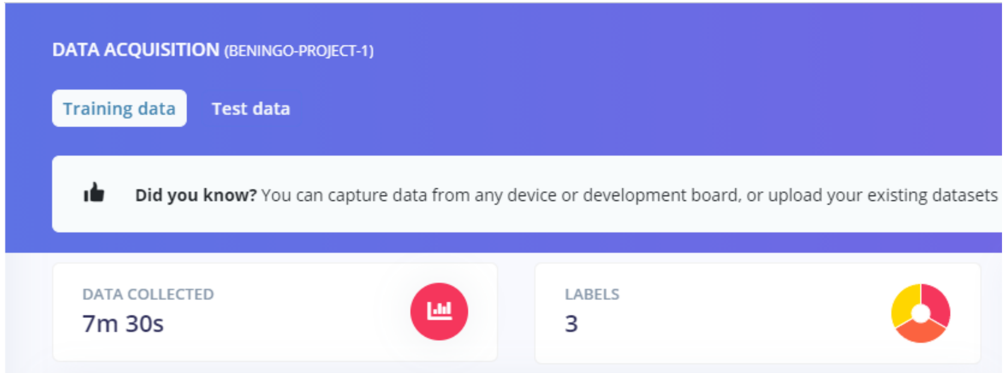
Sensor

Built-in accelerometer

Frequency

62.5Hz





DATA ACQUISITION (BENINGO-PROJECT-1)

Did you know? You can capture data from any device or development board, or upload your existing datasets

DATA COLLECTED
7m 30s

LABELS
3

What type of ML project are you most likely to work on in the near future?
- Hobby project
- Work project
- Other

3 Training a Model

Training a Model Overview

Time Series Parameter Setup

Feature Analysis

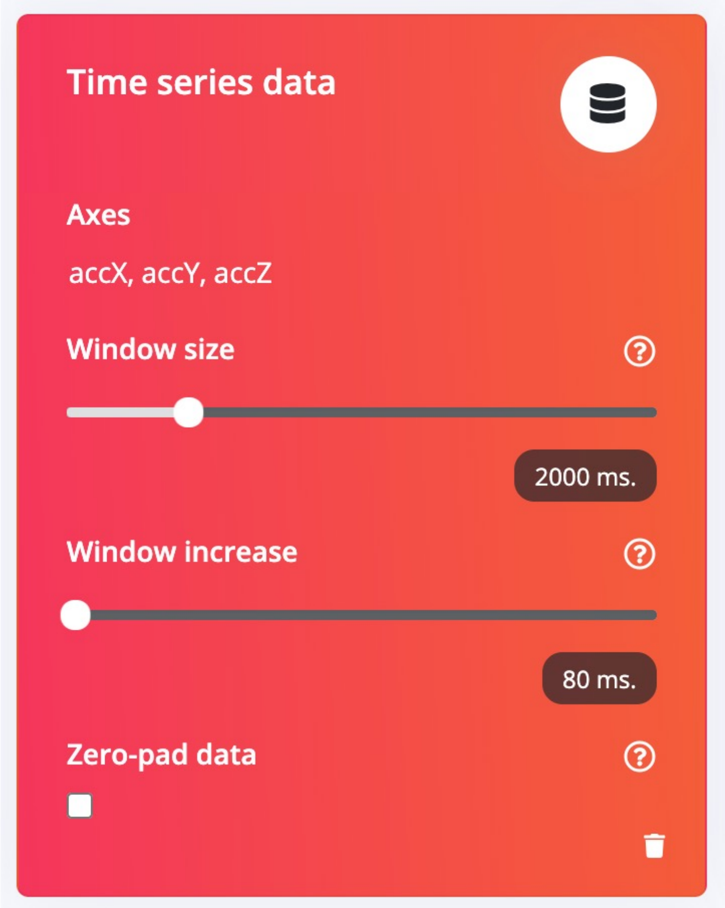
Neural Network Design

Training

Model Validation

Deploy

Training a Model – Impulse Design



Time series data

Axes

accX, accY, accZ

Window size

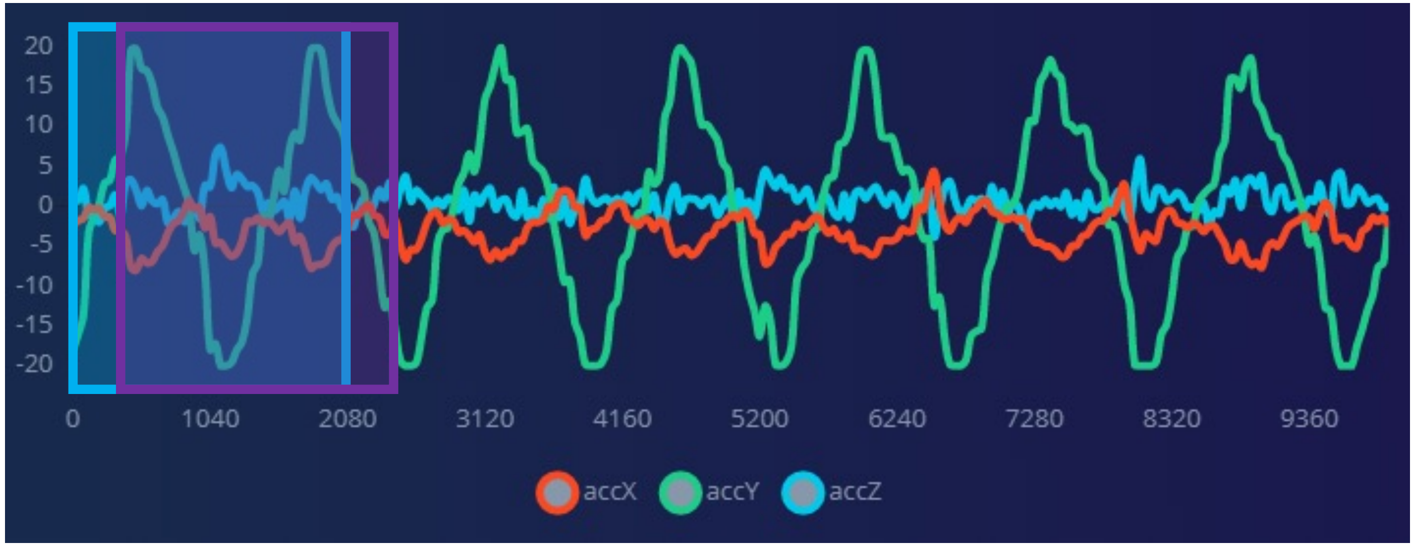
2000 ms.

Window increase

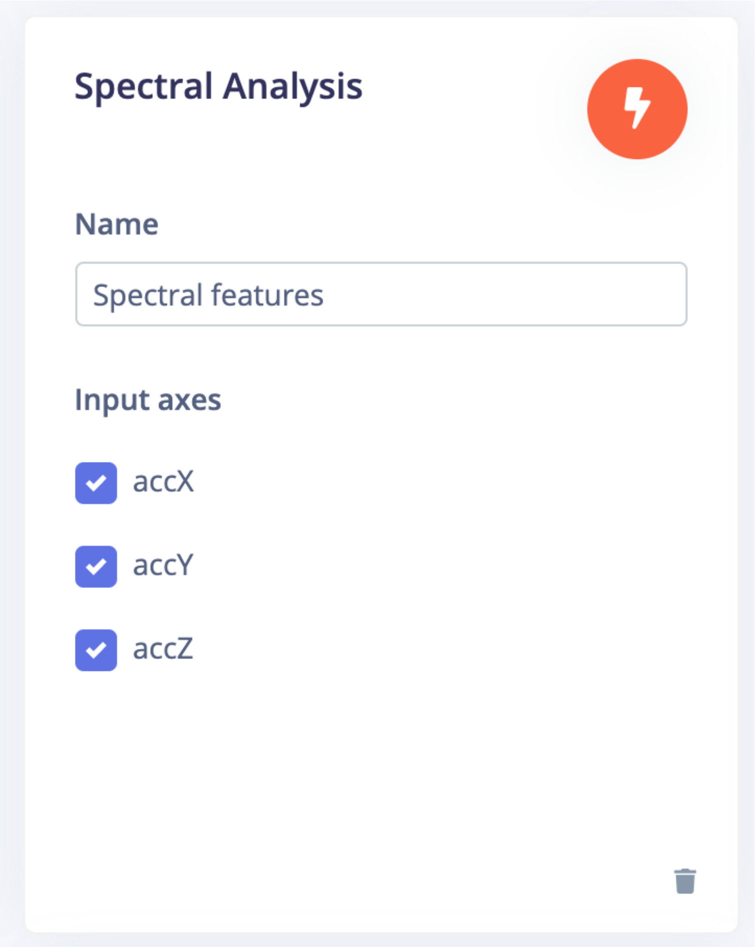
80 ms.

Zero-pad data

□



Training a Model Impulse Design



Spectral Analysis

Name

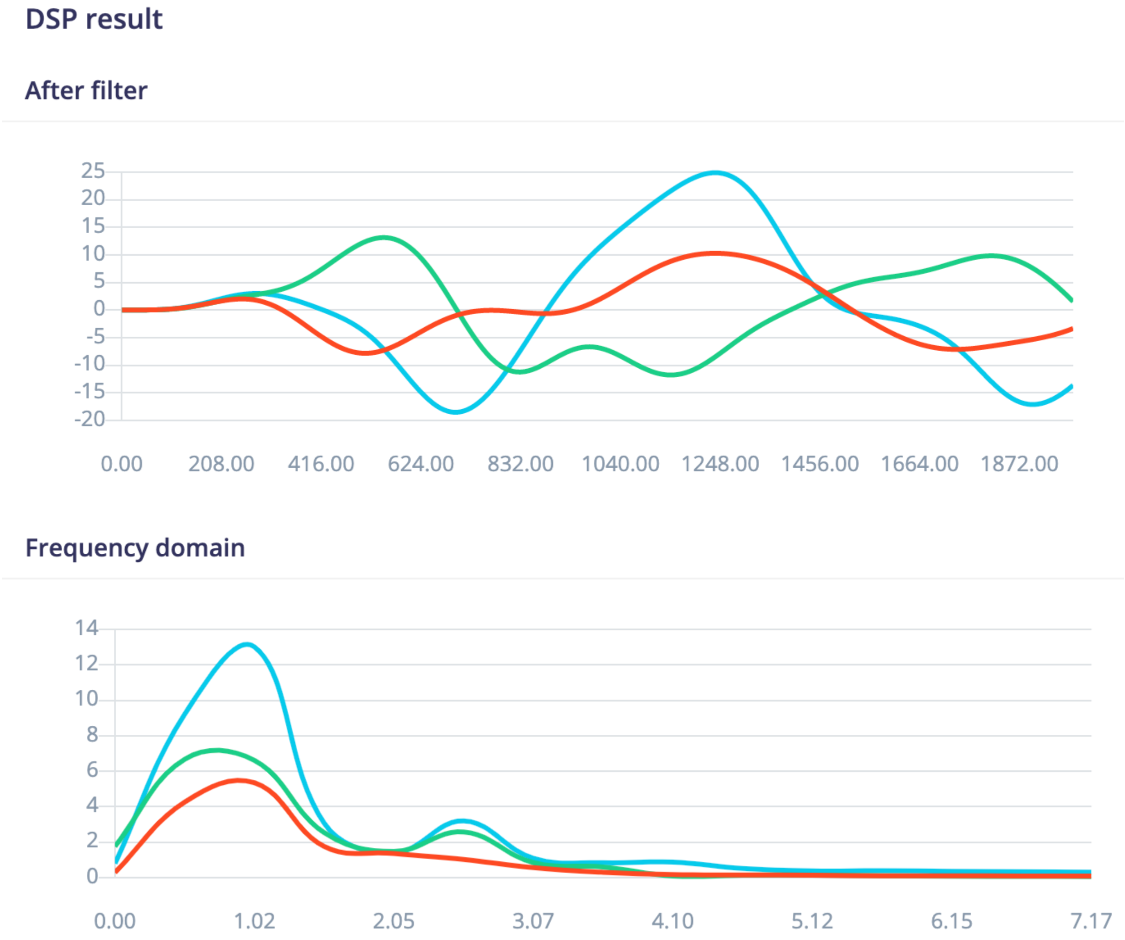
Spectral features

Input axes

☑ accX

☑ accY

☑ accZ

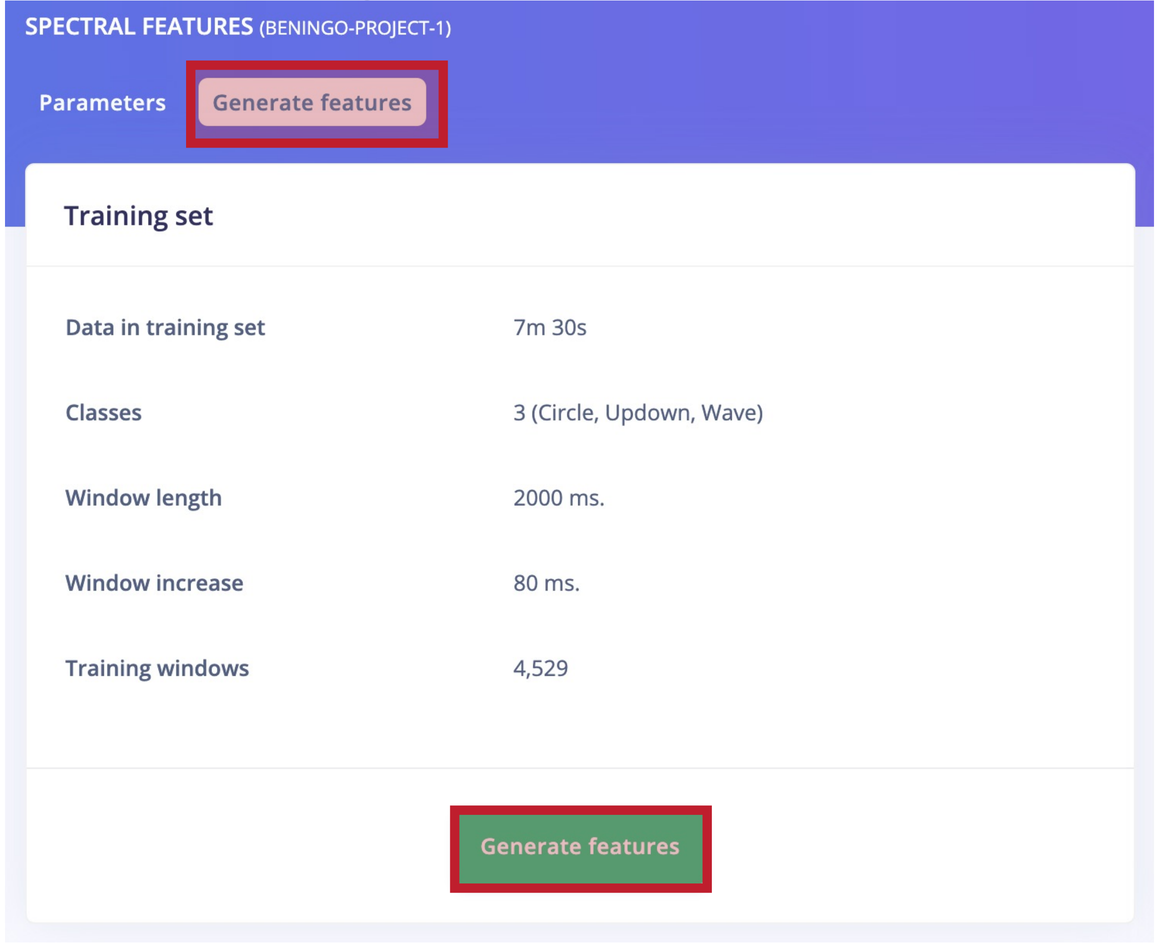


DSP result

After filter

Frequency domain

Training a Model – Impulse Design

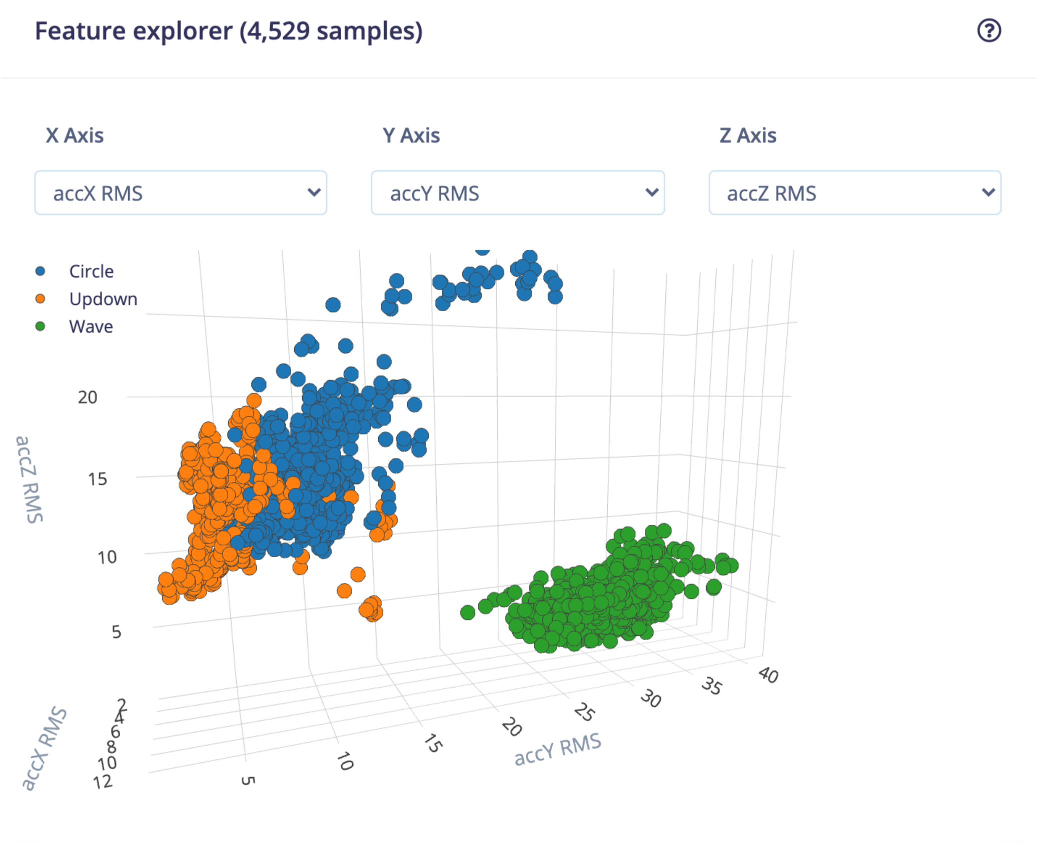


SPECTRAL FEATURES (BENINGO-PROJECT-1)

Parameters | Generate features

Training set

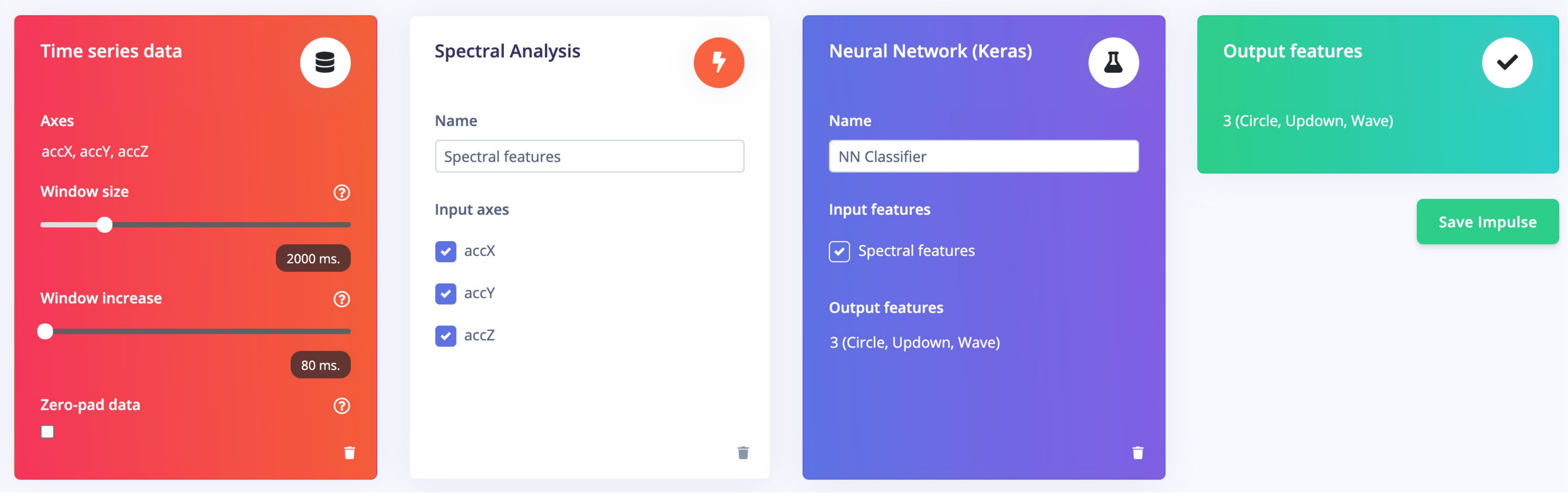
Data in training set	7m 30s
Classes	3 (Circle, Updown, Wave)
Window length	2000 ms.
Window increase	80 ms.
Training windows	4,529



Feature explorer (4,529 samples)

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

Training a Model – Impulse Design



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

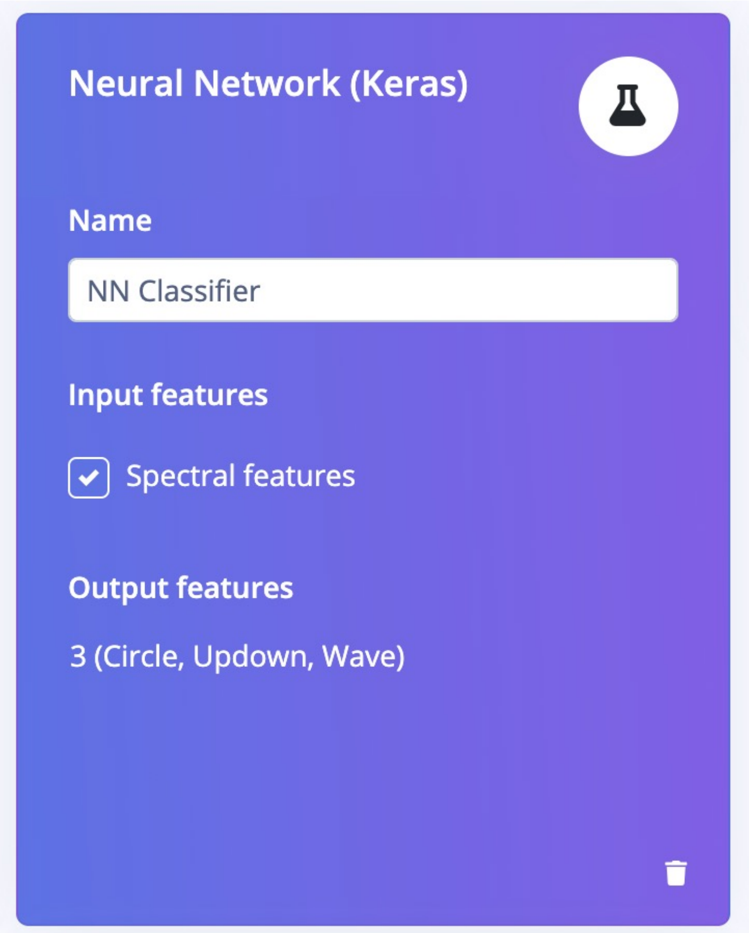
Output features

3 (Circle, Updown, Wave)

Output features

3 (Circle, Updown, Wave)

Training a Model Impulse Design



Neural Network (Keras)

Name

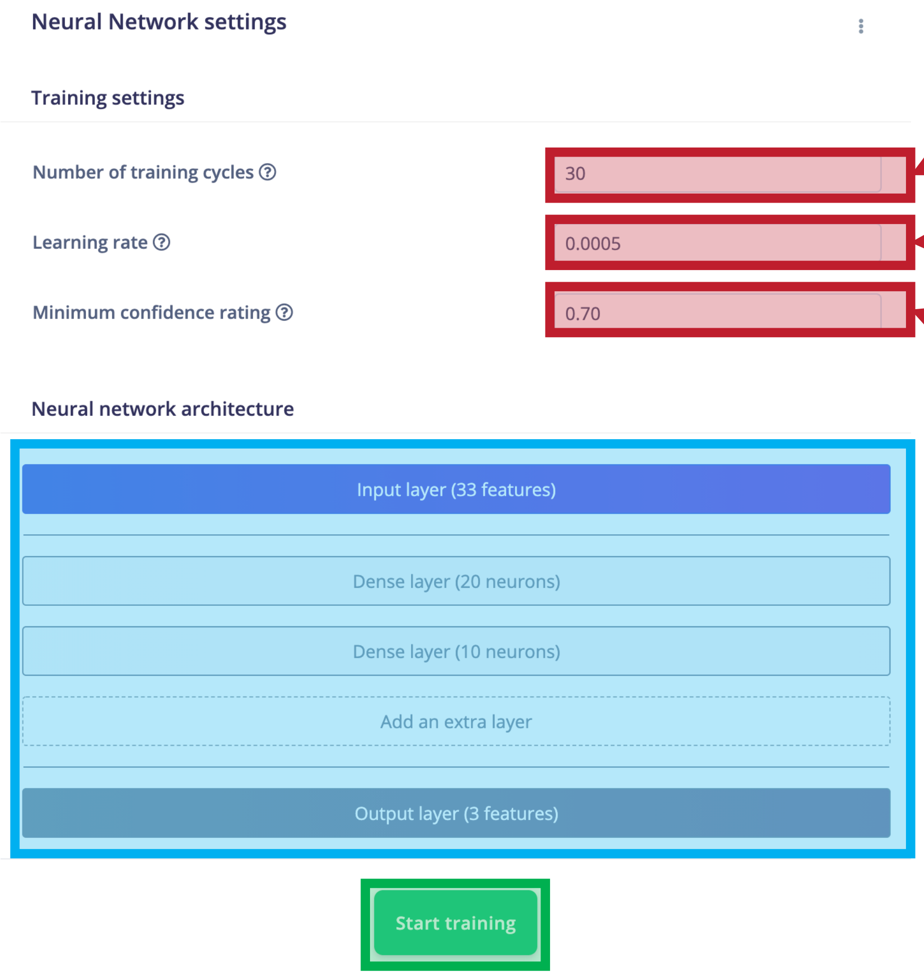
NN Classifier

Input features

☑ Spectral features

Output features

3 (Circle, Updown, Wave)



Neural Network settings

Training settings

Number of training cycles: 30

Learning rate: 0.0005

Minimum confidence rating: 0.70

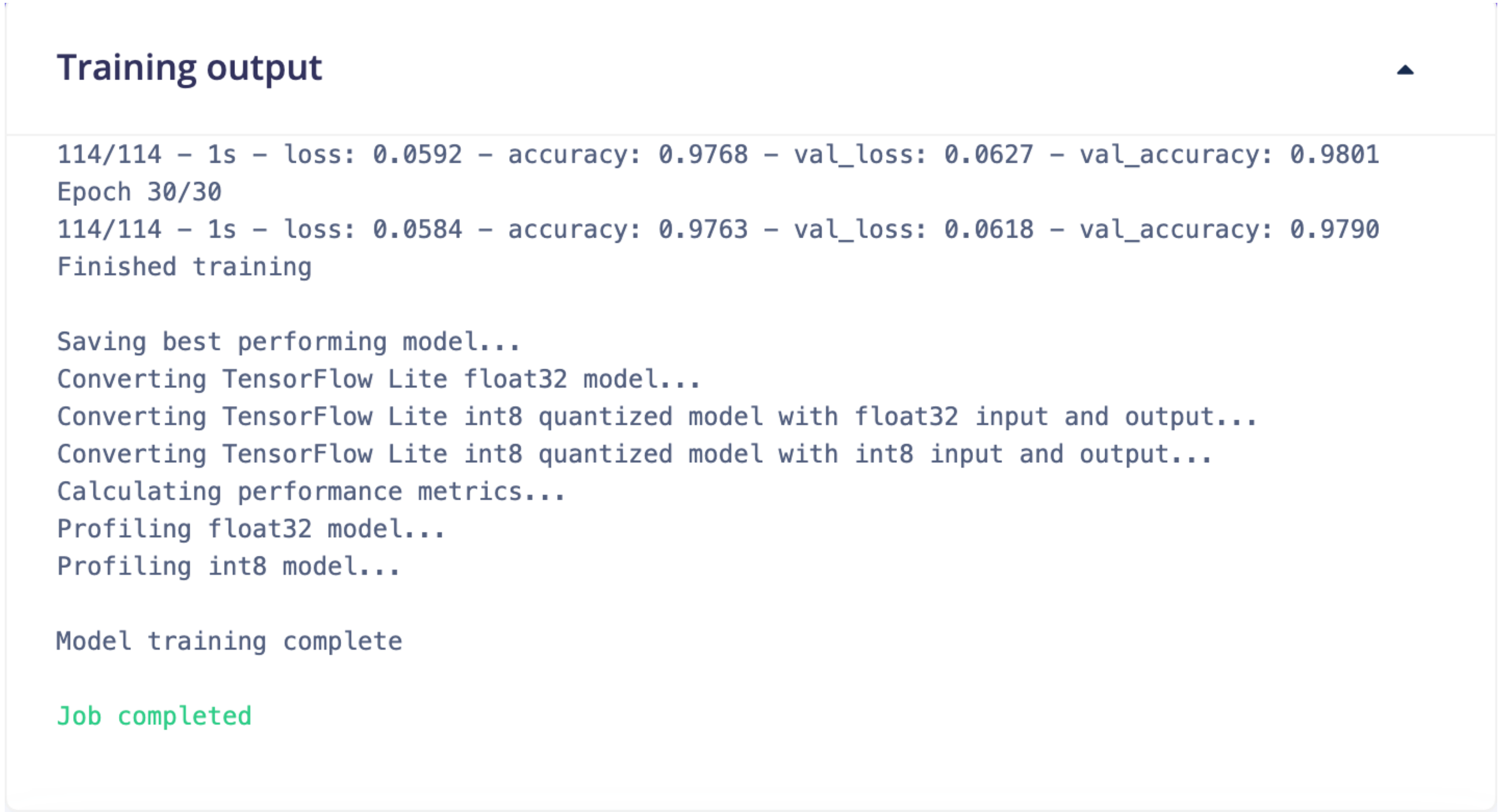
Neural network architecture

100 - 150

0.0001

0.80

Training a Model - Results



Training output

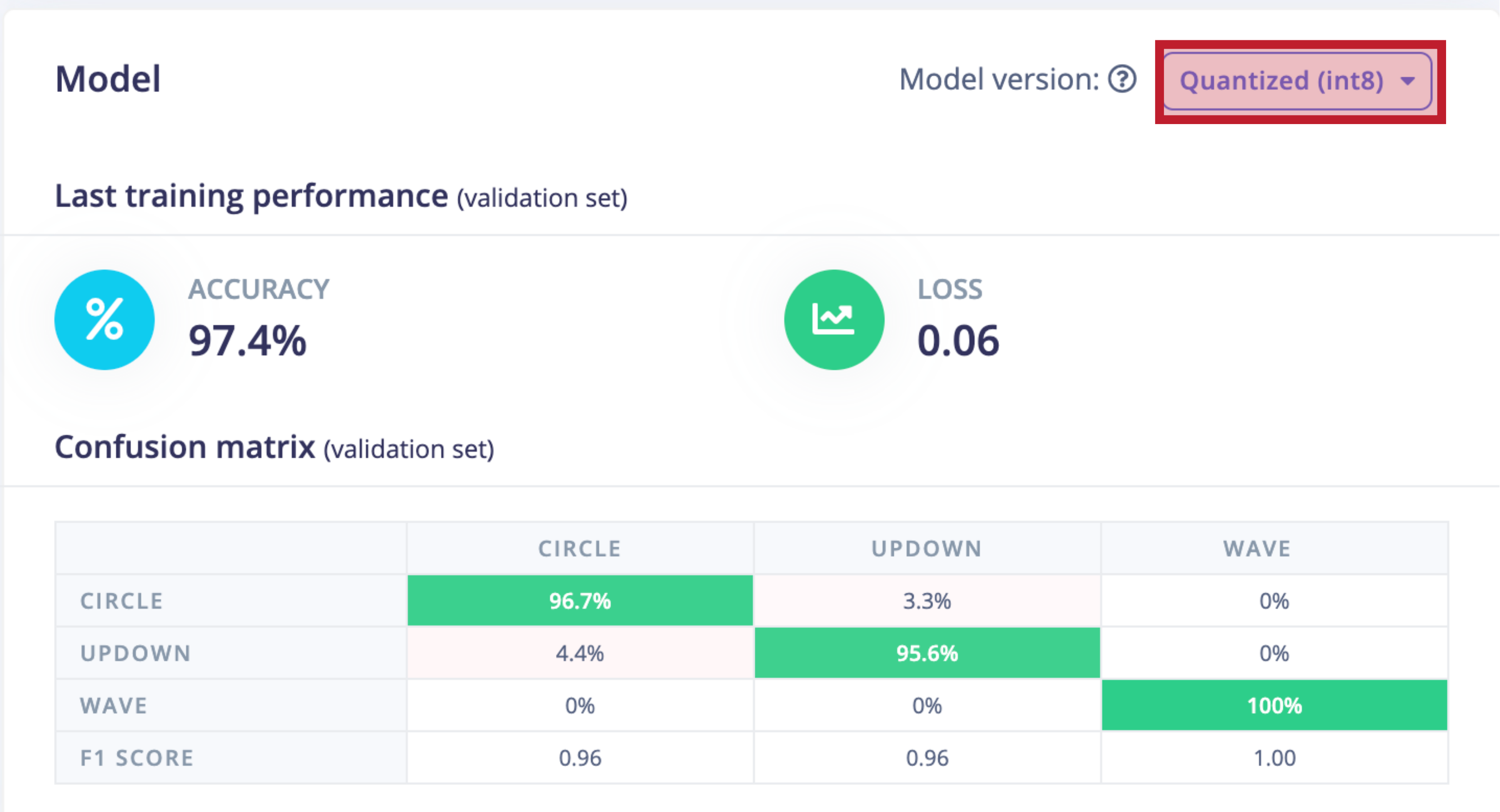
```
114/114 - 1s - loss: 0.0592 - accuracy: 0.9768 - val_loss: 0.0627 - val_accuracy: 0.9801
Epoch 30/30
114/114 - 1s - loss: 0.0584 - accuracy: 0.9763 - val_loss: 0.0618 - val_accuracy: 0.9790
Finished training

Saving best performing model...
Converting TensorFlow Lite float32 model...
Converting TensorFlow Lite int8 quantized model with float32 input and output...
Converting TensorFlow Lite int8 quantized model with int8 input and output...
Calculating performance metrics...
Profiling float32 model...
Profiling int8 model...

Model training complete

Job completed
```

Training a Model - Results



Model

Model version: Quantized (int8)

Last training performance (validation set)

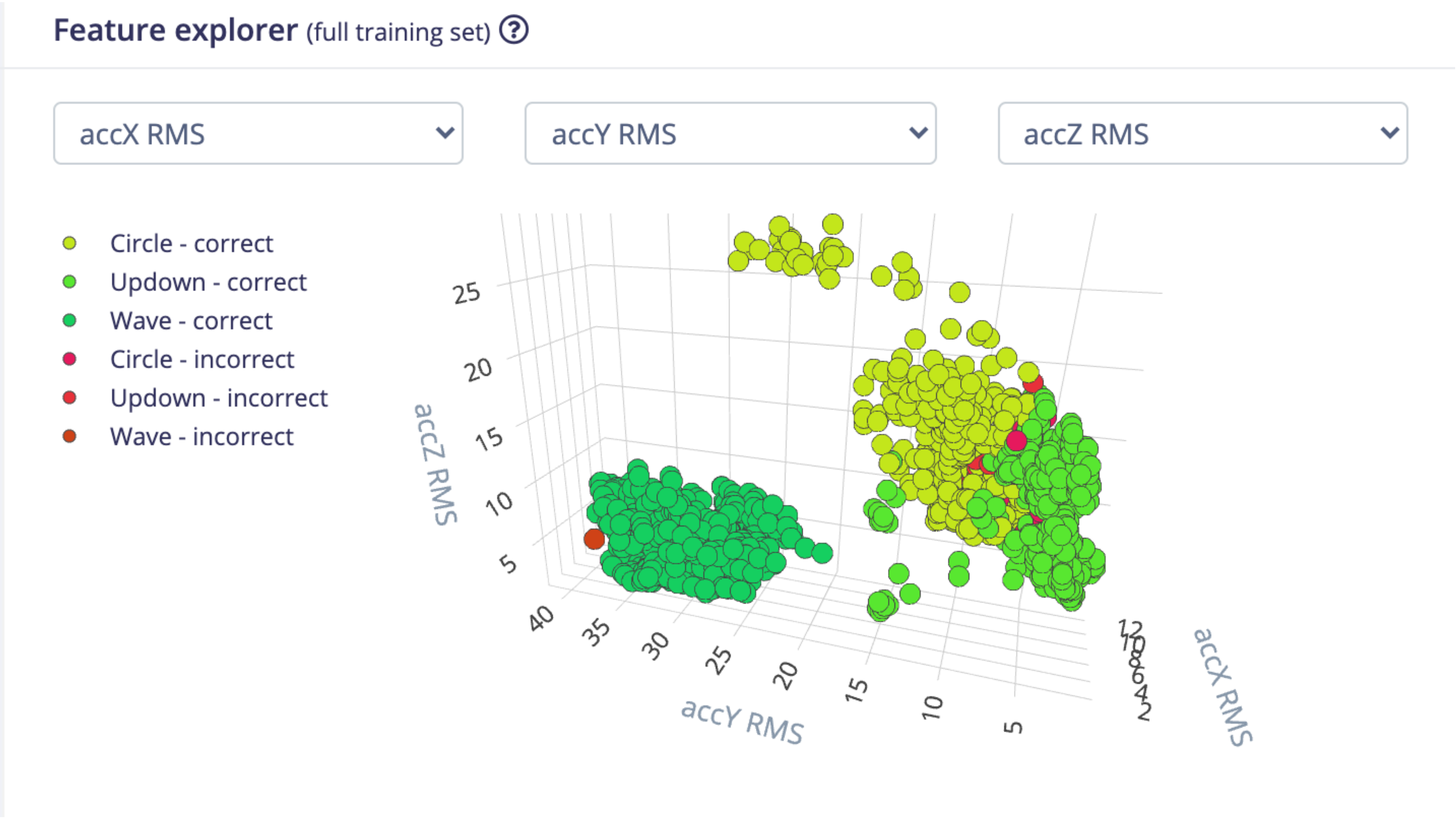
ACCURACY 97.4%

LOSS 0.06

Confusion matrix (validation set)

	CIRCLE	UPDOWN	WAVE
CIRCLE	96.7%	3.3%	0%
UPDOWN	4.4%	95.6%	0%
WAVE	0%	0%	100%
F1 SCORE	0.96	0.96	1.00

Training a Model - Results



Feature explorer (full training set) ?

accX RMS ⌄ | accY RMS ⌄ | accZ RMS ⌄

- ● Circle - correct
- ● Updown - correct
- ● Wave - correct
- ● Circle - incorrect
- ● Updown - incorrect
- ● Wave - incorrect

How do you feel about these results? Are they …

- Good
- Okay
- horrendous
- Other (put your though in the chat box please)

4 Going Further

AI and ML Resources

- Jacob's AI Blogs
- Jacob's CEC courses
- Jacob's ML Blogs

- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

BENINGO EMBEDDED GROUP

www.beningo.com

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets