

# Workout Exercise Classification

Edge AI demo by Ludwig Stumpp



MACHINE  
LEARNING  
TOKYO

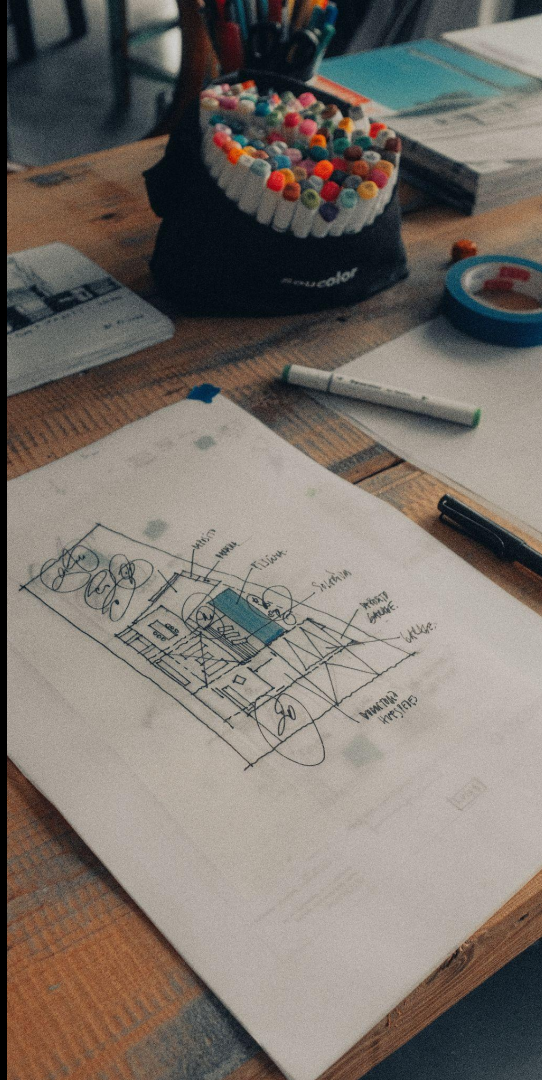
# Motivation

- reaching the smartphone app after exercises to start/stop the tracking feels unnatural
- familiarize with edge ai workflow and gather useful hands-on experience in the field
- motivate other fellow MLT members

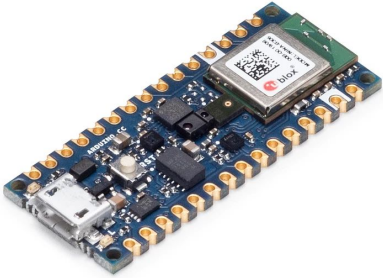


# Project Scope

- continuous classification of three exercise types:  
Jumping-Jacks, Push-Ups, Squats
- inference is running on an edge device
- total duration per exercise is accumulated and shown in  
some kind of visualization
- project time: 4 weeks besides regular job



# Hardware Requirements



Arduino Nano 33 BLE Sense

- 45mm x 18mm
- 9 axis inertial sensor
- TensorflowLite support
- BLE capabilities

32 €



Silicone sleeve

- protects the board
- improves handling and stability in pocket

5 €



USB A to Micro B cable

- power and data connection

7 €



Powerbank

- allows for mobile usage of arduino board
- alternative: external battery

~25 €



# Software Requirements



## Arduino IDE

- write code and upload it to the board
- support of external libraries

0 €



## Edge Impulse Platform

- offers end-to-end deep learning solution for edge ai devices
- free for individuals

0 €



## Chrome Web Browser

- Web browser that supports bluetooth low energy (BLE)
- for recording and visualization of data

0 €



## Python Environment

- for resampling of time series data

0 €



MACHINE  
LEARNING  
TOKYO

# Edge AI Pipeline - Data Collection & Storage

- record accelerometer data (x, y, z) while doing exercises
- board in my pocket connected to power bank
- sending data over Bluetooth Low Energy (BLE) from the board to a web app running in my browser and storing it as .csv files on my local machine
- 30s per exercise in total, including data of 4 different board orientations each
- repository of ble-recorder: <https://github.com/LudwigStumpp/ble-recorder>
- data then uploaded to Edge Impulse Platform



# Edge AI Pipeline - Data Collection & Storage



### BLE Recorder

A Web-based Tool for recording BLE Data into a .csv - Dataset.  
Note: The Web Bluetooth API, that this tool is based on, is only supported in the [here](#) listed browsers.

#### 1. Step - Specify Connection Details and Message Format

Service UUID of Peripheral:

Message Format:  Delimiter Symbol:

Establish the Connection:

#### 2. Step - Check the Connection and specify Recording Settings

Raw Data:

ts:  accX:  accY:  accZ:

Preparation Time:  s Recording Time:  s

☐ Add additional UNIX - Timestamp? (Use only if peripheral device does not send timestamp!)

Start the Recording:

#### 3. Step - Wait until Recording has finished and download .csv file

Recording starts in:  Recording Time left:  Recording Duration:

Made with ❤️ in Bavaria by [coding Stumps](#)

pushups\_1.csv

	ts	accX	accY	accZ
1	ts	accX	accY	accZ
2	116628	-3.93	-1.10	9.21
3	116648	-3.22	-1.66	10.27
4	116668	-3.19	-1.26	10.74
5	116690	-3.32	-0.01	9.75
6	116710	-3.41	-0.03	8.81
7	116734	-3.58	-1.50	9.34
8	116754	-3.21	-1.61	10.64
9	116774	-2.29	-0.44	10.48
10	116794	-2.61	0.28	9.74
11	116814	-3.44	-0.52	9.44
12	116876	-3.02	0.02	10.12
13	116896	-3.55	0.07	10.01
14	116916	-4.01	-0.30	10.53
15	116936	-3.39	-0.08	10.87
16	116956	-3.24	0.05	11.00
17	116980	-3.55	0.00	11.67
18	117000	-3.90	-0.02	11.72
19	117020	-3.84	0.10	11.62
20	117042	-3.93	0.16	10.62
21	117062	-3.59	-0.31	10.84
22	117082	-3.37	-0.44	10.96
23	117106	-3.46	-0.28	10.56
24	117126	-3.86	-0.17	10.00
25	117146	-3.71	-0.05	9.75
26	117166	-3.53	-0.21	9.62



# Edge AI Pipeline - Preprocessing & Training Pipeline

## Preprocessing:

- resampled time series data to have constant rate of 50Hz using python script and standard linear interpolation

## Training Pipeline:

- use standard spectral analysis preprocessing of Edge Impulse platform
  - low pass
  - fast fourier transform on sliding windows
  - spectral analysis to generate 33 features for each window
- 3000ms window length led to 84 training samples
- Fully Connected FNN, 33 inputs, two hidden layers, 3 outputs





# Edge AI Pipeline - Preprocessing & Training Pipeline

CREATE IMPULSE (WORKOUT-CLASSIFICATION)

Ludwig

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

3000 ms.

Window increase

300 ms.

Frequency (Hz)

50

Zero-pad data

Spectral Analysis

Name

Spectral features

Input axes

☒ accX

☒ accY

☒ accZ

Classification (Keras)

Name

NN Classifier

Input features

☒ Spectral features

Output features

3 (jumpingjacks, pushups, squats)

Output features

3 (jumpingjacks, pushups, squats)

Save Impulse

Add a processing block

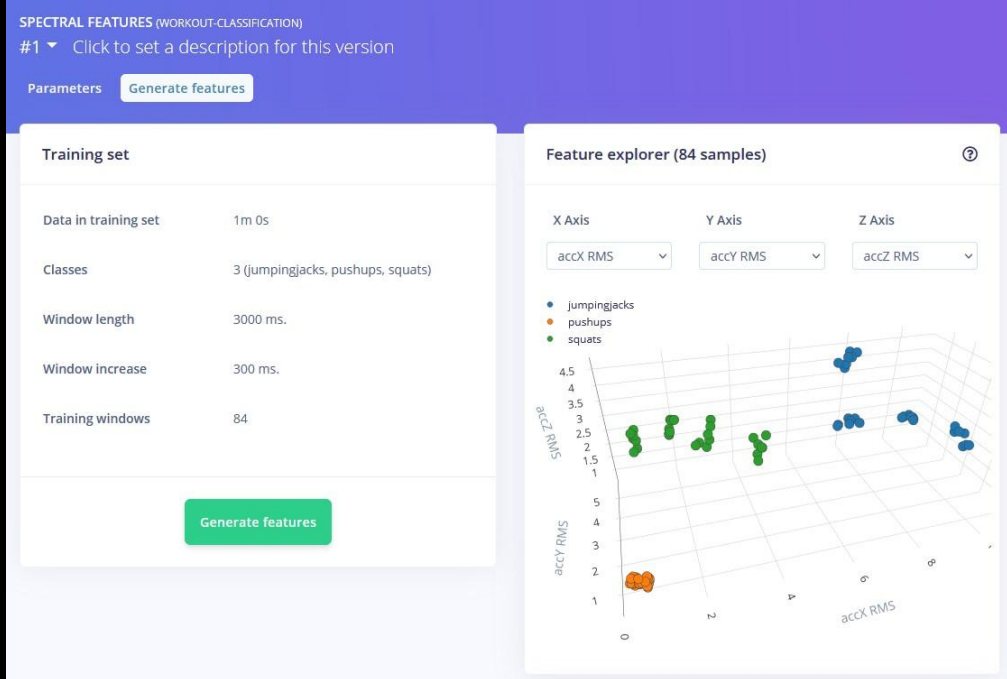
Add a learning block

The logo for Machine Learning Tokyo, featuring a stylized 'M' and 'T' inside a circle.

MACHINE  
LEARNING  
TOKYO

Workout Classification Project by Ludwig Stumpp

# Edge AI Pipeline - Preprocessing & Training Pipeline



**NN CLASSIFIER (WORKOUT-CLASSIFICATION)**  
#1 ▾ Click to set a description for this version

**Neural Network settings** ⓘ

**Training settings**

Number of training cycles ⓘ

Learning rate ⓘ

**Neural network architecture**

Input layer (33 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (3 features)

**Start training**



# Edge AI Pipeline - Model Deployment & Inference

## Model Deployment:

- compiled pipeline into arduino library using the Edge Impulse Deployment features
- used optimized version (int8) to increase on-device performance

## Inference:

- extended sample code provided by Edge Impulse with BLE capabilities
- Inference in continuous fashion on the board and results sent via BLE to web app
- web app accumulates exercise durations

# Edge AI Pipeline - Model Deployment & Inference

## 2. Step - Check the Connection

Raw Data:

uncertain

Start the Workout:

Start

Stop

Reset

Delta per Classification

0,4

s

No Exercise Identifier

uncertain

## 3. Step - Perform your Workout

jumpingjacks:

34

s

squats:

62.4

s

pushups:

22.4

s

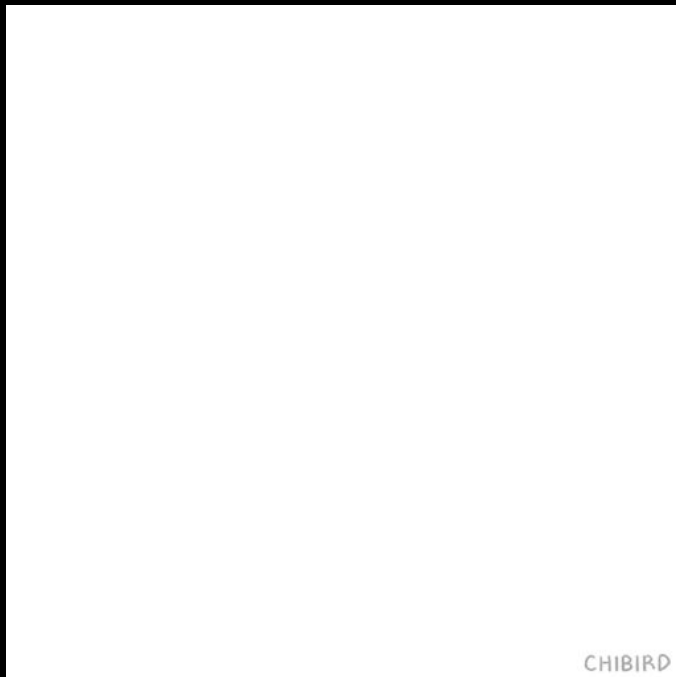


# Challenges & Solutions

- recording of data by default requires connection to local machine via usb-cable:
  - developed tool to record data via Bluetooth Low Energy (BLE)
  - one then only requires connection to power bank
- features heavily rely on orientation of the board:
  - collect more data in different board orientations to make prediction more robust
  - alternative: apply PCA on raw data - untested since not supported by Edge Impulse
- board is slightly moving while performing the exercises:
  - still an issue
  - happening due to high weight of power bank and missing attachment to sport pants
  - ideas: use light weight battery instead of power bank, improve packaging



# Call to Action



- getting started is very easy using the Edge Impulse platform
- no need of any prior knowledge regarding electrical engineering
- if I could do it, then you can too!



# Thank you!



<https://github.com/LudwigStumpp/ble-recorder>

<https://github.com/LudwigStumpp/arduino-workout-classification>

[https://twitter.com/ludwig\\_stumpp](https://twitter.com/ludwig_stumpp)



MACHINE  
LEARNING  
TOKYO