Cycling Pedal Analysis

DGMD S-14 Summer 2021

Elaine Kristant

Outline

- Problem
- Background
- Data Collection
- Data segmentation and cleaning
- Analysis

Problem

Cycling computers that attach to bikes typically give you data related to trip time, speeds, and sometimes cadence, but they typically don't provide any data on the mechanics of the rider unless you're willing to spend over $1000. This project aims to use a set of sensors to analyze the pedal and foot movement to increase efficiency of the ride.

Current state of the art - standard bike computer

Lower End

- Inexpensive: $20 - $300
- Self contained to one device but sometimes can stream data to smartphone
- Typical data collected:
  - Current speed
  - Maximum speed
  - Average speed
  - Elapsed time
  - Trip distance
  - Odometer
  - GPS*
  - Heart rate*

*Typically only found on the more expensive models

Higher End

- Expensive: $300 - $1500
- Built into pedal or crank
- Communicates to smartphone or lower end bike computer
- Typical data collected:
  - Everything that a lower end computer can do
  - Forces applied throughout the stroke
  - Power
  - Position analysis

Prices and features found from https://www.nashbar.com/bike-computers-gps-powermeters

Lower End

https://www.nashbar.com/cateye-urban-bike-computer-black-wireless-1604380/p-reqcqqqa2taeyaa2

https://www.nashbar.com/garmin-edge-520-plus-cycling-computer-black-010-02083-00/p-ruqxtarqxuqeyaa2

Higher End

https://www.sram.com/en/sram/models/pm-red-d1

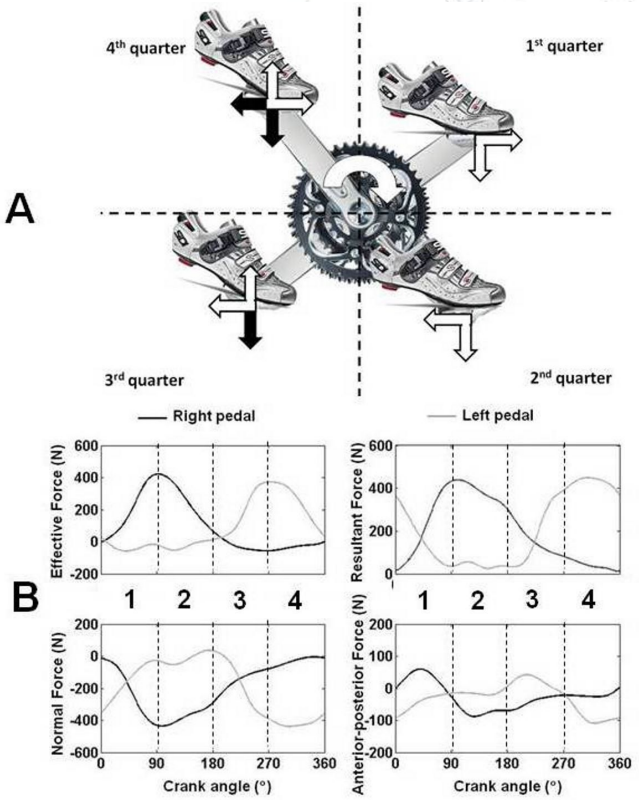
https://static.garmincdn.com/en/products/010-01787-02/g/cf-lg-bf0b13cc-487b-41e1-ba67-3fad8423f174.jpg

Stroke Analysis

- Even power throughout cycle
- Foot/pedal angle matters

Theory:

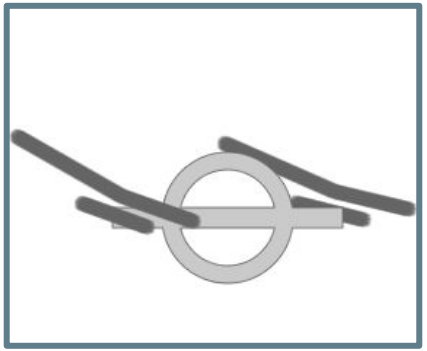
- This is a two-dimensional problem
- Z Acceleration will be used to determine the stroke
- The rotation around the X axis (X for the gyro) will determine the foot position
- Y axis acceleration might also detect foot position



Bini, Rodrigo & Hume, Patria & Croft, James & Kilding, Andrew. (2013). Pedal force effectiveness in cycling: A review of constraints and training effects. Journal of Science and Cycling. 2. 11-24.

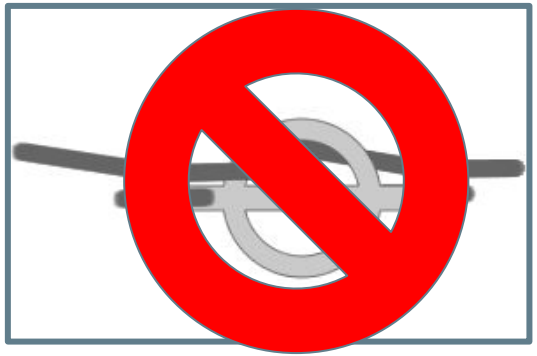
Classes

Ideal

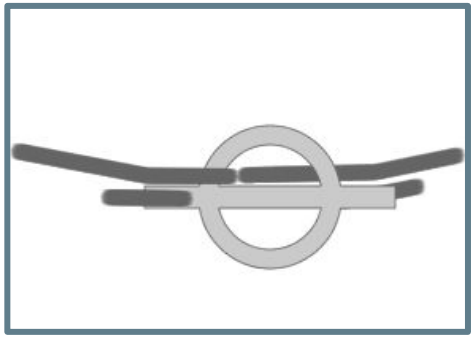


Toes down

Up hill



No ankle



Flat

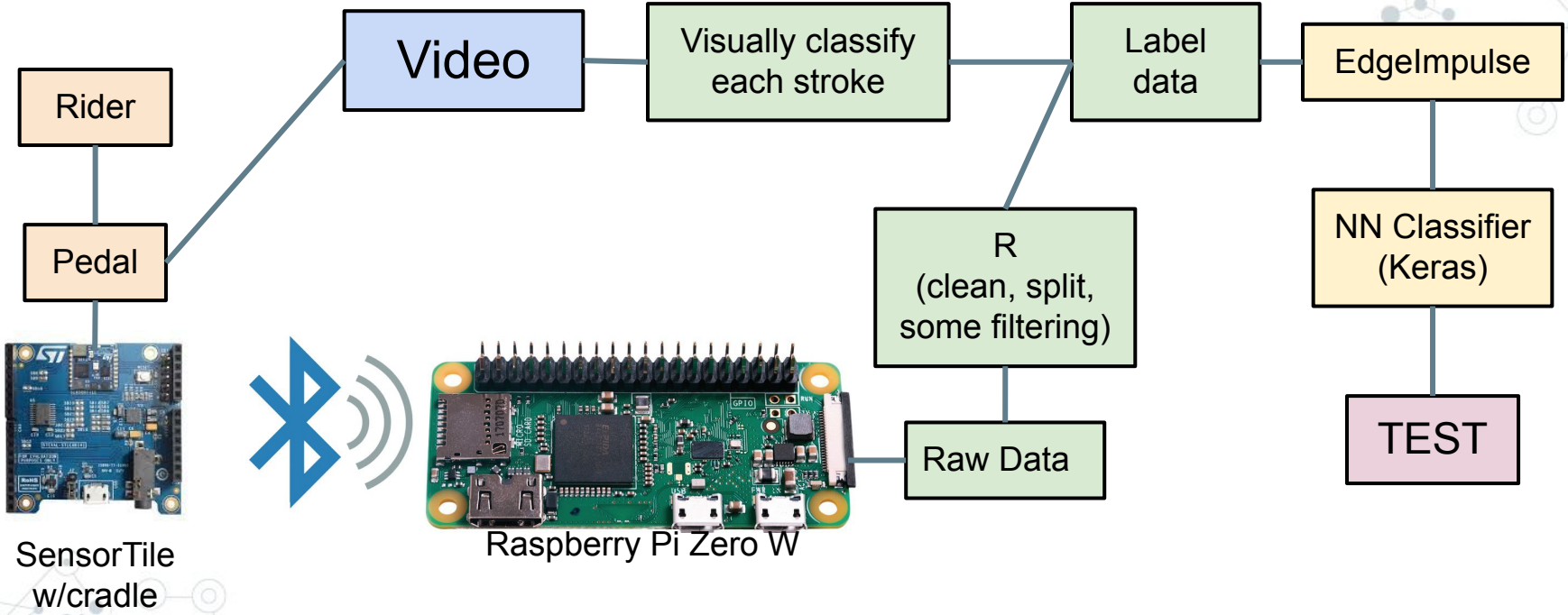
http://www.perfectcondition.ltd.uk/Articles/Pedalling/index.htm

Goal and approach

The goal of this project aims to determine if proper pedaling technique can be done through an IMU instead of expensive power analysis pedals.

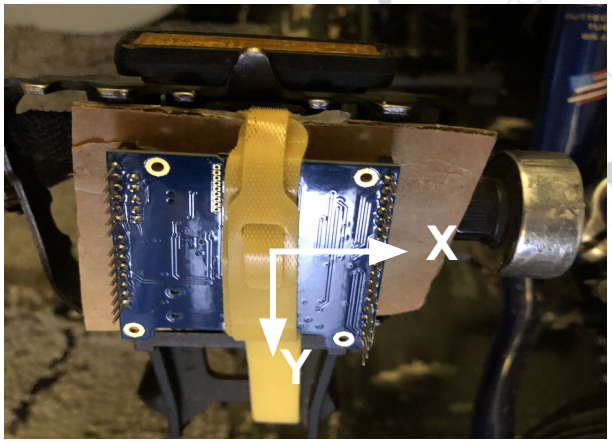
By looking at the acceleration and angle of the pedal, a profile can be created and analyzed against optimal performance and feedback can be provided.

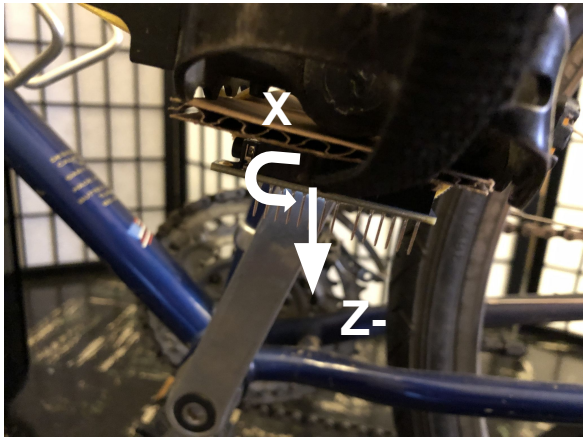
Block diagram



Setup

- Commuter bike on stationary trainer
- Low force on trainer
- Pedals with toe-clips
- Cycling shoes
- SensorTile with cradle expansion board attached to bottom of pedal
  - Acceleration Z axis vertical
  - Gyroscope X axis in line with pedal axis
  - Wall power
- RaspberryPi located within bluetooth range





Data collection

SensorTile running slightly modified “ALLMEMS1_V3.1.0”

↓

Data transmitted to RasPi through BLE

↓

Data then transferred to PC for analysis

```
Characteristic value was written successfully
Notification handle = 0x0011 value: 1d 6f 30 00 43 fc db fe fe ff e1 ff 01 00 cc 00 98 00 25 02
Notification handle = 0x0011 value: 23 6f 30 00 43 fc dd fe fe ff e0 ff 01 00 d2 00 98 00 28 02
Notification handle = 0x0011 value: 29 6f 30 00 45 fc de fe ff ff e0 ff 01 00 d5 00 98 00 25 02
Notification handle = 0x0011 value: 2f 6f 2f 00 44 fc dd fe fe ff e1 ff 01 00 d2 00 a0 00 1f 02
Notification handle = 0x0011 value: 36 6f 31 00 43 fc dd fe fe ff e1 ff 01 00 cc 00 95 00 28 02
Notification handle = 0x0011 value: 3c 6f 2f 00 43 fc dd fe fe ff e0 ff 01 00 cf 00 95 00 24 02
Notification handle = 0x0011 value: 42 6f 30 00 44 fc de fe ff ff e2 ff 01 00 cd 00 9b 00 1e 02
Notification handle = 0x0011 value: 48 6f 31 00 44 fc dd fe fe ff e1 ff 01 00 d0 00 9a 00 24 02
Notification handle = 0x0011 value: 4f 6f 30 00 44 fc dd fe fe ff e0 ff 01 00 cf 00 94 00 22 02
Notification handle = 0x0011 value: 55 6f 31 00 44 fc de fe fe ff e2 ff 01 00 cf 00 97 00 21 02
Notification handle = 0x0011 value: 5b 6f 31 00 44 fc de fe fe ff e1 ff 01 00 ca 00 9d 00 1f 02
Notification handle = 0x0011 value: 61 6f 31 00 44 fc de fe fe ff e0 ff 01 00 d2 00 9b 00 25 02
Notification handle = 0x0011 value: 68 6f 31 00 43 fc dd fe fe ff e2 ff 01 00 cd 00 9b 00 2a 02
Notification handle = 0x0011 value: 6e 6f 31 00 43 fc dc fe fe ff e1 ff 01 00 d0 00 94 00 24 02
Notification handle = 0x0011 value: 74 6f 2f 00 43 fc dc fe fe ff e0 ff 01 00 cf 00 9d 00 2b 02
Notification handle = 0x0011 value: 7a 6f 30 00 44 fc dc fe fe ff e1 ff 01 00 d5 00 97 00 21 02
Notification handle = 0x0011 value: 81 6f 31 00 43 fc dc fe fe ff e1 ff 01 00 d2 00 94 00 25 02
Notification handle = 0x0011 value: 87 6f 30 00 43 fc db fe ff ff e0 ff 01 00 d5 00 9b 00 1e 02
Notification handle = 0x0011 value: 8d 6f 30 00 44 fc dc fe fe ff e2 ff 01 00 cf 00 97 00 24 02
Notification handle = 0x0011 value: 93 6f 31 00 43 fc dc fe fe ff e1 ff 01 00 d5 00 9d 00 28 02
Notification handle = 0x0011 value: 9a 6f 31 00 45 fc db fe fe ff e0 ff 01 00 d8 00 98 00 28 02
Notification handle = 0x0011 value: a0 6f 2f 00 43 fc da fe fe ff e2 ff 01 00 cd 00 94 00 25 02
Notification handle = 0x0011 value: a6 6f 30 00 43 fc da fe fe ff e1 ff 01 00 db 00 9d 00 24 02
Notification handle = 0x0011 value: ac 6f 30 00 43 fc db fe fe ff e0 ff 01 00 db 00 9a 00 1e 02
Notification handle = 0x0011 value: b3 6f 30 00 44 fc db fe fe ff e1 ff 01 00 d0 00 9b 00 24 02
Notification handle = 0x0011 value: b9 6f 30 00 43 fc dc fe fe ff e1 ff 01 00 d3 00 a0 00 28 02
Notification handle = 0x0011 value: bf 6f 31 00 43 fc dc fe fe ff e0 ff 01 00 d0 00 91 00 1e 02
Notification handle = 0x0011 value: c5 6f 31 00 43 fc dd fe fe ff e2 ff 02 00 d6 00 92 00 21 02
```

Data interpretation, cleaning, and segmentation

Sensor data converted from hex to dec

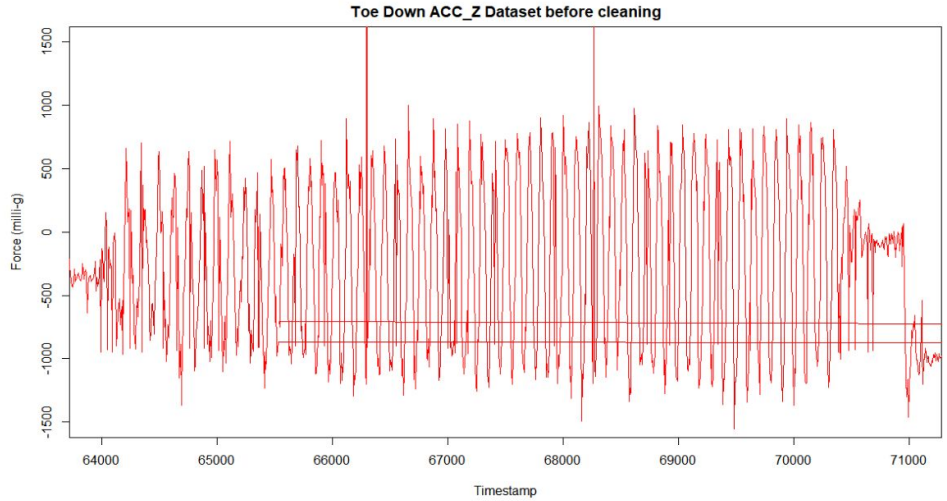
- Segment out by sensor
- Adjust for Hex 2's
- Zero ACC Z (get rid of gravity)

Data Smoothed

- Moving Average
- Clean data

Data Segmented

- Split individual strokes
- Standardized timestamp
- Export CSV

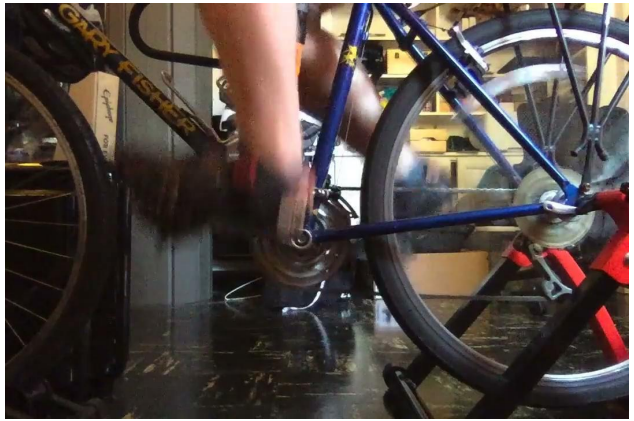


Classification

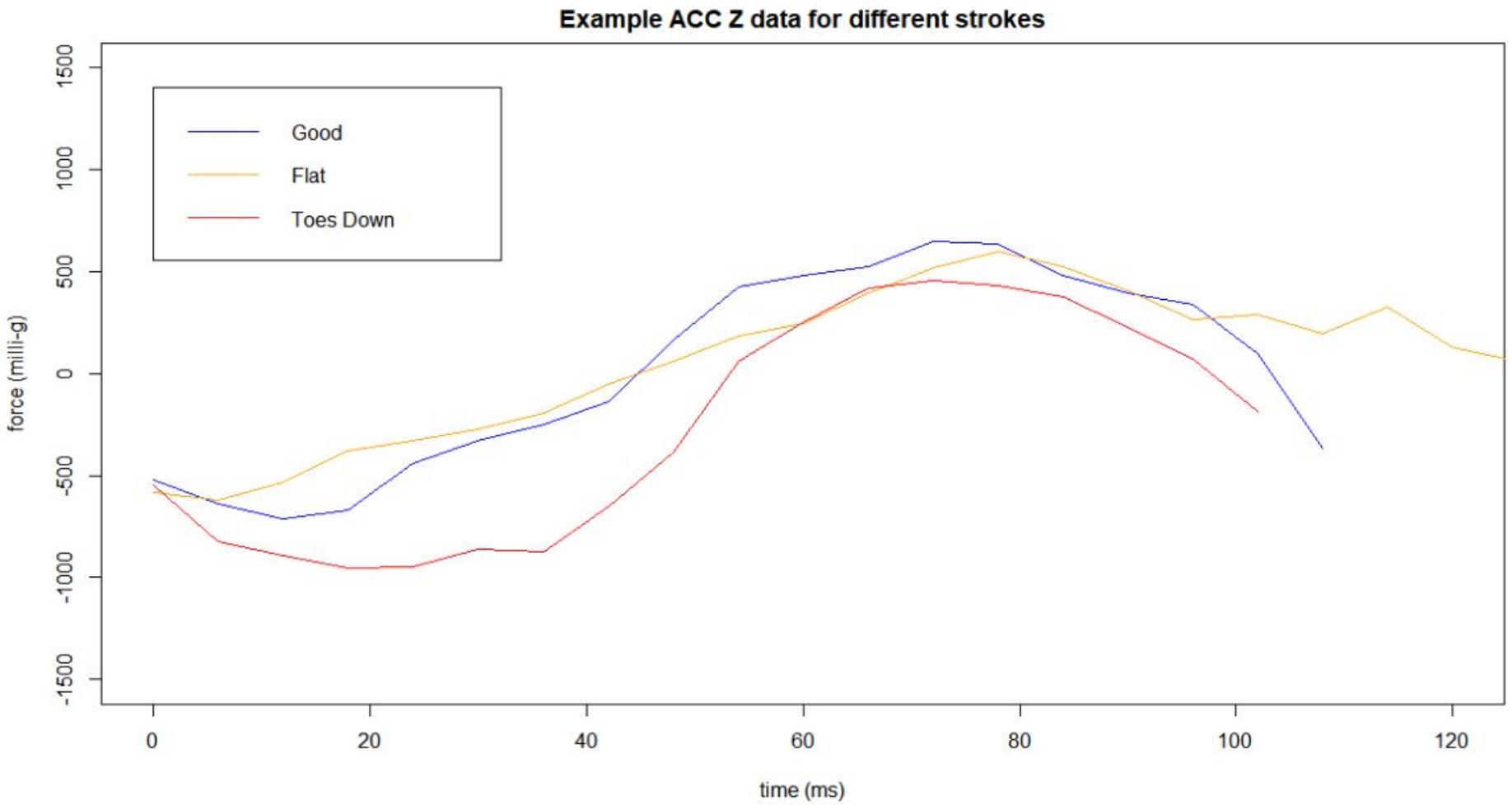
- Used video to classify each stroke
- Looked at pedal angle
- Looked at heel

Good

Flat

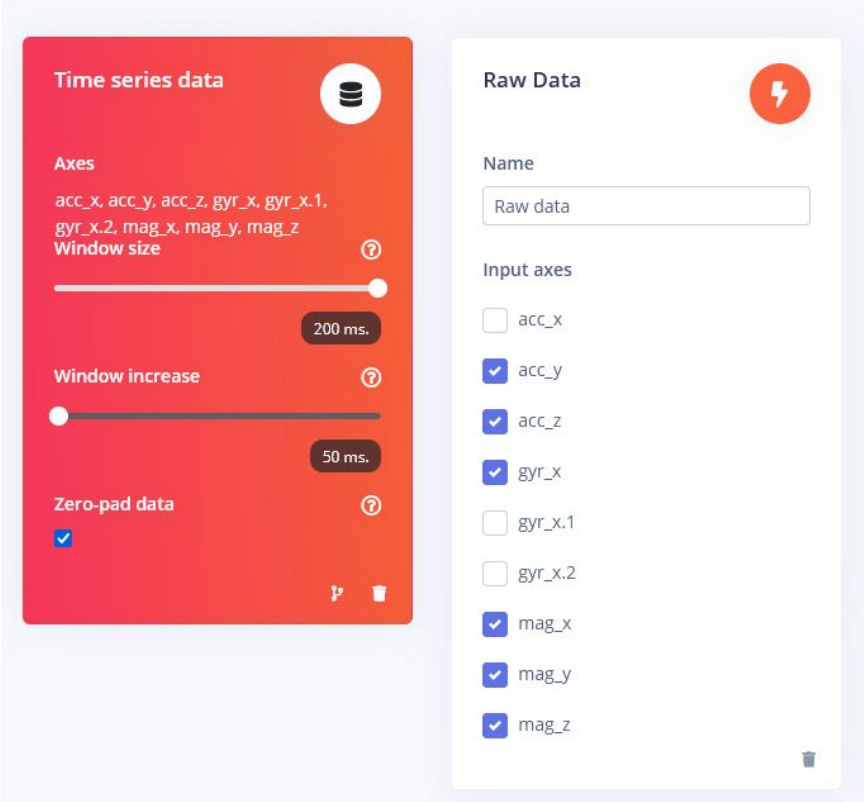


Toes Down


Example ACC Z data for different strokes
Good
Flat
Toes Down
force (milli-g)
1500
1000
500
0
-500
-1000
-1500
0
20
40
60
80
100
120
time (ms)

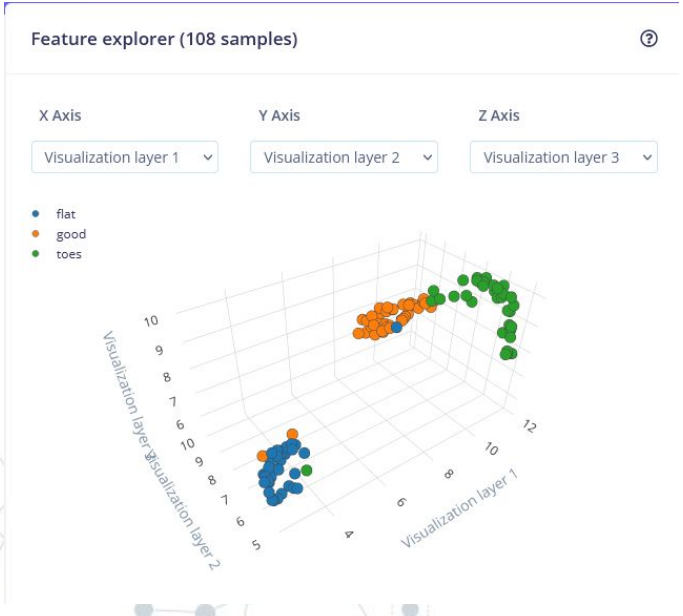
Data analysis

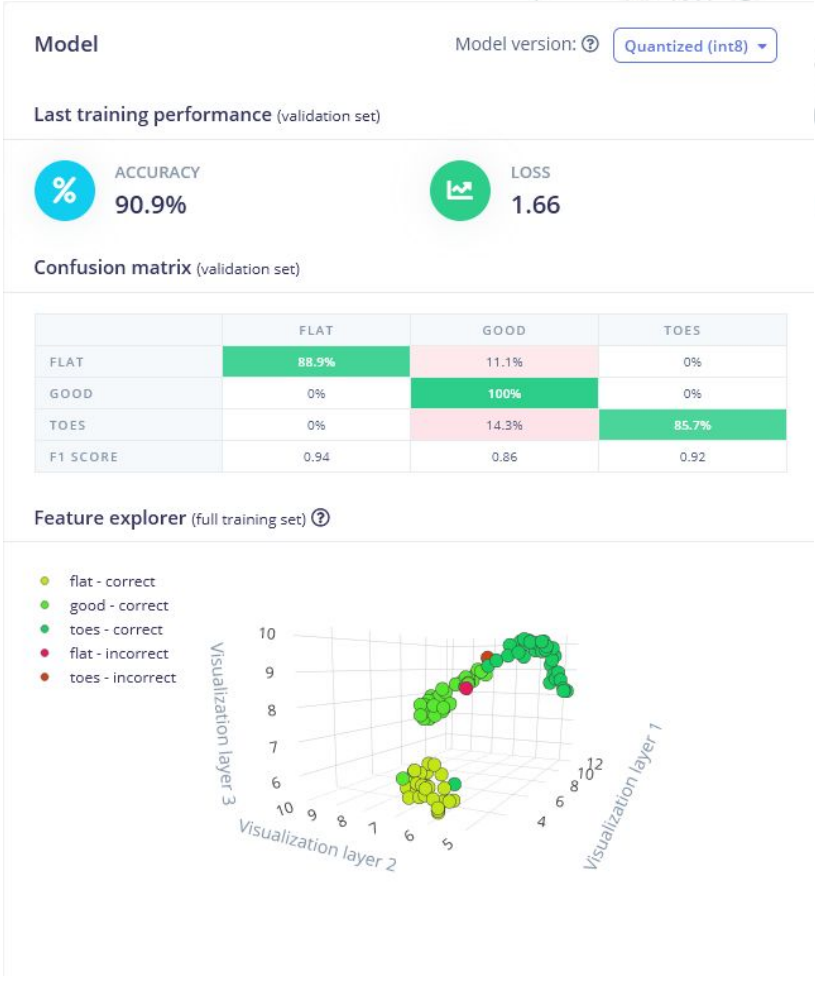
- EdgeImpulse
- Training Data:
  - Good: 40
  - Flat: 33
  - Toes down: 35
- Testing Data:
  - Good: 10
  - Flat: 11
  - Toes down: 10
- Data:
  - acc_y
  - acc_z
  - gyr_x
  - mag_x
  - mag_y
  - mag_z



Learning

- NN Classifier (keras)
- Training cycles: 30
- Learning rate: 0.01
- Minimum confidence: 0.6





Testing

- One data point mis-classified

Model testing results

ACCURACY
96.77%

	FLAT	GOOD	TOES	UNCERTAIN
FLAT	90.9%	9.1%	0%	0%
GOOD	0%	100%	0%	0%
TOES	0%	0%	100%	0%
F1 SCORE	0.95	0.95	1.00	

- Cloned project, rebalanced, and retrained

Model testing results

ACCURACY
91.67%

	FLAT	GOOD	TOES	UNCERTAIN
FLAT	91.7%	8.3%	0%	0%
GOOD	0%	100%	0%	0%
TOES	0%	14.3%	85.7%	0%
F1 SCORE	0.96	0.87	0.92	

Final thoughts

- More data with more riders
- Better pedals
- Try to increase # of classes
- Live classification