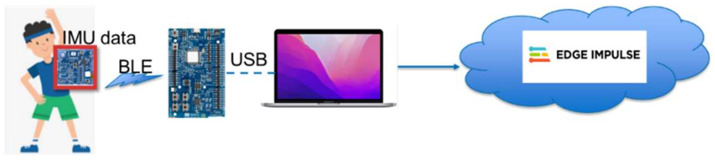
## I. Introduction

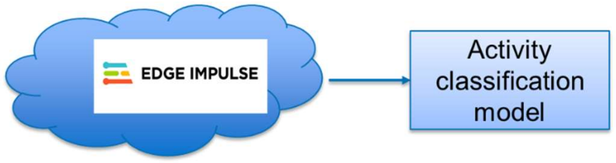
This document contains instructions for you to build and run a machine learning model on an edge device. In particular, you will develop a motion classification model, and run this on a Thingy52, a multi-sensor prototyping platform from Nordic Semiconductor. The classification model is built by using an online platform named EdgeImpulse, and is then integrated into prebuilt Thingy firmware. Your main task is building and optimizing the model on EdgeImpulse. The firmware integration requires only a few configurations, for which no programming skill is required.

There are 7 steps to build the classification model from scratch:

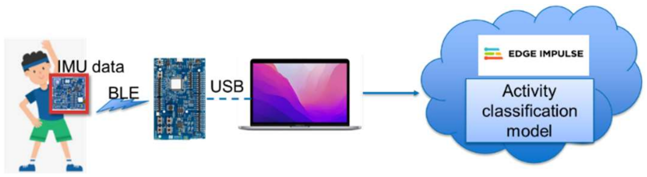
1. Preparation:
    - Hardware: Thingy52, nrf52DK, an Android phone
    - Install software
2. **2.** Create an EdgeImpulse account and project
3. Collect data:



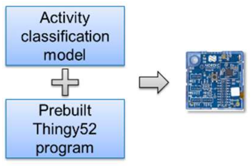
4. Build, Train, Test and Tune a model:



5. Live test and export the best model



6. Download model to Thingy52



7. Demonstrate the real-world results:

## II Preparation:

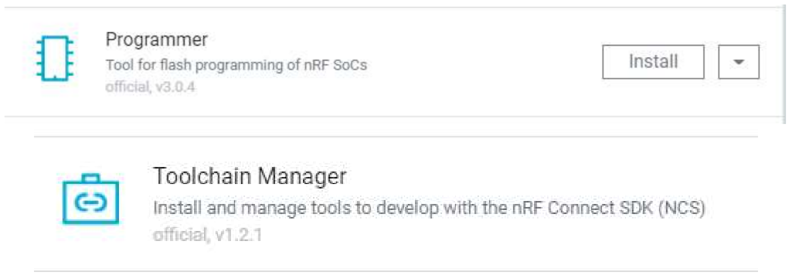
1. Hardware:
   - Thingy52
   - Nrf52DK
   - JTAG cable 1.27mm

2. Software:
   a. **nRF Connect for Desktop** : This is a cross-platform tool from Nordic Semiconductor. In the course, we will use two of its tools: “Programmer” to flash compiled program onto the Thingy52 and the nRF52DK, “Toolchain manager” to manage the developing tool for the Thingy52.

- Download and install “nRF Connect for desktop” from:

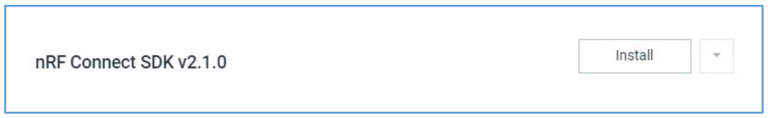
https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-desktop

  **Note:** Make sure that during installation you also install the “SEGGER J-Link” software (included in the “nRF Connect for desktop” software as a separate installation process), as this is required to properly detect and connect to the nRF52DK in a later step.
- Open nRF Connect for desktop, install “Programmer” and “Toolchain manager”


Programmer
Tool for flash programming of nRF SoCs
official, v3.0.4

Toolchain Manager
Install and manage tools to develop with the nRF Connect SDK (NCS)
official, v1.2.1

- After installing the Toolchain Manager, click “Open” on the Toolchain Manager
- Install nRF Connect SDK **v2.1.0**.

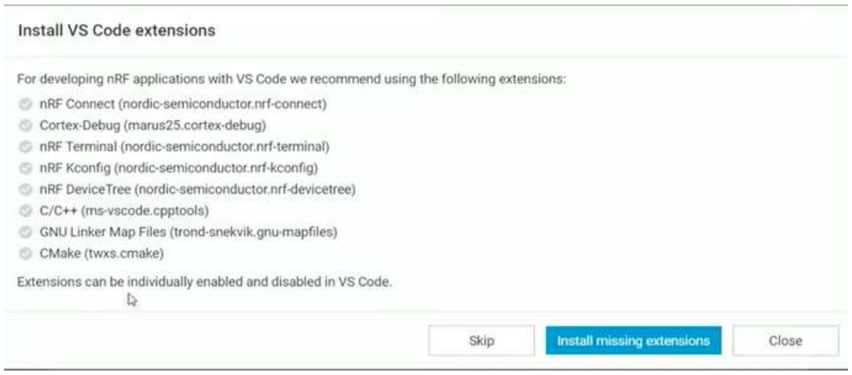

nRF Connect SDK v2.1.0

- Close all tools and nRF Connect for desktop after installation has completed.

b. **VSCode**: This is a source/code editor which we use to integrate the machine learning model code into a prebuilt program.
Download VSCode from here:
[https://code.visualstudio.com/](https://code.visualstudio.com/)

c. **Update extensions for VS code:**

- Open nRF Connect for desktop → Toolchain manager → Open VS Code (in nRF Connect SDK v2.1.0)
- A popup window will appear as shown below, click on "Install missing extensions"



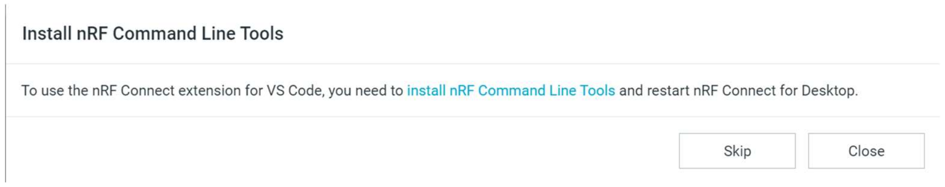
Install VS Code extensions

For developing nRF applications with VS Code we recommend using the following extensions:

- nRF Connect (nordic-semiconductor.nrf-connect)
- Cortex-Debug (marus25.cortex-debug)
- nRF Terminal (nordic-semiconductor.nrf-terminal)
- nRF Kconfig (nordic-semiconductor.nrf-kconfig)
- nRF DeviceTree (nordic-semiconductor.nrf-devicetree)
- C/C++ (ms-vscode.cpptools)
- GNU Linker Map Files (trond-snekvik.gnu-mapfiles)
- CMake (twxs.cmake)

Extensions can be individually enabled and disabled in VS Code.

- Another popup might appear (see below screenshot) stating that you should install "nRF Command Line Tools" to be able to use the "nRF Connect extension for VS Code". You can download and install this additional tool from here (Windows, Linux, MacOS):
[https://www.nordicsemi.com/Products/Development-tools/nRF-Command-Line-Tools/Download#infotabs](https://www.nordicsemi.com/Products/Development-tools/nRF-Command-Line-Tools/Download#infotabs)



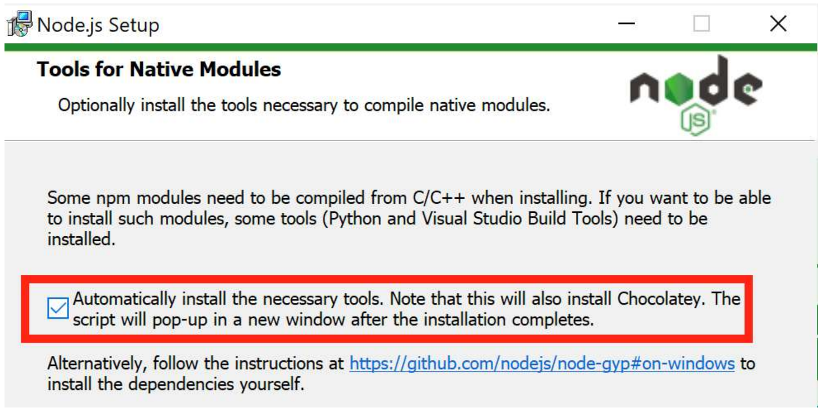
Install nRF Command Line Tools

To use the nRF Connect extension for VS Code, you need to install nRF Command Line Tools and restart nRF Connect for Desktop.

d. **Edge Impulse Data Forwarder**: Download and install the Edge Impulse CLI from the below link, this command line interface is used to stream data to the Edge Impulse platform. To install it, follow the instructions listed in below link:
[https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation](https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation)

Please note:

- you need Node JS v14 or higher to be able to install the Edge Impulse CLI. Note that you can use nvm (Windows / Linux / MacOS)to install multiple Node JS versions on your system and use it to easily switch between Node JS versions.
- During node JS installation, you must select the option as in this figure, which will install Chocolatey, Python (if not yet installed), VS Studio 2019 packages.


Node.js Setup

**Tools for Native Modules**

Optionally install the tools necessary to compile native modules.

Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python and Visual Studio Build Tools) need to be installed.

☑ Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.

Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.

- If installing the tools following the Node JS is error, it can be come from the mismatch between the installed Python version and Chocolatey. Please **uninstall** Node Js, all Python versions, Chocolatey, VS Studio 2019 packages and restart install only NodeJS (all uninstalled packages will be automatically installed in correct versions).

## III. Create an account and project on Edge Impulse

### 1. Account registration:

- Edge Impulse is a free online development platform for machine learning on edge devices. You can build, test and adjust a machine learning model using this platform. The platform will export the final model into firmware code or a C++ library to be deployed on your edge devices.

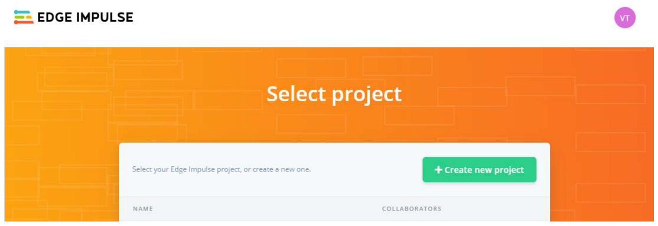
- This section instructs you on how to create a new machine learning project on Edge Impulse for motion classification.

### 2. Account registration:

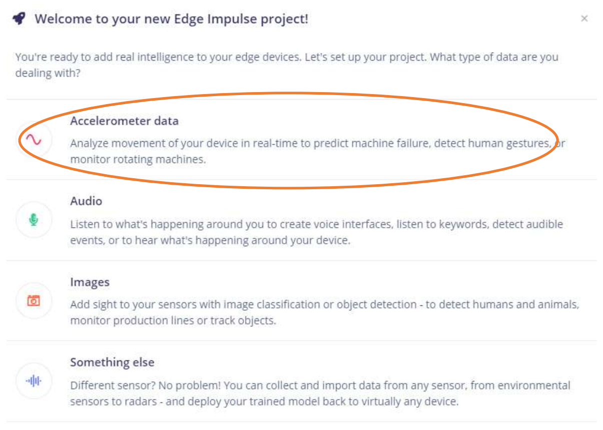
To start using Edge Impulse, first sign up for an account using this link: https://www.edgeimpulse.com/

### 3. Create an Edge Impulse project:

- Log in
- Click on "Create new Project" and name the project in the pop up window, e.g: CPS_Test1



- In the next pop up window, select “Accelerometer data”, which sets the project up to be used for motion classification.



Welcome to your new Edge Impulse project!

You're ready to add real intelligence to your edge devices. Let's set up your project. What type of data are you dealing with?

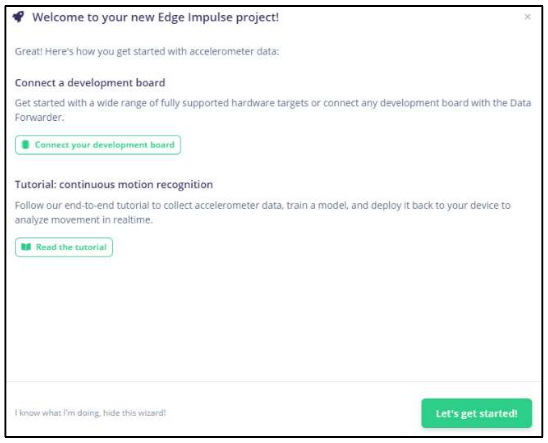
**Accelerometer data**
Analyze movement of your device in real-time to predict machine failure, detect human gestures, or monitor rotating machines.

**Audio**
Listen to what's happening around you to create voice interfaces, listen to keywords, detect audible events, or to hear what's happening around your device.

**Images**
Add sight to your sensors with image classification or object detection - to detect humans and animals, monitor production lines or track objects.

**Something else**
Different sensor? No problem! You can collect and import data from any sensor, from environmental sensors to radars - and deploy your trained model back to virtually any device.

- Finally, press “Let’s get started” in the next window to finish creating this project.

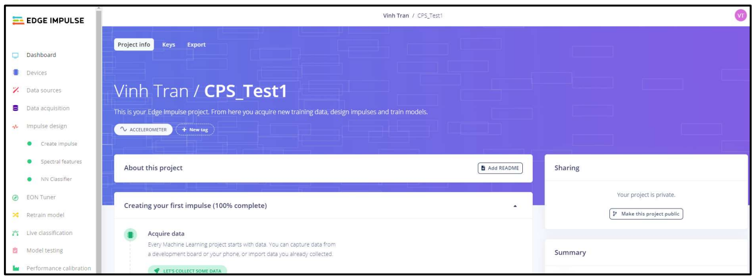


Welcome to your new Edge Impulse project!

Great! Here's how you get started with accelerometer data:

**Connect a development board**
Get started with a wide range of fully supported hardware targets or connect any development board with the Data Forwarder.

**Tutorial: continuous motion recognition**
Follow our end-to-end tutorial to collect accelerometer data, train a model, and deploy it back to your device to analyze movement in realtime.



Vinh Tran / **CPS_Test1**

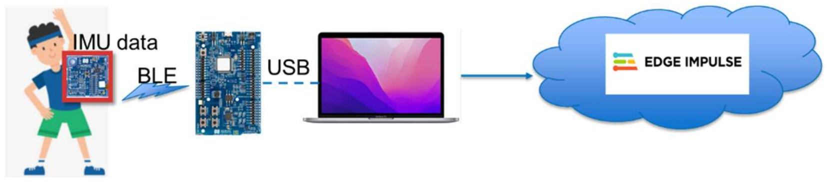
This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

Your project is private.

**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.

## IV. Collect data:

**Introduction**: this section instructs you on how to build a dataset for building a motion recognition model on EdgeImpulse. Motion data from the Thingy's accelerometer is streamed to the Edge Impulse project built in the previous section, the streamed data is then automatically saved to the created EdgeImpulse project.



The data collection involves four steps:

- Flash firmware to the Thingy52 and nrf52DK board.
- Connect and transfer collected data from the Thingy52 to  the nRF52DK over Bluetooth
- Connect your PC to your Edge Impulse project using the EdgeImpulse Data Forwarder
- Collect and label data on the Edge Impulse platform

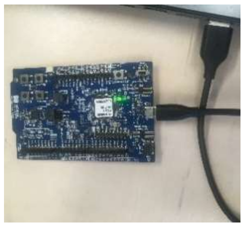
### 1. Flash firmware to Thingy52 and nrf52DK board for streaming sensor data:

a. First, you must download the streaming firmware to the Thingy52 and nrf52DK board.

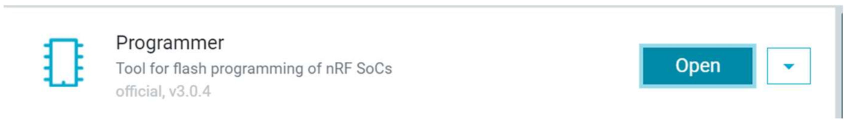
i. Download the two files (central_uart.hex and datastream.hex) from: https://github.com/vinhtqc/streamThingy

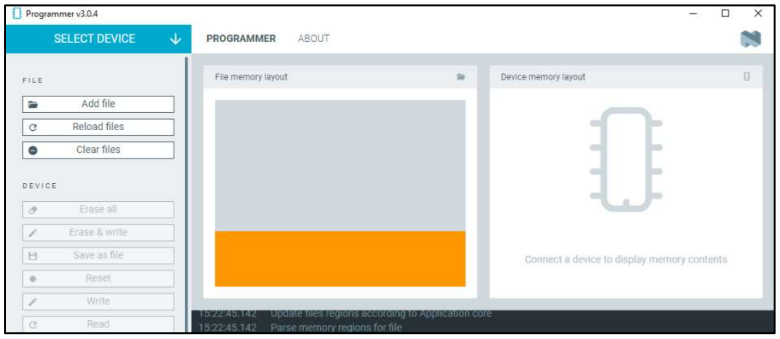
ii. Program nRF52DK:

- Connect the nRF52DK board to PC, then turn on its Power switch.



- Open "nRF Connect for Desktop" → then open "Programmer"



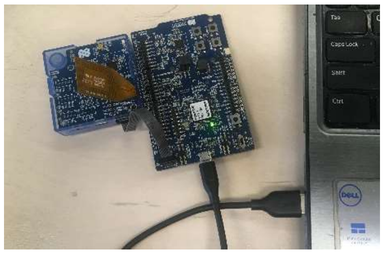


```
15:22:45.142    Update files regions according to Application core
15:22:45.142    Parse memory regions for file
```

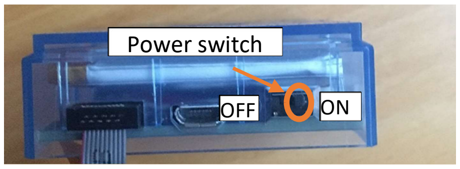
- Click “Select device” at the top left, then select nRF52DK
    a. **Note:** if you do not see the “nRF52DK”, but do see a device named similarly to “J-LINK”, you have likely not installed “SEGGER J-Link” (correctly) during the installation of “nRF Connect for desktop”. You can see installation details by referring to Section II (Part 2 - Software) of this document.
- Click “Add file”, select the file: “central_uart.hex”
- Click “Erase & Write” to flash the device and wait for finishing
- Close the “Programmer” window

iii. Program Thingy52:

- Connect Thingy52 to nRF52DK board using the JTAG cable, then connect nRF52DK board to PC.



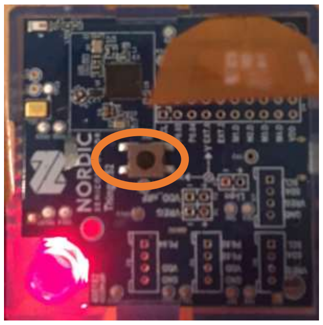
- Turn on Thingy52 and nRF52DK board



- Open nRFConnect → “Programmer”
- Click “select device” at the top left, then select nRF52DK again
- Click “Add file”, and select the file: “datastream.hex”
- Click “Erase & Write” to program to the device and wait for finishing
- Close the “Programmer” window

## 2. Connect and transfer data from Thingy to nRF52DK via Bluetooth:

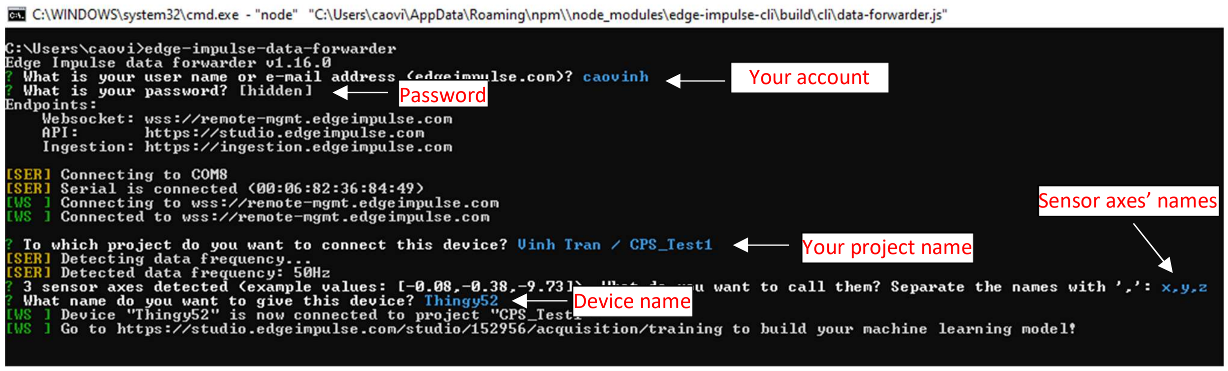
- After finishing the programing of the two devices, *carefully* unplug the JTAG cable.
- Hold the main button on the Thingy52 for 5 seconds → the LED should now be flashing quickly in Red: this means the Thingy52 is has connected and is transferring data to nRF52DK over Bluetooth.



- If the LED does not flash quickly, restart the Thingy52 (OFF then ON), then hold button for 5 seconds again. If it still does not work, try to re-program the two devices using the steps described in this section (IV – Part 1) and try again.

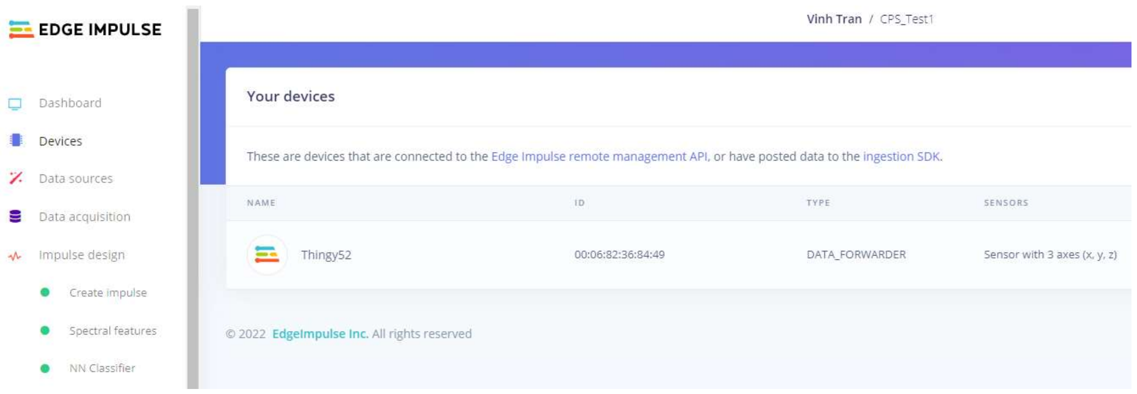
## 3. Connect your PC to your Edge Impulse project using the EdgeImpulse Data Forwarder:

- Make sure the Thingy52 is connected and transferring data to the nrf52DK board over Bluetooth (Led flashes quickly in Red) .
- On your PC, open the command line terminal.
- Type "edge-impulse-data-forwarder" (note that you should have the "Edge Impulse CLI" installed for this, see Section II – Part 2) and follow step by step the steps shown in below figure to log into EdgeImpulse and add the Thingy52 as a data collection device



```
C:\WINDOWS\system32\cmd.exe - "node"  "C:\Users\caovi\AppData\Roaming\npm\\node_modules\edge-impulse-cli\build\cli\data-forwarder.js"

C:\Users\caovi>edge-impulse-data-forwarder
Edge Impulse data forwarder v1.16.0
? What is your user name or e-mail address (edgeimpulse.com)? caovinh
? What is your password? [hidden]
Endpoints:
    Websocket: wss://remote-mgmt.edgeimpulse.com
    API:       https://studio.edgeimpulse.com
    Ingestion: https://ingestion.edgeimpulse.com

[SER] Connecting to COM8
[SER] Serial is connected (00:06:82:36:84:49)
[WS ] Connecting to wss://remote-mgmt.edgeimpulse.com
[WS ] Connected to wss://remote-mgmt.edgeimpulse.com

? To which project do you want to connect this device? Vinh Tran / CPS_Test1
[SER] Detecting data frequency...
[SER] Detected data frequency: 50Hz
? 3 sensor axes detected (example values: [-0.08,-0.38,-9.73]). What do you want to call them? Separate the names with ',': x,y,z
? What name do you want to give this device? Thingy52
[WS ] Device "Thingy52" is now connected to project "CPS_Test1"
[WS ] Go to https://studio.edgeimpulse.com/studio/152956/acquisition/training to build your machine learning model!
```

- Go to your Edge Impulse project →"Devices": you will see your device named Thingy52 (or however else you have called it) has been registered.



Your devices

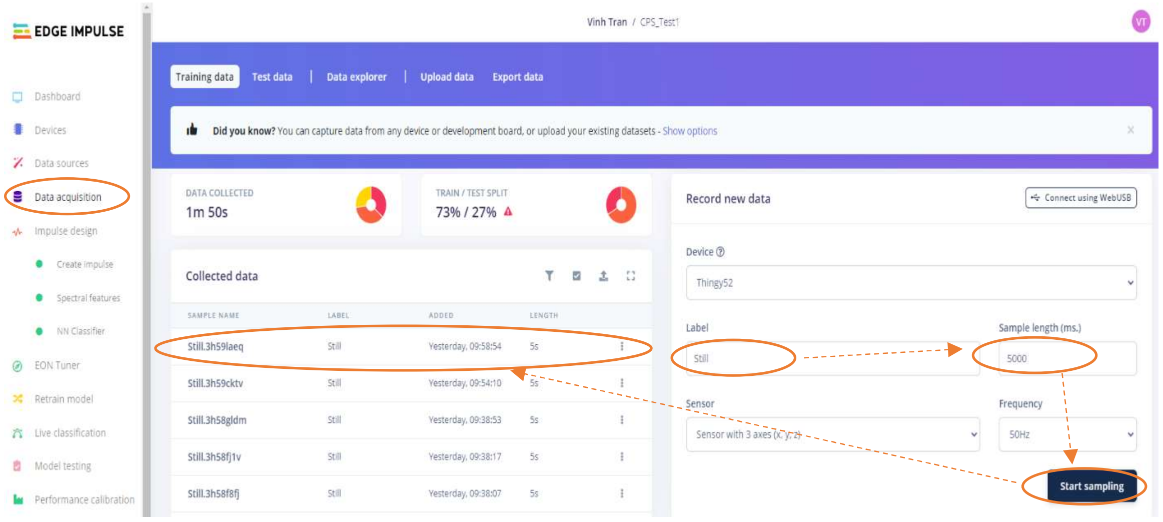
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

| NAME | ID | TYPE | SENSORS |
|---|---|---|---|
| Thingy52 | 00:06:82:36:84:49 | DATA_FORWARDER | Sensor with 3 axes (x, y, z) |

© 2022 EdgeImpulse Inc. All rights reserved

## 4. Collect and label data on Edge Impulse platform

- In your Edge Impulse project, click on "Data acquisition".



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

DATA COLLECTED 1m 50s

TRAIN / TEST SPLIT 73% / 27%

Collected data

| SAMPLE NAME | LABEL | ADDED | LENGTH |
|---|---|---|---|
| Still.3h59laeq | Still | Yesterday, 09:58:54 | 5s |
| Still.3h59cktv | Still | Yesterday, 09:54:10 | 5s |
| Still.3h58gldm | Still | Yesterday, 09:38:53 | 5s |
| Still.3h58fj1v | Still | Yesterday, 09:38:17 | 5s |
| Still.3h58f8fj | Still | Yesterday, 09:38:07 | 5s |

Record new data

Device: Thingy52

Label: Still

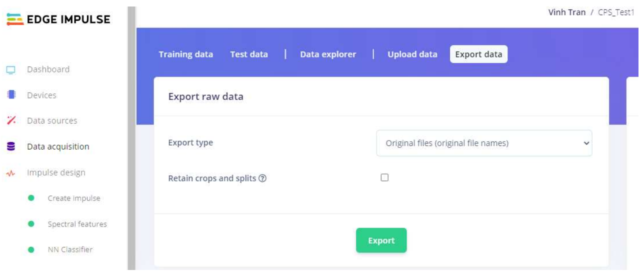
Sample length (ms.): 5000

Sensor: Sensor with 3 axes (x, y, z)

Frequency: 50Hz

- Make sure the device name you have just connected is listed as the "Device" in the "Record new data" modal on the right
- Fill in the data label (e.g: still) , the sample length of each data sample (e.g: 5000 for 5 sec)
  **Note: label should be less than 10 characters.**
- Click "Start sampling", then carry out the corresponding activity with the Thingy52. Accelerometer data is sampled at 50Hz and streamed to Edge Impulse for 5 seconds. After finishing, the data is saved and displayed in the middle modal (see screenshot).
- Continue to collect more samples. Make sure that the Thingy keeps operating while collecting. If it stops (LED off), press the middle button on Thingy (once) to wake it up.

- After collecting all data, you can download it to your PC to back-up the data or reuse it in other projects:

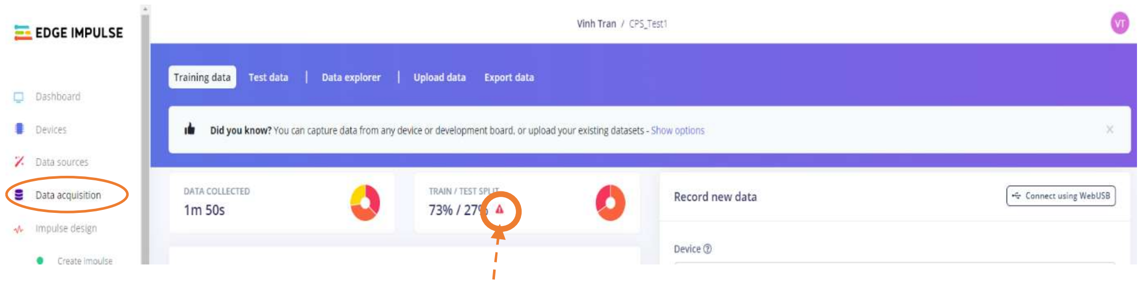


Export raw data

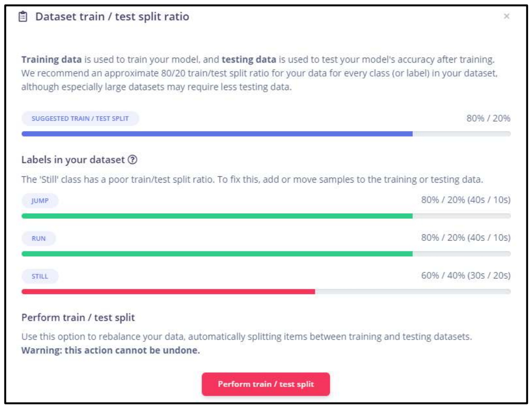
Export type: Original files (original file names)

Retain crops and splits

- Separate the dataset for training and testing purposes: After collecting all data, you need to first separate data into a training and a testing set.
  i. Click the alarm/danger icon behind TRAIN/TEST SPLIT (see screenshot) on the same page where you collected the data (Data acquisition page, Training data tab)



  ii. A popup window appears, click on the "Perform train/test split" button (please note that the "Perform train/test split" button only appears after having collected enough data samples for each class; it appears that you need at least 10 samples per label)



Dataset train / test split ratio

**Training data** is used to train your model, and **testing data** is used to test your model's accuracy after training. We recommend an approximate 80/20 train/test split ratio for your data for every class (or label) in your dataset, although especially large datasets may require less testing data.

SUGGESTED TRAIN / TEST SPLIT 80% / 20%

Labels in your dataset

The 'Still' class has a poor train/test split ratio. To fix this, add or move samples to the training or testing data.

JUMP 80% / 20% (40s / 10s)

RUN 80% / 20% (40s / 10s)

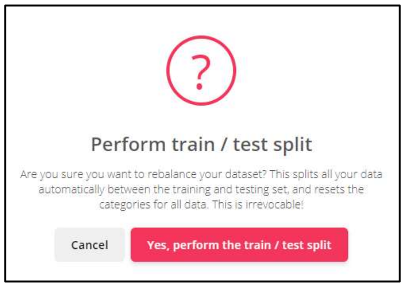
STILL 60% / 40% (30s / 20s)

Perform train / test split

Use this option to rebalance your data, automatically splitting items between training and testing datasets. **Warning: this action cannot be undone.**

Perform train / test split

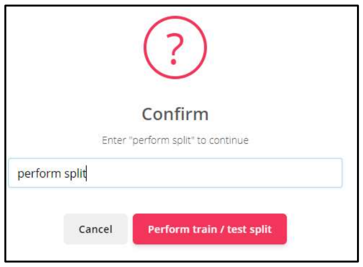
iii. Then confirm the split by clicking the "Yes, perform the train/test split" button in the next pop up window



Perform train / test split

Are you sure you want to rebalance your dataset? This splits all your data automatically between the training and testing set, and resets the categories for all data. This is irrevocable!

iv. Then, type "perform split" in the textbox and press "Perform train/test split" to confirm the split again

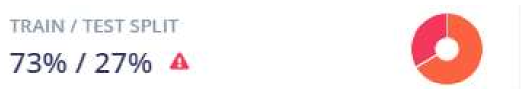


Confirm

Enter "perform split" to continue

perform split

v. After this, you can see your data has been split



TRAIN / TEST SPLIT

73% / 27%

Please note that you can also manually add data to the test dataset by navigating to the "Test data" tab on the "Data acquisition" page, and sampling data in the exact same fashion as described above for collecting training data.

## V. Build a machine learning model:

**Introduction**: This section instruct you to build and test a machine learning model for motion classification  on Edge Impulse with the collected dataset.

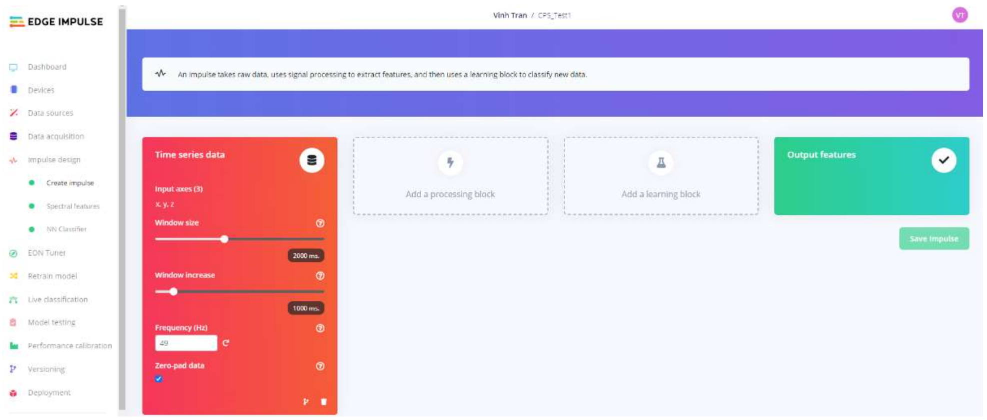
It includes the following subtopics:

- Build a model
- Test the built model.
- Save the built model.
- Tune the model
- Live Testing
- Exporting the model
- Advanced analysis

### 1. Build model:

#### *a. Create general blocks:*

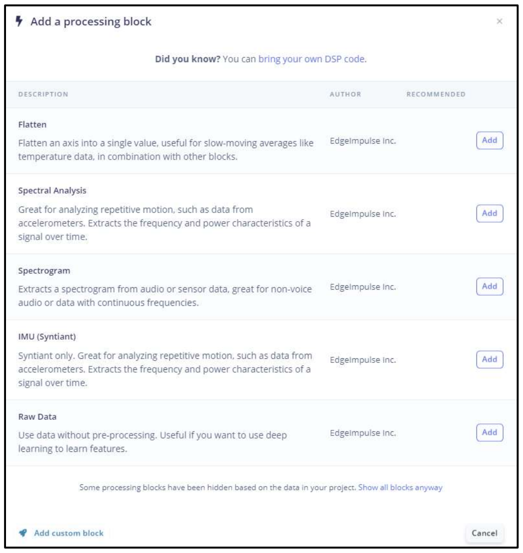
- Click on “Impulse design” → “Create Impulse”.
- Select the “window size” and “window increase”, e.g 2000 and 1000.
- Make sure the “frequency” input field corresponds to the frequency of the sampled data (collected in Section IV -  Step 4)



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

Time series data
Input axes (3)
x, y, z
Window size
2000 ms.
Window increase
1000 ms.
Frequency (Hz)
49
Zero-pad data
✓

- Then click on “Add a processing block”, select a feature extraction method from the pop-up window. E.g: “spectral analysis”



Add a processing block

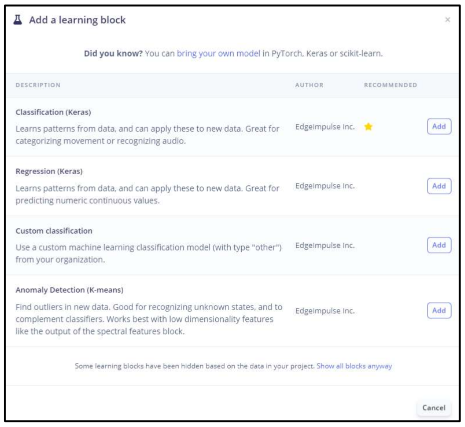
**Did you know?** You can bring your own DSP code.

| DESCRIPTION | AUTHOR | RECOMMENDED | |
|---|---|---|---|
| **Flatten** Flatten an axis into a single value, useful for slow-moving averages like temperature data, in combination with other blocks. | EdgeImpulse Inc. | | Add |
| **Spectral Analysis** Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time. | EdgeImpulse Inc. | | Add |
| **Spectrogram** Extracts a spectrogram from audio or sensor data, great for non-voice audio or data with continuous frequencies. | EdgeImpulse Inc. | | Add |
| **IMU (Syntiant)** Syntiant only. Great for analyzing repetitive motion, such as data from accelerometers. Extracts the frequency and power characteristics of a signal over time. | EdgeImpulse Inc. | | Add |
| **Raw Data** Use data without pre-processing. Useful if you want to use deep learning to learn features. | EdgeImpulse Inc. | | Add |

Some processing blocks have been hidden based on the data in your project. Show all blocks anyway

Add custom block    Cancel

- Click on "add a learning block" and select "Classification" in the pop up window.



Add a learning block

**Did you know?** You can bring your own model in PyTorch, Keras or scikit-learn.

| DESCRIPTION | AUTHOR | RECOMMENDED | |
|---|---|---|---|
| **Classification (Keras)** Learns patterns from data, and can apply these to new data. Great for categorizing movement or recognizing audio. | EdgeImpulse Inc. | ★ | Add |
| **Regression (Keras)** Learns patterns from data, and can apply these to new data. Great for predicting numeric continuous values. | EdgeImpulse Inc. | | Add |
| **Custom classification** Use a custom machine learning classification model (with type "other") from your organization. | EdgeImpulse Inc. | | Add |
| **Anomaly Detection (K-means)** Find outliers in new data. Good for recognizing unknown states, and to complement classifiers. Works best with low dimensionality features like the output of the spectral features block. | EdgeImpulse Inc. | | Add |

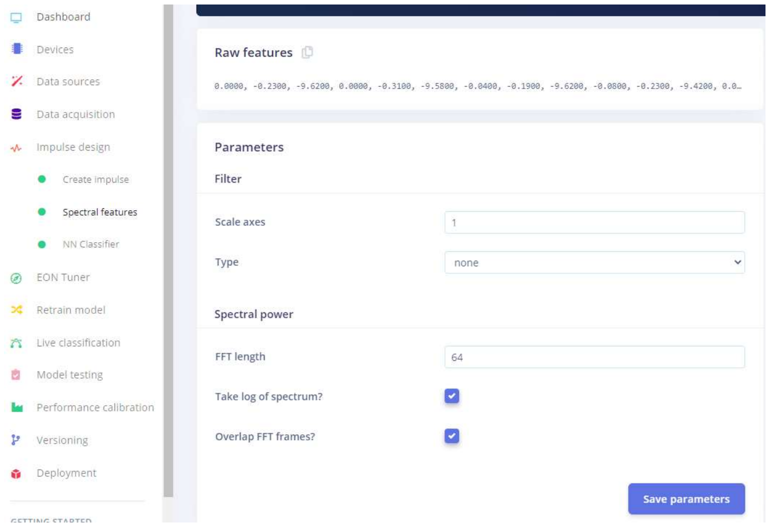
Some learning blocks have been hidden based on the data in your project. Show all blocks anyway

Cancel

- Click on "Save Impulse"

## *b. Configure and Run feature extraction:*

i. Click on "Impulse Design" -> "Spectral features", change the filter and FFT length (must be the power of 2), then click save parameters (this saves the set (default) parameters).



Raw features

```
0.0000, -0.2300, -9.6200, 0.0000, -0.3100, -9.5800, -0.0400, -0.1900, -9.6200, -0.0800, -0.2300, -9.4200, 0.0…
```

Parameters

Filter

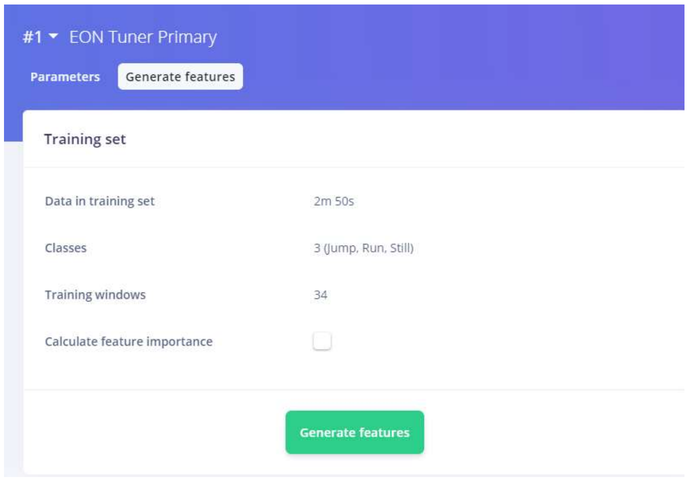
| | |
|---|---|
| Scale axes | 1 |
| Type | none |

Spectral power

| | |
|---|---|
| FFT length | 64 |
| Take log of spectrum? | ✓ |
| Overlap FFT frames? | ✓ |

Save parameters

ii. After this, click “Generate Features”. This will run a feature generation job. You can proceed to the next instruction after the feature generation job has completed.

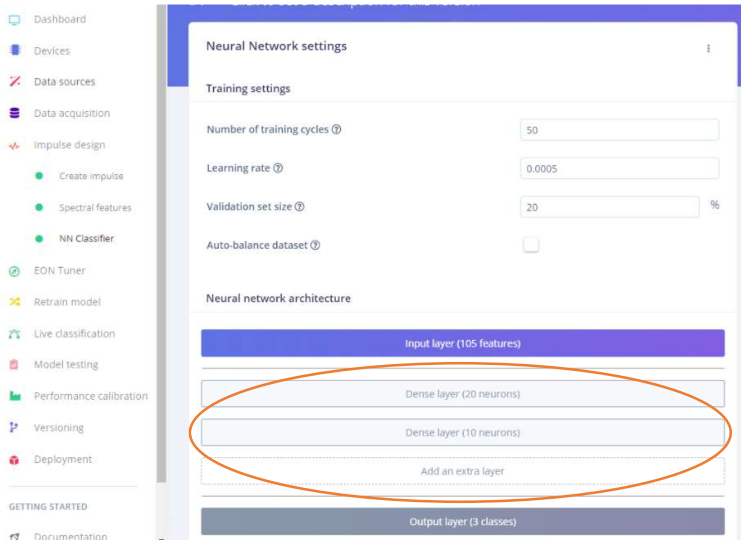


Training set

| | |
|---|---|
| Data in training set | 2m 50s |
| Classes | 3 (Jump, Run, Still) |
| Training windows | 34 |
| Calculate feature importance | ☐ |

*c. Build and run a neural network:*

i. Click on “NN Clasifier”, here you can make changes to the “Neural network architechture” section to configure and make changes to your NN. After configuring your architecture, click on Start Training.



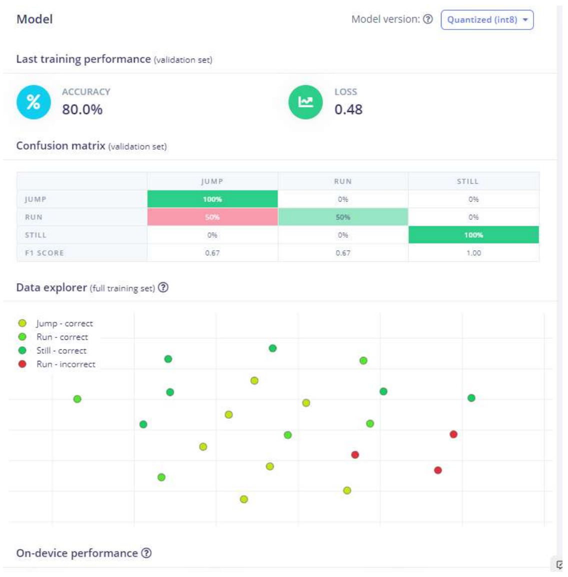
Neural Network settings

Training settings

| | |
|---|---|
| Number of training cycles | 50 |
| Learning rate | 0.0005 |
| Validation set size | 20 % |
| Auto-balance dataset | ☐ |

Neural network architecture

ii. After training, results will be shown on the right



Model — Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY 80.0%

LOSS 0.48

Confusion matrix (validation set)

| | JUMP | RUN | STILL |
|---|---|---|---|
| JUMP | 100% | 0% | 0% |
| RUN | 50% | 50% | 0% |
| STILL | 0% | 0% | 100% |
| F1 SCORE | 0.67 | 0.67 | 1.00 |

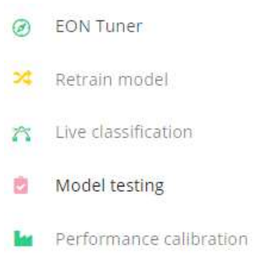
Data explorer (full training set)

On-device performance

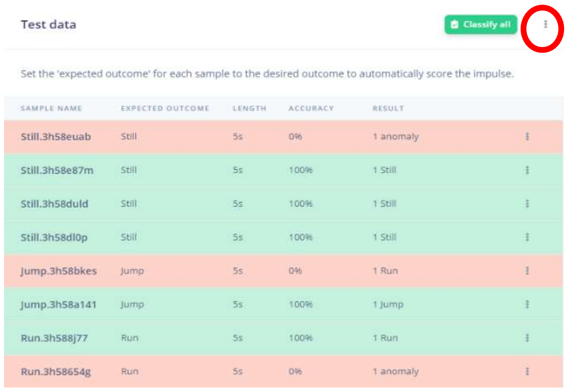
## 2. Test the model:

After training has completed, you can now test the model.

- Go to "Model testing" on the left panel



- Click on the three dot icon at the top of the middle panel

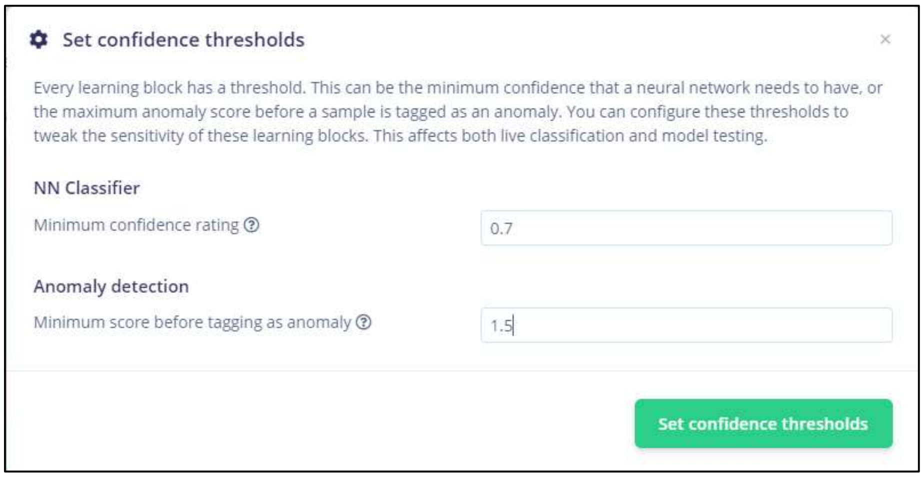


Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

| SAMPLE NAME | EXPECTED OUTCOME | LENGTH | ACCURACY | RESULT |
|---|---|---|---|---|
| Still.3h58euab | Still | 5s | 0% | 1 anomaly |
| Still.3h58e87m | Still | 5s | 100% | 1 Still |
| Still.3h58duld | Still | 5s | 100% | 1 Still |
| Still.3h58dl0p | Still | 5s | 100% | 1 Still |
| Jump.3h58bkes | Jump | 5s | 0% | 1 Run |
| Jump.3h58a141 | Jump | 5s | 100% | 1 Jump |
| Run.3h588j77 | Run | 5s | 100% | 1 Run |
| Run.3h58654g | Run | 5s | 0% | 1 anomaly |

- Set the confidence threshold level of the NN Classifier (0..1). A class is predicted as TRUE only if its confidence value is higher than this threshold, unless the result is set to "Uncertain".
- The "Anomaly detection" is only  available if you have added an anomaly block to your Neural Network architecture in section V – Part 1 (Create general blocks)



Set confidence thresholds

Every learning block has a threshold. This can be the minimum confidence that a neural network needs to have, or the maximum anomaly score before a sample is tagged as an anomaly. You can configure these thresholds to tweak the sensitivity of these learning blocks. This affects both live classification and model testing.

NN Classifier

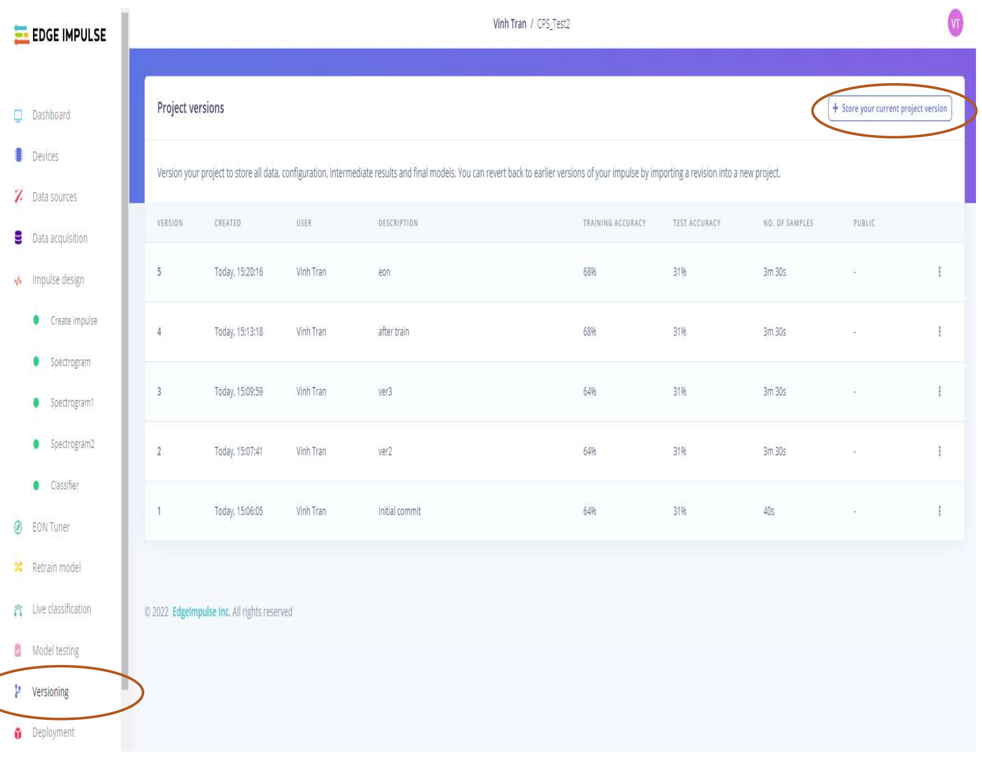
Minimum confidence rating: 0.7

Anomaly detection

Minimum score before tagging as anomaly: 1.5

## 3. Save the built model

After finishing a model, you can save the current one by: Versioning → Store your current project version



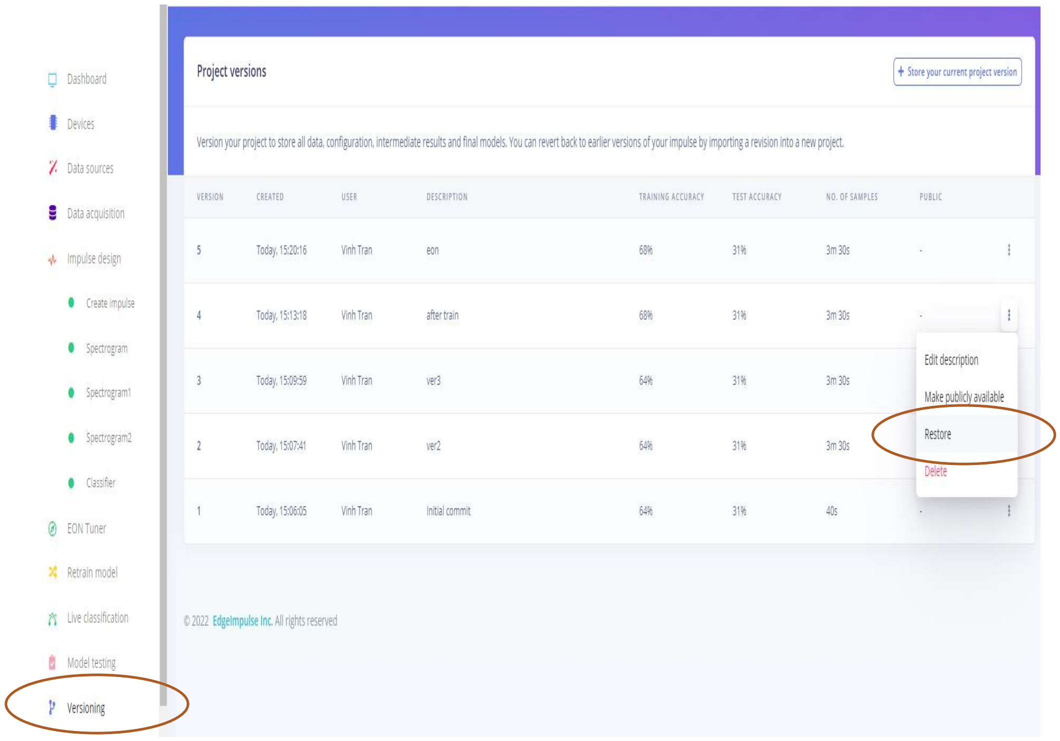
Project versions

Version your project to store all data, configuration, intermediate results and final models. You can revert back to earlier versions of your impulse by importing a revision into a new project.

| VERSION | CREATED | USER | DESCRIPTION | TRAINING ACCURACY | TEST ACCURACY | NO. OF SAMPLES | PUBLIC |
|---|---|---|---|---|---|---|---|
| 5 | Today, 15:20:16 | Vinh Tran | eon | 68% | 31% | 3m 30s | - |
| 4 | Today, 15:13:18 | Vinh Tran | after train | 68% | 31% | 3m 30s | - |
| 3 | Today, 15:09:59 | Vinh Tran | ver3 | 64% | 31% | 3m 30s | - |
| 2 | Today, 15:07:41 | Vinh Tran | ver2 | 64% | 31% | 3m 30s | - |
| 1 | Today, 15:06:05 | Vinh Tran | Initial commit | 64% | 31% | 40s | - |

© 2022 EdgeImpulse Inc. All rights reserved

Then later you can restore the saved one by go to Versioning → select the version you want to restore → click on the triple dot → Restore



Project versions

Version your project to store all data, configuration, intermediate results and final models. You can revert back to earlier versions of your impulse by importing a revision into a new project.

| VERSION | CREATED | USER | DESCRIPTION | TRAINING ACCURACY | TEST ACCURACY | NO. OF SAMPLES | PUBLIC |
|---|---|---|---|---|---|---|---|
| 5 | Today, 15:20:16 | Vinh Tran | eon | 68% | 31% | 3m 30s | - |
| 4 | Today, 15:13:18 | Vinh Tran | after train | 68% | 31% | 3m 30s | - |
| 3 | Today, 15:09:59 | Vinh Tran | ver3 | 64% | 31% | 3m 30s | |
| 2 | Today, 15:07:41 | Vinh Tran | ver2 | 64% | 31% | 3m 30s | |
| 1 | Today, 15:06:05 | Vinh Tran | Initial commit | 64% | 31% | 40s | - |

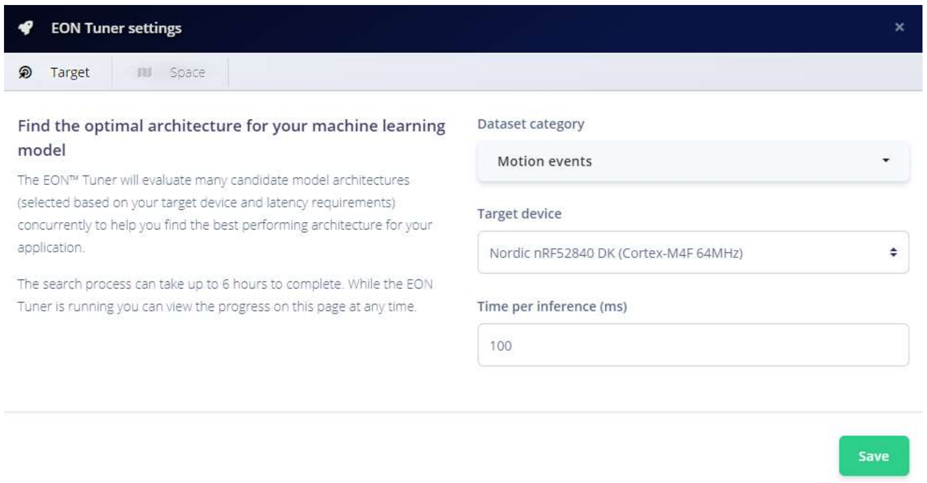
© 2022 EdgeImpulse Inc. All rights reserved

## 4. Tune the model:

In the previous section (Section V – Part 1), when creating the model, there are many parameters to consider, such as (but not limited to) the input window size, feature extraction method and the NN network architecture. These parameters can influence the model performance, and we need to investigate these influences to be able to select the most appropriate model.
The "EON Tuner" by EdgeImpulse can help you to evaluate the influence of these parameters automatically. It adjusts the parameters and evaluates the performance of each setting, after which it outputs a report for you to evaluate.

This section instructs you to use the EON Tuner to be able to find the best model based on the configurable parameters.

- On the left panel, select “EON Tuner” → “Start EON Tuner”
- A pop-up panel will appear, select the Nordic nRF52840 DK (Cortex-M4F 64MHz).


EON Tuner settings

Target | Space

**Find the optimal architecture for your machine learning model**

The EON™ Tuner will evaluate many candidate model architectures (selected based on your target device and latency requirements) concurrently to help you find the best performing architecture for your application.

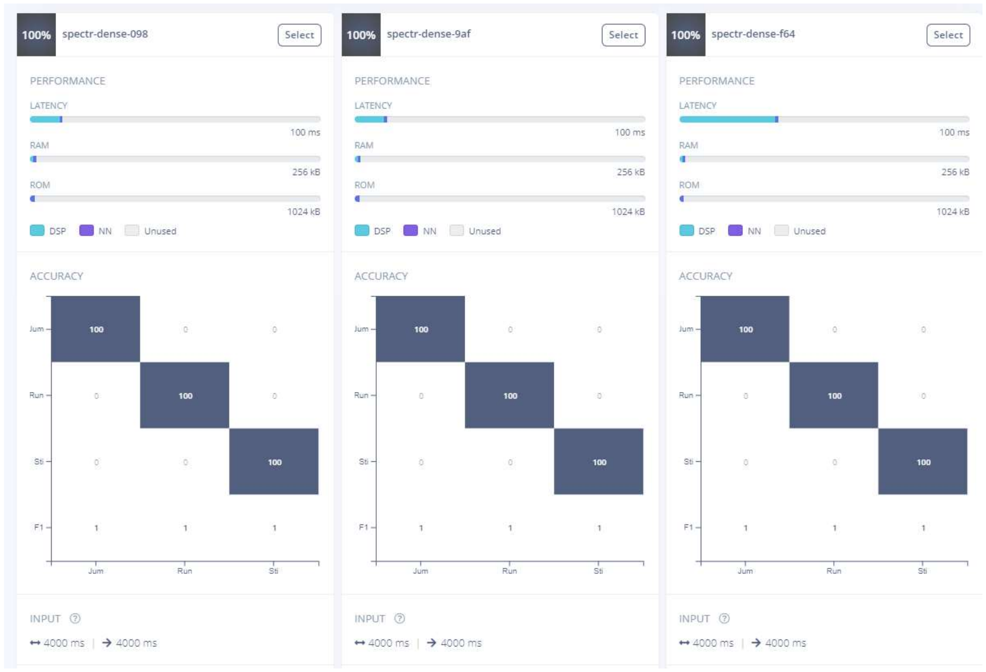
The search process can take up to 6 hours to complete. While the EON Tuner is running you can view the progress on this page at any time.

Dataset category: Motion events

Target device: Nordic nRF52840 DK (Cortex-M4F 64MHz)

Time per inference (ms): 100

- Click on Save, after which you can click on “Start EON tuner”, after which the EON tuner will run. Note that it can take quite long for the EON Tuner to finish. The result after finishing could look something like the below:


100% spectr-dense-098 | 100% spectr-dense-9af | 100% spectr-dense-f64

PERFORMANCE: LATENCY 100 ms; RAM 256 kB; ROM 1024 kB

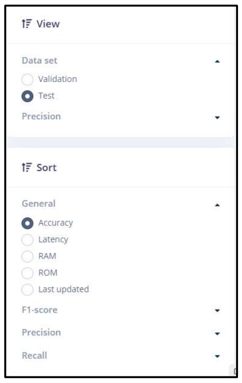
DSP NN Unused

ACCURACY

| | Jum | Run | Sti |
|---|---|---|---|
| Jum | 100 | 0 | 0 |
| Run | 0 | 100 | 0 |
| Sti | 0 | 0 | 100 |
| F1 | 1 | 1 | 1 |

INPUT: ↔ 4000 ms | → 4000 ms

- You can analyze the results from different parameter settings with regard to different metrics (accuracy,pressision, latency etc.). Each set of parameters produces different results.



View

Data set
- Validation
- [✓ Test]

Precision

Sort

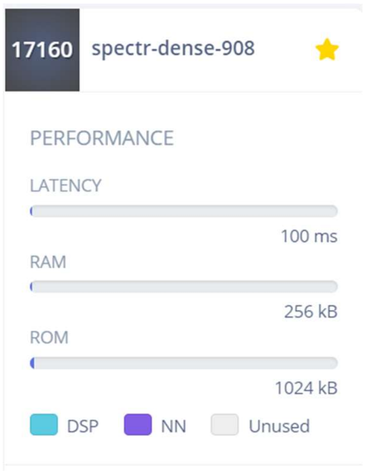
General
- [✓ Accuracy]
- Latency
- RAM
- ROM
- Last updated

F1-score

Precision

Recall

- Select what you think is the best model by clicking on “Select” on the model. Note: The Thingy52 has less resources available than the nRF52832DK. You must select a model that uses less than **128KB** ROM and less than **12KB** RAM (you can evaluate this by seeing how much of the resource usage bar is filled in, the number under the bars for the RAM and ROM reflect the maximum available RAM and ROM available on target board, which in our case is the  nRF52832DK)



17160 spectr-dense-908

PERFORMANCE

LATENCY 100 ms

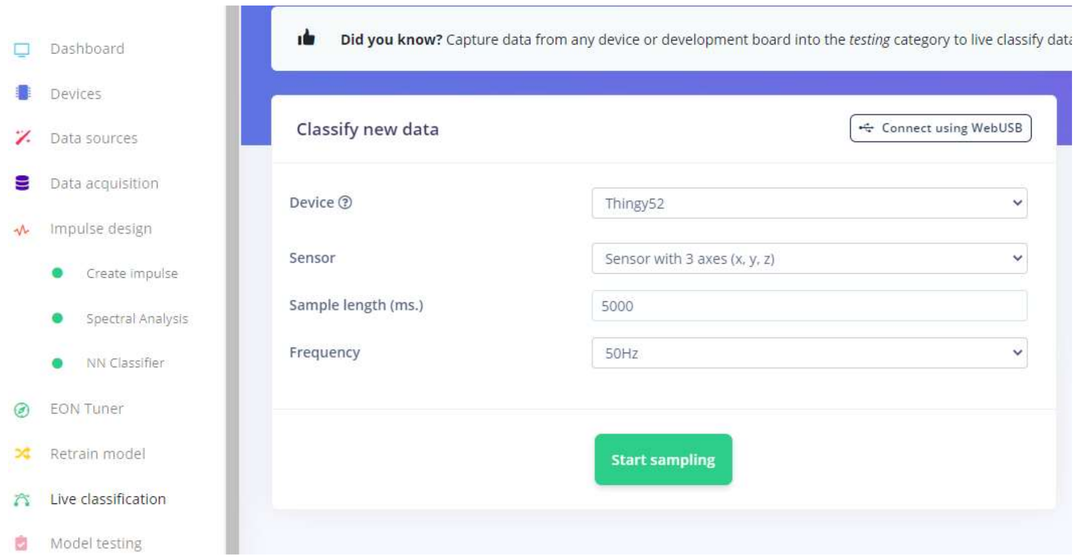
RAM 256 kB

ROM 1024 kB

**Note: the current model will be lost when you apply a model built by EON. You should save the current model before applying a model built by EON unless the current one will be lost and can not be recovered.**

## 5. Live Test

- After building the model, you can test it live with a data stream provided by the Thingy52.
- Go to "Live classification", fill in the required parameters and click "Start sampling"



**Did you know?** Capture data from any device or development board into the *testing* category to live classify dat

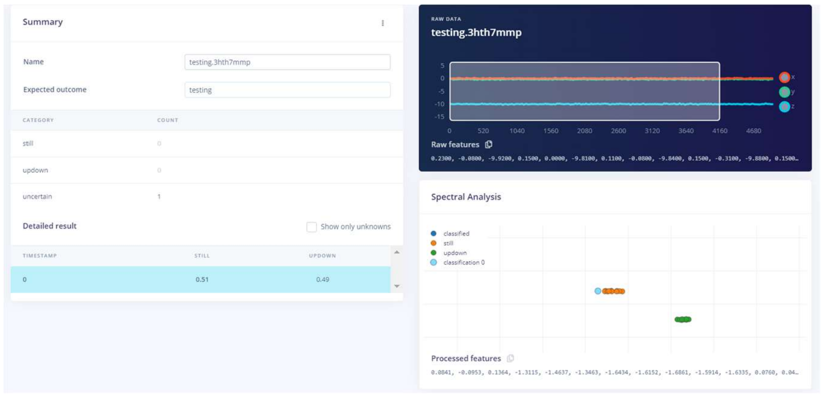
Classify new data

Connect using WebUSB

|  |  |
|---|---|
| Device | Thingy52 |
| Sensor | Sensor with 3 axes (x, y, z) |
| Sample length (ms.) | 5000 |
| Frequency | 50Hz |

Start sampling

- After a sample was taken, a classification result is provided. For spectral analysis feature extraction, a 2D graph is shown in which the new datapoint is plotted together with the labeled datapoints (to provide a visual intuition with regard to the classification result)



Summary

|  |  |
|---|---|
| Name | testing.3hth7mmp |
| Expected outcome | testing |

| CATEGORY | COUNT |
|---|---|
| still | 0 |
| updown | 0 |
| uncertain | 1 |

Detailed result

Show only unknowns

| TIMESTAMP | STILL | UPDOWN |
|---|---|---|
| 0 | 0.51 | 0.49 |

testing.3hth7mmp

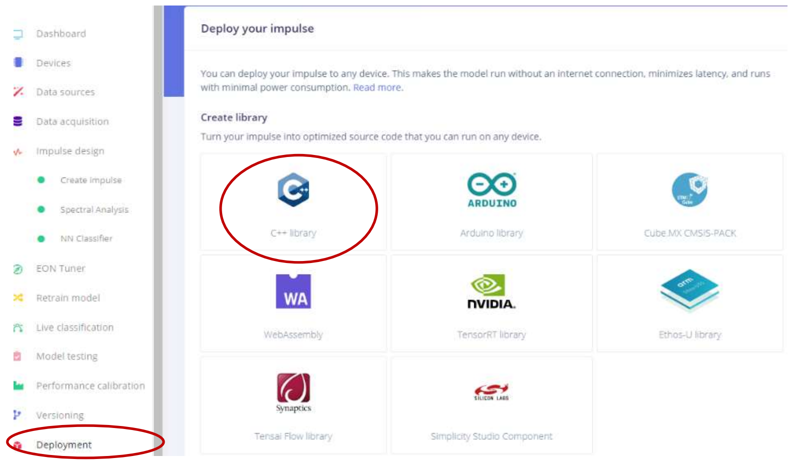
Raw features

```
0.2300, -0.0800, -9.9200, 0.1500, 0.0000, -9.8100, 0.1100, -0.0800, -9.8400, 0.1500, -0.3100, -9.8800, 0.1500…
```

Processed features

```
0.0841, -0.0953, 0.1364, -1.3115, -1.4637, -1.3463, -1.6434, -1.6152, -1.6861, -1.5914, -1.6335, 0.0760, 0.04…
```

6. Export the model:
   - When satisfied with the model performance, you can export it to integrate the model into the Thingy52.
   - Go to “Deployment”, select “C++ library”, then scroll down to the end of the middle window, keep the optimization method as “Quantized (int8)” and click “Build”.



You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Turn your impulse into optimized source code that you can run on any device.

   - Wait for the building to finish, after which a popup window will appear prompting to save the file, e.g: “cps_test1.zip”

7. Advanced analyzing:
   After you have gotten used to the basic functionalities of Edge Impulse, you can play with the advanced features to further improve your model, which is elaborated this video:

   https://www.youtube.com/watch?v=7vr4D_zlQTE

## VI. Integrate the model into Thingy52:

This section instructs you on how to integrate the model built in the previous section (V) to a prebuilt firmware and flash it to the Thingy52.

This process includes 5 steps :

- Preparation
- Load firmware file to the VSCode SDK
- Configuration for integrating the built ML model.
- Compile firmware
- Flash firmware

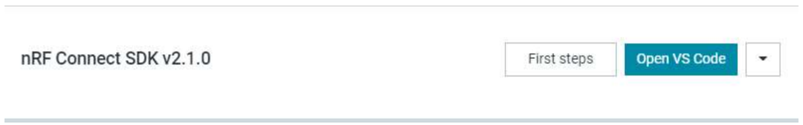
## 1. Preparation:

- Connect the Thingy52 to the nrf52DK board using the JTAG cable, then connect the nrf52DK board to your PC using a usb port. Make sure all devices are turned *ON*.

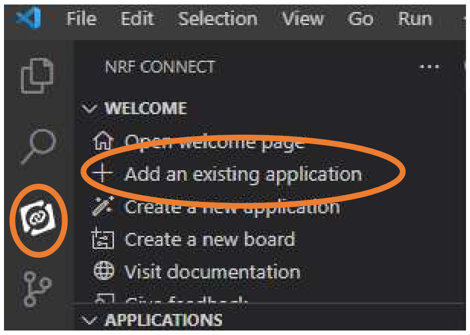
- Download and unpack the file “machine_learning2.rar” to your PC from: https://github.com/vinhtqc/CPS2022_ML

## 2. Load firmware file to the VS Code SDK:

- Open “nRF Connect for Desktop” → Toochain manager → Click “Open VsCode” under the nrF SDK version downloaded in Section II to open Visual studio code

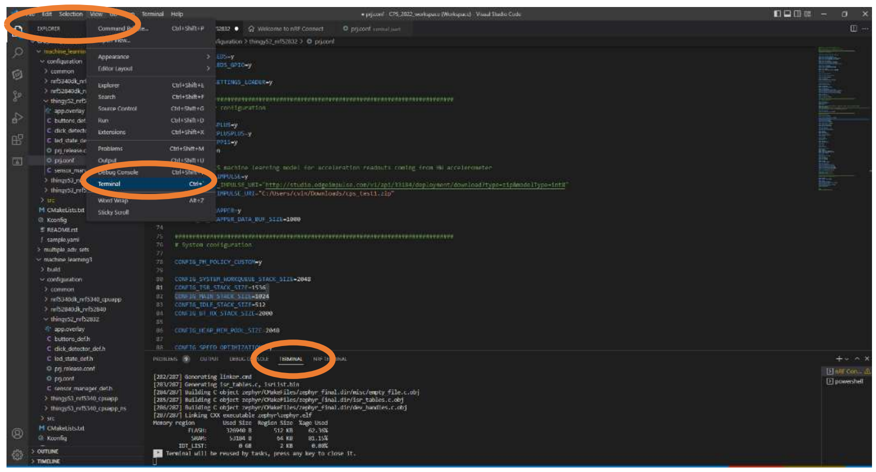


- In “VSCode”, on the left sidebar, click on the “nRF Connect” icon (or press Ctrl + Alt + N)
- Then, click “Add an existing application”



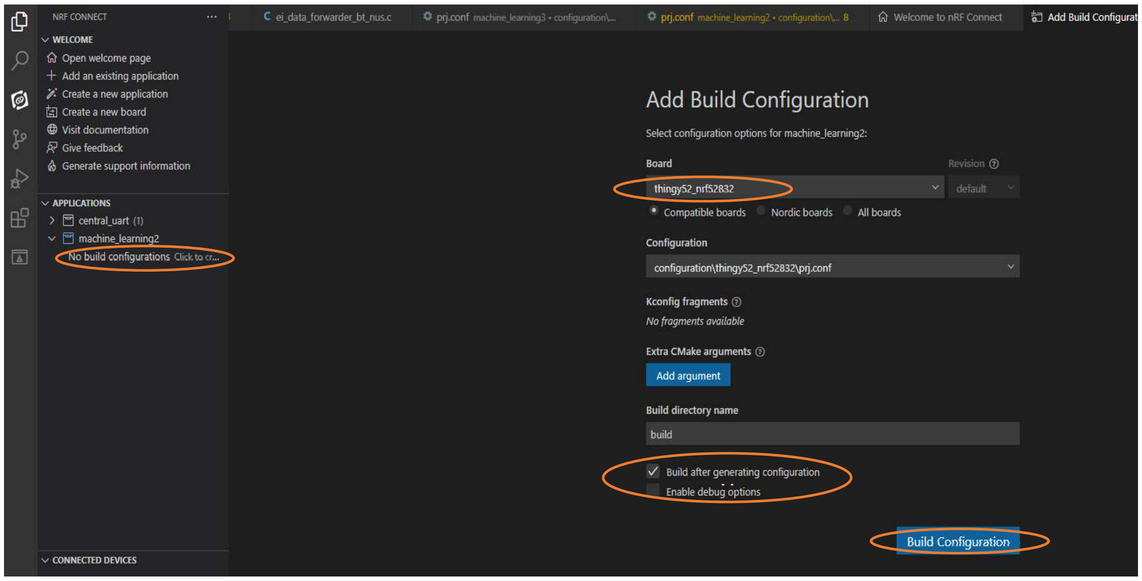
- Select the folder of “machine_learning2” which has been (unpacked). Wait for the loading of the software to be done.

- Click on View -> Terminal (or Ctrl + `). The Terminal tab will appear at the bottom. It will help you to trace your work progress.



```
[282/287] Generating linker.cmd
[283/287] Generating isr_tables.c, isrList.bin
[284/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/misc/empty_file.c.obj
[285/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/isr_tables.c.obj
[286/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/dev_handles.c.obj
[287/287] Linking CXX executable zephyr\zephyr.elf
Memory region         Used Size  Region Size  %age Used
           FLASH:      326940 B       512 KB     62.36%
            SRAM:       53184 B        64 KB     81.15%
        IDT_LIST:          0 GB         2 KB      0.00%
 Terminal will be reused by tasks, press any key to close it.
```

- In the “Applications” bar on the left side, select “machine_learning2”, click on “No build configurations”, a configuration window will appear as the below picture on the right. **Note**, if you receive an error message stating: “Couldn’t resolve Zephyr base from …”, please make sure you have installed nRF Command Line Tools (and all other dependencies as well), as described in Section II – Part 2)
- After this, set the parameters to those shown in the below figure. Wait for the initialization to complete, you can track the progress in the “Terminal” tab (bottom)



NRF CONNECT
WELCOME
Open welcome page
Add an existing application
Create a new application
Create a new board
Visit documentation
Give feedback
Generate support information
APPLICATIONS
central_uart (1)
machine_learning2
No build configurations Click to cr...
CONNECTED DEVICES
ei_data_forwarder_bt_nus.c
prj.conf machine_learning3 • configuration\...
prj.conf machine_learning2 • configuration\... 8
Welcome to nRF Connect
Add Build Configurat

Add Build Configuration

Select configuration options for machine_learning2:

Board: thingy52_nrf52832 — Revision: default

● Compatible boards ○ Nordic boards ○ All boards

Configuration: configuration\thingy52_nrf52832\prj.conf

Kconfig fragments: *No fragments available*

Extra CMake arguments: Add argument

Build directory name: build

✓ Build after generating configuration

☐ Enable debug options

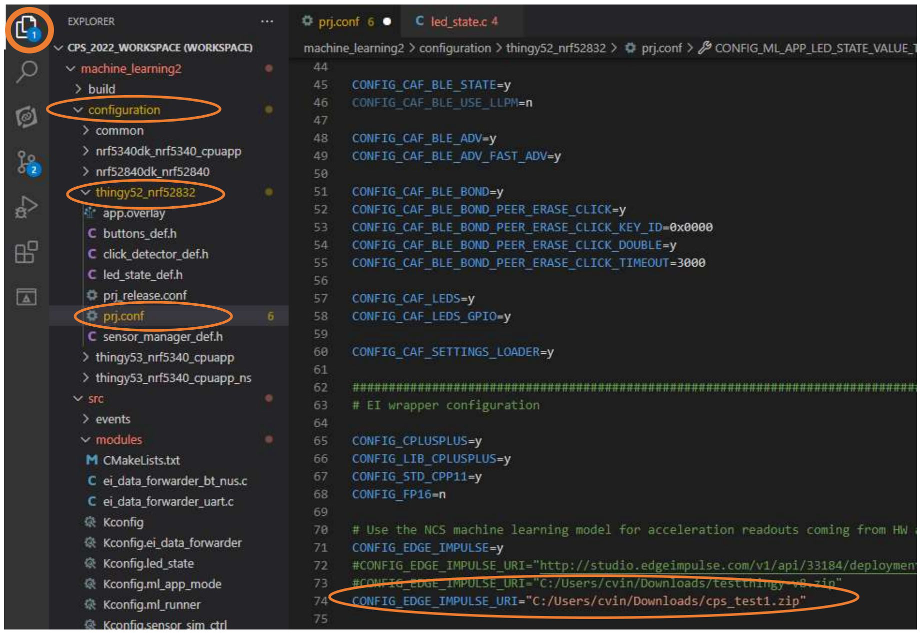
Build Configuration

## 4. Configuration for integrating the built machine learning model

a. Select the **"Explorer"** icon in the left column, then select **"machine_learning2" → "configuration" → "thingy52_nrf52832"**

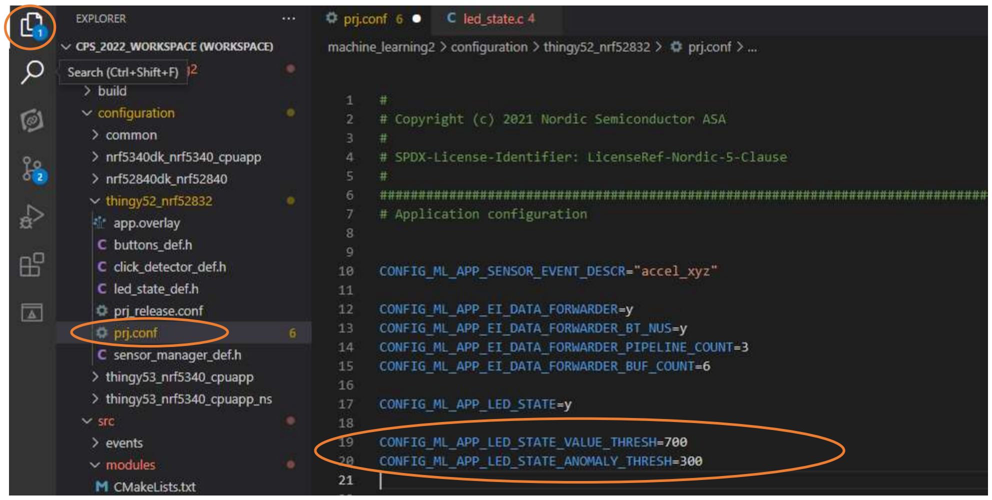
   i. Select **"prj.conf"**, the file should now appear on the right panel

   - On the right panel, scroll to line with the text: "CONFIG_EDGE_IMPULSE_URI", change the path to the location of the model file (cps_test1.zip, **unpacked folder**) that you built and downloaded before.
     - Note: slashes must be forward slashes ('/').



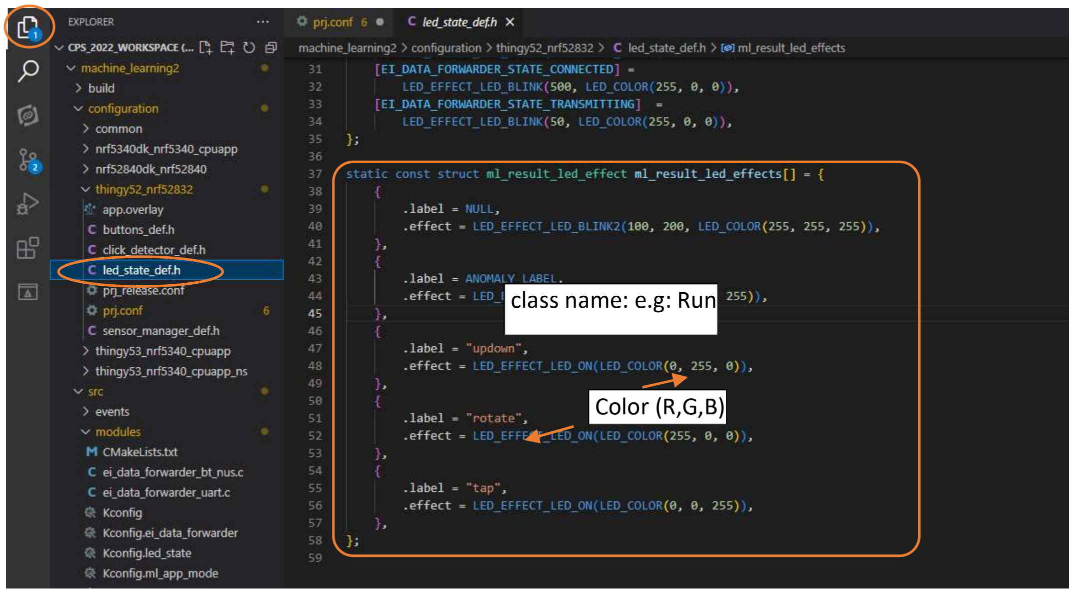
```

CONFIG_CAF_BLE_STATE=y
CONFIG_CAF_BLE_USE_LLPM=n

CONFIG_CAF_BLE_ADV=y
CONFIG_CAF_BLE_ADV_FAST_ADV=y

CONFIG_CAF_BLE_BOND=y
CONFIG_CAF_BLE_BOND_PEER_ERASE_CLICK=y
CONFIG_CAF_BLE_BOND_PEER_ERASE_CLICK_KEY_ID=0x0000
CONFIG_CAF_BLE_BOND_PEER_ERASE_CLICK_DOUBLE=y
CONFIG_CAF_BLE_BOND_PEER_ERASE_CLICK_TIMEOUT=3000

CONFIG_CAF_LEDS=y
CONFIG_CAF_LEDS_GPIO=y

CONFIG_CAF_SETTINGS_LOADER=y

################################################################################
# EI wrapper configuration

CONFIG_CPLUSPLUS=y
CONFIG_LIB_CPLUSPLUS=y
CONFIG_STD_CPP11=y
CONFIG_FP16=n

# Use the NCS machine learning model for acceleration readouts coming from HW
CONFIG_EDGE_IMPULSE=y
#CONFIG_EDGE_IMPULSE_URI="http://studio.edgeimpulse.com/v1/api/33184/deploymen
#CONFIG_EDGE_IMPULSE_URI="C:/Users/cvin/Downloads/testthingy-v8.zip"
CONFIG_EDGE_IMPULSE_URI="C:/Users/cvin/Downloads/cps_test1.zip"

```

   - Scroll through the right panel to the line with the text:
     **CONFIG_ML_APP_LED_STATE_VALUE_THRESH** and
     **CONFIG_ML_APP_LED_STATE_ANOMALY_THRESH**
     Set these two values equal to the two values you defined in Section V – Part 2 (Test the model), and multiply this number by 1000.



```
#
# Copyright (c) 2021 Nordic Semiconductor ASA
#
# SPDX-License-Identifier: LicenseRef-Nordic-5-Clause
#
##########################################################################################
# Application configuration


CONFIG_ML_APP_SENSOR_EVENT_DESCR="accel_xyz"

CONFIG_ML_APP_EI_DATA_FORWARDER=y
CONFIG_ML_APP_EI_DATA_FORWARDER_BT_NUS=y
CONFIG_ML_APP_EI_DATA_FORWARDER_PIPELINE_COUNT=3
CONFIG_ML_APP_EI_DATA_FORWARDER_BUF_COUNT=6

CONFIG_ML_APP_LED_STATE=y

CONFIG_ML_APP_LED_STATE_VALUE_THRESH=700
CONFIG_ML_APP_LED_STATE_ANOMALY_THRESH=300
```

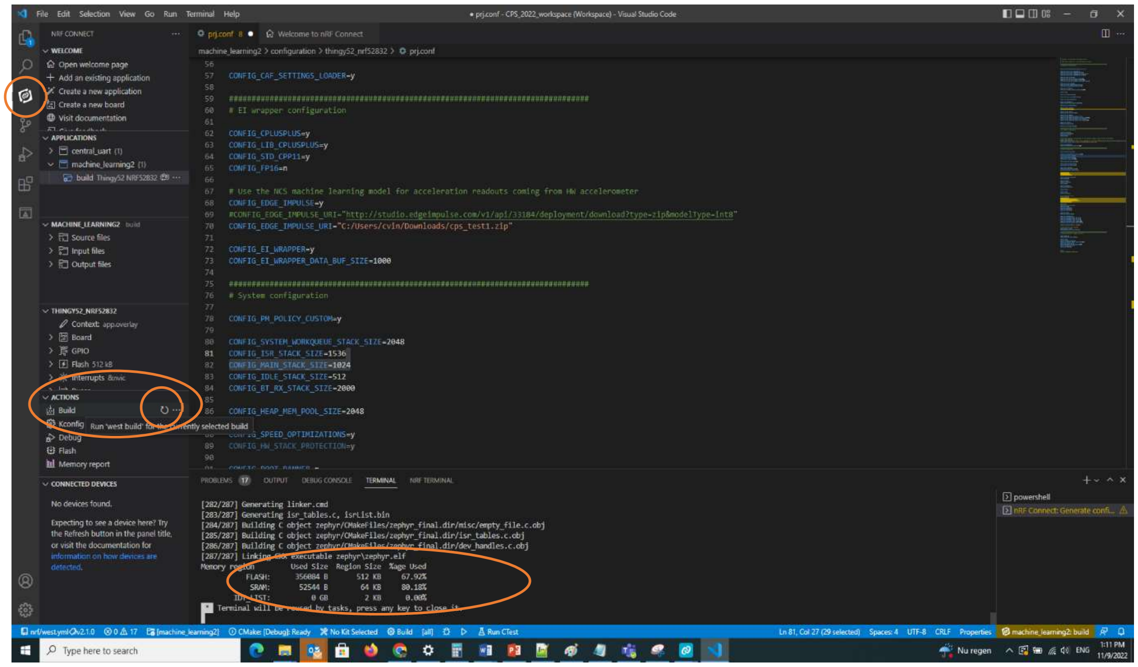
ii. Select **"led_state_def.h"** in the left column. Then, in the right window, edit the **"ml_result_led_effect"** to match your model's label names defined in section IV – Part 4 (Collect and label data on Edge Impulse platform.) Add or remove fields to this structure if your built model has more or less labels than the ones predefined. Furthermore, you can define a color for the added labels.



```
    [EI_DATA_FORWARDER_STATE_CONNECTED] =
        LED_EFFECT_LED_BLINK(500, LED_COLOR(255, 0, 0)),
    [EI_DATA_FORWARDER_STATE_TRANSMITTING]  =
        LED_EFFECT_LED_BLINK(50, LED_COLOR(255, 0, 0)),
};

static const struct ml_result_led_effect ml_result_led_effects[] = {
    {
        .label = NULL,
        .effect = LED_EFFECT_LED_BLINK2(100, 200, LED_COLOR(255, 255, 255)),
    },
    {
        .label = ANOMALY_LABEL,
        .effect = LED_                              255)),
    },
    {
        .label = "updown",
        .effect = LED_EFFECT_LED_ON(LED_COLOR(0, 255, 0)),
    },
    {
        .label = "rotate",
        .effect = LED_EFFECT_LED_ON(LED_COLOR(255, 0, 0)),
    },
    {
        .label = "tap",
        .effect = LED_EFFECT_LED_ON(LED_COLOR(0, 0, 255)),
    },
};
```

## 5. Compile firmware

- Select the "nRF Connect" icon in the left sidebar.
- Go to the "Actions" section in the bottom-left, Click the "Pristine Build" icon (self-pointing arrow, next to "Build"). Wait for the build to complete, after this, you will see a report of the amount of RAM and FLASH memory that will be used on the Thingy52 in the Terminal window

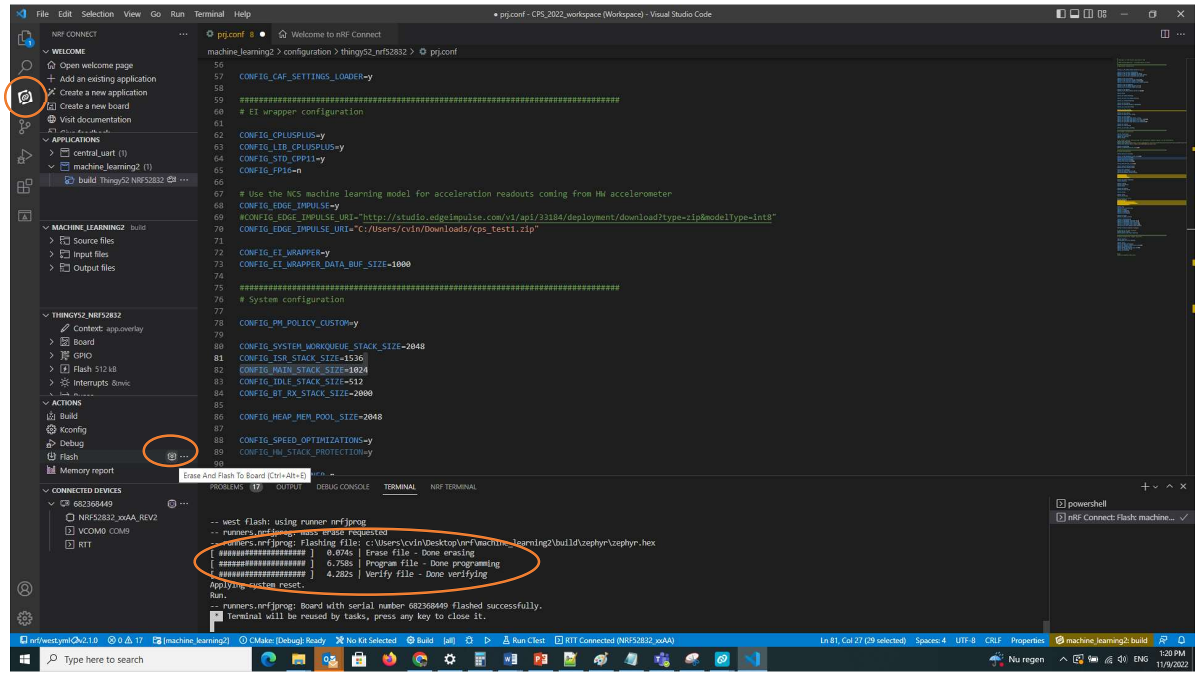


```
CONFIG_CAF_SETTINGS_LOADER=y

################################################################################
# EI wrapper configuration

CONFIG_CPLUSPLUS=y
CONFIG_LIB_CPLUSPLUS=y
CONFIG_STD_CPP11=y
CONFIG_FP16=n

# Use the NCS machine learning model for acceleration readouts coming from HW accelerometer
CONFIG_EDGE_IMPULSE=y
#CONFIG_EDGE_IMPULSE_URI="http://studio.edgeimpulse.com/v1/api/33184/deployment/download?type=zip&modelType=int8"
CONFIG_EDGE_IMPULSE_URI="C:/Users/cvin/Downloads/cps_test1.zip"

CONFIG_EI_WRAPPER=y
CONFIG_EI_WRAPPER_DATA_BUF_SIZE=1000

################################################################################
# System configuration

CONFIG_PM_POLICY_CUSTOM=y

CONFIG_SYSTEM_WORKQUEUE_STACK_SIZE=2048
CONFIG_ISR_STACK_SIZE=1536
CONFIG_MAIN_STACK_SIZE=1024
CONFIG_IDLE_STACK_SIZE=512
CONFIG_BT_RX_STACK_SIZE=2000

CONFIG_HEAP_MEM_POOL_SIZE=2048

CONFIG_SPEED_OPTIMIZATIONS=y
CONFIG_HW_STACK_PROTECTION=y
```

```
[282/287] Generating linker.cmd
[283/287] Generating isr_tables.c, isrList.bin
[284/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/misc/empty_file.c.obj
[285/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/isr_tables.c.obj
[286/287] Building C object zephyr/CMakeFiles/zephyr_final.dir/dev_handles.c.obj
[287/287] Linking CXX executable zephyr\zephyr.elf
Memory region         Used Size  Region Size  %age Used
           FLASH:      356084 B       512 KB     67.92%
            SRAM:       52544 B        64 KB     80.18%
        IDT_LIST:          0 GB         2 KB      0.00%
 Terminal will be reused by tasks, press any key to close it.
```

## 6. Flash firmware to Thingy52

Click the "Erase and Flash" icon (small downward pointing arrow, next to "Flash") to flash the program to the Thingy52. Wait until the Terminal reports the flashing process has finished. After this, disconnect the Thingy52.

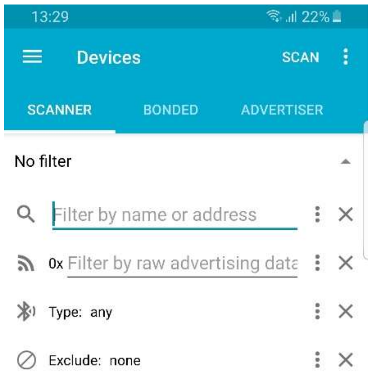


```
-- west flash: using runner nrfjprog
-- runners.nrfjprog: mass erase requested
-- runners.nrfjprog: Flashing file: c:\Users\cvin\Desktop\nrf\machine_learning2\build\zephyr\zephyr.hex
[ #################### ]   0.074s | Erase file - Done erasing
[ #################### ]   6.758s | Program file - Done programming
[ #################### ]   4.282s | Verify file - Done verifying
Applying system reset.
Run.
-- runners.nrfjprog: Board with serial number 682368449 flashed successfully.
 Terminal will be reused by tasks, press any key to close it.
```

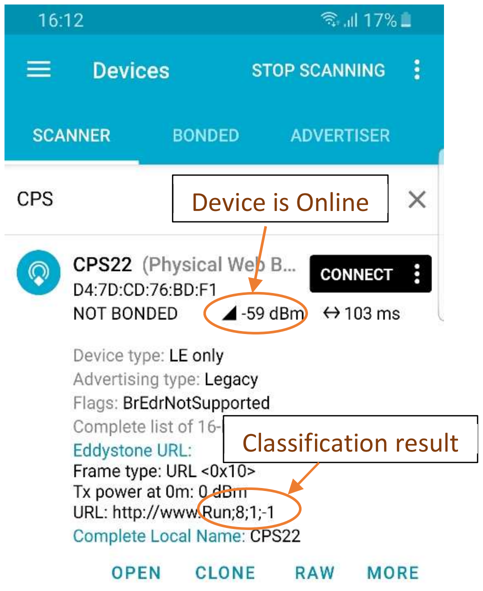
## VII. Demonstration:

This section instructs you to view the Thingy52 motion classification results on an Android phone.

- Go to Android Store (Google Play) and download the “nrfConnect” app to your phone
- Open the app and allow the required permissions
- Click the small triangle on the right edge to how the filter settings. Type “CPS” to the name field. After this, click the triangle to close the filter.



- Click the “SCAN” button. The Thingy52 should now appear in the list of devices found. The result of classification can be found in the line: “URL:....www.Class;n,m,k” , with “Class” being the name of one of your model classes, namely the predicted class (i.e. “Run”), “n” being the feature extraction time, “m” being the classification time and “k” being the anomaly detection time (if there is an anomaly block in your model)



Device type: LE only
Advertising type: Legacy
Flags: BrEdrNotSupported
Complete list of 16-
Eddystone URL:
Frame type: URL <0x10>
Tx power at 0m: 0 dBm
URL: http://www.Run;8;1;-1
Complete Local Name: CPS22

## VII. Updated history:

- Manual_CPS_5: add trouble shooting for installing NodeJs tools in section II.2.d