## Introduction

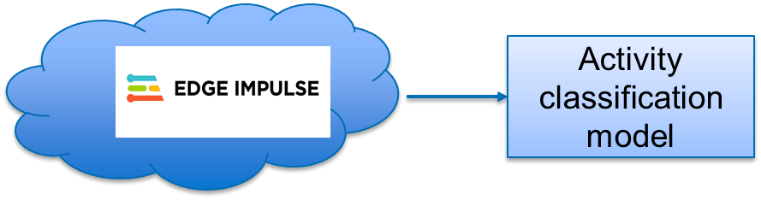
This document contains instructions for you to build and run a machine learning model on an edge device. In particular, you will develop a motion classification model, and run this on a Thingy52, a multi-sensor prototyping platform from Nordic Semiconductor. The classification model is built by using 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. Create an EdgeImpulse account and project
  3. Collect data:



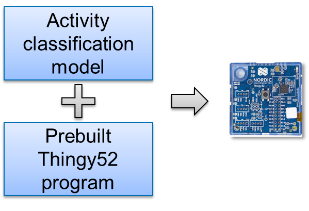
* 1. Build, Train, Test and Tune a model:



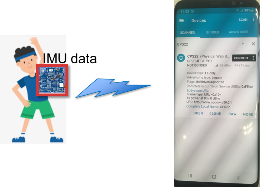
* 1. Live test and export the best model



* 1. Download model to Thingy52



* 1. Demonstrate the real-world results:



## II Preparation:

### Hardware:

* Thingy52
* Nrf52DK
* JTAG cable 1.27mm

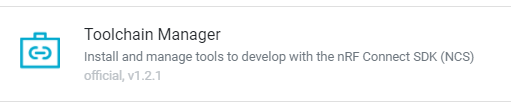
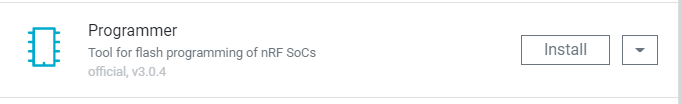
### Software:

* 1. **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”



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



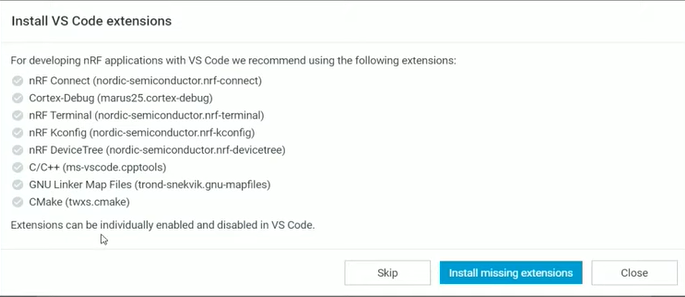
* Close all tools and nRF Connect for desktop after installation has completed.

* 1. **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/>

* 1. **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”



* 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>

Graphical user interface, text, application

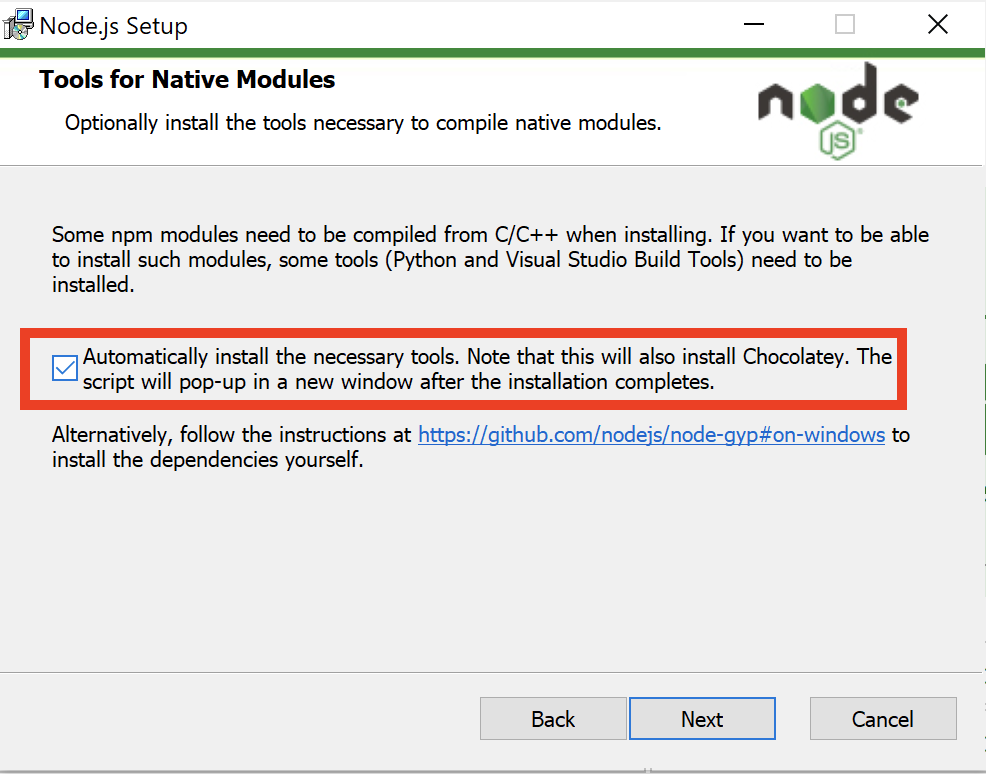
Description automatically generated

* 1. **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>

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](https://github.com/coreybutler/nvm-windows) / [Linux / MacOS](https://github.com/nvm-sh/nvm))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.



* + - * 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).

## Create an account and project on Edge Impulse

### 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.

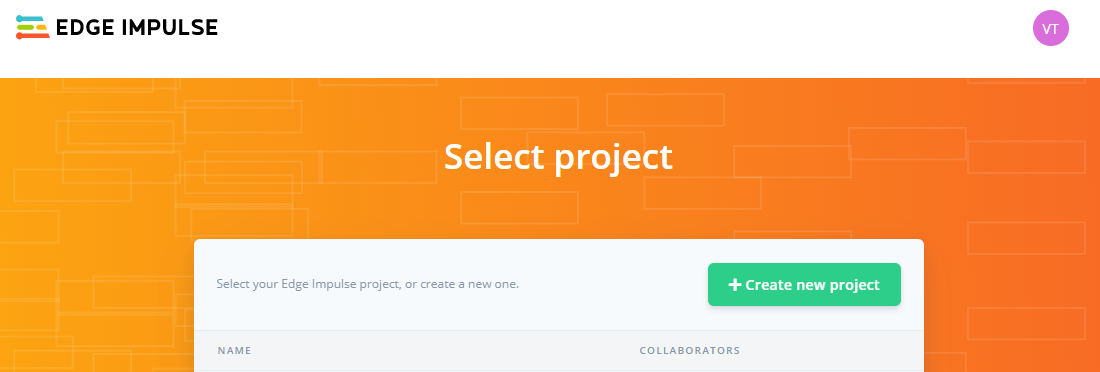
### Account registration:

To start using Edge Impulse, first sign up for an account using this link:

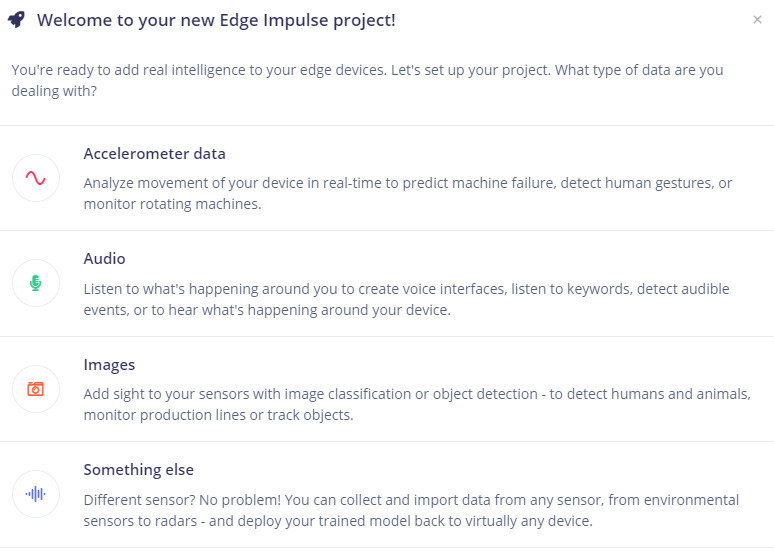
<https://www.edgeimpulse.com/>

### Create an Edge Impulse project:

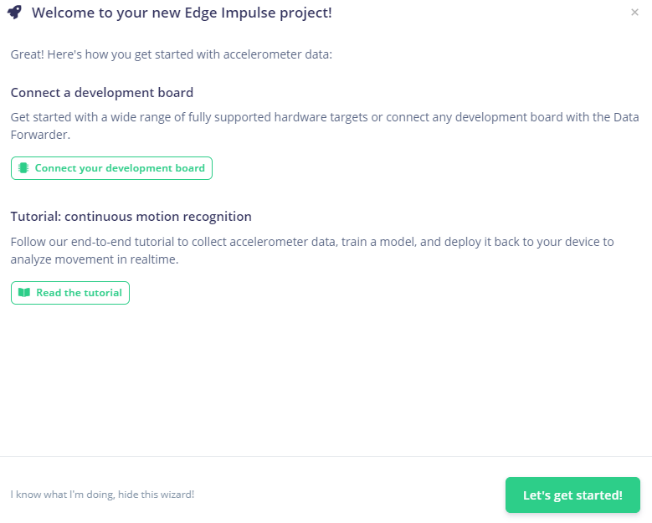
* + - * Log in
      * Click on “Create new Project” and name the project in the pop up window, e.g: CPS\_Test1

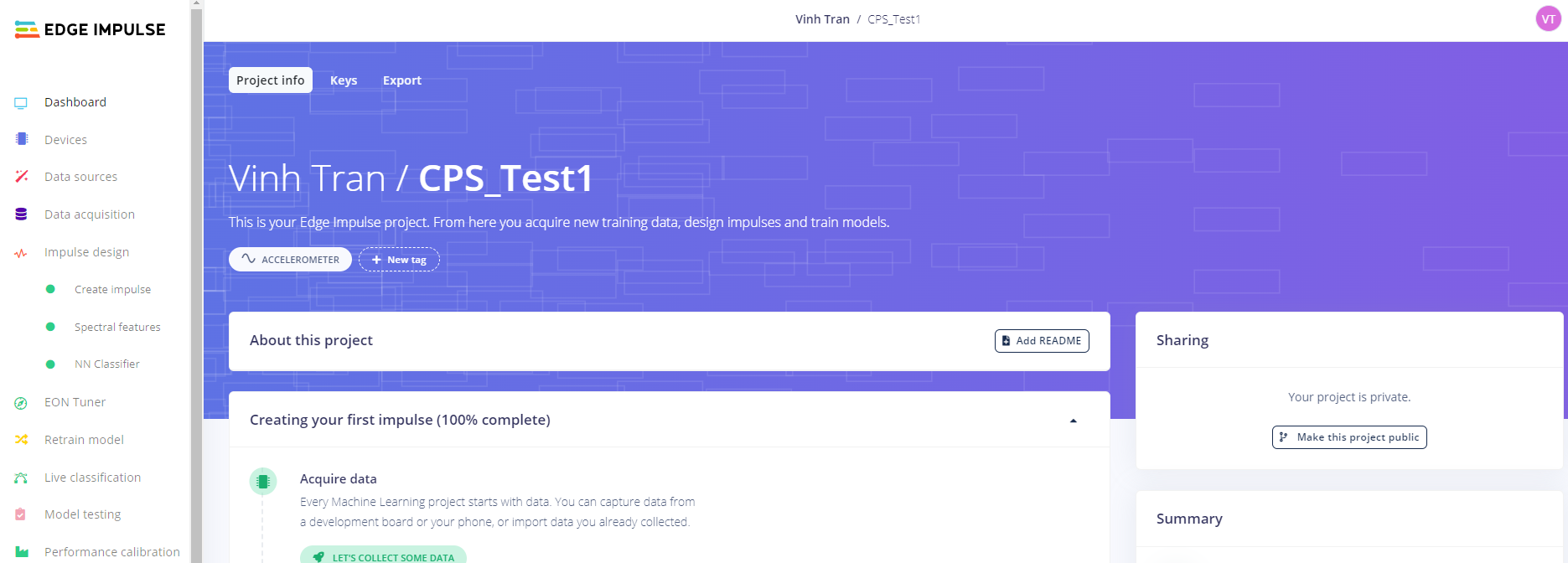


* + - * In the next pop up window, select “Accelerometer data”, which sets the project up to be used for motion classification.



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





## 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

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

* 1. First, you must download the streaming firmware to the Thingy52 and nrf52DK board.
     1. Download the two files (central\_uart.hex and datastream.hex) from:

<https://github.com/vinhtqc/streamThingy>

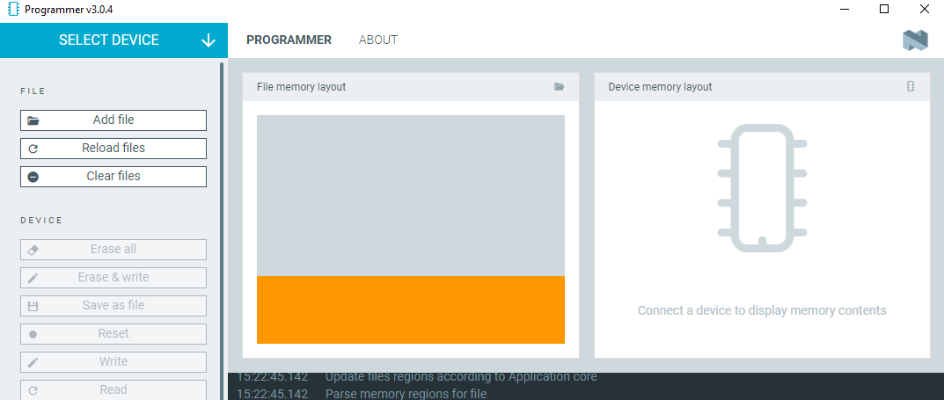
* + 1. Program nRF52DK:
       - Connect the nRF52DK board to PC, then turn on its Power switch.



* + - * Open “nRF Connect for Desktop” 🡪 then open “Programmer”

Text

Description automatically generated



* + - * Click “Select device” at the top left, then select nRF52DK
        1. **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
    1. Program Thingy52:
       - Connect Thingy52 to nRF52DK board using the JTAG cable, then connect nRF52DK board to PC.



* + - * Turn on Thingy52 and nRF52DK board



Power switch

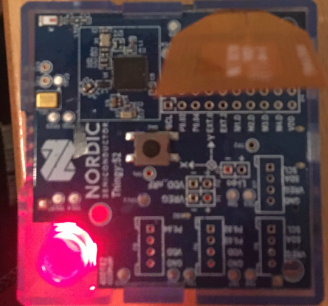
ON

OFF

* + - * 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

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

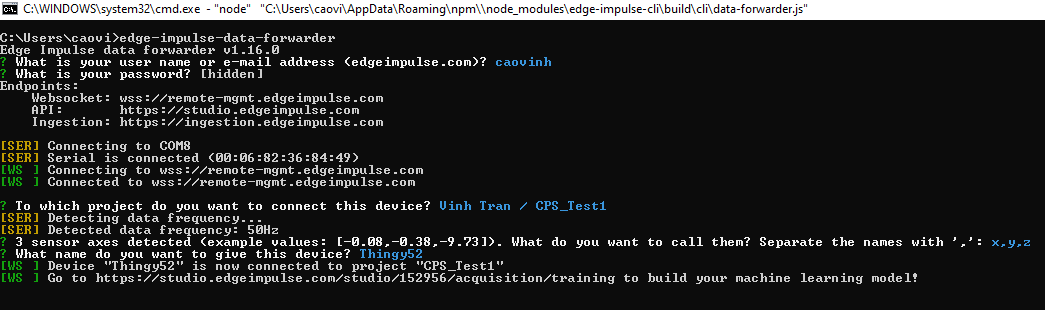
* + - * After finishing the programing of the two devices, *carefully* unplug the JTAG cable.
      * Hold the main button on the Thingy52 for 5 seconds 🡪 the LED should now be flashing quickly in Red: this means the Thingy52 is has connected and is transferring data to nRF52DK over Bluetooth.



* + - * 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.

### 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



Your account

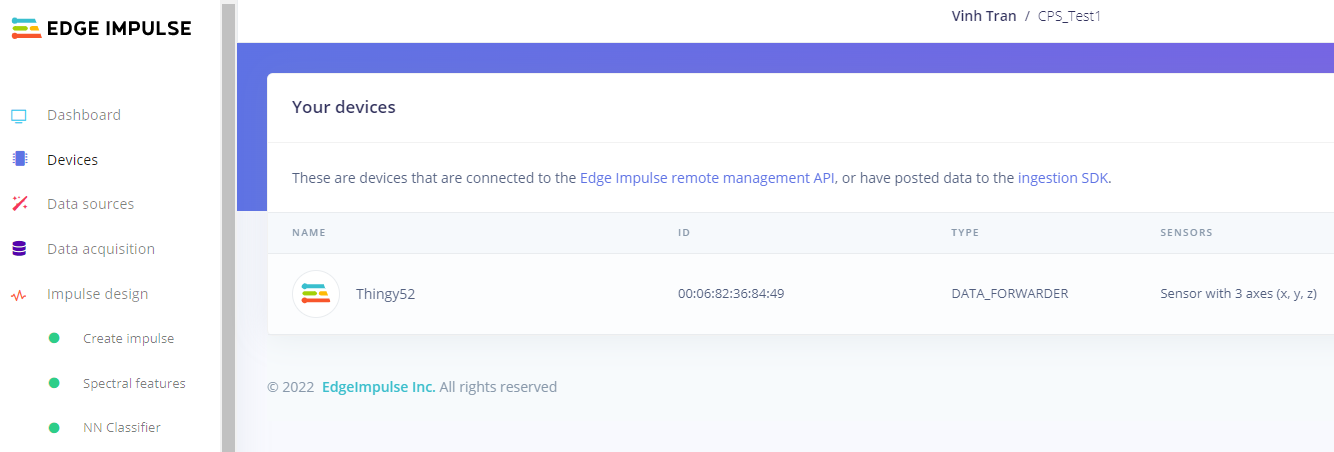
Password

Your project name

Sensor axes’ names

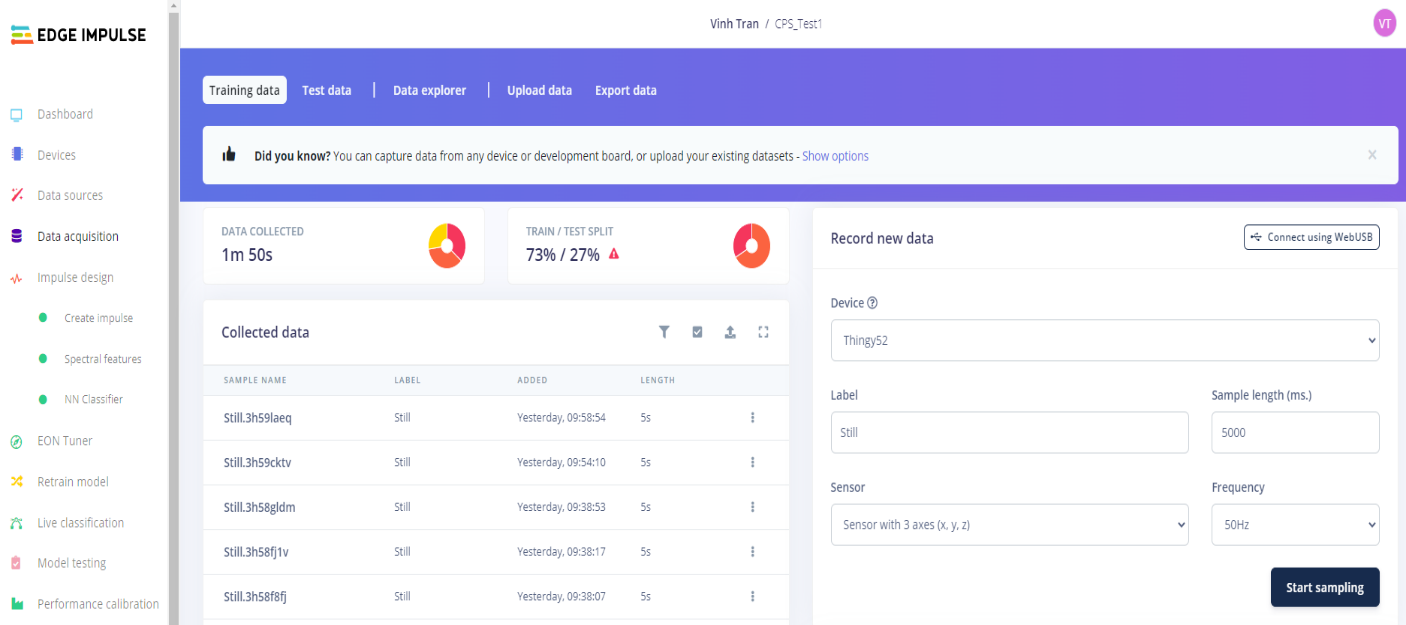
Device name

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



### Collect and label data on Edge Impulse platform

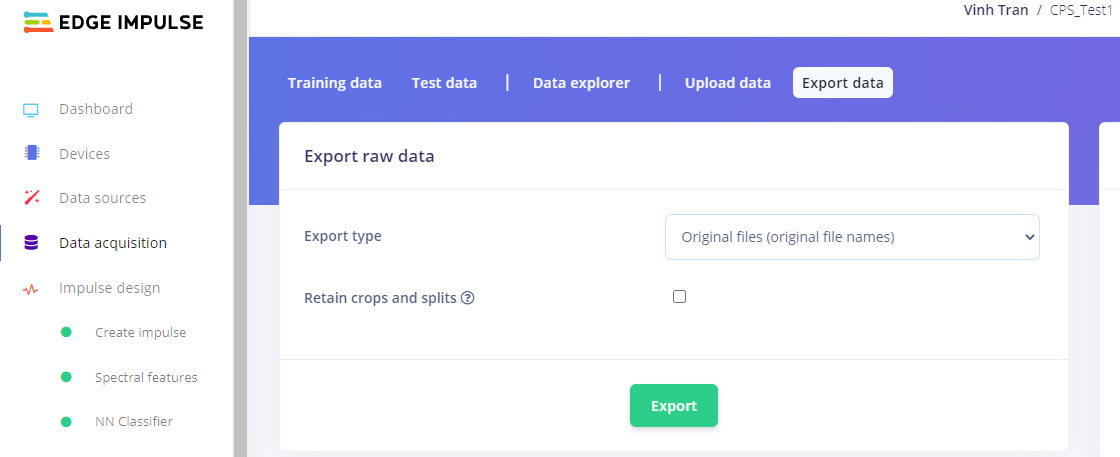
* + - * In your Edge Impulse project, click on “Data acquisition”.



* + - * 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)

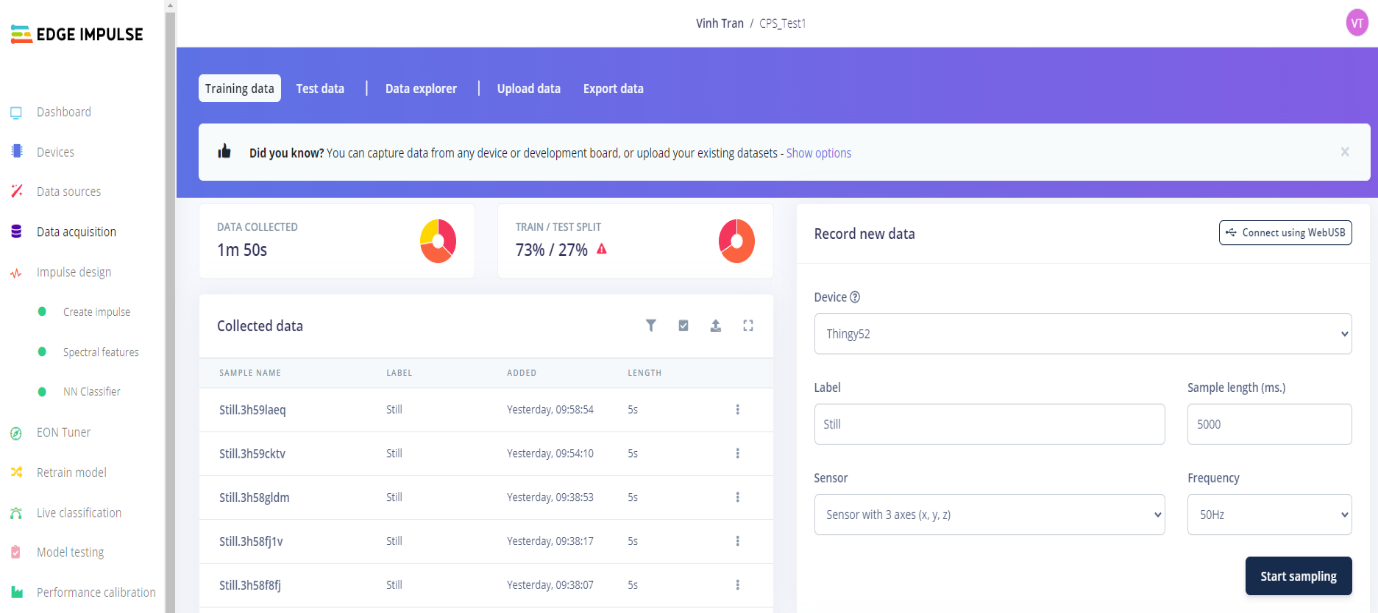
**Note: label should be less than 10 characters.**

* + - * 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:

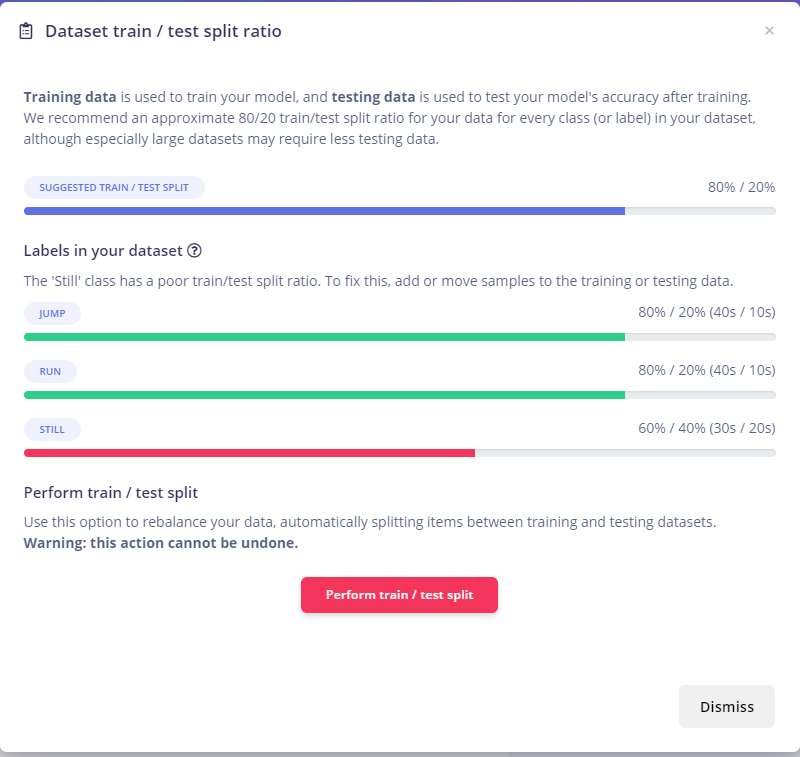


* + - * Separate the dataset for training and testing purposes: After collecting all data, you need to first separate data into a training and a testing set.

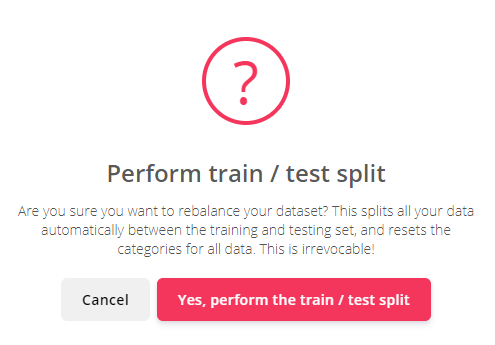
1. Click the alarm/danger icon behind TRAIN/TEST SPLIT (see screenshot) on the same page where you collected the data (Data acquisition page, Training data tab)



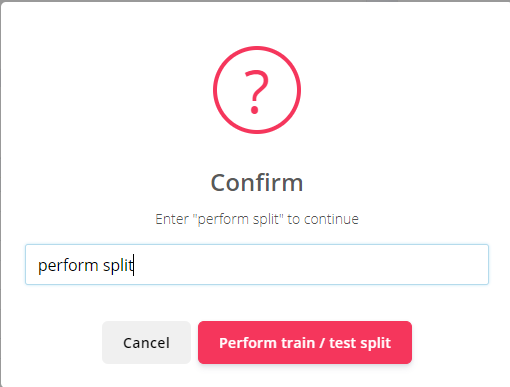
* + - * 1. A popup window appears, click on the “Perform train/test split” button (please note that the “Perform train/test split” button only appears after having collected enough data samples for each class; it appears that you need at least 10 samples per label)



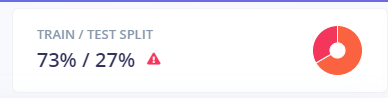
* + - * 1. Then confirm the split by clicking the “Yes, perform the train/test split” button in the next pop up window



* + - * 1. Then, type “perform split” in the textbox and press “Perform train/test split” to confirm the split again



* + - * 1. After this, you can see your data has been split



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.

## 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.

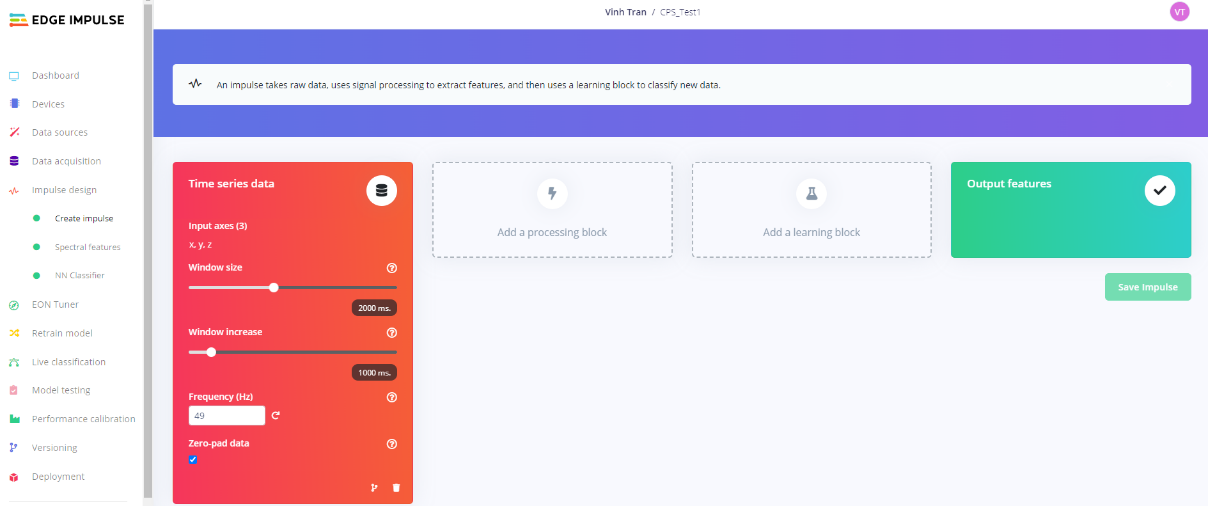
It includes the following subtopics:

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

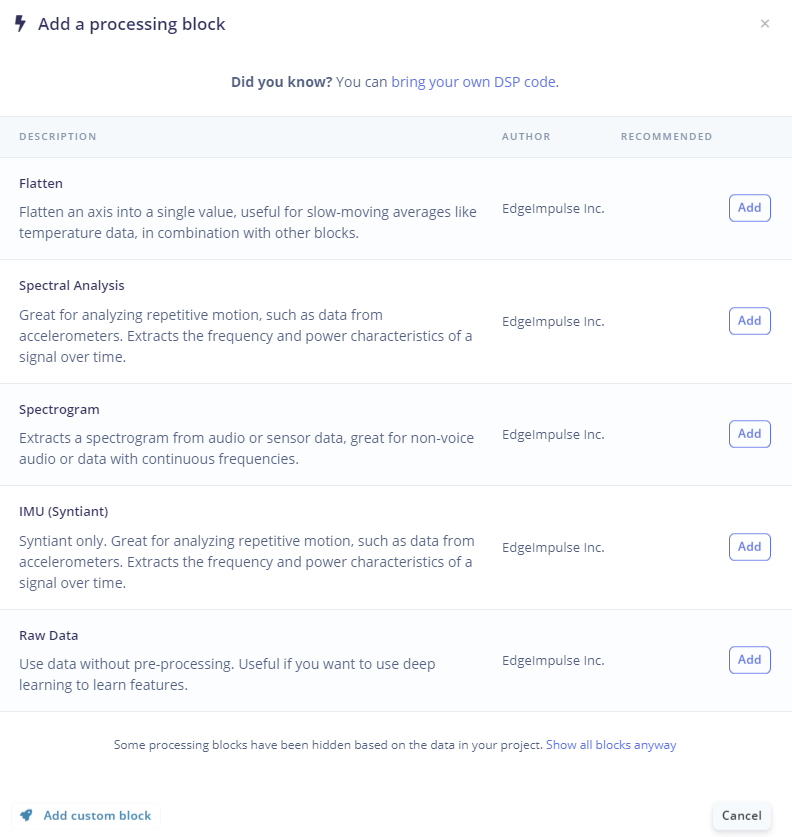
### Build model:

#### Create general blocks:

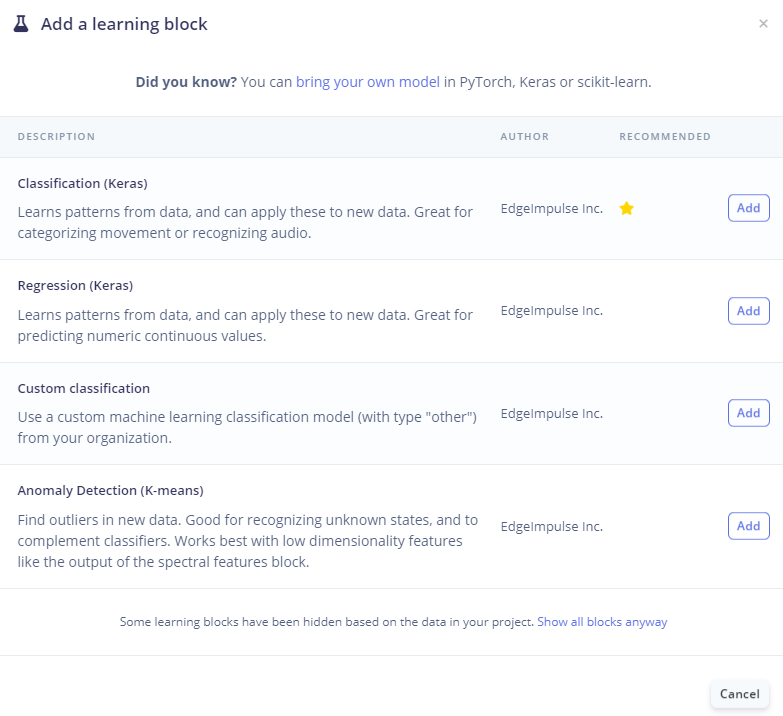
* Click on “Impulse design” 🡪 “Create Impulse”.
* Select the “window size” and “window increase”, e.g 2000 and 1000.
* Make sure the “frequency” input field corresponds to the frequency of the sampled data (collected in Section IV - Step 4)



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



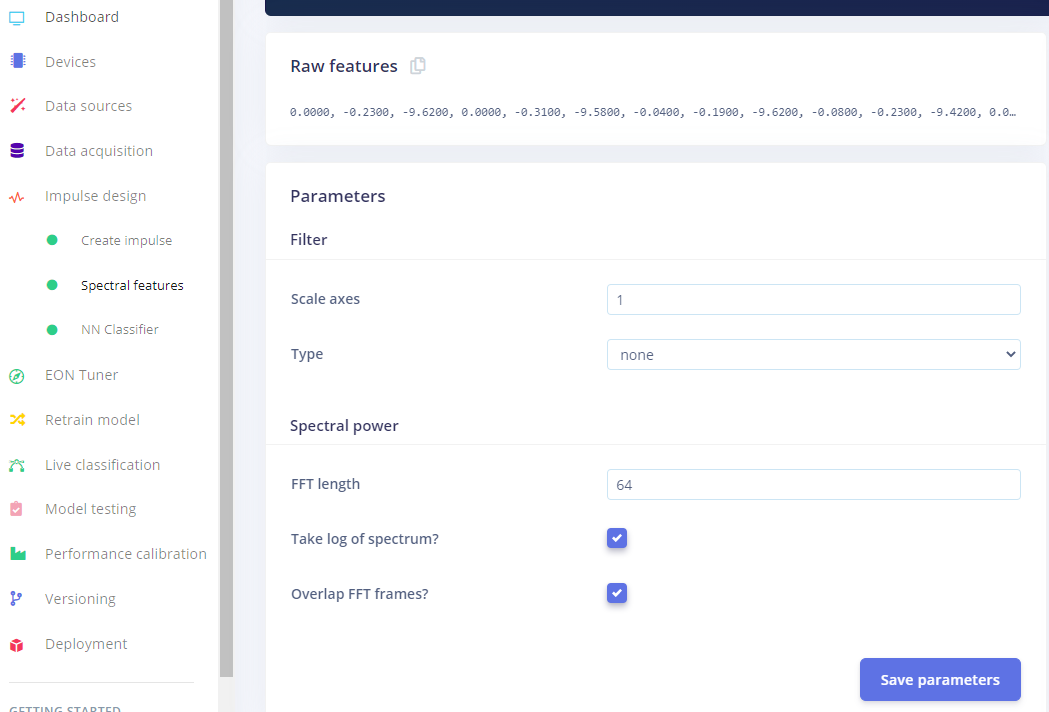
* Click on “add a learning block” and select “Classification” in the pop up window.



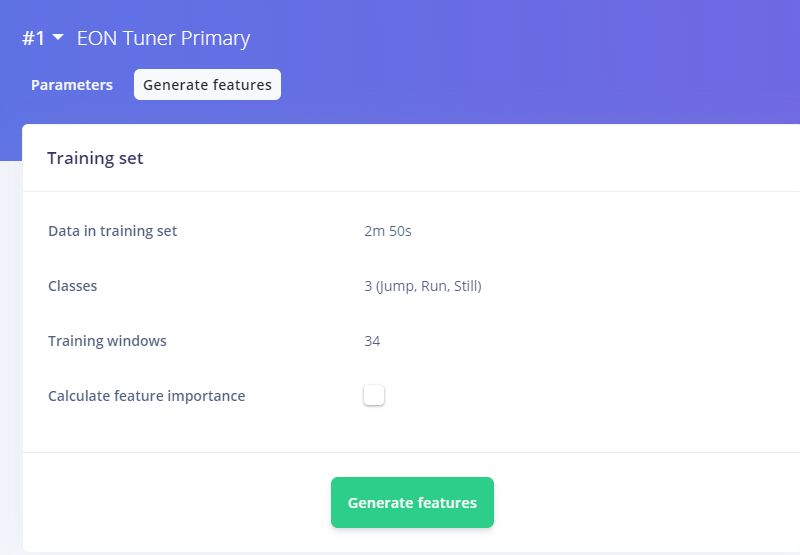
* Click on “Save Impulse”

#### Configure and Run feature extraction:

* + 1. 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).

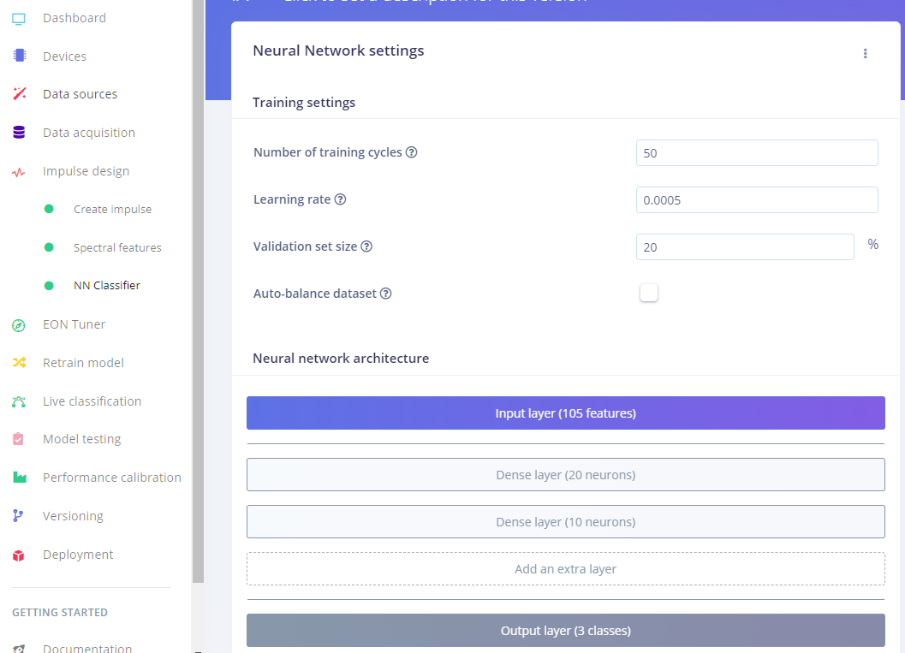


* + 1. After this, click “Generate Features”. This will run a feature generation job. You can proceed to the next instruction after the feature generation job has completed.

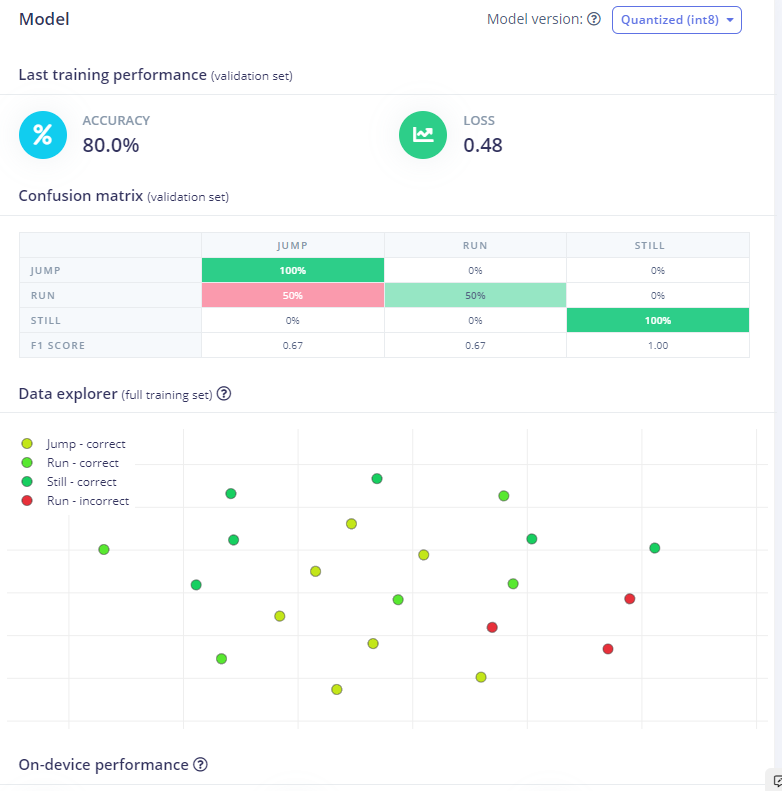


#### Build and run a neural network:

* + 1. 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.



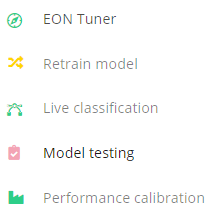
* + 1. After training, results will be shown on the right



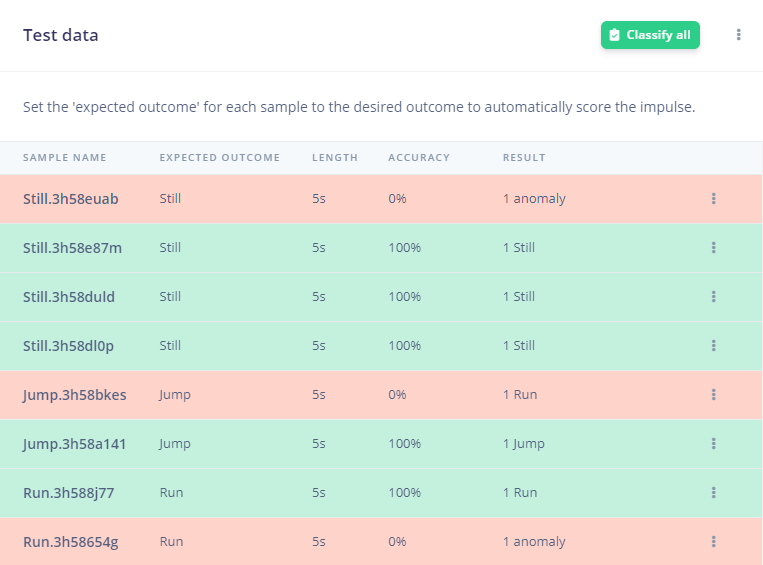
### Test the model:

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

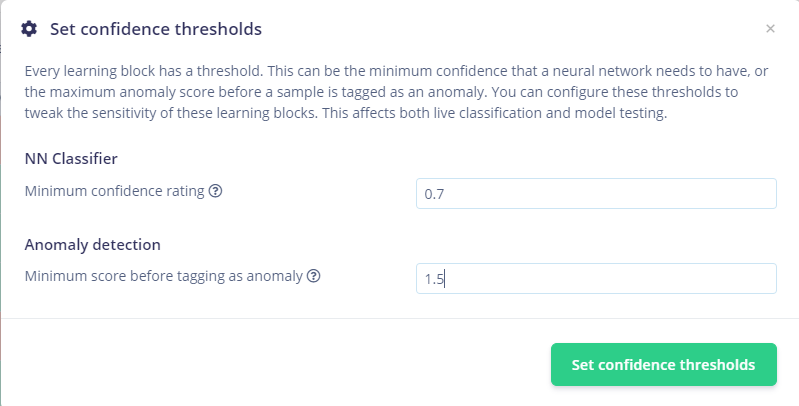
* Go to “Model testing” on the left panel



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

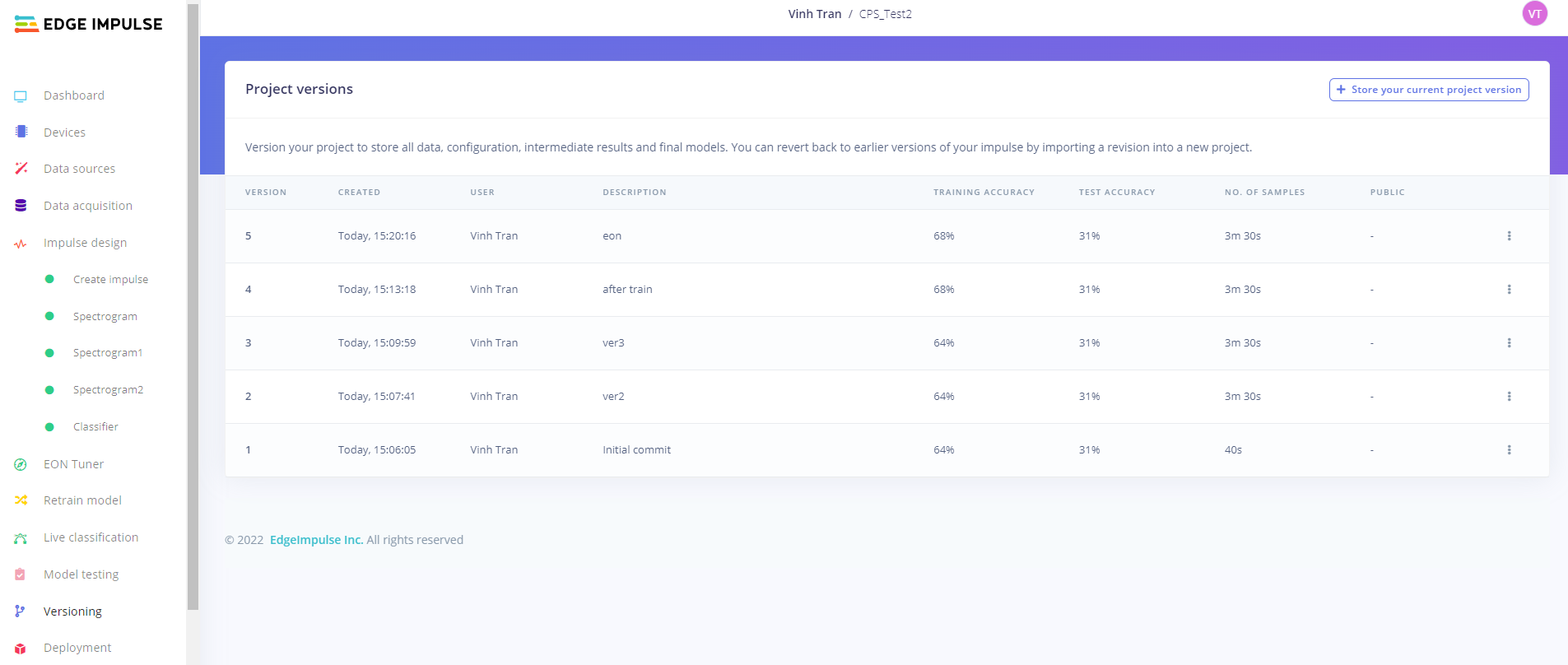


* Set the confidence threshold level of the NN Classifier (0..1). A class is predicted as TRUE only if its confidence value is higher than this threshold, unless the result is set to “Uncertain”.
* The “Anomaly detection” is only available if you have added an anomaly block to your Neural Network architecture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

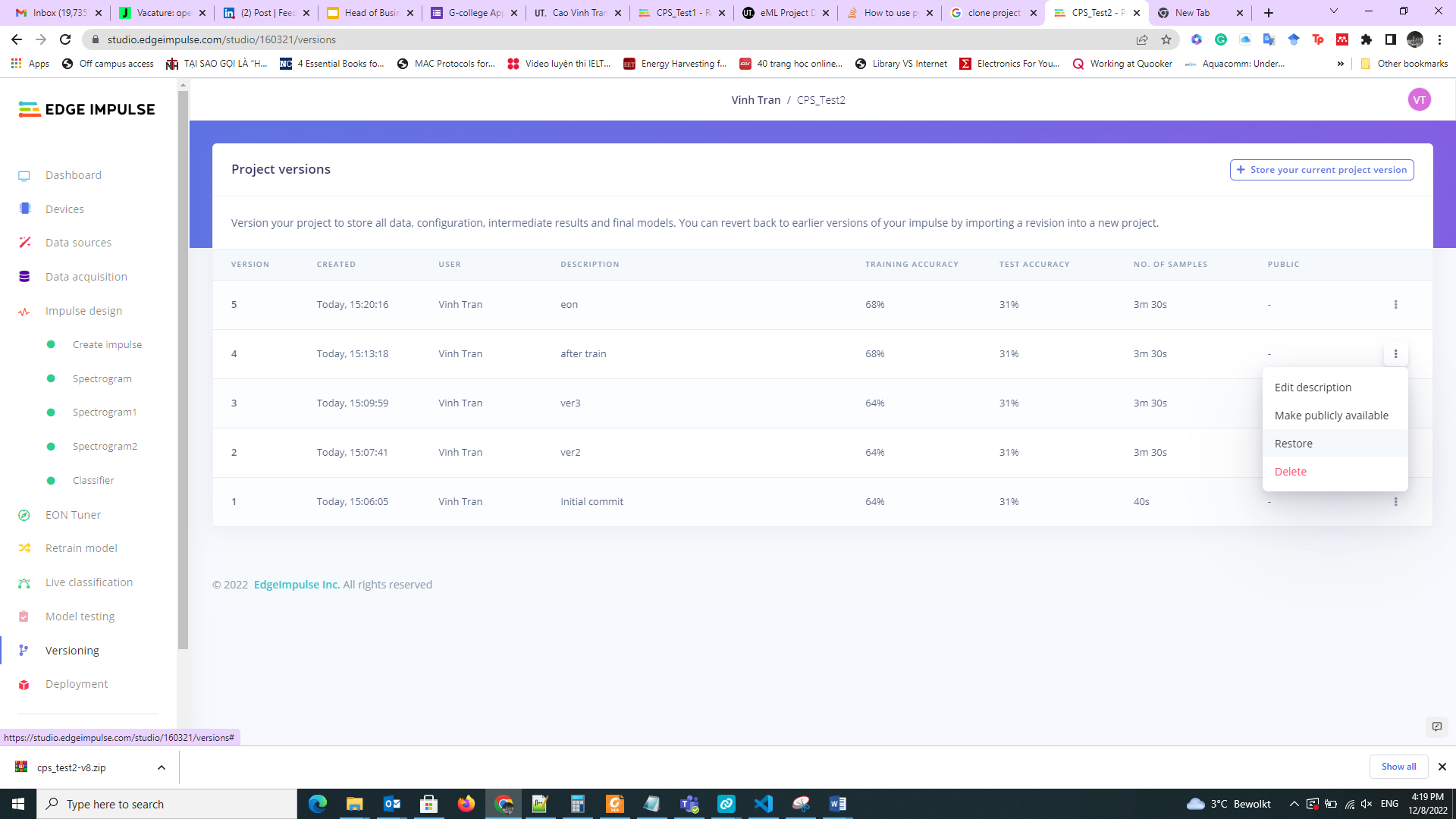


### Save the built model

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



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



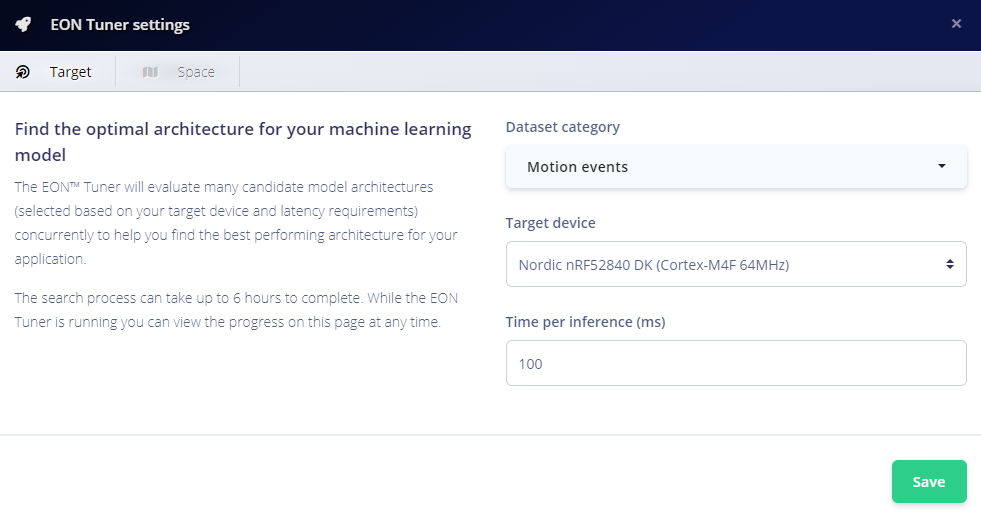
### Tune the model:

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.

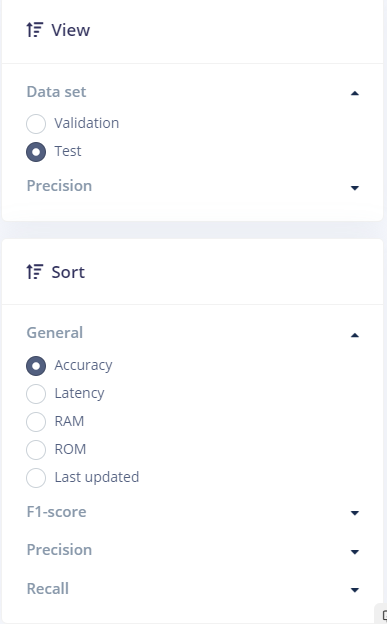
* On the left panel, select “EON Tuner” 🡪 “Start EON Tuner”
* A pop-up panel will appear, select the Nordic nRF52840 DK (Cortex-M4F 64MHz).



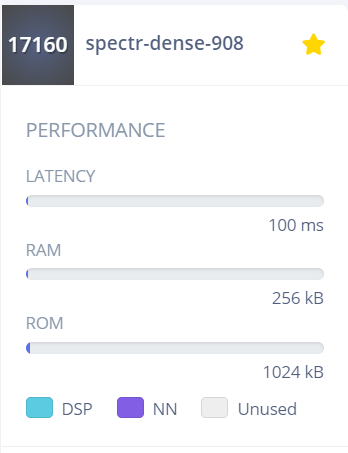
* Click on Save, after which you can click on “Start EON tuner”, after which the EON tuner will run. Note that it can take quite long for the EON Tuner to finish. The result after finishing could look something like the below:



* 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.



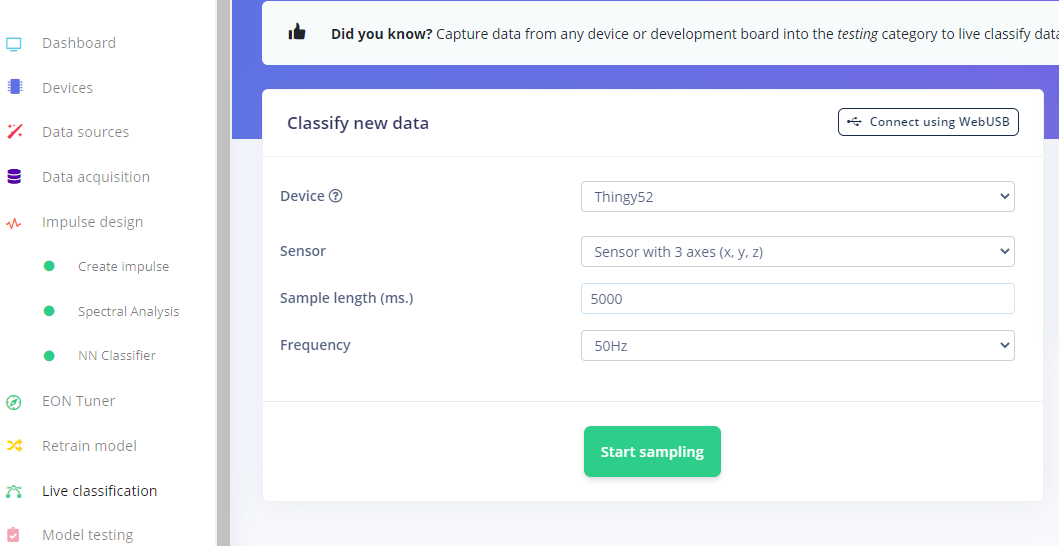
* Select what you think is the best model by clicking on “Select” on the model. Note: The Thingy52 has less resources available than the nRF52832DK. You must select a model that uses less than **128KB** ROM and less than **12KB** RAM (you can evaluate this by seeing how much of the resource usage bar 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)



**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.**

### 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”



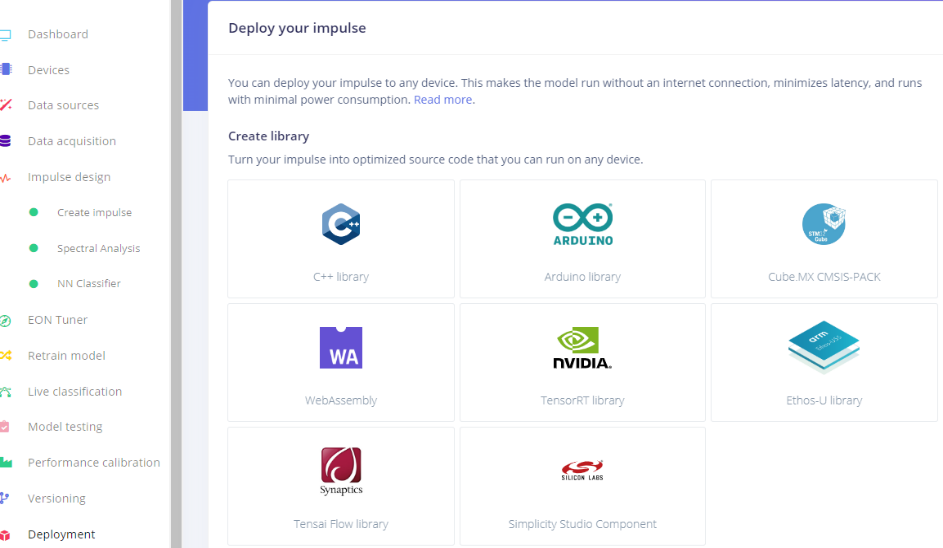
* After a sample was taken, a classification result is provided. For spectral analysis feature extraction, a 2D graph is shown in which the new datapoint is plotted together with the labeled datapoints (to provide a visual intuition with regard to the classification result)

Graphical user interface, application

Description automatically generated

### 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”.



* Wait for the building to finish, after which a popup window will appear prompting to save the file, e.g: “cps\_test1.zip”

### 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>

## 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

### 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:

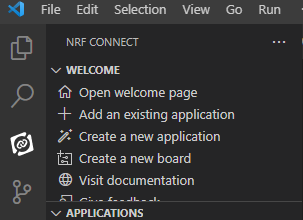
<https://github.com/vinhtqc/CPS2022_ML>

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

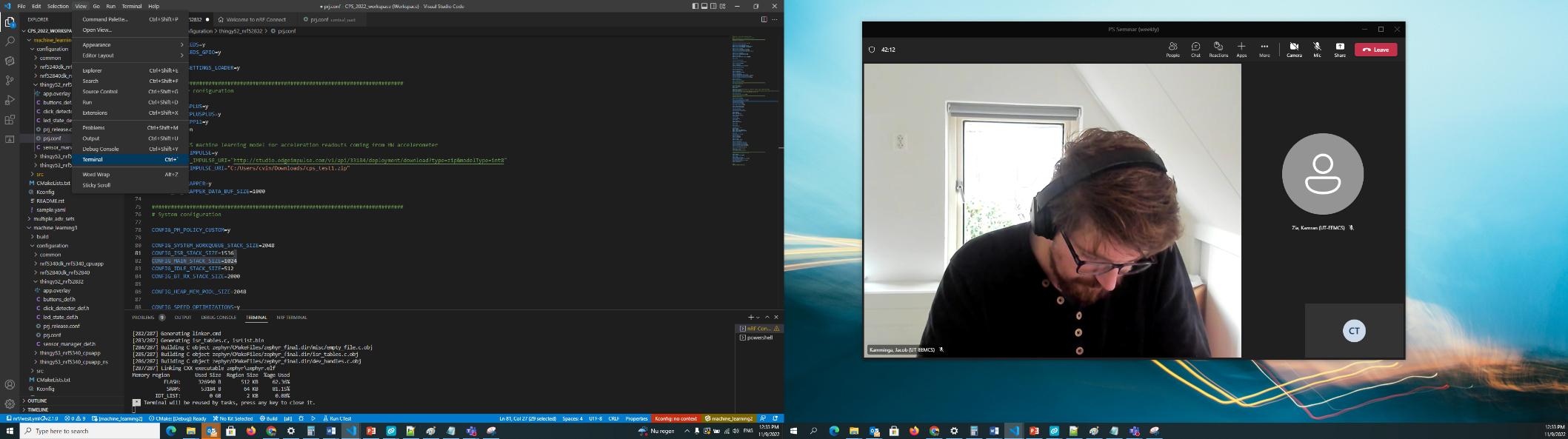
* Open “nRF Connect for Desktop” 🡪 Toochain manager 🡪 Click “Open VsCode” under the nrF SDK version downloaded in Section II to open Visual studio code



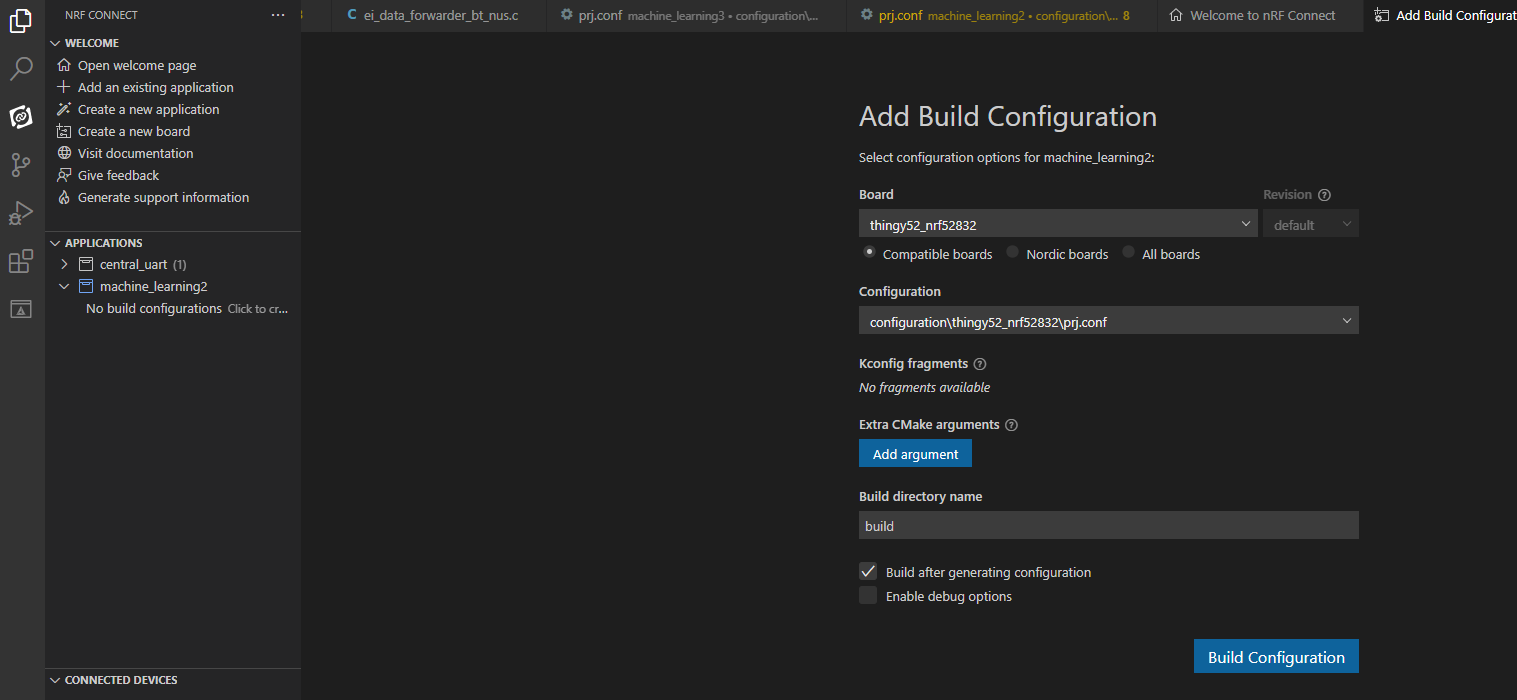
* In “VSCode”, on the left sidebar, click on the “nRF Connect” icon (or press Ctrl + Alt + N)
* Then, click “Add an existing application”



* 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.

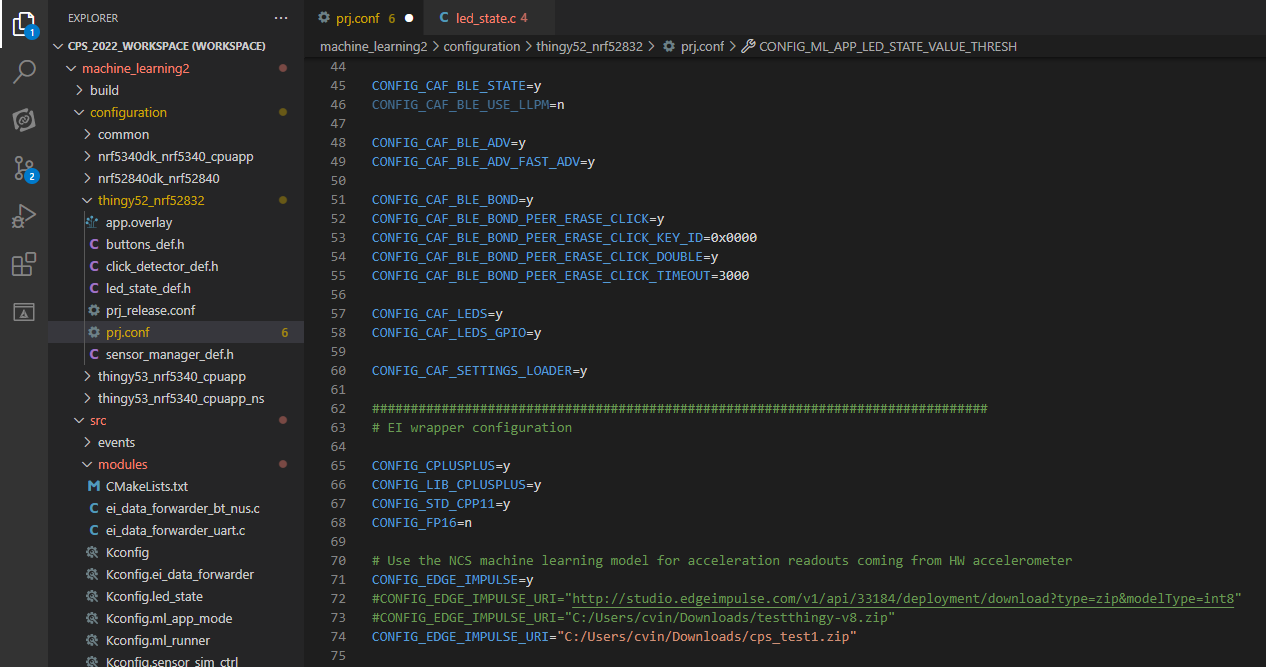


* In the “Applications” bar on the left side, select “machine\_learning2”, click on “No build configurations”, a configuration window will appear as the below picture on the right. **Note**, if you receive an error message stating: “Couldn’t resolve Zephyr base from …”, please make sure you have installed 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)



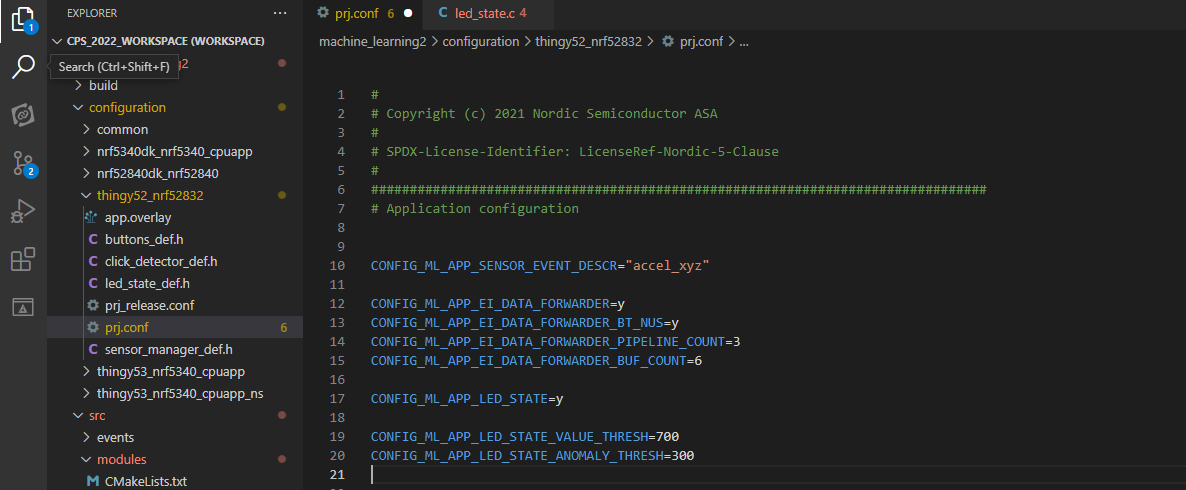
### Configuration for integrating the built machine learning model

* + 1. Select the **“Explorer”** icon in the left column, then select **“machine\_learning2” 🡪 “configuration” 🡪 “thingy52\_nrf52832”**
       1. 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 (‘/’).

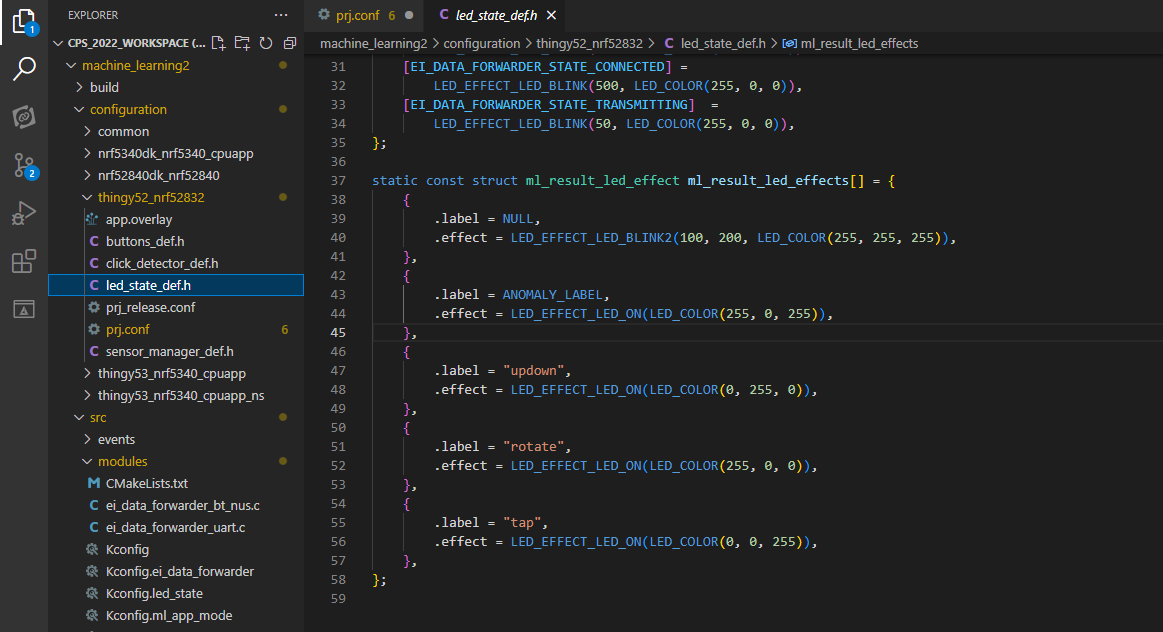


* Scroll through the right panel to the line with the text: **CONFIG\_ML\_APP\_LED\_STATE\_VALUE\_THRESH** and **CONFIG\_ML\_APP\_LED\_STATE\_ANOMALY\_THRESH**

Set these two values equal to the two values you defined in Section V – Part 2 ([Test the model](#_Test_the_model:)), and multiply this number by 1000.



* + - 1. 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](#_Collect_and_label).) 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.

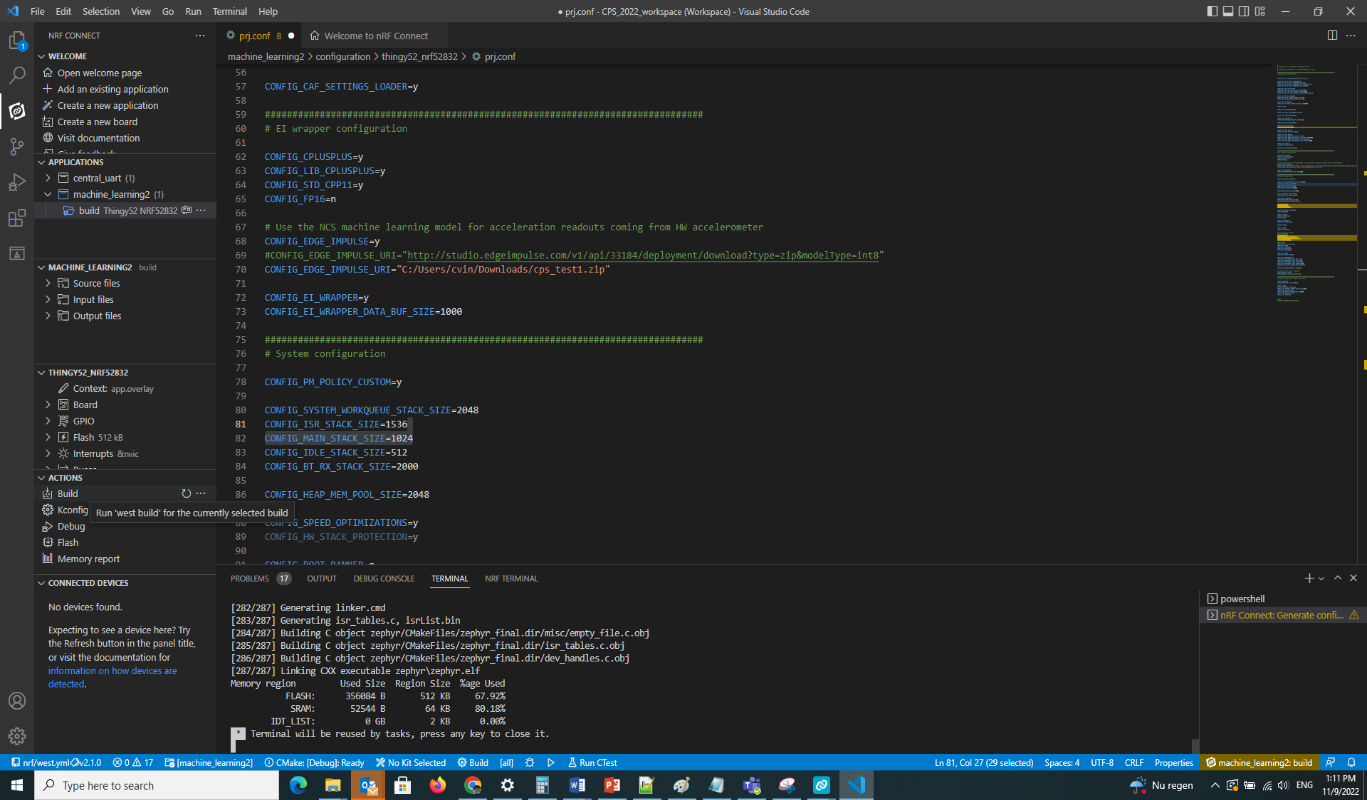


class name: e.g: Run

Color (R,G,B)

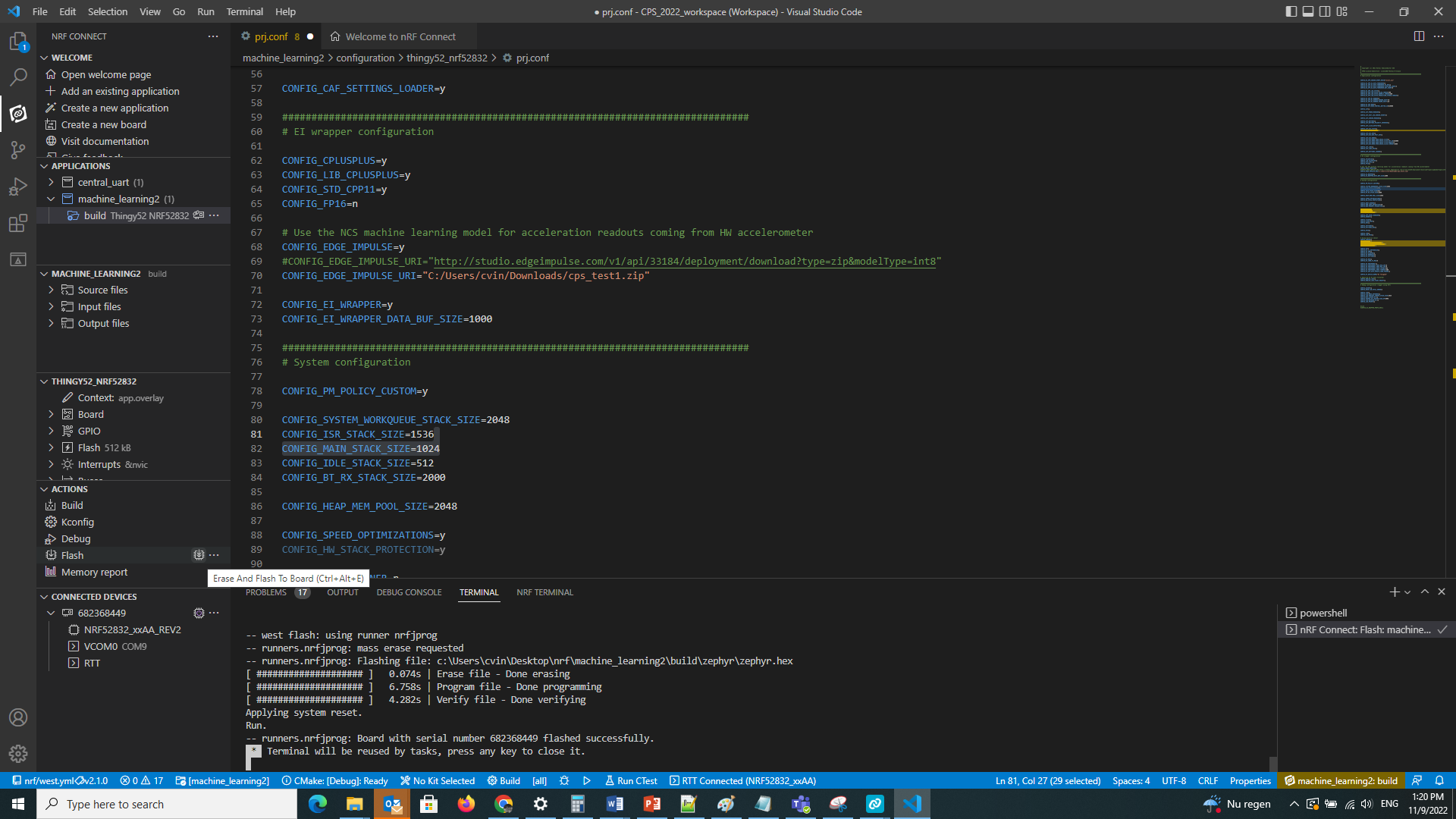
### Compile firmware

* 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



### Flash firmware to Thingy52

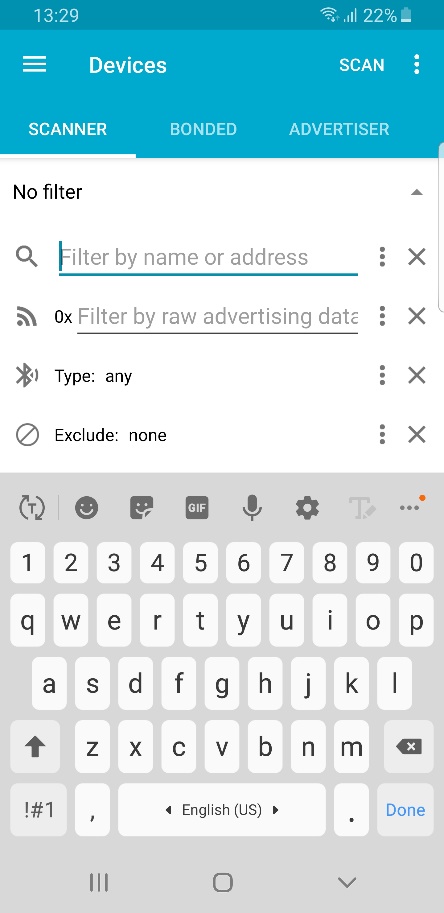
Click the “Erase and Flash” icon (small downward pointing arrow, next to “Flash”) to flash the program to the Thingy52. Wait until the Terminal reports the flashing process has finished. After this, disconnect the Thingy52.



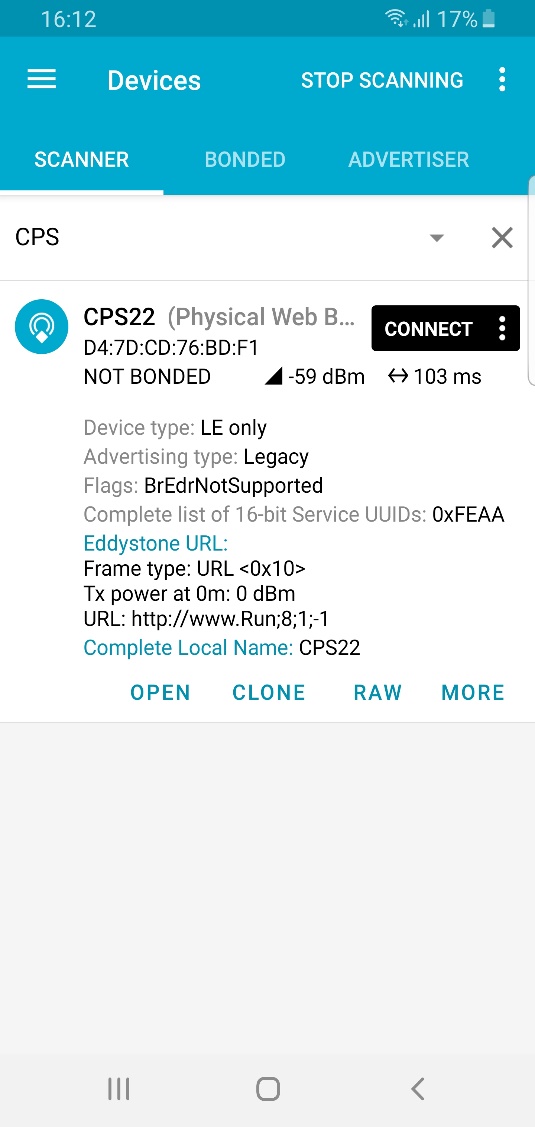
## Demonstration:

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 is Online

Classification result

## Updated history:

* + - * Manual\_CPS\_5: add trouble shooting for installing NodeJs tools in section II.2.d