

## **Edge Computing Lab**

**Class: TY-AIEC**

**School of Computing, MIT Art Design Technology University**

*Academic Year: 2024-25*

### **Experiment No. 8**

**Name: Rahul Bhati**

**Roll no: 2223416**

#### **Introduction**

**The "magic wand" project that can recognize gestures using an accelerometer and an ML classification model on Edge Devices**

**Objective:** Build a project to detect the accelerometer values and convert them into gestures

#### **Tasks:**

- Generate the dataset for Accelerometer Motion (Up-Down, Left-Right)
- Configure BLE Sense / Mobile for Edge Impulse
- Building and Training a Model
- Deploy on Nano BLE Sense / Mobile Phone

#### **Introduction**

Edge Impulse is a development platform for machine learning on edge devices, targeted at developers who want to create intelligent device solutions. The " Accelerometer Motion "sensor reading equivalent in Edge Impulse would typically involve creating a simple machine learning model that can run on an edge device, like classifying sensor data or recognizing a basic pattern.

#### **Materials Required**

- Nano BLE Sense Board

#### **Theory**

GPIO (General Purpose Input/Output) pins on the Raspberry Pi are used for interfacing with other electronic components. BCM numbering refers to the pin numbers in the Broadcom SOC channel, which is a more consistent way to refer to the GPIO pins across different versions of the

Here's a high-level overview of steps you'd follow to create a "Hello World" project on Edge Impulse:

#### **Steps to Configure the Edge Impulse:**

1. Create an Account and New Project:
  - Sign up for an Edge Impulse account.
  - Create a new project from the dashboard.

## 2. Connect a Device:

- You can use a supported development board or your smartphone as a sensor device.
- Follow the instructions to connect your device to your Edge Impulse project.

## 3. Collect Data:

- Use the Edge Impulse mobile app or the Web interface to collect data from the onboard sensors.
- For a "Hello World" project, you could collect accelerometer data, for instance.

## 4. Create an Impulse:

- Go to the 'Create impulse' page.
- Add a processing block (e.g., time-series data) and a learning block (e.g., classification).
- Save the impulse, which defines the machine learning pipeline.

## 5. Design a Neural Network:

- Navigate to the 'NN Classifier' under the 'Learning blocks'.
- Design a simple neural network. Edge Impulse provides a default architecture that works well for most basic tasks.

## 6. Train the Model:

- Click on the 'Start training' button to train your machine learning model with the collected data.

## 7. Test the Model:

- Once the model is trained, you can test its performance with new data in the 'Model Testing' tab.

## 8. Deploy the Model:

- Go to the 'Deployment' tab.
- Select the deployment method that suits your edge device (e.g., Arduino library, WebAssembly, container, etc.).

- Follow the instructions to deploy the model to your device.

## 9. Run Inference:

- With the model deployed, run inference on the edge device to see it classifying data in real-time.

## 10. Monitor:

- You can monitor the performance of your device through the Edge Impulse studio.

Paste your Edge Impulse project's Results:

The image displays two screenshots of the Edge Impulse Studio web interface. Both screenshots show a project named 'Rahul Bhati / R\_bhati-project-1' with a target device of 'Arduino Nano 33...'. The interface includes a left sidebar with navigation options like Dashboard, Devices, Data acquisition, Experiments, EON Tuner, and Impulse design. The main area shows the 'Dataset' tab with a table of collected samples and a 'Collect data' button. The 'TRAIN / TEST SPLIT' section indicates 76% training and 24% testing data.

**Top Screenshot Dataset Table:**

SAMPLE NAME	LABEL	ADDED	LENGTH
circular.5n2tbs8n	circular	Mar 25 2023, 14:34:20	4s
circular.5n2tbsdr	circular	Mar 25 2023, 14:34:05	4s
circular.5n2taq70	circular	Mar 25 2023, 14:33:45	4s
circular.5n2tack1	circular	Mar 25 2023, 14:33:32	4s
circular.5n2t9uq	circular	Mar 25 2023, 14:33:17	4s
circular.5n2t9hr	circular	Mar 25 2023, 14:33:03	4s
circular.5n2t8bb	circular	Mar 25 2023, 14:32:32	4s
circular.5n2t84es	circular	Mar 25 2023, 14:32:18	4s
circular.5n2t5skt	circular	Mar 25 2023, 14:31:04	4s
circular.5n2t5dah	circular	Mar 25 2023, 14:30:48	4s
circular.5n2t4uld	circular	Mar 25 2023, 14:30:33	4s

**Bottom Screenshot Dataset Table:**

SAMPLE NAME	LABEL	ADDED	LENGTH
left and right.5n2u5r51	left and right	Mar 25 2023, 14:48:31	4s
left and right.5n2u5dni	left and right	Mar 25 2023, 14:48:17	4s
left and right.5n2u5030	left and right	Mar 25 2023, 14:48:03	4s
left and right.5n2u4f0a	left and right	Mar 25 2023, 14:47:51	4s
left and right.5n2u451p	left and right	Mar 25 2023, 14:47:36	4s
left and right.5n2u3l0u	left and right	Mar 25 2023, 14:47:19	4s
left and right.5n2u2cao	left and right	Mar 25 2023, 14:46:38	4s
left and right.5n2tvq0e	left and right	Mar 25 2023, 14:45:13	4s
up and down.5n2tq7p7	up and down	Mar 25 2023, 14:42:11	4s
up and down.5n2tplei	up and down	Mar 25 2023, 14:41:52	4s
up and down.5n2tp742	up and down	Mar 25 2023, 14:41:37	4s

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Spectral features

Classifier

Retrain model

Live classification

Model testing

Deployment

Versioning

Upgrade Plan

Get access to higher job limits and more collaborators

View plans

Rahul Bhati / R\_bhati-project-1

PERSONAL

Target: Arduino Nano 33

Dataset

Data explorer

Data sources

Synthetic data

AI labeling

CSV Wizard

DATA COLLECTED

5m 37s

TRAIN / TEST SPLIT

76% / 24%

Collect data

Connect a device to start building your dataset.

Dataset

Training 40

Test 0

SAMPLE NAME	LABEL	ADDED	LENGTH
idle.5n2uqgtv	idle	Mar 25 2025, 14:59:49	4s
idle.5n2uq2pa	idle	Mar 25 2025, 14:59:34	4s
idle.5n2upbhj	idle	Mar 25 2025, 14:59:17	4s
idle.5n2up2mp	idle	Mar 25 2025, 14:59:01	4s
idle.5n2uokpn	idle	Mar 25 2025, 14:58:47	4s
idle.5n2unmvm	idle	Mar 25 2025, 14:58:18	4s
idle.5n2umut6	idle	Mar 25 2025, 14:57:51	4s
idle.5n2ugf8g	idle	Mar 25 2025, 14:54:20	4s
idle.5n2uf583	idle	Mar 25 2025, 14:53:36	4s
idle.5n2uenou	idle	Mar 25 2025, 14:53:23	4s
idle.5n2ue9ms	idle	Mar 25 2025, 14:53:08	4s

RAW DATA

Click on a sample to load...

Resume tutorial

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Spectral features

Classifier

Retrain model

Live classification

Model testing

Deployment

Versioning

Upgrade Plan

Get access to higher job limits and more collaborators

View plans

Rahul Bhati / R\_bhati-project-1

PERSONAL

Target: Arduino Nano 33

Impulse #1

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 (9)

accX, accY, accZ, gyx, gyY, gyZ, magX, magY, magZ

Window size

2,000 ms

Window increase (stride)

200 ms

Frequency (Hz)

100

Zero-pad data

☒

Spectral Analysis

Name

General Impulse

Input axes (3)

☒ accX

☒ accY

☒ accZ

☐ gyx

☐ gyY

☐ gyZ

☐ magX

☐ magY

☐ magZ

Classification

Name

Classifier

Input features

☒ Spectral features

Output features

4 (circular, idle, left and right, up and down)

Add a learning block

Output features

4 (circular, idle, left and right, up and down)

Save Impulse

Resume tutorial

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Spectral features

Classifier

Retrain model

Live classification

Model testing

Deployment

Versioning

Upgrade Plan

Get access to higher job limits and more collaborators

View plans

Rahul Bhati / R\_bhati-project-1

PERSONAL

Target: Arduino Nano 33

Parameters

Generate features

Raw data

Show: All labels

idle.5n2uqgtv (idle)

0ms 210ms 420ms 630ms 840ms 1050ms 1260ms 1470ms 1680ms 1890ms 2100ms 2310ms 2520ms 2730ms 2940ms 3150ms 3360ms 3570ms 3780ms 3990ms

Raw features

0.3621, 0.1115, 9.9196, 0.4762, 0.4318, 10.8688, 0.2927, 0.2885, 10.8634, 0.3789, 0.8738, 10.1381, ...

Label

idle

Parameters

Autotune parameters

Filter

Scale axes

1

Input decimation ratio

1

Type

none

Analysis

Type

FFT

FFT length (N)

4096

DSP result

After filter

0.0 0.5 1.0

Value

0.00 210.00 420.00 630.00 840.00 1050.00 1260.00 1470.00 1680.00 1890.00

Sample #

Spectral power (log)

0 -0.5 -1.0 -1.5 -2.0

Energy

0.00 210.00 420.00 630.00 840.00 1050.00 1260.00 1470.00 1680.00 1890.00

Sample #

Resume tutorial

EDGE IMPULSE

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

Create impulse

Spectral features

Classifier

Retrain model

Live classification

Model testing

Deployment

Upgrade Plan

Get access to higher job limits and more collaborators.

View plans

Rahul Bhati / RJ\_bhati-project-1

Target: Arduino Nano 33 ...

Neural Network settings

Training settings

Number of training cycles

30

Use learned optimizer

☐

Learning rate

0.0005

Training processor

CPU

Advanced training settings

Neural network architecture

Input layer (39 features)

Dense layer (20 neurons)

Dense layer (10 neurons)

Add an extra layer

Output layer (4 classes)

Save & train

Training output

Model

Model version: Quantized (int8)

Last training performance (validation set)

ACCURACY

100.0%

LOSS

0.04

Confusion matrix (validation set)

	CIRCULAR	IDLE	LEFT AND RIGHT	UP AND DOWN
CIRCULAR	100%	0%	0%	0%
IDLE	0%	100%	0%	0%
LEFT AND RIGHT	0%	0%	100%	0%
UP AND DOWN	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00

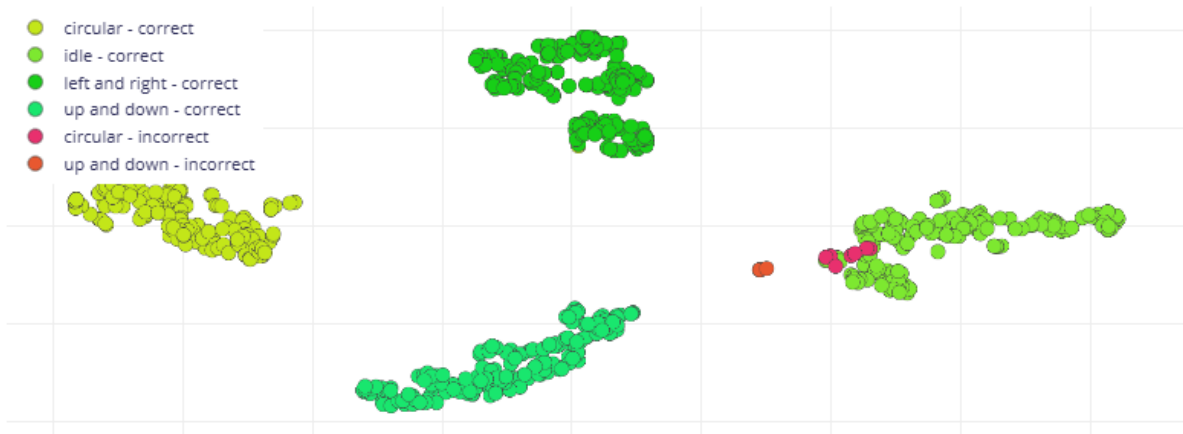
Metrics (validation set)

METRIC	VALUE
Area under ROC Curve	1.00
Weighted average Precision	1.00
Weighted average Recall	1.00
Weighted average F1 score	1.00

Data explorer (full training set)

Resume tutorial

## Data explorer (full training set) ?



## On-device performance ?

Engine: ? [EON™ Compiler](#)

INFERENCING TIME  
1 ms.

PEAK RAM USAGE  
1.4K

FLASH USAGE  
16.1K

```
nano_ble33_sense_accelerometer_continuous | Arduino IDE 2.3.5
File Edit Sketch Tools Help
Arduino Nano 33 BLE
nano_ble33_sense_accelerometer_continuous.ino
17 /* Includes ----- */
18 #include <Rj_bhati-project-1_inferencing.h>
19 #include <Arduino_LSM9DS1.h> //Click here to get the library: https://www.arduino.cc/reference/en/libraries/arduino\_lsm9ds1/
20
21 /* Constant defines ----- */
22 #define CONVERT_G_TO_MS2 9.80665f
23 /**
24  * When data is collected by the Edge Impulse Arduino Nano 33 BLE Sense
25  * firmware, it is limited to a 2g range. If the model was created with a
26  * different sample range, modify this constant to match the input values.
27  * See https://github.com/edgeimpulse/firmware-arduino-nano-33-ble-sense/blob/master/src/sensors/ei\_lsm9ds1.cpp
28  * for more information.
29  */
30 #define MAX_ACCEPTED_RANGE 2.0f
31
32 /*
33  ** NOTE: If you run into TFLite arena allocation issue.
34  **
35  ** This may be due to may dynamic memory fragmentation.
36  ** Try defining "-DEI_CLASSIFIER_ALLOCATION_STATIC" in boards.local.txt (create
37  ** if it doesn't exist) and copy this file to
38  ** <ARDUINO_CORE_INSTALL_PATH>/arduino/hardware/<mbed_core>/<core_version>/'.
39  **
40  ** See
41  ** (https://support.arduino.cc/hc/en-us/articles/360012076960-Where-are-the-installed-cores-located-)
42  ** to find where Arduino installs cores on your machine.
43  **
44  ** If the problem persists then there's not enough memory for this model and application.
45  */
46
47 /* Private variables ----- */
48 static bool debug_nn = false; // Set this to true to see e.g. features generated from the raw signal
49 static uint32_t run_inference_every_ms = 200;
50 static rtos::Thread inference_thread(osPriorityLow);
51 static float buffer[EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE] = { 0 };
```

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 132.291000 ms., Classification: 0.580000 ms., Anomaly: 0ms.):
#Classification results:
  circular: 0.371094
  idle: 0.523437
  right_left: 0.042969
  up_down: 0.062500
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 133.824997 ms., Classification: 0.571000 ms., Anomaly: 0ms.):
#Classification results:
  circular: 0.000000
  idle: 0.996094
  right_left: 0.000000
  up_down: 0.000000
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 129.904007 ms., Classification: 0.571000 ms., Anomaly: 0ms.):
#Classification results:
  circular: 0.000000
  idle: 0.996094
  right_left: 0.000000
  up_down: 0.003906
```