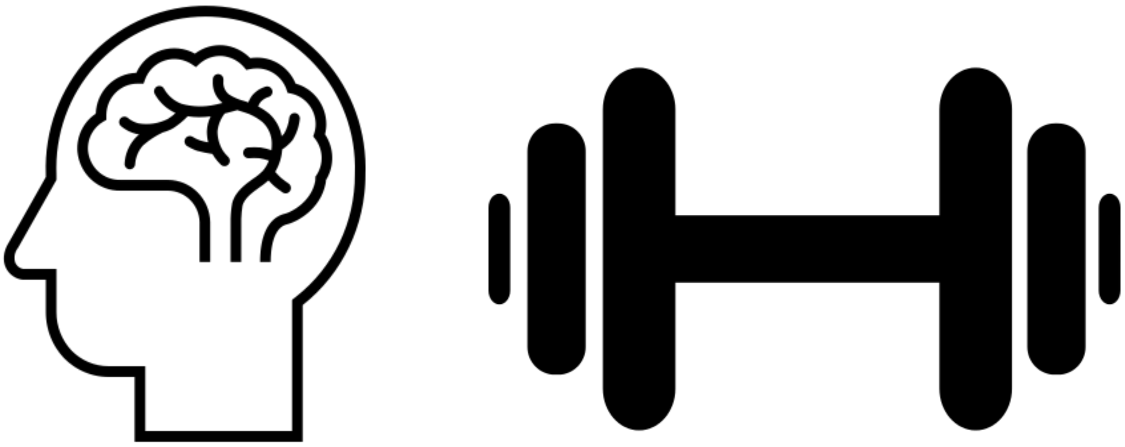
DECEMBER 29, 2022

EE-425: EMBEDDED SYSTEMS

COMPLEX ENGINEERING PROBLEM (CEP)



Smart Dumbbell for Real-time Exercise Classification

Group Members:
Atta ul Haleem (2019093)
Mustafa Anique Ansari (2019415)

Instructor:
Dr. Muhammad Irfan

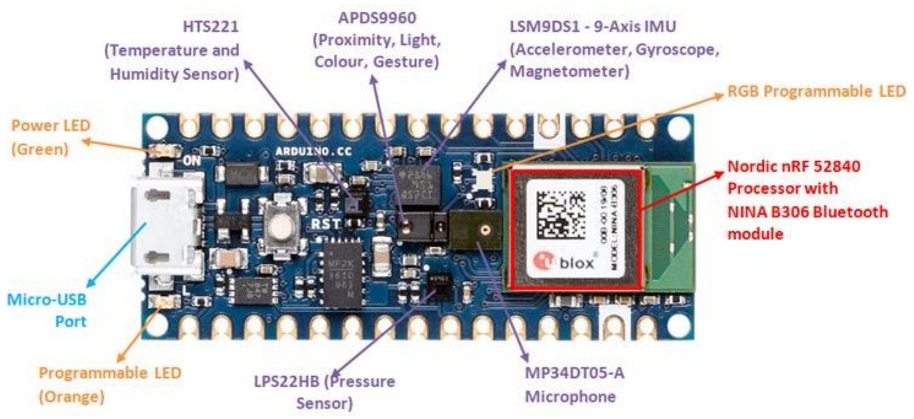
Introduction

The dumbbell is a type of free weight which is used in weight training. In this project, we propose attaching a small, embedded device to a dumbbell to make it a smart dumbbell or Smartbell. The Smartbell can identify certain common exercises using Tiny Machine Learning (TinyML). TinyML is a field which shrinks the performative power of Machine Learning to fit on tiny hardware. We use the Arduino Nano 33 BLE Sense as the edge device with Edge Impulse being used for data collection, impulse design, model training, testing and deployment. Moreover, we utilize the BLE feature of the Arduino for wireless communication of real-time inferences which are then displayed onto a Web Bluetooth app called the Smartbell Dashboard.

Methodology

The hardware for this project consists of an Arduino Nano 33 BLE Sense, a small breadboard, and a battery. These are mounted directly to one of the dumbbell plates. The Nano 33 BLE has many built-in modules as shown in Figure 1.

Figure 1: Arduino Nano 33 BLE Sense [1]


HTS221 (Temperature and Humidity Sensor)
APDS9960 (Proximity, Light, Colour, Gesture)
LSM9DS1 - 9-Axis IMU (Accelerometer, Gyroscope, Magnetometer)
RGB Programmable LED
Power LED (Green)
Nordic nRF 52840 Processor with NINA B306 Bluetooth module
Micro-USB Port
Programmable LED (Orange)
LPS22HB (Pressure Sensor)
MP34DT05-A Microphone

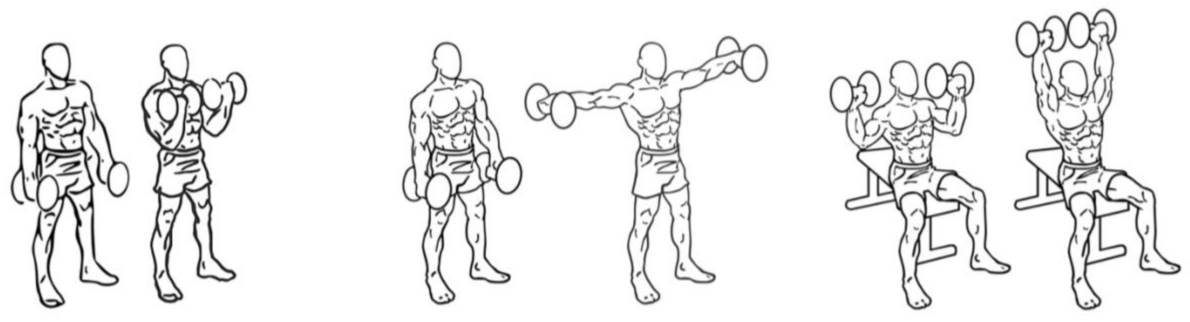
The primary sensor used in this project is the LSM9DS1 9-axis Inertial Measurement Unit (IMU) built into the Nano 33 BLE. The IMU contains an accelerometer, a gyroscope, and a magnetometer, which can capture precise movement data. By extracting features such as the Power Spectral Density (PSD) and the Root Mean Square (RMS) value from the captured data, different movements can be classified using a Nearest Neighbor (NN) model.

The NINA B306 Bluetooth module on the Nano 33 BLE enables data to be sent to other devices in real-time. The Smartbell can be connected to a Bluetooth device such as a laptop or a phone, and the inference results can be seen on the Smartbell Dashboard. In Bluetooth Low Energy (BLE), devices are either central or peripheral. The Arduino is set up as a peripheral device i.e., it broadcasts the data to the central device such as a laptop. For setting up the BLE services and characteristics, the ArduinoBLE library [2] is used.

Data Collection

The Smartbell will be able to identify three different exercises namely the biceps curl, lateral raise, and overhead press. Figure 2 shows the movements for these exercises.

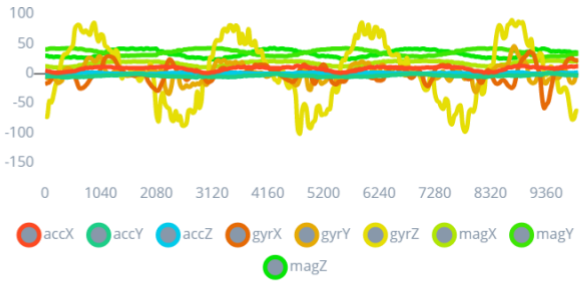
Figure 2: Different Exercises: (a) Biceps Curl (b) Lateral Raise (c) Overhead Press [3]



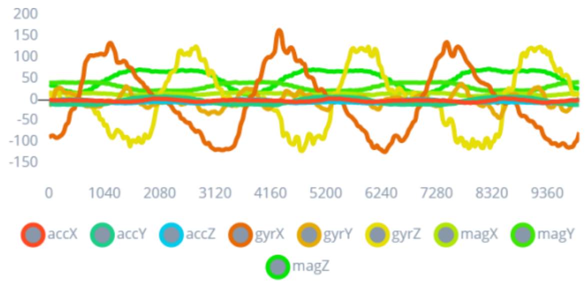
The data acquisition and model training are achieved using Edge Impulse which highly accelerates the deployment of Machine Learning models on edge devices. The Nano 33 BLE can be read data using either the edge-impulse-daemon command from the Edge Impulse Command Line Interface (CLI) or the WebUSB feature on the website. However, the CLI method is the recommended one and is reliable for longer periods of data collection.

The exercise data was collected for four classes: Lateral Raise, Bicep Curl, Overhead Press, and Idle. To avoid overfitting in the training model, data was collected from four different individuals for both right and left hands. Each data was sampled at 100 Hz for a duration of 10 seconds. About 5 minutes of inertial data was collected for each exercise. Figure 3 shows the raw data plots of the different classes.

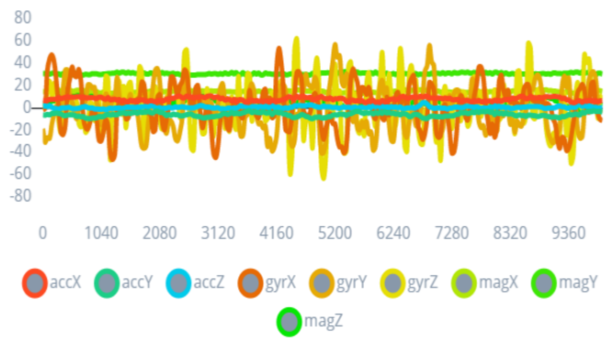
Figure 3: Raw Data Plot of (a) Lateral Raise (b) Bicep Curl (c) Overhead Press (d) Idle



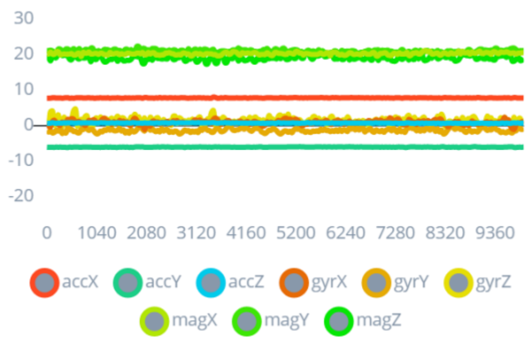
(a)



(b)



(c)



(d)

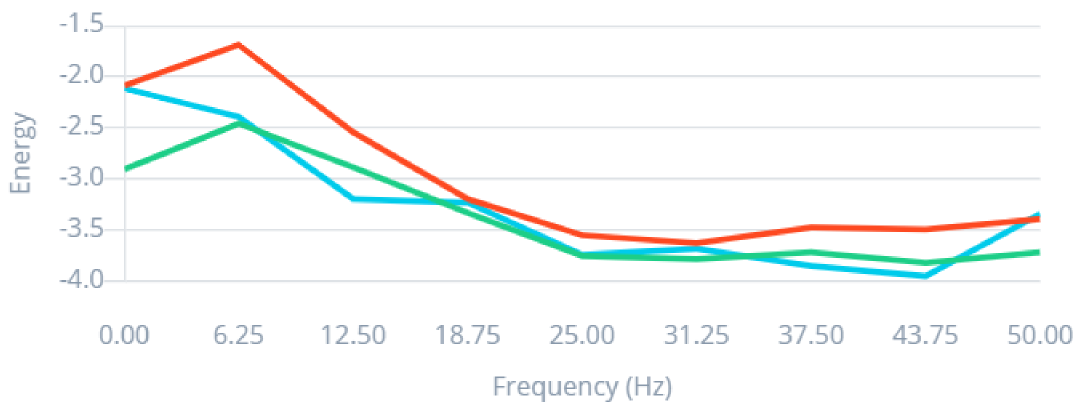
Digital Signal Processing

The acquired raw data is first processed, via filtering, before it can be analyzed. This involves cleaning and formatting the data, removing outliers and noise, and normalizing the data to a consistent scale. To train a machine learning model, the data needs to be labeled with the correct output or class.

The on-device processing of the data required a peak RAM usage of 8KB and processing time of about 20ms which are both ideal for our use case.

The spectral power, shown in Figure 4, gives us a measure of the distribution of energy across different frequency bands in the signal. It provides information about the frequency content of the signal and how the energy is distributed across different frequencies.

Figure 4: Spectral Power (log)



Classification

Once the data is filtered and noise is removed, the next step is to extract features from the data that will be used to train the machine learning model. This involves applying transformations to the data, such as windowing or spectral analysis, and selecting specific features from the data that are relevant such as the gyroscope and accelerometer readings. The additional features used to classify the data include: spectral power, skewness, RMS values and kurtosis.

Once the data is preprocessed and features are selected, it can be used to train a machine learning model. Edge Impulse provides a range of algorithms and tools for training machine learning models. After the model is trained, it can be deployed to an edge device for real-time prediction.

The classified data is shown in Figure 5, and the discrepancies between the different datasets can clearly be seen.

Figure 5: Classified Data

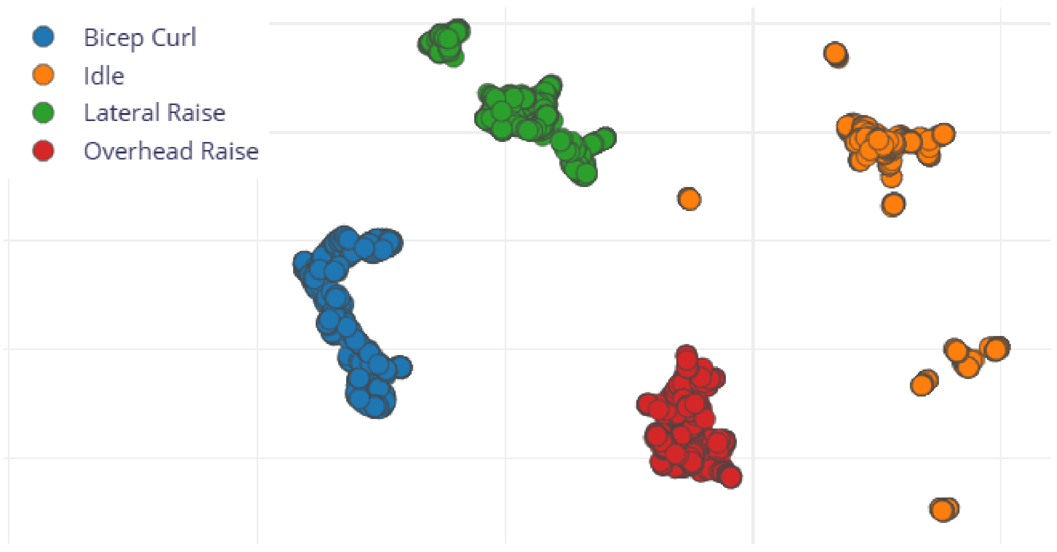
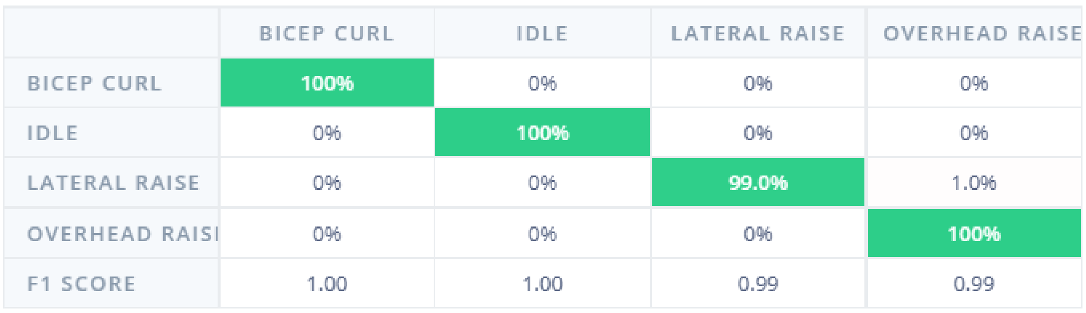


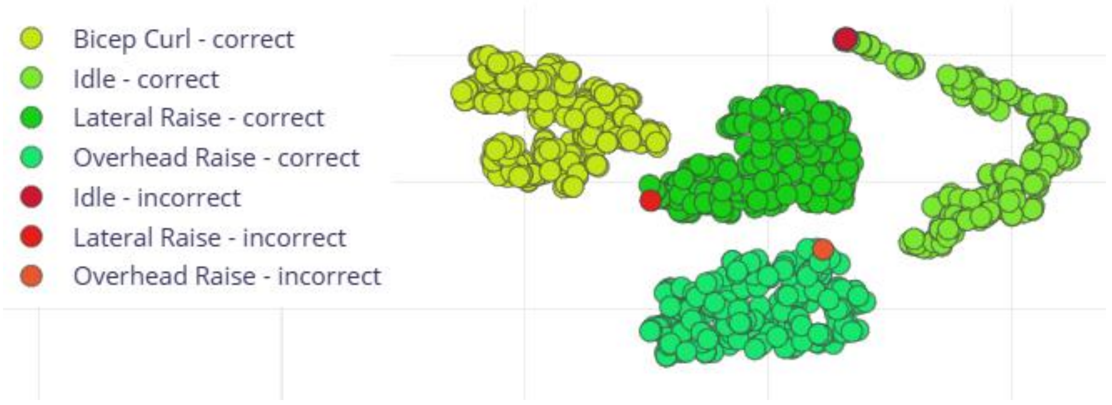
Figure 6 shows the confusion matrix of the classification model. A confusion matrix is a table that is used to evaluate the performance of a classification algorithm. It is a way to visualize the predicted classes of a classification model compared to the true classes. The confusion matrix allows you to see how many instances of each class the model predicted correctly and incorrectly.

Figure 6: Confusion Matrix



	BICEP CURL	IDLE	LATERAL RAISE	OVERHEAD RAISE
BICEP CURL	100%	0%	0%	0%
IDLE	0%	100%	0%	0%
LATERAL RAISE	0%	0%	99.0%	1.0%
OVERHEAD RAISI	0%	0%	0%	100%
F1 SCORE	1.00	1.00	0.99	0.99

Figure 7: Training Results

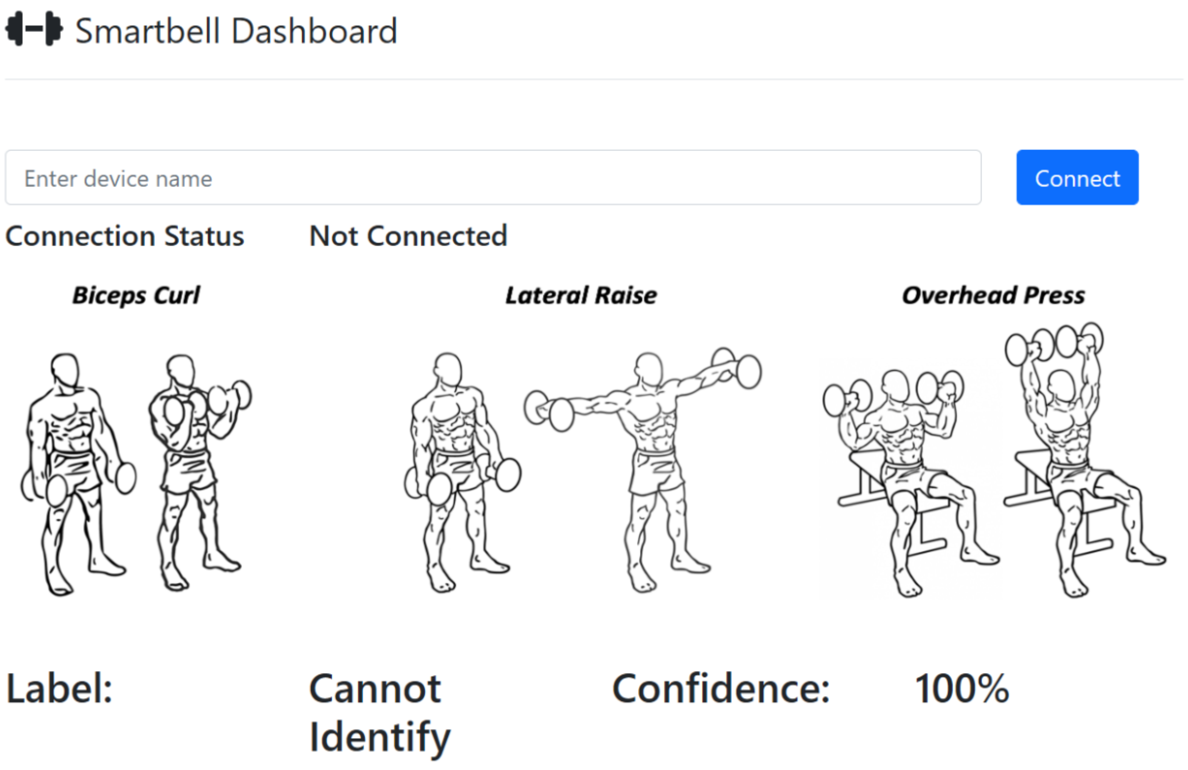


Dashboard

The Smartbell Dashboard is created using HTML, CSS, JavaScript, and the Web Bluetooth Application Programming Interface (API). Since, we did not have much experience with web development, we followed an online tutorial [4] and tweaked the design and code to our needs. The Web Bluetooth API is supported on most mainstream browsers such as Google Chrome, Microsoft Edge, and Opera. The only major browsers currently not supporting the API are Mozilla Firefox and Safari. However, this problem can be circumvented by installing a browser that does support Web Bluetooth. The web application can be stored locally and opened in a laptop. However, for using it on a phone, a website hosting service such as Tiiny Host can be used.

The Dashboard can search for and connect to nearby BLE devices. The connection status of the Bluetooth, the predicted label, and the confidence percentage can be viewed from the dashboard in real-time. The Smartbell Dashboard is shown in Figure 8.

Figure 8: Smartbell Dashboard



Smartbell Dashboard

Connection Status Not Connected

Label: Cannot Identify

Confidence: 100%

References

[1] A. Raj, "Arduino Nano 33 BLE Sense Review - What's New and How to Get Started?," Circuit Digest, 18 11 2019. [Online]. Available: https://circuitdigest.com/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide. [Accessed 10 12 2022].

[2] Arduino, "ArduinoBLE - Arduino Reference," Arduino, 2022. [Online]. Available: https://www.arduino.cc/reference/en/libraries/arduinoble/. [Accessed 10 12 2022].

[3] Microchip Technology, Inc., "Create a Smartbell with Edge Impulse," Microchip Technology, Inc., 2022. [Online]. Available: https://microchipdeveloper.com/machine-learning:smartbell-with-edge-impulse. [Accessed 10 12 2022].

[4] M. Afaneh, "Web Bluetooth: A Getting Started Guide for Developers," NovelBits, 8 11 2022. [Online]. Available: https://novelbits.io/web-bluetooth-getting-started-guide/. [Accessed 10 12 2022].