# Syntiant: Training Neural Networks for Sensors

SYNTIANT

Team Members: Marco Conati(Team Lead), Alexandra Loumidis, Yoo-Jin Hwang, Kevin Kong, Kip Macsai-Goren, Penny Hernandez, Nick Zemtzov

Liaisons: David Garrett, Will McDonald

Clinic Advisor: Professor Daniel Contreras

HARVEY MUDD COLLEGE

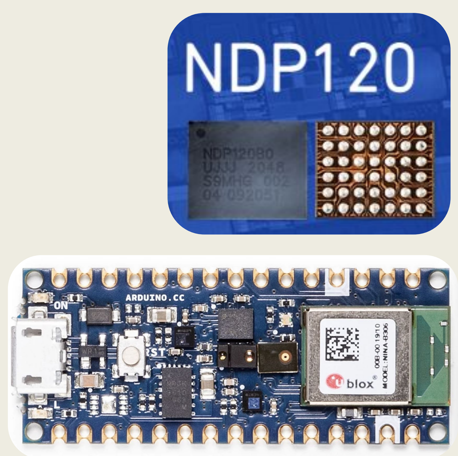
## Background

Syntiant Corporation produces ultra-low power, high-performance neural decision processors (NDP) for use in sensor applications.

- Particular focus on always-on applications & audio keyword interfaces
- Readily integrated with off-the-shelf battery-powered electronics

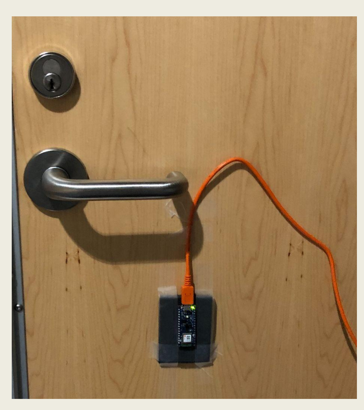
## Problem Statement

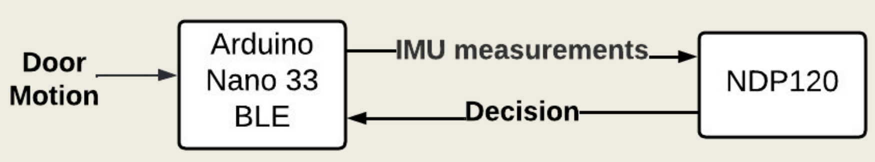
- **Train a neural network (NN)** on Syntiant's NDP120 with data collected from a 9-axis IMU to identify sensor events in a home environment.
- **Design a printed circuit board (PCB)** to integrate the NDP120 with the IMU to test the NN's predictions for the initial use case.




## Door Use Case & Data Collection

The ultimate goal of the project is for the team to use the PCB with the on-board NDP120 and Arduino to test the neural network's predictions in a real-time application. The data was collected from placing the 9-axis IMU on a door in different orientation and then labeling them manually for each test case.






Beyond the base use-case of predicting open, close, and non-events, the model was trained on a more complex dataset to validate its robustness in different scenarios. The new dataset had four cases: open, close, tampering, and non-event (Fig. 1). Additionally, a third of the new dataset was generated with the sensors in random locations on the door. This was meant to simulate a situation where a consumer of the potential product may not precisely place sensors.

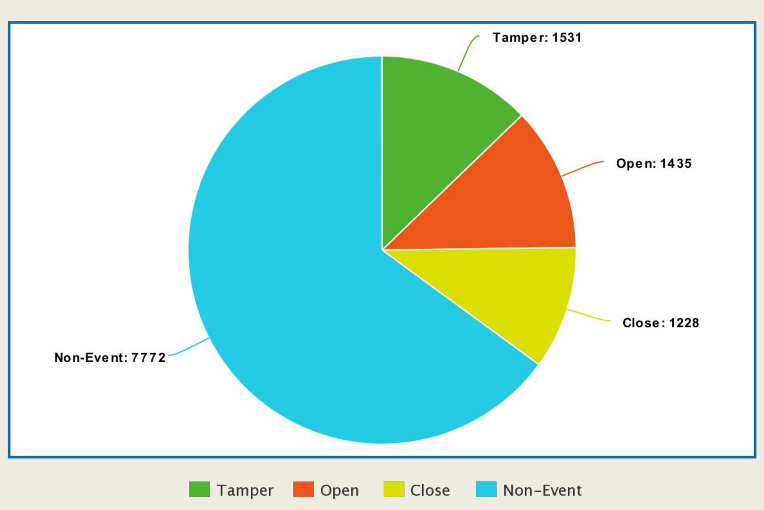
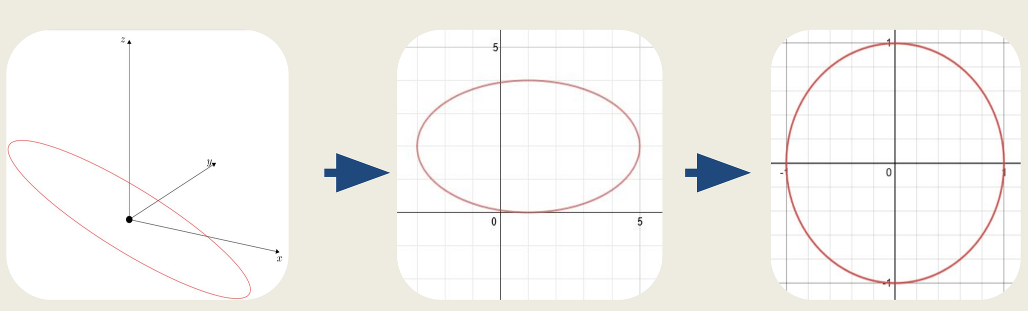



Figure 1: Data collection distribution of different test cases

## Door State Estimation

In order to give users and the neural network more information about what is happening around a house, the team made use of the rest of the sensors on the Arduino.`



The team used a resource provided by the Ref 1 in order to implement a tilt-compensated algorithm that incorporates 3-axis accelerometer and magnetometer sensors. The idea is to calculate the pitch, roll and yaw that define the PCB orientation on the door in order to estimate the door's angle using the rotation matrix as seen in Fig 2. The tilt compensation algorithm can be defined from the equations below assuming that θ is the pitch and ϕ is the roll. There are fixed magnetic offsets called hard iron offsets that add to the true magnetometer sensor output. This is eliminated by calibrating and finding the center of the sphere coordinates as seen in Fig 3.

$$\begin{pmatrix}\cos\theta & \sin\theta\sin\phi & \sin\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi\end{pmatrix}\begin{pmatrix}B_{px}-V_x\\ B_{py}-V_y\\ B_{pz}-V_z\end{pmatrix}$$

Figure 2: Tilt-compensation rotation matrix provided by Reference [1]

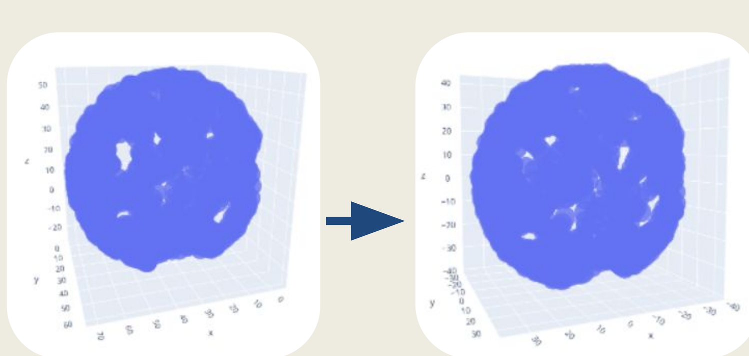


Figure 3: Uncompensated accelerometer measurements where the origin of the sphere is the hard-iron offset (left), measurements after hard iron offset calibration (right)

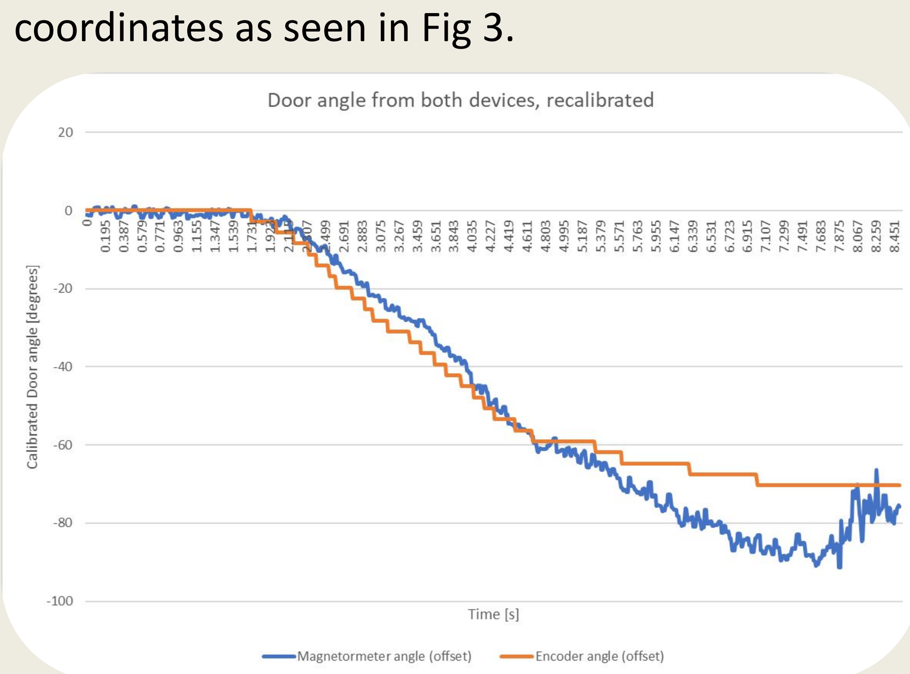



Figure 4: Measurements from opening a door to 90 degrees. Measurements of the magnetometer angle estimated from the tilt-compensation algorithm (blue) and ground truth from our encoder readings (orange)

A Kalman Filtering algorithm used the tilt-compensated magnetometer and gyroscope values to determine the door's pitch. The gyroscope was used to predict changes in pitch, while the magnetometer was used in a correction step.

## Neural Network Design

The chosen model is a 1D convolutional neural network. The original design was motivated by Jason Brownlee's single headed network in the article: 1D Convolutional networks for Human Activity Recognition [2]. In his article, the model was used on 6-axis IMU data to detect human activities like walking, sitting, and standing. This model was used as a starting point since it had demonstrated success in event detection using accelerometer data.

For the training/validation/testing dataset split, a 60/20/20 division was used. The neural network architecture used is shown in Fig. 5. The architecture was found by testing various kernel widths and calculating the validation loss for each layer dimension. There was no noticeable effect on the overall performance for the changes in the kernel dimensions so kernels of three was chosen to minimize model parameters.

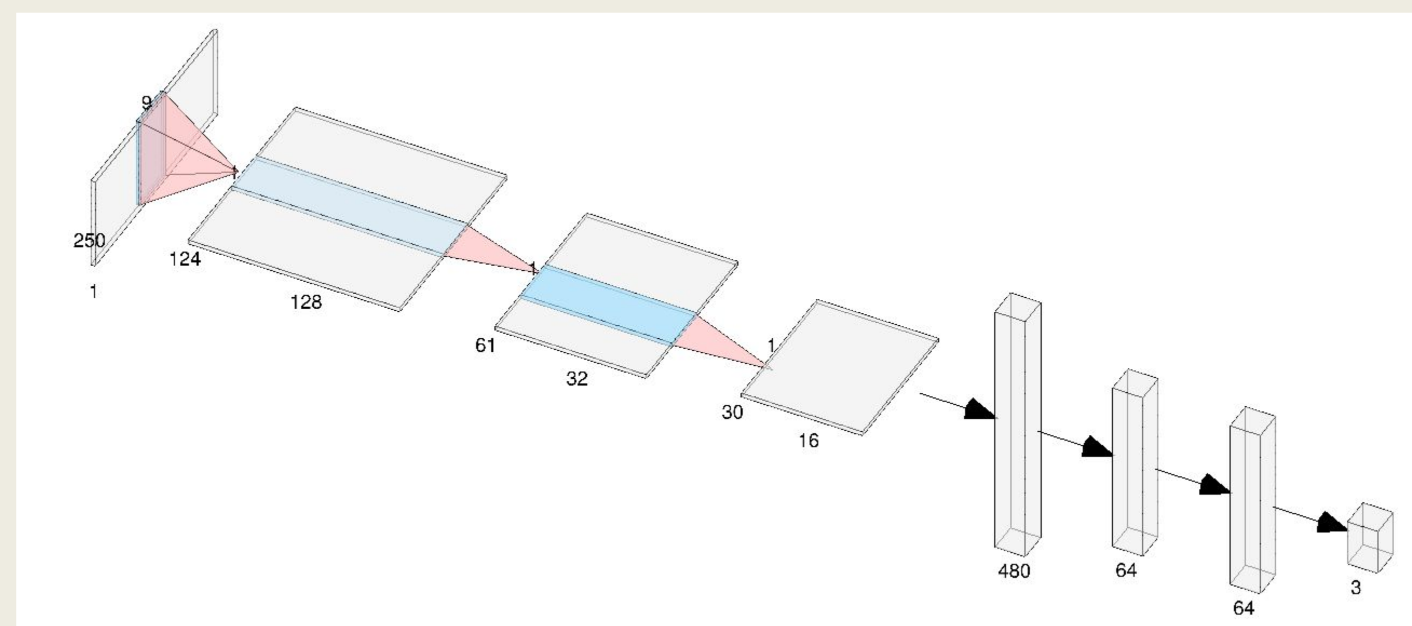
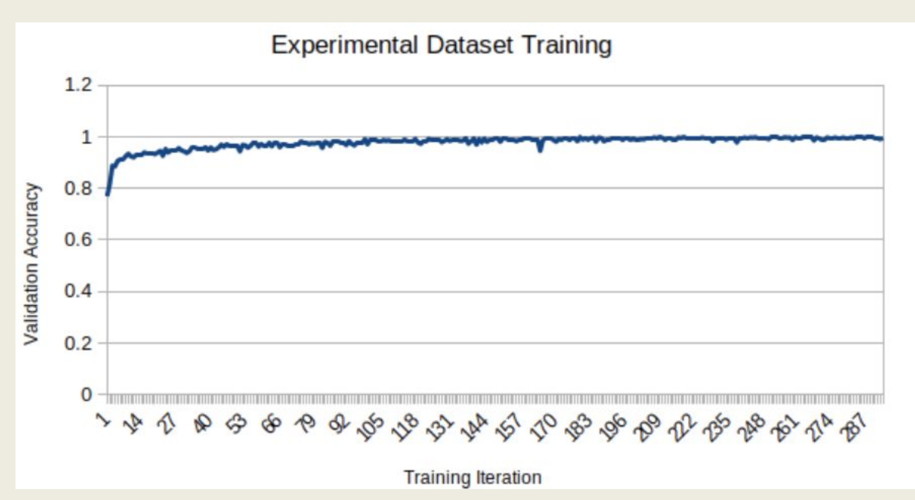



Figure 5: Picture of final neural network architecture. Specific dimensions from left to right are: (1,9,250), (128,1,124), (32,1,61), (16,1,30), (480), (64), and finally (3)

The accuracy of the model during training is shown and quantified with the confusion matrix in Figure 6.

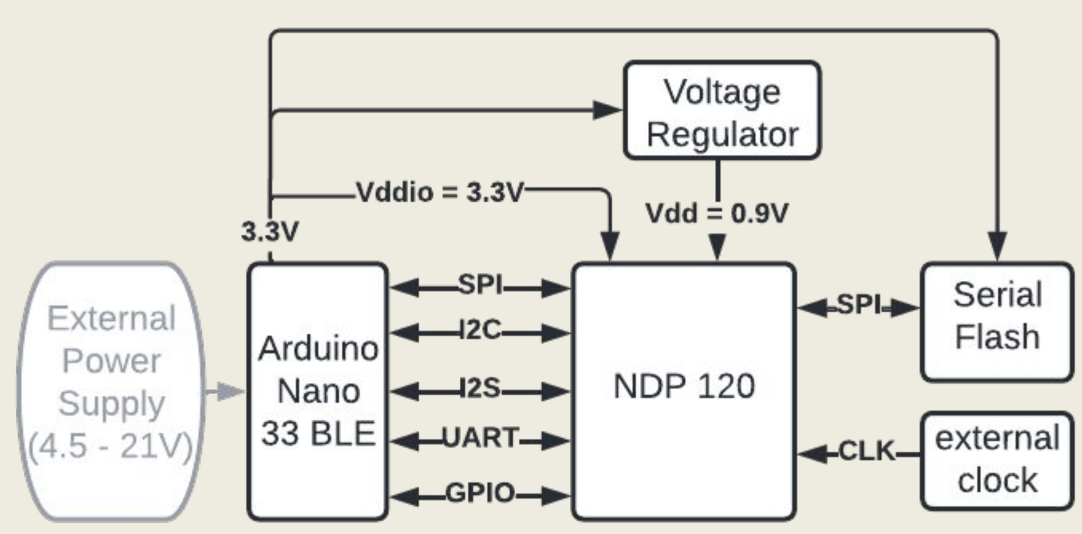



Predicted

| Actual | Open | Close | Tamper | Non-Event |
|---|---|---|---|---|
| Open | 277 | 0 | 0 | 0 |
| Close | 0 | 239 | 1 | 0 |
| Tamper | 0 | 0 | 308 | 1 |
| Non-Event | 0 | 0 | 0 | 1551 |

Figure 6: Picture of experimental dataset training on neural network (left) and the corresponding confusion matrix with each event (right)

## PCB Design




## PCB - Board Layout

A PCB was designed with the goal of holding and supplying connections between the NDP120 and an Arduino Nano 33 BLE device, which included an onboard 9-axis IMU sensor. Additional components such as a voltage regulator, serial flash memory, and external clock were also included in the design. The board was made to be 4 layers with power, ground, and two signal planes and has headers to insert the Arduino device on headers as seen in Fig. 7

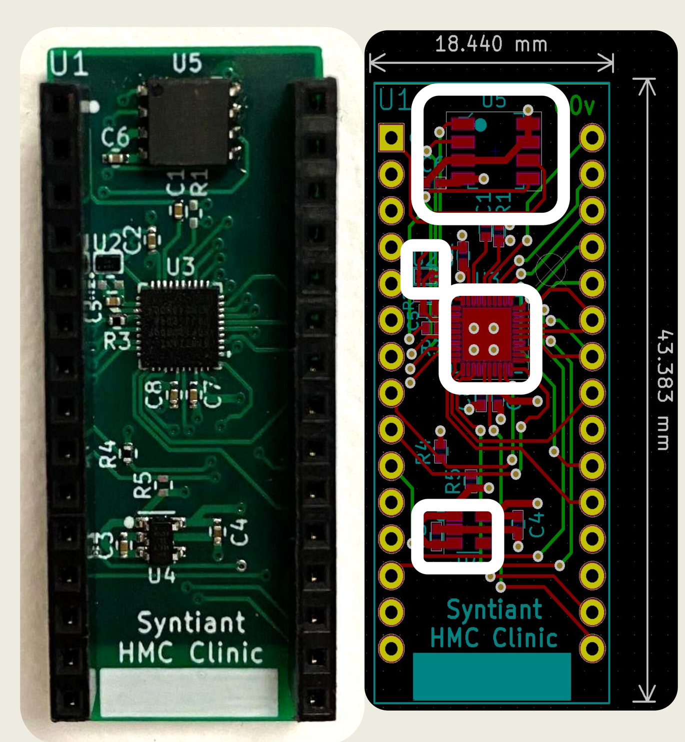



Figure 7: Picture of assembled PCB (left) and schematic view of top and bottom layer with silkscreen (right). White highlighted components (right) from top to bottom are: Serial flash, external clock, NDP120, voltage regulator

The team has successfully tested the power supplies, SPI communication, and the clock frequency between the motherboard and the sensor using Syntiant's firmware package. Next steps will be to test communication lines like I2S and send the data reading the predictions from the neural network and the ground truth reading of the door angle.

## References

[1] T. Ozyagcilar, "Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors," Freescale Semiconductor Application Note. https://www.mikrocontroller.net/attachment/292888/AN4248.pdf

[2] Brownlee, Jason. "1D Convolutional Neural Network Models for Human Activity Recognition." Machine Learning Mastery, 27 Aug. 2020, https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/.