# ARCHITECTURE DESIGN

## Gesture Prediction using sensors

| | |
|---|---|
| Written By | Prakhyath Bhandary |
| Revision Number: | 1.0 |
| Last date of revision: | 04/12/2022 |

## Contents

## Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 04-12-2022 | V 1.0 | Initial AD – V1.0 | Prakhyath Bhandary |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# 1 Introduction

## 1.1 Why this Architecture Design Document?

The goal of the Architecture design document (AD) is to give the internal logic design of the actual program code for Gesture Prediction. AD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

## 1.2 Scope

Architecture design document (AD) is a component-level design process that follows a step- by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

## 1.3 Constraints

The application should be user friendly as automated as possible. Users can easily use the application and not needed to know any of the workings

## 2 Architecture

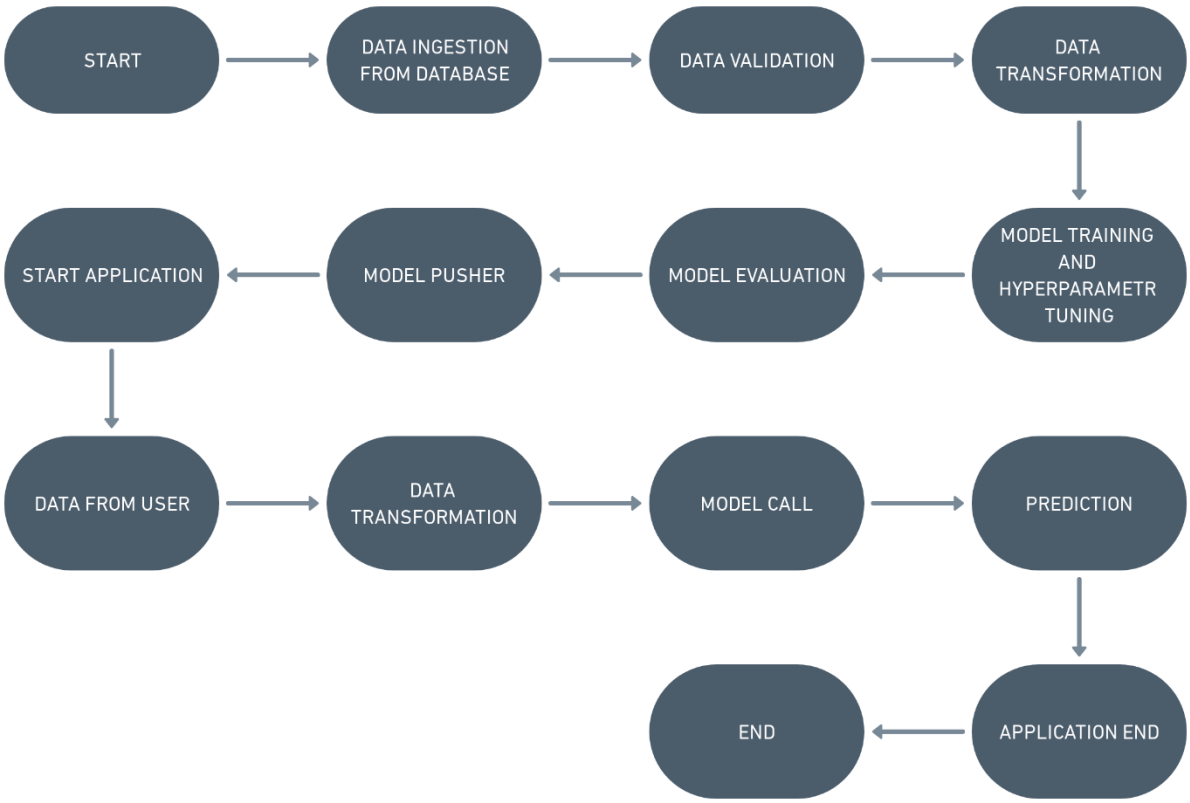


Figure 1: Process flow

# 3 Architecture Description

## 3.1. Raw Data Collection

For recording patterns, we used a MYO Thalmic bracelet worn on a userâ€™s forearm, and a PC with a Bluetooth receiver. The bracelet is equipped with eight sensors equally spaced around the forearm that simultaneously acquire myographic signals. The signals are sent through a Bluetooth interface to a PC. We present raw EMG data for 36 subjects while they performed series of static hand gestures.The subject performs two series, each of which consists of six (seven) basic gestures. Each gesture was performed for 3 seconds with a pause of 3 seconds between gestures.

## 3.2. Export Data from Database

Data Export from Cassandra Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing and Model Training.

## 3.3. Data Pre-Processing

Before building any model, it is crucial to perform data pre-processing to feed the correct data to the model to learn and predict. Model performance depends on the quality of data fed to the model to train. This Process includes.

a) Handling Null/Missing Values
b) Outliers Detection and Removal

## 3.4. Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a data set.

a) Validate number of columns and their types
b) Drop columns with zero standard deviation
c) Detect data set drift

### 3.5. Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

### 3.6. Reporting

Reporting is a most important and underrated skill of a data analytic field. Because being a Data Analyst you should be good in easy and self-explanatory report because your model will be used by many stakeholders who are not from technical background. a) High Level Design Document (HLD)

b) Low Level Design Document (LLD)

c) Architecture

d) Wireframe

e) Detailed Project Report

f) Power Point Presentation

### 3.7. Modelling

Data Modelling is the process of analyzing the data objects and their relationship to the other objects. It is used to analyze the data requirements that are required for the business processes. The data models are created for the data to be stored in a database. The Data Model's main focus is on what data is needed and how we have to organize data rather than what operations we have to perform.

### 3.8. Data Validation

Here Data Validation will be done on the test set.

### 3.9. Deployment

The final model is deployed on GCP by dockerizing using docker, and using the tool Github Actions.

# 4. Technical Specification

## 4.1. Dataset

For recording patterns, we used a MYO Thalmic bracelet worn on a user's forearm, and a PC with a Bluetooth receiver. The bracelet is equipped with eight sensors equally spaced around the forearm that simultaneously acquire myographic signals. The signals are sent through a Bluetooth interface to a PC. The signals are processed using machine learning techniques to predict the gesture

```
df.head()
```

| | time | channel1 | channel2 | channel3 | channel4 | channel5 | channel6 | channel7 | channel8 | class |
|---|---|---|---|---|---|---|---|---|---|---|
| 2287 | 2400 | -0.00001 | 0.00000 | -0.00001 | 0.00000 | 0.00000 | -0.00001 | -0.00001 | 0.00001 | 1.0 |
| 2288 | 2401 | -0.00001 | -0.00002 | 0.00000 | -0.00001 | -0.00001 | -0.00001 | -0.00003 | -0.00002 | 1.0 |
| 2289 | 2402 | -0.00001 | -0.00002 | 0.00000 | -0.00001 | -0.00001 | -0.00001 | -0.00003 | -0.00002 | 1.0 |
| 2290 | 2403 | -0.00001 | -0.00002 | 0.00000 | -0.00001 | -0.00001 | -0.00001 | -0.00003 | -0.00002 | 1.0 |
| 2291 | 2404 | -0.00001 | -0.00002 | 0.00000 | -0.00001 | -0.00001 | -0.00001 | -0.00003 | -0.00002 | 1.0 |

The dataset consists of 10 columns. Time column is the amount of time taken for the user to perform the gesture. 8 columns named channel1,channel2, …, channel8 are the readings of 8 sensors present in the bracelet which surrounds the forearm of the user. Classes 1 to 8 represent the various gestures performed by the user.

0 - unmarked data,

1 - hand at rest,

2 - hand clenched in a fist,

3 - wrist flexion,

4 – wrist extension,

5 – radial deviations,

6 - ulnar deviations,

7 - extended palm (the gesture was not performed by all subjects)

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes

| | time | channel1 | channel2 | channel3 | channel4 | channel5 | channel6 | channel7 | channel8 | class |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237908e+06 | 4.237907e+06 |
| mean | 3.113689e+04 | -7.911481e-06 | -9.416077e-06 | -9.548735e-06 | -9.637838e-06 | -1.599724e-05 | -1.085528e-05 | -9.364637e-06 | -9.696860e-06 | 1.265671e+00 |
| std | 1.868079e+04 | 1.631110e-04 | 1.192228e-04 | 1.241846e-04 | 2.257728e-04 | 2.724188e-04 | 2.151405e-04 | 1.527311e-04 | 1.720933e-04 | 1.989693e+00 |
| min | 0.000000e+00 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | -1.280000e-03 | 0.000000e+00 |
| 25% | 1.529000e+04 | -3.000000e-05 | -4.000000e-05 | -4.000000e-05 | -6.000000e-05 | -8.000000e-05 | -6.000000e-05 | -4.000000e-05 | -3.000000e-05 | 0.000000e+00 |
| 50% | 3.054600e+04 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | -1.000000e-05 | 0.000000e+00 |
| 75% | 4.580400e+04 | 2.000000e-05 | 2.000000e-05 | 3.000000e-05 | 4.000000e-05 | 5.000000e-05 | 3.000000e-05 | 2.000000e-05 | 1.000000e-05 | 2.000000e+00 |
| max | 9.714400e+04 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 1.270000e-03 | 7.000000e+00 |

Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for the median, play an important factor in deciding which value to be chosen as a priority for further exploration tasks and analysis.

### 4.2 Logging

We should be able to log every activity done by the user

- The system identifies at which step logging require.
- The system should be able to log each and every system flow.
- The system should not be hung even after using so much logging. Logging is just because we can easily debug issuing so logging is mandatory to do.

## 5. User Input/Output Workflow