Imperial College London

MEng Final Year Project

Imperial College London

Department of Electrical and Electronic Engineering

No Data is Bad Data: Classification of food type and calorie intake from In-Ear signals

Author:
Yuna Valade

Supervisors:
Matteo Bermond,
Prof. Danilo Mandic

Second Marker:
Dr. Ayush Bhandari

June 21, 2023

Plagiarism Statement

I affirm that I have submitted, or will submit, an electronic copy of my final year project report to the provided EEE link.

I affirm that I have submitted, or will submit, an identical electronic copy of my final year project to the provided Blackboard module for Plagiarism checking.

I affirm that I have provided explicit references for all the material in my Final Report that is not authored by me, but is represented as my own work.

I have used ChatGPT v3.5 as an aid in the preparation of my report. I have used it to improve the quality of my English throughout, however all technical content and references comes from my original text.

Abstract

Traditionally, the gold-standard method to measure the electrical activity of the brain, the Electroencephalogram (EEG), demands professional supervision and the use of multiple electrodes attached to the subject's scalp [1]. However, the Hearables, an in-ear wearable device placed in, on and around the ear, introduces a more portable, unobstructive and discrete method to measure EEG signals and continuously monitor patient's vital signals [2][3]. This device aims to enhance patient's comfort and user-friendliness of the recording process. Nevertheless, continuous monitoring comes with inherent challenges stemming from patients' daily activities such as speaking, chewing and walking which introduces undesired information (i.e. noise) in recordings which are known as artefacts.[4][5]

Noises derived form muscular activity are usually undesired and for this reason removed from recordings using different methods [5] . This work aims to make sense of the information embedded in the artifacts, in particular induced chewing and swallowing artefacts to identify patterns that distinguish types of ingested foods and ultimately attempt to estimate calorie intake.

A protocol to record in-ear physiological bio (EEG) and non-bio signals (audio) was defined and implemented to record a subject ingesting different types of food and drinks. The goal is to extract and identify different time and frequency features to classify them in different categories ("Hard food", "Medium-Soft Food", "Soft Food", "Flat drink", "Fizzy drink"). Moreover, use some of the features implemented in the classification, for a calorie intake estimation. A three steps classifiers in series, has then be implemented to label activity versus non activity, food versus drink, food hardness and drink type. The architecture's performances has been evaluated both individually for each classifier and overall for the whole pipeline. An f1-score up to 87% is achieved on individual classifiers and 70% on the whole architecture. The calorie intake was calculated from the bite count, the subject-tailored average mass per bite and the identified food category which had a caloric density associated with it. Results indicate good potential for this approach. With further research and refinement, this approach could improve the dietary monitoring and provide valuable insights into individuals' eating patterns and calorie intake.

Acknowledgements

I would like to thank Professor Danilo Mandic for the opportunity to work on this impactful research, where my knowledge built over the course of my four-year degree could be put to practice.

I would like to dedicate special thanks to Matteo Bermond, who had to endure 180 sips of coke for this project, whose constructive feedback and collaboration has enabled me to navigate through this project.

Finally, I would like to express my heartfelt gratitude to my family for their constant support and encouragements, and my friends for making this university life a memorable experience. They never cease to motivate and inspire me.

Contents

List of Figures

1 Background

Traditionally EEG requires medical expertise to assist the patient, place gelled electrodes across their scalp and interpret the measured signals. The in-ear device is aims to significantly improve the healthcare sector introducing a continuous and discreet 24/7 monitoring and diagnosis of patients. Continuously monitoring also comes with challenges, as patients carry their day to day activities which affects the recording process. This project focuses specifically on chewing and swallowing patterns and follows the concept that "no data is bad data" meaning that useful information can be collected from unwanted artefacts instead of discarding them.

Hospital admissions for eating disorders in the United-Kingdom have increased by 84% in the last five years reaching a total of 24,268 admissions [6]. The Royal College of Psychiatrists describes that: "Children and young people with eating disorders are the worst affected with a rise of 90% in the five-year period, from 3,541 to 6,713 episodes, and a 35.4% increase in the last year alone. A stark rise of 128% is seen in boys and young men — from 280 hospital admissions in 2015/16 to 637 in 2020/2021. An alarming increase of 79% is also seen in adults across the five years.[6]" Eating disorders include obesity, anorexia, bulimia, binge eating and any other unhealthy eating behaviours[7]. These numbers highlight the importance of studies related to chewing and swallowing patterns which with a monitoring device could provide tailored and data-driven diagnostics to the patients.

Before diving into the implementation, building additional background knowledge on the topic, both on medical and research context, is key to gain a more thorough understanding of approaches to consider for this project.

1.1 Medical Context

The used data consists of non-bio (microphone) and bio-signals (in-ear EEG). Electroencephalogram, EEG, is a recording of brain activity, which referring to the NHS website, is a method that can be utilized to help diagnose and monitor a number of brain conditions including seizures, dementia, encephalitis, brain tumours, and sleep apnoea [1]. When analysing EEG signals, there are 5 main recognized human brain frequencies [8]:

- Delta (0.5-4Hz), which is associated with deep sleep and healing.
- Theta (4-8Hz), which is associated with level of attention during a task. They may assist emotional processing, deep relaxation, intuition, and memory consolidation.
- Alpha (8-13Hz), which is associated with level of attention during a task. More specifically they assist during memory usage, creative flow states and also during states of mild anxiety. As this signal is correlated to the Occipital area of the brain, it is well visible when the subjects close their eyes.

- Beta (13-30Hz), which is the most frequently seen rhythm in normal adults and children associated with focus, alertness, and clear thinking.
- Gamma (>30Hz), mainly up to 45Hz, correlates to sustained attention and high cognitive performance. In other words, gamma frequencies are linked to higher mental functioning involving learning, concentration, and self-control.

This knowledge is key when considering pre-processing recordings of EEG, to understand how the signal can be filtered such that relevant frequencies containing information are kept and observed.

1.2 The In-Ear Device

For further context on the device, a picture of the used earplug for the in-ear EEG recording as well as its set up in the ear has been inserted:

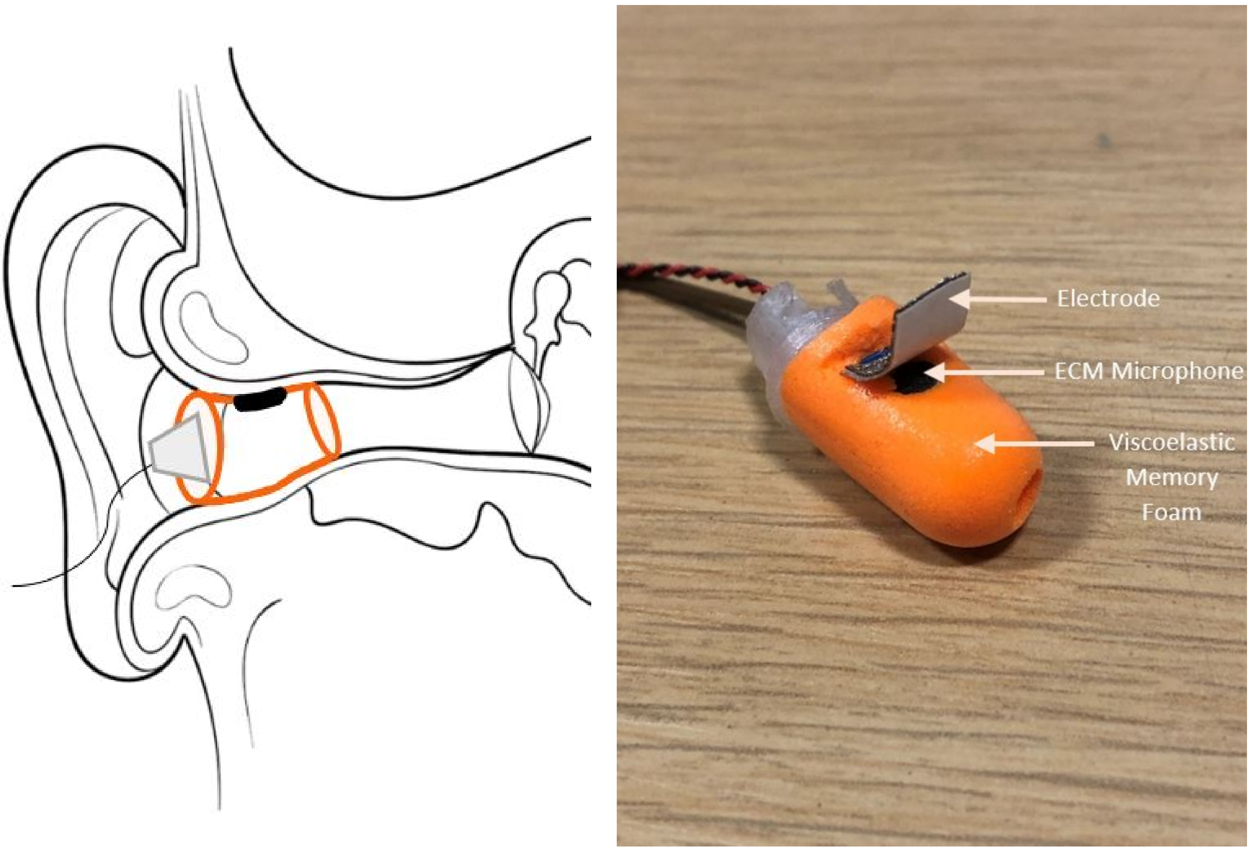


Figure 1: Diagram of the Earplug Set-Up and Picture of the Earplug

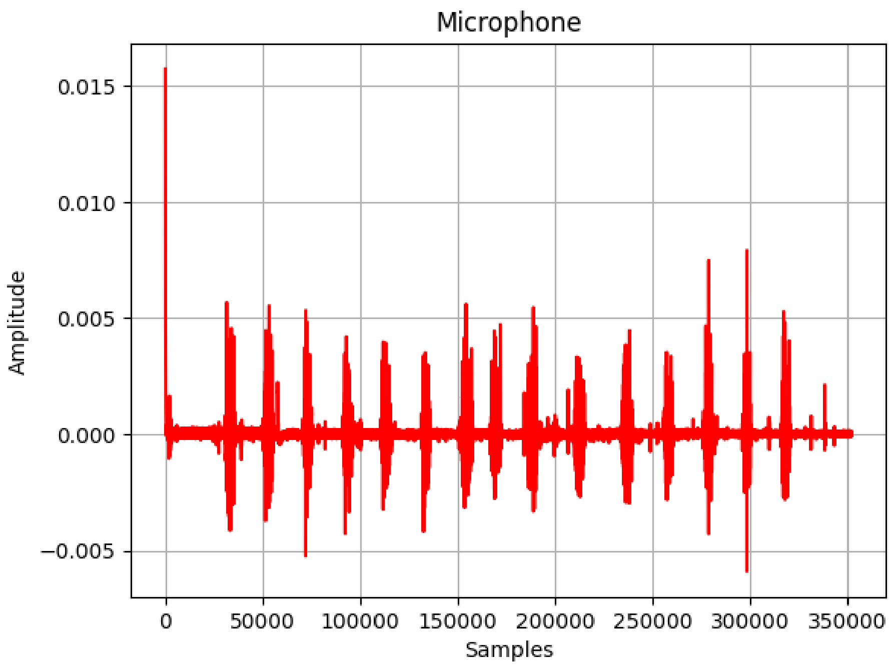
The earplug is realized with a viscoelastic memory foam which is a flexible material that conforms to any shape of an object under the influence of heat and pressure, and it slowly returns to its original shape when the pressure is released. [9]. Its compression ability during insertion in the ear canal makes it ideal for a generic earmould concept to make ear-EEG suitable for immediate and widespread use. Additionally, its conformity provides secure contact between the electrodes and the ear canal skin for any shape of ear-canal, enabling consistent good-quality recordings of EEG signals across patients [10]. Previous models included silicone molds or customized earpieces which were not only more costly but also required lengthier manufacturing processes [10].

The electrodes used to detect EEG from the ear canal, described further in-depth in previously published paper [10], are made of a conductive cloth of silver-coated nylon interwoven with elastic fibres; these are glued to the viscoelastic earplug and gelled before insertion. This enables the electrode to stretch in both directions and has a very low impedance (only of 0.5 Ohm/sq). To produce the electrodes, strips of fabric 3 mm wide and 1 cm long were cut; then standard wires were attached to the strips using a silver conductive epoxy adhesive.

The ECM Microphone placed underneath the electrode is a type of microphone that uses an electret diaphragm and condenser to convert sound waves into electrical signals. As the use of the earplug requires conductive gel to be applied on the device, it is key to have a microphone robust to moist environment. The diaphragm in the ECM microphone helps to shield the internal components from direct contact with its external environment [11]. Thus making this type of microphone an ideal component for this type of application.

The recording process usually involves subjects following a protocol to stimulate their senses and obtain relevant recordings. For example, in the In-Ear EEG-Based Attention State Classification Using Echo State Network paper [12], the research team asked their subjects to press a button corresponding to the target stimulus presented at the center of the screen as quickly as possible.

For this study, this device captured two bio-signals such as cross-head EEG and unilateral (left) EEG, as well as non-electrical bio-signals such as microphone data. The plots (3) below demonstrate an example of raw recorded data of a subject chewing on 15 bites of banana. For the microphone recordings, the defined chews can be perceived, on the EEG recording, slight perturbations can be observed on the Crosshead EEG signal.



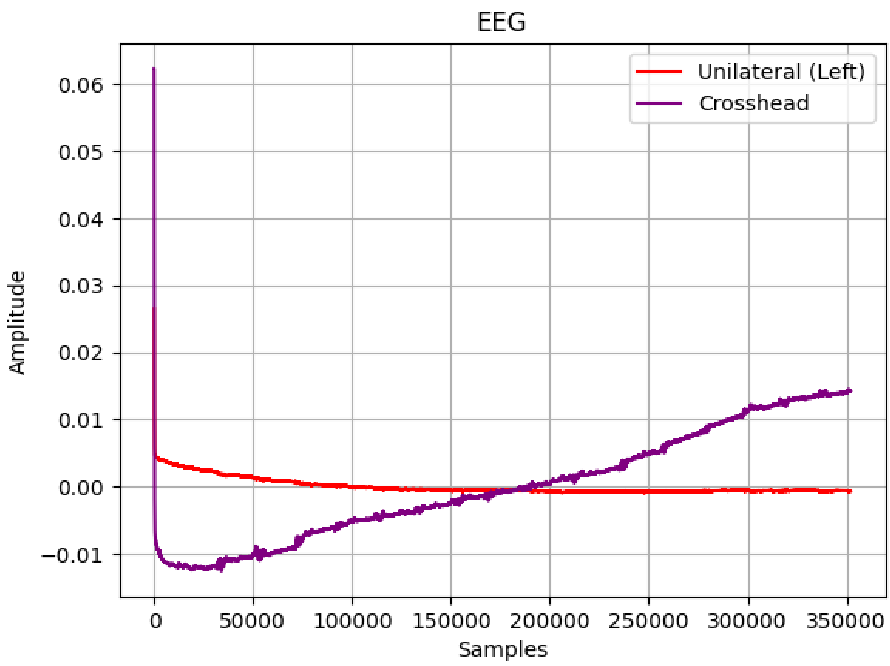


Figure 3: Raw Recordings of 15 Banana bites

1.3 Current Research

Understanding the research undertaken on the device helps to provide context on the recording process, available processing techniques, and tools implemented using data collected from the device. Regarding this project, the paper "Hearables: Making Sense from Motion Artefacts in Ear-EEG for Real-Life Human Activity Classification paper"[4], offers a comprehensive insight into artefact classification. It focuses on collecting data and identifying patterns to classify different artefacts generated by human activities such as sitting, walking, speaking, and chewing. After pre-processing the data, relevant time and spectral metrics are computed and defined as features. Using the One-Way Analysis of Variance (ANOVA) test, the most significant features were selected fed into models such as Decision Trees, Discriminant Analysis, Naive Bayes Classifiers, Support Vector Machines(SVM), K-Nearest Neighbors (KNN) Classifiers and Ensemble Classifiers.

ANOVA is a popular feature selecting test when the data worked with has numerical input variables and categorical outputs. The test is used to identify the most relevant features by comparing the means of at least two groups and determines the differences between them. It measures the feature's significance using two statistical metrics: the $\boldsymbol{f-value}$ and the $\boldsymbol{p-value}$. The $\boldsymbol{f-value}$ quantifies the extent of the differences between the class means relative to the variability within each group:

$$\boldsymbol{f-value} = \frac{\sum n_j(\bar{X}_j - \bar{X})^2}{\sum\sum(X - \bar{X}_j)^2} \tag{1}$$

$\bar{X}_j$ is the sum of the $\boldsymbol{j}$th group, $\bar{X}$ is the overall mean of all group's means, X is a data point in the $\boldsymbol{j}$th group and n_j is the sample size of the $\boldsymbol{j}$th group. The higher the $\boldsymbol{f-value}$ metric the higher the significance of that feature. The $\boldsymbol{p-value}$ represents the probability of obtaining the observed $\boldsymbol{f-value}$, it describes how likely it is that the data would have occurred by random chance, which infers that the lower the $\boldsymbol{p-value}$ the more significant the feature is [13]. The rule of thumb defines that the $\boldsymbol{p-value}$ should be less than 0.05 which indicates strong evidence against the null hypothesis, as there is less than a 5% probability that the results are random [13].

Using this method, an effective selection of features which reduces the probability of misclassified signals can be exerted thus improving the model's ability to classify and performance. In the mentioned paper, each performance is then compared and the best-obtained model is further discussed. As a reminder of what these classifiers are, they will quickly be introduced and explained in the following section.

1.3.1 Classifiers

The 6 presented classical machine learning algorithms are the most commonly used for classification tasks [14]:

1. **Decision Trees**
   Decision Trees are supervised machine-learning algorithms that make decisions in a tree-like model; each internal node represents a feature that branches out according to defined rules. Finally, leaf nodes represent the result of the algorithm. They are commonly used for classification and regression problems. The most impactful parameter to tune for this model is the maximum depth. It determines the number of levels or splits the tree can have which impacts the model's ability to capture patterns or overfit on data.

2. **Discriminant Analysis**
   Discriminant Analysis is a statistical technique aiming to find a linear combination of features that maximizes the separation between different classes. It finds this combination by projecting the data onto a lower dimensional space and searching for sets of linear discriminants that maximize the ratio between class variance to within-class variance. The classes are expected to be normally distributed, meaning that they are assumed to have a Gaussian distribution. The solver used to computed and optimized the discriminant vectors is a tunable parameter. Experimenting with different solvers is key to evaluate the performance of this model.

3. **Naive Bayes Classifiers**
   The Naive Bayes classifier is a Probabilistic Classifier commonly used for classification. It is based on the Bayes theorem that assumes that all features are conditionally independent given their class label. During training, the classifier estimates the mean and variance of each feature in each class based on the training set. During prediction, the classifier computes the z-score distance between the new instance and each class-mean and uses

the Gaussian probability density function to determine which class it belongs to. This classifier has no particular tunable hyperparameters.

4. **Support Vector Machines (SVM)**
   SVMs locate the most suitable function of classification to separate classes during training. This function is known as the hyperplane which maximizes the distance between data points from different classes. Common hyperparameters to improve the model's accuracy include: C (regularisation) parameter and the used kernel.

   $>$ The C (Regularisation) parameter is penalty parameter which represents misclassification. This parameter tells the SVM optimisation how much error is bearable and how to adjust the boundary accordingly. The higher the C parameter the less regularization is applied to the model which allows the model to fit the training data more closely but also has higher chances of overfitting while a smaller C increases regularization.

   $>$ Kernels take low dimensional input spaces and transforms them into higher dimensional spaces. Different kernels enable different types of classifications, the most common options consist of :

   (a) Linear Kernel : $K(X, Y) = X^T \cdot Y$
   (b) Gaussian RBF Kernel : $K(\vec{x}, \vec{l^i}) = e^{-\frac{||\vec{x}-\vec{l^i}||^2}{2\sigma^2}}$
   (c) Sigmoid Kernel : $K(X, Y) = tanh(\gamma \cdot X^T Y + r)$
   (d) Polynomial Kernel: $K(X, Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$

5. **Logistic Regression**
   Despite its name, Logistic Regression is a commonly used classification algorithm which uses the sigmoid function to return the probability of a label between 0 and 1. It is the commonly used for binary classification problems and is defined as follow : $LR = \frac{1}{1+exp(-\eta)}$ The most impactful hyperparameter to consider is the regularization parameter C which controls the amount of regularization applied to the model. The regularization effects are similar to the SVM one.

6. **K-Nearest Neighbour (KNN)**
   When classifying, this algorithm finds the k-nearest neighbors to a new instance based on a distance metric. For classification, the majority class among the neighbors determines the prediction. Its performance is heavily impacted by the choice of $\boldsymbol{k}$ neighbours to consider during runtime.

7. **Random Forest**
   As its name suggests, it is an ensemble of decision trees. It uses bagging techniques to train each tree on a random sampling of the original dataset and a random subset of features. Finally, it aggregates their results to make predictions, it could be based on a majority voting for classification or averaging for regression. They are famously known for their

robustness against outliers and resistance to over-fitting. Usually, the most impactful hyperparameter considered for this model is the number of trees (n_estimators).

1.3.2 Neural Nets

In addition to considering classifiers for the artefact classification, the study on Motion Artefacts [4] considered a 2-layer Neural Network approach. Comparisons between the two machine learning models were drawn where the study concluded that the KNN-model with k=10 neighbours outperformed the Neural Network with an overall performance of 85% to a performance of 81%. These results are good base points to consider when designing the approach to classify in-ear chewing and swallowing signals.

The paper "In-Ear EEG Based Attention State Classification Using Echo State Network"[12], similarly to this study, uses EEG data recorded with an in-ear device but focuses on classifying different attention states using an Echo State Network. also introduces a similar pre-processing approach but defines 5 features of interest to extract. The Echo State Network is a type of recurrent neural network (RNN) with sparsely connected hidden layers. The provided methodology offers an approach to consider and explore when pre-processing, feature-extracting and considering a machine learning implementations with the recorded data. The Echo State Network is a type of recurrent neural network (RNN) with a sparsely connected internal unit layer, the diagram below depicts the structure of the Echo State Network:

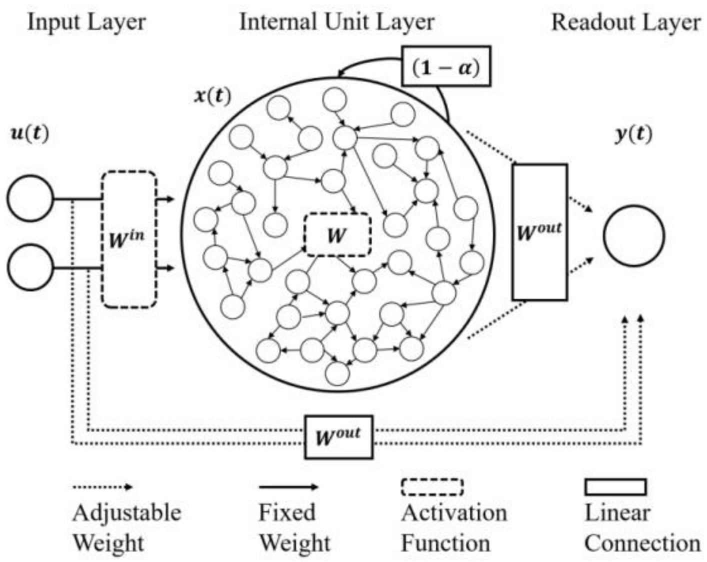


Figure 4: Echo State Network Diagram

Following the tuning of its hyperparameters, the model achieved an accuracy of 81.16% with in-ear EEG signals. Since this study uses similar in-ear EEG signals it is an interesting approach to consider. However, the paper states that the ESN "greatly fluctuates in presence of external artifacts" which would imply that smoothing of recorded signals to achieve good performance. Only patterns amongst the different types of foods are unfamiliar, it could be reflected in rapid changes or oscillations present in EEG signals that could be oversimplified if a smoothing filtered is applied. These studies incline the exploration of classical machine learning approaches with classifiers rather than focus on developping Neural Networks.

1.3.3 Calorie Intake

In recent advancements related to dietary assessment, three notable research papers have introduced different machine learning related methods to estimate calorie intake. They have been insightful to set guidance on potential implementation directions. An identified approach focused [15] on extracting features such as pixel brightness, RBG or contrast from a dataset of food images with labelled calories to train a Regression Model which estimates the calories of the food from its testing data. Calorie estimation using pictures has been an increasingly popular method to keep a dietary record. Only, since the project focuses on the analysis of recorded signals from microphone and EEG, incorporating images which primarily conveys visual information detracts from the project's core objective.

Another approach [16] focuses on estimating the weight of individual food bites using a set of features extracted from both non-audio, manually annotated timestamps, and audio features collected from earbud's microphone. Specifically, 6 non-audio features are extracted, including the number of chews, the mean and standard deviation of chew duration, and the mean and standard deviation of the subject's chewing rate, and finally the food type as a categorical variable. The audio features are extracted time and frequency features such as signal energy in log-scale energy bands, higher order statistics (including skewness and kurtosis), and fractal dimension obtained after preprocessing the microphone recordings. These features are then used to train four different models such as Logistic Regression (LR), Support vector regression (SVR), Feed-forward neural network (FFNN) and a Generalized Regression Neural Network (GRNN). Considering the type of data is similar to the one used in this study, it is a relevant approach to keep in mind when designing the solution for calorie intake.

The "Energy intake estimation from counts of chews and swallows" [17] paper adapts a model using subject-specific parameters such as the average mass per chew of solid foods, the average mass per swallow of solid foods and the average mass per swallow of liquids to estimate the calorie intake. In their study, subjects were recorded using wearables specialised in chewing and swallowing monitoring, and were asked to eat three identical meals used for the training set and a different one used for the testing set. Each meal was labelled by a professional nutritionist and features such as the total number of chews and swallows were extracted from the recordings and used to train the model. Their model is inspired by the Sazonov method [18] which introduces the following relationships to estimate the total ingested mass and deduce calorie intake: The total mass ingested is defined as the sum of the mass of solid foods (M_S)

and the mass of liquids (M_L) ingested:

$$M_T = M_S + M_L \tag{2}$$

1. $M_S = w_S * MPSw_S * N_{SW}^S + w_C * (MPChew * c_f)N_{chew}$
   (a) The mass of solid foods can be split into the addition of two elements. First, the average mass of swallows of solid food times the number of swallows which determines the total mass when swallowing. And, in a second part, the average mass of chew of solid food times the number of chews which determines the total mass chewed.
   (b) Two weight factors can also be noticed, they are indicators of chews or swallows being taken into account. The relationship between the two is of : $w_S + w_C = 1$, where $w_S, w_C \in 0, 0.5, 1$. This is such that if only chews were recorded for a meal the weight factors would be of: $w_S = 0$ and $w_C = 1$ and vice-versa if only swallows were recorded. For the case of both chews and swallows recorded : $w_C = w_S = 0.5$
   (c) c_f is a correction factor which ta such that : $c_f = \frac{CPS_{w_{training}}}{CPS_{w_{validation}}}$
2. Mass of Liquid Food: $M_L = MPS_{w_L} * N_{SW}^L$

Calorie Intake:

$$CI = \sum_i^N m_{T_i} CD_i \text{ where} \tag{3}$$

1. m_{T_i} is the mass of the concerned food or liquid ingested
2. CD_i is the caloric density associated to the food type

This is a very relevant approach to keep in mind as these parameters can be annotated during recording sessions, and the identification of the food type could also be used to calculate calorie estimate.

2 Materials and Methods

The final aim of this project consists of searching for patterns in subject's chewing and swallowing signals, classifying different types of food based on these patterns and approximate the calorie intake. This study will follow the following outline:

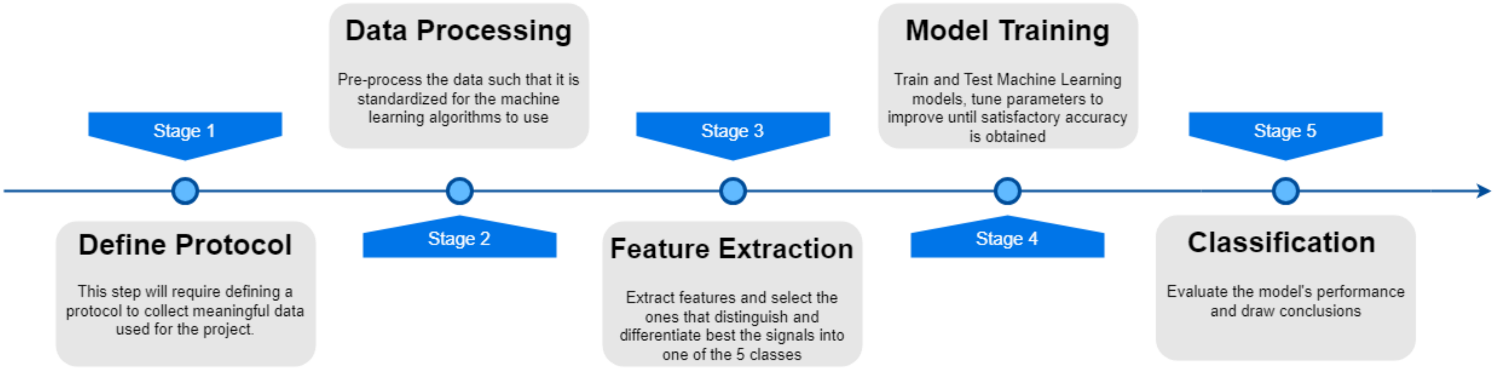


Figure 5: Highlevel Timeline of thr Implementation Process

As there are no available datasets online, containing recordings of both EEG and microphone data needed for the purpose of this study, a protocol was put into place to record subjects ingesting different types of food and liquids. Implementing an accurate protocol which enables optimal extraction of relevant information from recordings has proven to be challenging.

2.1 Recording Set Up

The recordings are performed within the EEE department laboratories under the Imperial College London ethics committee approval JRCO 20IC6414 with subjects full informed consent.

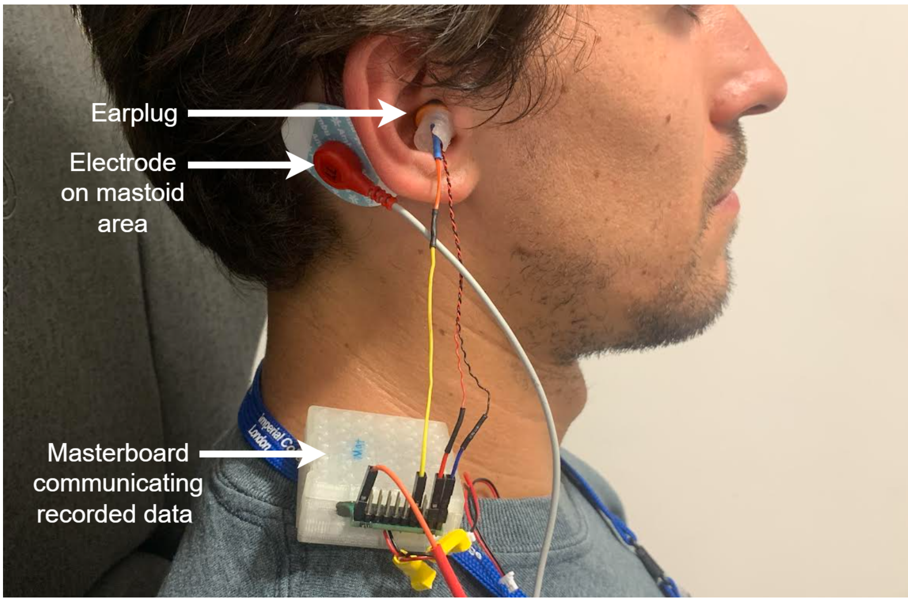


Figure 6: Subject Set Up

The subject is set up as the picture (6) above illustrates; after applying conductive gel to the earplugs they are inserted into the ear canal, an electrode is placed as reference on the mastoid area. The subject is set up such that: the in-ear electrodes focus on capturing EEG artefacts caused by the muscular activity induced by the chewing process, the microphone focuses on recording the friction between the electrode and the ear canal skin caused by mastication.

2.2 Recording Protocol

To begin with, food types were split into 3 categories such as: 'Soft Food', 'Medium Soft Food' and 'Hard Food'. On the other hand, liquid types were split into 2 categories such as: 'Flat Liquid' and 'Fizzy Liquid'. In order to best capture the variety of foods and liquids, a first approach for the protocol consisted of gathering aliments of an ascending range of calories:

Food	Banana	Orange	Boiled Egg	Slice of Bread	Fruit Jelly Candy
Type	Soft	Soft	Soft	Soft	Soft
Calories	89	47	154	266	324
Quantity (g)	100	100	100	100	100
Food	Strawberry	Grape	Rice	Pasta	Scone
Type	Medium	Medium	Medium	Medium	Medium
Calories	33	67	130	130	353
Quantity (g)	100	100	100	100	100
Food	Carrot Stick	Apple	Chocolate	Nacho Chips	Peanuts
Type	Hard	Hard	Hard	Hard	Hard
Calories	41	52	335	465	567
Quantity (g)	100	100	100	100	100
Food	Water	Coffee	Orange Juice	Semi-skimmed milk	Shaken Udder Chocolush Milkshake
Type	Flat Liquid	Flat Liquid	Flat Liquid	Flat Liquid	Flat Liquid
Calories	0	26	156	170	244
Quantity (mL)	330	330	330	330	330
Food	Perrier	Sprite	Schweppes Lemon	7UP	Coke
Type	Fizzy Liquid	Fizzy Liquid	Fizzy Liquid	Fizzy Liquid	Fizzy Liquid
Calories	0	46	86	135	139
Quantity (mL)	330	330	330	330	330

Figure 7: Table of all the selected foods and calorie breakdown

This protocol required the subject to undergo 5 rounds, sitting down in a relaxed position, while focusing on a fixed point to remove head swing motion artefacts, and chew or swallow for 30 seconds followed by 30 seconds of rest. The table below illustrates the patterns followed by the subject when recording the data:

Round 1	Egg	rest	Pasta	rest	Carrot	rest	Water	rest	Sparkling Water	rest
Round 2	Bread	rest	Rice	rest	Doritos	rest	Coffee	rest	Redbull	rest
Round 3	Banana	rest	Grape	rest	Nuts	rest	Juice	rest	Lemonade	rest
Round 4	Orange	rest	Blueberry	rest	Apple	rest	Milk	rest	7UP	rest
Round 5	Gummies	rest	Scone	rest	Chocolate	rest	Choco Milkshake	rest	Coke	rest

Table 1: Round Breakdown of Protocol 1

The idea behind this protocol relied on obtaining a range of different types of foods and drinks as well as calories to estimate from. Following this protocol, 5 recording sessions were performed. Although timestamps were carefully denoted to be of 30 seconds for each activity, in practice, each chewing/swallowing segment more or less varied due to the unpredictable nature of the chewing process when ingesting food. As a result, an average over the whole recording for each activity was taken for segmentation purposes. After further analysis of the obtained results, the identification of chewing patterns proved to be difficult. The selected foods demonstrated no apparent patterns within the same category rendering the differentiation process complex to analyse and evaluate. In addition, the 30-second segments of continuous chewing or swallowing made the artefacts challenging to accurately extract, segment, and label thus negatively affecting the performance of classifying models.

To overcome the complexity of the data, the protocol was adapted to better focus on capturing individual artefacts and increase the likeliness of identifying patterns between foods of different types. Instead of taking a variety of foods, one food per category was selected and bites were standardized such that clearer segments can be extracted.

Round 1	1 Banana Bite	rest (30 sec)	x15
Round 2	1 Grape	rest (30 sec)	x15
Round 3	1 Carrot Bite	rest (30 sec)	x15
Round 4	1 sip of Water	rest (30 sec)	x15
Round 5	1 sip of Coke	rest (30 sec)	x15

Table 2: Round Breakdown of Protocol 2

Similarly to the previous protocol, the subject undergoes 5 rounds sitting down in a relaxed position while focusing on a fixed point to remove head swing motion artefacts. They take one bite, chew and rest 30 seconds and repeat this process 15 times per round.
This recording method demonstrated much clearly the differences between artefacts obtained from different types of foods, visual differences could already be inferred making the data favorable for classification purposes. To ensure the regularity of the data, all recordings were performed on one test subject such that the trained machine learning algorithms would focus on identifying the subject's chewing and swallowing patterns. A total of 8 food recordings for each food type and 12 drink recordings for each drink type were performed. The additional 4 drink recordings were included to mitigate the data imbalance caused by the presence of 3 food classes to 2 drinking classes.

2.3 Signal Pre-processing

Once the data has been recorded, pre-processing is required before feeding these inputs into an ML algorithm. The recordings are filtered and normalized such that they are less scale variant and more homogeneous before being segmented and labeled into their respective categories.

As mentioned in the Background section, all relevant frequencies regarding EEG signals are present between 0.5-30Hz. Thus, considering a passband filter amongst the Butterworth, Chebyshev Type I and Type II filter seems the most reasonable option. When closely observing the response of these filters, the Butterworth Filter has the smoothest roll-off response in the stop-band, the Chebyshev Filter Type I has the quickest roll-off but the pass-band is affected by ripples, and finally, the Chebyshev Filter Type II has a slower roll-off response than the Type I but no ripples affect the pass-band. Furthermore, the roll-off of the Chebyshev Type II is quicker than the Butterworth thus making the Chebyshev filter Type II a better choice. After trying different filter orders, the 5th order gave the best pre-processing results on EEG and Mic data. The Figure below illustrates the responses of these filters offering visual comparison.

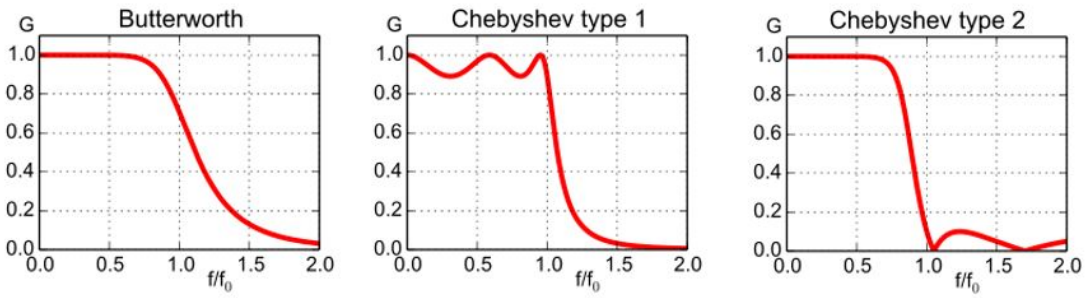


Figure 8: Comparison between different Linear Filters [19]

Following the signal's filtering, normalization is applied to ensure they are scaled to identical levels making the algorithm less scale sensitive when training/testing on the data. The L2 norm, also referred to as Euclidean Normalization, which consists of the square root of the sum of the squares of each value is applied. The sklearn pre-processing library was used which applies the following math to pre-process: $x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$

Once the data is filtered and normalized, segmentation is required to separate the different resting, chewing, and swallowing segments. An automatic segmentation process was implemented which includes a thresholding function taking in the absolute value of an inputted signal. If the defined threshold is exceeded, a flag is triggered which takes the next 5500 or 1000 samples depending on whether the concerned segment is a chewing or swallowing artefact. The thresholds for food and liquid segments were defined after closely studying a specific recording and adapted when necessary. The remaining non-flagged segments were separately stored and constituted the rest segments. Around 10% of extracted segments were incorrectly thresholded; these could have been detected due to noise induced by the hardware or the subject during rest time. These were manually removed. For further specificity, for ingested food recordings, if signals breached the threshold of 0.0015, the flag was triggered. A segmentation example can be observed for Soft, Medium, and Hard foods in the figure down below:

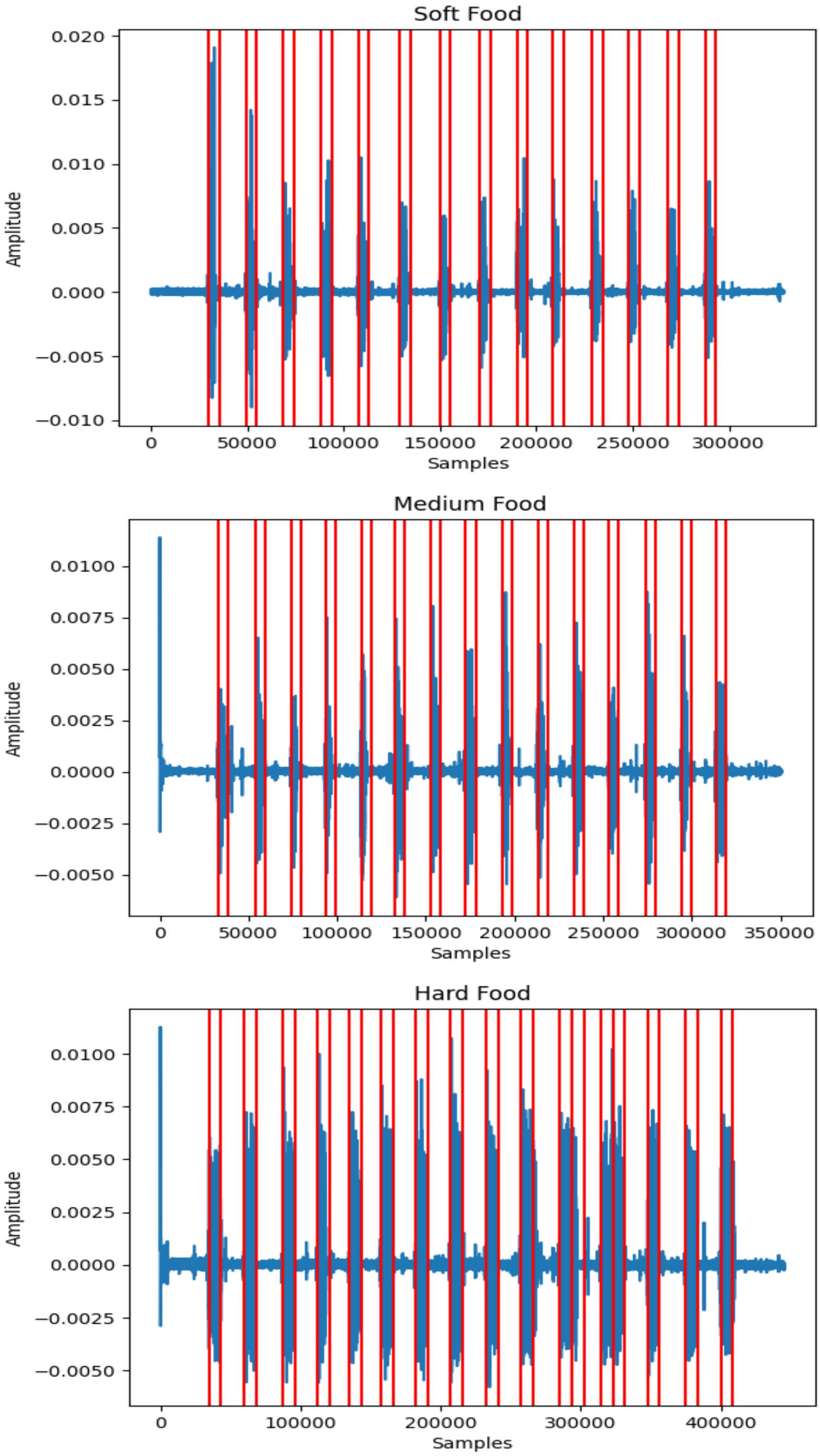


Figure 9: Segmentation of Signals

Visually, it can be observed that the Hard Food data is quite distinguishable to the other two classes. The length of the chews are longer and at a higher frequency while the other two classes have similar amplitudes, length and chewing frequencies within a bite.

When segmenting swallowing artefacts, the defined threshold has been defined higher than the average observed peak value. It has been noted that in rest state, subjects naturally swallow their saliva which creates swallowing artefacts within rest segments thus resulting in confusions between the swallowing and resting segments. To help mitigate this confusion as best as possible, only swallowing artefacts above the average amplitude of rest segments were extracted, which consists of artefacts above the threshold of 0.007. Below a comparison between Timestamp Segmentation and Threshold Segmentation can be observed for both Flat and Fizzy Drinks. Around every dotted line, the subject took a sip of their drink. For better visualization purposes, a close up of the drink segmentation process has been inserted below :

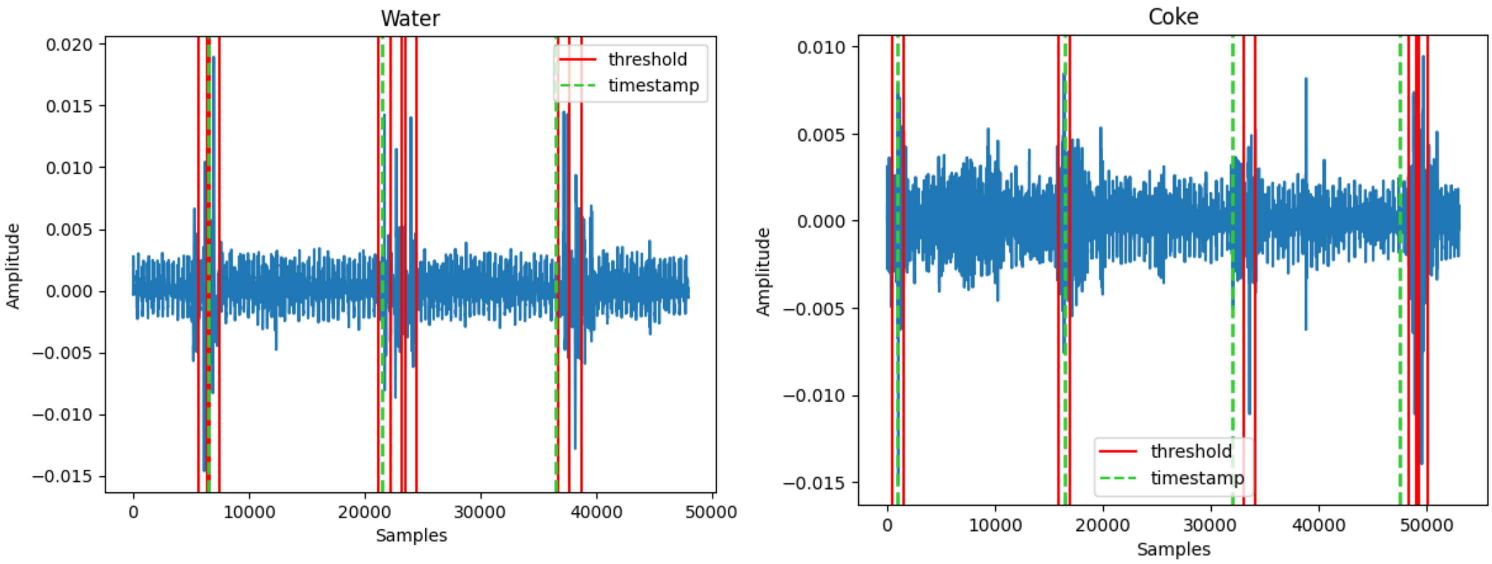


Figure 10: Close up of Segmentation and Extraction of Drink Segments

2.4 Feature Extraction

As each segment has between 1000 to 8000 samples, it is difficult to efficiently train models on such large data inputs. Feature selection enables the identification of key features that allows algorithms to distinguish and identify different categories. Signals have features that can be divided into two types [20]:

- Time-Domain Analysis which provides information on how a signal, X, changes over time.
- Frequency-Domain Analysis which demonstrates how much a signal's energy is distributed over a range of frequencies. The Fast Fourier Transform (FFT) was applied to obtain the Power Spectrum Density (PSD), $S(f)_i$, which is the square of each FFT value divided by 2 times the frequency spacing on the x-axis.

Table 3: Table of Features

Feature	**Acronym**	**Formula**
Minimum	MIN	$\min(X)$
Maximum	MAX	$\max(X)$
Mean	MEAN	$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$
Root Mean Square	RMS	$\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Variance	VAR	$\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{X})^2)$
Standard Deviation	STD	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{X})^2}$
Power	POWER	$\frac{1}{N}\sum_{i=1}^{N}(x_i^2)$
Peak	PEAK	$P_m = \max(\lvert x_i \rvert)$
Peak to Peak	P2P	$P_k = \max(\lvert x_i \rvert) - \min(\lvert x_i \rvert)$
Crest Factor	CRESFT	$\frac{P_m}{RMS}$
Skew	SKEW	$\frac{\sum_{i=1}^{N}(x_i - \bar{X})^3}{(N-1)(N-2)\sigma^3}$
Kurtosis	KURT	$\frac{1}{N}\sum_{i=1}^{N}(\frac{x_i - \bar{(X)}}{\sigma})^4$
Max Frenquency	MAX_f	$\max(S(f)_i)$
Sum of total PSD	SUM_f	$\sum_{i=1}^{N} S(f)_i$
PSD Mean	MEAN_f	$S_\mu = \frac{1}{N}\sum_{i=1}^{N} S(f)_i$
PSD Variance	VAR_f	$\frac{1}{N-1}\sum_{i=1}^{N}(S(f)_i - S_\mu)^2)$
Peak of PSD	PEAK_f	$\max(\lvert S(f)_i \rvert)$
Skew of PSD	SKEW_f	$\frac{\sum_{i=1}^{N}(S(f)_i - S_\mu)^3}{S_V^{3/2}}$
Kurtosis of PSD	KURT_f	$\frac{1}{N}\frac{\sum_{i=1}^{N}(S(f)_i - S_\mu)^4}{S_V^{4/2}}$
Note: N = length(X)		

To avoid biasing the model, only features from the training dataset were considered for training.

2.5 Machine Learning Algorithms

The final aim of this project focuses on classifying recordings into their respective categories. To achieve this, two main approaches were considered :

1. Machine Learning Classifiers

2. Deep Neural Network or Convolutional Neural Net.

2.5.1 Model Selection

In order to decide which approach to focus on, simple models were implemented for both options and were cross-compared. As all recordings were taken from the same subject, the dataset was split into a randomised 70/30 training-to-testing ratio. The approach focusing on Machine Learning Classifiers was first considered. The problem was simplified to a Classifier which differentiates between Rest and Activity signals. Using the One-Way Analysis of Variance (ANOVA) test, the most significant features were selected and the following machine learning classifiers, introduced in the Background Section, were considered:

- Logistic Regression (LR)
- Decision Tree (DT)
- Random Forest (RF)
- Support Vector Machine (SVM)
- K-Nearest Neighbours (KNN)
- Gaussian Naive Bayes (GNB)
- Discriminant Analysis (LDA)

The highest-performing classifier reached a maximum of 98% and its testing accuracy reached as high as 88%.

On the other hand, a simple Multi Layer Perceptron Neural Network method was implemented. Further information on the NN's architecture can be found in the Appendix (C). Without any additional parameter tuning or architectural improvements, by inputting the 19 computed features, the Neural Net achieved 73% training accuracy and 56% testing accuracy. The model's Learning Curves demonstrating promising results can be observed on the graph below:

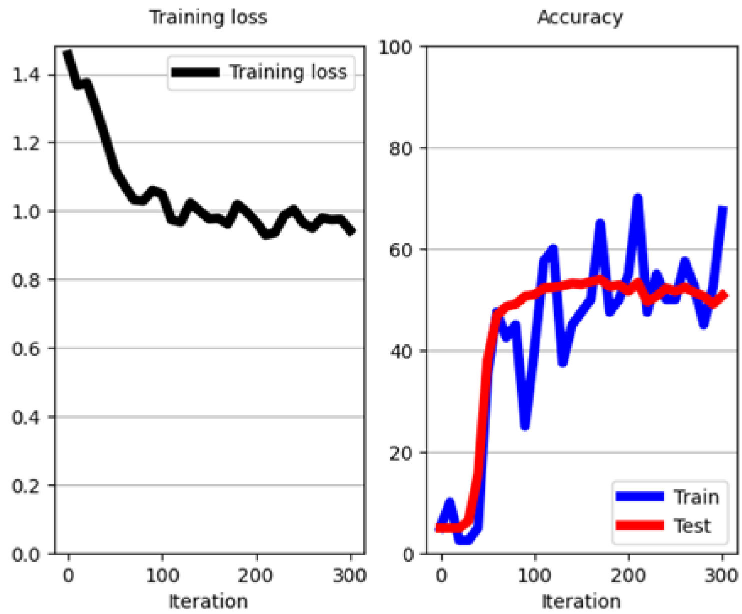


Figure 11: MLP Learning Curves during model training

After comparing the obtained results, focusing on tuning and improving the Neural Nets was deemed over-engineering the problem leading to its development put on hold. Considering they require large datasets to train and enhance their performance, this project requires time and resources to obtain recordings which makes data sparse and limited. On the other hand, classifiers are better at dealing with smaller datasets, problems that can be solved using linear boundaries, and are interpretable and explainable models. Classifiers' interpretability and explainability make them straightforward to debug and follow their algorithm's learning process. Additionally, the data worked with provides well-defined features in space which does not require complex black box models where it is difficult to interpret and understand the way the algorithm selects features to make decisions. Finally, classifiers have rapid training time which is beneficial for debugging purposes.

The implementation of the simple Classifier spurted the idea of creating an architecture of Classifiers in series where each Classifier specializes in the classification of certain classes. They would all train on the same training data but different labels such that:

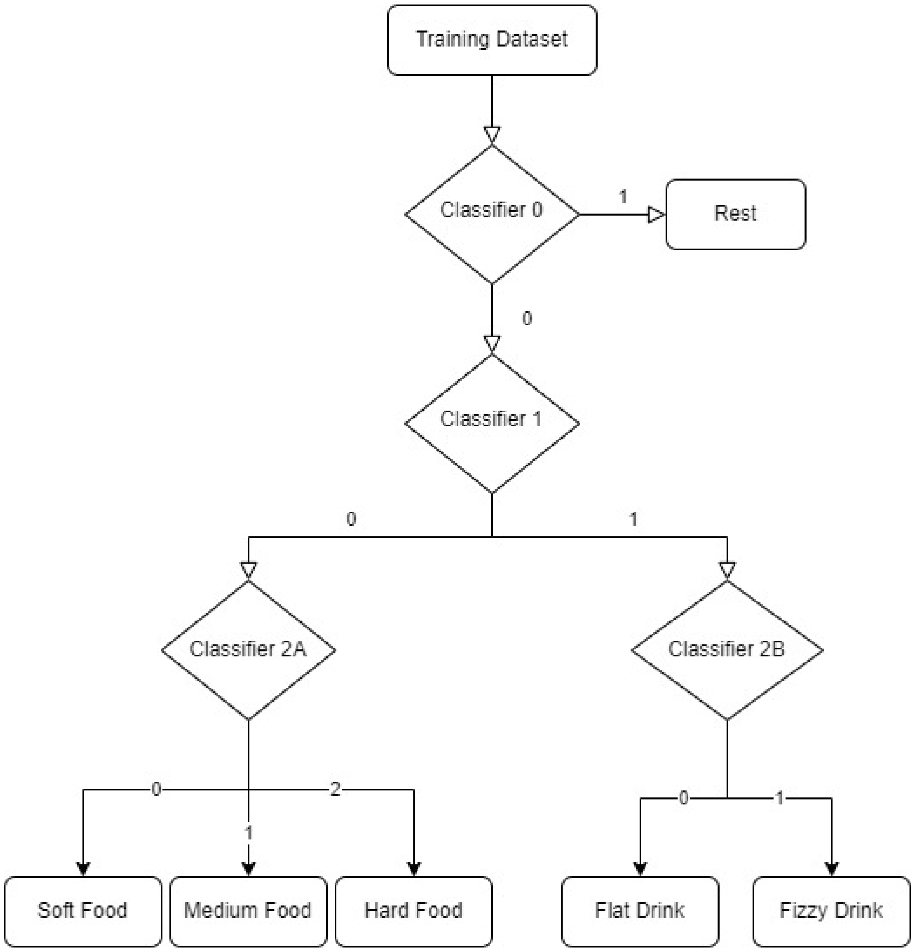
1. Classifier 0: specialized in classifying segments which contain Activity (chewing/swallowing) and Rest. This classifier tests on the testing dataset.

2. Classifier 1: specializes in classifying segments of Chewing and Swallowing. This classifier tests on predicted outputs of Classifier 0 labeled as Activity.

3. Classifier 2A: specializes in classifying between Soft, Medium and Hard Food. This classifier takes only the test outputs of Classifier 1 labeled as Chewing and predicts the type of food it is.

4. Classifier 2B: specializes in classifying between Flat and Fizzy Drinks. This classifier takes only the test outputs of Classifier 1 labeled as swallowing and predicts the type of drink it is.

The figure below illustrates this architecture:



Class	Associated Tuple
Soft Food	(0,0,0)
Medium Food	(0,0,1)
Hard Food	(0,0,2)
Flat Drink	(0,1,0)
Fizzy Drink	(0,1,1)
Rest	(1,2,3)

Figure 12: Depiction of the Classifiers in Series and its associated tuples

This chain of classifiers operates like a Decision Tree where each 'leaf node' represents a final prediction as to what food or drink type is inputted. As previously mentioned, the ANOVA test is used to select the inputted features. As the type of data worked with has numerical input variables and categorical outputs, the use of the ANOVA test to identify the most relevant features is the most coherent. The test compares the means of at least two groups and determines the differences between them.

In this context, all 19 features of two groups are inputted and the ones which differentiate the most the classes are returned. When evaluating, mainly the f-values will be discussed and the p-values will be referred to in specific cases. Out of the 19 computed features only the top $\boldsymbol{k}$ are kept. How the features were selected for each classifier and how they distinguish the concerned classes is further discussed in the next section.

2.5.2 Model's Feature Selection

To begin with, for Classifier0, the training data is split into two classes: 'Activity' and 'Rest' which are then inputted into the ANOVA-test. The parameter $\boldsymbol{k}$ representing the number of selected features is tuned to obtain the highest performing Classifier0. After trial and error, $\boldsymbol{k}$ = 7 resulted in the best performance. Below is an example of F-values returned:

Table 4: Activity Vs Rest

Feature	F-value	Feature	F-value
RMS	1315.47	**PEAK**	1296.03
STD	1296.03	**MIN**	926.58
MAX	1195.80	**SUM_f**	308.9
P2P	1315.47		

The obtained f-values are on average remarkably high, which is an indicator that the selected features highly differentiate the two classes. When further breaking down the selection of features, it can be reasoned why they would be so significant. The Root Mean Square, a measure which takes the average magnitude of the inputted segment, provides information on the overall magnitude and amplitude of the data. Rest signals are on average uniform in comparison to Chewing and Swallowing which are likely to have peaks of varying amplitudes. This also explains why standard deviation, which provides insight into how disperse and variable the values within a signal are, how much they deviate from the mean, is a distinguishable feature between the classes. Maximum, minimum, peak-to-peak, and peak are all indicators of the signal's amplitude over time that can be observed when an example of a Rest and Activity signal are put side by side. Classifier0 is thus expected to classify with high accuracy.

Moving onto Classifier1, the training data is split into 'Swallowing' and 'Chewing' and is inputted into the ANOVA-test. The parameter $\boldsymbol{k}$ is tuned through trial and error, which results in $\boldsymbol{k}$ = 6 for best performance. When comparing Swallowing and Chewing artefacts, the obtained f-values are comparatively high. Although they do not exceed 1000, they are sufficiently high to infer that the difference between the signals are pronounced. The table below demonstrates an example of returned f-values when comparing Chewing and Swallowing. The Crest Factor is an indicator of the difference between the peak amplitude and the average amplitude of the signal, chewing tends to have varying amplitudes due to the mechanical jaw movement when ingesting food which would explain a higher peak than the average amplitude compare to swallowing liquids where no mechanical movement movements are required. The

RMS, STD and MIN can be found again which are traits that can be observed when visually comparing the two signals, since a singular sip of water takes significantly less time than chewing on food, it is expected that chewing artefacts have higher RMS and STD values while swallowing artefacts have similar MIN values in comparison to the variety of chewing artefacts. Thus Classifier1 is expected to accurately perform.

Table 5: Chewing vs Swallowing

Feature	F-value	Feature	F-value
CRESFT	440.69	**KURTOSIS**	115.52
RMS	142.41	**KURTOSIS_f**	115.52
STD	133.07	**MIN**	66.50

From the point mentioned above, it is clear that by nature the act of resting, chewing and swallowing are highly different. Distinguishing between the food types and drink types proved to be more challenging as the f-values revealed that by nature certain classes are similar. The Table [6] below, illustrates a breakdown of the types of food compared two by two as well as an overall comparison between the 3 classes.

Table 6: Comparison of Food types

Soft vs Medium Soft Food

Feature	F-value	Feature	F-value
MAX_f	12.64	**RMS**	8.76
PEAK_f	12.64	**SUM_f**	6.59
STD	8.78	**POWER**	6.54

Soft vs Hard Food

Feature	F-value	Feature	F-value
SUM_f	162.00	**MAX**	75.14
PEAK	81.25	**MAX_f**	73.02
P2P	76.18	**PEAK_f**	73.02

Medium Soft vs Hard Food

Feature	F-value	Feature	F-value
SUM_f	141.54	**SKEW**	57.61
CRESFT	123.58	**SKEW_f**	57.61
PEAK	62.83	**MAX**	57.59

Overall Comparison of Food Types

Feature	F-value	Feature	F-value
KURT	145.07	**STD**	65.63
KURT_f	66.07	**MIN**	56.96
CRESFT	66.07	**RMS**	54.85

For Classifier2A, the optimal choice of $\boldsymbol{k}$ has been identified as $\boldsymbol{k} = 6$. When only looking at the overall comparison amongst the type of foods, it appears that the f-values are sufficiently high to have features that differentiate efficiently the types of food. Nevertheless, when looking closer, it can be observed that when compared to Soft and Medium Soft Food signals, the Hard Food category is evidently distinguishable. The soft and medium soft foods have fairly low f-values indicating that the inputted features aren't significant enough to distinguish them. It can be observed that overall the sum of total PSD distinguished the most the Hard Food class from the other two. Additionally, since chewing on hard food produces the highest amplitudes due to the high impact between the teeth and the solid food, metrics related to amplitude such as PEAK or Crest Factor additionally significantly differentiate the hard food class to the other two. According to the obtained results, it can be expected that the Hard Food class would have less difficulty in being accurately classified in comparison to the Soft and Medium Soft food category which are highly likely to get interchangeably confused.

Finally, when observing the ANOVA test results obtained for Classifier2B by comparing Fizzy and Flat Drinks, all f-values do not exceed 2. In addition, all previously identified p-values conformed to the rule of thumb of values below 0.05 but in this case the p-values range between 0.099 to 0.246. Thus highlighting how difficult the distinction between both classes is. An example of obtained f-values for Fizzy and Flat Drink can be observed below:

Table 7: Fizzy vs Flat Drinks

Feature	F-value	Feature	F-value
SKEW	1.89	**PEAK_f**	1.22
MAX_f	1.89	**SKEW_f**	1.22
VAR_f	1.34		

Thus, by evaluating selected features for each classifier, expectations can be set on the performance and potential limitations that could cap the accuracy of the models can be identified. Although f-values for Classifier 0, 1 and 2A demonstrate encouraging prospects, considering our available data, Classifier 2B is likely to have limited performance because the calculated features do not distinguish enough the fizzy and flat drink signal's. A potential solution to this limitation is further discussed in the section Future Works.

2.5.3 Model Training Method

To train the model, optimising methods were considered to maximise the performances of the classifiers. Indeed, each model gets hyperparameters tuned using the Gridsearch library which iterates through a range of defined values to find the optimal combination. For example, for KNN the best k neighbour is searched, for Decision Trees the optimal tree depth is searched or for Random Forest the optimal number of trees within the ensemble is tuned. Gridsearch will train the models with each parameter, compare and returns the best performing one. Below is a table containing all the defined hyperparameters per models, as introduced in the Background section, for each ML model:

Table 8: Tuned Parameters

Classifier	Tuned Parameter	Classifier	Tuned Parameter
LR	C:[0.1,1,10]	**SVM**	C:[0.1,1,10]
DT	max_depth:[5,10,None]		kernel:[linear,rbf,sigmoid]
RF	n_estimators : [5,10,50,100]	**GNB**	None
KNN	n_neighbors: range(1,45)	**LDA**	solver:[svd,lsqr,eigen]

Following the identification of the optimal model, a 5-fold Cross Validation is applied to training data which splits the dataset into 5 folds and alternates between each fold keeping each time one as a validation set and the rest as regular training data. Only the best-performing model is kept and evaluated with its confusion matrix outputted.

3 Results and Discussions

3.1 Dataset Evaluation

Building a complete, accurate, and consistent dataset before evaluating the model's performance is an important element which highly influences how well a Machine Learning Model performs. A dataset with features that are representative of their classes greatly helps classifiers make accurate predictions and reduces their confusion. As previously discussed, each classifier was tuned such that only the most significant features for its classified classes were considered. Further evaluation was required to determine whether these used features were indeed the most optimal and whether there was a relationship to establish between the segmentation length and the significance of the features.

3.1.1 Segmentation Evaluation

Features extracted from pre-processed signals of various lengths were evaluated using the ANOVA test. These were labeled into the classes Food, Drink and Rest which were compared two by two. Thus investigating on the relationship between segmentation and the significance of extracted features for the built dataset. Regarding food signals, different lengths such as full length (5500 samples), undersampled (2500 samples) and oversampled (7000 samples) segments were tested. Similarly, for drinking signals, full length (1000 samples), undersampled (500 samples) and oversampled (2000samples) segments were tested.

In the case of food signals, full-length and under-sampled signals returned similar feature choices and f-values when performing the ANOVA test. On the contrary, over-sampled food segments made the distinction between the rest class difficult for classifiers, thus negatively affecting their performance. It was observed that the Hard class is the most evident to distinguish among the 3 food classes while the Soft and Medium food have similar features making it more difficult to distinguish them. Including segments of different lengths was also attempted to observe if it could result in more distinguishable features. The final method included both segments of full-length and further segmented signals which provided better generalization and thus contextual information to capture both short-term and long-term patterns for a more comprehensive understanding of the classes.

Similarly to the food signals, oversampling drinking signals created more confusion and increased segments resemblance to rest signals while the full-length and further segmented resulted in similar feature choices and f-values. Distinguishing between Fizzy and Flat drinks has proven even more difficult than the food types as the f-values are small inferring that the available features struggle to distinguish the two classes. Since the f-values were nearly identical, all drinking segments were fully segmented into 500 sample segments to increase the amount of available data for the Swallowing class. Thus, the dataset was robustly compiled and evaluated such that a reliable model's performance evaluation can be performed.

It was concluded that oversampling would reduce the significance of the features while undersampling doesn't affect their significance, meaning we could increase the number of available data within our dataset without augmenting the chances of confusing the Classifiers. The final outlook of each segment per class can be further visuallised in the Appendix B.

3.2 Classification Evaluation

To perform a complete evaluation of the used classifiers, 2 methods have been implemented: an Individual evaluation of each Classifier, and an Evaluation of the Classifiers in a pipeline which considers misclassified elements. The first method provides insights on how the classifiers perform on their own, if they recognize patterns and classify accurately the labels they have been trained to classify. The second method provides an assessment that more closely reflects the real-world application, how the architecture deals with outliers can be evaluated. Both methods rely on confusion matrices to evaluate the performance. They provide insights on metrics such as accuracy, precision, recall and f1-score.[21] Each metric provides information on the following:

- Accuracy: number of correct predictions overall predictions
- Precision: how many of the positive predictions made are true positives
- Recall: how many positive cases the classifier correctly predicts over all the positive cases in the data
- F1-score: harmonic mean of precision and recall, $\frac{2*Precision*Recall}{Precision+Recall}$

Throughout the evaluation, the F1-Score will be the main metric used to evaluate performance. Indeed the F1-score considers both the classifier's Precision and Recall for each class. It is more representative of how well the class is identified by the model as the accuracy tends to be biased towards the highest performing class. If the model achieves high accuracy by simply predicting the majority class it is not precise enough to evaluate the performance. As a result, the F1-score for the classification of each class will be discussed.

3.2.1 Classifier's Individual Evaluation

To individually evaluate each classifier, they will each use an adapted version of the testing dataset where the labels are to exactly match the labels each classifier has been trained with.

- Classifier 0: uses the testing dataset with the first label defined as 0 or 1.
- Classifier 1: uses the testing dataset with the second label defined as 0 or 1.
- Classifier 2A: uses the testing dataset where the second label is defined as 0 and the third one as 0, 1 or 2.
- Classifier 2B: uses the testing dataset where the second label is defined as 1 and the third one as 0 or 1.

This first step enables the evaluation of the classifiers on their own to understand their strengths and weaknesses. At this stage, they can be tuned and optimized so the highest f1-scoring models are kept for the pipeline. To ensure the displayed F1-score values are representative of the performance, the evaluations performed below are averages of F1-scores over 5 runs. The purpose of Classifier0 focuses on classifying between Activity and Rest, most specifically ensuring that the classification of the class Activity is optimal. The following classifiers would in theory take the predicted values of that class.

Both ratios of 70-30 and 80-20 training to testing have been tested on the classifiers before collecting results. It was noted that the 70-30 split resulted in better performance for Classifier 0 and Classifier 1 but the split 80-20 resulted in a 2-4% performance decrease for Classifiers 0 and 1 and a 10% increase in the classification of Hard Food for Classifier 2A. As could be observed in the feature evaluation, when comparing classes certain have features that distinctly differentiate them from others while others have under-representative features. Increasing the training data could allow models to better learn about the Hard Food Class and thus better classify it than for a 70-30 split. Since the final aim of this project focuses on classifying the food types, working with the 80-20 split was considered more relevant.

To further consider the feature significance imbalance a comparison between models with and without weighted classes has been performed. Most selected classifiers have a weight_class parameter except from KNN, Gaussian Nayve Bayes and LDA. Although, KNN has a similar parameter performing a 'uniform' nearest neighbour meaning that all points in each neighborhood are weighted equally or a 'distance' search which means that closer neighbors of a new instance point have a greater influence on the prediction than neighbors distant neighbours. The figure below demonstrates the F1-Score obtained for each model on the testing dataset.

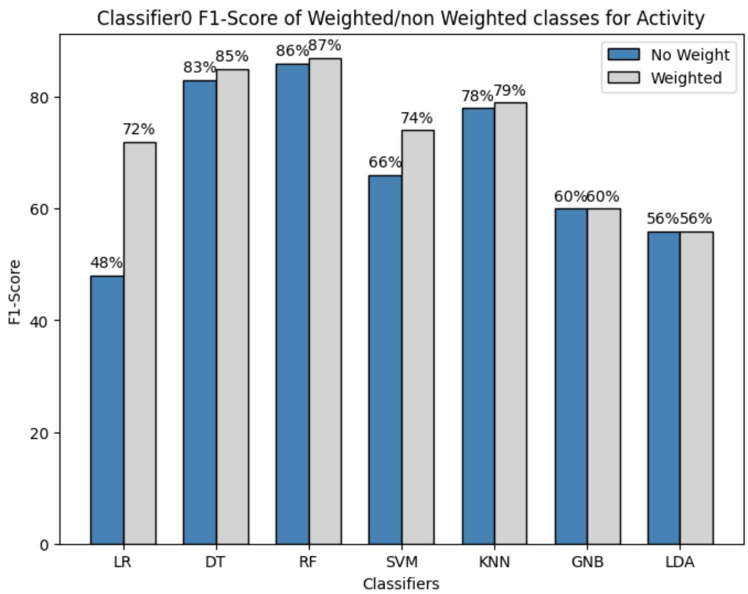
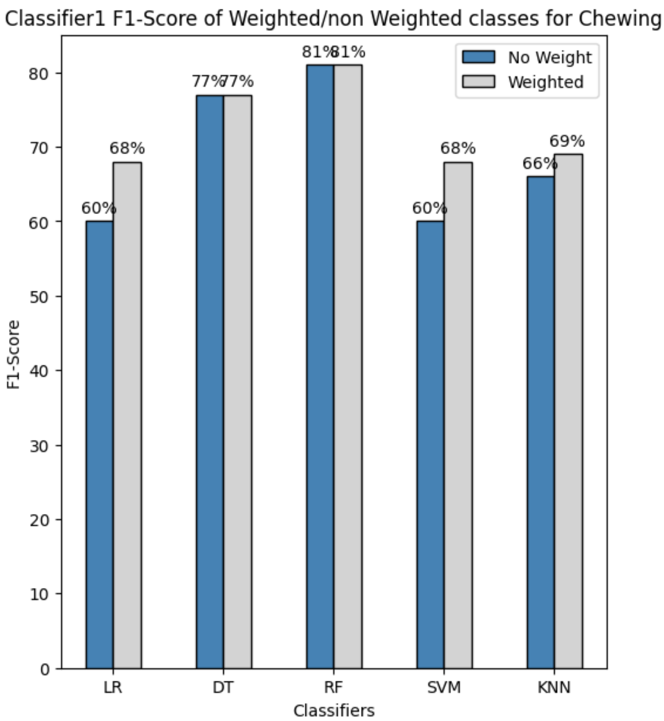


Figure 13: Classifier0 Individual Performance Evaluation

It can be observed that the best-performing model is the Random Forest with an F1-score of 87%. Focusing on algorithms that performed with above 70% F1-score, the underlying algorithms behind the models can explain why such results can be observed. Decision trees make splits based on feature importance at each node, during training they will pick up on signals that have less significant features and create a branch. They can capture relationships between features and the target variable even when the significance is low. A Random Forest is an ensemble of trees with a majority voting system which explains why it exhibits slightly better performance than the DT. On the other hand, KNN performs classification based on the majority class of the nearest neighbors, if certain features have larger significance it leads to a biased influence when determining new neighbors. Similarly, Logistic Regression and Support Vector Machines rely on optimizing a global decision boundary when classifying which makes them susceptible to feature significance imbalance. If certain features dominate the optimization process, the models will make biased predictions which decreases the performance. This explains why such an increase in performance can be observed for LR and SVM when considering class weights or non-weighted classes.

Considering that out of the 5 classifiers, this first classifier has its training dataset the least affected by feature significance imbalance, it was apparent that models performing less than 70% may not provide reliable results. As a result, the next classifiers were trained exclusively considering Logistic Regression, Decision Trees, Random Forests, SVMs and KNNs.

The figures below depict the F1-score of Classifier1. In a similar style as for Classifier 0, models with class weight and without class weighting were evaluated.



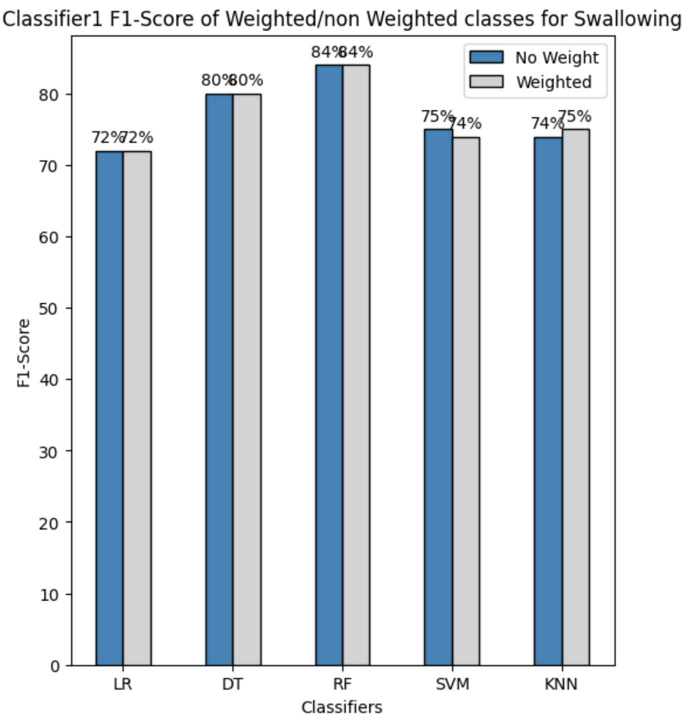


Figure 14: Classifier1 Individual Performance Evaluation

Once again, it can be observed that the Random Forest model performs the best, with an F1-

Score of 81% for the Chewing Class and 84% for the Swallowing Class. The class weighting does not significantly impact the performance of Classifier 1 although a neat improvement can be observed for the Logistic Regression model. This suggests that there is no imbalance in the way features represent each class. Given the Classifier 0 classifies Activity signals with an accuracy of 87%, the overall accuracy would in theory be of 70.47% for Chewing ($0.87 \times 0.81 = 0.7047$) and of 73.08% for Swallowing.

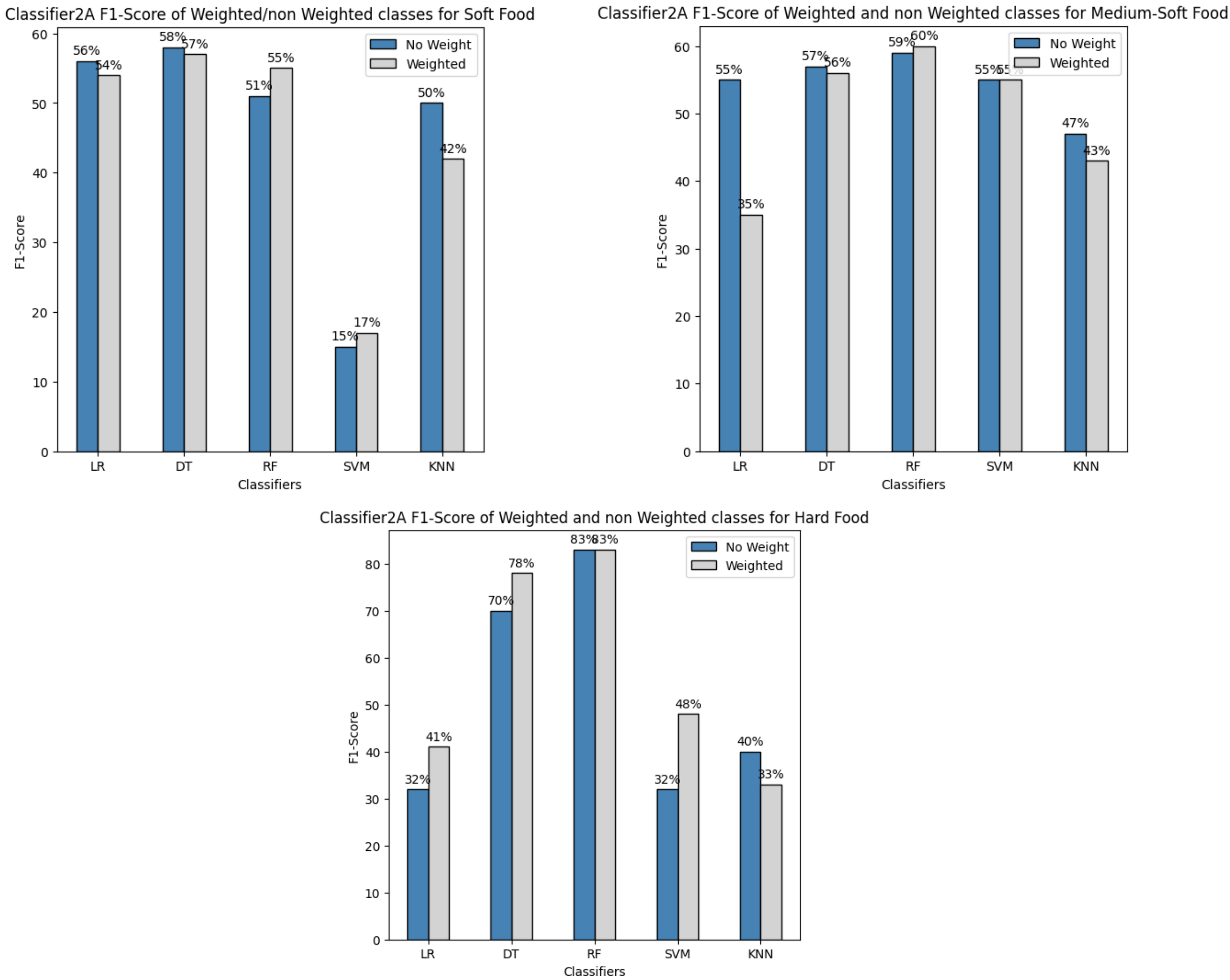


Figure 15: Classifier2A Individual Performance Evaluation

As the feature selection f-values suggested, Classifier2A's overall performance is significantly lower than the two previous ones. The models struggle to distinguish between the Soft and Medium-Soft Food cathegory. Decision Trees and Random Forest still manage to perform best. Random Forest classifies Soft-Foods with a F1-score of 55%, Medium-Soft Foods at 60% and Hard Foods at 83%.

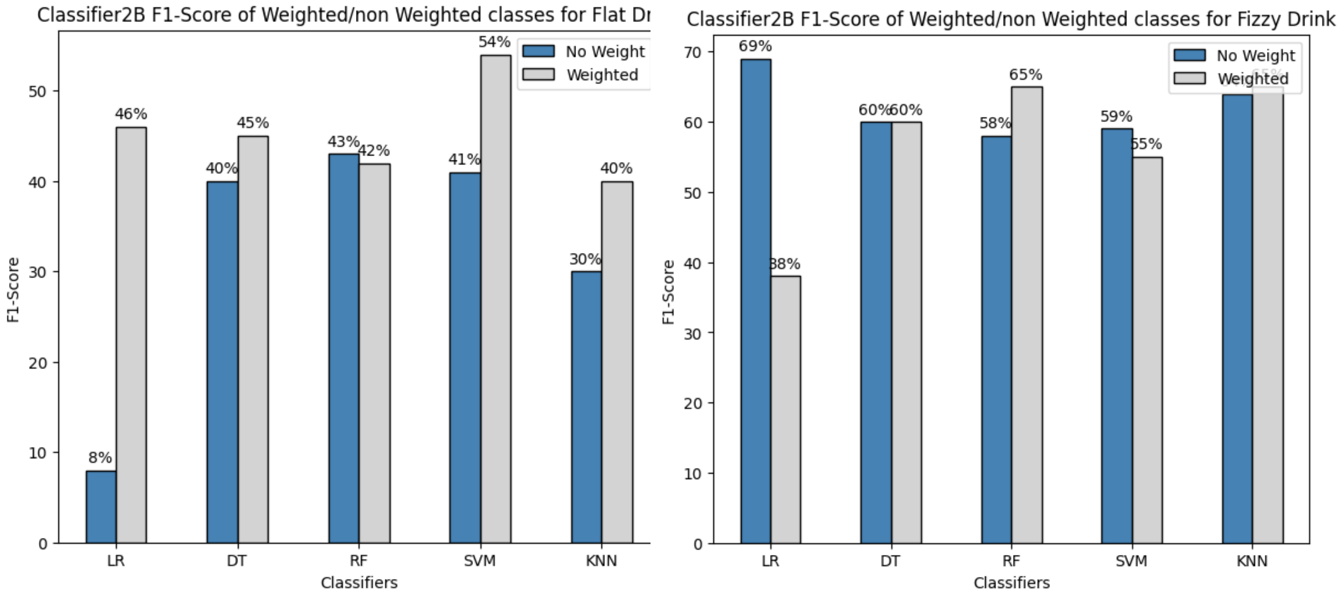


Figure 16: Classifier2B Individual Performance Evaluation

It can be observed that the models struggle to classify between Fizzy and Flat drinks, which is likely linked to the under-representative features which do not distinguish enough the two classes.

3.2.2 Classifier's Pipeline Evaluation

For the evaluation of the Classifiers as a pipeline, each classifier uses as testing data predicted outputs from the previous classifier. Only Classifier0 is directly fed the testing data with the first label defined as 0 or 1. As mentioned above, this evaluation reflects more closely the real-world application; how it is capable of classifying a signal into a food type and how the architecture deals with outliers can be evaluated.

To construct the best-performing architecture, as each algorithm specializes in different styles of classifications, mixing them and evaluating their performance could result in a better model than selecting a single algorithm to perform all stages of the classification. The testing process can be broken down in the following ways:

- Classifier 0: uses the testing dataset with the first label defined as 0 or 1.
- Classifier 1: uses prediceted values that have been classified as label 0 by Classifier 0. Classifier 0 has predicted labels from Logistic Regression, Decision Tree, Random Ensemble, SVM and KNN. Each prediction is taken and tested again on these 5 models tailored with the Classifier 1 selected features
- Classifier 2A: uses the testing dataset where the second label is defined as 0 and the third one as 0, 1 or 2.
- Classifier 2B: uses the testing dataset where the second label is defined as 1 and the third one as 0 or 1.

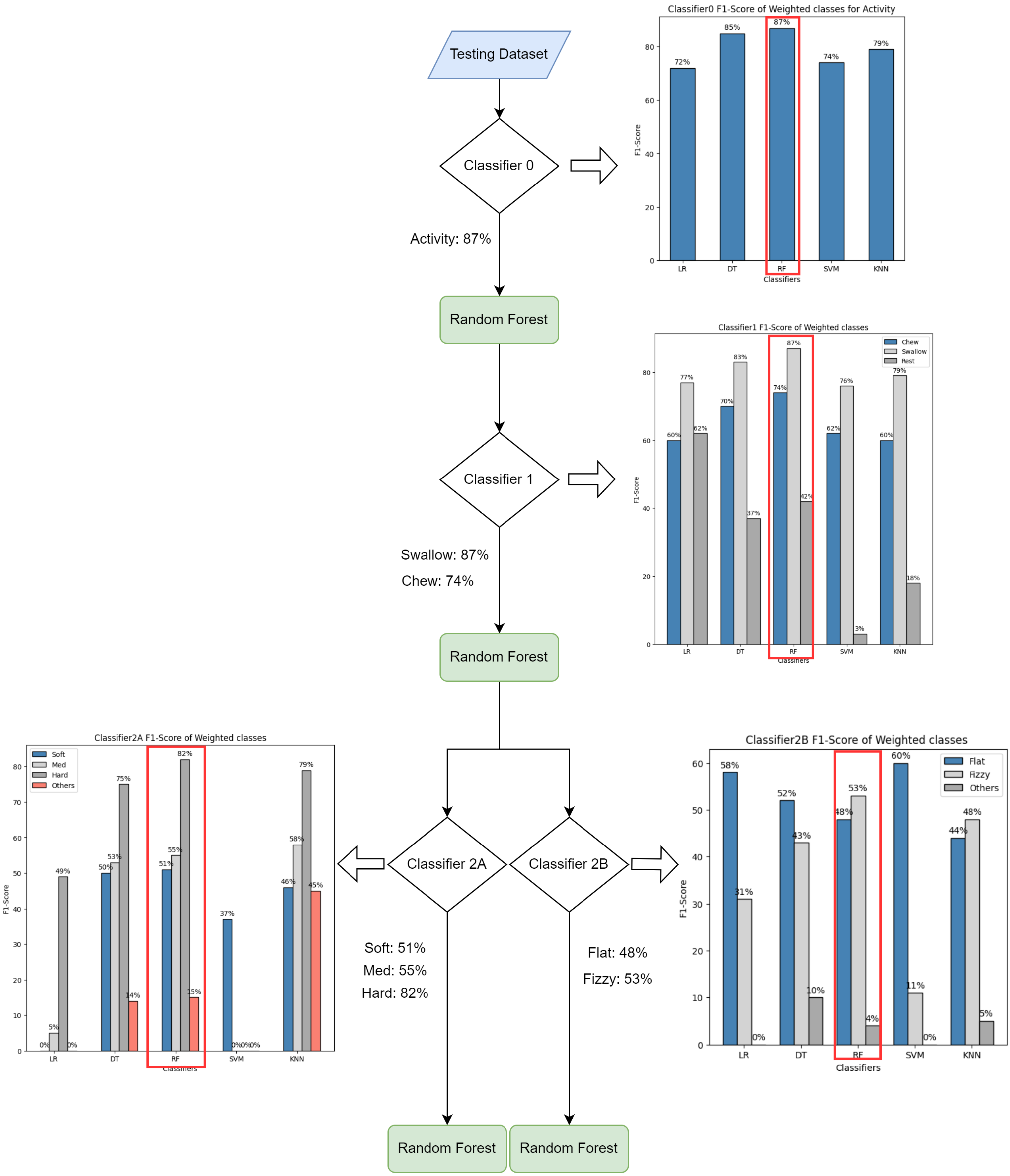


Figure 17: Classifiers Pipeline

The diagram below illustrates how the pipeline was evaluated. Considering the results obtained in the Individual Evaluation of Classifier 0, the GNB and LDA models have been excluded from the pipeline:

In a pipeline of classifiers, it is important to carefully select the model with the best accuracy at each stage. Indeed, the performance of each classifier has a cascading effect on the overall performance of the pipeline. By choosing the model with the highest accuracy, it is ensured that the most accurate predictions are fed into the following classifier. Thus an optimized sequence of algorithms would maximize the potential for an accurate and reliable prediction model.

It can be observed that, as expected from the Individual Evaluation, the Random Forest algorithm outperformed all the other models at each stage of the classification. Random forest excels in situations where there are non-linear relationships and handles best outliers. Their algorithm relying on a majority voting system makes them better for generalization and robustness against outliers. Through the misclassified predictions, Random Forests can identify more subtle features that other models fail to detect. When comparing the performances of the other models in a pipeline in comparison to individually, we can observe an overall decrease in performance which is expected due to the introduced outliers. For Classifier 1, the Chew accuracy decreases of 7% compared to when tested individually. As Classifier 1 takes in the predicted values of Classifier 0 using the Random Forest algorithm, the decrease can be linked to misclassified chewing signals which result in less testing data. On the other hand, Classifier 2A performs similarly to its individually evaluated model, the Hard Food type is highly accurately classified compared to the other two classes. The final obtained f1-scores can be regrouped in the following table:

Table 9: Obtained Results from the Pipeline Architecture

Class	**F1-Score**	**Class**	**F1-Score**
Soft Food	51%	**Flat Drink**	48%
Med-Soft Food	55%	**Fizzy Drink**	53%
Hard Food	82%		

The overall performances could be deducted by counting the number of predicted values per class and dividing it by the total number of values in per class in the testset such that:

$$\text{Accuracy} = \frac{count(y_{pred})}{count(y_{test})} \tag{4}$$

For each run, the accuracies for each food type fluctuated but could be concluded and reduced to the following range: the Soft Food type has an accuracy between 20-25% , the Medium Soft Foods between 30-35% and finally 65-70% for Hard Foods.

3.3 Calorie Estimate

To elaborate on the implemented architecture of Classifiers in series, an approach to estimate calorie intake considered the identified type of food, and associate a caloric density to each type. As classifiers are accessible and have straightforward training processes, it would be feasible to expand the existing architecture to include additionally trained classifiers to identify more foods based on chewing signals. For this project, the calorie intake (CI) method follows the following formula:

$$CI = \text{Bite Number} \times \text{Average Mass per Bite (g)} \times \text{Caloric Density (cal/g)} \tag{5}$$

Each classified food type is associated to the caloric density of the ingested food representative of the class. If a signal is classified as Hard Food the caloric density of a carrot would be used, if it is classified as a Soft Food, the caloric density of a banana would be used. The same concept is applied for all 5 classes. Additionally, the Average Mass per Bite parameter is a tailored parameter that is specific to the subject. During the recording, all bites were standardized such that the subject consumed 1 banana and a half, 15 grapes and 1 carrot and a half for each recording. By referring to the labels of the bought products the total ingested mass per recording could be inferred:

$$M_{banana} = 177g \text{ (118g per banana)} \tag{6}$$

$$M_{grape} = 75g \text{ (5g per grape)} \tag{7}$$

$$M_{carrot} = 120g \text{ (80g per carrot)} \tag{8}$$

$$M_{tot} = M_{banana} + M_{grape} + M_{carrot} = 372g \tag{9}$$

Each type of food required 15 bites which results in a total of 45 bites per recording. Thus, the Average Mass per Bite in this case is :

$$Avg_{g/bite} = \frac{372}{45} = 8.2g/bite \tag{10}$$

Lastly the caloric densities (CD) were calculated according to each product's label: bananas are defined as 85cal for 100g, grapes as 66cal for 100g and carrots as 41 cal for 100g, thus:

$$CD_{banana} = 0.85cal/g \; , CD_{grape} = 0.66cal/g \; , CD_{carrot} = 0.41cal/g \tag{11}$$

3.3.1 Calorie Estimate Evaluation

Considering this method of estimating calories depends on the predicted number of signals within a category, the performance of this calorie estimation is highly dependent on the performance of the pipeline of classifiers. As has been previously observed, the pipeline classifies the Soft Food type with an accuracy contained between 20-25%, 30-35% for Medium Soft Foods and finally 65-70% for Hard Foods.

To evaluate calorie estimation:

1. Count the signals that have been labeled as Soft, Medium, and Hard Food to identify Bite Number for each food type

2. Calculate the calorie intake for each food according to the Formula (5)

3. Compare with the real calorie value by computing the ratio :

$$Ratio = \frac{\text{Estimated CI}}{\text{Real CI}} * 100 \tag{12}$$

An example of obtained results during a run where the model could classify Soft Foods with an accuracy of 23%, Soft foods with an accuracy of 32% and Hard foods at 70% the following estimations were observed:

Table 10: Activity Vs Rest

Food Type	**Estimated Calories (cal)**	**Actual Calories (cal)**	**Ratio**
Soft	299.71	1296.42	23.1
Medium Soft	368.02	1147.34	32.1
Hard	342.92	490.85	69.8

As expected, the calorie estimation ratio reflects the performance of the model. As demonstrates the Hard Food category, including signals that contain artefacts that represent better Soft and Medium Soft food types, this method could predict with higher accuracy the calorie intake.

4 Future Work

4.1 Classifier Approach

From the results discussed and evaluated through this report, it can be observed that further research should be led to identify artefacts that distinguish even further the types of food and liquids. The obtained artefacts did demonstrate that the type of food could be distinguished from artefacts as hard foods can be fairly accurately identified from the rest of the food types. Nevertheless, the current artefacts are not fully sufficient to best distinguish between Soft and Medium foods and Flat and Fizzy Drinks.

As was observed from the feature selection, classes such as Fizzy and Flat Drinks didn't have representative features that could sufficiently distinguish both classes. The Soft and Soft-Medium Food classes were also victim of this issue. On the other hand, the Hard Food features which displayed higher f-values resulted in significantly better-performing classifiers. Considering these results, an approach that could considerably improve the pipeline's performance would consist of including more types of signals such as attaching an extra microphone at the tip of the earplug (as illustrated in the figure below) and/or tape an accelerometer to the subject's jaw. The current position of the microphone located at the top mainly detects the friction between the electrode and the skin which is good to detect physical jaw movement such as chewing, clenching, speaking. Adding the additional microphone at the tip of the earplug would capture sounds produced by the food inside the mouth, within the oral cavity. This could give further information on the ingested food texture, and additional artefacts that cannot be picked up by the microphone on the top of the earplug.

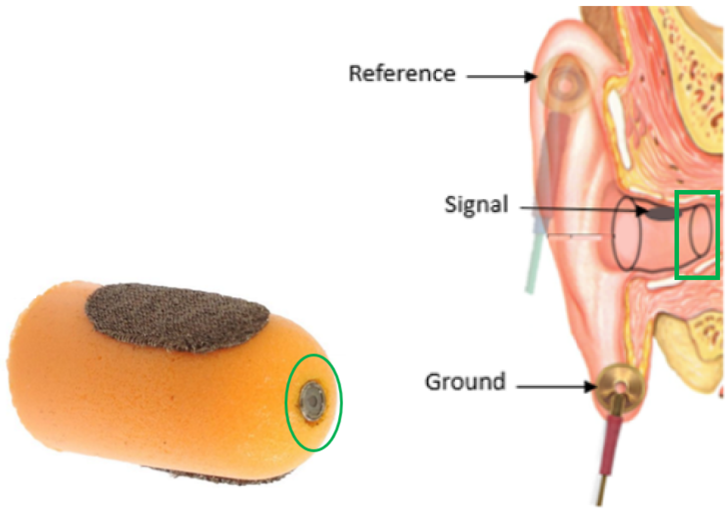


Figure 18: Emplacement of the 2nd microphone

Furthermore, the accelerometer placed on the jaw could focus on retrieving information on the kinematics and dynamics of the chewing process. Information on the jaws movements (opening and closing), chewing frequency and bite force estimation would be made available.

These additional signals could provide further information on the characteristics of chewing and swallowing. Thus more complete features could be extract that have the potential of further differentiating the types of ingested foods.

Finally, when the model reaches satisfactory classification accuracies, recordings obtained from the First Protocol could be used as testing data to estimate calories and further explore the relationships between the foods categorized within the same type of food or drinks.

Lastly, it is interesting to highlight that throughout the evaluation of the model as a pipeline, it was observed that the Random Forest Classifier outperformed linear-based models such as Logistic Regression, as well as non-strictly linear models such as KNN and SVM. These last two models should handle non-linear relationships but their sensitivity to outliers make them unreliable. For comparison purposes, implementing a more complex model which possesses similar characteristics to Random forest such as being able to handle higher dimensional data, being robust to outliers and successfully capturing non-linear relationships could be interesting to study. Some models that possess similar traits include other ensemble learning methods such as AdaBoost, which combines multiple weak classifiers to create strong classifiers, or Bagging, which combines multiple models trained on different bootstrap samples of the data could be relevant models to consider.

4.2 Calorie Estimate Approach

The current calorie estimation's performance is dependent on the pipeline's accuracy; if the model's classification improves due to the suggested additions the improvements would reflect in the calorie estimation. Further improvements could be implemented to fully remove any user input, such as a regression model which estimates bite weight, as introduced in the paper 'Bite-Weight Estimation Using Commercial Ear Buds' [16]. Caloric densities would still need to be associated to classified food types but the bite weight would be fully estimated from the audio signals and non-audio signals extracted from recording sessions.

A more general approach to calorie estimation that could be interesting to consider would focus on a generalized regression model which separately to the food classifier could take features such as chewing frequency, chewing amplitude, chewing length and accelerometer data labeled with defined calories. The model would train to associate certain features to calorie ranges and would output a calorie estimate for the testing data. Identifying patterns and studying the relationship between chews, swallows and food calories could be another approach to consider.

4.3 Conclusion

This study aimed to make sense of chewing and swallowing in-ear physiological bio (EEG) and non-bio signals (microphone). Its focus relied on identifying patterns that distinguish different types of ingested foods, classify them and ultimately attempt to estimate calorie intake. To study these patterns, a subject ingesting standardized bites of 'Soft Food', 'Medium Soft Food', 'Hard Food' and sips of 'Flat Drink' and 'Fizzy Drink' were recorded.

After careful pre-processing and feature extraction, the ANOVA test specializing in identifying the features that best distinguish n inputted classes, suggested that the features obtained from the recordings were not sufficient to represent all five of our classes. When comparing Fizzy and Flat drinks, for example, all features returned f-values below 2 indicating that none of the features differentiate the classes. However, the test also highlighted classes which had very distinguishable features and demonstrated good potential for a successful classification of food types.

A pipeline consisting of four classifiers in series was implemented with the purpose of food type classification. To decide on the components of its architecture, different classification approaches were tuned and explored such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machines, KNN, Gaussian Naive Bayes and Discriminant Analysis models. This selection provided a wide range of algorithms to test and identify the best performing model. After evaluating individual classifiers, the Random forest outperformed all other models for each classifier, reaching f1-scores as high as 87% when classifying. Therefore, this model was used for the four components of the pipeline. This architecture successfully classified Hard Foods with an overall accuracy of 70%. The successful classification of the 'The Hard Food' category demonstrates that accurate classification can be achieved when the features have strong significance. However, the lower accuracies observed in the Soft and Soft-Medium Food classes can be attributed to the lack of distinguishable features between them.

This approach is a first step toward making diagnostics out of chewing and swallowing artefacts. With further research and improvement, it holds the potential to advance the field of dietary monitoring. Future investigations could focus on integrating additional signals such as an additional microphone or accelerometer data. These additions could identify more characteristical features such as sounds within the oral cavity during food ingestion or capture jaw kinematics during mastication. Moreover, further exploring the relationship between chewing patterns, swallowing dynamics, and food calories could unlock new insights and pave the way for a more comprehensive and personalized approach to calorie estimation. Hopefully, this work opens up exciting avenues for advancing the understanding and application of in-ear physiological sensing in the domain of dietary assessment.

5 Appendix

A User Guide to Navigate through the Jupiter Notebook

1. First, identify the icon indicated in the figure below which can be found on the vertical left bar of the notebook, within the 'Folder' icon. Load the used data by selecting the provided files from the unzipped 'Recordings.zip' folder :

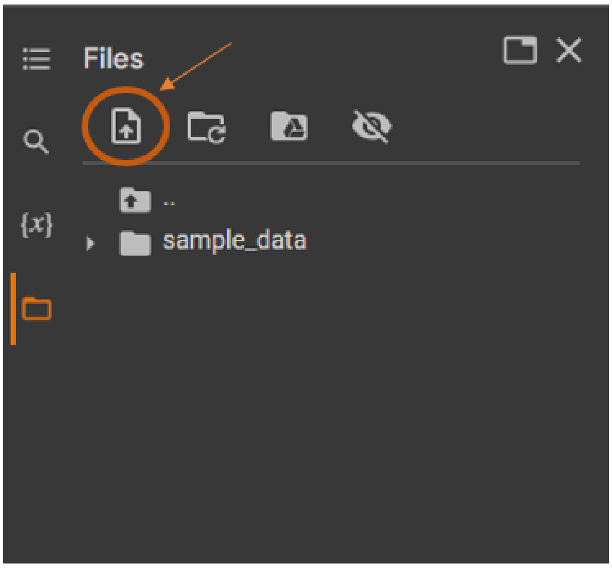


Figure 19: Loading Data into Jupiter Notebook Icon guide

2. Once all the data is run the entire notebook by pressing the short cut 'Ctrl+F9' or by clicking on the 'Runtime' Tab at the top of the notebook, followed by 'Run all':

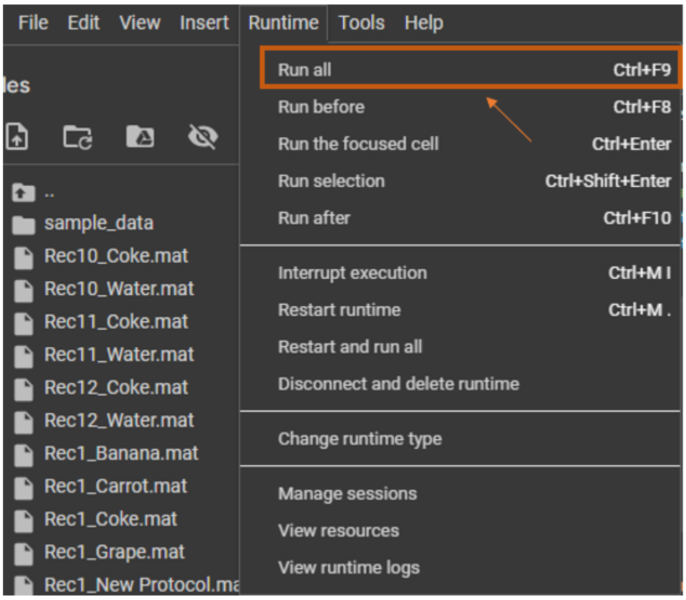


Figure 20: Running the whole notebook picture guide

3. The Table of Content will give a further breakdown on the purpose of each cell. Navigate freely through the notebook refering to the Table of Content for a specific section of interest.

B Extract of segments from each class

B.1 Samples for Soft Food

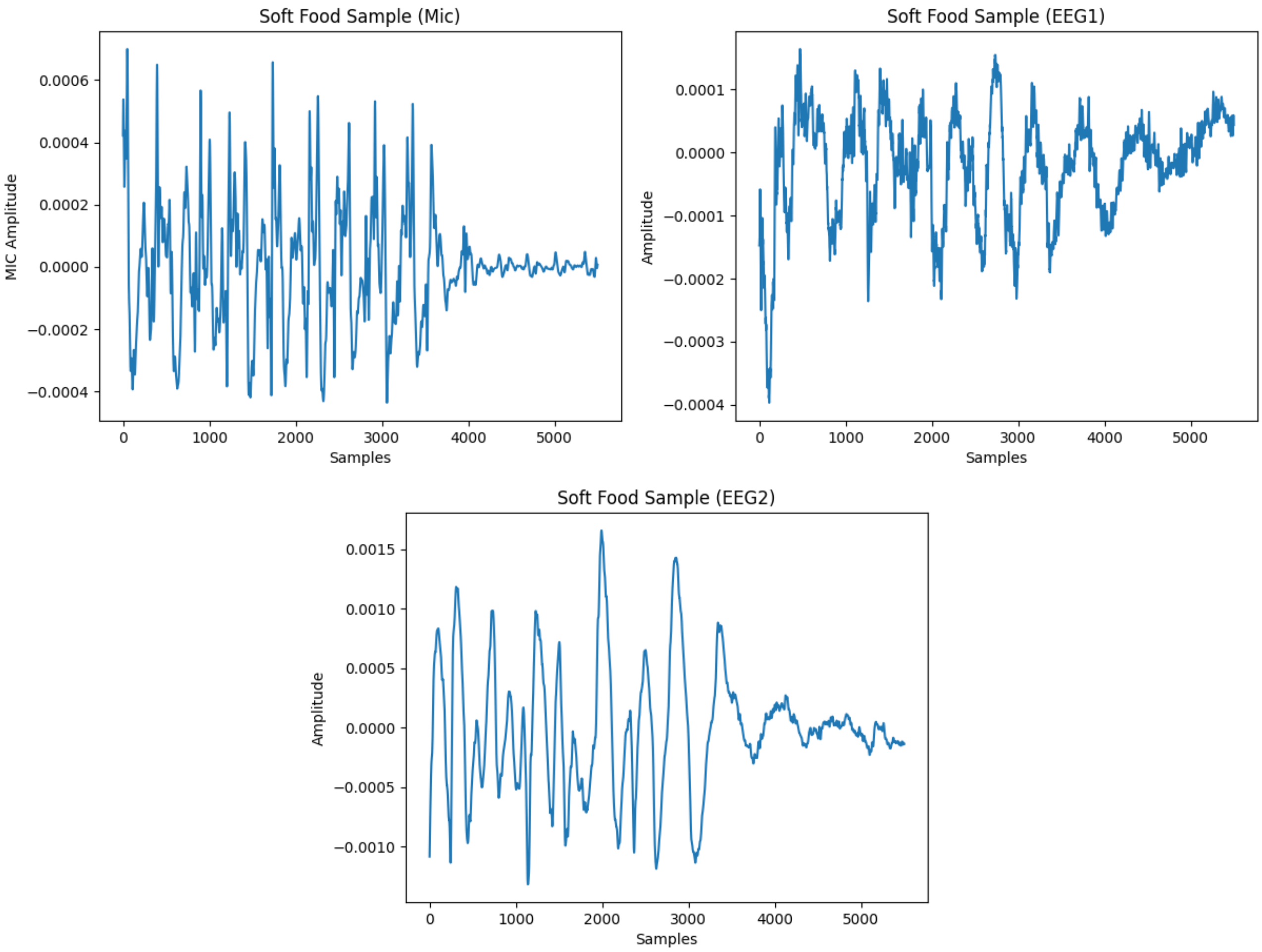
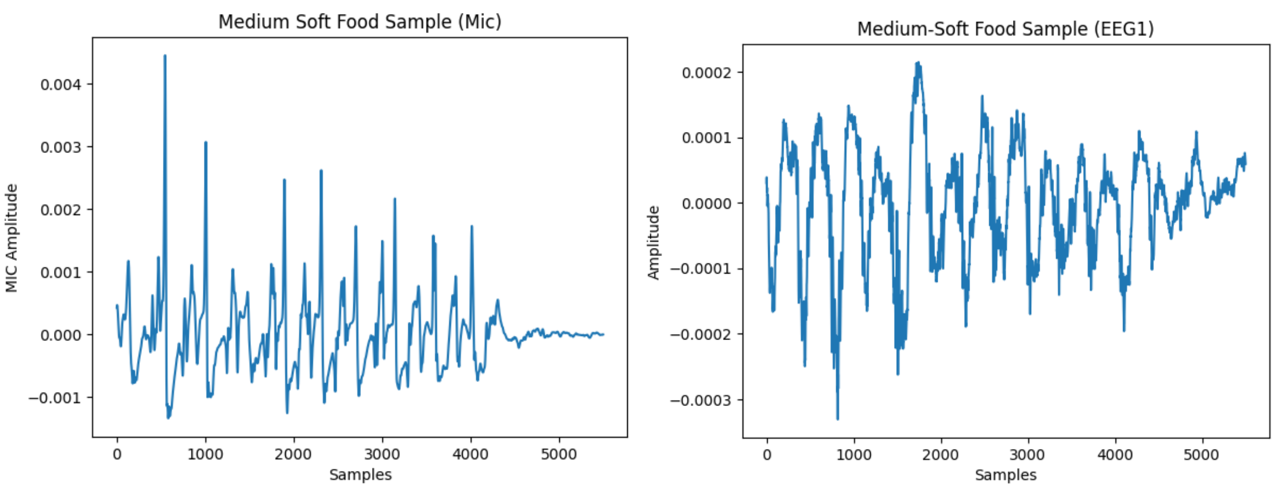


Figure 21: Mic, EEG1 and EEG2 samples of class Soft Food

B.2 Samples for Medium Soft Food



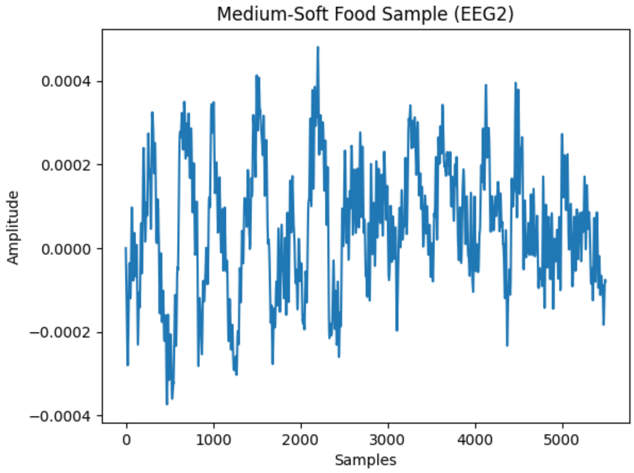


Figure 22: Mic, EEG1 and EEG2 samples of class Medium Soft Food

B.3 Samples for Hard Food

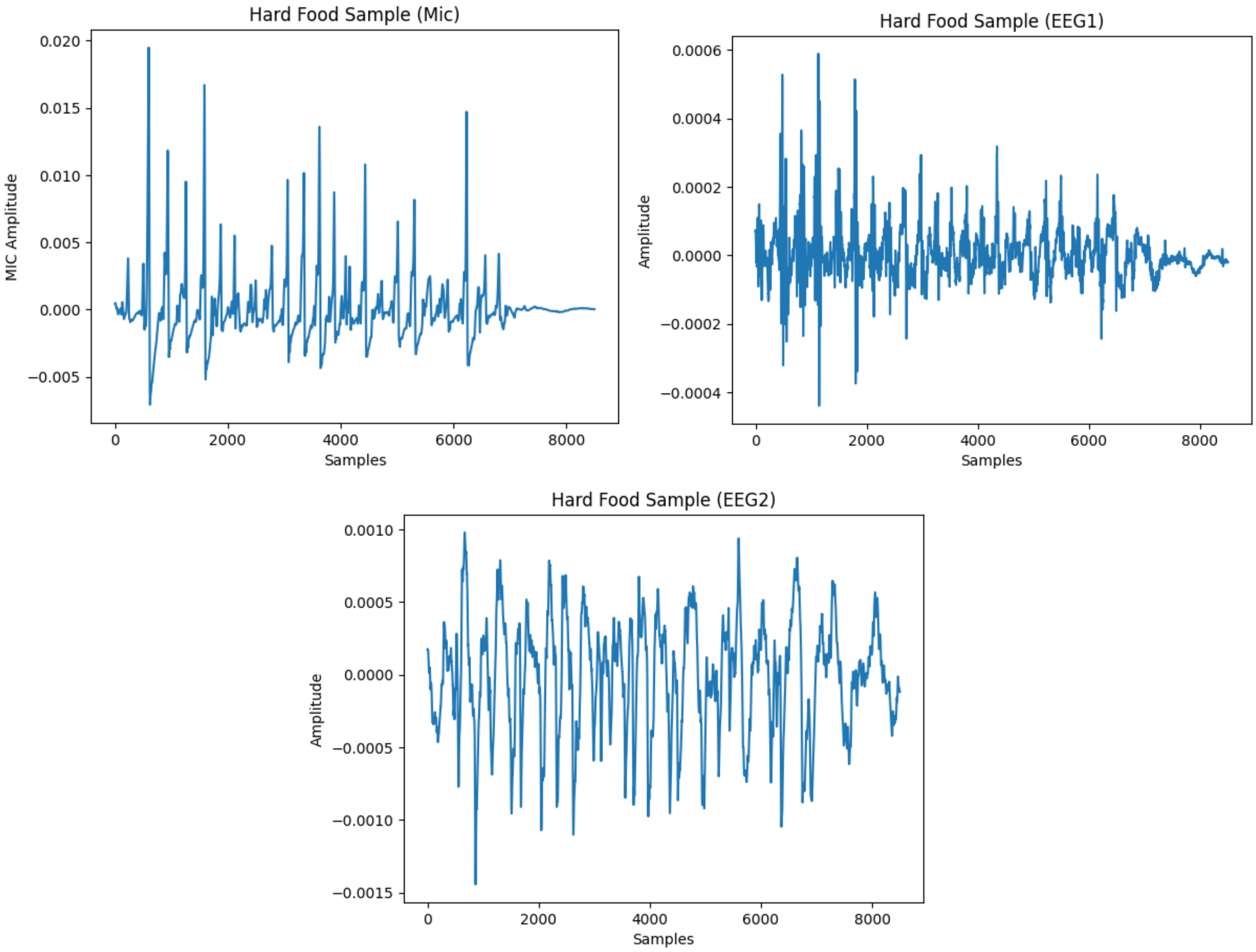


Figure 23: Mic, EEG1 and EEG2 samples of class Hard Food

B.4 Samples for Flat Drink

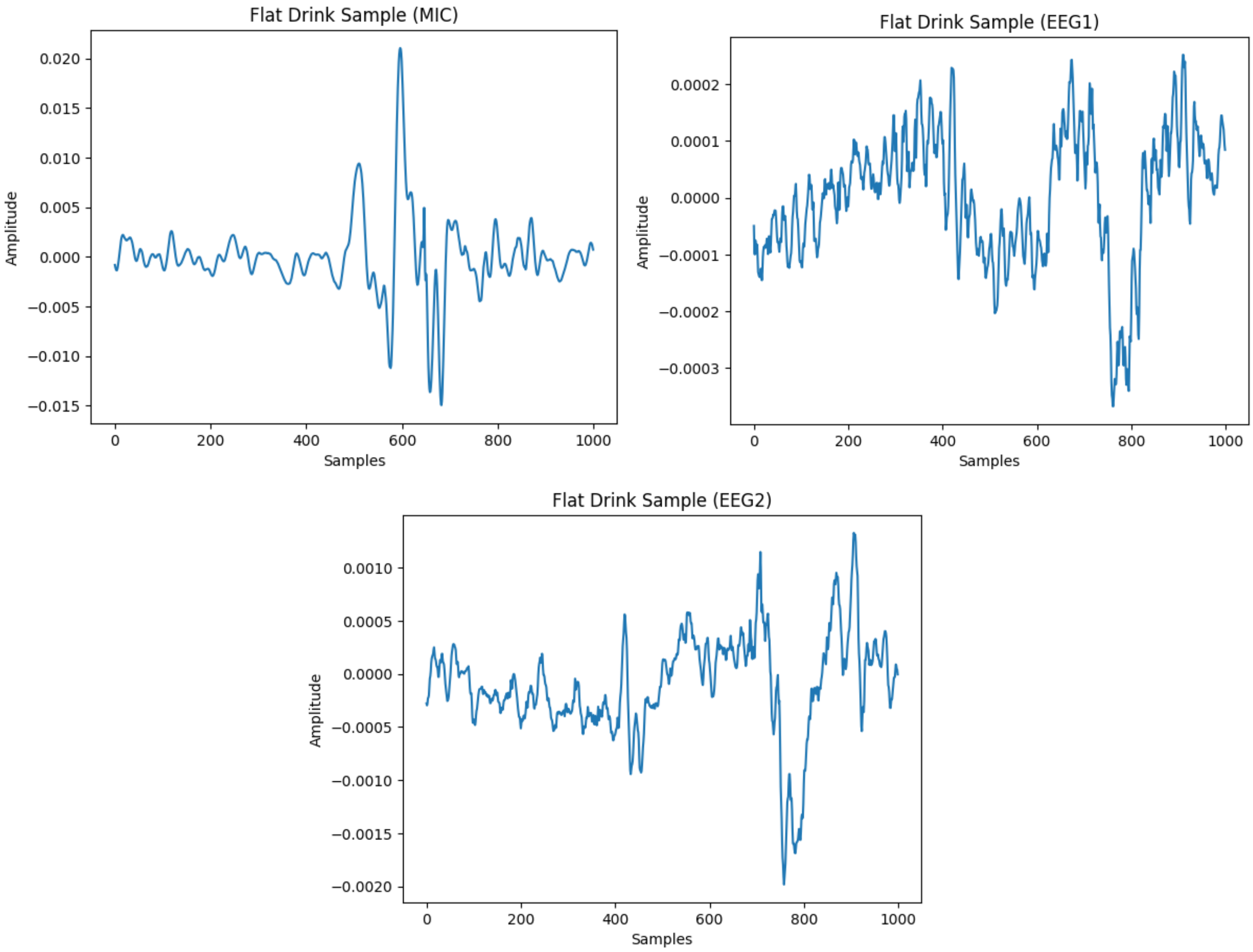
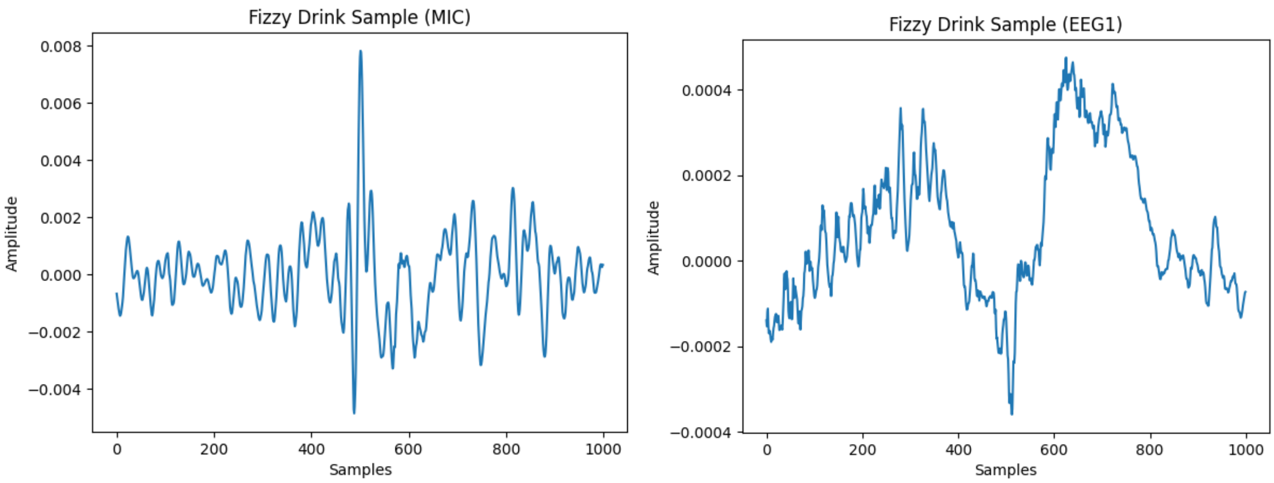


Figure 24: Mic, EEG1 and EEG2 samples of class Flat Drink

B.5 Samples for Fizzy Drink



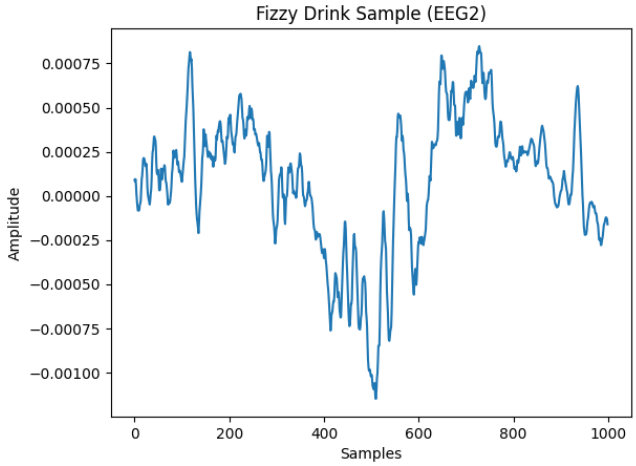


Figure 25: Mic, EEG1 and EEG2 samples of class Fizzy Drink

C MLP Neural Network

The code below represents a simple MultiLayer Perceptron (MLP) Neural Network implemented using Pytorch. An MLP is a feedforward neural network that is composed of multiple interconnected layers of artificial neurons. The implementation structure was inspired from the Department of Computing, Machine Learning for Imaging's tutorial.

The '__init__' method initializes the parameters of the network.

The 'forward_pass' performs the forward propagation of the network.

Comments provided in the code below provide a walk-through of the algorithm implemented.

```
import torch

class MLP_NN():
    def __init__(self, rng):
        # Construct and initialize network parameters
        D_in = 19 # Dimensions of feature-vectors.
        D_hid = 50 # Dimensions of hidden layer.
        D_out = C_classes # Dimension of Output layer (Number of classes)

        # Initialize parameters of the MLP using Xavier initialization
        fan_in = D_in # +1 to add bias in input layer
        fan_out = D_hid
        w1_init = rng.normal(loc=0.0, scale=np.sqrt(2/fan_in), size=(fan_in, D_hid))
        fan_in = D_hid # +1 to add bias in hidden layer
        fan_out = D_out
        w2_init = rng.normal(loc=0.0, scale=np.sqrt(2/fan_in), size=(fan_in, D_out))

        w1 = torch.tensor(w1_init, dtype=torch.float, requires_grad = True)
        w2 = torch.tensor(w2_init, dtype=torch.float, requires_grad = True)
        self.params = [w1, w2]
        #########################################

    def forward_pass(self, batch_sigs):
        # compute predicted y
        [w1, w2] = self.params

        #Convert input to Tensor
        batch_sigs_t = torch.tensor(batch_sigs, dtype=torch.float)

        #Add Bias for better performance [specific to ReLU]
        #feature_for_bias = torch.ones(size=(batch_sigs.shape[0], 1)) # [N, 1] column
            vector.
        #x = torch.cat((batch_sigs_t, feature_for_bias), dim=1) # Concatenate
```

```
    previous features with our Extra feature=1 for bias.

x = batch_sigs_t #Input when using

# Build network of form: x -> fully connected -> scaled relu -> fully
    connected -> softmax
# Hidden layer [LAYER 1]
h1 = x.mm(w1) #Dot product of X with Weight 1
h1_relu = h1.clamp(min=0) #applies ReLU activation function, set all negative
    values to zero and leaving positive values unchanged

# Fully-connected classifier (aka Densely connected classifier) [LAYER 2]
#h1_ext = torch.cat((h1_tanh_scaled, unary_feature_for_bias), dim=1) #Add
    bias term
h2 = h1_relu.mm(w2) #dot product btw output of prev layer and Weight 2
logits = h2
#################################

# Softmax activation function.
exp_logits = torch.exp(logits)
y_pred = exp_logits / torch.sum(exp_logits, dim=1, keepdim=True)
# Keepdim=True returns [N,1] array.
# Torch broadcasts [N,1] to [N,D_out] via repetition, to divide elementwise
    exp_h2 (which is [N,D_out]).

# Store activations. Will be needed for backprop.
self.__activations = [x, h1, h1_relu, h2, y_pred] # h1_ext,

return y_pred
```

References

[1] URL: https://www.nhs.uk/conditions/electroencephalogram/#:~:text=An%5C%20electroencephalogram%5C%20(EEG)%5C%20is%5C%20a,looked%5C%20at%5C%20by%5C%20a%5C%20doctor..

[2] David Looney et al. “The in-the-ear recording concept: User-centered and wearable brain monitoring”. In: *IEEE pulse* 3.6 (2012), pp. 32–42.

[3] Valentin Goverdovsky et al. “Hearables: Multimodal physiological in-ear sensing”. In: *Scientific reports* 7.1 (2017), p. 6948.

[4] Ghena M Hammour and Danilo P Mandic. “Hearables: Making Sense from Motion Artefacts in Ear-EEG for Real-Life Human Activity Classification”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2021, pp. 6889–6893.

[5] Edoardo Occhipinti et al. “Hearables: Artefact removal in Ear-EEG for continuous 24/7 monitoring”. In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2022, pp. 1–6.

[6] May 2022. URL: https://www.rcpsych.ac.uk/news-and-features/latest-news/detail/2022/05/18/hospital-admissions-for-eating-disorders-increased-by-84-in-the-last-five-years.

[7] Feb. 2021. URL: https://www.nhs.uk/mental-health/feelings-symptoms-behaviours/behaviours/eating-disorders/overview/.

[8] *Brainwave frequencies and effects*. URL: https://www.neurosonica.com/the-science/brainwave-types-frequencies.html.

[9] URL: https://www.pfa.org/wp-content/uploads/2019/02/InTouch_v11.1.pdf.

[10] Valentin Goverdovsky et al. “In-ear EEG from viscoelastic generic earpieces: Robust and unobtrusive 24/7 monitoring”. In: *IEEE Sensors Journal* 16.1 (2015), pp. 271–277.

[11] Ryan Smoot. *ECM vs. MEMS microphones: Does new mean better?* May 2021. URL: https://www.electronicproducts.com/ecm-vs-mems-microphones-does-new-mean-better/.

[12] Dong-Hwa Jeong and Jaeseung Jeong. “In-ear EEG based attention state classification using echo state network”. In: *Brain sciences* 10.6 (2020), p. 321.

[13] Saul Mcleod. *P-value and statistical significance: What it is amp; why it matters*. May 2023. URL: https://www.simplypsychology.org/p-value.html.

[14] Abdulhamit Subasi. *Practical machine learning for data analysis using python*. Academic Press, 2020.

[15] First Teddy Surya Gunawan et al. “Food intake calorie prediction using generalized regression neural network”. In: *2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*. IEEE. 2018, pp. 1–4.

[16] Vasileios Papapanagiotou, Stefanos Ganotakis, and Anastasios Delopoulos. “Bite-weight estimation using commercial ear buds”. In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2021, pp. 7182–7185.

[17] Juan M Fontana et al. “Energy intake estimation from counts of chews and swallows”. In: *Appetite* 85 (2015), pp. 14–21.

[18] Edward S Sazonov et al. “Toward Objective Monitoring of Ingestive Behavior in Free-living Population”. In: *Obesity* 17.10 (2009), pp. 1971–1975.

[19] Geek3. *requency response curves of four linear analog filters: Butterworth filter, Chebyshev filter of type 1 and 2 and Elliptic filter, each one as 5th order filter*. Dec. 2016.

[20] Matteo Gambera. *How to extract features from signals*. Feb. 2021. URL: https://m-gambera.medium.com/how-to-extract-features-from-signals-15e7db225c15.

[21] Teemu Kanstren. *A look at precision, recall, and F1-score*. May 2021. URL: https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec#:~:text=F1%5C%2DScore%5C%20is%5C%20a%5C%20measure,than%5C%20the%5C%20traditional%5C%20arithmetic%5C%20mean..