Controlling IoT devices with our brain, eyes, and facial muscles.

James Beavis - 21046104
Supervised by Nhat (Nick) Pham
Moderated by Hantao Liu
BSc Computer Science (Year in Industry)
Cardiff University, School of Computer Science and Informatics
08/05/2025

1. Abstract

For the 17.8% of people in England and Wales who are disabled, using conventional means of controlling technology such as keyboards, mice and touch screens may be difficult or even impossible. Accommodations in the form of alternative means of control exist, but they often come with caveats. This calls for a new accommodation for these individuals, particularly those with motor impairments.

This study explores the use of biosignals, specifically EEG, EOG and EMG, as an alternative means of controlling technology. Signals from the brain, eyes and facial muscles are captured and filtered, and useful time and frequency domain features are extracted from the raw data. These features are used to train a random forest machine learning model. The resulting classifier is used to predict a user's actions in real time. The model is designed to identify when the user is looking up, down, left and right, clenching their jaw, and generating alpha waves. The system's ability to predict the user's actions is demonstrated through the scenario of controlling the playback of music; the person's actions are mapped to controls.

5-Fold cross-validation showed that the ML model has an average accuracy of 95.71%. The system represents a promising proof-of-concept for biosignal-driven assistive interfaces, though further validation on a broader user base is required.

2. Acknowledgements

I would like to thank my supervisor, Dr Nick Pham, for his invaluable guidance throughout the development of this project. I would like to extend my thanks to Kha Huynh and Loïc Lorente Lemoine for their support. Additionally, I would like to thank my parents, siblings, and friends.

3. Acronyms

IoT – Internet of Things

ML – Machine Learning

EEG – Electroencephalography

EMG – Electromyography

EOG – Electrooculography

ExG – Electrophysiology

BCI – Brain-Computer Interface

BLE – Bluetooth low energy

CNN – Convolutional Neural Networks

LSL – Lab Streaming Layer

SVM – Support Vector Machines

Table of Contents

List of figures

List of tables

4. Introduction

4.1 Motivation

An estimated 1 in 400 babies born in the UK have cerebral palsy (Scope 2024). It is usually caused by an injury to the brain, either before or after birth, but it has no one cause. It is a lifelong, incurable condition. The 2021 census reports that 17.8% of people in England and Wales are disabled (Office for National Statistics 2021).

For those with disabilities that limit their movement, like people with cerebral palsy, or people with limb differences, there may be some difficulty in controlling devices using conventional methods. This includes buttons on a screen, keyboards, mice, or possibly even speech control, depending on how severe their disability is. This has contributed to the fact that disabled people are over 50% more likely to face barriers to accessing digital and online services than able-bodied people (Ward 2021).

To accommodate disabled people and to make technology more accessible, alternative means of controlling devices have been developed, including eye trackers and more accessible keyboards. However, these alternative solutions come with their own drawbacks. For example, some eye trackers require the user's eyes to be visible to a camera, making them impossible to use in low light (Majaranta and Räihä 2002). Other solutions like the P300 speller have high latency (Farwell and Donchin 1988), whereas this project will focus on a system that can operate in real time.

Brain-computer interfaces (BCIs) and biosignal classification offer a potential pathway for more accessible technology. EEG, EOG, and EMG offer non-invasive ways to measure brain and muscular activity. These signals can be reliably linked to voluntary actions such as blinking, eye movement, and muscle tension, allowing systems to infer user intent without requiring speech or limb control, making them attractive as alternative input methods (Nicolas-Alonso and Gomez-Gil 2012). This study attempts to solve the issue of accessibility for disabled people by exploring the use of biosignals for system control. In this study, the BCI chosen was the OpenBCI Ganglion.

4.2 Aims

The aims of this project are as follows:

1. Collect and annotate EEG, EOG and EMG signals.
    a. Use an OpenBCI Ganglion board to collect the signals.
    b. Develop a labelling tool to annotate the signals.
2. Prepare collected biosignals.

    a. Filter the data.
    b. Extract time and frequency domain features from the raw signal.
3. Train a random forest classifier to detect user actions.
4. Demonstrate the random forest classifier by implementing real-time prediction.
    a. Use these predictions to control the playback of music.
    b. Provide a video demonstration of this application.
5. Evaluate system accuracy.
    a. Use cross-validation to calculate accuracy.
    b. Generate a confusion matrix to assess class performance.

The system will be written in Python, and will use the Scikit-learn library for machine learning.

4.3 Scope

Due to time constraints, some areas of study are out of scope. The majority of the technologies used in this project, including collecting biosignals, filtering the signals, extracting useful features from raw data, evaluating machine learning models, and to some extent training the random forest classifier, were not taught in the course. Due to this, time has been allocated to learning and researching these topics.

Firstly, EEG, EOG and EMG biosignals will only be collected from one participant. This may affect generalisability. The system will be tested only on the same participant. Additionally, though the project is designed to make technology more accessible to disabled people, the participant is not disabled. Finding disabled volunteer participants is out of scope for this project.

Additionally, though it was considered, deploying the project to a Raspberry Pi is not in scope for this project due to time constraints, and lack of BLE libraries on the Raspberry Pi Zero W used.

The following sections will outline the background to this project, the system design and evaluation.

5. Background

5.1 Brain-Computer Interfaces

Brain-computer interfaces (BCIs) allow communication between a person's brain and a device. They are often used in healthcare to repair or augment communication or sensory-motor functions after neurological damage (Krucoff et al. 2016), though they can also be used as a human-machine interface, allowing for the control of a device without moving a person's limbs (Wolpaw and Wolpaw 2012), as is the case in this project.

BCIs operate by detecting signals such as brain activity, eye movements and muscle contractions (Nicolas-Alonso and Gomez-Gil 2012). For example, when a person moves their eyes, or makes an expression with their face, electrical signals are transmitted to their muscles which can be detected by BCIs and passed on to a computer. Likewise, a person's brain is constantly emitting electrical signals, though the kinds of signals it sends depends on a person's mental state (Trejo et al. 2015). BCIs are relevant for those with physical disabilities, who may not be able to use conventional input methods such as keyboards and mice (Mak and Wolpaw 2009). By capturing bioelectrical signals from the brain and other parts of the body, BCIs offer an alternative mode of interaction with computers.

The BCI chosen for this project was the OpenBCI Ganglion Biosensing Board. This device communicates with a computer via Bluetooth low energy (BLE, a wireless network technology designed for low power operations), transmitting the signals it collects using the electrodes attached to it (OpenBCI 2024). The board is capable of collecting EEG, EOG, and EMG signals using its four channels. The signals can be read and visualised in real time and saved for later processing via OpenBCI's graphical user interface (OpenBCI 2025). This board was chosen over other BCIs, such as the ANT Neuro EEGO Sports BCI cap (ANT Neuro 2024) for its fast setup time. Attaching the electrodes and preparing the system typically took no longer than ten minutes. The ANT Neuro EEGO Sports BCI cap sits on the scalp and requires electrode gel, meaning that it would have likely required a second person to set up. Adhesive electrode pads are compatible with the Ganglion, which were preferred rather than using electrode gel, as they require no post-session cleanup and are easier to attach to one's self. This added allows for greater independence for disabled people if an individual can set the system up themselves.

Figure 1 The OpenBCI Ganglion used in this project, with the laptop used to process the signals collected behind it.

5.2 Biosignals

Bioelectrical signals are generated by brain, muscle activity, and eye movements, and these can be detected using BCIs (Kawala-Sterniuk et al. 2021). These signals can be measured, and this type of study is called electrophysiology. When these signals are recorded from the brain, the process is called electroencephalography (EEG), from the eyes, electrooculography (EOG), and from skeletal muscle, electromyography (EMG) (Schomer and Lopes Da Silva 2017). This study focuses on using brain waves, eye movements, and facial movement to control the playback of music, and so EEG, EOG and EMG are used.

5.2.1 EEG

Figure 2 An adhesive electrode.

EEG stands for electroencephalography. This is a method to record the voltage fluctuations created by neurons in the brain, which can represent brain activity (Amzica and Lopes Da Silva 2017). These fluctuations can be analysed to make inferences about the state of the brain; for example, the voltage fluctuations created when a person is tired, versus when they are excited, look very different (Trejo et al. 2015). Usually during these tests, electrodes are placed on the scalp according to the International 10-20 system (Nuwer et al. 1998), allowing a researcher to see the activity of all areas of the brain.

In this project two electrodes were placed on opposite sides of the forehead, above the eyebrows. This placement was chosen due to the limitations of the adhesive electrodes used, they are unreliable when applied to hair-covered areas of the scalp, they require contact with the skin. One of the adhesive electrodes used can be seen in figure 2. While this positioning limits access to signals from posterior regions of the brain, it does not hinder this project, as the only brain activity being measured is the presence of alpha waves.

Alpha waves are a type of brain wave that can be detected via EEG. They are strongest when a person is at rest, and so they are thought to represent the brain in an “idle state” (Malik and Amin 2018). They are usually detected at the back of the scalp, when measuring signals generated by the occipital lobe, however they can also be detected at frontal sites, like on the forehead (Teplan 2002). In this project, training data of alpha waves were collected when the participant had their eyes shut, deliberately relaxed. An example of these alpha waves can be found in figure 3. In the proposed music controlling scenario, the user would also close their eyes whilst relaxing and clench their jaw to change the playlist of music.

A strength of using biosignals from the brain detected by EEG is that brain activity does not require the use of muscles. In fact, alpha waves are strongest when a person is not moving their body, with their eyes shut. This has a useful application in allowing a disabled person with limited control over their motor functions to control a device without using their limbs. An unfortunate weakness is that EEG signals are more prone to noise than EOG and EMG signals (Hu et al. 2015). Note the spike around sample 350 in figure 3; this could be an artifact from eye movements. Additionally, alpha waves are harder to detect on the forehead than at the back of the scalp (Britton et al. 2016), though the hardware used in this project leaves no alternative.

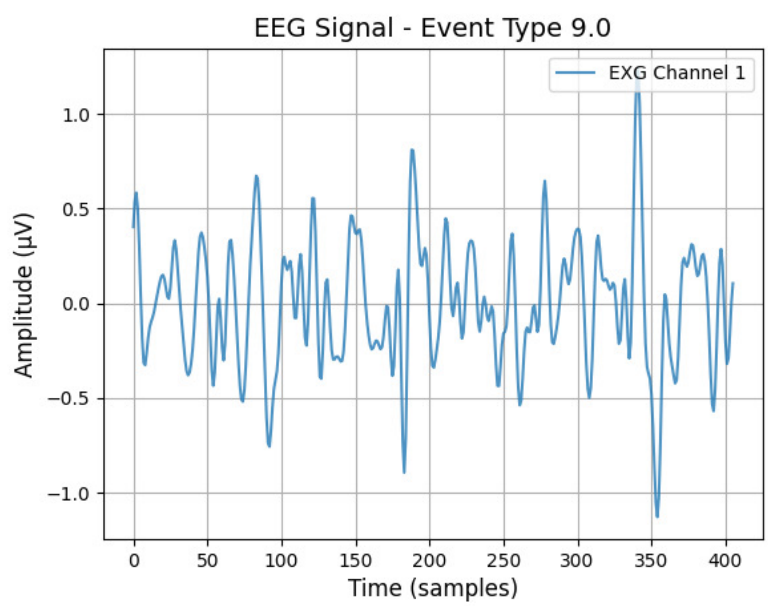

Figure 3 Alpha waves.

5.2.2 EOG

The front of the eye is electrically positive compared to the back of the eye. This means that any movements of the eye create a voltage difference, which can be detected using EOG – electrooculography (Bulling et al. 2008). Electrodes placed around the eyes, like on the temple and forehead, can detect changes in voltage and this can be used to infer direction or blinking. Figure 4 shows a blink. In the proposed music control scenario, movement of the eyes correspond to multiple different commands; when a person looks up, the volume increases, down, the volume decreases, left, the track changes to the previous song in the playlist, right, the track skips to the next song in the playlist.

A key advantage of using biosignals detected via EOG in this project is that these signals are very easy to detect; the signals are strong and distinct, allowing for accurate classification (Sharma et al. 2020). However, a limitation is that eye movements can happen accidentally, far more often than other control means like facial expressions. Blinking happens involuntarily, and so it is not used as a means of control in the proposed scenario. Samples were captured of blinking, and the random forest model was trained to detect them, but this is to avoid misclassification of other classes. Looking left, right, up and down will be detected by the classifier regardless of the user's intent to provide an input to the system, however meaning that the system is inherently sensitive to unintended inputs when relying on EOG.

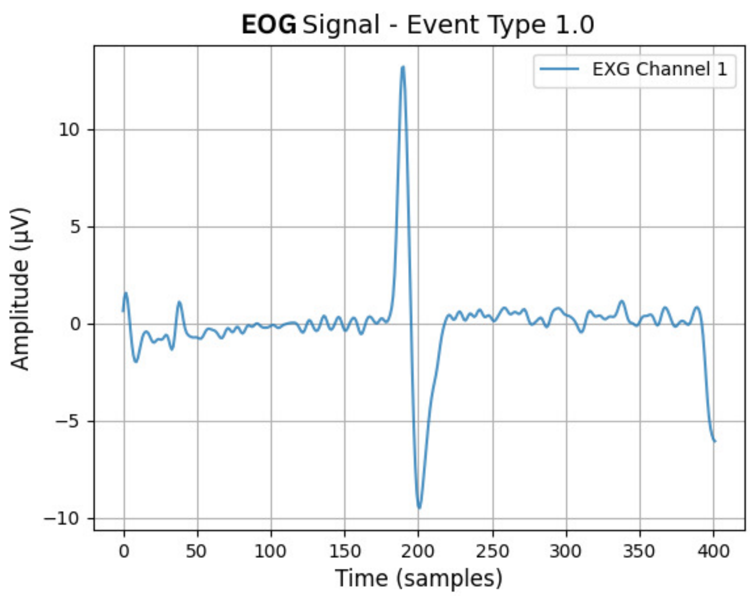

Figure 4 A blink.

5.2.3 EMG

When a muscle is activated, its fibres generate voltage changes. Electromyography (EMG) can detect these small electrical signals when an electrode is placed on the skin over the muscle (Robertson 2014). In this project, EMG is used to detect jaw clenching, and the key muscle being measured here is the masseter, the primary muscle used for jaw clenching and chewing. In the proposed music control scenario, when a jaw clench is detected, the music pauses or resumes playing depending on its current state.

A strength of using biosignals detected via EMG is that like EOG signals, they are easy to detect and easily distinguishable (Nazmi et al. 2016). Figure 5 clearly shows a jaw clench where there is a large spike of activity. This makes classification for this event more accurate. However, clenching the jaw can become tiring after repeated use, especially for a disabled person who may have muscle fatigue. Additionally, in this project, an electrode is placed on the temple to detect both EOG and EMG signals. This means that unintentional eye movement could cause artifacts to appear in the signal, making class classification more difficult. In figure 5, the spike around sample 350 could be one of these artifacts.

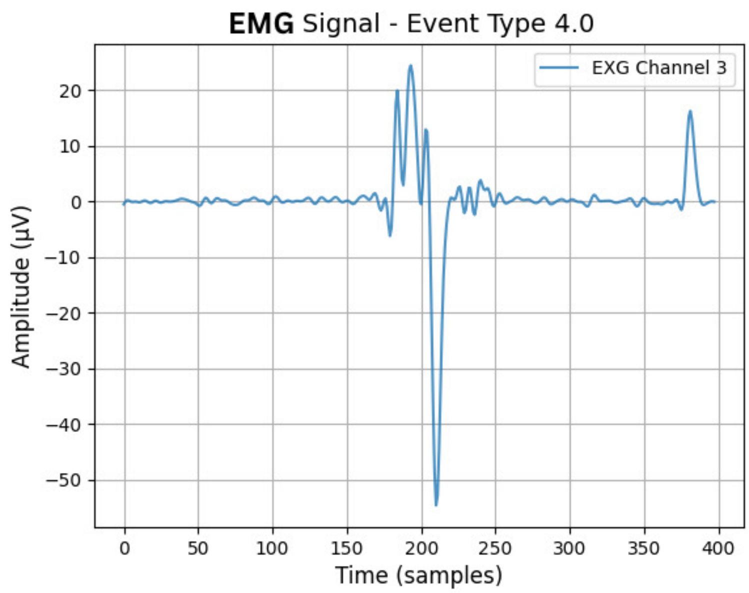

Figure 5 A jaw clench.

5.3 Machine Learning

The machine learning algorithm used in this project is ‘Random Forest’ (Breiman 2001). The random forest model is a supervised algorithm, meaning it uses labelled data to train models. The training data is already marked with which class it belongs to, for example signals representing a jaw clench are labelled as such; this is done during data collection and is explained further in section 6.2.1. In contrast, unsupervised learning explores unlabelled data.

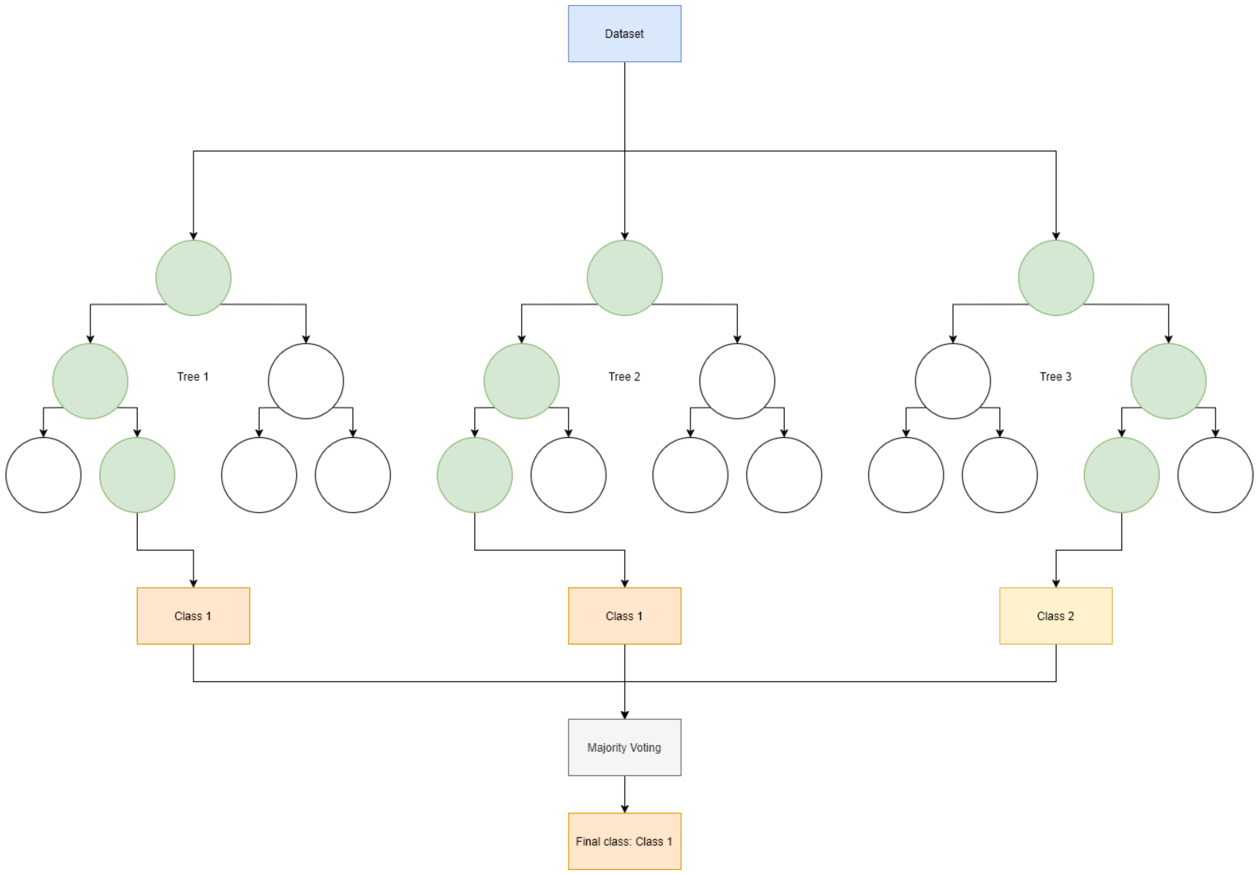

Figure 6 A random forest classifier.

The random forest model is made up of multiple decision trees, which are another type of algorithm. A depiction of a random forest can be seen in figure 6. Decision trees have a flowchart-like tree structure, starting with a root node, which feeds into internal nodes called decision nodes. The leaves of the tree represent class labels. The model splits the data at each node based on a feature, and whether that feature meets the threshold. The decision made here determines which decision node is chosen next, and this repeats until it reaches a leaf node. In a random forest model, many decision trees vote together before making a final classification (Rokach and Maimon 2005). Each tree is trained on a random subset of the data, using the same number of samples as in the original dataset. This means that some samples are reused, which is called training with replacement. This technique is called bagging, or bootstrap aggregating. This reduces overfitting by increasing diversity between trees and increases robustness (Breiman 2001). Overfitting occurs when the model becomes too specific to the training set rather than learning general patterns which can be applied to new data (Heaton 2018).

The random forest classifier was chosen for a variety of reasons. Firstly, it works well with small datasets (Han et al. 2021). Due to the short timeframe of development, and as a precaution against a change in planned methods, it was anticipated that a relatively small amount of training data would be collected. Approximately two hours of usable recordings were collected by the end of development, which is more than enough for the random forest classifier to be reasonably accurate. Secondly, random forests support multi-class classification well; they are resistant to overfitting (Prinzie and Van Den Poel 2008). In this project, there are ten distinct classes, which are explored more in section 6.2.3. Thirdly, random forests are fast to train (Yates and Islam 2021). In this project, after preprocessing the training data, training the model took only thirty seconds. Lastly, random forests handle complex, variable, nonlinear data such as biosignals well .

Another machine learning algorithm that was considered at the start of development was convolutional neural networks (CNNs) (Lecun et al. 1998). They are commonly used in biosignal classification (Craik et al. 2019); however it was decided that they were not appropriate for this project. Firstly, random forests provide interpretable feature importance scores; its decisions are easily and directly explainable. Deep learning models such as CNNs can instead be thought of as a “black box”, data goes in and classifications come out, but it can be harder to understand which channels or signal features can influence classification, which is important as it can inform future design and refinement (Salahuddin et al. 2022). More information about which features are important in this project can be found in section 6.3.

Secondly, though the system currently performs on a general-purpose laptop, future iterations may be deployed on low-power hardware such as a Raspberry Pi. Random forests are more suited to this rather than CNNs due to their low memory cost (Breiman 2001). More information about the planned deployment to a Raspberry Pi can be found in section 7.3.

5.4 Physical Disabilities

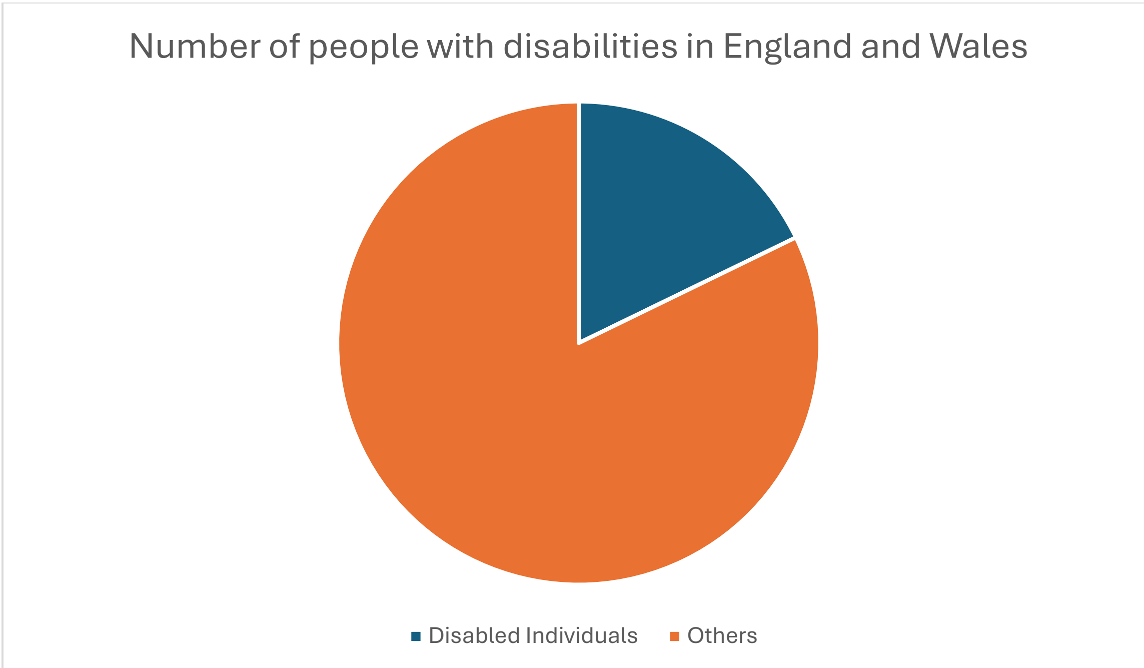

Figure 7 A pie chart showing the percentage of people with disabilities in the UK.

An estimated 1 in 400 babies born in the UK have cerebral palsy (Scope 2024). It is usually caused by an injury to the brain, either before or after birth, but it has no one cause. It is a lifelong, incurable condition. Though this project focuses on those with cerebral palsy, there are other disabilities which can make it more difficult to us technology, and so this study will also aim to provide an accessible means of control for the 17.8% of people in England and Wales with a disability (Office for National Statistics 2021). This ratio of disabled to able bodied people can be seen in figure 7.

Cerebral palsy is a group of incurable conditions that effect movement and coordination (Rosenbaum 2007). Symptoms can include weak limbs, clumsy and uncontrolled movements, and a feeling of being too “stiff” or too “floppy”. These symptoms can make controlling technology through conventional means, such as using a hand to press a button, difficult, or in some cases impossible (Saavedra et al. 2009). Cerebral palsy can also affect speech, meaning that current accommodations in the form of speech control may not be appropriate (NHS 2023).

The actions used as inputs in this project have been identified as possible for a person with cerebral palsy to carry out. Individuals often have good eye movement control, even if they have impairments with limb movement or speech (Saavedra et al. 2009). Though the condition can affect the jaw during speech, individuals with cerebral palsy are often capable of simply clenching their jaw, though some with severe cerebral palsy may find this taxing . Finally, producing strong alpha waves does not require the use of muscles at all (Teplan 2002).

5.5 Existing Solutions

5.5.1 Conventional Assistive Technologies

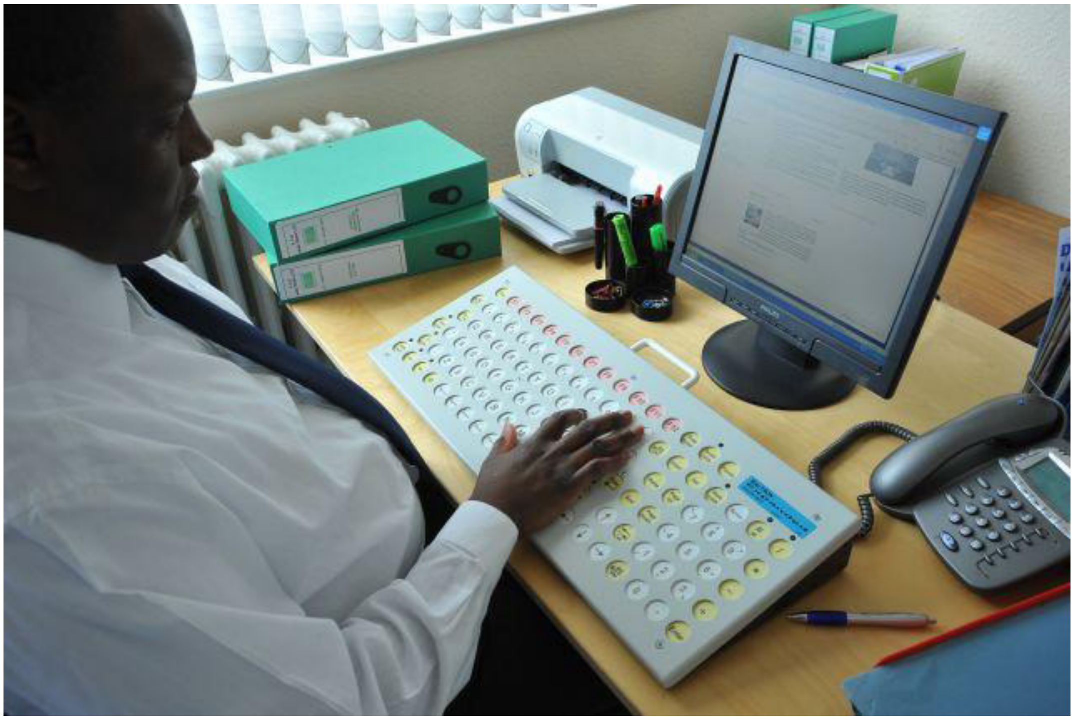

Figure 8 A Maltron expanded keyboard (Maltron 2024).

Commercial assistive technologies exist to allow people with physical impairments to use technologies. These include screen readers, braille displays and screen magnifiers for the blind or visually impaired, but also adaptive keyboards for those with limb differences or difficulties with mobility. For example, as seen in figure 8, the Maltron expanded keyboard is designed for people with muscle control problems, like those with cerebral palsy (Maltron 2024). It includes a raised key guard to prevent accidental keystrokes. The idea of accidental input prevention is also used in this project, each input action is distinct.

Additionally, eye movement trackers can be used to plot where the user is looking on a screen, which can be used to control a system. The system developed in this project similarly uses eye movements as input, using biosignals rather than visual tracking. Eye movement trackers may be difficult to use in the dark; if the user’s eyes are not visible to the camera, it cannot track them (Majaranta and Räihä 2002). This project can be used in the dark or low light.

Technologies exist to allow a user to control a device using their voice, which has been used as an accommodation for disabled people (Vigouroux et al. 2023). Unfortunately, people with cerebral palsy may have trouble speaking, it has been estimated that 50% of children with cerebral palsy have motor speech disorders (dysarthria) (Pennington 2012). This makes speech control impractical for many individuals.

5.5.2 Biosignal-based Systems

Brain, eye and muscle biosignals can provide control mechanisms for those who cannot use traditional inputs.

Due to the complexity and variability of biosignals, machine learning techniques have been widely applied to improve classification performance (Craik et al. 2019). Support Vector Machines (SVMs), Random Forests, and Neural Networks have all been used to classify EEG and EMG signals in applications such as epilepsy detection, sleep stage classification, and gesture recognition. These studies demonstrate that machine learning models can handle the noise and individual variability inherent in biosignal data.

Biosignal-based systems have been explored in a variety of applications. EEG signals have been used in P300 spellers, allowing users to select letters based on focused attention patterns (Farwell and Donchin 1988). These systems depend on users maintaining visual attention on a grid of flashing stimuli. Here, a grid of letters is shown to the user, with the columns flashing in turn. When the user concentrates, the currently flashing column is selected, making the rows flash, so that the user can concentrate again and select a letter. This technology is unsuitable for this project's aims due to its high latency; it can take multiple seconds for a user to select an input, whereas some applications such as music playback control require real time control; for example, if a person wanted to change the currently playing track to the previous one in the playlist, but in the time it takes for a system to process the input, the current song ends and the next one begins, the system will incorrectly change the track to the original song.

Another biosignal-based system is Earable. Earable is a device worn behind the ear, which similarly to this project collects EEG, EOG and EMG data to control a technology. A photo of the Earable device can be found in figure 9. Designed to monitor neurological disorders, such as epilepsy and insomnia (rather than make a device more accessible to a disabled individual), it has also been demonstrated to control IoT devices, meaning that it could be used for this purpose. The Earable system has been used to control a drone's movements, using a person's biosignals as control inputs; eye movements and facial muscle contractions were used to control the drone to take off, land, move and rotate. In another demonstration, a person's alpha wave rhythm was used to dim a light when the user tried to relax or sleep (Pham et al. 2019).

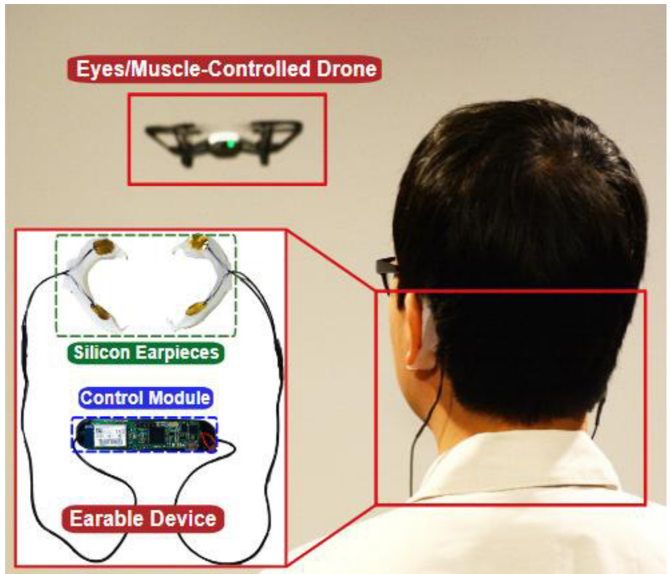

Figure 9 Earable (Pham et al. 2019).

Earable shares much in common with this project, using similar EEG, EOG and EMG signals to control a device. However, this project will allow its users to control a device using their alpha waves intentionally, rather than using the alpha waves passively. With the scenario of music playback control in mind, the user will be able to consciously generate alpha waves and use them to change the playlist used. Allowing a person with severe motor impairment to control a device with their brain waves as a deliberate control input is important; though they may find it difficult to use their limbs, entering a relaxed state does not require fine motor skills.

Additionally, researchers Kanungo et al have developed a method of using steady-state visually evoked potentials (SSVEPs) and eye blinks to control a wheelchair, increasing accessibility for disabled people. In their system, users focus on flashing stimuli to indicate directional commands, much like the P300 speller. A series of eye blinks serve as a stop signal. It was decided that eye blinks would not be used as a means of control for this project. Though eye movements can occur accidentally, with enough focus, the user can prevent this; however the user will eventually have to blink, which could be wrongly interpreted as an input. Furthermore, Kanungo et al.'s system reports an average command execution time of approximately four seconds and an accuracy of 87%. This project aims to improve upon both latency and classification accuracy by leveraging a machine learning-based, multi-signal approach (Kanungo et al. 2021).

Table 1 compares the described existing solutions.

Approach	Description	Biosignals?	Cons
Maltron keyboard (Maltron 2024)	A keyboard with a raised keyguard to prevent accidental keystrokes.	None	Requires the use of hands – this project requires only eyes, jaw and brain.
Voice recognition (Vigouroux et al. 2023)	Users control devices with spoken commands.	None	Individuals with cerebral palsy are likely to have difficulties with speech (Pennington 2012).
P300 Speller (Farwell and Donchin 1988)	Allows users to select letters from a grid by focusing their attention on flashing targets.	EEG	High latency and requires concentration – this project aims to be real-time.
Earable (Pham et al. 2019)	A device which sits behind the ear and monitors eye movements, brain waves and facial contractions.	EEG, EOG, EMG	This project aims to build upon this idea and use alpha waves as an intentional method of control.
Eye movement trackers (Majaranta and Räihä 2002)	Detects where a person is looking to control a system.	None/ EOG	May require the user to sit somewhere the camera can see.
BCI controlled wheelchair (Kanungo et al. 2021)	The user controls wheelchair movement using focus and blinking.	EEG, EOG	Uses blinking as controls, low accuracy and high latency.

Table 1 List of existing solutions.

6. Methodology

This project aimed to create a system which could classify actions such as eye movements, facial expressions and the brain's alpha waves in real time, and use these as inputs to control the playback of music. The process of creating this system included collecting training data using the Ganglion, filtering and processing the captured bio-signals, training a machine learning model using the resulting data, optimising and improving the accuracy of the model, and developing a program that uses the results from the model to play music. The project was written entirely in Python, and used libraries such as Scikit-learn and PyGame for machine learning and playing music respectively (PyGame 2024).

6.1 System Overview

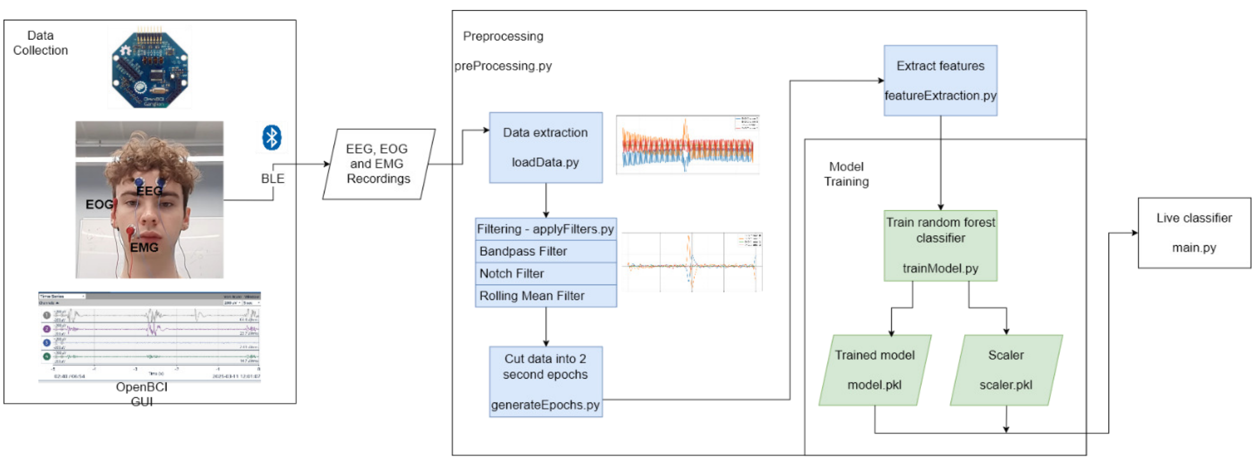

Figure 10 System overview diagram.

Figure 10 displays a high-level overview of the system architecture of this project, from signal acquisition, through data filtering and feature extraction, to classification and action execution. Before this pipeline can be carried out and operate in real time, the machine learning model must be trained. The following sections will explain each stage individually, including the process of collecting training data and model training during development.

6.2 Signal Acquisition and Preprocessing

Before the random forest model can classify data, it first needs to learn how to recognise patterns; thus, the model needs to be trained. Two hours and ten minutes of EMG, EEG and EOG recordings were used to train the model. This section will explain the process of collecting, filtering and preprocessing the training data.

6.2.1 Data Collection

As explained in section 5.2, electrodes were placed on the right temple, above the eyebrows, and on the right cheek for data collection.

Figure 11 The electrode placement on the participant, two on the forehead, one on the cheek, one on the temple and two behind the ears.

The data collection phase involved repeating the same action, recording the resulting biosignals using the Ganglion, and marking the data when each event occurs. A Python script was developed to interface with the OpenBCI GUI and annotate recordings in real time. Certain buttons on the keyboard would correspond to certain actions; for example, the 'C' key would correspond to a jaw clench. Each time an action would occur, the button was pressed, marking the data. What this looked like on the terminal can be seen in figure 12. This was done for all actions: blinking, looking left, looking right, looking up, looking down, clenching the jaw, smiling and frowning. Although blinking did not correspond to a command for controlling the playback of music, the classifier was trained to recognise it so that an accidental blink would not be confused for another action. For recording alpha waves, the recording started after the eyes were closed and the mind entered a relaxed state, and ended before the eyes opened. The data was then marked after recording. This is because the action of pressing the button to mark the data could disrupt the restful state and reduce the alpha waves.

```
Marker Guide:
    b: blink (1)
    l: look left (2)
    r: look right (3)
    c: clench jaw (4)
    u: look up (5)
    d: look down (6)
    f: frustrated emotions (7)
    h: happy emotions (8)
Press 'q' to exit.

Blink marked!
Blink marked!
Blink marked!
Left look marked!
Left look marked!
Right look marked!
Up look marked!
Up look marked!
```

Figure 12 The marker script terminal output.

6.2.2 Filtering

The next stage was filtering the recorded data. Bioelectrical signals are very sensitive, which means that recording these signals can result in capturing unwanted electrical signals, like powerline interference, or artifacts if the electrode cables are tugged on accidentally. As a result, raw signals can sometimes contain noise and other artifacts which can obscure useful features and make pattern recognition difficult. Additionally, 'slow drift' can occur, which is when the baseline of the signal gradually changes over time. Filtering the signals reduces the noise and removes irrelevant components in the raw data, and corrects slow baseline drift (Teplan 2002).

Three filters were used. The first is a bandpass filter. It filters out frequencies below 0.5 Hz and above 30 Hz. This removes very high and very low frequencies, where electrical interference and motion artifacts tend to appear, leaving the part of the signal where relevant content tends to exist. The second is a notch filter. This is to remove powerline interference. In Wales, where this project was carried out, mains electricity runs at 50 Hz (UK Government 2013). The notch filter removes this interference. Although the bandpass filter filters this already, the notch filter is applied as a precaution, in case the signal is particularly strong and affects the signal in the passband. The third filter is a rolling mean filter. This takes a moving average of recent values of the signal, which is subtracted from the signal, “flattening” the baseline and removing baseline shift (Sörnmo and Laguna 2005).

Examples of a signal before and after filtering can be found in figures 13 and 14.

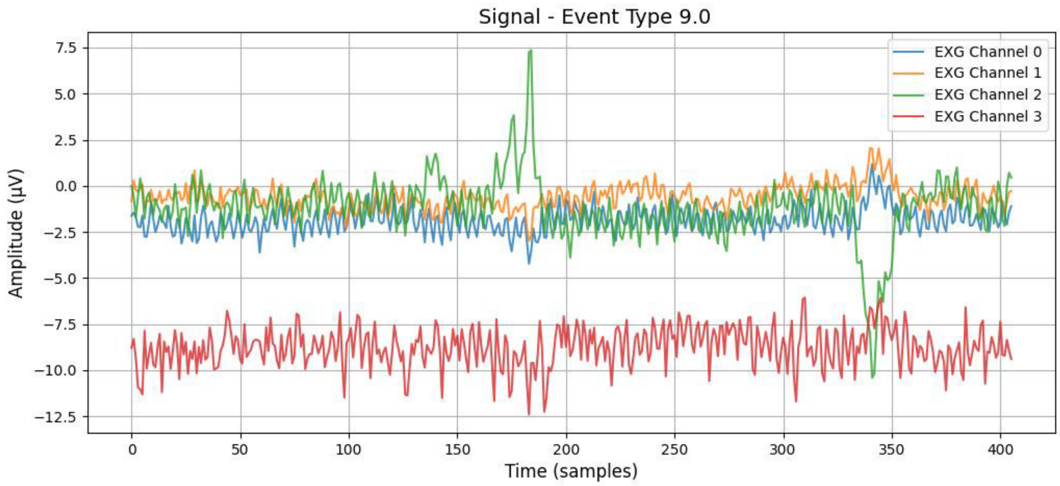

Figure 13 Alpha waves signal before filtering, the raw signal from all 4 channels.

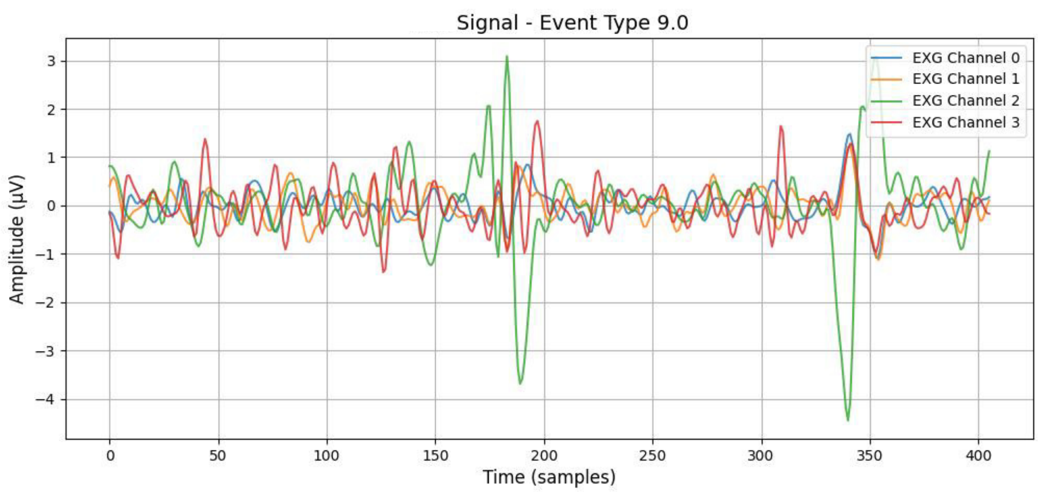

Figure 14 Alpha waves signal after filtering, all four channels.

6.2.3 Generating Epochs

Once the data is filtered, it is ready to be used to train the machine learning model.

The ML classifier needs to learn patterns from the input events, and so these events need to be isolated from the rest of the recorded data. Each event takes around two seconds to carry out. The training pipeline module searches through the training data, looking for the markers inserted earlier which indicate an event has occurred. A series of signal samples one second before and one second after the marked timestamp is extracted, which is called an epoch. It should be noted that the Ganglion board collects 200 samples a second, thus two seconds of data are shown in the figures. During live classification, the system classifies a two second window, and so the training epochs must consistently match this size. The random forest model needs each training example to be roughly the same length of time. The event epochs may overlap, though effort was taken during recording to not perform actions too close together. These epochs are assigned values corresponding to the action that created them:

1. Blinking

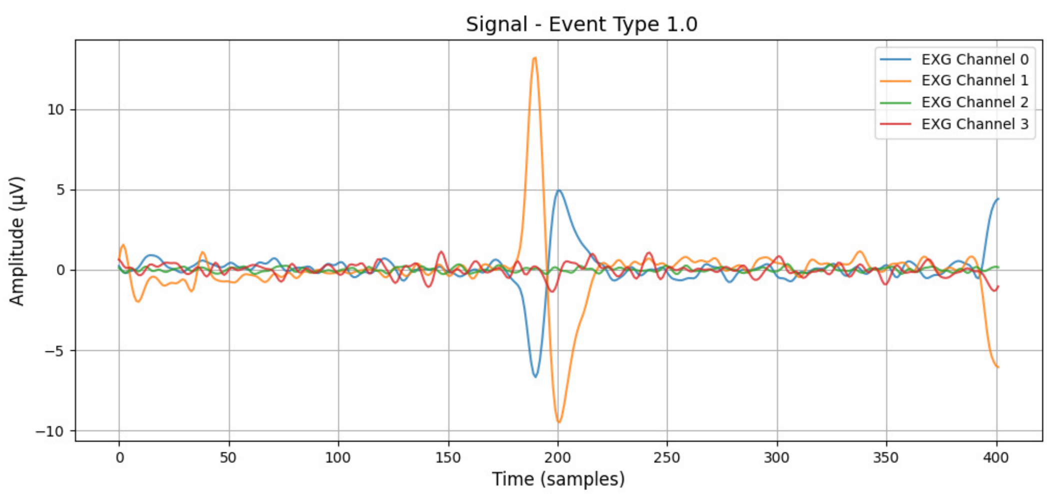

Figure 15 The signals generated by a blink.

Here in figure 15, there is a spike of activity in channel 1, corresponding to the electrode detecting EOG signals on the right eyebrow.

2. Looking left

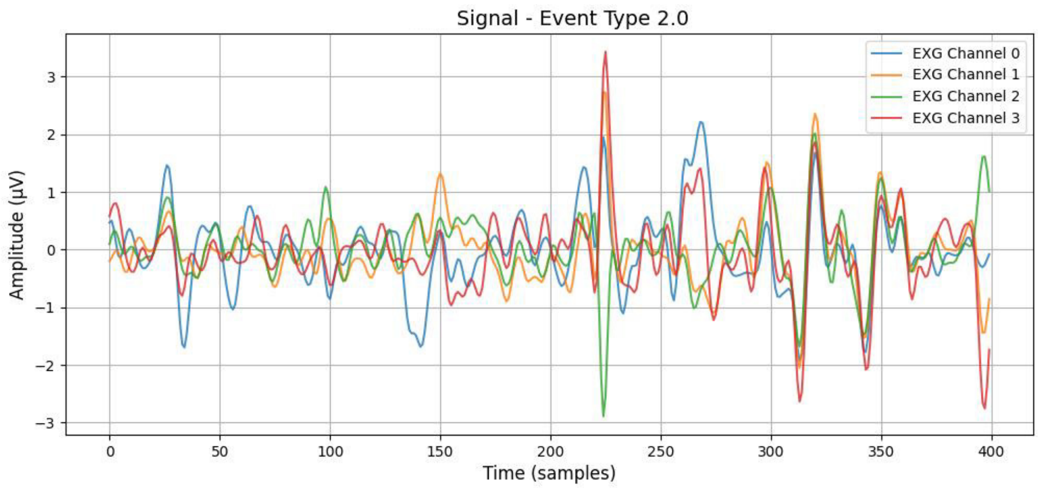

Figure 16 The signals generated by looking left.

3. Looking right

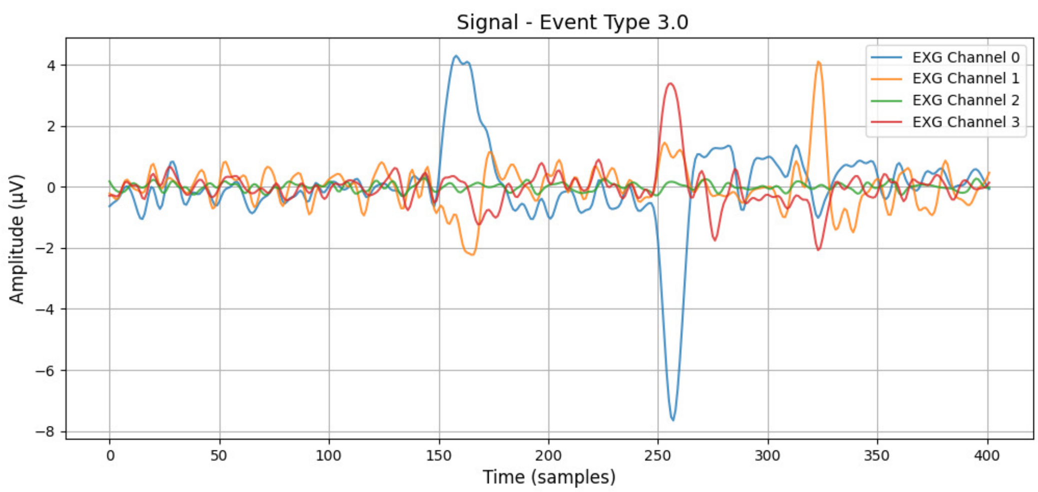

Figure 17 The signals generated by looking right.

It can be seen in figures 16 and 17, when looking left and right there is activity in channel 0, corresponding to the electrode on the right temple, detecting EOG signals. The two signals have the opposite polarity, for opposite directions of looking.

4. Clenching the jaw

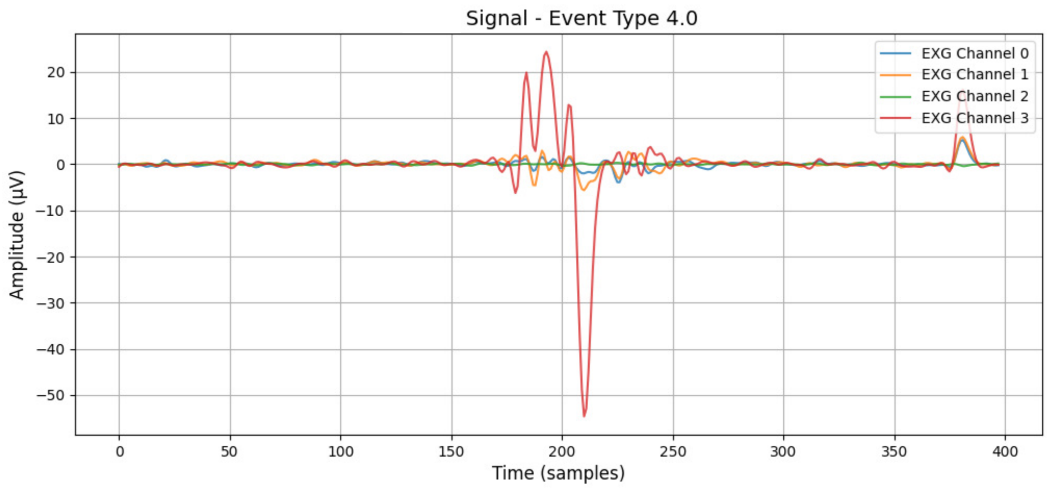

Figure 18 The signals generated by clenching the jaw.

In figure 18 a large spike of activity can be seen in channel 3, which corresponds to the electrode that sits on the cheek, near the mouth. This electrode detects EMG signals.

5. Looking up

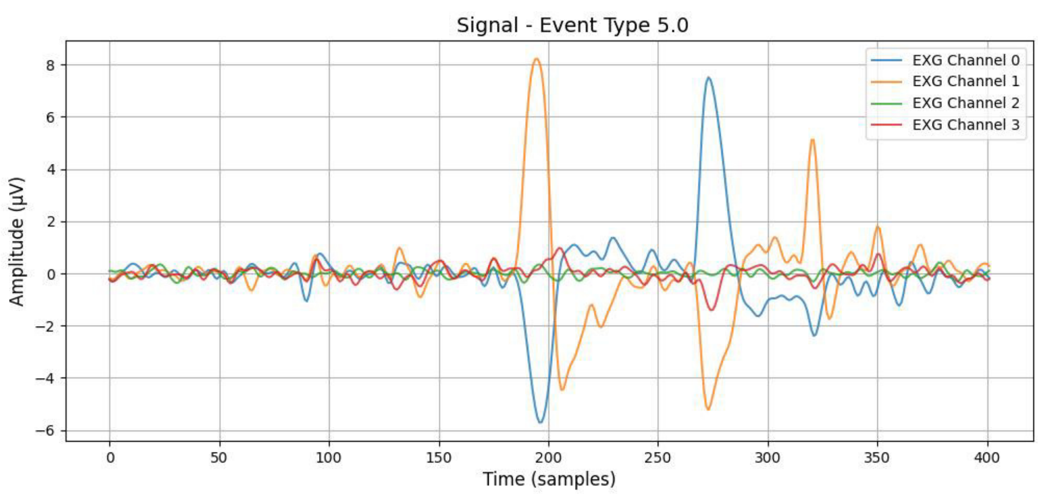

Figure 19 The signals generated by looking up.

6. Looking down

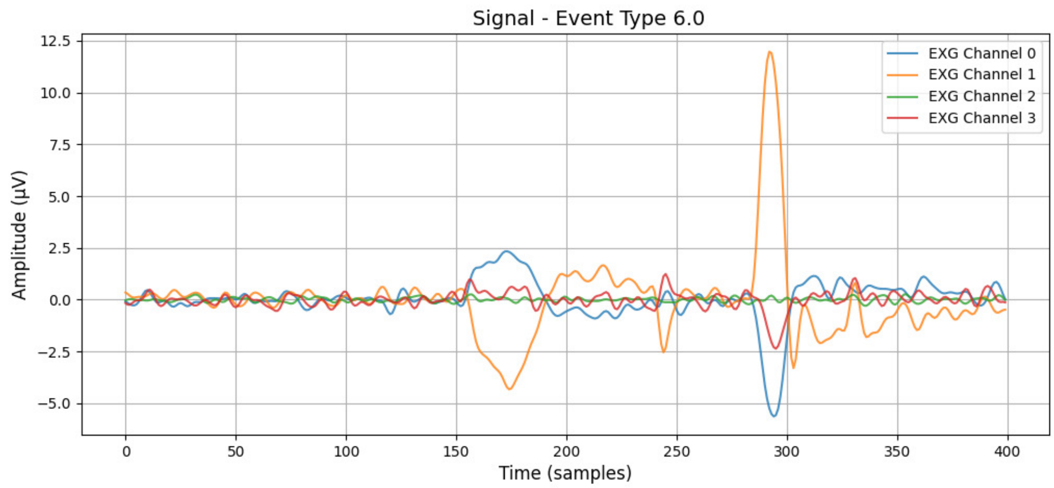

Figure 20 The signals generated by looking down.

In figures 19 and 20, when looking up and down spikes of activity can be seen in channels 1 and 0. In channel 0, the two actions have opposite polarities corresponding to the direction the participant is looking.

7. Smiling

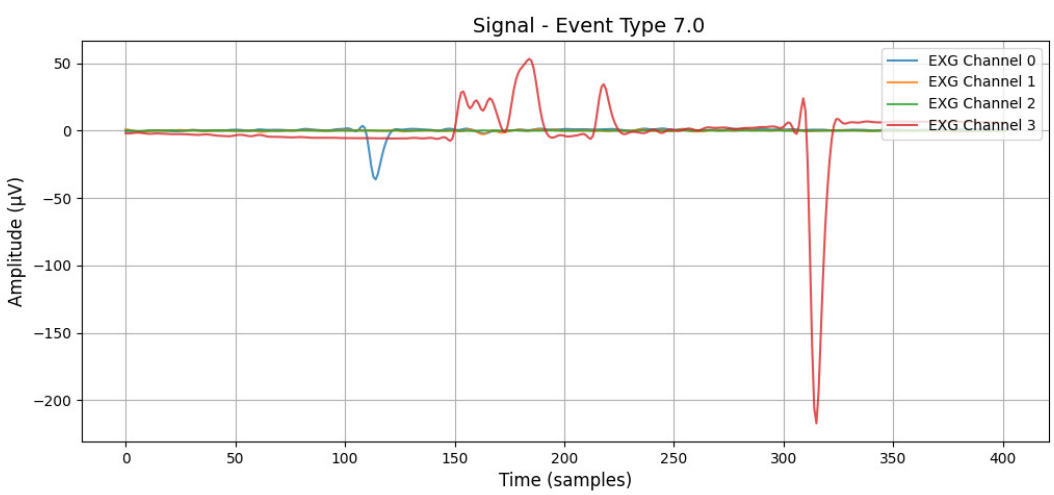

Figure 21 The signals generated by smiling.

In figure 21, when smiling a very large spike of activity can be seen in channel 3, corresponding to the electrode detecting EMG activity near the mouth.

8. Frowning

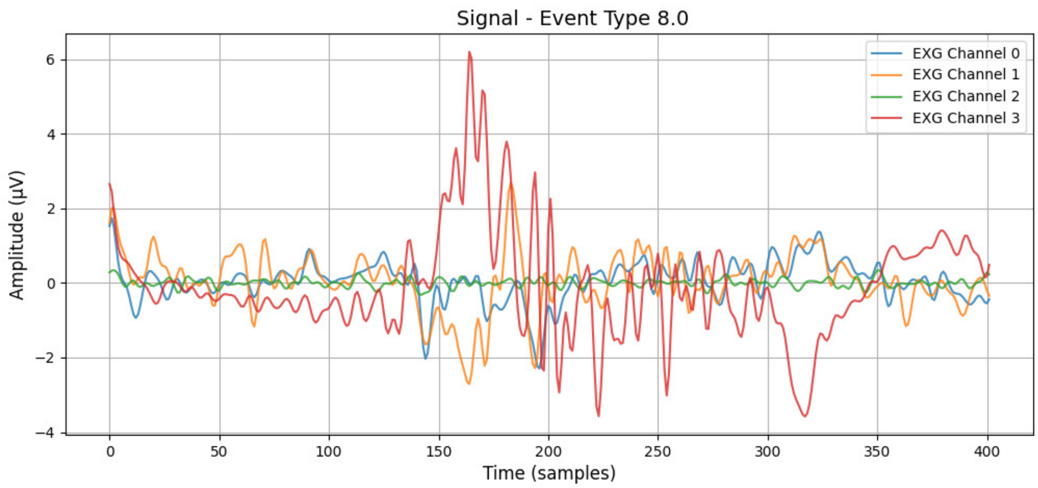

Figure 22 The signals generated by frowning.

In figure 22, when frowning a large spike of activity can be seen in channel 3, corresponding to the electrode detecting EMG activity near the mouth.

9. Alpha waves

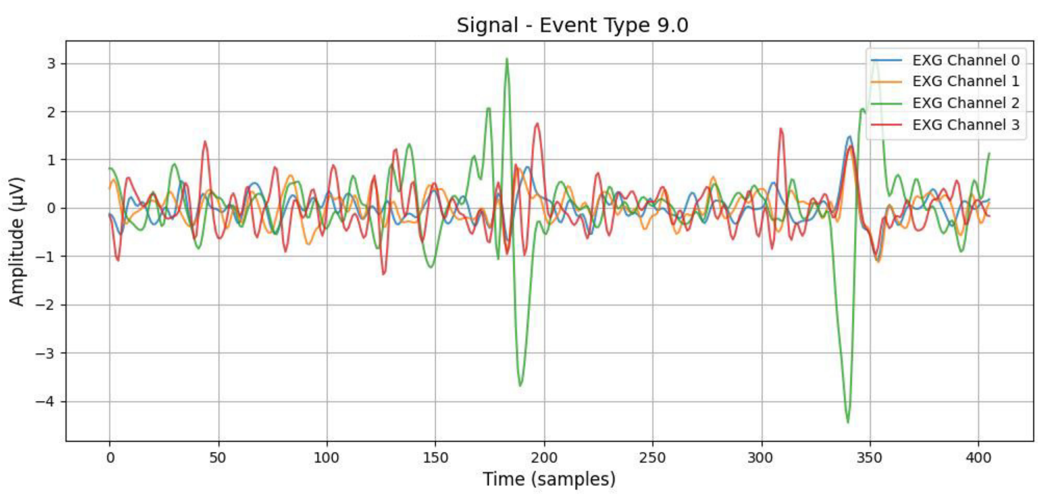

Figure 23 The signals generated by generating alpha waves.

In figure 23, when generating alpha waves, waves can be seen in channel 1, corresponding to the electrode on the left eyebrow, which collects EEG signals. Alpha waves can be detected in this area, and these waves are what are depicted in the figures. Other spikes could correspond to artifacts created by unintentional eye movements.

The ML classifier also needs to be able to know when no action is happening – when the user is idle. Additionally, the program scans the recording for windows of time where no event occurs. It looks for a signal with no marker to signify that an event occurred. It then checks to see if there is any overlap at all with an event. If it can be said that no event occurred in a two second window, it marks the window as "idle". This ensures that the model not only learns to recognise specific actions, but also learns to distinguish them from periods of no activity.

6.3 Feature Extraction

Raw EEG, EMG and EOG signals are complex, and difficult to analyse. The random forest classifier tends to perform poorly when raw time series data has been used to train it (Breiman 2001). Instead, only the most relevant information should be used to train the model. Feature extraction aims to reduce data complexity, otherwise known as data dimensionality (Subasi 2019). Identifying which information is important and most discriminating allows the model to recognise patterns more easily, leading to more accurate classification. The features extracted include the shape, variability and frequency content of the signal in an epoch.

Features that capture how a signal changes over time describe the time domain, while those that capture which frequencies a signal contains and how powerful they are describe the frequency domain.

Feature	Description	Domain	Formula (If applicable)
Mean	Average amplitude of the signal (per channel)	Time Domain	$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$
Variance	Measure of signal variability	Time Domain	$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$
Skewness	Asymmetry of the signal distribution	Time Domain	$$\text{Skewness} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^3$$

Kurtosis	“Tailedness” of the signal distribution	Time Domain	$\text{Kurtosis} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \mu}{\sigma}\right)^4$
RMS	Root Mean Square, gives an idea of signal power	Time Domain	$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Entropy	Signal unpredictability, useful for distinguishing idle	Time Domain	$H = -\sum_{i=1}^{n} p_i \log_2 p_i$
Zero Crossing Rate	Rate of sign changes, indicates signal frequency characteristics	Time Domain	$ZCR = \frac{1}{N-1}\sum_{i=1}^{N-1} 1[x_i \cdot x_{i+1} < 0]$
Rolling Std Dev	Standard deviation of rolling mean, indicates instability	Time Domain	$\sigma_{\text{rolling}} = \text{std}(x_{i:i+w})$
Mobility	Hjorth parameter: signal activity relative to variance	Time Domain (Hjorth)	$\text{Mobility} = \sqrt{\frac{\text{Var}(x')}{\text{Var}(x)}}$
Complexity	Hjorth parameter: change in frequency, signal irregularity	Time Domain (Hjorth)	$\text{Complexity} = \frac{\text{Mobility}(x')}{\text{Mobility}(x)}$
Peak-to-Peak	Amplitude range of the signal	Time Domain	$A_{\text{ptp}} = \max(x) - \min(x)$

Energy	Total signal energy (sum of squares)	Time Domain	$E = \sum_{i=1}^{N} x_i^2$
Delta Band Power	Power in 0.5–4 Hz, related to deep relaxation	Frequency Domain	$P_{\text{band}} = \sum_{f=f_1}^{f_2} \lvert X(f) \rvert^2$
Theta Band Power	Power in 4–8 Hz, related to drowsiness	Frequency Domain	$P_{\text{band}} = \sum_{f=f_1}^{f_2} \lvert X(f) \rvert^2$
Alpha Band Power	Power in 8–13 Hz, associated with relaxation	Frequency Domain	$P_{\text{band}} = \sum_{f=f_1}^{f_2} \lvert X(f) \rvert^2$
Beta Band Power	Power in 13–30 Hz, associated with alertness	Frequency Domain	$P_{\text{band}} = \sum_{f=f_1}^{f_2} \lvert X(f) \rvert^2$

Table 2 Extracted Features

Table 2 describes which features were extracted from the generated epochs, and which domain they belong to.

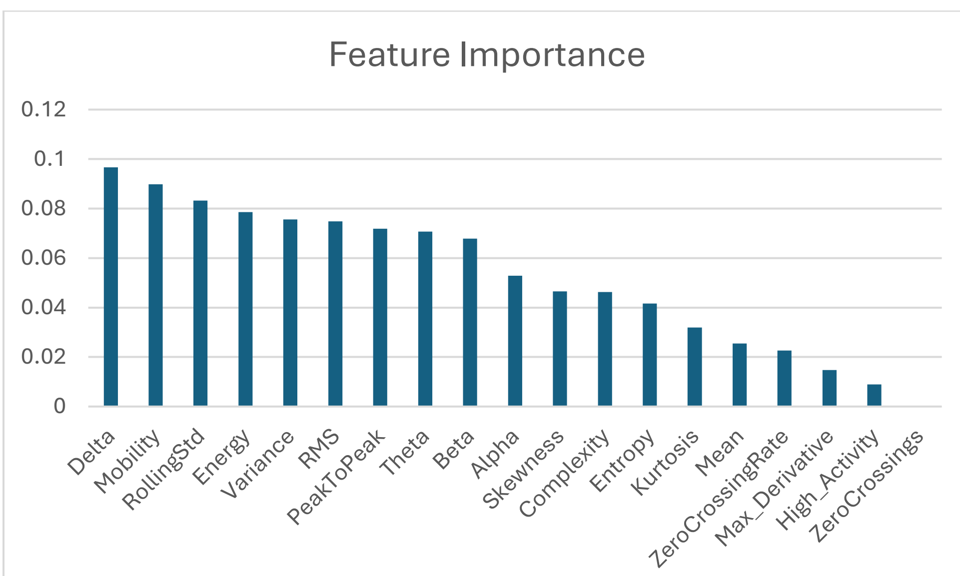

Figure 24 Feature importance bar graph.

Scikit-learn supports displaying which features are most important to the model in contributing to its decision making. Figure 24 ranks features by their importance to the random forest's decision making. Analysis showed that the most important feature extracted was channel 1's delta band power. Channel 1 corresponded to the signals gathered by the electrode at the right temple.

Additionally, we learn that the top eight most important features were extracted from channel 1. This indicates that the electrode at the right temple consistently captured the most discriminative signal patterns across the different actions. This is expected given that the placement allows detection of EMG from jaw clenching and also EOG from eye movements. This could create an issue where EOG and EMG signals get confused, if for example a user moved their eye and clenched their jaw simultaneously. This could interfere with accurate classification and so this should be explored in future work. It may also indicate that the other electrode placements were suboptimal, and the system could have benefited from alternative placements.

6.4 Model Training

The random forest model was trained on features extracted from two second epochs of labelled biosignal data. After extraction, the data was split into training and test sets. After training, performance was evaluated using K-Fold cross validation (Kohavi 1995).

6.4.1 Preprocessing

After data collection, there was an imbalance in class distribution. Most significantly, there were a large number of 'blink' samples. This is because blinks occur accidentally and involuntarily. To avoid the model being biased towards classes with more examples, the data was balanced using random undersampling. This is a method which decreases the size of the majority class in the dataset whilst keeping the data in the minority class. The Imbalanced-learn Python library provides a method for undersampling which was used in this project (Lemaıtre and Nogueira 2017). This reduced the number of epochs used from 12224 to 2496, a much lower number due to the low number of the least represented class.

Additionally, a scaler was used to standardise the range of the feature values. This scaler was saved and reused for real-time prediction. The scaler used was the RobustScaler provided by the Scikit-learn Python library (Pedregosa 2011). This centres the data by subtracting the median and scales the data by using the interquartile range. Biosignals can have occasional spikes, like a jaw clench or a blink, and scaling the data this way helps maintain signal consistency without being too influenced by these spikes.

6.4.2 Train/Test split

After preprocessing, the dataset was randomly split. 80% of the samples were used for training, and 20% were used for testing the model after training. It is important that the test set is not used for training, so that generalisation performance on unseen data can be evaluated, giving a more realistic estimate of real-world performance.

The train-test split is visualised in figure 25. Overall, 1996 samples were used for training, and 500 were used for testing.

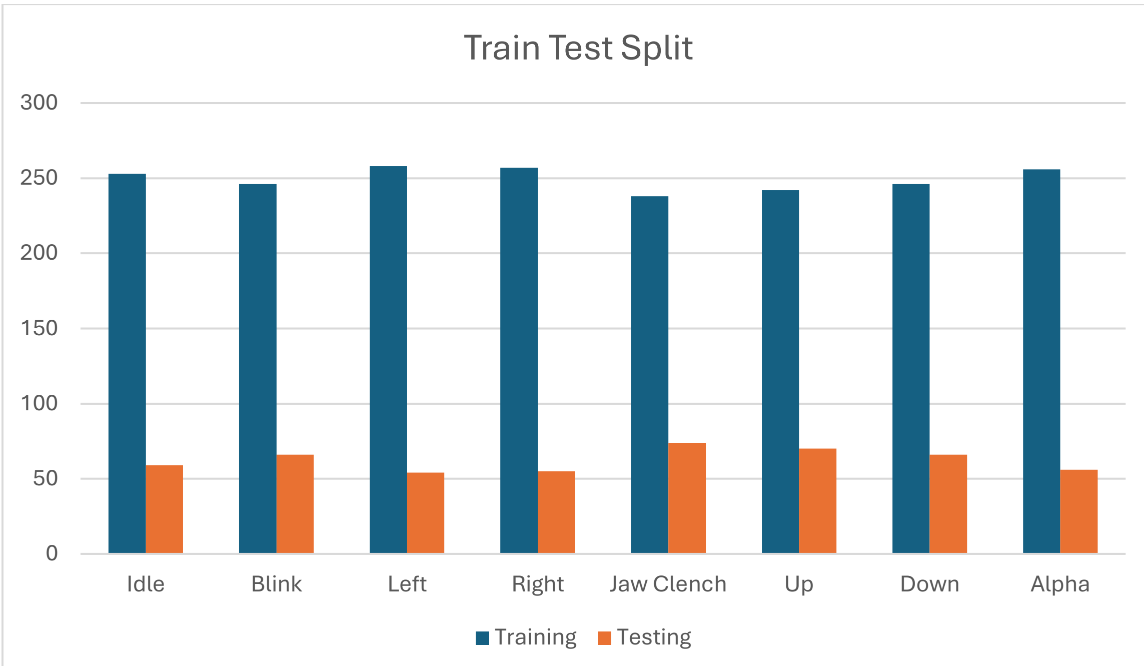

Figure 25 Train test split bar chart.

6.4.3 Model Configuration and Calibration

Hyperparameters are configuration values which control the behaviour of the random forest classifier. Tuning these parameters helps optimise the model's performance. A grid search was performed using Scikit-learn's GridSearchCV method (Pedregosa 2011). A grid search is a method for finding the best combination of hyperparameters for a model (Ogunsanya et al. 2023). A set of possible values for each hyperparameter is chosen, and the algorithm tries every possible combination of these values. The model created by each combination of values is evaluated via cross-validation, and the model with the best performance is selected, returning the best possible set of hyperparameters. The grid in this project tested 3 values for the number of decision trees, four values for the maximum depth of each decision tree, five values for the minimum number of samples required to create a new branch in the tree, four values for the number of samples which must be present in a leaf node, and two values for whether bootstrap samples or the full dataset should be used. This resulted in 480 combinations of hyperparameters, each trained and evaluated. The best possible combination was then used to train the model. Table 3 explains each hyperparameter and its value (Scikit-learn 2024).

Hyperparameter	Meaning	Effect	Value
min_samples_leaf	The minimum number of samples required to be in a leaf node.	Smaller leaves lead to complex trees but risk overfitting, larger values can smooth the model but may underfit.	1
n_estimators	The number of trees in the forest.	More trees improve accuracy but increase training time and memory usage	500
max_features	The number of features to consider when looking for the best split at a node.	Smaller subsets reduce correlation between trees, but too few can limit tree quality	Log2
max_depth	The maximum depth of each tree.	Deep trees can overfit, shallow trees can underfit.	None
min_samples_split	The minimum number of samples needed to split a node.	Small values capture more detail but may overfit, higher values may produce shallower tress and may underfit.	2

Table 3 Hyperparameters.

Additionally, Platt scaling was used to calibrate the classifier (Platt 1999). This method takes the outputs from a model like a random forest and applies a logistic regression model to map them to better calibrated probabilities, adjusting the confidence scores of a class to make them more realistic. This is important, as during real time classification, decisions are made based on how confident the model is that a certain action has occurred; for example even if the classifier returns that a jaw clench action has occurred, the system only executes the appropriate method if the confidence value is above a certain threshold.

6.4.4 Testing

After training, the model was evaluated using the 20% of data reserved for testing. This method generated the precision, recall, and f1-score of the model. Additionally, K-fold cross-validation was used to estimate the accuracy of the model (Kohavi 1995). K-fold cross-validation involves splitting the dataset into 'k' subsets, or folds, in this case 5, iteratively training the model on k-1 folds and validating it on the remaining fold. This is repeated k times, ensuring each fold is used as a validation set once. The final results are discussed further in section 7.

6.5 Real-Time Classification Pipeline

The real-time classification system, written in Python, operates in three daemon threads: data acquisition, signal classification, and music control. Each thread communicates via thread-safe queues. This structure is important because it ensures that the system can acquire more data from the Ganglion board whilst simultaneously classifying data it has already received, ensuring low latency and allowing the system to work in real time. The OpenBCI GUI interfaces with the Ganglion board, and streams data to the classification script via the Lab Streaming Layer (LSL) protocol (Kothe 2019). The LSL protocol was chosen for its buffer; it will store an amount of data if there is some latency in system processing.

6.5.1 Data Acquisition

Data is streamed from the OpenBCI Ganglion board using LSL and received by a dedicated thread. Since the random forest model was trained on features extracted from two second epochs, the buffer holds the most recent two seconds of ExG data. To avoid missing events that might occur too late in the captured window, each window overlaps with the next; the buffer updates every 0.45 seconds, removing the oldest 90 samples and allowing 90 newer samples to populate the buffer. This update rate was chosen as the fastest achievable without introducing noticeable latency.

6.5.2 Signal Classification

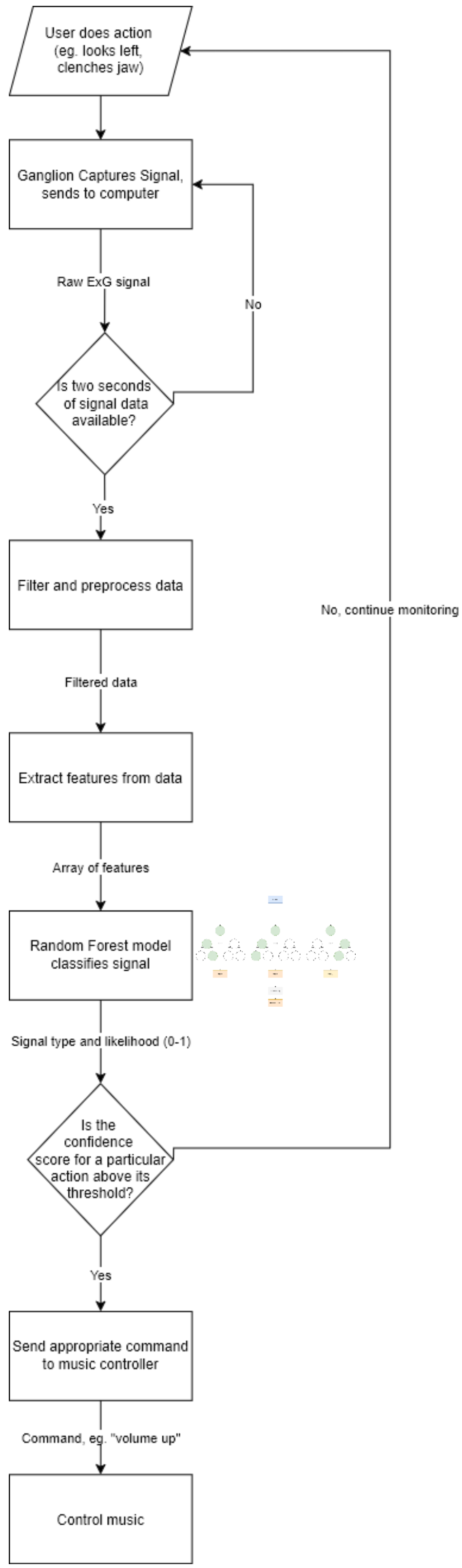

Figure 26 Flow of data during signal classification

Once the buffer is full, it is passed to the signal classification thread. The data is passed to the same filtering pipeline used when training the model. Then, features are extracted from the filtered data in the same format. The RobustScaler used and saved earlier is also applied to the data, to match training conditions. The filters and scaler are discussed in detail in section 6.2.

The scaled features are then passed to the trained random forest model. The model predicts the probabilities of the 2 second window containing any of the events. The system checks if the maximum probability exceeds a class specific threshold. If it does, it is decided that the classifier is confident enough that an action has occurred in the two second window. If not, the prediction is discarded.

To reduce false positives, blinks are ignored as actionable commands. Additionally, due to the nature of the rolling window, two consecutive predictions could be for the same one action. To avoid repeated commands, predictions only trigger actions for music control if no action had occurred within the past 2.2 seconds, or if the current prediction differs from the last and no action had occurred in the past 1 second.

Additionally, though the classifier estimates the probability of the user generating alpha waves, this alone does not trigger control. Instead, alpha wave activity only enables control if the classifier also detects a jaw clench, and the alpha probability exceeds a predefined threshold. This is done to avoid any accidental control of the system using alpha waves, as the user has less voluntary control over their brain waves compared to their facial expressions.

6.5.3 Application

The trained classifier can be used for a multitude of applications. It has been chosen that this project's usefulness will be demonstrated by controlling the playback of music using biosignals.

Once a confident enough prediction is made, the associated command is passed to the music control thread. Here, music is controlled via a custom class created using the PyGame library. The actions performed by the user control the music in the following way in table 4.

Action	Result
Looking left	Track is changed to the previous song in the playlist.
Looking right	Track is changed to the next song in the playlist.
Looking up	Volume increases (by 0.1, to a maximum value of 1).
Looking down	Volume decreases (by 0.1, to a minimum value of 0).
Jaw is clenched	The playback pauses if a song is playing, or resumes if the song has been paused.
Jaw is clenched while alpha waves are detected	The playlist switches between either an upbeat selection of songs, or a sadder selection of songs.

Table 4 Actions and playback result.

The music used in this project was sourced from the YouTube Studio Audio library, which is a library of royalty free music.

6.6 Ethical Considerations

This study involved only one participant, the author, who provided their informed consent. Data collection and testing was conducted solely with the author. No personal or medical information was shared externally.

Data collection was non-invasive and posed no risk to the participant. EEG use has been associated with a small risk of causing seizures in people who are predisposed to having them, such as those who suffer with epilepsy, but the participant had no known neurological conditions.

This study had the goal of creating an accessible interface for people living with disabilities, but the only participant had no disabilities. Though it was decided that gaining ethical approval and finding disabled participants was not feasible considering the project's short timeframe, this is nonetheless a limitation in the study, and is discussed further in sections 7 and 8.

7. Results and Evaluation

This section will cover the accuracy of the project, as well as its practicality as an accommodation for disabled people, and its limitations.

7.1 Accuracy

	PRECISION	RECALL	F1-SCORE	SUPPORT
IDLE (-1.0)	0.92	0.95	0.93	59
BLINK (1.0)	0.98	0.94	0.96	66
LOOK LEFT (2.0)	0.98	0.94	0.96	54
LOOK RIGHT (3.0)	0.98	0.98	0.98	55
CLENCH JAW (4.0)	0.97	0.97	0.97	74
LOOK UP (5.0)	0.97	0.99	0.98	70
LOOK DOWN (6.0)	0.99	1.0	0.99	66
ALPHA WAVES (9.0)	0.98	1.0	0.99	56
ACCURACY			0.97	500
MACRO AVERAGE	0.97	0.97	0.97	500
WEIGHTED AVERAGE	0.97	0.97	0.97	500

Table 5 Model classification report

The system was evaluated using a variety of means. Firstly, a classification report was generated using Scikit-learn. This provided a detailed report on how accurate the model was at predicting each class. This report can be found in table 5.

Class -1.0 is “idle”, where the user has performed no action. 1.0 is blinking. 2.0 is looking left. 3.0 is looking right. 4.0 is clenching the jaw. 5.0 is looking up. 6.0 is looking down. 9.0 is where the user has tried to generate alpha waves.

One of the aims of this project was to evaluate how accurate the system is. Precision measures how many of the model’s predictions for a given class were actually correct. A high precision score means few false positives, indicating that the prediction for that class is reliable. The recall value measures what percentage of each class was correctly predicted. A high recall score means few false negatives, and it indicates how complete the model is for a particular class. The f1-score is the harmonic mean of precision and recall, balancing false positives and false negatives, giving a summary of overall class performance.

From this, we can see that the model struggled to classify class -1.0, or "idle". This may be attributed to the reduced number of informative features in idle-state signals. The model performed best for classes 6.0 and 9.0 (looking down and generating alpha waves). This may be because these actions produce distinct biosignal patterns. Eye movements create very strong signals, and alpha waves are a well-defined frequency band, a feature that is extracted specifically from the signal. Additionally, 5-fold cross-validation was used to create an overall accuracy score of the model. The details of how this process works can be found in section 6.4.4. This resulted in an average accuracy score of 95.71%, with a standard deviation of ±0.79%, indicating stable performance across different folds and strong classification accuracy. High accuracy implies that the quality of the recorded biosignal data is good (Lee et al. 2024). The model's accuracy is higher than some other biosignal-based solutions; for example, the BCI controlled wheelchair described in 5.5.2 had an accuracy of only 87% (Kanungo et al. 2021).

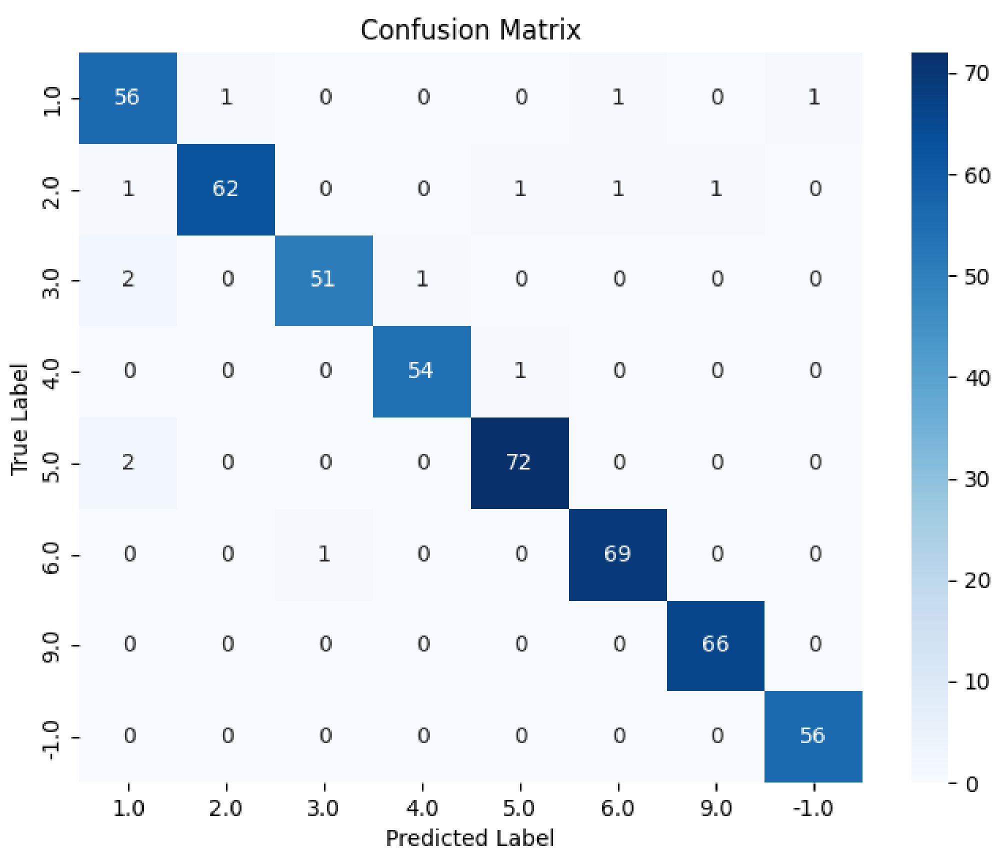

Figure 27 Model confusion matrix

Additionally, figure 27 shows the confusion matrix created using Scikit-learn. Most values lie on the diagonal, meaning that the model is accurately predicting each class, and no one class was commonly mistaken for another.

7.2 Practicality

Setting up the system for live classification took fewer than ten minutes. The system was set up independently by the participant without assistance, though it is important to note that the participant was not disabled. A person with cerebral palsy or other disabilities may find it challenging to set up on their own; due to the specific placement of the electrodes, they require fine motor skills and the use of a person's hands to apply.

The trained random forest model occupies approximately 135 MB when serialized. While this is suitable for execution on a general-purpose laptop such as the one used for development, it may currently be too large for use on smaller embedded systems such as Arduinos without modification. It is likely small enough for use on a microcomputer such as a Raspberry Pi. Future work could involve reducing model size by limiting tree depth or reducing the number of estimators.

The system achieves its objectives: it allows a user to control the playback of music using their eyes, brain and facial muscles. A demonstration of this was recorded and submitted along with this report[1] (*Controlling IoT devices with our brain, eyes and facial expressions* 2025). The demonstration video shows low latency, the time between the participant performing an action and the corresponding command being carried out is around a second.

7.3 Limitations

Though the system meets its objective, it does not do this without limitations. As mentioned in section 6.6, this study had only one participant. This makes it difficult to generalise results; it is possible that the system may not perform as well on other individuals. Additionally, there was a lack of testing with target users. Though the system was designed with improving accessibility for people with disabilities, the participant did not have a disability, and so the system's real-world effectiveness is unverified.

Electrode placement also requires fine motor control, which the target user might not have, meaning that the user would have to rely on another individual to help them. This does not increase independence the way the study hoped.

[1] https://youtu.be/WC70dQBcvec

The study originally planned to use the ANT Neuro EEGO Sports BCI cap rather than the OpenBCI Ganglion Board (ANT Neuro 2024; OpenBCI 2024). However, it was deemed that using the cap would be impractical; though it would have made detecting EEG signals easier, setting up the cap takes much longer than setting up the Ganglion. The cap requires two people to set up, and requires electrode gel, which is unpleasant for the participant and must be cleaned off the electrodes after use. The Ganglion board can be set up by the user, and the adhesive electrodes are simple to apply and remove.

Additionally, it was found in testing that the user must remain very still whilst using the system. If the wires attaching the electrodes to the Ganglion board are moved, the board interprets this as a signal, confusing the system. In the demonstration video, the participant is shown holding these wires to prevent this from happening. This may be impractical for users who have little control over their movements, or for users who want to move around whilst listening to music. However, the system may still be practical for users with some other disabilities, like those who are paralysed below the neck. Had the ANT Neuro EEGO Sports BCI cap been chosen for this project instead of the OpenBCI Ganglion board, this problem would have been mitigated, as the cap sits on the head and thus does not have trailing wires which can be moved. Future work could explore how alternative BCIs could fix this problem.

Some of the actions used as inputs in this project are easy to do accidentally. Tracks are changed by looking left or right, which users will want to do even if they don't want to change the music. This can be mitigated by creating a method for preparing the system to receive an input, such as squeezing a fist whilst looking left or right, but this has not yet been implemented.

The study originally planned to deploy the system by streaming biosignals from the Ganglion board directly to a Raspberry Pi Zero W, rather than using the OpenBCI GUI on a laptop. A script was developed for this purpose; however, the necessary libraries for the Raspberry Pi to connect to the Ganglion via Bluetooth Low Energy (BLE) could not be found or configured within the project's timeframe. As a result, the system was developed and tested on a laptop computer instead.

8. Conclusions and Future Work

The aims of this project were as follows:

1. Collect and annotate EEG, EOG and EMG signals.

As shown in section 6.2, this aim has been achieved. Using the OpenBCI Ganglion board, EEG signals in the form of alpha waves, EOG signals in the form of looking up, down, left, right, and blinking, and EMG signals in the form of jaw clenching. As seen in figure 12, a Python script was developed to annotate these signals during data collection.

2. Prepare collected biosignals.

As shown in sections 6.2.2 and 6.2.3, the captured biosignals were successfully filtered and segmented into two second epochs. Following that, time and frequency domain features were extracted from the raw signals, as discussed in section 6.3.

3. Train a random forest classifier to detect user actions.

Section 6.4 covers how the random forest classifier was trained to detect user actions, including how sample features were separated to train and test the classifier.

4. Demonstrate the random forest classifier by implementing real-time prediction.

As seen in the demonstration video [2], the random forest classifier successfully predicts user input in real time. In the scenario discussed in section 6.5.3, the predictions have been shown to control the playback of music.

5. Evaluate system accuracy.

As discussed in section 7.1, the accuracy of the random forest classifier was found using 5-fold cross-validation to be 95.71%, with a standard deviation of ±0.79%, indicating stable performance across different folds and strong classification accuracy. The confusion matrix found in figure 27 also shows that the model accurately predicts each class.

Because of these reasons, it can be said that this project has met its aims.

The project was limited in that the study was tested using only one participant, who did not have a disability, making generalisation difficult. If this project were to continue, testing with a broader participant group, including with people who have disabilities, would be necessary. Furthermore, future work could include deploying the system on a Raspberry Pi or similar single-board computer, making the system more practical for everyday use.

[2] https://youtu.be/WC70dQBcvec

Additionally, future work could explore using other BCIs, such as the ANT Neuro EEGO Sports BCI cap.

Though the project demonstrated the use of using biosignals to control a device by using the signals as inputs for controlling the playback of music, there are many other possible use cases for such a technology. For example, controlling a ‘smart home’ using the brain, eyes and facial expressions could be very useful in making a disabled person’s living space more accessible. A person who requires the use of a wheelchair might find it taxing to travel to a light switch, if they are already in bed for example, but this technology could allow them to turn their light off by clenching their jaw or blinking. It could allow a person with cerebral palsy to turn on a coffee machine without moving their limbs, they could look left or right a few times to activate it. Similarly to how the Earable technology has been demonstrated (Pham et al. 2019), a person’s alpha waves could be detected and used to control a ‘dimmer’ switch in a person’s bedroom when they feel tired or relaxed.

Alternatively, the technology could be used to control a wheelchair or similar mobility aid, similar to the approach described in section 5.5.2 (Kanungo et al. 2021). The project as it is now supports four directions controlled by eye movements – up, down, left and right, which could correspond to the direction the wheelchair user wants to move.

The brain, eyes and facial expressions could also be used to control other kinds of digital applications. This could include controlling a mobile phone, as some disabled individuals may find using their fingers to control the screen difficult or even impossible. This could allow them to make calls, use social media or browse the internet without using voice commands.

In conclusion, the project has been demonstrated to show promise as a possible accommodation for disabled people. The biosensing and machine learning technologies explored and developed could evolve into any number of accommodations for disabled people in future, with music playback control being just one possible use.

9. Reflection on Learning

This section reflects on the technical and personal learning outcomes of the project, including insights gained in machine learning, brain-computer interfacing, and my broader development as a computer scientist.

During my time at university, I have undertaken only one module that focused on machine learning, Informatics. As part of the coursework, I created a simple image classifier using the approaches taught in lectures, as well as analysing a classifier given for biases. Of all the courses I enrolled in over the three years, this was one of my favourites, which influenced me to choose to work on this project. I am pleased that this project has given me the opportunity to learn more about machine learning, I now understand additional approaches, such as the random forest approach used in this project as well as other approaches I considered during early development. However, the approaches taught in the Informatics module as well as the random forest approach are supervised. I have knowledge of how unsupervised learning works, but unfortunately no experience in using it. I could have implemented an unsupervised approach to consolidate my learning, as mentioned in section 5.3, but I decided that this was the wrong choice for the project. Regardless, I am glad that I now have experience with the Python library Scikit-learn, which I feel will be invaluable in my future career.

Though I knew before choosing the topic for this project that I wanted to expand my knowledge about machine learning, before development, I had no experience in working with brain-computer interfaces. I had originally thought that biosignals were difficult to classify; that they might not have a use outside of medicine due to preconceived ideas stemming from the fact that there are not many commercial uses for them currently. I learned that this is not the case, and I am now excited to see where the study of this topic will go in future, and whether in the near future, people will be controlling their homes with their facial expressions and thoughts. Learning more about this topic has broadened the horizons of my learning, and I am grateful that I was able to learn about something that I might otherwise never have been able to study. Though I do not feel that working with these technologies is necessarily something that I would seek to do in my future career, I remain open to the possibility.

One of the most challenging aspects of development was diagnosing issues with classification. At the start of development, the model was no more than 60% accurate. This meant that using it to classify signals would have been only marginally better than chance. I learned to debug iteratively, testing and isolating components one by one. I found that the training data I had been using was sometimes low quality, and I learned that this was because my mobile phone was sending Bluetooth signals which were interfering with the

OpenBCI Ganglion. Additionally, I tested the classification of each action one by one and found that the classifier found it difficult to tell right and left eye movements apart, and so I used Hjorth parameters to extract features which would allow the classifier to better differentiate between the two. As part of university modules, I learned good programming techniques, particularly modularisation. I am grateful for this, as it taught me to split each step of preprocessing, such as filtering or feature extraction, into its own module, making debugging much easier.

Throughout my academic career, including during my placement in industry, I have struggled with my time management skills. Knowing that this was a weakness of mine, I decided that I would have to focus on meeting self-imposed deadlines and sticking to a timeline of tasks and milestones. At the beginning of development, I created a weekly plan outlining which tasks, deliverables and milestones needed to have been completed and by when. I have largely stuck to this plan throughout development. I found that planning specifically what I needed to achieve and when improved my time management skills greatly. When I found that I had completed certain tasks ahead of schedule, I decided that I would attempt to deploy the project on a Raspberry Pi. However, I found that I had not set aside enough time to do so, as it was not a part of the original plan. I was unable to finish deployment once unexpected roadblocks appeared, namely that the Raspberry Pi Zero W did not have the BLE libraries required to communicate with the OpenBCI Ganglion. I had to make the decision that working on deployment any longer would negatively affect my report writing, and so I chose to stop. In future, I would create more than one plan at the start of development, one optimistic and one more conservative.

Overall, completing this project has significantly developed my technical and personal skills. I have improved upon my knowledge of machine learning, now to the point that I am considering a career using it. My understanding of biosignal processing has strengthened, as well as my ability to plan and manage a long-term project independently. While I encountered limitations in scope, I now have a clearer understanding of how to structure iterative development, debug complex pipelines, and manage uncertainty in self-directed work. In future work, I will begin evaluation testing even earlier to ensure the results can shape development decisions.

10. References

Amzica, F. and Lopes Da Silva, F.H. 2017. *Cellular Substrates of Brain Rhythms*. Schomer, D. L. and Lopes Da Silva, F. H. eds. Oxford University Press. Available at: https://academic.oup.com/book/35515/chapter/305245903 [Accessed: 7 May 2025].

ANT Neuro. 2024. *eego™ sports - Mobile EEG recording system*. Available at: eego™ sports - Mobile EEG recording system [Accessed: 27 April 2025].

Breiman, L. 2001. Random Forests. *Machine Learning* 45(1), pp. 5–32. doi: 10.1023/A:1010933404324.

Britton, J., Frey, L. and Hopp, J. 2016. *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants.* Chicago: American Epilepsy Society.

Bulling, A., Roggen, D. and Tröster, G. 2008. It’s in your eyes: towards context-awareness and mobile HCI using wearable EOG goggles. In: *Proceedings of the 10th international conference on Ubiquitous computing*. Seoul Korea: ACM, pp. 84–93. Available at: https://dl.acm.org/doi/10.1145/1409635.1409647 [Accessed: 7 May 2025].

Controlling IoT devices with our brain, eyes and facial expressions. 2025. Available at: https://youtu.be/WC70dQBcvec.

Craik, A., He, Y. and Contreras-Vidal, J.L. 2019. Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering* 16(3), p. 031001. doi: 10.1088/1741-2552/ab0ab5.

Farwell, L.A. and Donchin, E. 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology* 70(6), pp. 510–523. doi: 10.1016/0013-4694(88)90149-6.

Han, S., Williamson, B.D. and Fong, Y. 2021. Improving random forest predictions in small datasets from two-phase sampling designs. *BMC Medical Informatics and Decision Making* 21(1), p. 322. doi: 10.1186/s12911-021-01688-3.

Heaton, J. 2018. Deep Learning. *Genetic Programming and Evolvable Machines* 19(1–2), pp. 305–307. doi: 10.1007/s10710-017-9314-z.

Hu, J., Wang, C., Wu, M., Du, Y., He, Y. and She, J. 2015. Removal of EOG and EMG artifacts from EEG using combination of functional link neural network and adaptive neural fuzzy inference system. *Neurocomputing* 151, pp. 278–287. doi: 10.1016/j.neucom.2014.09.040.

Kanungo, L., Garg, N., Bhobe, A., Rajguru, S. and Baths, V. 2021. Wheelchair automation by a hybrid BCI system using SSVEP and eye blinks. Available at: https://arxiv.org/abs/2106.11008 [Accessed: 6 May 2025].

Kawala-Sterniuk, A. et al. 2021. Summary of over Fifty Years with Brain-Computer Interfaces—A Review. *Brain Sciences* 11(1), p. 43. doi: 10.3390/brainsci11010043.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*. pp. 1137–1143.

Kothe, C.A. 2019. *Lab Streaming Layer*. Available at: https://labstreaminglayer.readthedocs.io/index.html [Accessed: 28 April 2025].

Krucoff, M.O., Rahimpour, S., Slutzky, M.W., Edgerton, V.R. and Turner, D.A. 2016. Enhancing Nervous System Recovery through Neurobiologics, Neural Interface Training, and Neurorehabilitation. *Frontiers in Neuroscience* 10. Available at: http://journal.frontiersin.org/article/10.3389/fnins.2016.00584/full [Accessed: 17 April 2025].

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), pp. 2278–2324. doi: 10.1109/5.726791.

Lee, Y.J., Park, C., Kim, H., Cho, S.J. and Yeo, W.-H. 2024. Artificial intelligence on biomedical signals: technologies, applications, and future directions. *Med-X* 2(1), p. 25. doi: 10.1007/s44258-024-00043-1.

Lemaıtre, G. and Nogueira, F. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.

Majaranta, P. and Räihä, K.-J. 2002. Twenty years of eye typing: systems and design issues. In: *Proceedings of the symposium on Eye tracking research & applications - ETRA '02*. New Orleans, Louisiana: ACM Press, p. 15. Available at: http://portal.acm.org/citation.cfm?doid=507072.507076 [Accessed: 28 April 2025].

Mak, J.N. and Wolpaw, J.R. 2009. Clinical Applications of Brain-Computer Interfaces: Current State and Future Prospects. *IEEE Reviews in Biomedical Engineering* 2, pp. 187–199. doi: 10.1109/RBME.2009.2035356.

Maltron. 2024. *Expanded Keyboard for Users with Disabilities*. Available at: maltron.com [Accessed: 27 April 2025].

Nazmi, N., Abdul Rahman, M., Yamamoto, S.-I., Ahmad, S., Zamzuri, H. and Mazlan, S. 2016. A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions. *Sensors* 16(8), p. 1304. doi: 10.3390/s16081304.

NHS. 2023. *Cerebral Palsy*. Available at: https://www.nhs.uk/conditions/cerebral-palsy/ [Accessed: 27 April 2025].

Nicolas-Alonso, L.F. and Gomez-Gil, J. 2012. Brain Computer Interfaces, a Review. *Sensors* 12(2), pp. 1211–1279. doi: 10.3390/s120201211.

Nuwer, M.R. et al. 1998. IFCN standards for digital recording of clinical EEG. *Electroencephalography and Clinical Neurophysiology* 106(3), pp. 259–261. doi: 10.1016/S0013-4694(97)00106-5.

Office for National Statistics. 2021. Disability, England and Wales census.

Ogunsanya, M., Isichei, J. and Desai, S. 2023. Grid search hyperparameter tuning in additive manufacturing processes. *Manufacturing Letters* 35, pp. 1031–1042. doi: 10.1016/j.mfglet.2023.08.056.

OpenBCI. 2024. *Ganglion Board*. Available at: https://docs.openbci.com/Ganglion/GanglionLanding/ [Accessed: 28 April 2025].

OpenBCI. 2025. *OpenBCI User Guide*. Available at: https://docs.openbci.com/Software/OpenBCISoftware/GUIDocs/ [Accessed: 28 April 2025].

Pedregosa, F. 2011. Scikit-learn: Machine Learning in Python. *MACHINE LEARNING IN PYTHON*.

Pennington, L. 2012. Speech and communication in cerebral palsy. *Eastern J Med* 17, pp. 171–177.

Pham, N., Kim, T., Thayer, F.M., Nguyen, A. and Vu, T. 2019. Earable -- An Ear-Worn Biosignal Sensing Platform for Cognitive State Monitoring and Human-Computer Interaction (demo). In: *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. Seoul Republic of Korea: ACM, pp. 685–686. Available at: https://dl.acm.org/doi/10.1145/3307334.3328582 [Accessed: 22 April 2025].

Platt, J. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers* MIT Press, pp. 66–74.

Prinzie, A. and Van Den Poel, D. 2008. Random Forests for multiclass classification: Random MultiNomial Logit. *Expert Systems with Applications* 34(3), pp. 1721–1732. doi: 10.1016/j.eswa.2007.01.029.

PyGame. 2024. *PyGame - Python Game Development*. Available at: www.pygame.org [Accessed: 27 April 2025].

Robertson, D.G.E. ed. 2014. *Research methods in biomechanics*. Second edition. Champaign, IL: Human Kinetics. doi: 10.5040/9781492595809.

Rokach, L. and Maimon, O. 2005. Top-Down Induction of Decision Trees Classifiers—A Survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 35(4), pp. 476–487. doi: 10.1109/TSMCC.2004.843247.

Rosenbaum, P.L. 2007. The Definition and Classification of Cerebral Palsy. *Developmental Medicine & Child Neurology* 49, pp. 1–44. doi: 10.1111/j.1469-8749.2007.00001.x.

Saavedra, S., Karve, S.J., Woollacott, M. and Van Donkelaar, P. 2009. Eye hand coordination in children with cerebral palsy. *Experimental Brain Research* 192(2), pp. 155–165. doi: 10.1007/s00221-008-1549-8.

Salahuddin, Z., Woodruff, H.C., Chatterjee, A. and Lambin, P. 2022. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Computers in Biology and Medicine* 140, p. 105111. doi: 10.1016/j.compbiomed.2021.105111.

Schomer, D.L. and Lopes Da Silva, F.H. eds. 2017. *Niedermeyer's Electroencephalography*. Oxford University Press. Available at: https://academic.oup.com/book/35515 [Accessed: 28 April 2025].

Scikit-learn. 2024. *sklearn.ensemble.RandomForestClassifier*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accessed: 8 May 2025].

Scope. 2024. *Cerebral Palsy Introduction*. Available at: https://www.scope.org.uk/advice-and-support/cerebral-palsy-introduction [Accessed: 6 May 2025].

Sharma, K., Jain, N. and Pal, P.K. 2020. Detection of eye closing/opening from EOG and its application in robotic arm control. *Biocybernetics and Biomedical Engineering* 40(1), pp. 173–186. doi: 10.1016/j.bbe.2019.10.004.

Sörnmo, L. and Laguna, P. 2005. *Bioelectrical signal processing in cardiac and neurological applications*. Amsterdam Boston: Elsevier Academic Press.

Subasi, A. 2019. Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques. In: *Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques*. Elsevier, pp. 1–26. Available at: https://linkinghub.elsevier.com/retrieve/pii/B9780128174449000015 [Accessed: 28 April 2025].

Teplan, M. 2002. Fundamentals of EEG Measurement. *Measurement Science Review* 2.

Trejo, L.J., Kubitz, K., Rosipal, R., Kochavi, R.L. and Montgomery, L.D. 2015. EEG-Based Estimation and Classification of Mental Fatigue. *Psychology* 06(05), pp. 572–589. doi: 10.4236/psych.2015.65055.

UK Government. 2013. *Electric and magnetic fields: sources and exposure*. Available at: https://www.gov.uk/government/publications/electric-and-magnetic-fields-sources-and-exposure/electric-and-magnetic-fields-sources-and-exposure.

Vigouroux, N., Vella, F., Lepage, G. and Campo, É. 2023. Design Recommendations Based on Speech Analysis for Disability-Friendly Interfaces for the Control of a Home Automation Environment. Available at: https://arxiv.org/abs/2311.13223 [Accessed: 8 May 2025].

Ward, E. 2021. *Accessibility and disability: UK research and statistics*. Available at: https://business.scope.org.uk/accessibility-and-disability-facts-and-figures/#Access_to_digital_technology_and_services [Accessed: 7 May 2025].

Wolpaw, J. and Wolpaw, E.W. 2012. *Brain–Computer Interfaces: Principles and Practice*. Oxford University Press. Available at: https://academic.oup.com/book/1700 [Accessed: 17 April 2025].

Yates, D. and Islam, M.Z. 2021. FastForest: Increasing random forest processing speed while maintaining accuracy. *Information Sciences* 557, pp. 130–152. doi: 10.1016/j.ins.2020.12.067.