

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ  
**ΠΑΤΡΩΝ**  
UNIVERSITY OF PATRAS

Τομέας: Τεχνολογία της Πληροφορίας  
Εργαστήριο: VVR Group / Cognix

Διπλωματική Εργασία

Του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας  
Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Ελευθέριος Παπαδόπουλος του Κωνσταντίνου

Αριθμός μητρώου: 1059386

Θέμα

**Εκτίμηση της Ετοιμότητας Οδηγού Αυτόνομου Οχήματος με Χρήση  
Βιοσημάτων**

Επιβλέπων

Καθηγητής Κωνσταντίνος Μουστάκας, Πανεπιστήμιο Πατρών

Αριθμός Διπλωματικής Εργασίας:

Πάτρα, Μήνας 201X



# ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

## Εκτίμηση της Ετοιμότητας Οδηγού Αυτόνομου Οχήματος με Χρήση Βιοσημάτων

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας  
Υπολογιστών

Ελευθέριος Παπαδόπουλος του Κωνσταντίνου

(Α.Μ.: 1059386)

παρουσιάτηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και  
Τεχνολογίας Υπολογιστών στις

\_\_\_\_ / \_\_\_\_ / \_\_\_\_

Ο Επιβλέπων

Ο Διευθυντής του Τομέα

Κωνσταντίνος Μουστάκας  
Καθηγητής

Ονοματεπώνυμο Διεθυντή Τομέα  
Τίτλος Διευθυντή Τομέα



# Στοιχεία διπλωματικής εργασίας

Θέμα: **Εκτίμηση της Ετοιμότητας Οδηγού Αυτόνομου Οχήματος με Χρήση  
Βιοσημάτων**

Φοιτητής: **Ελευθέριος Παπαδόπουλος του Κωνσταντίνου**

Ομάδα επίβλεψης  
**Καθηγητής Κωνσταντίνος Μουστάκας, Πανεπιστήμιο Πατρών**

Εργαστήρια  
VVR Group / Cognix

Περίοδος εκπόνησης της εργασίας:  
Μάϊος 2021 - Ιούνιος 2022

This thesis was written in L<sup>A</sup>T<sub>E</sub>X.



# Περίληψη

Τα αυτόνομα οχήματα έχουν πλέον εισχωρήσει στη καθημερινότητά μας και προβλέπεται πως τα επόμενα χρόνια θα γίνουν κυρίαρχα στους δρόμους του ανεπτυγμένου κόσμου. Παρόλα αυτά, για την ώρα υπάρχουν σοβαρά προβλήματα στην ασφάλειά τους. Η μεγαλύτερη μερίδα ατυχημάτων στα οποία εμπλέκονται τα αυτόνομα οχήματα γίνεται κατά τη μεταβίβαση του ελέγχου από τον αλγόριθμο που ελέγχει το αυτοκίνητο στον οδηγό. Αρκετές έρευνες έχουν δείξει ότι οι οδηγοί αυτόνομων οχημάτων μπορεί να προκαλέσουν ατυχήματα λόγω πανικού κατά τη διάρκεια μετάβασης του ελέγχου. Αυτά τα στοιχεία αναδεικνύουν τη σημασία εκτίμησης της ετοιμότητας ενός οδηγού μέσα από συστήματα ασφαλείας του αυτοκινήτου.

Στη παρούσα εργασία, δόθηκε μεγάλη έμφαση στη μελέτη της σχετικής βιβλιογραφίας για τον καθορισμό του στόχου και τη σχεδίαση των επιμέρους βημάτων υλοποίησής της. Τα κύρια ευρήματα από σύγχρονες μελέτες έδειξαν ότι μια από τις ποιο ακριβείς μεθόδους εκτίμησης της νοητικής κατάστασης ενός οδηγού είναι τα ηλεκτροεγκεφαλογραφήματα (ΗΕΓ).

Ένα ΗΕΓ είναι πρακτικά η καταγραφή της εγκεφαλικής δραστηριότητας ενός ατόμου με χρήση ηλεκτροδίων που εφάπτονται στο κεφάλι του. Τα ηλεκτρόδια αυτά, μπορούν να ανιχνεύσουν τη τάση που οφείλεται στη συγχρονισμένη αποστολή πολλών ηλεκτρικών σημάτων από νευρώνα σε νευρώνα. Αυτά τα σήματα αποτελούν τη βάση της λειτουργίας του εγκεφάλου και παρουσιάζουν διαφορετική τοπολογία ανάλογα με τη νοητική δραστηριότητα. Πολλές από τις έρευνες που μελετήσαμε έκαναν λόγο για περιοχές όπως η υπερχείλια και η γωνιώδης έλικα, οι οποίες βρίσκονται στο βρεγματικό λοβό και θεωρούνται σημαντικά κέντρα εγκεφαλικής δραστηριότητας κατά την ανάγνωση. Ακόμη, άλλες παράμετροι, όπως η συχνότητα των σημάτων ενός ΗΕΓ, είναι αξιόπιστοι δείκτες της ψυχικής κατάστασης του ατόμου. Συγκεκριμένα, έχουν οριστεί συχνοτικές ζώνες με ονόματα θήτα, δέλτα, άλφα, βήτα και γάμμα, οι οποίες από τις μικρότερες στις μεγαλύτερες τιμές, χωρίζουν το διάστημα συχνοτήτων από 0 έως 40 Hz. Για παράδειγμα, οι συχνότητες δέλτα έχουν συσχετισθεί με κατάσταση υπνηλίας σε μελέτες κούρασης οδηγών, οι συχνότητες άλφα σε κατάσταση ηρεμίας και οι συχνότητες βήτα και γάμμα σε νοητικά απαιτητικές καταστάσεις όπως επίλυση μαθηματικών προβλημάτων.

Με βάση τα παραπάνω, θεωρήσαμε πως το ΗΕΓ αποτελεί ένα πολύ χρήσιμο σύνολο πληροφοριών που μπορούμε να εκμεταλλευτούμε για την εκτίμηση της ετοιμότητας ενός οδηγού. Πέρα από την αποδοτικότητά του σε προηγούμενες εφαρμογές που μελετήσαμε, αυτή η μέθοδος προσφέρει και επεξηγησιμότητα στο μοντέλο μας καθώς τα αποτελέσματά μας μπορούν να διασταυρωθούν με

ευρήματα της γνωσιακής νευροεπιστήμης. Ωστόσο, το ΗΕΓ είναι μια ιδιαίτερα ευαίσθητη μέτρηση. Επηρεάζεται από τις πιο ανεπαίσθητες κινήσεις ενός ατόμου, ακόμα και από αυτόματες κινήσεις όπως το άνοιγμα και κλείσιμο των ματιών ή ο χτύπος της καρδιάς. Είναι φανερό πως σε ένα πραγματικό περιβάλλον οδήγησης, παράγοντες όπως η δόνηση της μηχανής ή λακκούβες στο δρόμο θα εισήγαγαν θόρυβο στις μετρήσεις μας σε καταστροφικό βαθμό. Για αυτό το λόγο, θεωρούμε πως είναι σημαντικό να μελετηθεί το ΗΕΓ ως βάση γνώσης για την εκπαίδευση άλλων αισθητήρων οι οποίοι μπορούν να λειτουργήσουν στο πλαίσιο της οδήγησης. Το δεύτερο σημαντικό εύρημα της βιβλιογραφικής μας μελέτης ήταν η συσχέτιση των εγκεφαλικών σημάτων με συσκευές ιχνηλάτησης της κίνησης των ματιών, αισθητήρων θερμοκρασίας σώματος, ρυθμού αναπνοής, ηλεκτροκαρδιογραφημάτων και ηλεκτρομυογραφημάτων.

Έπειτα από τη παραπάνω μελέτη, καταλήξαμε στο ότι ο στόχος της εργασίας μας θα είναι ο εξής. Θα προσπαθήσουμε να φτιάξουμε ένα σύνολο δεδομένων ΗΕΓ από πείραμα σε ένα σενάριο οδήγησης αυτόνομου οχήματος στο οποίο η προσοχή του οδηγού θα αποσπάται από την ανάγνωση κειμένων. Στη συνέχεια, θα επεξεργαστούμε αυτά τα δεδομένα και ευελπιστούμε να αποδείξουμε πως υπάρχει μια νευρολογική βάση πάνω στην οποία μπορούμε να στηριχτούμε για να θεωρήσουμε το ΗΕΓ ως αξιόπιστη βάση γνώσης για την εκπαίδευση περιφερειακών συστημάτων σε επόμενα πειράματα. Η

επεξεργασία των δεδομένων έγινε με δύο προβλήματα προς επίλυση. Το πρώτο ήταν η πρόβλεψη της απόκρισης το οδηγού σε περιστατικό εμφάνισης ενός εμποδίου (κρούση ή αποφυγή) και το δεύτερο η εκτίμηση του χρόνου αντίδρασης του.

Για την υλοποίηση του σεναρίου οδήγησης αυτόνομου οχήματος, υλοποιήθηκε μια προσομοίωση χρησιμοποιώντας το εργαλείο Unity. Οι συμμετέχοντες του πειράματος βρίσκονταν στη θέση του οδηγού σε ένα αυτόνομο όχημα το οποίο κινούνταν σε έναν επαρχιακό δρόμο. Το όχημα είχε τη δυνατότητα να πλοηγείται σε αυτό το δρόμο χωρίς να χρειάζεται η συμμετοχή του οδηγού. Ο οδηγός, κατά τη διάρκεια της αυτόνομης οδήγησης, διάβαζε ένα σύνολο κειμένων γραμμένα στα αγγλικά με στόχο να δώσει τη περίληψή τους και να απαντήσει σε ερωτήσεις κατανόησης στο τέλος του πειράματος. Το κείμενο βρισκόταν σε μια διαδραστική οθόνη που τοποθετήσαμε στο ταμπλό του αυτοκινήτου. Ανιχνεύσαμε την κίνηση των χεριών του με τη χρήση του συστήματος Leap Motion, δίνοντας τη δυνατότητα χρήσης αυτής της οθόνης με τα χέρια του αλλά και τη προβολή του κειμένου σε ένα τάμπλετ στο αριστερό χέρι του οδηγού, εφόσον πραγματοποιούσε μια συγκεκριμένη χειρονομία.

Σε τακτά χρονικά διαστήματα, ο συντονιστής του πειράματος ειδοποιούσε τον οδηγό για μεταβίβαση του χειρισμού του οχήματος από το αυτόνομο σύστημα στον ίδιο. Η ειδοποίηση μπορούσε να γίνει με τρεις διακριτούς τρόπους. Είτε με οπτική ειδοποίηση στην οθόνη και στο τάμπλετ που διάβαζε ο οδηγός το κείμενο, είτε με ακουστικό ερέθισμα, είτε με απτικό ερέθισμα, μέσω της συσκευής Myo Controller που τοποθετήσαμε στο μπράτσο του κάθε συμμετέχοντα. Με αυτό το τρόπο, έφτανε η ειδοποίηση στον οδηγό ότι υπάρχει ένα εμπόδιο στο δρόμο το οποίο καλείται να αποφύγει. Στις περισσότερες περιπτώσεις, ο συντονιστής δημιουργούσε ένα κύβο στη μέση της λωρίδας του οδηγού, ο οποίος έπρεπε να αποφευχθεί με ελιγμό του οχήματος. Σε λιγότερες περιπτώσεις, η ειδοποίηση ήταν ψευδής, δηλαδή δεν υπήρχε εμπόδιο στο δρόμο, ώστε να ελέγχουμε και τη συχνότητα με την οποία οι οδηγοί δημιουργούσαν ατυχήματα από τον πανικό τους. Καθ όλη τη διάρκεια του πειράματος, καταγράφαμε την εγκεφαλική δραστηριότητα των συμμετεχόντων με το σύστημα καταγραφής ΗΕΓ, ActiveTwo.

Μόλις ολοκληρώσαμε τη διεξαγωγή των πειραμάτων, είχαμε συλλέξει δεδομένα από 17 άτομα που οδήγησαν για περίπου 20 λεπτά το καθένα με 45 περιστατικά εμφάνισης εμποδίου και 15 περιστατικά ψευδούς ειδοποίησης. Οι τρεις διαφορετικοί τρόποι ειδοποίησης μοιράστηκαν σχεδόν ίσα ανάμεσα στα περιστατικά, με περίπου 15 ειδοποίησεις σε πραγματικά εμπόδια και 5 σε ψευδή ανά κάθε μέσο ειδοποίησης. Στη συνέχεια, επεξεργαστήκαμε τα δεδομένα με μεθόδους μηχανικής μάθησης για την αποθορυβοποίησή τους, την εξαγωγή χρήσιμων χαρακτηριστικών και την εκπαίδευση μοντέλων πρόβλεψης για τα δύο προβλήματα που ορίσαμε πρότερα.

Για τη προεπεξεργασία των δεδομένων, αρχικά μειώσαμε τη συχνότητα δειγματοληψίας για να είναι ευκολότερο να υλοποιήσουμε απαιτητικούς υπολογισμούς και θέσαμε την ονοματολογία του παγκόσμια αναγνωρισμένου προτύπου 10-20. Στη συνέχεια εφαρμόσαμε τον αλγόριθμο Independent Component Analysis (ICA) για την απομάκρυνση των τεχνικών σφαλμάτων. Όπως είχαμε αναφέρει και νωρίτερα, κινήσεις όπως το άνοιγμα και κλείσιμο των ματιών πρέπει να απομακρυνθούν ώστε να μην επηρεάσουν τις μετρήσεις μας. Για να το κάνουμε αυτό, χρησιμοποιούμε την μέθοδο ICA η οποία εξάγει τα μεγίστως ανεξάρτητα σήματα από το ΗΕΓ. Διαισθητικά, αυτός ο αλγόριθμος ξεδιαλύνει τα σήματα τα οποία συμβάλλουν σε κάθε κανάλι του ΗΕΓ. Με αυτό το τρόπο καταλήγουμε να έχουμε κανάλια που απομονώνουν τον θόρυβο από κινήσεις. Αφαιρώντας αυτά τα κανάλια από τα αρχικά μας δεδομένα οδηγεί στον καθαρισμό της καταγραφής από ανεπιθύμητο θόρυβο. Έπειτα, εφαρμόσαμε ένα ζωνοπερατό φίλτρο στις συχνότητες 1 Hz και 40 Hz. Τελευταίο βήμα της προεπεξεργασίας των δεδομένων ήταν η εξαγωγή των γεγονότων του πειράματος τα οποία είχαν συγχρονιστεί κατά τη διάρκεια διεξαγωγής του. Έτσι, λάβαμε τις ακριβείς χρονικές στιγμές κατά τις οποίες ειδοποιούσαμε τους συμμετέχοντες για την ύπαρξη εμποδίου, τις στιγμές που έπιαναν το τιμόνι και την κατάληξη του συμβάντος είτε με αποφυγή είτε με συντριβή. Από αυτά τα δεδομένα, υπολογίσαμε τους χρόνους αντίδρασης και απομονώσαμε τα 2 δευτερόλεπτα πριν την ειδοποίηση του οδηγού για να τα χρησιμοποιήσουμε στην εκπαίδευση των παρακάτω μοντέλων.

Οι αλγόριθμοι Common Spatial Patterns (CSP) και Source Power Modulation (SPoC), χρησιμοποιήθηκαν για την εξαγωγή χρήσιμων γνωρισμάτων από τις μετρήσεις μας. Ο αλγόριθμος CSP, χρησιμοποιεί την επισήμανση κλάσεων ενός προβλήματος για να εντοποίσει τα χωρικά μοτίβα τα οποία συναντώνται συχνότερα στη κάθε περίπτωση. Στη συνέχεια, μετασχηματίζει τα αρχικά δεδομένα σε έναν νέο χώρο και εμείς χρησιμοποιούμε τη μέση ισχύ του φάσματος τους για την εκπαίδευση των επόμενων μοντέλων. Διαισθητικά, αυτός ο αλγόριθμος εντοπίζει τη δραστηριότητα η οποία ξεχωρίζει καλύτερα τις περιπτώσεις σύγκρουσης του οχήματος και αποφυγής του εμποδίου και την τονίζει στον μετασχηματισμένο χώρο. Ακριβώς την ίδια λειτουργία διατελεί ο αλγόριθμος SPoC με μόνη διαφορά ότι αντί για τις ετικέτες κλάσεων που χρησιμοποιεί ο CSP αλγόριθμος, εκείνος χρησιμοποιεί μια συνεχή τιμή, όπως είναι ο χρόνος αντίδρασης των οδηγών. Κατά αντιστοιχία, τα αρχικά δεδομένα μετασχηματίζονται σε έναν χώρο στον οποίο τα μοτίβα που σχετίζονται περισσότερο με το χρόνο απόκρισης των οδηγών, έχουν τονιστεί. Τέλος, και σε αυτή τη περίπτωση, χρησιμοποιούμε τη μέση ισχύ φάσματος για να εκπαιδεύσουμε τα μοντέλα μηχανικής μάθησης.

Για την πρόβλεψη των αποτελεσμάτων μας χρησιμοποιήσαμε τους αλγορίθμους Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR) και Decision Tree (DT) για το πρόβλημα κατηγοριοποίησης μεταξύ των κλάσεων αποφυγής και σύγκρουσης. Αντίστοιχα, στο πρόβλημα εκτίμησης της τιμής του χρόνου αντίδρασης χρησιμοποιήθηκαν πάλι οι αλγόριθμοι SVM και RF και προστέθηκε ο αλγόριθμος Ridge.

Πριν τη παρουσίαση των τελικών αποτελεσμάτων έγινε μια οπτικοποίηση των μοτίβων συμπεριφοράς των συμμετεχόντων και μια στατιστική ανάλυση των δεδομένων του ΗΕΓ. Από την ανάλυση της συμπεριφοράς των οδηγών είδαμε ότι οι περισσότεροι οδηγοί απέφυγαν το εμπόδιο περισσότερες φορές με το απτικό ερέθισμα και ελάχιστες με το ακουστικό. Ένα απροσδόκητο αποτέλεσμα που εντοπίσαμε ήταν πως ο χρόνος αντίδρασης ανάμεσα στις συγκρούσεις και στις αποφυγές δεν

παρουσίαζε στατιστικά σημαντική διαφορά. Παράλληλα, παρατηρήσαμε πως το οπτικό και το ακουστικό ερέθισμα είχαν παρόμοια κατανομή χρόνων αντίδρασης, ενώ το απτικό ερέθισμα επέφερε γρηγορότερους χρόνους αντίδρασης. Μέσα από τη στατιστική ανάλυση των χρονοσειρών του ΗΕΓ, αποφανθήκαμε ότι ο μηχανισμός ανάλυσης του οπτικού ερεθίσματος παρουσιάζει καθυστέρηση στην αντίληψη του σήματος αλλά ταχύτατη λήψη απόφασης. Αντίστοιχα, ο μηχανισμός ανάλυσης του ακουστικού ερεθίσματος είχε ταχύτερη επεξεργασία του σήματος αλλά μεγάλη καθυστέρηση στη λήψη απόφασης και για αυτό το λόγο οι χρόνοι αντίδρασης στις δύο περιπτώσεις είναι παρόμοιοι. Ακόμη, ο μηχανισμός ανάλυσης του απτικού ερεθίσματος είναι γρηγορότερος και στις δύο περιπτώσεις. Πέρα από αυτά, παρατηρήσαμε μια αύξηση της έντασης των υψηλών συχνοτήτων (βήτα, θήτα) στις περιπτώσεις σύγκρουσης και αντίστοιχη αύξηση έντασης χαμηλών συχνοτήτων (δέλτα, θήτα, άλφα) στη περίπτωση αποφυγής. Τέλος, παρατηρήσαμε μια αύξηση στη τυπική απόκλιση των σημάτων στη περίπτωση σύγκρουσης σε σχέση με τη περίπτωση αποφυγής.

Τα τελικά αποτελέσματα των προβλημάτων κατηγοριοποίησης των συγκρούσεων και των αποφυγών και της πρόβλεψης του χρόνου αντίδρασης των οδηγών ήταν σε μεγάλο βαθμό επιτυχής. Το καλύτερο μοντέλο που εφαρμόσαμε ήταν το RF με ποσοστό επιτυχίας στο 85%. Αναφορικά με τη πρόβλεψη του χρόνου αντίδρασης παρατηρήσαμε μέσο σφάλμα 0.12 msec, ωστόσο πιο προηγμένες μετρικές έδειξαν πως το μοντέλο μας δεν κατάφερε να εξηγήσει μεγάλο μέρος της διασποράς των δεδομένων.

Τα συμπεράσματα που βγάλαμε από το πείραμά μας ήταν κυρίως θετικά, με ορισμένες βελτιώσεις της διαδικασίας του πειράματος που θα μπορούσαν να γίνουν σε μελλοντικές έρευνες. Αρχικά, θεωρούμε πως ένας συνδυασμός ιχνηλάτησης της κίνησης των ματιών και ερωτήσεων κατανόησης του κειμένου, κατά τη διάρκεια του πειράματος, θα μας εξασφάλιζε ότι οι συμμετέχοντες αφοσιώνονταν στο κείμενο. Παράλληλα, θεωρούμε πως οι συμμετέχοντες, σε μεγάλο βαθμό, έμαθαν να αντιδρούν μηχανικά στο εμπόδιο κάτι το οποίο μπορεί να εξηγήσει τους σχεδόν ίδιους χρόνους αντίδρασης στις περιπτώσεις σύγκρουσης και αποφυγής. Μια σχεδίαση με περισσότερες τροχιές αποφυγής και διακύμανση στην απόσταση εμφάνισης του εμποδίου θα εξασφάλιζε ότι οι

συμμετέχοντες έπρεπε να κρίνουν το κάθε εμπόδιο ξεχωριστά. Σε τελική ανάλυση θεωρούμε πως οι υποθέσεις μας, ως ένα βαθμό, επαληθεύτηκαν και υπάρχει έφορο έδαφος για μελλοντικές έρευνες και υλοποίηση εφαρμογών σε αυτό τον τομέα.

# Abstract

Autonomous vehicles have been introduced in our daily lives and are projected to become the norm in the near future. Despite their impressive technology, however, autonomous vehicles still lack in some safety aspects. Specifically, most of the accidents autonomous vehicles are involved happen during the take over requests that notify the driver to assume the control of the vehicle in the face of a potentially dangerous situation, which is usually attributed to driver's distraction. We aim to prove that using electroencephalography (EEG), we can correlate a driver's mental distraction with respective brain activity. We built an autonomous vehicle driving scenario that included distractions, in the form of text reading, and conducted experiments on multiple subjects, creating a take over request based EEG dataset that was missing from the literature. Additionally, we built a machine learning based system to predict the outcome of take over requests in our simulation using the pre-stimulus data from the EEG. In the end, we achieved an accuracy of 85%, providing strong incentive for further work containing EEG recordings, in the field of autonomous vehicle safety.



## Acknowledgements

I would like to express my gratitude to the supervisor of this thesis, professor Konstantinos Moustakas for his guidance and motivation. I would also like to thank associate professor Christos Feidas for providing the necessary equipment for the EEG recordings. The assistance provided by Dr Delis Ioannis, lecturer at Leeds university, for his expert guidance on the set up of the experiment and the processing stage of the recordings. The aid of Evripidis Avraam was also much appreciated as he created the simulation. Lastly, I wish to extend my special appreciation to all the volunteers who participated in the experiment for free, without which, this thesis would not be complete.



# Contents

<b>List of Figures</b>	<b>xix</b>
<b>List of Tables</b>	<b>xxiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Context of the problem at hand . . . . .	1
1.2 Goals of this work . . . . .	1
1.3 Contributions of this work . . . . .	2
<b>2 The Role of Biosignals in Road Accident Prevention</b>	<b>3</b>
2.1 . . . . .	3
2.2 Definitions of distraction and fatigue effects . . . . .	3
2.3 Inattentiveness measurement in previous works . . . . .	4
2.3.1 Subjective measurements . . . . .	4
2.3.2 Biological measures . . . . .	4
2.3.3 Driving performance measures . . . . .	4
2.3.4 Driver physical measures . . . . .	5
2.3.5 Hybrid approaches . . . . .	5
<b>3 Neurological Background and Related Materials</b>	<b>7</b>
3.1 The physiology of the brain . . . . .	7
3.1.1 Structural division of the brain . . . . .	7
3.1.2 The skull and meninges . . . . .	7
3.1.3 The lobes and their functions . . . . .	8
3.2 Communication between nerve cells . . . . .	9
3.2.1 Nerve cell shape . . . . .	9
3.2.2 The chemical excitation of the nerve cell . . . . .	9
3.2.3 The excitation-inhibition mechanism . . . . .	10
3.3 Electroencephalography . . . . .	10
3.3.1 Origins of the EEG . . . . .	11
3.3.2 The physics of the EEG . . . . .	11
3.3.3 Brain Rhythms . . . . .	12
3.3.4 References . . . . .	14
3.3.5 Artifacts . . . . .	15
3.4 Localization of activity in the brain . . . . .	16
<b>4 Experiment</b>	<b>19</b>
4.1 The Simulation . . . . .	19
4.1.1 General information and goals . . . . .	19
4.1.2 The setting . . . . .	19
4.1.3 The self-driving vehicle . . . . .	19

4.1.4	Distraction mechanism . . . . .	21
4.1.5	Obstacle creation and takeover request . . . . .	22
4.2	Simulation system's arrangement . . . . .	25
4.2.1	Simulation computer . . . . .	25
4.2.2	Simulation monitor . . . . .	25
4.2.3	Steering wheel . . . . .	25
4.2.4	Pedals . . . . .	25
4.2.5	Leap Motion Controller . . . . .	27
4.2.6	Sound system . . . . .	27
4.2.7	Myo Gesture Control Armband . . . . .	27
4.3	ActiveTwo acquisition system . . . . .	29
4.3.1	AD-box . . . . .	29
4.3.2	Electrodes . . . . .	30
4.3.3	Battery-box . . . . .	31
4.3.4	Charger . . . . .	31
4.3.5	Receiver . . . . .	32
4.3.6	Gel and syringes . . . . .	32
4.4	Neurospec triggering system . . . . .	33
4.4.1	MMBT-S Trigger Interface Box . . . . .	34
4.4.2	MMBT-S features . . . . .	35
4.4.3	Modality . . . . .	35
4.4.4	Recording computer . . . . .	35
4.4.5	Remote control monitor . . . . .	35
4.5	Brainwave acquisition and triggering software . . . . .	36
4.5.1	Actiview902 . . . . .	36
4.5.2	Python script for trigger synchronization . . . . .	37
4.6	Experiment protocol . . . . .	37
4.6.1	Briefing . . . . .	37
4.6.2	During the experiment . . . . .	38
4.7	Subjects . . . . .	38
<b>5</b>	<b>EEG Processing</b> . . . . .	<b>39</b>
5.1	MNE Terminology . . . . .	39
5.1.1	Raw . . . . .	39
5.1.2	Epoch . . . . .	39
5.1.3	Evoked . . . . .	39
5.2	Standardization . . . . .	39
5.2.1	Removing excess channels . . . . .	39
5.2.2	Extracting event data . . . . .	40
5.2.3	Setting a standard montage and channel references . . . . .	40
5.3	Downsampling . . . . .	41
5.4	Epoch Extraction . . . . .	41
5.4.1	Classification problem . . . . .	41
5.4.2	Regression problem . . . . .	41
5.5	Filtering . . . . .	42
5.6	Artifact Removal . . . . .	43
5.7	Common Spatial Patterns . . . . .	45
5.8	Source Power Modulation . . . . .	49
5.9	Machine Learning Fundamentals . . . . .	50
5.9.1	Types of Machine Learning Problems . . . . .	51

5.9.2	Overfitting and Underfitting . . . . .	51
5.9.3	Training and Validation . . . . .	52
5.10	Classification Task . . . . .	52
5.10.1	Decision Trees . . . . .	52
5.10.2	Random Forest Classifier . . . . .	53
5.10.3	Logistic Regression . . . . .	54
5.10.4	C-Support Vector Classification . . . . .	57
5.11	Classification Metrics . . . . .	58
5.11.1	Precision . . . . .	58
5.11.2	Recall . . . . .	58
5.11.3	F1-score . . . . .	59
5.11.4	Accuracy . . . . .	59
5.11.5	Macro average . . . . .	59
5.11.6	Weighted average . . . . .	59
5.12	Regression Task . . . . .	59
5.12.1	Support Vector Regression . . . . .	59
5.12.2	Random Forest Regression . . . . .	60
5.12.3	Ridge Regression . . . . .	60
5.13	Regression Metrics . . . . .	61
5.13.1	Mean square error . . . . .	61
5.13.2	Mean average error . . . . .	61
5.13.3	R squared . . . . .	62
<b>6</b>	<b>Results</b> . . . . .	<b>63</b>
6.1	Statistical Analysis . . . . .	63
6.1.1	Crashes versus Evasions analysis . . . . .	63
6.1.2	Reaction time analysis . . . . .	69
6.1.3	Conclusion . . . . .	71
6.1.4	Statistical analysis of EEG Signals . . . . .	72
6.1.5	Conclusions . . . . .	77
6.2	Classification Results . . . . .	78
6.2.1	SVC with polynomial kernel . . . . .	79
6.2.2	SVC with radial basis function kernel . . . . .	81
6.2.3	SVC with linear kernel . . . . .	84
6.2.4	Decision Tree . . . . .	85
6.2.5	Random Forset Classifier . . . . .	87
6.2.6	Logistic Regression . . . . .	89
6.2.7	Classification scores . . . . .	90
6.3	Regression Results . . . . .	92
6.3.1	SVR . . . . .	93
6.3.2	Random Forest Regression . . . . .	95
6.3.3	Ridge Regression . . . . .	97
6.3.4	Regression scores . . . . .	99
<b>7</b>	<b>Limitations and Future Work</b> . . . . .	<b>101</b>
7.1	Uncertainties about participant's attention . . . . .	101
7.2	Reflex response hypothesis and alternative distractions . . . . .	101
7.3	Furthering the collection of data . . . . .	102
7.4	Alternative feature extraction methods . . . . .	102
<b>Bibliography</b>		<b>103</b>



# List of Figures

3.1	Illustration of the human brain depicting the cerebrum, cerebellum and brainstem . . . . .	7
3.2	Illustration of a section of the human head, depicting the skull and the meninges around the human brain . . . . .	8
3.3	Illustration of the lobes of the human brain . . . . .	8
3.4	Illustration of some of the different functions each lobe of the brain is capable of controlling . . . . .	9
3.5	Illustration of a typical nerve cell . . . . .	10
3.6	Diagram of the potential of the neuron during excitation . . . . .	10
3.7	Image of the first ever EEG recording made by Hans Berger in 1921 . . . . .	11
3.8	Arrangement of the electrodes on a subject's head according to the 10-20 standard. . . . .	12
3.9	Depiction of typical waveforms of different brain rhythms. . . . .	13
3.10	Relative percentile amplitude of the different brain rhythms that can be measured in an EEG recording. . . . .	14
3.11	The effect of EOG artifacts on EEG recordings. . . . .	15
3.12	The effect of ECG artifacts on EEG recordings. . . . .	16
4.1	Preview of the setting chosen for the experiment, taken from the Unity game development software . . . . .	20
4.2	Preview of the interior of the car in developer mode . . . . .	20
4.3	Application of fog in the simulation in order to reduce driver's visibility and increase framerate performance . . . . .	21
4.4	The tablet implemented at the interior of the car as a distraction mechanism . . . . .	21
4.5	Recording hand position using Leap Motion Controller . . . . .	22
4.6	Text chosen by the participants after navigation the tablet interface using their hands . . . . .	22
4.7	Pop-up module controlled by the left hand to magnify the text and cover part of the participant's view of the road . . . . .	23
4.8	Preview of the obstacle. It consists of one wooden cube that is created in the middle of the driver's lane. This is a true positive case . . . . .	23
4.9	Visual feedback in a takeover request scenario . . . . .	24
4.10	A false alarm takeover request event . . . . .	24
4.11	Rewarding visual cue presented to the participants after a successful takeover request event . . . . .	24
4.12	Ultrawide Curved Monitor used in the experiment to show the simulation . . . . .	25
4.13	Logitech G29 Driving Force steering wheel used to control the car in the simulation . . . . .	26
4.14	Logitech G29 Driving Force pedals used to control acceleration and deceleration of the vehicle . . . . .	26
4.15	Leap Motion Controller used for hand tracking. The infrared emitters are visible in the photograph . . . . .	27
4.16	Picture of the sound system used to produce the audio notification. The simulation computer is situated underneath . . . . .	28

4.17	Picture of Myo Gesture Control Armband used for the haptic notification . . . . .	28
4.18	Image of AD-Box on top of the Battery . . . . .	29
4.19	Image of the electrodes including the 32 EEG channels and CMS/DRL electrodes . . . . .	31
4.20	Image of the front of the charger . . . . .	31
4.21	Image of the back of the charger . . . . .	32
4.22	Image of a bottle of signa gel used in the experiments . . . . .	33
4.23	Image of a syringe used to apply gel to the subject's head during the experiments . . . . .	33
4.24	Image of the MMBT-S Trigger Interface Box . . . . .	34
4.25	Image of the MMBT-S Trigger Interface Box's SUB-D port . . . . .	34
4.26	Image of the MMBT-S Trigger Interface Box's USB port and runtime indicator . . . . .	35
5.1	Graph showing the Hamming window in time and frequency domains . . . . .	42
5.2	Unfiltered display of the first 20 electrodes . . . . .	44
5.3	Extracting the 32 ICA components . . . . .	44
5.4	Difference between unfiltered (red) and filtered (black). The above diagram depicts all the channels and the below average across the whole EEG recording. . . . .	45
5.5	Display of the first 20 electrodes after the filtering . . . . .	45
5.6	Effect of spatial CSP filtering in motor imagery task . . . . .	46
5.7	A toy example of CSP filtering in 2-D showcasing the discriminative power of spatial filtering. . . . .	47
5.8	Example of visualization of CSP analysis on a motor imagery classification task . . . . .	48
5.9	Geometric mapping between body parts and motor/somatosensory cortex . . . . .	48
5.10	Illustration of different approaches to relating spectral power to a target variable . . . . .	49
5.11	Example of a decision tree made to differentiate a set of numbers based on their color and underlining . . . . .	53
5.12	Example of a decision tree ensemble. Each individual tree votes for the label of an observation and the majority determines the prediction of the model . . . . .	54
5.13	Graph depicting the sigmoid function . . . . .	55
5.14	Diagram showing the value of the log loss cost function depending on the values of the label ( $y_i$ ) and the model's prediction ( $a_i$ ) . . . . .	56
5.15	Image showing the Support Vectors used to solidify the hyperplane's position and orientation. . . . .	57
6.1	Aggregate results of each event type . . . . .	64
6.2	Chart depicting the number of total crashes (in red) versus total evasions (in green) . . . . .	64
6.3	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a visual notification to the subject. . . . .	65
6.4	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a audio notification to the subject. . . . .	66
6.5	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a haptic notification to the subject. . . . .	66
6.6	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a visual notification to the subject. . . . .	67
6.7	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a audio notification to the subject. . . . .	68
6.8	Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a haptic notification to the subject. . . . .	68
6.9	Histogram displaying the reaction times of the three different modalities of notifications used in the experiment . . . . .	69

6.10	Plot of the empirical Cummulative Distribution Function of the three modalities. Visual stimulus is represented by the orange line, audio stimulus by the blue line and haptic by the green.	70
6.11	Violin plot of the crash and evasion conditions with respect to reaction time. The blue area is the distribution of evasion events and the orange area is the distribution of the crash events.	71
6.12	Event Related Potenital of visual notifications.	72
6.13	Event Related Potenital of audio notifications.	73
6.14	Event Related Potenital of haptic notifications.	73
6.15	Joint plot of the crash cases	74
6.16	Joint plot of the evasion cases.	75
6.17	Topological map of the evasion cases using the average over the 2.5 second window before the stimulus onset.	75
6.18	Topological map of the evasion cases using the average over the 2.5 second window before the stimulus onset.	76
6.19	Topological map of the evasion cases, divided into frequency zones.	76
6.20	Topological map of the crash cases, divided into frequency zones.	76
6.21	Comparison of standard deviation between crashes (orange) and evasions (blue).	77
6.22	Diagram of the regularization parameter C with regards to the model's accuracy score.	79
6.23	Diagram of the kernel's polynomial degree with regards to the model's accuracy score.	80
6.24	Diagram of the CSP components with regards to the accuracy score of the SCV with polynomial kernel.	81
6.25	Diagram of the regularization parameter C with regards to the model's accuracy score using Radial Basis Function.	82
6.26	Diagram of the CSP components with regards to the accuracy score of the SCV with RBF kernel.	83
6.27	Diagram of the regularization parameter C with regards to the model's accuracy score when using Linear kernel.	84
6.28	Diagram of the CSP components with regards to the accuracy score of the SCV with polynomial kernel.	85
6.29	The decision tree created by training on 80% of the data	86
6.30	Same decision tree, zoomed in order to	86
6.31	Diagram of the CSP components with regards to the decision tree's accuracy score	87
6.32	Diagram of the number of estimators (decision trees) with regards to the model's accuracy score.	88
6.33	Diagram of the CSP components with regards to the Random Forest Classifier accuracy score	88
6.34	Diagram of the regularization parameter C with regards to the Logistic Regression's accuracy score.	89
6.35	Diagram of the CSP components with regards to the Logistic Regression's accuracy score	90
6.36	Patterns extracted by the CSP algorithm	91
6.37	Mean Decrease in Impurity of each feature	92
6.38	Diagram of the Mean Absolute Error with respect to the degree of the SVR kernel's polynomial	93
6.39	Diagram of the Mean Squared Error with respect to the degree of the SVR kernel's polynomial	94
6.40	Diagram of the R squared metric with respect to the degree of the SVR kernel's polynomial	94

6.41	Diagram of the Mean Absolute Error with respect to the degree of the number of estimators used . . . . .	95
6.42	Diagram of the Mean Squared Error with respect to the degree of the number of estimators used . . . . .	96
6.43	Diagram of the R squared metric with respect to the degree of the number of estimators used . . . . .	96
6.44	Diagram of the Mean Absolute Error with respect to the regularization parameter alpha of the model . . . . .	97
6.45	Diagram of the Mean Squared Error with respect to the regularization parameter alpha of the model . . . . .	98
6.46	Diagram of the R squared metric with respect to the regularization parameter alpha of the model . . . . .	98
6.47	Patterns extracted by the SPoC algorithm . . . . .	99
6.48	Average power of SPoC components correlated to Reaction Time. A line is fitted through every feature to show the best linear fit possible in the data. . . . .	100
6.49	Error of the prediction of each testing sample . . . . .	100

# List of Tables

4.1	Codes sent to the MMBT-S in relation to the events of the simulation . . . . .	37
6.1	Mean reaction times of the three notification modalities . . . . .	70
6.2	Accuracy, macro F1 average and Weighted F1 average score of each model . . . . .	90
6.3	Precision and Recall scores of both crash and evasion cases of each model . . . . .	91
6.4	Precision and Recall scores of both crash and evasion cases of each model . . . . .	99



# 1. Introduction

## 1.1 Context of the problem at hand

Modern day technology is on the verge of transforming transportation of people by introducing autonomous vehicles for public usage. The National Highway Traffic Safety Administration (NHTSA) defined five levels of automation of vehicles ranging from level 0, including vehicles where there exist no automation and the driver is in complete and sole control of it, to level 4, referring to vehicles that control every safety-related aspect of driving and the only input of users is their destination [1]. Most of the self-driving vehicles available to the public today reach up to level 3, meaning that they can control safety-critical functions under certain circumstances and the human driver must be alert at any moment as he might be required to take control of the vehicle.

The control of the vehicle is exchanged between the self-driving car's software and the driver by events which are called Take-over Requests. During the Take-over Request, the vehicle notifies the driver through various means including visual, audio and vibrotactile haptic means or combinations of the above [2].

There are enough indications that automated vehicles can reduce road accident numbers and severity, especially considering the involvement of human factors in non-autonomous vehicle accidents [3]. Despite the promise of safety made by automating driving, there are still serious challenges related to system failures and other factors. One of these factors is the disengagement from automated to manual control of the vehicle which can amount up to 20% to 27% of accidents involving self-driving cars [4], [5]. Recent works have stated the importance of studying the effects of Take-over Request modalities with regards to reaction time [6] as well as building systems to predict the driver's Take-over performance [7].

## 1.2 Goals of this work

The main goal of this work is to setup a driving experiment and create a dataset containing biosignal data, specifically the electroencephalograph of the experiment's participants. The choice of this approach was based on the extensive works made by the scientific community on driver attention and fatigue using biosignals. Further analysis of these works will be made in a future chapter. Additionally, we aim at creating a system that is able to predict the performance of the driver before the Take-over Request happens.

Our efforts will focus on proving that there exists a scientific foundation upon which, a take over notification system can become significantly safer. As we will discuss in further chapters, a system based solely on the EEG recordings of the driver can not be applied to a real world scenario. Instead, the EEG recording will be used as the "ground truth" upon which peripheral systems will be trained to spot driver's inattentiveness. A likely scenario would be the integration of eye tracking systems. Yet the first step is to make sure that a link between distraction and brain activity can be found.

### 1.3 Contributions of this work

The novelty of this work can be attributed on two factors. The first one, is the apparent lack of publicly available EEG recording on a Take-over Request setting with multiple feedback modalities. The second, is our methods for building the prediction system which are based on advanced feature extraction and machine learning techniques, all of which will be analyzed with greater detail later.

## 2. The Role of Biosignals in Road Accident Prevention

### 2.1

There has been a significant increase in the literature discussing the problem of driver's inattentiveness in the past years. Most researchers try to address one of two problems, driver's distraction or driver's fatigue. Different types of measures have been proposed for the prediction of inattentive state of the driver and the subsequent prevention of road accidents. These measures can be divided into five categories:

1. Subjective report measures
2. Driver biological measures
3. Driver physical measures
4. Driving performance measures
5. hybrid measures

Furthermore, most of the previous works applied some kind of method of data analysis that was based on machine learning or artificial intelligence. In the following sections we will discuss the literature that inspired our work.

### 2.2 Definitions of distraction and fatigue effects

It is crucial for our work to define the concepts of distraction and fatigue. Even though our work was exclusively focused on distraction, works on fatigue detection and prevention of fatigue related accidents offered crucial insight on techniques that were also applied on distraction detection.

The definition of distraction we followed [8] was determined as performing a cognitively demanding task while driving that influences both the driver's visual behaviour and driving performance. The effects of distraction can be observed in driver's behavioural patterns, physiological responses and driving performance.

Regarding behavioural pattern alteration, there has been evidence that driver's change their inspection patterns on the forward and outward view [9], [10], [11]. Research indicated that distracted drivers would spend more time looking directly ahead, reducing the inspection of the instruments and mirrors and glances at traffic signals and the area around the vehicle. This indicated that eye-glance patterns could be used to discriminate driving while performing a secondary distracting task. As of other physiological responses, pupil dilation and heart rate increase was correlated with imposed cognitive loads [12]. Additionally, temperature alterations in some parts of the body were indicative of the driver performing a distracting task [13], [14]. Lastly, and most important of the above for our

work was the evidence of EEG signals containing information that could indicate the driver's distraction from a cognitively demanding secondary task [15]. Driver distraction also affected the driving performance by lowering the driver's control over the vehicle which resulted in sudden speed changes and drifting outside of lane boundaries [16]. Furthermore, delays in checking behaviour were observed as well as reduced steering smoothness [17]. Both of these are very important in our study as a means of determining an inadequate reaction from the driver.

Driver fatigue was defined as the mental and physical ineffectiveness as a result of extensive workload. Similarly to distraction, the driver's behavioural patterns changed when fatigued. In particular, the driver was inclined to relieve muscular tension in the neck more often, yawning was observed more frequently and a more relaxed hand position was adopted [18]. In regards to the EEG effects of fatigue, it was observed that low frequencies had a significant relationship with sleepiness while alertness was correlated with higher frequencies called beta waves that are going to be analyzed in a future chapter. Higher alpha wave activity was also observed in the central regions of the brain in a drowsy state [19]. Lastly, steering wheel reversals were lower in sleep deprived drivers [20] as well as less frequent steering maneuvers or correction [21], low velocity steering and large amplitude steering movements [22].

## 2.3 Inattentiveness measurement in previous works

### 2.3.1 Subjective measurements

One of the most commonly used tool for subjective self-assessment of sleepiness is the KSS (Karolinska Sleepiness Scale). Its reliability was investigated through extensive testing using EEG. The problem with subjective testing is that it requires feedback from the user, something that might reengage the driver's attention by making him self conscious of his state. In the scope of our work, subjective measures were only used to assess the engagement of the participants to the distraction task.

### 2.3.2 Biological measures

Biological measures are a very useful tool to determine a driver's distraction or fatigue as they are objective, automatic functions that the driver can not control. Biological signals used in the literature were electroencephalograms (EEG) [23], electrocardiograms (ECG) [24], electrooculograms (EOG) [25] and electromyograms (EMG) [26]. Both statistical variables such as standard deviation, average value, and frequency domain features, like the energy content of each frequency band, are used to determine changes of driver's state. In principal, power spectrum features were more reliable and consistent at predicting fatigue. Second by second classifications were more successful at predicting the driver's distraction state. A distracted driver's EEG recording required a more sophisticated feature extraction method as simple statistical and power spectrum analysis were not consistent among different cognitive tasks.

Other biological sensors were used in order to monitor the activity of other parts of the body that could indicate a driver's change in mental state. Specifically, in [27] four types of sensors were used, namely ECG and EMG sensors as we discussed earlier, skin conductivity sensors and respiration sensors. These measurements were used to assess a driver's stress levels.

### 2.3.3 Driving performance measures

Changes in a driver's performance can be induced by a change of mental state. Steering angle and vehicle tracking were used in order to predict a driver's attention levels [28], [29]. Other approaches included measuring the pressure distribution on the driver's seat [30] or the forces on the pedals and the vehicle's speed [31].

### 2.3.4 Driver physical measures

As the aforementioned biological measures are usually very sensitive to noise, as we are going to discuss in a further chapter, applying them to a real driving scenario seems difficult. Other physical measures, though, that can correlate with these recordings could be a viable alternative for monitoring a driver's state. Specifically, indicators such as PERCLOS, head-nodding frequency, slouching frequency and posture adjustment frequency were used in [32]. Work on attention analysis was done in [33]. They tried to detect visual distraction with head pose and fuse it with yawning, eyebrow raising and PERCLOS for fatigue detection.

Physical measures like eye movement seem to correlate highly with EEG recordings that indicate a person's drowsiness state [34]. This leaves space for approaches combining multiple measurements from a driver and fusing them together into more reliable classifiers. EEG recordings could be used as ground truth for the state of the driver in order for classifiers, using features from physical and performance measurements to be applied in a real world situation.

### 2.3.5 Hybrid approaches

Most influential for our work was the prospect of combining features that are easily attainable in a real world scenario and using EEG recordings as ground truth to provide physiologically based labeling for the classifier training. Specifically, eye tracking and EEG recordings seem to be closely related in many different situations. In [35], apart from the EEG data, eyelid movement, pupil diameter, gaze direction and head position were measured in order to create and train a dynamic encoder-decoder model for fatigue detection. Heart rate variability in association with EEG recordings as predictors for driver's drowsiness were studied in [36]. A combination of EEG and EMG signals was used in [37] to estimate a driver's vigilance and respiration signals were used alongside EEG recordings in [38] to automatically detect sleep-onset.



# 3. Neurological Background and Related Materials

## 3.1 The physiology of the brain

The brain is one of the most important organs in human anatomy located in the head. It is the central piece of the nervous system and plays a crucial role in controlling the functions of the human body, the interpretation of information from the outside world and the higher mental capabilities of the human species.

### 3.1.1 Structural division of the brain

It is composed of three functionally and structurally different regions, the cerebrum, the cerebellum and the brainstem [39] as depicted in 3.1. The cerebrum is the larger part of the brain, it is separated in two hemispheres and it controls higher mental functions like emotion, language usage, vision, learning and fine control of movement. The cerebrum is located under the cerebellum and it regulates muscle movement, maintaining posture and balance among other lower complexity functions. Finally, the brainstem acts as a connection between the cerebrum, the cerebellum and the rest of the body and controls automatic functions such as breathing and regulating the heart rate.

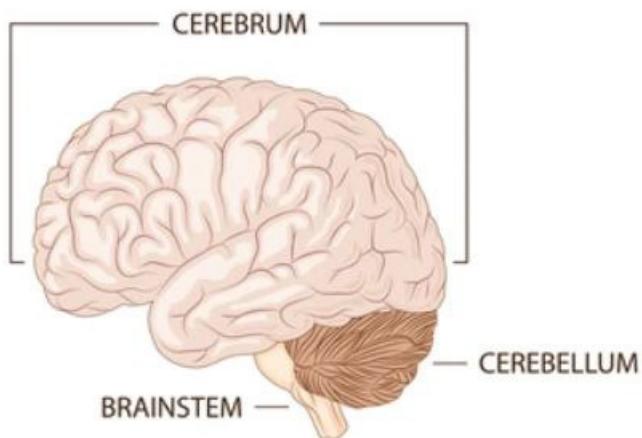


Figure 3.1: Illustration of the human brain depicting the cerebrum, cerebellum and brainstem

### 3.1.2 The skull and meninges

The brain is enclosed by the meninges and the skull as shown in 3.2. The skull is a bone structure that forms a protective cavity for the brain. On the other hand, the meninges consist of three layers of tissue that protect the brain and supply it with sufficient blood [40]. In particular, the outermost

meninx is called dura mater and its role is to protect the brain and separate it's two hemispheres as well as the cerebrum from the cerebellum. The intermediate layer is called arachnoid mater and provides elasticity to the meninges structure. The space between the dura mater and the arachnoid mater is called subdural space and has a significant importance in acquiring data from the brain. The inner most meninx is called pia mater and provides the brain with blood vessels that reach deep inside its structure.

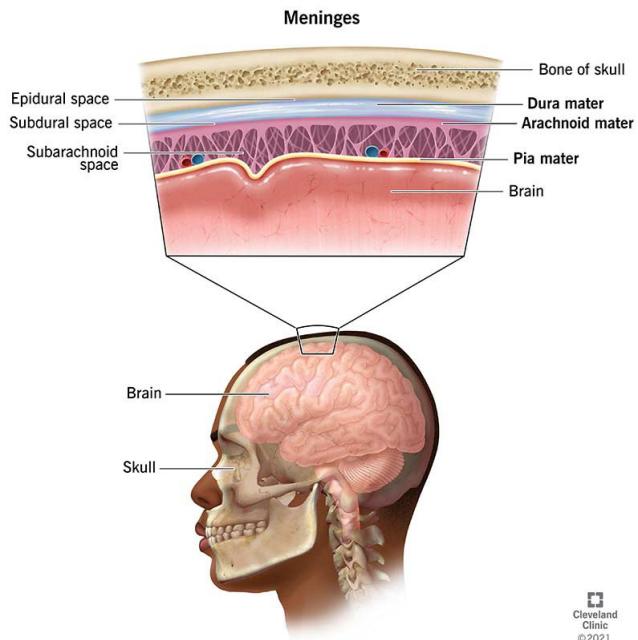


Figure 3.2: Illustration of a section of the human head, depicting the skull and the meninges around the human brain

### 3.1.3 The lobes and their functions

Apart from its structural components, the brain is separated into functionally different areas as well. The ones that are of high importance for the task at hand are located near the surface. In particular, each hemisphere is divided into four lobes as depicted in 3.3, the frontal lobe, located ,appropriately to its name, at the front of the brain, the temporal lobe, located on the sides, above the ears, the parietal lobe which is found in the middle of the brain and the occipital lobe at the back. These four lobes have significantly different functions. [39]

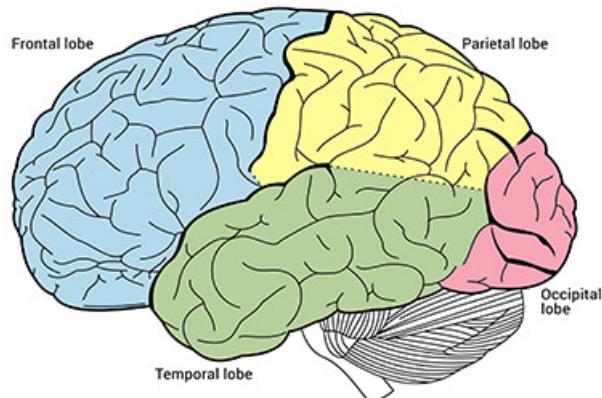


Figure 3.3: Illustration of the lobes of the human brain

The frontal lobe is responsible for intelligence, concentration, problem solving abilities and judgement as well as regulation of emotions and personality. There, also, exists a smaller area, called the Broca's area [39], which is responsible for the formation of speech and writing. The temporal lobe is used to understand speech, to form and use memories, sequencing, organizing and hearing while the occipital lobe is used to interpret visual signals . Last but not least, the parietal lobe, which is the lobe that will play a very important role in our experiments, is dedicated to interpreting external signals from motor and sensory stimulation, spatial and visual interpretation and language interpretation. Some of these functions are shown in 3.4 in relation to their appropriate lobe.

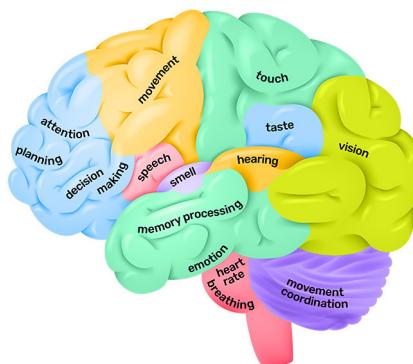


Figure 3.4: Illustration of some of the different functions each lobe of the brain is capable of controlling

## 3.2 Communication between nerve cells

### 3.2.1 Nerve cell shape

The human brain consists of a large number of nerve cells. Some researchers approximate the number of nerve cells in an adult human somewhere between 80 and 90 billion [41], [42]. Not all neurons have the same shape and size, yet some of their structural characteristics are found in almost all healthy neurons. Namely, all neurons consist of a cell body called soma, an axon and the dendrites [43]. An illustration of these shapes can be seen in 3.5 When the dendrites of two different nerve cells come close to each other, we say that a synapse is created. Synapses between neurons allow them to communicate using electrical and chemical signals, forming a large network of interconnected nodes.

### 3.2.2 The chemical excitation of the nerve cell

In order for the communication to begin, a chemical excitation must happen in a neuron [44]. Initially, when the neuron is at rest, an electric potential of approximately  $-70\text{mV}$  [45] between the inside and the outside of the cell, exists. The neuron's cell has selective permeability and is able to regulate this electric potential [46]. When the inside of the axon is depolarized, increasing the potential by reducing the negative charge within the neuron, there is a chance for the neuron to fire. If the potential surpasses the threshold of excitation, which is situated around  $-55\text{mV}$  to  $-65\text{mV}$  [47], a rapid depolarization and an inversion of the polarity is observed. This temporary hyperpolarization reaches a peak shortly after its initialization and then returns to its initial, resting state potential. This peak is propagated through the axon of the neuron, transmitting itself into neighbouring neurons through the already made synapses [48]. This whole process is clearly depicted in 3.6

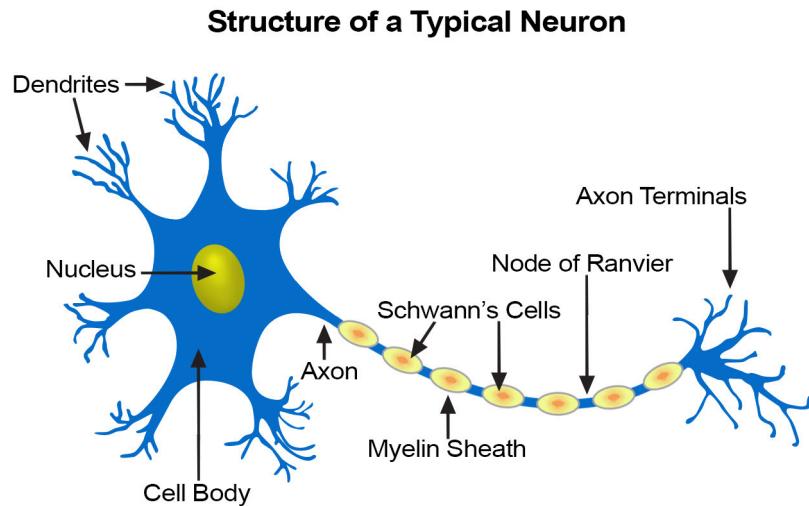


Figure 3.5: Illustration of a typical nerve cell

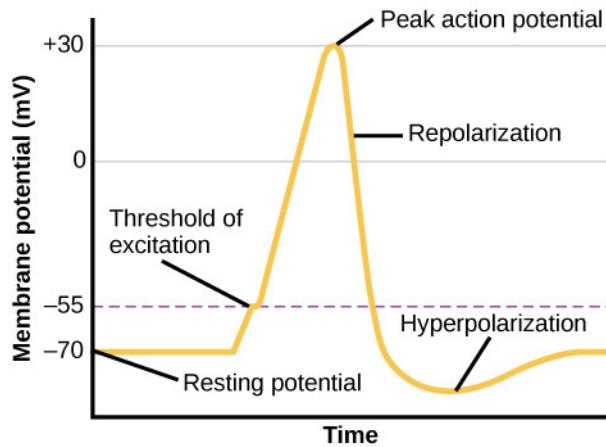


Figure 3.6: Diagram of the potential of the neuron during excitation

### 3.2.3 The excitation-inhibition mechanism

Another factor of the neuron-to-neuron communication is the excitation and inhibition mechanism [49]. Certain chemical substances act as transmitters of excitation or inhibition in a neuronal network, making the transmission of the aforementioned hyperpolarization easier or harder respectively. This mechanism plays a significant role in functions of cognition in the human brain as an imbalance of either the excitation transmitters or the inhibition transmitters may result in serious health issues. Namely, an excess of excitation transmitters might result in recurrent bursts of excitation in the brain similar to an epileptic seizure [50]. Conversely, the excess of inhibition transmitters will make the communication between neurons weaker. This imbalance has been associated with conditions such as schizophrenia and other psychiatric conditions [51], [52].

## 3.3 Electroencephalography

Having described the anatomy and the main mechanisms of neuron-to-neuron communication, we are able to understand the ways scientists have found in order to monitor the brain's activity.

### 3.3.1 Origins of the EEG

The scientist who is credited with the discovery of the brain's electrical properties is Richard Caton who recorded the electrical activity of animal's brains during sleep using a sensitive galvanometer [53]. In 1924, the German psychiatrist Hans Berger [54] recorded the first human EEG. In the following years, significant breakthroughs were made in the field of medicine after the recording of epileptiform spikes and discharges [55], establishing electroencephalography as a valid and important way of research related to medicine and psychiatry. An image of the first EEG recording made in 1921 is depicted in 3.7.

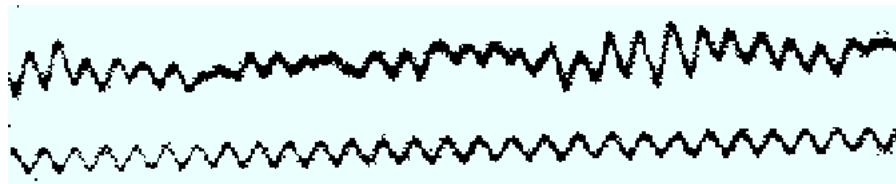


Figure 3.7: Image of the first ever EEG recording made by Hans Berger in 1921

### 3.3.2 The physics of the EEG

The phenomenon the electroencephalogram tries to record is the propagation of charge between neurons as mentioned in the previous section. The exchange of charge can be measured using an electrode to capture the voltage field and calculate it using a voltmeter. Yet it is not as simple as measuring the voltage in an electrical circuit. The activity of the brain has to be measured from the surface of the human head, the scalp. Unfortunately, the voltage field gradients are reduced with the square of the distance between the source and the electrode. This means that the already weak signals of the neurons are further attenuated due to the thickness of the skull and the meninges. Some types of EEG try to alleviate this problem by invasive means, attaching the electrodes directly onto the brain's tissue [56] (invasive EEG) or in the subdural space [57] (semi-invasive EEG). In addition to this, the signal from a single neuron is too weak to measure, so the EEG reflects the summation of the synchronous activity of millions of neurons firing at the same time in a similar spatial orientation [58].

It is evident that the electrode's material and shape play a very important role in the acquisition of the brain's activity. The most common type of electrodes is passive electrodes, which are usually made using silver or silver chloride ( $\text{Ag} - \text{AgCl}$ ) with typical wires attached to them, leading the signal into an amplifier, situated away from the electrodes. These electrodes are placed on the scalp using a type of conductive gel or paste. Active electrodes, in contrast to passive ones, try to reduce the noise of the signals and the preparation time by moving the amplifiers as close to the electrodes as possible. This makes it easier for the conductors of the experiment, to lower the impedance of the scalp without the excessive use of gel and skin preparation [59]. It, also, increases the signal to noise ratio, as it amplifies the signal before its contamination by external fields. An extension of the active electrode technology is the dry electrode [60] technology, which consists of high quality active electrodes that do not require the use of conductive gel or paste. Lastly, recent advances in materials technologies have resulted in carbon fiber based, spongy electrodes. The electrodes, when wet, are sufficiently conductive and do not require conductive materials in order to reduce the impedance. In addition, their lower cost and flexibility provide a useful alternative to the widely used, metal based electrodes [61].

In order to capture the general activity of the brain, the use of multiple electrodes is needed. The spatial resolution of an EEG is severely limited [62], which means that patches of coherent activity of sizes smaller than about 5 cm are generally not revealed, even when electrode spacing is less than 1 cm. [63] In the confines of an experiment, the recording of the activity of different lobes of the

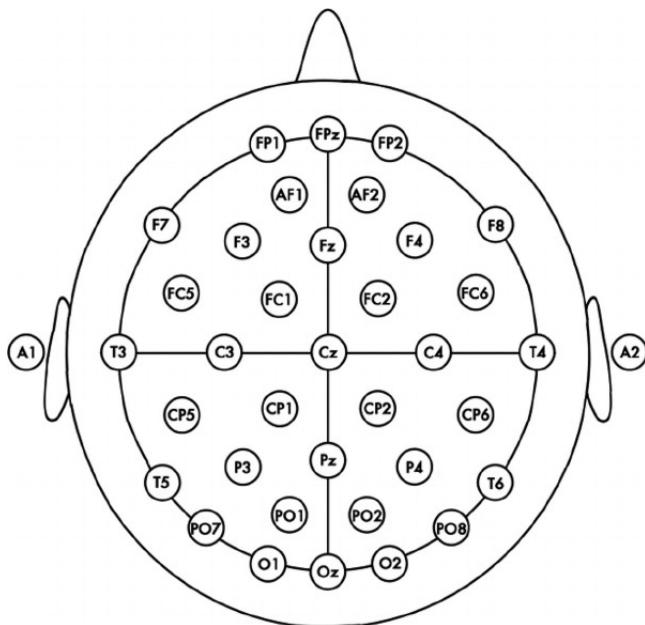


Figure 3.8: Arrangement of the electrodes on a subject's head according to the 10-20 standard.

brain might be necessary, thus the use of 16 to 256 electrodes, called channels, is applied. In a multi-electrode setting, the signal of each electrode is derived as the output of a differential amplifier [64]. The amplifier has two inputs and a number of variations of channel inputs is used. The amplification of the difference between two signals decreases the noise that affects the electrodes uniformly and the common signal that two electrodes might capture at once. In this way, the EEG is capable of emphasizing the difference between the activity of two spots of the brain rather than just their raw signals.

The placement of the electrodes is equally important to the aforementioned hardware specifications of a typical EEG recording device. We are intuitively inclined to say that the placement of the electrodes over different lobes of the brain, according to the needs of the experiment, will provide us with the best possible results. Yet the need for standardization requires of us to comply to a universally accepted placement of the electrodes. There are many standards, usually defined by the placement of the electrodes on a specific company's cap. Common examples include the Biosemi, EasyCap and HydroCel standards. The internationally recognized standard is the 10-20 standard [65] depicted in 3.8. The 10 and 20 refer to the fact that the distance between neighbouring electrodes is either 10% or 20% of the total front-to-back or right-to-left distance of the skull. The labeling of the electrodes is based on the lobe over which the electrode is placed. In particular, electrodes based over the pre-frontal lobe are labeled as Fp, over the frontal as F, over the temporal as T, over the parietal as P, over the central as C and over the occipital as O.

Additionally to the characterizing letter, numbers are used to indicate the hemisphere that the electrode is put, odd numbers correspond to the left hemisphere while even numbers correspond to the right hemisphere. The letter Z indicates that the electrode is put on the center of the head, between the two hemispheres. Electrodes that are situated between lobes share their respective characterizing letters, for example an electrode placed between the temporal and parietal lobe is labeled as TP. The 10-20 system is a higher density variant of the 10-20 system that accommodates more than 64 electrodes [66].

### 3.3.3 Brain Rhythms

Some of the most important aspects of an EEG recording are its frequency components. There are five major frequency bands, also called brain rhythms, that reflect specific functions of the brain [67].

These frequency bands, from low frequency to high frequency, are named delta ( $\delta$ ), theta ( $\theta$ ), alpha( $\alpha$ ), beta ( $\beta$ ) and gamma ( $\gamma$ ). A depiction of a typical recording of each wave is presented in 3.9.

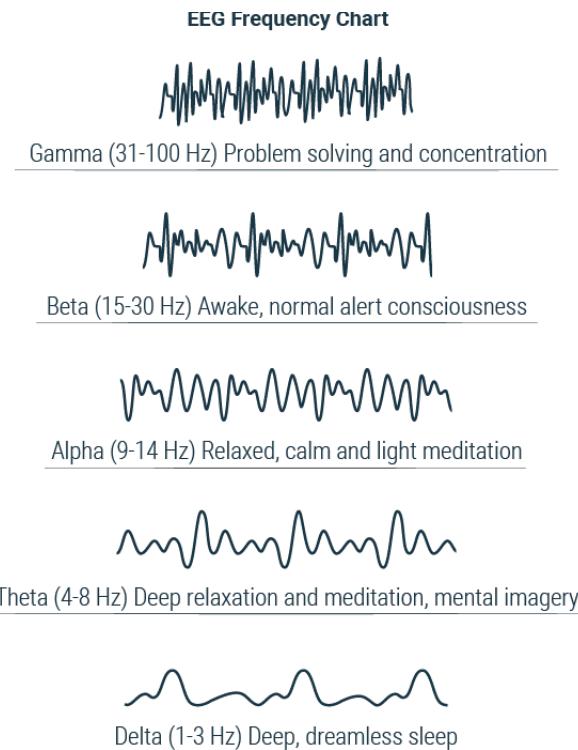


Figure 3.9: Depiction of typical waveforms of different brain rhythms.

Each one of them has been correlated with different types of brain functions. Specifically, delta waves, which are observed at 1-3 Hz, have been associated with non-rapid eye movement sleep in mammals, characterized as deep sleep, and has been a useful parameter in EEG sleep experiments [68]. Theta waves, found at 4-7 Hz, are usually correlated with light sleep or meditative states and have been used to investigate the relation of sleep deprivation or hypnagogic states with perception and other psychological phenomena [69]. Alpha waves are considered to be one of the most important waves of the human brain activity as they have the greatest relative amplitude percentage among the brain rhythms under normal circumstances, as shown in 3.10. They are found at 8-13 Hz and are correlated with a relaxed, awake, non-focused state. A link has been found between alpha wave oscillations and the regulation of cortical excitability versus inhibition, making them a great predictor of perceptual readiness and vigilance [70]. This, in particular, makes the alpha waves crucial to our efforts just like the beta waves, that are found at 14-30 Hz. Beta waves are associated with high alertness, problem solving and is an indicator of an active mind. In particular, there has been a lot of research that indicates the relation between high beta wave activity and sensorimotor tasks [71]. Lastly, gamma waves found at ranges greater than 30 Hz have been correlated with high concentration tasks including feature-binding processes [72].

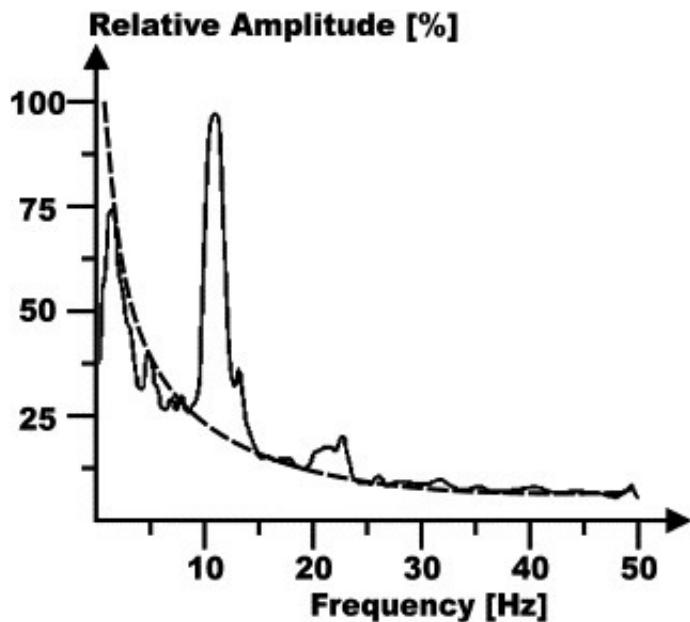


Figure 3.10: Relative percentile amplitude of the different brain rhythms that can be measured in an EEG recording.

### 3.3.4 References

As we mentioned before, the signal displayed at any channel is in fact the difference in electrical potential to some other recording site. Primarily, this other recording site is the ground electrode. As the name indicates, this electrode is connected to the ground circuit of the amplifier, and it thereby picks up electrical noise that does not reach the other scalp electrodes. Apart from the ground electrode, there exists the reference (or multiple) electrodes which have some differences.

The ground is used for common mode rejection. The primary purpose of the ground is to prevent power line noise from interfering with the small biopotential signals of interest. By design, amplifiers should not be affected by large changes in potential at both the active and reference sites. A ground electrode for EEG recordings is often placed on the forehead (but could be placed anywhere else on the body; the location of the ground on the subject is generally irrelevant). For our purposes, the ground electrode was placed on the top of the participants scalp, near the Cz electrode. The reference lead is the lead that connects the reference electrode; in EEG recordings, this electrode is usually placed at the ear or, in the case of “summed ears,” to a pair of electrodes, one at each ear. The measured electrical potential differences are ideally the voltage drops from the active electrode (connected to Vin+ on the amplifier) to the reference electrode (connected to Vin- on the amplifier).

There are some commonly used reference schemes. The one we employed, the average of the two mastoids is often used as offline reference, especially if the signal in central areas or along the midline is of interest. The electrodes on the mastoids are expected to pick up relatively little cortical activity from the top of the head. At the same time, they are located closely enough to the other scalp channels to pick up similar noise. Averaging across the left and right mastoid furthermore offers a symmetric reference that should not prefer one of the hemispheres. Dedicated mastoid electrodes can be placed behind the ears (often called M1 and M2 or A1 and A2) or the mastoids can be approximated by the electrode positions TP9 and TP10. Other methods include applying the so-called Common Average Reference, the new reference is the average electrical activity measured across all scalp channels. A new method was the reference at infinity method, which supposes that an electrode exists infinitely far from the subject, recording only the common noise that exists in space [73].

### 3.3.5 Artifacts

Just like in any signal processing task, the signals acquired by the recording equipment are not completely independent from external sources. In other fields, noise is included in the signals by physical phenomena that are not absent in our research. Yet, signals that are recorded but are not relevant to the process that we are studying are referred to by the broader concept of artifacts because they might be caused by physiological reactions of the subject and not an exclusively external parameter. We are going to make a discrimination between physiologic artifacts and non-physiologic artifacts and mention some of their most common sources [74].

The non-physiologic have their sources in the pieces of equipment we use to acquire the EEG recordings. Specifically, bad electrode contact with the scalp will reduce the quality of the recording significantly. Another source of electrode artifacts is the unequal impedance between electrodes. This may happen if a subject with dental fillings rub against each other during speech. If the subject moves or touches the electrodes during recordings this will also reduce the quality of the data. A very important artifact that is impossible to avoid is the 60 Hz or 50 Hz artifact, depending on the frequency of the power grid of the country. This artifact is the result of the electrodes working as tiny antennas, picking up the frequency of the power grid that unavoidably covers most modern day facilities. This type of artifact can be removed by filtering as we will see in a future chapter. Other non-physiological artifacts include environment-related noise, such as electronic devices that emit signals, a phone ringing or even unsheltered headphones.

On the other hand, some artifact's sources are to be found on the subject's automatic functions. Namely, the beating of the heart, the blinking of the eyes, respiration and glossokinetic artifacts are common in EEG recordings. Additionally, laughing or sneezes as well as tremors and shivers might affect the EEG recording. Swallowing, another automatic motion, is also a very common artifact that we are faced with.

Because the blinking and heart rate artifacts are so common in EEG recording we are going to spend some more time analyzing them. Starting from the blinking artifacts, the EOG has been attributed to the fact that the external surface of the cornea (at the front of the eye) is electrically charged with respect to the posterior surface of the retina (at the back of the eye) [75]. Therefore, each eyeball acts like a battery and generates an electric field that propagates across the whole surface of the head, including the scalp. The results of this phenomenon can be seen in 3.11.

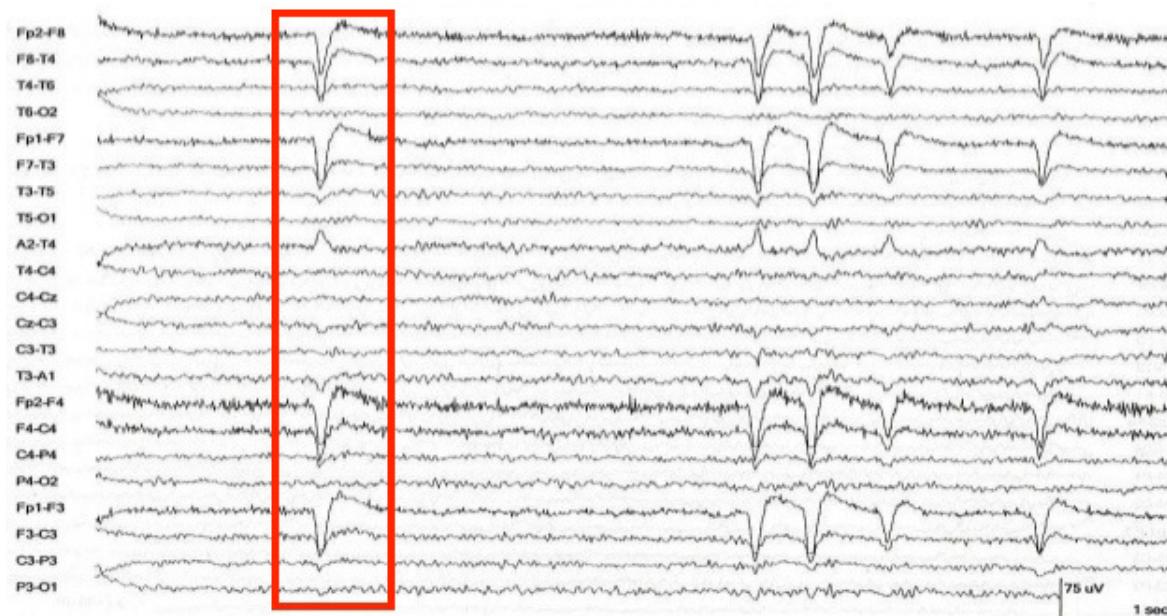


Figure 3.11: The effect of EOG artifacts on EEG recordings.

Respectively, the heart rate, or ECG artifact, is caused by the movement of the heart during a heartbeat. Its effect on the EEG is in a way similar to a traditional ECG recording, its main features being a sharp spike in the EEG followed by another spike on the negative plain [76] as seen in 3.12. An important thing to consider is that ECG artifacts are observed best in referential montages using earlobe electrodes A1 and A2.

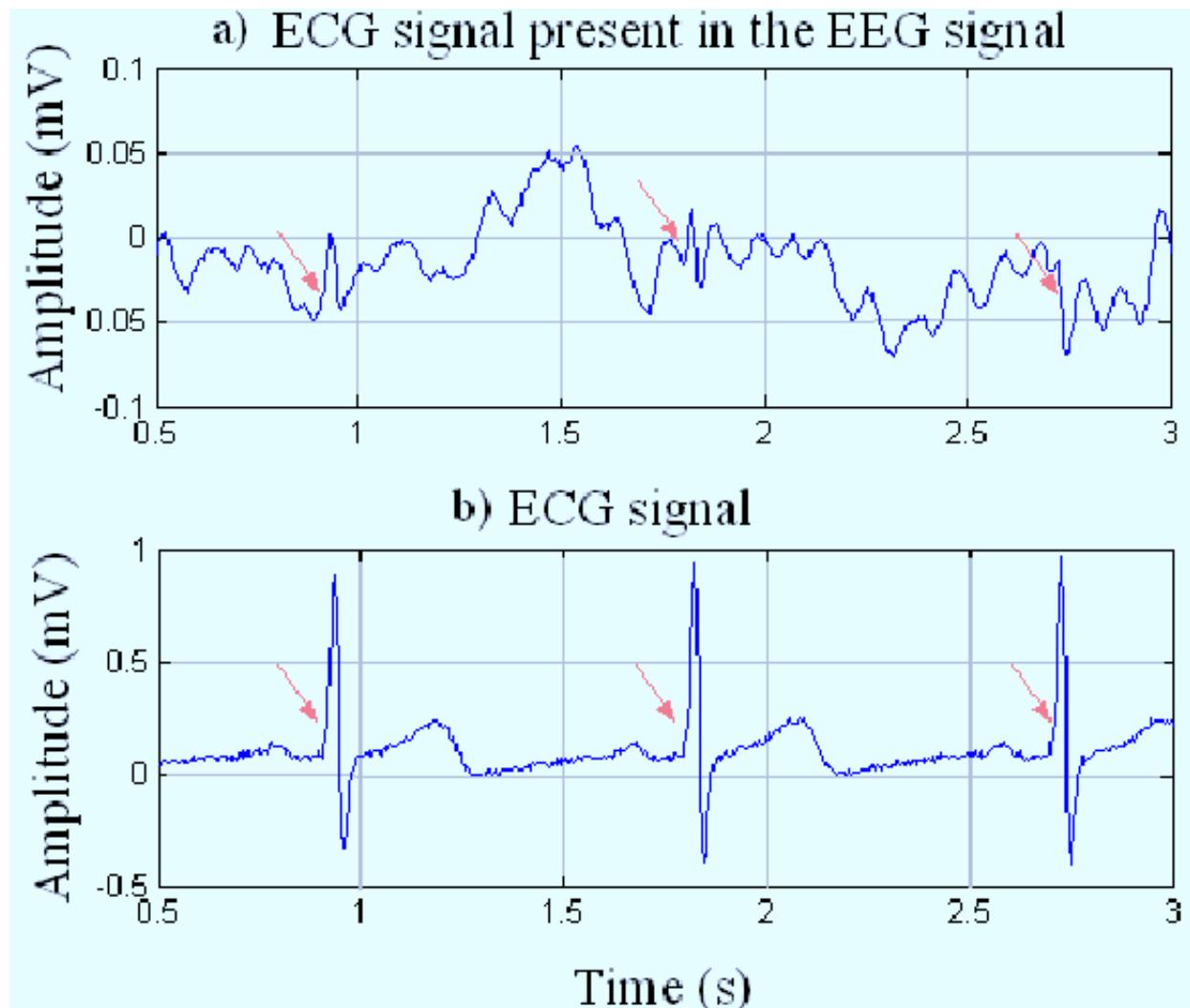


Figure 3.12: The effect of ECG artifacts on EEG recordings.

Lets not forget that in order to capture the brain's electric activities we have to record very small signals. This means that even the most trivial of causes, like a phone ringing, or swallowing, can show up in a recording. At the worst case that these artifacts appear at the moment that the useful brain activity appears, our methods of analysis might not work properly. In conclusion, we must show sufficient regard to the artifact problem and try to reduce the most common occurrences of it.

### 3.4 Localization of activity in the brain

A very important part of our work is to be able to discern between different types of activities with respect to their place of activation in the brain. Specifically, the cognitive function we would like to know about is reading as this will be chosen to be a crucial part of our experiment.

A lot of studies have been conducted to localize the source of activity of the brain when the subjects read a text. Unfortunately for us, this task seemed to be way more complex than the initial hypotheses. Posner et al. [77] were some of the first researchers to describe the complexity of the

reading task. Functions like selective attention, which means to focus only on important stimuli for the task and ignore unimportant ones, visual word formulation and semantic decoding seem to be some of the building blocks of the reading process. In their research they state that selective attention appears to use neural systems separate from those involved in passively collecting information about a stimulus. In the posterior part of the brain, the ventral occipital lobe appears to develop the visual word form. Similarly, in the anterior brain the lateral left frontal lobe is involved in the semantic network for coding word associations.

In another rich collection of locations of the brain that correspond to a specific action, Weis et al. [78] layed light in some areas that play very important role in the process of reading. In particular the supramarginal gyrus and the angular gyrus play important roles in reading. The supramarginal gyrus is a portion of the parietal lobe of the brain. It is one of the two parts of the inferior parietal lobule, the other being the angular gyrus. The supramarginal gyrus plays a role in phonological processing (i.e. of spoken and written language) and emotional responses. The angular gyrus is a hub of several networks that are involved in various functions, including attention, self-processing, semantic information processing, emotion regulation, and mentalizing. This means that we can expect a wide variety of areas of the brain to be active while reading, both in the occipital lobe, concerning the visual understanding of the characters, the parietal lobe above the positions of the aforementioned gyri, and the frontal lobe which controls grammar understanding and language fluency.

Another important factor we would like to bring attention to is the motor movement and decision making in response to an external stimulus. Movement related spectral changes can be seen in the pre-motor (decision making about movement) and motor (actual movement) cortices. In [79], researchers localized the motor-related activity of the brain to those areas using both fMRI scans and EEG recordings. Motor imagery tasks have also been studied heavily due to their importance in Brain Computer Interface design, and show similar results. This will be very useful information for the explainability of our results.



# 4. Experiment

## 4.1 The Simulation

### 4.1.1 General information and goals

As we mentioned earlier, our task is to simulate a highly autonomous vehicle supporting take-over requests of different modalities in order to examine the drivers prestimulus readiness and predict his takeover performance. In order to achieve that, we created a simulation using the Unity engine. The main assets we wanted to implement were the following.

Firstly, a driving setting (road) was necessary. Secondly, a self-driving vehicle was to be implemented. An obstacle creation event and the appropriate takeover request feedback was also included. Lastly, a distraction mechanism needed to be implemented in order to take the driver's attention away from the road. Now we are going to go through each component in higher detail.

### 4.1.2 The setting

The setting we chose for our experiment was a two lane highway in a rural area as shown in [4.1](#). The road contained curves and slopes. Decorative foliage was also implemented in order to create a more realistic representation of a road. The usual approach was to create just a straight line, without a background decorations which breaks immersion which may have implications on the attentiveness of the participant. The road was finite and needed to be looped when the car reached a certain point. The looping was done in a way that seemed normal to the participant. Fog was also implemented as depicted in [4.3](#), in order to reduce the visibility of the participant and make it harder to avoid the obstacles. It was also implemented in order to make the framerate of the simulation more stable.

### 4.1.3 The self-driving vehicle

The self-driving vehicle was made using a realistic vehicle model seen in [4.2](#). Initially, it is stopped at a certain point of the highway and when the experiment starts, it starts driving in the automatic mode. Transition between automatic and manual driving modes was implemented with no delay. The user is not able to commence the manual mode unless a takeover request is presented to him.



Figure 4.1: Preview of the setting chosen for the experiment, taken from the Unity game development software



Figure 4.2: Preview of the interior of the car in developer mode

The navigation of the self-driving car was made through nodes we placed manually on the road. The main idea was that the car, at every time, has a target node that it needed to go to. Passing through the target node, exempted the current node from the role of target node, and the next node in a list containing all the nodes was chosen for the same role. We made sure to put the nodes densely on the road in order to make the navigation as smooth as possible. In automatic mode there is an upper speed limit that is not exceeded. In manual mode the user is able to surpass it but that was a rare occurrence. Acceleration and deceleration were freely controlled by the pedals during manual mode and the participants were instructed to use them to their own avail. A reset button was implemented in order to fix the vehicle’s position and speed in case of a crash, replacing the car at the previously passed node. The conductors of the experiment were able to press the R button on the keyboard in order to reset the vehicle.



Figure 4.3: Application of fog in the simulation in order to reduce driver's visibility and increase framerate performance

#### 4.1.4 Distraction mechanism

We chose to distract the participants by including a reading comprehension task while driving. The participants were instructed to pick one of six available texts and read it carefully as they would be asked questions on the contents of the text later. All the chosen texts were IELTS reading comprehension examples at a level of B2. We chose this intermediate level because that was the lowest English level of our participants. Furthermore, we picked some of the comprehension questions for the texts in order to keep the participants focused on their task. When the participants finished one of the available texts, they were asked two questions based on what they have just read. Their focus on the reading task was measured by their performance on those questions.

The means through which the participants could interact with the texts was through a tablet placed at the tableau of the car as seen in [4.4](#).

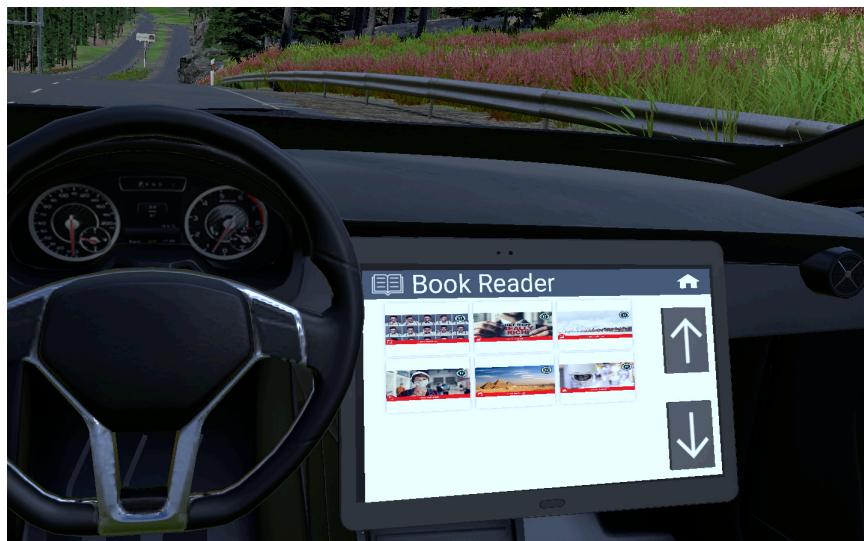


Figure 4.4: The tablet implemented at the interior of the car as a distraction mechanism

Using the Leap Motion Controller, we tracked the position of their hands [4.15](#), allowing them to use the tablet's interface with their hands. A gallery was presented to them, containing pictures that were relevant with their respective texts.



Figure 4.5: Recording hand position using Leap Motion Controller

The participants were given a choice of the topic in order to maximize their interest in the text they were about to read. Touching any of the gallery items opened the respective text 4.6.



Figure 4.6: Text chosen by the participants after navigation the tablet interface using their hands

Using their left hand, participants could display the text on their hand by turning their palms towards their bodies 4.7. This was implemented in order to physically cover part of their field of view, in order to make it more difficult to discern between true positive and false positive obstacles. The visual takeover request appeared both on the tablet and on the hand display.

#### 4.1.5 Obstacle creation and takeover request

An important aspect of the experiment was the obstacle creation, as this was the main way the user interacted with the simulation. We chose to simply create an immovable box in front of the vehicle at a distance that it would be slightly difficult to avoid the object. A green area was also highlighted that represented the end of the takeover request event. When the driver passed through that green area, the control of the vehicle was changed to auto. There were two modes regarding the obstacle creation. The first one was the true positive case in which the box was actually created in front of the car, seen in 4.8. Whenever that happened, the takeover request mechanism would notify the user

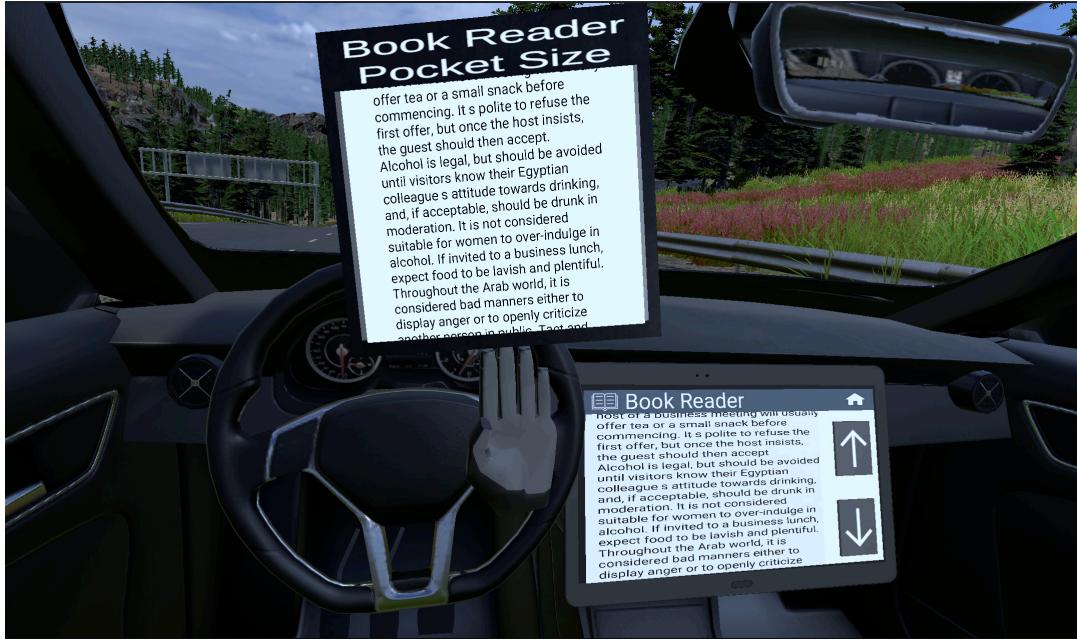


Figure 4.7: Pop-up module controlled by the left hand to magnify the text and cover part of the participant's view of the road

appropriately. The second case was a false alarm, meaning that the obstacle was not actually created in front of the car. Nevertheless, the takeover request mechanism, would still notify the participant that an obstacle was detected as seen in 4.10 and that he had to take manual control of the vehicle. This was implemented in order to simulate real life conditions in which the perceptive systems of a self-driving vehicle mistake something trivial, like a shadow, with an obstacle, requesting manual control from the driver.



Figure 4.8: Preview of the obstacle. It consists of one wooden cube that is created in the middle of the driver's lane. This is a true positive case

The takeover request system had three different modalities. In particular, a visual, an auditory and a haptic notification were implemented. The visual notification can be seen in 4.9. The conductors of the experiment were able to change the modalities with key bindings. Specifically, using the A button the modality changed to audio, using the V key the modality changed to visual and using the H key

the modality was changed to haptic. There was a tablet at the tableau of the car which was used for the distraction mechanisms that are going to be analyzed later. During the visual takeover request, a warning sign appeared at the tablet. Similarly, during the auditory takeover request, a warning recurring sound was heard. Lastly, during the haptic takeover request, there was a soft vibration on the Myo Gesture Control Armband used to give feedback to the user.



Figure 4.9: Visual feedback in a takeover request scenario



Figure 4.10: A false alarm takeover request event

Successfully completing a takeover request event rewarded the participants with a positive cue. In the visual takeover request scenario a green tick was show in the tablet,seen in 4.11 to notify the user that they successfully passed the takeover request area and that the control of the vehicle changed to automatic. The other two modalities also had a reward cue. In particular, the audio modality had a bell ringing sound notifying success and the haptic modality had similarly a rewarding vibration.



Figure 4.11: Rewarding visual cue presented to the participants after a successful takeover request event

## 4.2 Simulation system's arrangement

### 4.2.1 Simulation computer

The computer used for the simulation was a Gigabyte Technology X570 GAMING X. The operating system installed was Windows 10 Enterprise which was up to date. Its main components were: an AMD Ryzen 5 3600X ,3793 Mhz, 6 Core(s), 12 Logical Processor(s), a NVIDIA GeForce GTX 3060 Ti and an Intel SSD providing 1 TB of space.

### 4.2.2 Simulation monitor

The screen used to present the simulation to the participants was the Samsung Odyssey G9 Ultrawide Curved Gaming Monitor seen in [4.12](#).



Figure 4.12: Ultrawide Curved Monitor used in the experiment to show the simulation

The main features that made us choose this particular monitor was its great resolution at 5120x1440 pixels, the high refresh rate at 240 Hz that was more than enough for our simulation and its curve. In general, a wide, curved screen was thought as an asset that would improve the immersion of the participant as it covers most of his field of view. Additionally, the coverage of the field of view means that it will not be necessary for the participant to move his head around to look at all the parts of the simulation. This is important as, during the EEG recording, we want our participants to stay as still as possible.

### 4.2.3 Steering wheel

The steering wheel used was the Logitech G29 Driving Force seen in [4.13](#). It provides great immersion for the user as it provides realistic control of the vehicle.

Throughout the tests we conducted before the experiment we found that the vehicle in the simulation was easy to control, the steering wheel was responsive and there was no problem with the turning of the wheel, as a 900 degree turn was possible. The steering wheel was fixed on a rectangular surface placed on top of the work station. From a software standpoint, the steering wheel needed the support of the Logitech Hub software that installed all the necessary drivers.

### 4.2.4 Pedals

The Logitech G29 Driving Force used in the experiment, supports the use of pedals to control acceleration and deceleration seen in [4.14](#), as well as a gear stick.



Figure 4.13: Logitech G29 Driving Force steering wheel used to control the car in the simulation



Figure 4.14: Logitech G29 Driving Force pedals used to control acceleration and deceleration of the vehicle

We chose not to include the gear stick into our experiment because that would require from the participants to be familiar with changing gears in a vehicle. That would significantly reduce the amount of participants we could have, reducing also the quality of the dataset. The pedals were situated at the feet of the participants, adjusted before each run so as to be easy to use.

#### 4.2.5 Leap Motion Controller

In order for the user to control the user interface within the vehicle, the Leap Motion Controller was used, seen in [4.15](#), to determine the position and direction of both his hands.



Figure 4.15: Leap Motion Controller used for hand tracking. The infrared emitters are visible in the photograph

It was placed on each participants right leg as we found that was the best position in order to avoid unnecessary movement of the participants and glitches in the simulation.

#### 4.2.6 Sound system

A simple Logitech EasyCall Desktop speakerphone was used, as depicted in [4.16](#) in the experiment in order to support sounds. During testing we decided that the auditory notifications were clear to the participants and the small size of the speakerphones made it possible to hide it out of sight, in order to increase immersion.

#### 4.2.7 Myo Gesture Control Armband

The Myo Gesture Control Armband, seen in [4.17](#) was used to provide haptic feedback to the user. The main advantage of this module, was its open API and the free SDK publicly available which made its integration into our simulation seamless. The armband was placed firmly on the right arm of the participants. It was connected with the simulation computer via direct bluetooth connection.



Figure 4.16: Picture of the sound system used to produce the audio notification. The simulation computer is situated underneath



Figure 4.17: Picture of Myo Gesture Control Armband used for the haptic notification

## 4.3 ActiveTwo acquisition system

In this section we will go through the equipment used for the data acquisition.

The system used for the recording of the brain's electric potential was the ActiveTwo MKII-32+8+1 Channel EEG/ERP/EMG/EOG-Amplifier-System by Biosemi. This system provides 32 EEG channels, 8 channels for miscellaneous use, such as EMG or EOG recordings, and one external channel for event triggering. The bandwidth ranges from DC to 3276 Hz and has a maximum sampling rate of 16384 Hz per channel. The analog to digital conversion is achieved at 24 bits. Its input impedance is 300 MΩ at 50 Hz and the Common Mode Rejection Ratio (CMRR) is over 80 dB at 50 Hz. The system works using batteries that can last up to 15 hours and take 4 hours to recharge when depleted. The trigger input/output system consists of 16/15 bits respectively. Its computer interface consists of a USB 2 connector.

The main components of the system are 34 active electrodes, the AD-box, two batteries, a charger, a receiver and the cables necessary for interconnecting the different parts and the system, as a whole, with a computer. It is important to note that the aforementioned system is not authorized to be used as part of medical diagnoses and can only be part of scientific research. Now we are going to describe each part.

### 4.3.1 AD-box



Figure 4.18: Image of AD-Box on top of the Battery

The AD-box is the main module that is used for the recording of the brain's electric potential [4.18](#). On the front of the box, on the top left, we can see three ports labeled as CMS/DRL, Ergo1 and AUX3. The first port is used to connect CMS/DRL electrode sets in experiments that include recording of EMG/ECG data. The second port is used to connect the Ergo 1 channel input module. This module is used for the triggering of events by the subjects undertaking an experiment. In our experiment there was no use for this module. The third port, labeled AUX3, is used for additional sensors like galvanic skin response sensors, plethysmographs or even custom made sensors. To the right of those three ports we can see three LED indicators labeled as Power, CM in range and Low battery. The Power indicator emits green light whenever the AD-box's power circuits operate correctly. The CM in range indicator emits continuous blue light whenever the Common Mode voltage is within the normal operating limits. If the indicator is flashing it means that the voltage of our recordings is not in the normal range which means that there is a high probability our recordings will not be of high quality. The last indicator, labeled Low battery, will emit a red light whenever the battery's voltage is running low.

Additionally, in the bottom-middle of the front of the box, we can see an infrared emitter used to optically transfer the recordings either to the receiver or another computer. To the right of the emitter, we can find a rotary switch labeled speed mode. It has multiple positions that correspond to different speed settings ranging from 0 to 9. The speed mode determines the sampling rate and the bandwidth of our recordings. At a speed of 4, the sampling rate is at 2048 Hz and the bandwidth at 417 Hz. Increasing the speed setting by 1 will double both the sampling rate and the bandwidth to 4096 and 834 respectively. Due to the specifications of the ActiveTwo system we used, the only valid speed modes where the ones indicated by the number 4 to 7. Inferentially, the range of the sampling frequency is 2048 Hz to 16384 Hz and the range of the bandwidth is 417 Hz to 3276 Hz. Lastly, to the right of the speed mode rotary switch, we can find the DC input where the battery needs to be plugged in order for the system to start working.

The input for the electrodes is situated on the top surface of the AD-box. There are 8 inputs capable of supporting 32 channels each, except from the first that supports 34, including the CMS and DRL channels. To the left and right of each input socket, we can see the labels of the channels. To the bottom of the top surface, there are 8 additional, touch proof sockets that are used to connect the 8 additional channels.

### 4.3.2 Electrodes

The electrodes are the medium through which the brain electrical potential is measured. We used the electrodes depicted in [4.19](#) The ActiveTwo system's electrodes, contain a Ag-AgCl sintered electrode pellet, and a buffer amplifier with an input protection circuit. The electrodes are sealed in a water-tight resin. The electrodes can be used in connection with all commercially available electrode gels. The integration of the first amplifier stage on the electrode, allows impedance transformation on the electrode (the active electrode has an output impedance lower than 1 Ohm). This makes it possible to measure body surface potentials with via high electrode impedances, without encountering the noise and interference problems seen in measurements with high-impedance passive electrodes. The electrodes available to us were 34 in number and are connected on the surface of the AD-box. Specifically, we connect the electrodes to the socket labeled A1-A32. This socket supports the first 32 electrodes as well as two extra electrodes used as CMS and DRL.



Figure 4.19: Image of the electrodes including the 32 EEG channels and CMS/DRL electrodes

### 4.3.3 Battery-box

The battery-box is used to provide power to the AD-box and subsequently, to the whole recording system. There are two identical batteries in the ActiveTwo system. We can see one of them paired with the AD-box on 4.18. The on/off button is situated on the front of the battery to the left. Right next to it, there are two LED indicators labeled as power and shutdown. When the power indicator is green, it means that the battery is ready to provide the sufficient voltage at its output. When the shutdown indicator is illuminated red, it means that the internal protection circuit has disconnected the DC output from the battery. To the right of the two indicators, we can find a 5A fuse that protects the battery from short circuits. Lastly, to the right-most of the front of the battery we can see the DC output that is meant to be connected to the AD-box in order to provide power.

### 4.3.4 Charger



Figure 4.20: Image of the front of the charger

The charger is used to recharge the battery-boxes whenever they are depleted from continual usage. In the front of the charger, depicted in 4.20, we can find the charge output that is connected to the

DC output of the battery-box to charge it. Right next to it, there are three LED indicators labeled ready, charge and error. The ready labeled indicator illuminates a green light whenever the battery connected to the charger is fully charged and ready for use. The charge indicator turns orange when connected to a battery that is not fully charged, indicating that the charging is in progress. Finally, the error indicator emmits a red light if charging the battery is not possible due to some error. There are 5 different flashing modes for the error indicator that correspond to 5 reasons that might hamper the charging process.



Figure 4.21: Image of the back of the charger

At the back of the charger, which is shown in 4.21 there is a socket in which the DC input from the power grid is connected. To the left of that socket, there is a 5A fuse that protects the charger.

### 4.3.5 Receiver

The receiver is the module that collects all the data taken from the AD-box and the triggering devices and sends them to the recording computer. The front of the receiver contains three indicating LEDs. The back of the receiver has a receiving optical socket, to the far left of the device, that is connected to the AD-box's infrared emitter. At the center, there is a Sub-D connector that is used to transfer the triggers and the synchronization between the AD-box and the computer system that we use. Finally, a USB port to USB 2.0 is situated on the right that is used to connect the receiver with the recording computer.

### 4.3.6 Gel and syringes

The gel used in the experiments was the Signa Gel Electrode Gel shown in 4.22. This material is a highly conductive multi-purpose electrolyte recommended by the manufacturers of the ActiveTwo system.

To apply the gel we used the specialized, horn-shaped syringes provided by the manufacturers and depicted in 4.23. They are shaped like this in order for the conductors of the experiment to be able to apply the gel quickly around the curves of the head.



Figure 4.22: Image of a bottle of signa gel used in the experiments



Figure 4.23: Image of a syringe used to apply gel to the subject's head during the experiments

## 4.4 Neurospec triggering system

This section is dedicated to the Neurospec triggering system. We will discuss the hardware that is responsible for the synchronization of the computer containing the experiment's software and the EEG acquisition system.

To begin, it is important to state the necessity of a synchronization module between the computer containing the experiment's software and the ActiveTwo system. In EEG experimentation, it is really useful for the researchers to know the moments of high importance for their experiment. For example, in an experiment that focuses on reactions to a stimulus, the researchers need to know the moment that the stimulus is presented to the subject and the subject's reaction to said stimulus. Since the reaction time is on the scale of milliseconds, timing between the onset of the stimulus and the marking of the event onto the EEG recording must be very precise. Doing this using time measurements is also very difficult as the recording and the stimulus producing software must commence on the same time. Even a slight delay of one system or the other can produce significant problems to the experiment, like shortening or elongating the subject's reaction time. This means that the use of a synchronization module is necessary for any type of experiment that contains timed events like ours.

The synchronization module that we used came with the ActiveTwo system and was produced

by Neurospec. The module is named MMBT-S Trigger Interface Box shown in 4.24. Apart from the main module, we used the necessary connectors for the communication of the box with the simulation computer and the recording computer.



Figure 4.24: Image of the MMBT-S Trigger Interface Box

#### 4.4.1 MMBT-S Trigger Interface Box

The box itself is shaped like the letter N. On one side of the box there is a D-SUB port, shown in 4.25 that connects the module with the ActiveTwo receiver discussed on an earlier section.



Figure 4.25: Image of the MMBT-S Trigger Interface Box's SUB-D port

Additionally, the box has a USB port that is connected with the simulation computer. A LED indicator is situated near the USB port that flashes whenever a triggering signal is transferred between the simulation computer and the recording computer. Next to the USB socket, a switch is located that is used to control the runtime of the trigger signal pulse, which will be discussed further in a future subsection. The aforementioned system's features are depicted in 4.26.



Figure 4.26: Image of the MMBT-S Trigger Interface Box's USB port and runtime indicator

#### 4.4.2 MMBT-S features

The MMBT-S box has a sub-millisecond trigger marker transfer delay, which makes it suitable for the synchronization of events that take place on the scale of milliseconds. The triggering itself has a capability of 8-bit data transfer, giving the user more than enough signals to use for the purpose of their experiment. It is compatible with all major operating systems and supports Psychological Experiment and Stimulus Presentation Software. It is, also, compatible with most standard biosignal amplifiers that support trigger marking.

#### 4.4.3 Modality

The MMBT-S box has two modes of operation, Pulse Mode and Simple Mode. In Pulse Mode, the MMBT-S trigger interface box automatically sets trigger markers LOW after 8ms HIGH period; whereas in Simple Mode the MMBT-S trigger interface box keeps the appropriate trigger marker HIGH, which needs to be pulled LOW manually. The runtime mode of the MMBT-S trigger interface box can be adjusted using the runtime switch next to the USB socket. Using the Simple Mode, a researcher can specify long regions of the EEG that might be of interest, or simply label a section of the EEG, a practice used especially during the initial states of the EEG that are used to measure eye-movement activity and resting state. It, also, provides lower trigger mark period (under 8ms provided by the Pulse Mode) if needed. In our experiments, the Pulse Mode was sufficient as we wanted to mark the starting point of events and not larger sections of the experiment.

#### 4.4.4 Recording computer

The computer used for the recording of the brain activity was an HP Omen Laptop 15-en0xxx series. The operating system installed was Windows 10 which was up to date. Its main components were: an AMD Ryzen 5 4600H with Radeon Graphics, 3000 Mhz, 6 Core(s), 12 Logical Processor(s), a NVIDIA GeForce GTX 1650 Ti and a Samsung SSD providing 480 GB of space half of which was used in the Windows partition.

#### 4.4.5 Remote control monitor

A Samsung Smart TV was used to display the screen of the simulation computer through Team Viewer remote access control. Due to the large resolution of the display monitor, it was impossible to control

the simulation computer from the recording computer while at the same time monitoring the EEG through the recording software. We extended the screen of our recording computer onto the smart TV which was sufficient in order to control the environment of the experiment.

## 4.5 Brainwave acquisition and triggering software

After discussing the hardware that was used in the experiments, it is time to discuss the software that was used to bring everything together and make the whole setup work as intended.

The first software that needs to be mentioned are the necessary drivers for the triggering system. Neurospec provided us with a USB drive that contained the necessary MMBT drivers and the Tachyon software, used to send trigger markers to the EEG recording computer. This software was installed on the simulation computer and some initial tests were made to see the trigger marks on a live recording.

On the part of the recording computer, three different pieces of software were installed, also provided by Biosemi in a USB drive included in the ActiveTwo system. The first is the LabView Runtime Engine that contains the necessary drivers for the ActiveTwo system integration. The installation begins with the setup of National Instruments Package Manager and the LabView Package. This is necessary in order for the next software to work. The next thing we installed on the recording computer was the Actiview902 software that is used for live visualization of the recording.

The software that defines the events that need to be sent through the MMBT-S module was written by us in Python 3.9. The program uses the libraries: PySerial, PsychoPy, Pynput and PyGame. The first library is used for the communication of the computer with the MMBT-S module. PySerial allows the user to receive and send messages through COM ports. At the start of the experiment, we look at the simulation computer's device manager to see at which COM port the MMBT-S module is connected. PsychoPy is a library used to create software that covers the needs of a psychological experiment. Pynput is used to read keyboard input so that we can create the events at will. Lastly, Pygame is used to read input from the steering wheel which was not supported by Pynput.

### 4.5.1 Actiview902

The Actiview902 software was used for the live visualization of the participant's brain activity. We set the program to display the 32 channels we connected on the participant's head and the two extra channels, EXG1 and EXG2 that were used for reference. During the display we filtered the incoming data with a 2Hz high pass filter and a 40Hz low pass filter. These filters were only visually applied and did not have an effect on the recorded data. The filters were mainly used to verify that all the electrodes were placed correctly. If an electrode was not in contact with a participant's head, high frequency waveforms were present in the respective channel.

Another way to verify the good contact of the electrodes was the offset of the channels. In the offset tab, a visual representation of the difference measured between the CMS/DRL channels and any other channel. If the CMS/DRL channels are in good contact with a participant's head, a high offset is a great indicator that a channel does not have great contact and therefore, picks up higher amplitude external signals.

Lastly, the Actiview902 software provides a bar at the bottom of the screen that shows the incoming triggers. During the experiment, the conductors payed close attention to that bar in order to make sure that the participants followed the instructions correctly. For example, if the participant did not grasp the wheel during a takeover request, even though he was told to do so, the conductors could see that a steering wheel grasp trigger did not show up. In that case, they could incite the participant to grasp the wheel in future events and discard the current event.

### 4.5.2 Python script for trigger synchronization

We wrote a python script using the aforementioned Python libraries in order to synchronize the events of the simulation with the EEG recording. Firstly, we declare a connection with the MMBT-S trigger interface box using PySerial. This connection was made through a COM port that changed whenever the simulation computer shut down, so it had to be inputted manually every time it happened. We used Pynput's keyboard listener in order to send writing requests to the MMBT-s system when a specific key was pressed. Three keys were used to change the modality of the takeover request notification. Pressing V changed the modality to visual, A changed it to audio and H to haptic. The two keys used to pass the obstacle creation events to the MMBT-s were P and O. Depending on the modality chosen beforehand, pressing P created an obstacle and alerted the participant with the chosen notification. Pressing O, enabled the notification without actually creating the obstacle. This was used to create the false alarm events. These two keypresses also sent a write request to the MMBT-S.

Another point of interest we wanted to synchronize with the recordings was the reaction of the participants to the external stimulus. This was done by marking the moment when the participants touched the wheel. We were able to get data from the steering wheel using Python's library Pygame, by declaring it as a joystick. We observed that when participant's touched the wheel, it inevitably steered one way or another, even for a small margin. Using this, we monitored the position of the wheel after every obstacle creation event and marked the first deviation from its initial state.

Knowing when a crash occurs was also crucial for our analysis. For this reason, we bound the D key to mark a crash with the obstacle, deleting it in the process. Also, in case the participant did not crash with the obstacle but lost control of the vehicle, by pressing R, the conductors reset the vehicle while at the same time marking the respective event. In particular, the following table describes how the codes sent to MMBT-S associate with the events.

Event Description	Code
Real Obstacle Visual Feedback	1
Real Obstacle Audio Feedback	2
Real Obstacle Haptic Feedback	3
False Alarm Visual Feedback	4
False Alarm Audio Feedback	5
False Alarm Haptic Feedback	6
Wheel Feedback	7
Crash	8
Vehicle Reset	9

Table 4.1: Codes sent to the MMBT-S in relation to the events of the simulation

## 4.6 Experiment protocol

### 4.6.1 Briefing

The subjects were greeted by the conductors of the experiment and were lead to the room where the experiment would take place. They sat on their seat and were given time to familiarize themselves with the Leap Motion Controller and the driving simulation. More specifically, while the conductors attached the electrodes to their heads, they changed the mode of the simulation to manual. The participants were asked to drive using the steering wheel and the pedals in order to get used to the sensitivity of the wheel and the responsiveness of the pedals. A minimum of three loops was completed by all

participants. Additionally, they were requested to look through the given texts without reading them. The titles of all the texts were read to the participants before starting the experiment. Familiarizing themselves with the Leap Motion Controller was crucial for the experiment, so they were given ample time to get used to navigating through the menu and opening the palm reader. They were also advised to get their hands out of the Controller's field of view in case it did not work properly, in order to reset their hand's tracking.

When the participants felt ready, the lights were turned off and a small session began in which the conductors of the experiment explained to them what was about to happen. They were told that their two main goals were to keep the vehicle from crashing and to read at least one of the texts to the end. Afterwards, the takeover request system was explained to them. They were told that the conductors would create obstacles in front of their vehicle and that they could control the vehicle whenever a takeover request happened. They were informed about the different types of takeover notification modalities but they were not given examples in order not to become desensitised in the notifications. It was made clear that they needed to grasp the wheel at any case, with or without an obstacle, in order to have data on their reaction times. It was also mentioned that they had to get comfortable in their seat as they would be sitting there for approximately 20 minutes. When the briefing finished, they were left alone in the room and the conductors of the experiment started the simulation.

#### 4.6.2 During the experiment

We remind the reader that the conductors were able to control the simulation using a TeamViewer connection. When the experiment started, the first three events were always real obstacles and the notifications were one of each modality. We did that in order for the participant to experience all the different modalities and be prepared for the rest of the experiment. After that, the events were selected randomly in order for the participant to not be able to predict the next event and prepare for it. False alarms were favoured when the participant did not have a clear view of the road because we saw that their reactions were perfect otherwise. If they found the obstacles hard to evade, they were given multiple events of the same modality in order to get more used to them and eventually evade the obstacle. During the experiment, they were also advised not to move if the conductors saw large movement artifacts in the recording and not to speak unless they had something important to remark.

All the experiments had a point at which multiple events were given back to back and another point at which events were creates sparsely, in order for the participants to get more distracted with the texts. In the end, the number of real obstacle events was around 10 for each modality and the number of false alarms around 5 for each modality. After the 20 minutes, the participants were asked to provide a small summary of what they read and they were asked some of the comprehension questions. When they completed that last task, they were thanked for their time and left the lab.

### 4.7 Subjects

The EEG dataset was collected from 17 subjects with normal or corrected-to-normal vision. No subjects reported a history of psychiatric disorders, neurological disease or drug use disorders. All the subjects were recruited university students at University of Patras. The subject's ages ranged from 19 to 23 years old. The gender of the participants was split in half, with 9 female and 8 male participants.

# 5. EEG Processing

## 5.1 MNE Terminology

MNE [80], the main tool used for the processing of the data, is an open-source Python package for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, NIRS, and more. Due to its high organisation, the package has a few terms that we are going to use that need some clarity.

### 5.1.1 Raw

Raw objects hold continuous data (preprocessed or not), typically obtained from reading recordings stored in a file. Raw objects are by default in `fif` format, but a wide variety of formats are supported by the library. Raw objects also contain an `info` section. This data structure is where the measurement info is stored and it behaves like a dictionary. It contains all metadata that is available for a recording. However, its keys are restricted to those provided by the `FIF` format specification, so new entries should not be manually added.

### 5.1.2 Epoch

Epochs (sometimes called “trials” in other software packages) are equal-length segments of data extracted from continuous data. Usually, epochs are extracted around stimulus events or responses, though sometimes sequential or overlapping epochs are used (e.g., for analysis of resting-state activity).

### 5.1.3 Evoked

Evoked data are obtained by averaging epochs. Typically, an evoked object is constructed for each subject and each condition, but it can also be obtained by averaging a list of evoked objects over different subjects.

## 5.2 Standardization

There are some minor problems we need to solve before starting the preprocessing of the EEG recordings. The point of recording external channels was to create a common reference for each subject. Additionally, we need to extract the events from the data in order to know the exact point in time when each event occurred. Lastly, we need to set the montage to the standard 10-20 system.

### 5.2.1 Removing excess channels

During the experiment we recorded data from 32 electrodes placed on the subjects’ head and 2 electrodes placed behind their ears, on their mastoid bone. Additionally to those 34 channels, 6 more

external channels were included in the recording. However, these channels were not placed on the subjects, nor were they connected with an actual cable. Despite being flat recordings, we need to manually remove them from the dataset.

The MNE library, apart from supporting most file formats for EEG recordings, has a lot of functionalities for manipulating them. After loading the data, we used the command to drop the excess channels. Dropping any channel requires the channel's name. We made sure during the experiment to always connect the reference channels at ports EXG1 and EXG2. This made it possible to drop the channels EXG3 through EXG8 for all the subjects. This is of particular importance, as most methods that we use for the processing of the recordings have some strict statistical specifications and assumptions, like following a gaussian distribution and so on. Including flat recordings would raise various errors in said methods.

### 5.2.2 Extracting event data

We remind the reader that the equipment we used allowed us to synchronize the events we created in the simulation and save their exact times in the recording. Table 4.1 has a list of the codes assigned to each event for reference. These events are saved in an additional channel named Status channel. Status is for the most part a flat recording with spikes of specific length at the time of each event that corresponds to the event's code.

The MNE library gives its user the possibility to extract events from these kind of channels. We used the command find events to extract the event's times given in numbers of samples. In order to automatically find the events, we appointed the Status channel as the stimulation channel. This command returned an array that included all the events of each recording. Every element of the array corresponds to one event and contains the exact sample it was recorded on, the numbering of the first sample of the recording and the code of the event.

One problem was noticed during this procedure. In some recordings, the Status channel had an initial value that dropped down to zero a few seconds after the recording began. This was picked up by MNE as an initialization event, which is a very useful in general, but in our case was unnecessary. To solve this, after creating the events, we went through them and checked for the code of the initialization event, which is always set at 65536 by MNE. We had to look through the whole dataset as it was not always the first event recorded. After excluding this event from the data, the rest of the events were annotated on the EEG recording in order to make it easier to extract the data in the future. This was done by creating annotations from events and setting said annotations on each raw file. After that, the Status channel was not needed anymore, so it was also dropped.

### 5.2.3 Setting a standard montage and channel references

Montage, in the MNE terminology, is the set of data that specify the positions of electrodes on the scalp. In particular, while layouts are 2D locations, montages are 3D locations. A montage can also contain locations for HPI points, fiducial points, or extra head shape points. MNE provides some frequently used montages, including the standard 10-20 model that we used to get the recordings. In order to create a montage item, all the channels must, first, follow the standard naming of the channels specified in the respective model. The channel naming by the ActiveTwo system complies with the 10-20 standard requirements for the 32 electrodes placed on the participants' heads. We needed to rename the reference channels, as the generic naming of EXG1 and EXG2 was used. The standard 10-20 system names the reference channels as A1 and A2 respectively, so we used the rename channel command provided by MNE to do so.

Last step of this process was to set the montage onto the raw file and set the A1 and A2 channels as reference. After setting the montage and the reference, the A1 and A2 channels were dropped as their information was already included in all the channels by using them as reference.

## 5.3 Downsampling

One of the most important aspects of signal processing is sampling rate. The sampling process is used to reduce a continuous signal into a discrete one. The sampling rate is defined as the number of samples taken from a continuous signal in one second and it is measured in Hz. As we mentioned before, the sampling rate of our equipment is set at 2048 Hz. This is a large sampling rate, used to get a high signal resolution of the brain activity, but in the context of the signal processing methods we are going to use, it is redundant. Hence, we use the resample method provided by the MNE library. This method requires a new sampling frequency which was set at 250 Hz. This was done in order to reduce the computations needed for the following processing steps we applied.

## 5.4 Epoch Extraction

In order to address the problem at hand, we need to specify the time intervals of the EEG recording we need to process. We remind the reader that we want to find out if it is possible to estimate if the driver is distracted by another task, namely a reading task, up to a level that it will not be possible for him to take the control of the vehicle without causing an accident. In fact, we do not need the entire recording as the useful information for our problem resides just before the creation of obstacles.

### 5.4.1 Classification problem

For the classification problem, the important distinction between trials is whether or not the participants crashed their vehicle at the box obstacle or at the sides of the road. To make this possible, we check all the events of the recording that were already attached to the raw object. We store the contents of events that have a code assigned to an obstacle creation event. Passing through the following events, if we get to another event designated to an obstacle creation, it means that the trial was a successful evasion of the obstacle. In that case, we append the previously stored event at the new event list and store the new event in memory. If we get to an event that has the code 8 or 9, it means that there was a crash. In this case, we store the obstacle creation event with an altered code, specifically 10, to indicate that this trial had a crash.

The resulting list contains events that have codes from 1 to 6 or 10. The events that signify the holding of the wheel and the crashes, namely 7, 8 and 9 are not considered in this problem.

### 5.4.2 Regression problem

For the regression problem a different kind of event was needed. We not only needed to know when a crash happened or not, we also wanted to know the reaction times in order to compare the crash and the evasion cases. We remind the reader that a special event was created to signify when the participants grasped the wheel. According to 4.1, the wheel grasping event has the code 7. In order to find the reaction time, we need to calculate how many samples intervene between the obstacle creation events (1 through 6) and divide it by the sampling frequency. MNE provides the tools to do this process automatically.

We used the define target events function to calculate the time interval between obstacle creation events and grasping events. This function provides several tools for filtering these events. The most useful out of them was the minimum and maximum time constraints. Some of the events were not correct. In particular, some of the participants did not grasp the wheel during an event resulting in unrealistically long reaction times. Some others kept their hands on the wheel between events resulting in very short reaction times. In order to filter these events out, we decided to set a minimum reaction time of 300 milli seconds and a maximum time of 1.5 seconds.

After this process, two new arrays were created. One array contained the obstacle creation events that passed the filtering stage described above and the other contained the respective reaction times. The next step was to separate the crashes from the evasions. We used the same methods as in the classification problem, described in the previous subsection. This method only changed the codes of the crash events. We separated these events looping through the events and assigning events on two different arrays, one dedicated to crashes and one to evasions, along with the respective reaction times. We also kept the general array containing all events with their reaction times. We used the general array for the purpose of the regression problem and the other two arrays for statistical analysis.

## 5.5 Filtering

We chose to filter the data based on the basic waveforms described in chapter 5. The frequency zones discussed were the delta waves at 1 to 3 Hz, the theta waves at 3 to 7 Hz, the alpha waves at 7 to 14 Hz, the beta waves at 14 to 30 Hz and the gamma waves above 30 Hz. Considering the problem at hand, delta waves that are correlated with deep sleep are not useful and might even contain some low frequency drift noise that we want to avoid. Thus, we chose to use a high-pass filter above 2 Hz.

Another parameter we have to consider is the noise attributed to the power line frequency. This experiment was conducted in Greece where the power line frequency is 50 Hz. Getting rid of this frequency can be achieved in two ways. Either with a low pass filter before the 50 Hz threshold or with a notch filter exactly at 50 Hz. The first option is better at discarding the noise since notch filters are not perfect and a low-pass filter at 40 Hz, for example, attenuates the signal way before the 50 Hz threshold. The second option is better if we want to keep information close and above 50 Hz. We chose the first option as we considered that most of the information we need resides in alpha-beta waves.

The filter chosen in the end was a band-pass filter between 2 and 40 Hz. The designing of the filter was done automatically by MNE with the following specifications. Firstly, the filter was a Finite Impulse Response filter or FIR filter for short. This type of filter is most commonly implemented in software. The technique used to create this filter was using a Hamming Window. This technique employs a window function in order to be used as a filter depending on the specifications of the filter, like the cut-off frequencies. A depiction of the window function can be seen in 5.1.

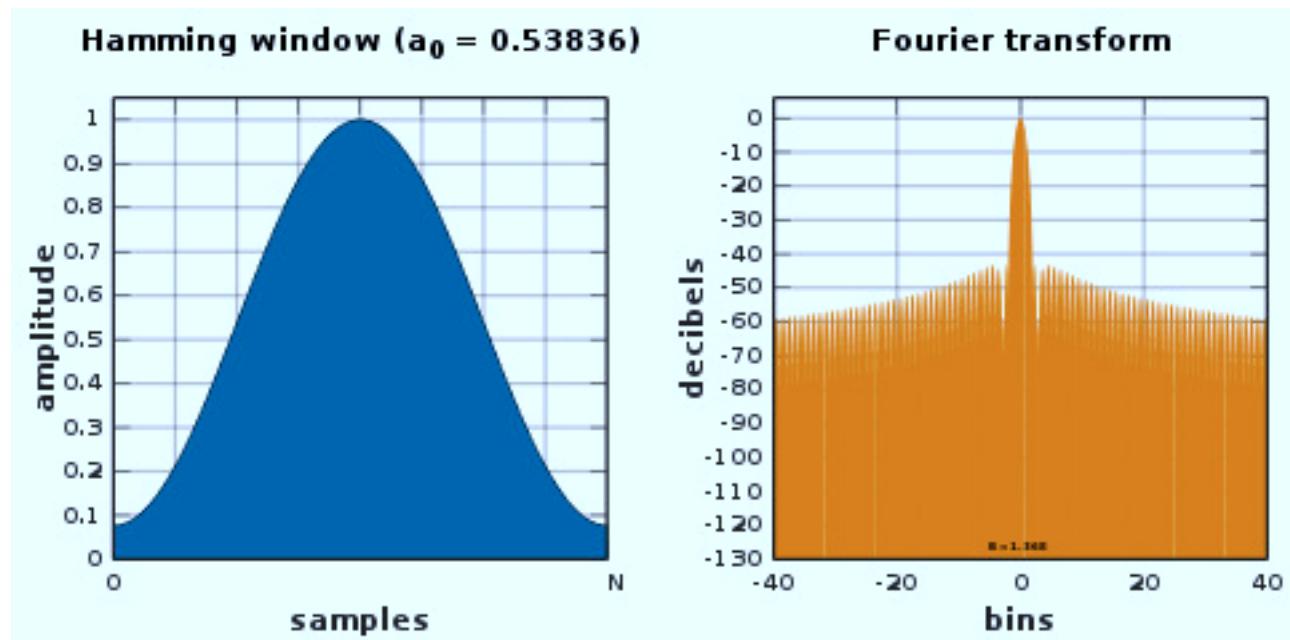


Figure 5.1: Graph showing the Hamming window in time and frequency domains

This method was chosen as the Hamming window has narrower transition bandwidth and minimum bandstop attenuation/decibel, making it a better choice for digital filters than other window functions like the rectangular and the Blackman Window [81].

## 5.6 Artifact Removal

As we mentioned in chapter 3, artifacts are a very serious issue that requires our attention if we want to consider our recordings credible. The main artifacts that we are going to try and repair are the blinking and heart rate artifacts.

The method we are going to use is called Independent Component Analysis or ICA for short. In general, ICA is a statistical and computational technique of revealing hidden factors that underlie sets of random variables. In our case, the EEG signals, being stochastic in nature, play the role of the random variables. It is supposed that these signals are a linear mixtures of latent signals that are observed on all channels with different magnitude. With this logic, the eye-blanks that affect every channel differently, can be extracted by ICA. This problem is usually called the "Cocktail Party Problem" and has many applications in fields outside of cognitive sciences like economics, sound engineering etc.

From a mathematical standpoint, ICA tries to transform our initial signals into a maximally independent set of signals. There are two key assumptions we have to consider before applying ICA. Firstly, the signals we try to uncover must be statistically independent. This is assured by the fact that the signal containing the eye-blanks is generated only by the blinking and not by the general brain activity. Secondly, they must be non-Gaussian, which also stands as EEG signals in general are not Gaussian signals and the changing of the participant's state enforces this argument.

We applied the ICA decomposition algorithm provided by Python's MNE library. The first step of the algorithm is to scale the data into unit variance and whiten them. Whitening is a step applied to EEG signal processing in order to make the assumption that the additive noise inserted into the channels is white noise, meaning gaussian with zero mean, and subsequently, be able to apply techniques such as adaptive spatial filters [82], [83] which require the noise to be white [84]. In our case, whitening is done by scaling all the data from every channel by the standard deviation across all channels. The goal is to remove the inter-channel correlation. This, also, helps in the case of different channel types, where EEG and MEG channels measure their signals in different units of measurement.

After this step, Principal Component Analysis or PCA is applied to the data as the last step of the whitening process. PCA is a popular dimensionality reduction technique used in machine learning applications. The steps of the PCA method are: the standardization of the data and the computation of the covariance matrix. The principal components are identified using the eigenvectors and eigenvalues of the covariance matrix [85]. The construction of the principal components is done in such a manner that the first principal component has the largest possible variance in the data set. In practice this is done by ordering the eigenvectors by their eigenvalues in descending order. After this step, we chose whether to keep all the principal components or discard the least significant ones by the ordering we applied in the last step. Lastly, we recast the data along the principal components axes to transform them linearly into a new space. If we reduce the number of principal components, dimensionality reduction occurs. In our case, recasting the original data onto the principal components axes removes the correlation between channels, resulting in whitened EEG signals.

The next step is passing the decomposed components of the PCA algorithm to the ICA algorithm [86]. Specifically, the algorithm used was the FastICA algorithm described in the following paper [87]. The aforementioned step, requires from us to specify how many components we want to keep. We choose to keep 32 components, equal to the number of channels. In this way, we hope to discern the 32 different signals captured by each one of the channels. Doing that will result in signals that have distinct features which we can choose to exclude by inspection. Components that had the characteristic spikes described in 3, related to blinking and ECG artifacts were removed. Components with other

artifacts like movement artifacts were removed in the case the movement happened before the events of the simulation, in order to clean up the pre-stimulus time field.

In the following images we can see the effect of the ICA technique. In figure 5.2 we can see the recordings of the EEG just as we got them from the recording system. We can see some obvious artifacts around almost every channel in specific intervals. We can see that the further from the front of the head, the electrodes are placed, the eye blinking artefacts become less powerful but they still exist.

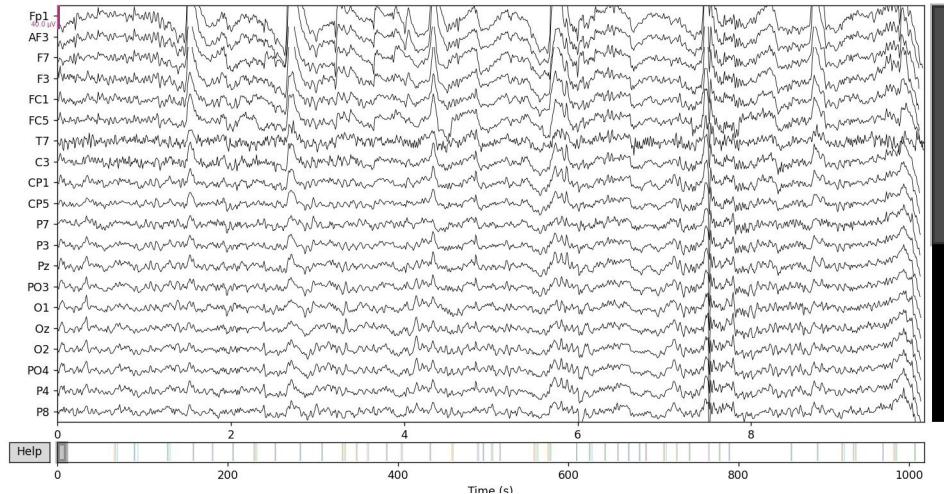


Figure 5.2: Unfiltered display of the first 20 electrodes

The following figure 5.3, shows the different components extracted from the algorithm. According to our previous analysis, we can see clearly that the first component (ICA000) contains the blinking information. Additionally, the components 4 and 9 contain other types of artefacts that we talked about earlier, like head movement (4) and bad electrode point of contact (9).

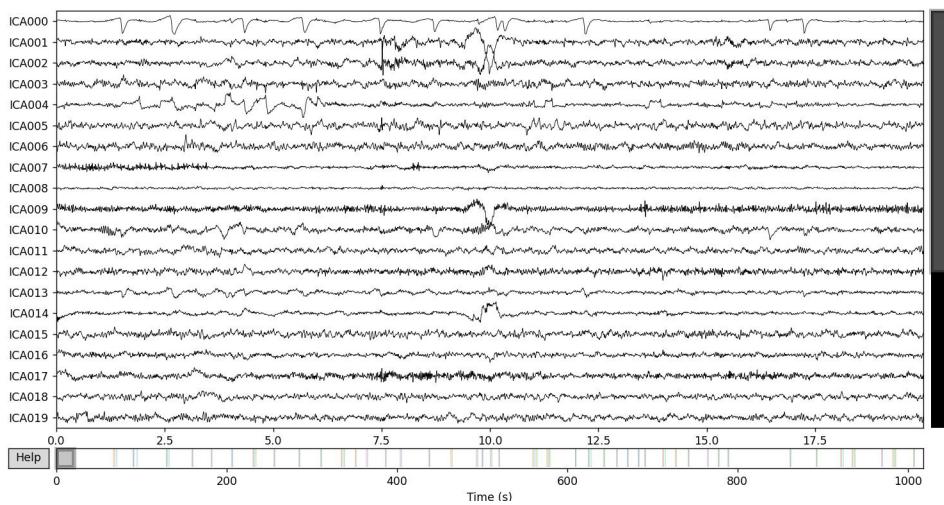


Figure 5.3: Extracting the 32 ICA components

We chose to remove the aforementioned components and the following diagrams show the difference between the initial EEG recordings (red) and the recording after the artefact removal (black).

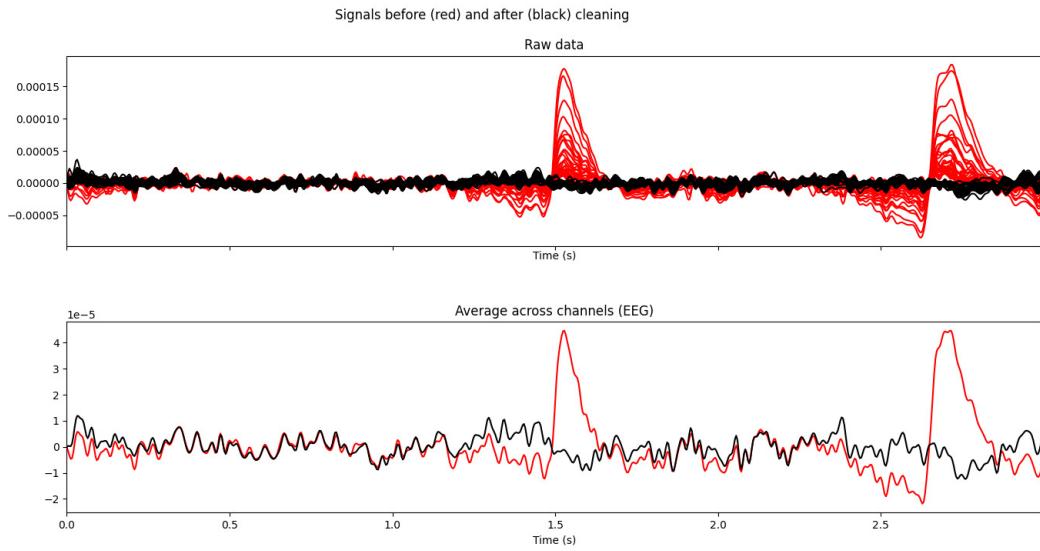


Figure 5.4: Difference between unfiltered (red) and filtered (black). The above diagram depicts all the channels and the below average across the whole EEG recording.

Lastly, we can see the effect of our method on the data. In figure 5.4 we can see that the huge peaks that were the eye blinking artefacts have been removed from the data. Also some smaller peaks have been removed that were probably attributed to head movement and heart rate. The high frequencies have also been nullified to some extent. The practical results of the method can also be seen in figure 5.5 in which we can see that the large artefacts of figure 5.2 have been removed.

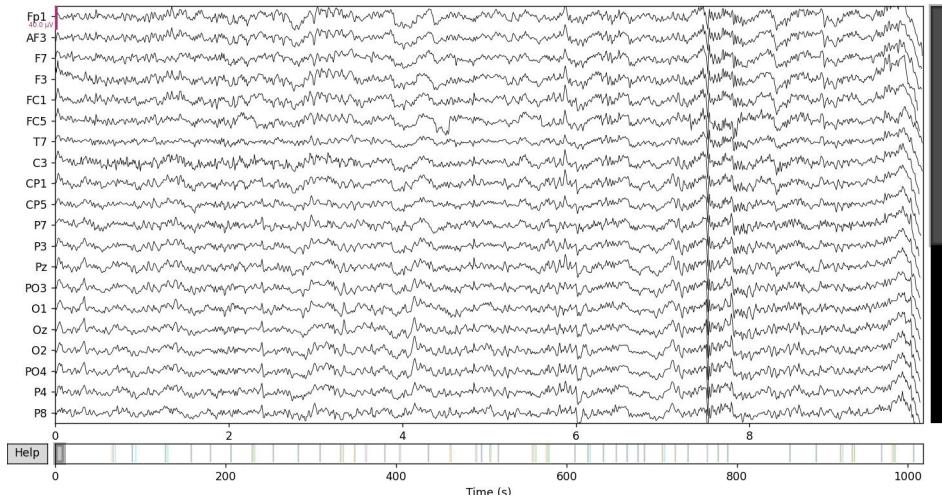


Figure 5.5: Display of the first 20 electrodes after the filtering

## 5.7 Common Spatial Patterns

Due to the volume conduction multichannel electroencephalogram recordings give a rather blurred image of brain activity. Therefore spatial filters are extremely useful in single-trial analysis in order to improve the signal-to-noise ratio [33]. Furthermore, spatial filtering gives better discriminative power between mental processes that are controlled by different regions of the brain. Considering our main hypothesis that a distracted driver would showcase higher activity in the frontal lobe as a result of concentration on a reading task, spatial filtering could provide us with a useful preprocessing step before conducting the classifier training.

Common Spatial Patterns or CSP for short, is a technique to analyze multichannel data based on recordings from two classes (conditions). CSP yields a data-driven supervised decomposition of the signal that projects it from the original sensor space to the CSP space. In a nutshell, CSP filters maximize the variance of the spatially filtered signal under one condition while minimizing it for the other condition. Since variance of band-pass filtered signals is equal to band-power, CSP analysis is applied to approximately band-pass filtered signals in order to obtain an effective discrimination of mental states that are characterized by ERD/ERS effects.

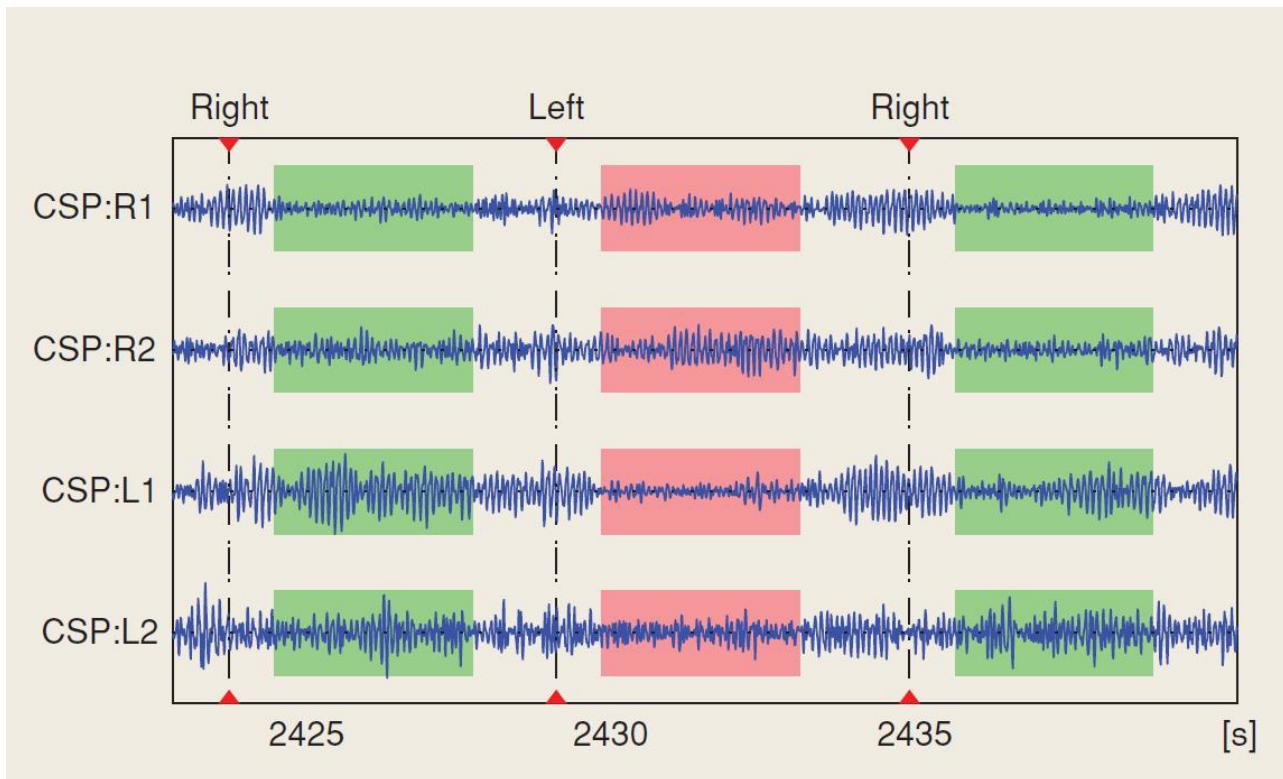


Figure 5.6: Effect of spatial CSP filtering in motor imagery task.

In 5.6 CSP analysis was performed to obtain four spatial filters that discriminate left from right hand motor imagery. The graph shows continuous band-pass filtered EEG after applying the CSP filters. The resulting signals in filters CSP:L1 and CSP:L2 have larger variance during right hand imagery (segments shaded in green) while signals in filters CSP:R1 and CSP:R2 have larger variance during left hand imagery (segment shaded red). From the above figure, we can intuitively say that CSP performs a transformation of the signals that accentuates the differences that are to be found between the two conditions, in our case, between crashes and evasions.

From a technical standpoint, the first step of the CSP analysis is the calculation of the covariance matrix in the two conditions. The spatial filters are computed by the simultaneous diagonalization of these two covariance matrices. Let  $\Sigma^{(+)}$  be the covariance matrix of the positive (crash) cases and  $\Sigma^{(-)}$  be the covariance matrix of the negative (evasion) cases.

$$\begin{aligned} W^T \Sigma^{(+)} W &= \Lambda^{(+)} \\ W^T \Sigma^{(-)} W &= \Lambda^{(-)} \\ (\Lambda^{(c)} \text{ is diagonal}) \end{aligned} \quad (5.1)$$

In order to simultaneously diagonalize them, we can solve the following generalized eigenvalue problem.

$$\Sigma^{(+)}w = \lambda \Sigma^{(-)}w. \quad (5.2)$$

Where  $w$  is the generalized eigenvectors and  $\lambda$  the corresponding diagonal element of the diagonalized covariance matrices. Note that a large value of  $\lambda$  close to one indicates that the corresponding spatial filter  $w$  yields high variance in the positive (negative) condition and low variance in the negative (positive) condition, respectively. This is the way the filters are calculated in order to increase the variance in one condition while decreasing it in the other. The transformed CSP space signals are given by the transformation:

$$x_{CSP}(t) = W^T x(t) \quad (5.3)$$

Where  $W$  is the matrix containing the filters  $w$  calculated in the above step,  $x(t)$  the original signals and  $x_{CSP}(t)$  the signals transformed into CSP space.

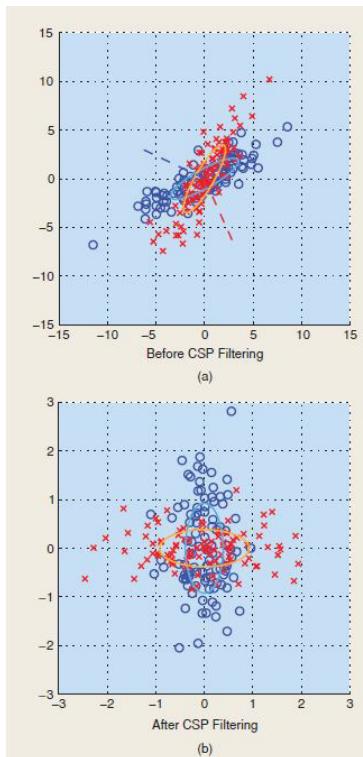


Figure 5.7: A toy example of CSP filtering in 2-D showcasing the discriminative power of spatial filtering.

If we were to intuitively understand the process described above, we could look at 5.7. Two sets of samples marked by red crosses and blue circles are drawn from two Gaussian distributions. In (a), the distribution of samples before filtering is shown. Two ellipses show the estimated covariances and dashed lines show the direction of CSP projections In (b), the distribution of samples after the filtering is shown. Note that both classes are uncorrelated at the same time; the horizontal (vertical) axis gives the largest variance in the red (blue) class and the smallest in the blue (red) class, respectively.

From the above, we can understand that the CSP method is based on the calculated spatial filters and the corresponding patterns. These byproducts of the process carry a useful amount of information that can be used in order to further our neurophysiological understanding of the participant's brain activity. Specifically, the spatial filters can not provide sufficient information to come to a neurophysiological explanation of the brain activity, as they are just the weights that are used to transform the initial data to the CSP space. On the other hand, the corresponding spatial patterns can be used to

check plausibility and to investigate neurophysiological properties. Based on the previous example of 5.6, one filter and the corresponding pattern in an analysis of a motor imagery task can be seen in 5.8

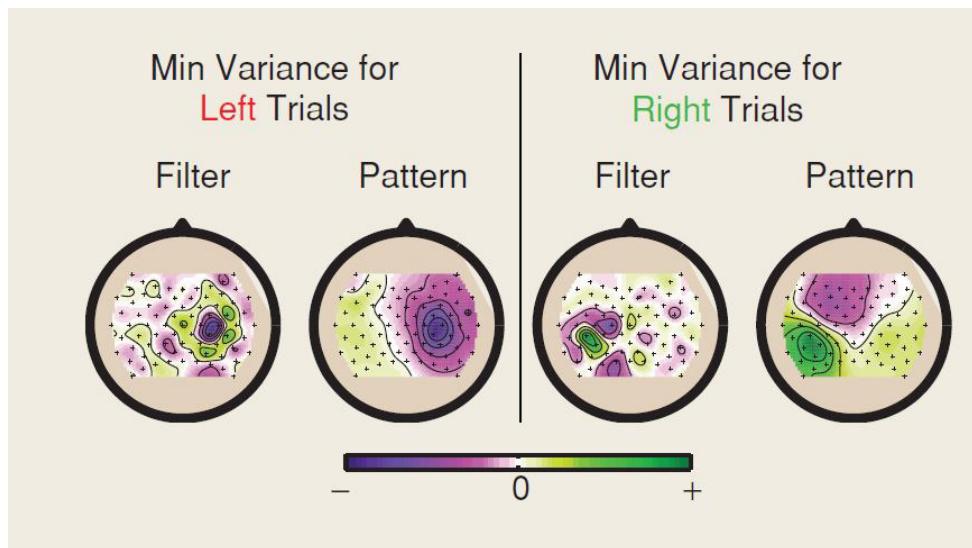


Figure 5.8: Example of visualization of CSP analysis on a motor imagery classification task

The patterns illustrate how the presumed sources project to the scalp. They can be used to verify neurophysiological plausibility. The filters are used to project the original signals. Here they resemble the patterns but their intricate weighting is essential to obtain signals that are optimally discriminative with respect to variance. In this case, we can see the difference between the patterns corresponding to imagining moving the left and the right arm. We can see in the left arm movement patterns that there is high activity in both hemispheres of the brain, but of opposite signs. Respectively, a similar picture can be seen in the patterns corresponding to imagining the right arm movement, but with approximately opposite polarity of the activity. In both cases the peak of the activity can be placed at the center of the brain, in the motor and sensory cortex. The exact place of the peaks also corresponds to the mapping of the functions seen in 5.9.

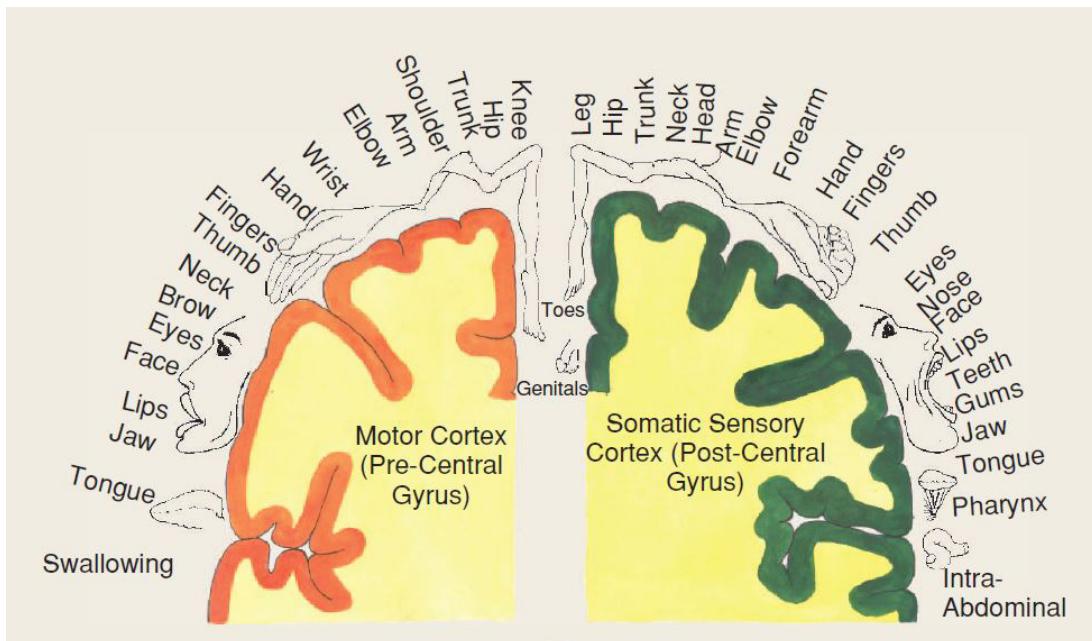


Figure 5.9: Geometric mapping between body parts and motor/somatosensory cortex

We implemented CSP filtering using the respective method provided by MNE. This method re-

quires the number of components, meaning the number of filters and patterns derived from the process. By default, the method does not regularize the covariance matrix and it calculates them using the empirical computation. We tried most of the regularization techniques provided by the library but none of them seemed to make a great impact on the accuracy of our model. Also, the default choice is the fastest so we decided to keep it as that. The output of this method can be chosen to be either the average power of each spatial filter or the whole transformed signal. In case we want the average power to get returned, a useful parameter of the method is the logarithmic transformation of the features, used as a standardization technique. Last but not least, the method can display the filters and the patterns calculated. This is a very useful piece of information as it can provide evidence for a neurological explanation of the classification efficiency.

## 5.8 Source Power Comodulation

Similar to the CSP method described above, the Source Power Comodulation [88], method or SPoC for short, is a supervised source separation framework. This framework makes use of the continuous target variable in the decomposition process in order to give preference to components whose power comodulates with the target variable. SPoC is a similar technique to other component analysis techniques like ICA described in a previous subsection. The advantage of this method is that the preference is given to components that better describe the target variable while techniques like ICA prioritize maximal independence between the components. In this way, SPoC is preferable when trying to predict a continuous value, which is exactly what we want to accomplish in the regression analysis.

The complete pipeline of the SPoC algorithm can be seen in 5.10 in contrast to other widely used methods of source separation techniques.

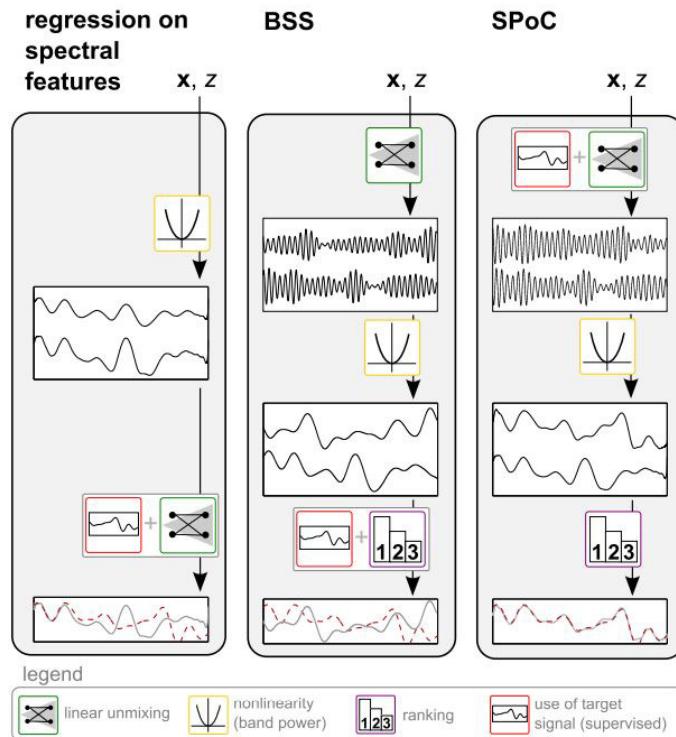


Figure 5.10: Illustration of different approaches to relating spectral power to a target variable

The SPoC pipeline starts with linearly unmixing the brain activity signals using the target variable as a guide. Next the band power is calculated and lastly the resulting signals are ranked based on their

fit to the target variable.

The novelty of this algorithm resides in the first step of the pipeline. In order to unmix the components that fit the target variable the best, a weight matrix  $W$  is calculated that contains the spatial filters. These spatial filters are computed the same way as in the CSP algorithm. A notable difference is the covariance matrix used for the extraction of the eigenvectors. In the CSP algorithm we had two different conditions and performed a simultaneous diagonalization task on both the covariance matrices of the two conditions. In the SPoC case, there are not two conditions that guide the unmixing process, but a continuous value.

This method employs an epoch based technique to substitute the conditionally computed covariance matrices with a covariance matrix that calculates the difference between the target value and the predicted one. Let  $C(e)$  be the covariance matrix of our initial signal  $x(t)$  on the  $e$ th epoch. The quantity of interest we will compute is defined as follows:

$$z'(e) = w^T(C(e) - C)w. \quad (5.4)$$

Where  $C$  is the covariance matrix computed from the average of all epochs,  $w$  the matrix containing the filters and  $z'$  the approximation of the target variable  $z$ . In fact, equation 5.4 is very similar to equations 5.1. In the place of  $\Sigma^{(c)}$  describing the covariance matrix of each condition, we have  $C(e) - C$ . We then define:

$$C_z = \langle C(e)z(e) \rangle \quad (5.5)$$

The chevrons denote that the matrix  $C_z$  is calculated by averaging the quantity inside the symbols. Finally, the covariance between  $z'$  and  $z$  is given by

$$\text{Cov}[z'(e), z(e)] = w^T C_z w \quad (5.6)$$

Just like in the CSP method, SPoC provides useful information for providing neurophysiological explanations to the correlation between brain activity and the target variable by visualizing the spatial patterns computed. We note that the spatial patterns that correspond to a specific filter are described as:

$$a \propto \sum_t x(t)(w^T x(t)) \quad (5.7)$$

Where  $a$  are the spatial patterns corresponding to the spatial filters  $w$ .

Likewise to the CSP algorithm, the MNE library has an implementation of the SPoC that follows roughly the same input rules as the CSP function. The only difference being that instead of a class label, the SPoC takes the continuous target value as input.

## 5.9 Machine Learning Fundamentals

In this section, some of the most important notions of Machine Learning will be explained in order to better understand the reasoning behind our model choices with respect to the problem at hand.

Starting with the notion of machine learning itself, it is the field of study that gives computers the capability to learn without being explicitly programmed. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. A lot of these algorithms are based on complex statistical methods. However, the most important aspect of machine learning is the use of mathematical optimization methods that are responsible for the learning part of these models.

### 5.9.1 Types of Machine Learning Problems

We mentioned above that machine learning algorithms make predictions and decisions given a set of data. While this seems concrete enough, depending on the nature of data and the output we want from the model, the nature of the models changes drastically.

The first distinction can be made between the way the output is supposed to be inferred. There are two main types of learning, supervised and unsupervised. Supervised learning is a machine learning approach that's defined by its use of labeled datasets. These datasets are designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately. Using labeled inputs and outputs, the model can measure its accuracy and learn over time. In this case, we know beforehand, in which category, each sample belongs. Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled data sets. These algorithms discover hidden patterns in data without the need for human intervention. In this case, the target is unknown beforehand and the algorithm tries to extract it in a sensible way.

The next distinction that can be made is based on the type of the problem the algorithm tries to solve. The four most common types of problems are Classification, Regression, Clustering and Anomaly Detection, although many others exist. Classification tasks are simply related to predicting a category of data (discrete variables). One of the most common examples is predicting whether or not an email is spam or ham. Regression tasks mainly deal with the estimation of numerical values (continuous variables). Some of the examples include estimation of housing price, product price, stock price etc. Clustering is a data mining technique for grouping unlabeled data based on their similarities or differences. For example, K-means clustering algorithms assign similar data points into groups, where the K value represents the size of the grouping and granularity. This technique is helpful for market segmentation, image compression, etc. Anomaly detection is the process of identifying unusual patterns in data that do not conform to expected behavior. It is often used in a wide range of applications, such as detecting fraudulent activity in financial data, detecting malicious behavior in network traffic data, and identifying equipment malfunctions in sensor data.

Specifically, for our problem, we want to predict whether or not a participant is going to crash or successfully evade an obstacle based on his or her EEG recording before the event. The input data is the EEG recording and the target is the crash or evasion distinction. Based on these, we can say that this is a Supervised Classification Task. The other problem we will try to solve, is predicting the reaction time of the drivers based on the EEG recordings. Again, the input data are the data points of the EEG recording and the target is the continuous value of the reaction time of each event. Hence, the second task is a Supervised Regression Task.

### 5.9.2 Overfitting and Underfitting

During the training phase, the model can either adapt adequately to the data and give good results, or it might fail to do so. This can happen due to many reasons but all of these can be organized into two categories, namely overfitting the model and underfitting the model.

Underfitting a model happens when the model could not grasp the underlying mathematical rules behind the data, resulting in poor guesses. It is like trying to predict the outcome of a game without knowing the rules. In machine learning terms, Underfitting refers to a model that can neither model the training data nor generalize to new data. An underfit machine learning model is not a suitable model and will be obvious as it will have poor performance on the training data. Underfitting is often not discussed as it is easy to detect given a good performance metric. The remedy is to move on and try alternate machine learning algorithms, expand the number of samples and so on. Nevertheless, it does provide a good contrast to the problem of overfitting.

While an overfit model might be able to grasp the underlying rules of a problem, it might get too specific on the given data, meaning that it fails to generalize given new, unseen data. A real life example of this is a student trying to study for an exam by keeping in memory every step of

solving an exercise. The student might be able to solve that particular exercise convincingly, but a slight alteration of the parameters of the problem will rend the student unable to solve it. Hence, we can say that overfitting refers to a model that models the training data too well. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize. Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns.

### 5.9.3 Training and Validation

In order to be able to tell when a model is overfit or underfit, we employ a technique called train-test split. The intuition behind this is that we reserve a small portion of data just to validate that the model can actually generalize given unknown data. When splitting the data one must keep in mind that there are two conditions that ensure a good train-test split. The first is for the two sets to be large enough to yield statistically meaningful results. The second is for the two sets to be representative of the data set as a whole. In other words we should not pick a test set with different characteristics than the training set. A common ratio between the training and testing sets is 80% to 20% respectively.

## 5.10 Classification Task

As we discussed earlier, our goal is to classify the end result of a participant's reaction to a takeover request event. Namely, the two outcomes of the event can be either the evasion of the obstacle or the crash of the vehicle either into the obstacle or at the sides of the road. We considered two different approaches for the classification task that differ in the type of the inputs we used.

The first approach was to use the average power of each signal in csp space after the transformation of our original data described in the sections above.

### 5.10.1 Decision Trees

Decision Trees is one of the most intuitive algorithms for decision making and classification. The induction of a decision tree is described in [89]. The idea behind creating a decision tree is that, given a group of observations, each with a set of features and described by a single label, what feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible? Is it possible to create a series of splits that can split the observations into groups that have elements that belong only to one label? This last proposition would create a perfect classification of the given observations. A simplistic example of a decision tree can be seen in 5.11

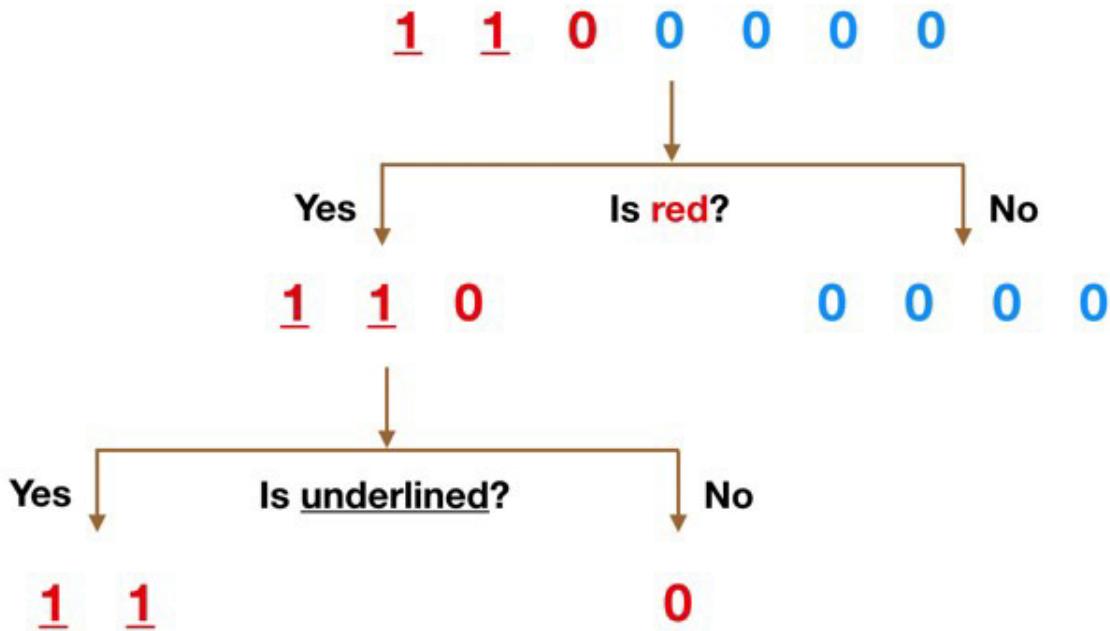


Figure 5.11: Example of a decision tree made to differentiate a set of numbers based on their color and underlining

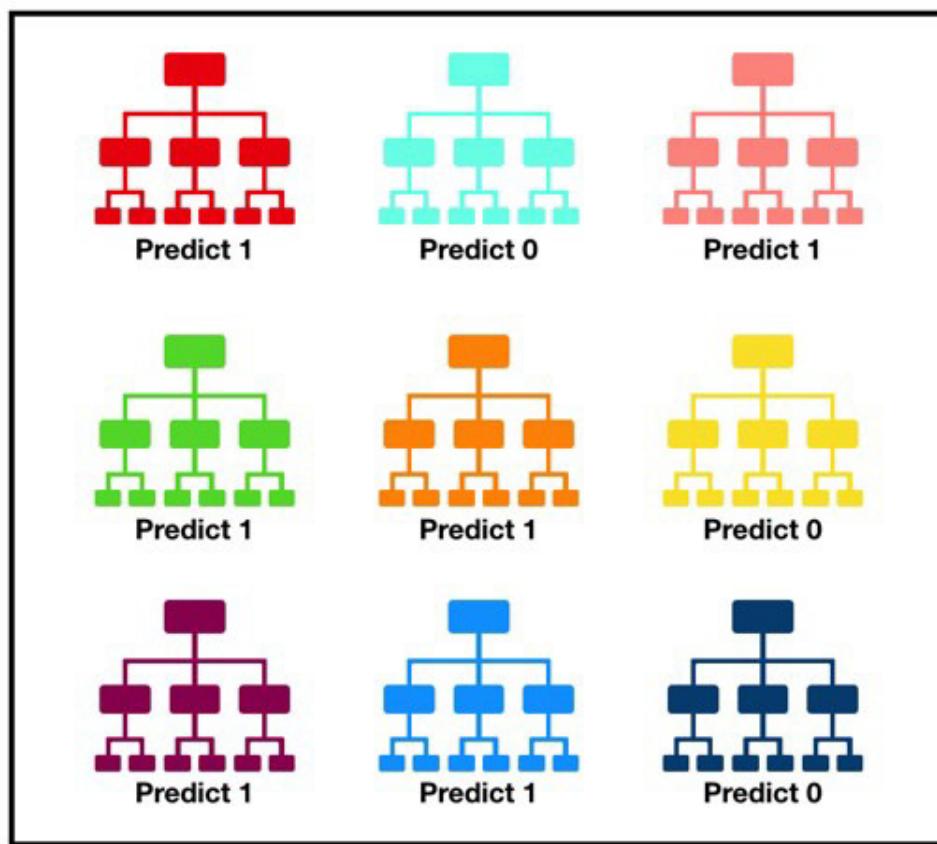
In a more formal description of the process, this machine learning algorithm considers, at every step, the information gain of splitting the observation group based on each provided feature. The information gain of the split can be measured by different metrics, such as Gini impurity and entropy. We chose entropy for our criterion. Repeating this process until the best total information gain is achieved, gives the tree that fits our dataset the best.

In practice, decision trees calculate simple rules that are used to split the observation group into the most homogeneous sub-groups based on label. Given a new observation, one must just follow these rules, choosing the corresponding branch, until one reaches a leaf of the tree, meaning an ending node where no other branching is done. The label that is represented more in that particular leaf, is chosen to be the prediction of our model.

While this approach is simple to understand, fast to build and fast to predict the label of a new observation it has some notable disadvantages. Specifically, Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Mechanisms such as pruning, setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem, although, in that way the optimization of the decision tree becomes a much harder problem. Also, Predictions of decision trees are neither smooth nor continuous, but piecewise constant approximations. Therefore, they are not good at extrapolation. Lastly, decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.

## 5.10.2 Random Forest Classifier

A random forest classifier [90] consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. This technique ameliorates the stability problems that a single decision tree presents, as discussed in the above subsection. An example of a decision tree ensemble is illustrated in 5.12.



Tally: Six 1s and Three 0s

**Prediction: 1**

Figure 5.12: Example of a decision tree ensemble. Each individual tree votes for the label of an observation and the majority determines the prediction of the model

The strength of this approach lies in the fact that the individual decision trees are uncorrelated. This means that an error made by a single tree can be compensated by the correct prediction from the majority of the remaining trees, as long as they are uncorrelated. In fact, being uncorrelated means that they do not make errors in the same direction. The algorithm ensures that the trees are uncorrelated by exploiting one of the main disadvantages of the single decision tree described above, namely its instability.

The different trees in the random forest, are created by randomly sampling the training dataset, with replacement. That process is called bagging and ensures that each decision tree is trained with  $N$  samples,  $N$  being the number of total samples provided in the training dataset, but with replacements, meaning that certain observations are repeated and consequently, a greater emphasis is given at the features that discern these observations better. Another parameter that ensures that trees are uncorrelated, is the random choice of features that are provided to each tree.

### 5.10.3 Logistic Regression

Logistic Regression is another very basic technique used for classification of observations based on a set of features. In contrast to the aforementioned methods, logistic regression does not create a set of rules to discern between different classes, but tries to uncover a mathematical rule that underlies them. Logistic regression is particularly beneficial when the features are continuous values, like in our case, either when we use the average power of each channel or when we use the whole csp space transformed signals.

One important difference between the tree based models and logistic regression is that this model provides a probability that a hypothesis is correct. Namely, if we consider as our null hypothesis that an observation belongs to class "crash", the output of the model gives the probability that the observation actually belongs to that class. An output greater than 50% would mean that the null hypothesis is valid, while an output less than 50% would suggest the opposite.

The mathematical foundation upon which the logistic regression technique is based is the following. Initially, we define a weighted linear combination of the features as follows:

$$Z = WX + b \quad (5.8)$$

Where  $X$  is a vector containing all the features of an observation,  $W$  a vector containing weights and  $b$  a bias. This quantity is then inputed in a non-linear unit, in particular, in a sigmoid function, the formula of which is the following:

$$\text{Sig}(x) = \frac{1}{1 + e^{-x}} \quad (5.9)$$

The sigmoid function is chosen because it is bound between 0 and 1, just like a probability is. A graph of the sigmoid function with the corresponding acceptance or rejection of the null hypothesis can be seen in 5.13.

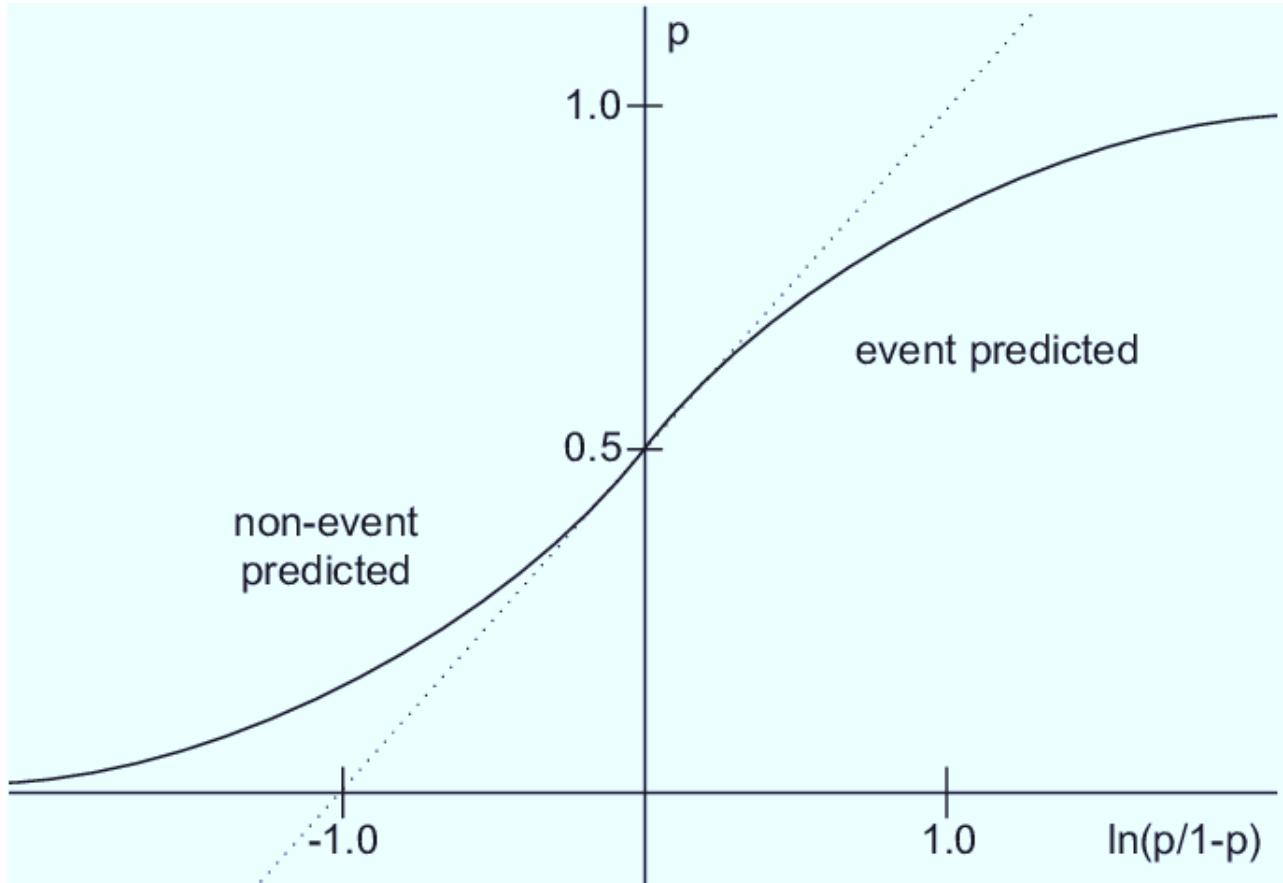


Figure 5.13: Graph depicting the sigmoid function

Combining equations 5.10 and 5.9, we get the prediction of our model as:

$$\hat{Y} = \frac{1}{1 + e^{-Z}} \quad (5.10)$$

Initially, since we do not know the relation between the features and the labels, the weights are randomly selected. This means that at first, our model predicts at random. In order to train the model

to give better predictions, the definition of a cost function that we will try to minimize is paramount. In logistic regression used in a two class classification task, the Log Loss is chosen as the cost function. The log loss cost function is defined as:

$$J(\hat{Y}, Y) = -\frac{1}{N} \sum_{i=1}^N [Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i)] \quad (5.11)$$

Where  $\hat{Y}_i$  is the predicted value of our model, ranging from 0 to 1,  $Y_i$  being the real class label of the observation and takes either 0 or 1 as a value. Let us consider the following cases.

- If an observation belongs to class 0 and our model predicts a probability near 0, then the  $Y_i \log(\hat{Y}_i)$  factor is zero because of  $Y_i$  and  $(1 - Y_i) \log(1 - \hat{Y}_i)$  factor is near zero because of  $\log(1 - \hat{Y}_i)$  resulting in a low cost value.
- If an observation belongs to class 1 and our model predicts a probability near 1, then the  $Y_i \log(\hat{Y}_i)$  factor is near zero because of  $\log(\hat{Y}_i)$  and  $(1 - Y_i) \log(1 - \hat{Y}_i)$  factor is zero because of  $(1 - Y_i)$  resulting in a low cost value.
- If an observation belongs to class 0 and our model predicts a probability near 1, then the  $Y_i \log(\hat{Y}_i)$  factor is zero because of  $Y_i$  but factor  $(1 - Y_i) \log(1 - \hat{Y}_i)$  is large because of  $\log(1 - \hat{Y}_i)$  resulting in a high cost value.
- If an observation belongs to class 1 and our model predicts a probability near 0, then the  $Y_i \log(\hat{Y}_i)$  factor is large because of  $\log(\hat{Y}_i)$  and  $(1 - Y_i) \log(1 - \hat{Y}_i)$  factor is zero because of  $(1 - Y_i)$  resulting in a high cost value.

A more comprehensive diagram can be seen in 5.14

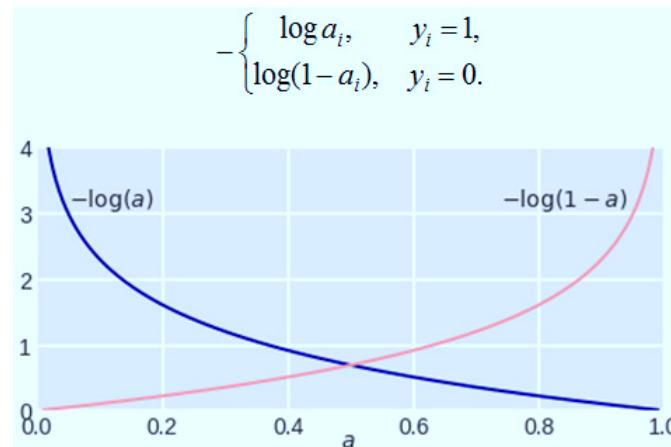


Figure 5.14: Diagram showing the value of the log loss cost function depending on the values of the label ( $y_i$ ) and the model's prediction ( $a_i$ )

After defining the cost function, the gradient descent algorithm is used to update the weights defined in equation 5.10. The gradient descent algorithm's basic idea is the following. In an iterative process, like the training of a machine learning model, the change of the model's parameters happens by subtracting the derivative of the cost function with respect to the parameter, by the parameter, multiplied by a factor called learning rate.

$$W_{ij} = W_{ij} - a \cdot \frac{\partial J}{\partial W_{ij}} \quad (5.12)$$

### 5.10.4 C-Support Vector Classification

Just like all the previous models the Support Vector Machines are models that aim to distinguish some given data points that belong to different classes. The algorithm achieves this by finding a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes in lower dimensions are easy to imagine. A hyperplane in two dimensions is a line while a hyperplane in three dimensions is a 2D plane and a hyperplane in 4 dimensions is a 3D space. In higher dimensions, hyperplanes are impossible to imagine, but the intuition we get from the low dimensions still applies. Specifically, in the 2 dimensional example, every data point above the line will be assigned to one class, while every data point below the line will be assigned to the other. This leads us to conclude that hyperplanes are just ways to split the space of the problem in two distinct areas, however many dimensions there might be.

Support vectors, seen in figure 5.15 are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane, also giving the name to the model. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

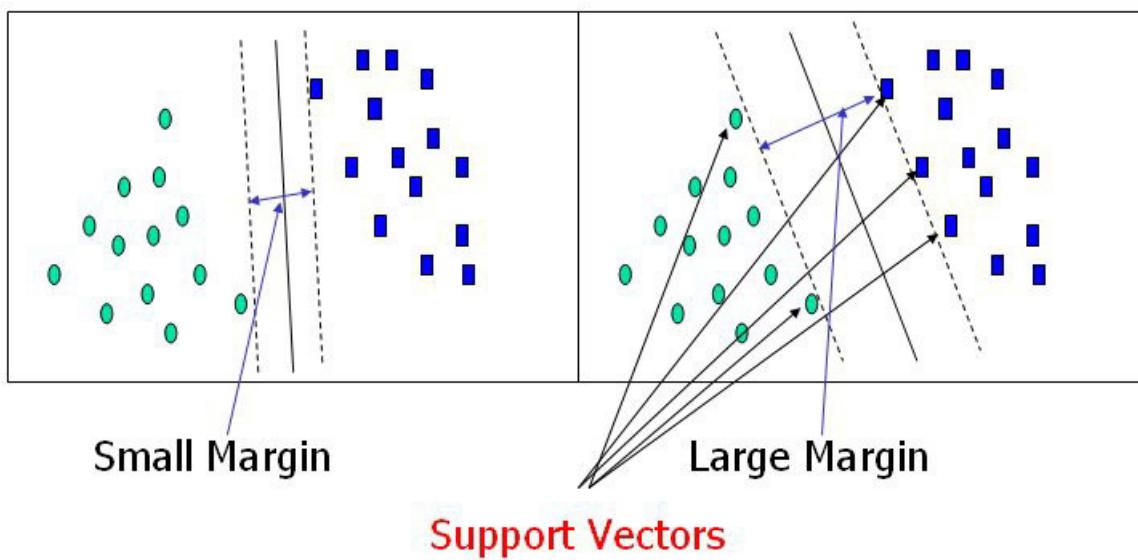


Figure 5.15: Image showing the Support Vectors used to solidify the hyperplane's position and orientation.

A very important aspect of the SVM model, giving it high versatility in most machine learning problems, is the kernel trick. The kernel trick, is a mathematical trick that enables a linear model, like the SVM, to classify non-linear data points by transforming them into a higher dimension in which they are linear. In order for a function to be considered a kernel, it must satisfy Mercer's constraints:

$$K(a, b) = \Phi^T(a)\Phi(b) \quad (5.13)$$

With  $\Phi$  being a kernel and  $K(a,b)$  a function.

Some very common kernels used in SVMs are:

- Gaussian Radial Basis Function:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (5.14)$$

- Polynomial Kernel Function:

$$K(x_i, x_j) = (x_i \cdot x_j + a)^b \quad (5.15)$$

- Linear Kernel:

$$K(x_i, x_j) = (x_i \cdot x_j) \quad (5.16)$$

We are going to use all the above kernels to induce some non linearity to our models since we can not know beforehand, if the data relation between the EEG recording and the crash/evasion cases will be linearly separable or non-linearly separable.

## 5.11 Classification Metrics

A classification task is defined as an attempt to create a model that given some input data can output a number signifying a class that the particular input is a part of. In order to evaluate this process a lot of different approaches are employed that provide different insight regarding the efficiency of our model.

### 5.11.1 Precision

The Precision metric is defined as the the number of true positives over the number of true positives plus the number of false positives.

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad (5.17)$$

Intuitively, we can consider the precision score to be an indication of how good our model is at finding the positive instances of a particular class. In fact, in our case we would prefer a high precision when trying to find crashes as correctly predicting when a crash will happen is more important from a safety standpoint than a general accuracy score.

### 5.11.2 Recall

Recall is defined as the number of true positives over the number of true positives plus the number of false negatives.

$$\text{Recall} = \frac{T_p}{T_p + F_n} \quad (5.18)$$

This metric focuses on how many of the elements classified in a specific class do, in actuality, belong to that class. While this also is a very important metric of the overall quality of our model, from a safety standpoint it is more important to predict correctly all the times a crash is going to happen, than predicting correctly the ones that a crash is not going to happen.

### 5.11.3 F1-score

The F1 score is the harmonic mean of the Precision and Recall metrics defined in equations 5.17 and 5.20. It is given by the following formula:

$$F1 = 2 \frac{Precision \cdot Recall}{Precision + Recall} \quad (5.19)$$

This metric takes values between 0 and 1 with 0 being a bad score and 1 being a good score. It is a general metric of the accuracy that encapsulates the precision and recall metrics.

### 5.11.4 Accuracy

The accuracy score is the most basic of the evaluation metrics. It is defined as the number of samples classified correctly over the total number of samples used for the validation of our model. The first three metrics that are going to be discussed, namely the precision, recall and f1 score, are calculated for each label separately.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} = \frac{T_p + T_n}{N} \quad (5.20)$$

Where N is the total number of inputs tested in the validation of our model.

### 5.11.5 Macro average

The macro average metric is practically the unweighted average of the previously discussed metrics. We calculated the macro average score for the precision, recall and accuracy.

### 5.11.6 Weighted average

The weighted average metric is the average of the previously discussed metrics but the number of samples of each label is taken into consideration.

## 5.12 Regression Task

In the regression task we tried to estimate the reaction time of the drivers using their brain's activity. The significance of this task is that a takeover request mechanism can estimate whether or not a driver will take the control of the vehicle in time to avoid the oncoming obstacles. In order to achieve this we used the SPoC algorithm to extract features from the time window 2 seconds before the onset of the notification. We used all 32 components of the SPoC algorithm and inputted them into the following three machine learning models.

### 5.12.1 Support Vector Regression

Similar to the Classification problem, the support vector idea can be applied to regression. The goal in the classification problem was to fit a hyperplane into the data in a way that there is maximum margin between what we called support vectors, and the fitted hyperplane. Doing that, assured us that the two classes could be devised by the maximum margin possible, meaning that the hyperplane can split the data space with equal terms to the two classes. Lets see how we can apply the same logic to a completely different problem, that of estimating the value of a continuous target variable.

SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data. The objective function of SVR

is to minimize the coefficients — more specifically, the l2-norm of the coefficient vector — not the squared error. The error term is instead handled in the constraints, where we set the absolute error less than or equal to a specified margin, called the maximum error,  $\epsilon$  (epsilon). We can tune epsilon to gain the desired accuracy of our model. Our new objective function and constraints are as follows:

$$\text{MIN} \frac{1}{2} \|x\|^2 \quad (5.21)$$

while also

$$|y_i - w_i x_i| \leq \epsilon \quad (5.22)$$

With  $w$  being the parameter vector of the hyperplane,  $y$  the target variable to be predicted,  $x$  the input data and  $\epsilon$  the maximum margin of error.

It is possible that some of the points might fall outside the specified margin of error. In that case we introduce a new concept of slack variables denoted by  $\xi$ . The concept of slack variables is simple: for any value that falls outside of  $\epsilon$ , we can denote its deviation from the margin as  $\xi$ . This new  $\xi$  is regulated by a factor  $C$  that we call regularization. The new objective function is written like this:

$$\text{MIN} \frac{1}{2} \|x\|^2 + C \sum_{i=1}^n |\xi_i| \quad (5.23)$$

while also

$$|y_i - w_i x_i| \leq \epsilon + |\xi_i| \quad (5.24)$$

With  $\xi_i$  being the distance of the  $i$ th data point from the error margin.

In the case of the SVC, we used the kernel trick to transform the data into a hyperspace in which they are linearly separable. In the regression task this is also possible. The data might follow a non-linear function like a polynomial or a logarithmic function. Using the kernels we saw in equations 5.14, 5.15, 5.16. We are able to use these in order to check for possible non linearities in our data.

## 5.12.2 Random Forest Regression

Random Forest Regression follows the same intuition as the Random Forest Classifier. The basic block of this model is the Decision Tree, which works almost exactly as the Decision Trees described in the previous section. The only difference is they try to estimate continuous variables. Each value of the data points is considered to be a single class in this case.

The difference of this model is that when we employ the ensemble process to get the end results, we average the values given by each decision tree. In this way we get continuous values that lie in between the values of the training dataset. This has one obvious problem though. Outliers, meaning data samples that have extreme values and are exceptions to the underlying mathematical rules, can not be estimated correctly by the model. Especially, if they do not show up in the training dataset but show up in the testing dataset.

We consider this model to be very good at grasping the underlying rules, especially when it comes to complex non-linear relationships. For that reason we used it to predict the reaction times of the participants.

## 5.12.3 Ridge Regression

The ridge regression model shares a lot in common with the Logistic Regression model we analyzed at the Classification Problem section. In particular, in the Logistic Regression model, we inputted a linear equation in the sigmoid function to create a decision boundary. If we just kept the linear equation of 5.10 we would have a model called Linear Regression. This model is the most basic regression model

which tries to fit a line through the data using Mean Squared Error as a cost function, which is going to be discussed later. The Ridge Regression model is a variation of the Linear Regression one.

The mathematical foundation of this model is the following:

$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta \in R} (\|y - XB\|_2^2 + C\|B\|_2^2) \quad (5.25)$$

In the above equation,  $\hat{\beta}^{ridge}$  represents the new weights of the ridge regression model after each recursion. The vector B corresponds to the previous weights, X is the input matrix and y the output value. C is the regularization parameter and the  $\|B\|_2^2$  symbolism represents the L2 norm of the vector B. The L2 norm is given as:

$$\|B\|_2^2 = \sqrt{\beta_0^2 + \beta_1^2 + \beta_2^2 + \dots + \beta_n^2} \quad (5.26)$$

The calculation of the argmin component is done with gradient descent, just like in the logistic regression case. This model is a simple, yet robust take on the fitting of linear equations in data. We consider this option in the case that the data do not have a non linear relationship.

## 5.13 Regression Metrics

Similarly to the classification task, we use certain accuracy metrics in order to evaluate the regression task. In particular, a regression task is defined as the creation of a model that tries to predict a continuous target value by uncovering the underlying mathematical rule that governs the behaviour of a specific phenomenon that is used as input. In contrast to the classification task, we can not simply check if the predicted values coincide with the real ones as it is impossible to get exactly the same value. As this is the case, the need rises for more sophisticated metrics that try to evaluate the deviation of the predicted values from the real ones.

### 5.13.1 Mean square error

The Mean Square Error or MSE is one of the most common evaluation metrics for regression tasks as it is simple to calculate and it usually plays the role of the cost function in many regression models. MSE is calculated as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 \quad (5.27)$$

Where N is the total number of inputs that we asked our model to predict, i is the index running through the validation set,  $Y_i$  is the real value of the target variable and  $\hat{Y}_i$  is the value of the target variable predicted by our model. We calculate the difference between the real and the predicted value of the target variable and square it in order to get a positive number. This metric is considered a distance function. We calculate the average of this quantity in order to see the digression of our predictions from the real values. It is evident that the smaller the MSE the better our model gets.

### 5.13.2 Mean average error

The Mean Average Error or MAE is a similar metric to the MSE score. The only difference between the two is that instead of squaring the difference between the predicted and real value, we take the absolute of that quantity. Formula 5.27 is altered as follows in order to describe the MAE:

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (5.28)$$

This metric can be easier to understand. In our problem, for example, we could say that the predicted reaction times digress, on average, at a factor of x ms. This can not be said for the MSE score as we square this value, which makes it a bit harder to understand intuitively. On the other hand, the MSE score is very useful because it is the second moment of the error, and includes both the variance of the estimator and its bias.

### 5.13.3 R squared

The R squared metric, or coefficient of determination, is a complex statistic that is defined as the proportion of the variation in the dependent variable that is predictable from the independent variable. The R squared score can be calculated as follows:

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} \quad (5.29)$$

The unexplained variance can be calculated by summing all the squared errors (same as MSE error but without the averaging). To calculate the total variance, you would subtract the average actual value from each of the actual values, square the results and sum them. From there, divide the first sum of errors (explained variance) by the second sum (total variance), subtract the result from one, and you have the R-squared. This metric can take values from  $-\infty$  to 1, 1 being the best score possible, meaning that all the variance of the dependent variable is predictable by our model. The negative values of the R squared score are attributed to the latent comparison made between the predictions of our model and the predictions made by a straight horizontal line, that many libraries, including sklearn, implement.

While a low R squared score does not necessarily mean that a model is bad, it can mean that the underlying rules that govern the phenomenon at hand, can not be explained by a linear model.

# 6. Results

## 6.1 Statistical Analysis

Firstly, a general statistical analysis of the data was made in order to get basic insight into the behavioural patterns of the participants. Later, these results will be used to evaluate the effectiveness of the experiment and how accurately we evoked the intended responses from the participants.

### 6.1.1 Crashes versus Evasions analysis

At first, a holistic review of the results will be presented. Figure 6.1 depicts the total number of events recorded during the experiment. In the x axis, we see the 9 different categories of events recorded based on table 4.1. Events 1 to 6 represent the creation or false alarm of obstacles with the respective notification modalities. Event 7 represents the wheel grasping moments and events 8 and 9 represent the crashes with the obstacle and the resetting of the vehicle respectively. On the y axis, the total number of events recorded is depicted.

We can see that events 1 to 3, being the real obstacle creation events, amount up to 200 events each, while the false alarm events have been recorded with one fourth of the frequency of the aforementioned events. Generally, the number of samples is even between the different modalities, which ensures us that the different modalities were evenly sampled. Event 7 is the most recorded event of them all. It represents every time the participants grasped the wheel. This means that the number of events coded with 7, can approximate the total number of obstacles created, apart from some very few cases during which the participants did not grasp the wheel. The total amount of crashes can also be estimated using events 8 and 9. The number of total crashes is a little above the recorded resets of the vehicle. As a rule, the reset button was pressed whenever a crash would be reported either with the obstacle, or the sides of the road, while 8 would only be reported when a crash happened with the obstacle.

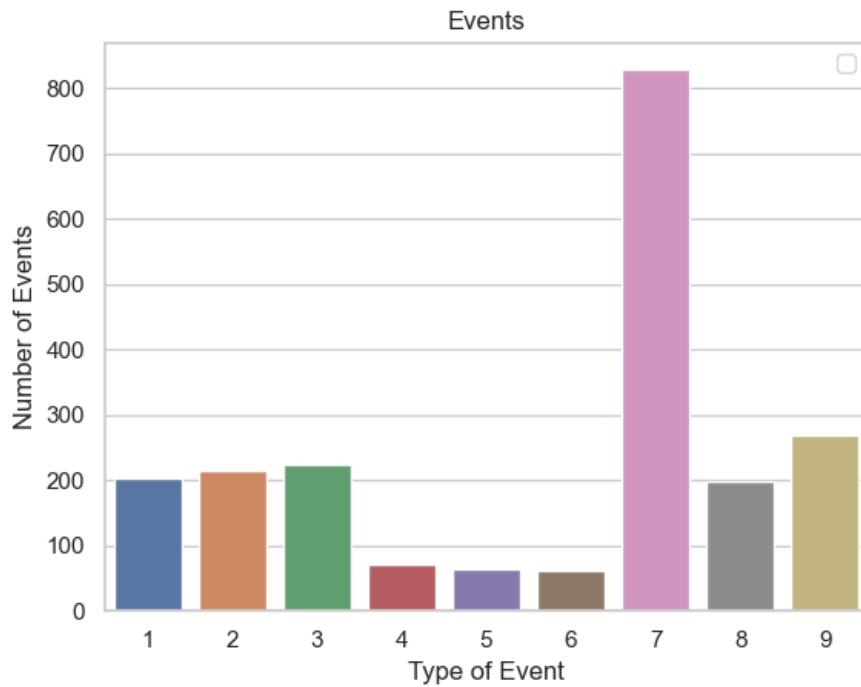


Figure 6.1: Aggregate results of each event type

The following segment is dedicated at comparing the crashes and the evasions in total and in the different types of events of the experiment.

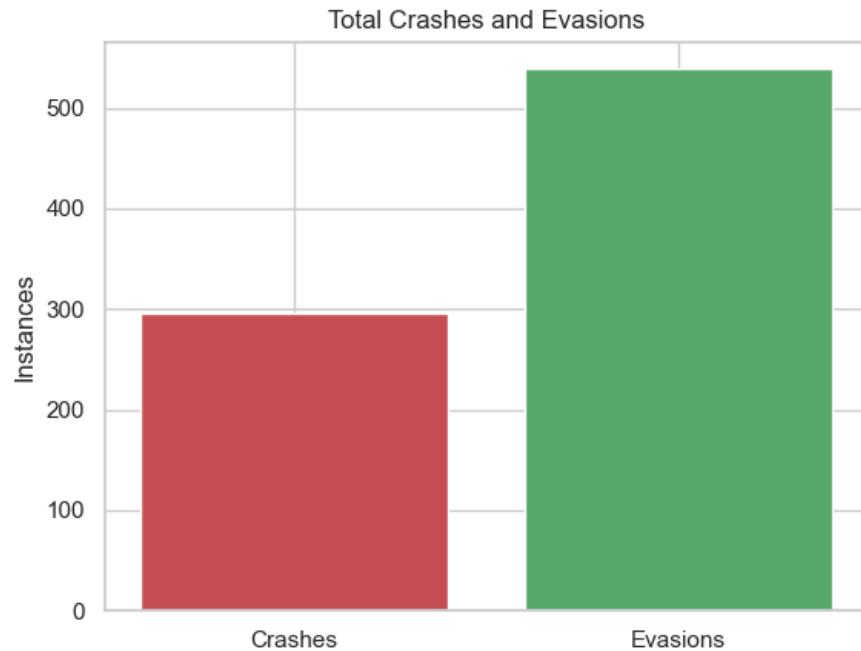


Figure 6.2: Chart depicting the number of total crashes (in red) versus total evasions (in green)

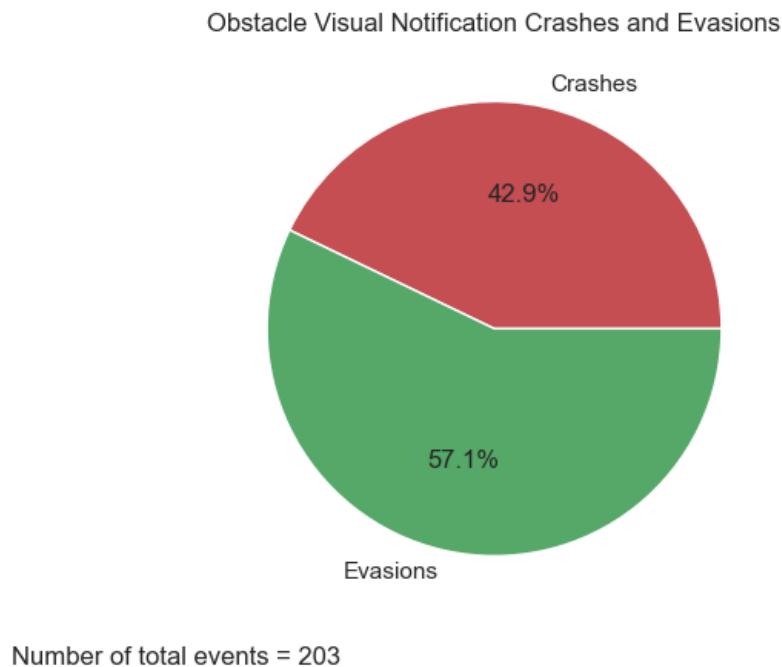


Figure 6.3: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a visual notification to the subject.

In the chart above 6.2, the total amount of crashes in comparison to the total amount of evasions is depicted. The bar plot consists of two bars in red and green color that represent the crashes and evasions respectively. In the y axis, the total amount of instances of each case is presented, with the crashes being almost 300 in number, while the evasions exceed the mark of 500 events. The accurate numbers for each case is 296 for the crashes and 539 for the evasions. We can see that the total amount of evasions exceeds by far that of the crashes. This means that our data is unbalanced and we should address this problem.

The pie chart depicted above 6.3 is used to compare the total crashes and evasions in the case of a real obstacle being created and a visual notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of real obstacle creation events with visual notification to the subject is 203. Of these 215 events, 51.2% were crashes while 48.8% were evasions.

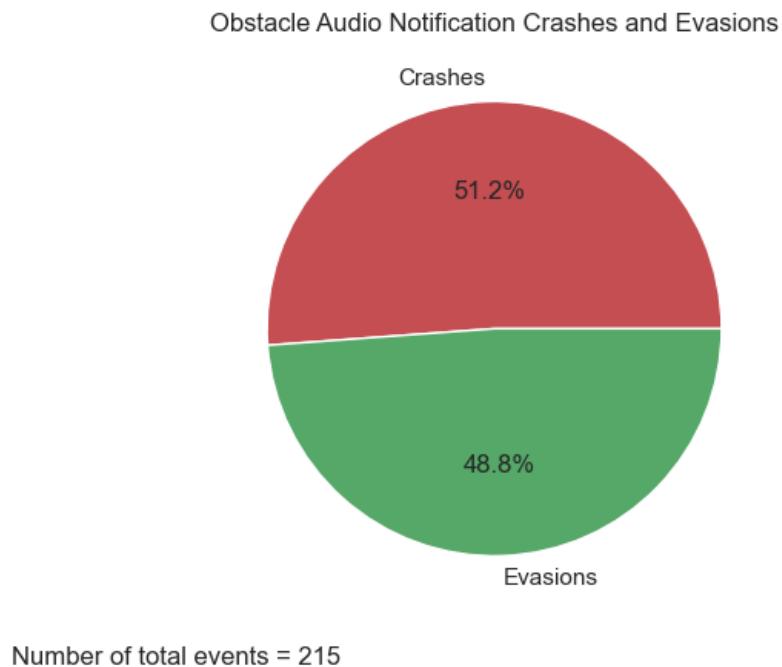


Figure 6.4: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a audio notification to the subject.

The pie chart depicted above 6.4 is used to compare the total crashes and evasions in the case of a real obstacle being created and a audio notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of real obstacle creation events with audio notification to the subject is 215. Of these 215 events, 42.9% were crashes while 57.1% were evasions.

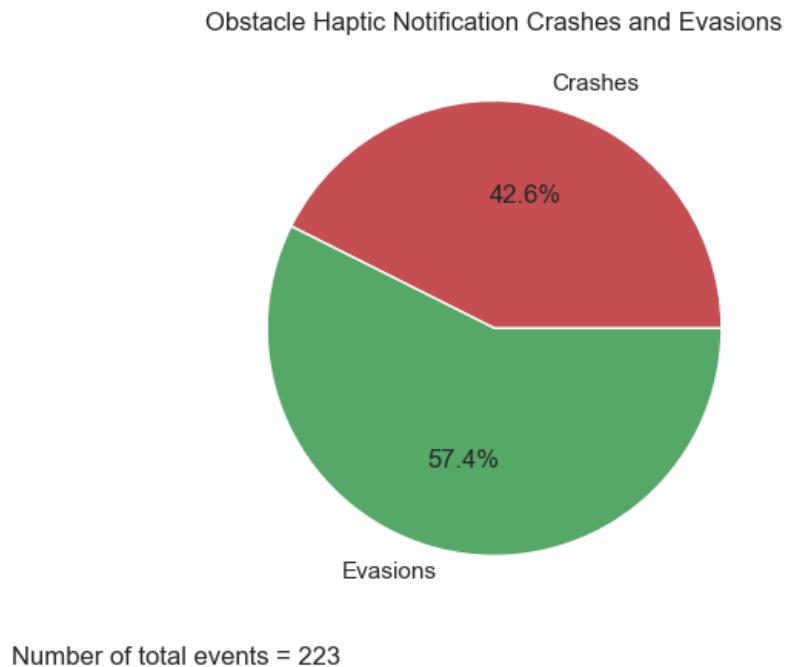


Figure 6.5: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a real obstacle created and a haptic notification to the subject.

The pie chart depicted above 6.5 is used to compare the total crashes and evasions in the case of a real obstacle being created and a haptic notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of real obstacle creation events with visual notification to the subject is 223. Of these 223 events, 42.6% were crashes while 57.4% were evasions.

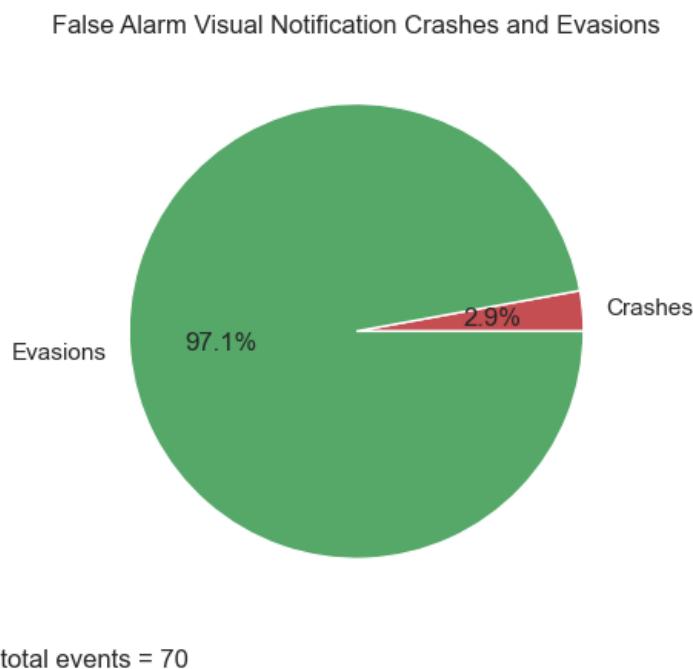


Figure 6.6: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a visual notification to the subject.

The pie chart depicted above 6.6 is used to compare the total crashes and evasions in the case of a false alarm and a visual notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of false alarm events with visual notification to the subject is 70. Of these 70 events, 2.9% were crashes while 97.1% were evasions.

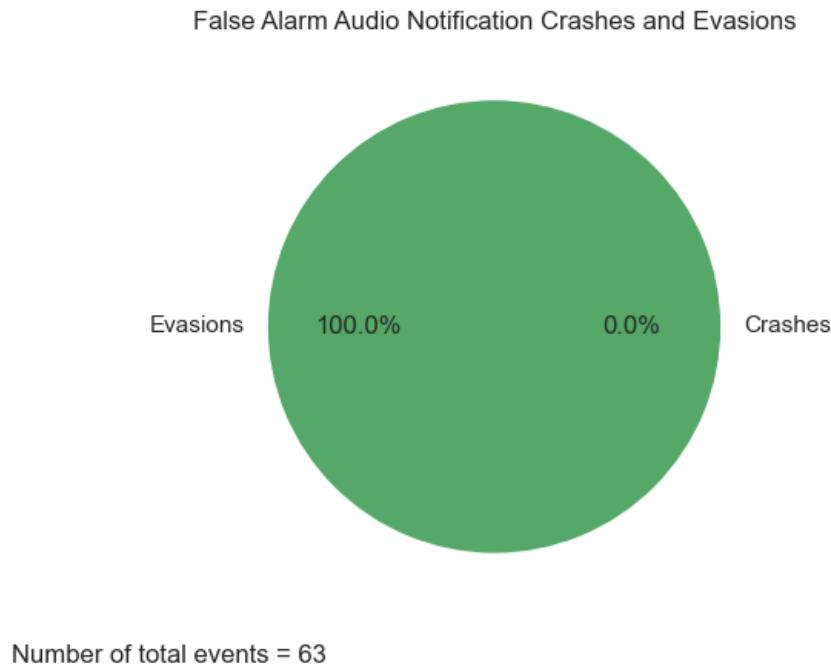


Figure 6.7: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a audio notification to the subject.

The pie chart depicted above 6.7 is used to compare the total crashes and evasions in the case of a false alarm and an audio notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of false alarm events with audio notification to the subject is 63. Of these 63 events, 0% were crashes while 100% were evasions.

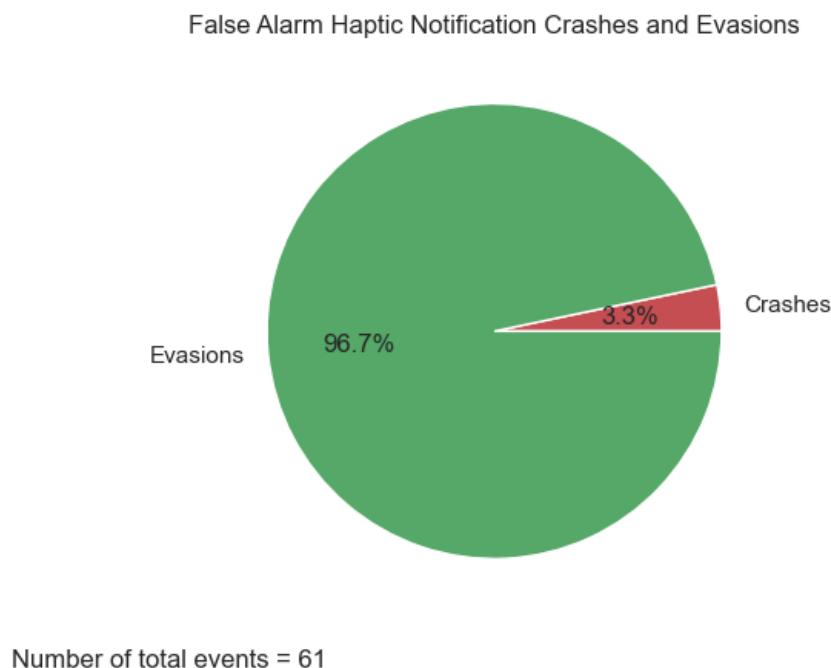


Figure 6.8: Pie chart depicting the percentage of total crashes (in red) versus total evasions (in green) in the event of a false alarm and a haptic notification to the subject.

The pie chart depicted above 6.8 is used to compare the total crashes and evasions in the case of a false alarm and an haptic notification given to the subject. The total number of events in this category is shown in the bottom left corner. The total number of false alarm events with haptic notification to the subject is 61. Of these 61 events, 3.3% were crashes while 96.7% were evasions.

We consider our experiment's balancing to be for the most part successful. All the events were distributed almost evenly while the ratio of 4 to 1 between the real obstacle creation events and the false alarms. The only imbalance was between the crashes and the evasions, the evasions being recorded twice as often as the crashes. We can also see from the charts above that the most successful way of notifying the participants for the takeover request is the haptic feedback given by the Myo controller. The second best way proved to be the visual notification on the tablet and the worst one the audio notification, in which the participants crashed more times into the obstacle or the road's sides, than they evaded them. On the other hand, in the case of false alarms, the audio notification did not have any crashes while the other two modalities had a few. However, the subjects seemed to not crash the vehicle very often in the false alarm events, meaning that, since the percentage of crashes is very low in all three modalities, their differences might not be statistically significant.

### 6.1.2 Reaction time analysis

In this subsection, we will look at the results of the Reaction Time analysis. As we mentioned in a previous chapter, we wanted to see if there is a difference between the three modalities we used with respect to the reaction time. We also wanted to see if there is a significant difference recorded between the crash and evasion conditions.

Our initial hypothesis was that in the event of a crash, there should be a greater reaction time, leaving a smaller time window for the participant to react to the obstacle, making the evasion maneuver harder, leading to either a crash on the obstacle or a hard steer of the wheel, leading to a crash on the side of the road.

Respectively, for the three different modalities, we presumed that haptic notifications would be the fastest to reach the brain, leading to a faster reaction from the participants. The next fastest stimulus to get processed by the brain would be the audio one and the slowest, the visual one. We expected to see a significant difference between these three cases. The histogram of the reaction times with respect to the different modalities can be seen in 6.9.

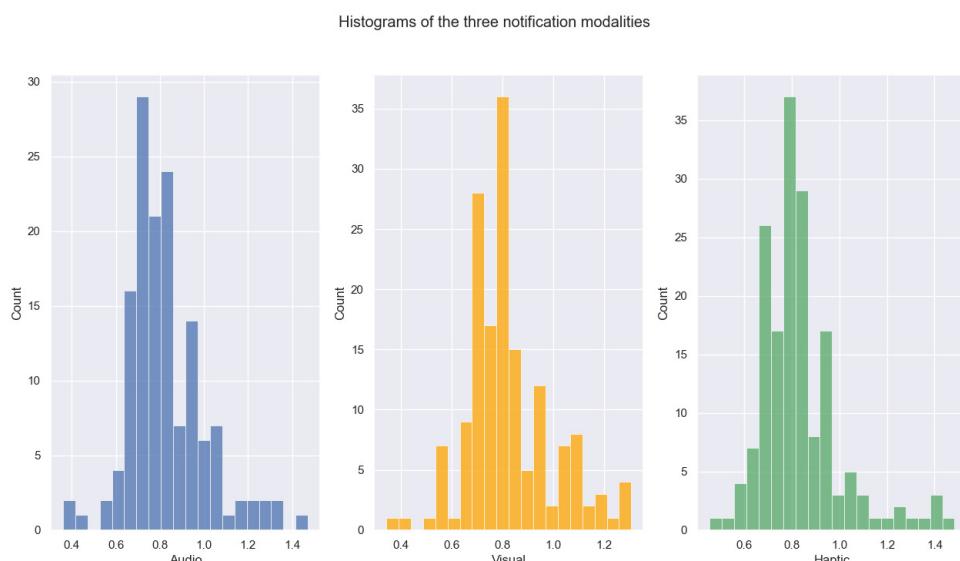


Figure 6.9: Histogram displaying the reaction times of the three different modalities of notifications used in the experiment

The above figure shows three histograms that describe the amount of reaction times recorded in a small time interval. We used 20 bins, meaning the total range of reaction times is split between 20 intervals. From left to right, the stimuli described are the audio stimulus, the visual stimulus and the haptic stimulus. All three of the modalities seem to follow a common distribution, seemingly a slightly skewed gaussian. The mean of the three modalities with respect to the reaction time is presented in the following table 6.1.

Modality	Reaction Time
Audio	0.837
Visual	0.838
Haptic	0.803

Table 6.1: Mean reaction times of the three notification modalities

In figure 6.10 we can see the distribution approximation from the data used to plot the histogram in figure 6.9. The method used for the approximation was the empirical Cummulative Distribution or eCDF.

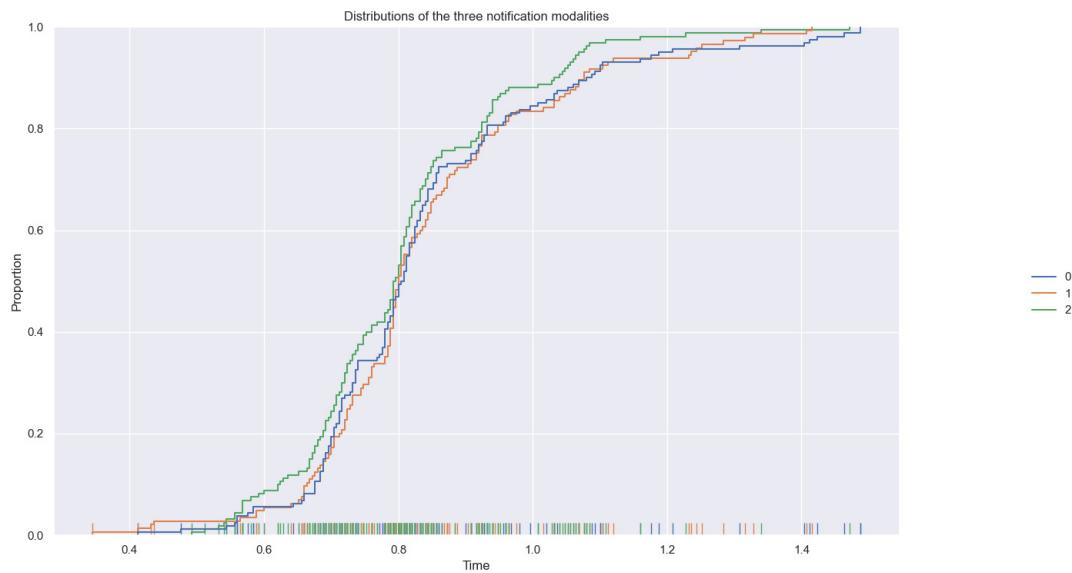


Figure 6.10: Plot of the empirical Cumulative Distribution Function of the three modalities. Visual stimulus is represented by the orange line, audio stimulus by the blue line and haptic by the green.

From figure 6.10 we can see that visual and audio stimulus have almost identical eCDFs while the haptic stimulus seems to be higher than the other two lines. This means that the haptic distribution shows less variance than the other two. The green line being above the orange and blue ones also means that the haptic stimulus results in generally faster responses. Next we will compare the crash and evasion conditions.

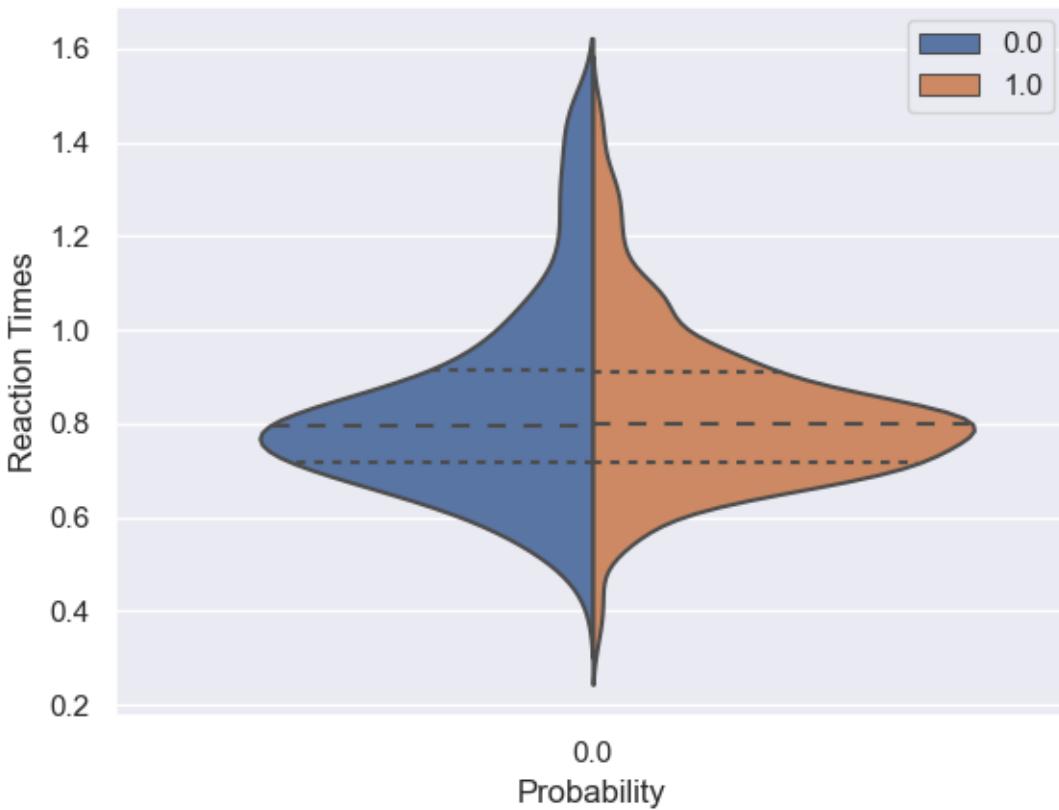


Figure 6.11: Violin plot of the crash and evasion conditions with respect to reaction time. The blue area is the distribution of evasion events and the orange area is the distribution of the crash events.

A violin plot is a hybrid of a box plot and a kernel density plot, which shows peaks in the data. It is used to visualize the distribution of numerical data. Unlike a box plot that can only show summary statistics, violin plots depict summary statistics and the density of each variable. The white dot represents the median, the thick gray bar in the center represents the interquartile range and the thin gray line represents the rest of the distribution, except for points that are determined to be “outliers” using a method that is a function of the interquartile range. On each side of the gray line is a kernel density estimation to show the distribution shape of the data. Wider sections of the violin plot represent a higher probability that members of the population will take on the given value; the skinnier sections represent a lower probability.

From our violin plot we can see that the distributions of crash and evasion cases are almost identical. We plotted dashed line to compare the median and interquartile range of the distributions and we see that the almost coincide with each other. There are slight differences in the form of the distribution function but we can not infer anything significant from these.

### 6.1.3 Conclusion

We set out to answer two different questions regarding the reaction times recorded during our experiment. The first one was if there is significant difference of the reaction time of the participants between the three notification modalities. We can safely say that visual and audio cues were similar with respect to the evoked reaction time of the participants. The haptic modality was slightly faster at a scale of 0.04 seconds which does not seem significant at the level of obstacle evasion. The other question we needed to address was, if there is a significant difference between the crash and the evasion cases when it comes to reaction times. Surprisingly, the two cases did not show any significant

difference, on the contrary, the results were almost identical, meaning that the reaction time did not play a significant role in distinguishing crashes from evasions.

### 6.1.4 Statistical analysis of EEG Signals

For the signals themselves we conducted two different investigation processes. At first, we wanted to compare the evoked potential from each modality in order to see if the previous Reaction Time results had a neurological foundation. Next, we wanted to see if there is any indicator that could differentiate the crash and evasion cases.

First, we tried to compare the event related potentials or ERPs of the three different notification modalities. There is a lot of literature that indicates that there is a point during the stimulation and reaction of the brain, that has to do with decision making after processing the stimulus itself. Most of the time, this mark is referenced as P300 because it is usually observed 300 msec after the onset of the stimulus. The P300 is described as a positive going amplitude around the area of 300-500 msec depending on the task the subjects are asked to undertake. An early, negative onset, is usually observed right before the positive peak. This peak is usually observed above the parietal lobe.

In order to study the aforementioned ERPs, we isolated the first 700 msec after the onset of the stimulus for every obstacle creation event. The following images show the average of all the epochs that correspond to a specific type of stimulus. We used the joint plots provided by MNE library, which plot the data as butterfly plots, adding the corresponding topomaps on top of time stamps that we decide beforehand. Note that in the haptic joint plot, we excluded the O1 electrode as it presented an obvious artifact, messing with the scale of the other signals.

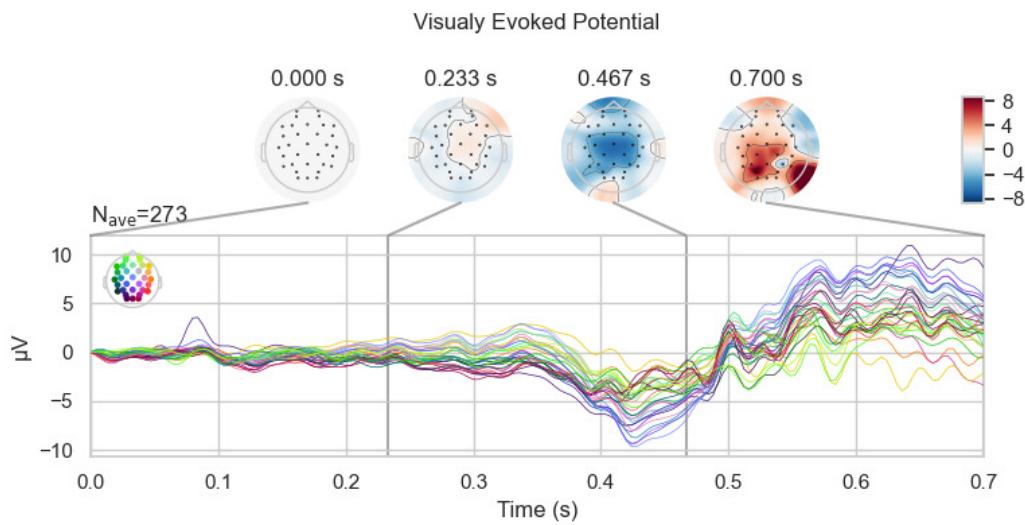


Figure 6.12: Event Related Potential of visual notifications.

From figure 6.12, we can see that the time area corresponding to the P300 ERP is detected between 350 and 500 msec. The negative onset of the ERP peak at around 410 msec and the positive peak happens at 500 msec and further.

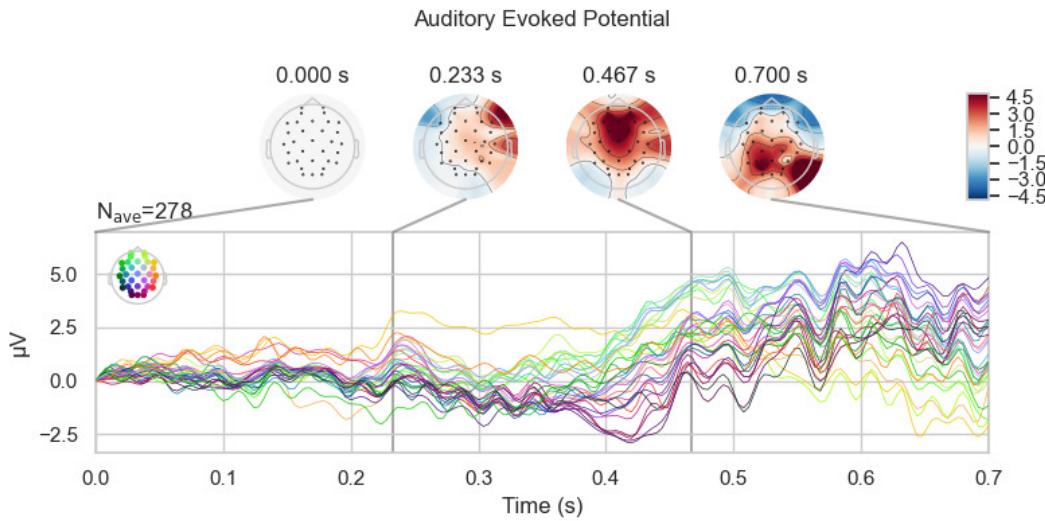


Figure 6.13: Event Related Potenital of audio notifications.

From figure 6.13, we can see that the time area corresponding to the P300 ERP is detected between 250 and 500 msecs. There is much more activity than the visually evoked potentials. The diagram is lacking a clear negative peak, but we can see that the onset of the P300 starts at around 250 to 300 msecs with a negative peak at 410 msecs and a positive peak further from 500 msecs. We can also see that there is a slight positive peak at exactly 250 msecs that is going to be discussed later.

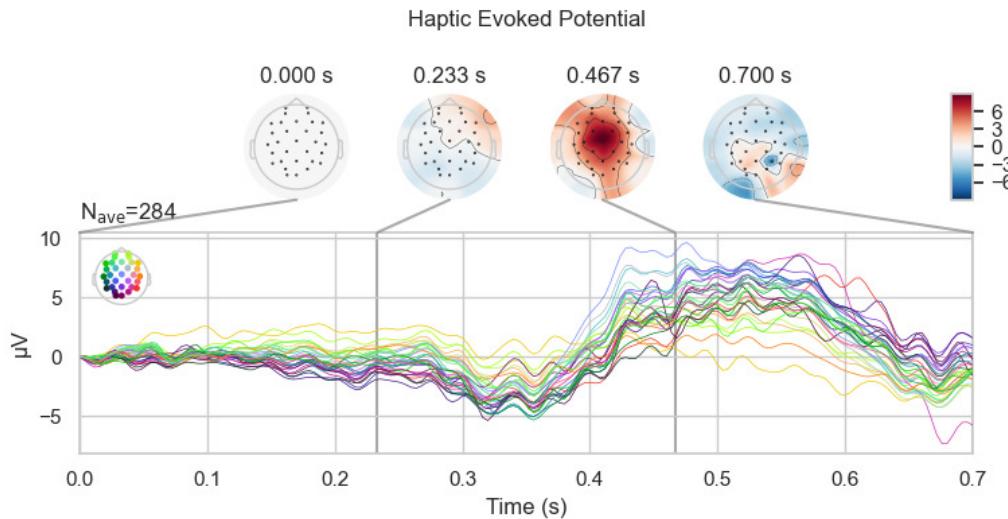


Figure 6.14: Event Related Potenital of haptic notifications.

From figure 6.14, we can see that the time area corresponding to the P300 ERP is detected between 250 and 500 msecs. Again, there is a diffusion of the signals earlier than in the visual stimulus diagram, similar to the activity of the audio stimulus diagram but with less amplitude. A negative peak can clearly be seen at 350 msecs and a positive one just before 500 msecs.

From the above diagrams we can conclude that the visual and audio ERPs are similar in timing but not with respect to the amplitude and variance of the signals. The haptic and audio stimuli are more similar with respect to the variance of the signals, but the haptic response is faster than that of both the visual and audio stimuli.

The higher diffusion of the signals in the audio and haptic cases indicates that the perception and processing of the signal is done faster in those cases than in the visual case. To add to that hypothesis,

the small positive peak at 250 msec can be characterized as a P200 ERP, usually correlated with higher order perceptual processes. We can say that the haptic and audio stimuli are processed faster than the visual stimulus.

On the other hand, the P300 ERP is observed at the same time period between the visual and audio ERPs, while the haptic stimulus clearly has an earlier P300 ERP. This can be attributed to the fact that, even though the audio stimulus is processed faster, the decision making in that case is slower than the other two cases. While visual ERP has a later onset, it is faster at decision-making, making it as fast as the audio stimulus. On the other hand, the haptic notifications, seem to be processed faster and also provide a decision faster than the other two types of stimulus.

The other type of analysis we conducted, was to compare the crash and evasion cases on the basis of the neurological response before the onset of the stimulus. We expected to see a difference in the amplitude of frequency components in certain regions of the brain, associated with higher cognitive functions like reading, in the case of a crash. We also expected higher amplitude in lower frequencies in the case of an evasion, indicating a more relaxed state of mind.

First we look at the joint plots of the two cases.

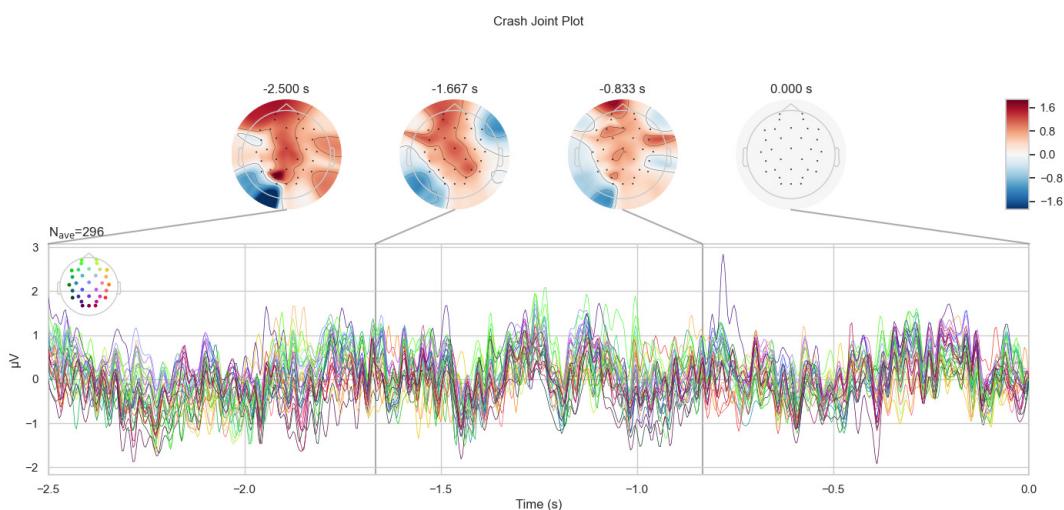


Figure 6.15: Joint plot of the crash cases

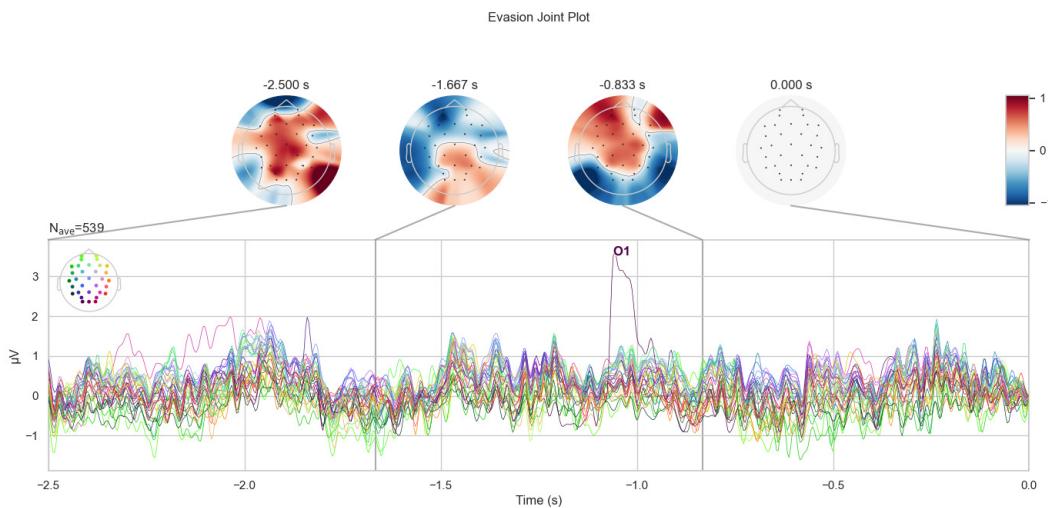


Figure 6.16: Joint plot of the evasion cases.

It is hard to extract concrete conclusions from the diagrams above. The main differences between the two, are the localization of higher activity in the time points we elected to study. On the crash cases, we can see a slight focus of activity at the center of the topological map, excluding some obvious artefacts. On the other hand, evasion cases seem to have a slightly more outspread activity. Note also, that the scale of the topomaps is different, the crash cases having higher peaks than the evasion ones.

The following diagrams show the topological map of the average of the whole 2.5 second window we showed above

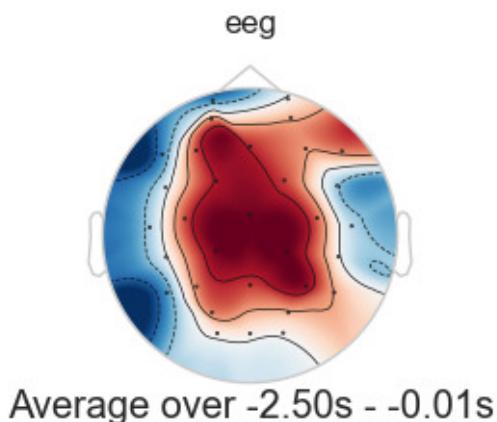


Figure 6.17: Topological map of the evasion cases using the average over the 2.5 second window before the stimulus onset.

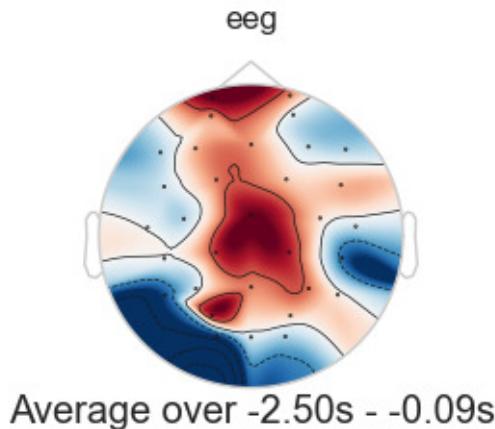


Figure 6.18: Topological map of the evasion cases using the average over the 2.5 second window before the stimulus onset.

From the two topomaps above we can see that in the crash cases, the brain activity is localized with higher density and intensity in the central regions of the brain, while in the case of the evasions, it is more spread out.

We now examine how the different frequency bands depict differences between the two conditions. We used the epochs to calculate the psd and divide it into frequency bands. We then used the information about electrode positions to insert a spatial factor into the following diagrams.

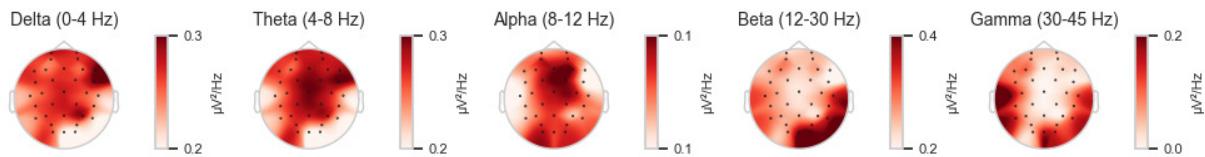


Figure 6.19: Topological map of the evasion cases, divided into frequency zones.

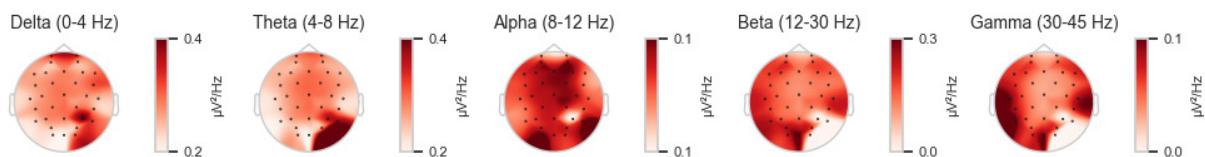


Figure 6.20: Topological map of the crash cases, divided into frequency zones.

From the above diagrams, we can see higher amplitude of lower frequencies (theta and below) in the evasion case. Higher density of alpha and beta waves can be spotted in the crash cases. However, the topographical maps are littered with single channel artefacts. This unfortunately affects the validity of our conclusions.

Lastly, we tried to directly compare the evoked data from the crash and evasion epochs. The following diagram is a comparison of the standard deviation of the channels between the two cases.

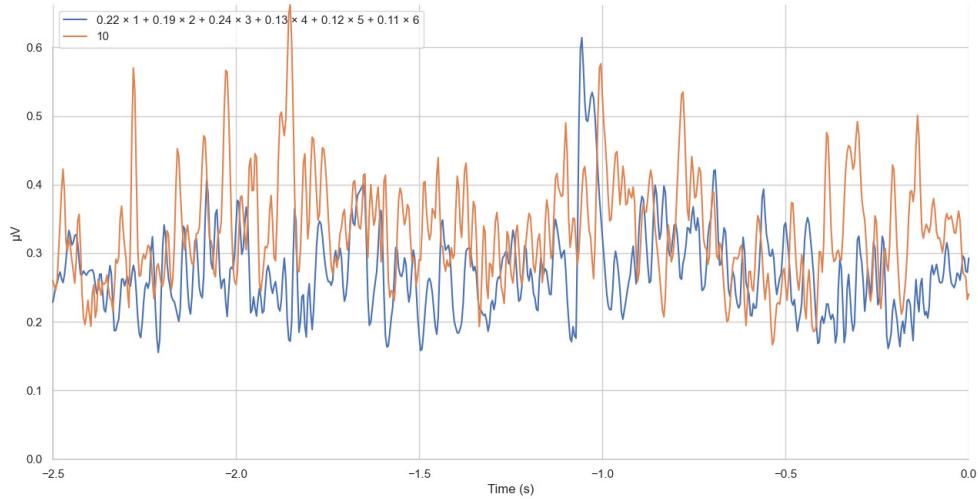


Figure 6.21: Comparison of standard deviation between crashes (orange) and evasions (blue).

We can clearly see that the inter-signal standard deviation in the case of crashes is higher throughout the 2.5 time window before the onset of the stimulus. This is useful information, as we can attribute the higher standard deviation to higher cognitive functions, resulting in a more focused participant on the distraction task.

### 6.1.5 Conclusions

Throughout this analysis we gained important insight regarding the validity of the experiment's methods, some of our initial hypotheses and the neurological interpretation behind them.

Firstly, the experiment was conducted successfully, with general balancing of the dataset. A clear oversight was the imbalance of the crash and evasion cases in the false alarm events. Even though this might end up serving as the idle case, our initial design lacked since in the false alarm cases, the crashes were almost non-existent.

During the reaction time analysis, we confronted surprising results, that did not support our initial hypotheses regarding the reaction time's correlation with accidents. The data showed that reaction time was almost identical between cases where the participants crashed the vehicle and cases where the participants evaded successfully the obstacle. It was also shown, that visual and audio notifications were very close in regards to evoking a reaction time, their means differing only some few milliseconds. Contrary to this, the haptic notifications seemed to have a lower mean reaction time, displayed in the eCDF plots.

Both these initially peculiar findings can be supported by analyzing the EEG signals. Through comparing the epoch and evoked data of the signals, we found that the mechanism behind decision making shows great timing differences between the three modalities we chose to include in the experiment. The visual notification took a longer time to be processed by the participant's brain but led to quicker decisions. On the other hand, audio stimulus was processed faster than visual stimulus, but had a longer period of deciding what to do with the provided information. Lastly, the haptic stimulus seemed to be both faster to process than the visual stimulus and faster to interpret and decide, than the audio stimulus. This can explain why the visual and audio signals had closer reaction times than the haptic one.

Another point of interest in our analysis was the higher crash rates when using the audio stimulus. Out of the three modalities, it had the highest accident rate, crashes being more than evasions. This could be explained as the audio signal took more time to be made into a decision by the participant's

brains. We could see high variance between the signals during the audio epochs, meaning that a lack of decision while the stimulus was processed earlier, could lead to anxiety. This could be also interpreted as, taking more time to calculate a decision, produced a worse decision and worse control over its execution.

Lastly, the EEG analysis between crash and evasion cases showed timid differences. Yet, the expected result, namely the higher frequency component's amplitude, was found in figures 6.19 and 6.20. Additionally, a constant increase of standard deviation was seen during the 2.5 second time window we chose for our analysis, in the crash cases. This can also be interpreted as a localization of higher frequency components in some parts of the brain that could indicate a clear neurological cause.

One blank spot of our analysis that is going to be analyzed in the Limitations and Future Work chapter, is that we can not be sure about the actual higher cognitive functions being recorded during the crash cases. Most of the participants reported that they had trouble using the Leap Motion Controller and that most crashes happened when they were trying to scroll through the text, open a new text or try to fix glitches in the simulation. This subjective claim can not be verified because we did not keep the hand tracking data.

## 6.2 Classification Results

We tried to classify the events, based on the EEG recording before the notification to the participant commenced, between the crash and evasion outcomes. We tried using different models in order to find if there are differences between the models and if we can achieve a higher score using a particular model. The models used in the classification problem were analyzed in detail in the previous chapter.

Namely, the models used were:

1. C-Support Vector Classification (SVC)
2. Decision Tree
3. Random Forest Classifier
4. Logistic Regression

In order to get the best accuracy score out of every model, we fine tuned each one of them separately with regards to their hyper-parameters.

There is a difference between a simple parameter and a hyper-parameter in machine and deep learning problems. A simple model parameter is usually referred to the parameters of the model that are used to calculate the output directly. Usually these parameters are referred to as weights and they are the ones that change during the learning process. A hyper-parameter is a determining factor of the model and stays the same during the training of the model. It can change when we validate the model in order to maximize its efficiency by adapting it to the data at hand. Examples of these hyper-parameters are regularization parameters, number of submodels used to solve a bigger problem, number of iterations in a computational procedure and so on.

During the validation of our models, we changed some of their hyper-parameters that we considered could provide better accuracy scores and in many cases we observed such results. We used 80% of the data to train the model and 20% of the data to validate it, as it is common practice for machine learning training. One exception was the simple Decision Tree, the hyper-parameters of which did not offer a better accuracy score when tampered with. One hyper-parameter common across all models was the number of components calculated through the Common Spatial Patterns feature extraction preprocessing. We used the optimal CSP components for each model in order to further validate its hyper-parameters. The following subsections are dedicated to each model used and the results gathered from the validation process.

### 6.2.1 SVC with polynomial kernel

We used the Python library sci-kit learn to implement the C-Support Vector Classification or SVC model. The implementation of this model is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. We tried to implement SVC with three different types of kernels, a polynomial kernel, the well known radial basis function or rbf kernel and a linear kernel in order to see in non-linearities affected the accuracy of the model.

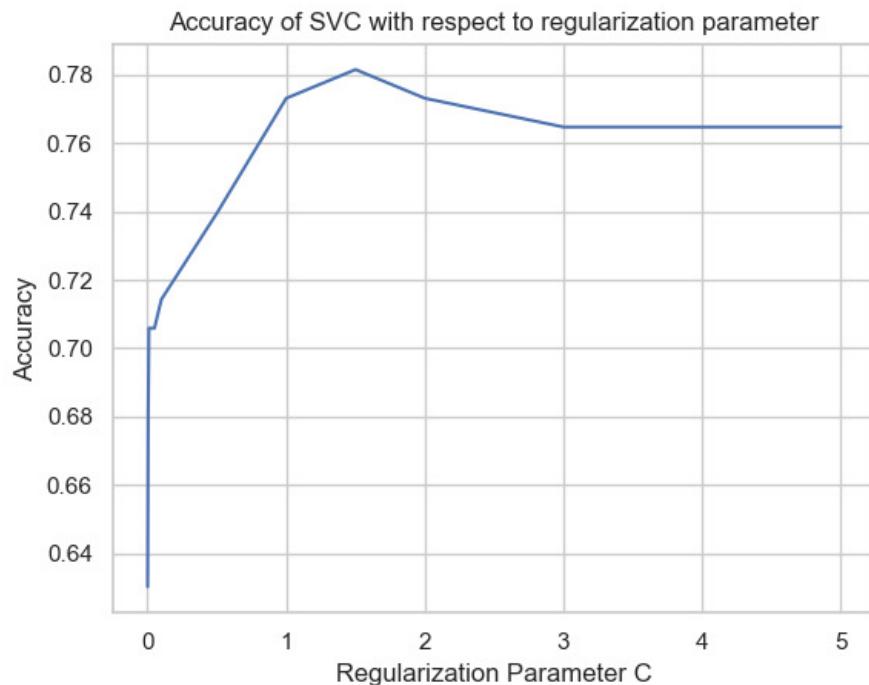


Figure 6.22: Diagram of the regularization parameter C with regards to the model's accuracy score.

The above graph depicts the relation between the regularization parameter C of the SVC model and the accuracy of its predictions. We can see a peak of accuracy at the 1.5 mark. When implementing the SVC with polynomial kernel in the future, the hyper-parameter C will be set at 1.5.

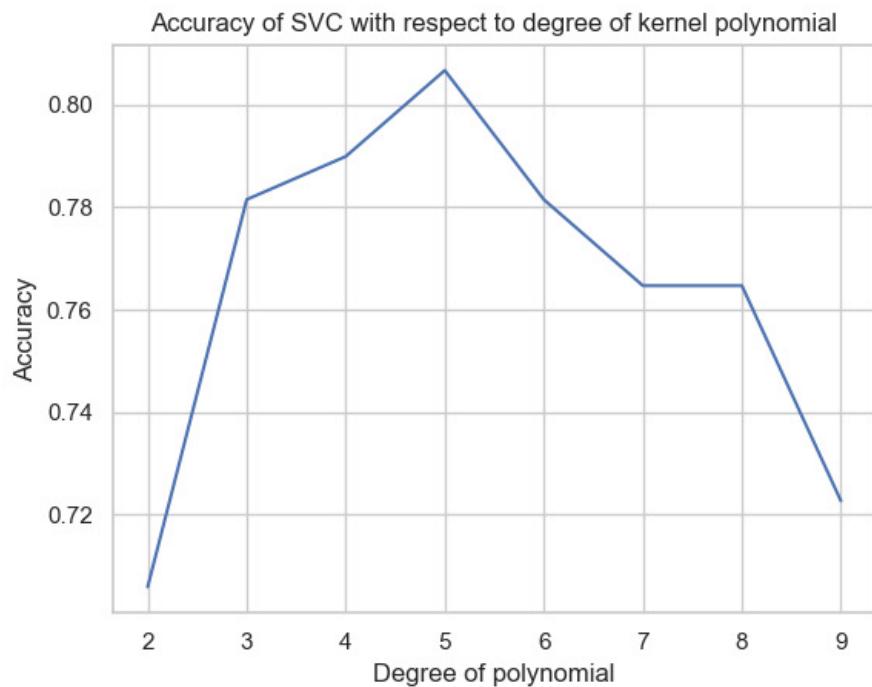


Figure 6.23: Diagram of the kernel's polynomial degree with regards to the model's accuracy score.

The above graph depicts the relation between the degree of the polynomial kernel of the SVC model and the accuracy of its predictions. We can see a peak of accuracy at the power of 5. When implementing the SVC with polynomial kernel in the future, the degree of the polynomial will be set at 5.

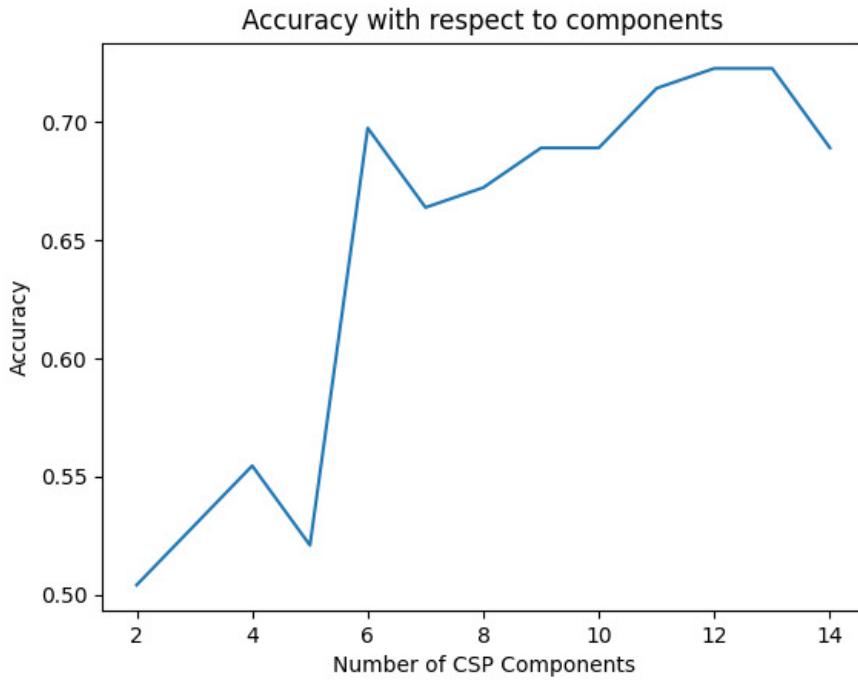


Figure 6.24: Diagram of the CSP components with regards to the accuracy score of the SCV with polynomial kernel.

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. We can see a peak of accuracy at 13 components, with a notable high score at 6 components. When implementing the SVC with polynomial kernel in the future, 13 components were used by the CSP filter.

### 6.2.2 SVC with radial basis function kernel

RBF kernels are the most generalized form of kernelization and is one of the most widely used kernels due to its similarity to the Gaussian distribution. Out of the three kernels we used (linear, polynomial and rbf) the rbf kernels are the most computationally expensive but transform the data in a highly non linear way. We used the rbf kernel because of that high non-linear nature, in order to see if our data can be linearly distinguished or if a non-linear separation is needed.

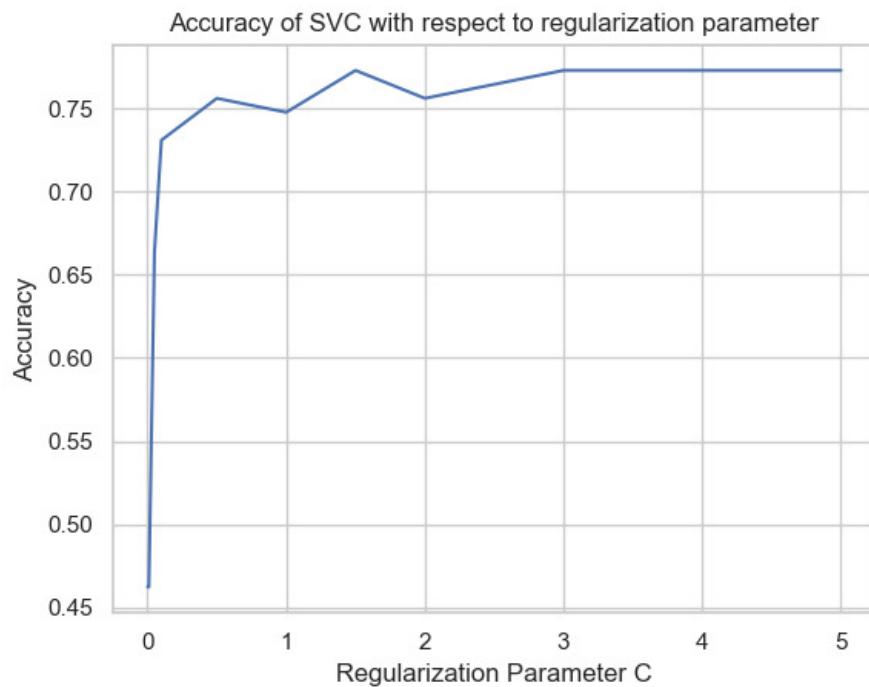


Figure 6.25: Diagram of the regularization parameter C with regards to the model's accuracy score using Radial Basis Function.

The above graph depicts the relation between the regularization parameter C of the SVC model and the accuracy of its predictions. We can see a peak of accuracy at the 1.5 mark, similar to the polynomial kernel. When implementing the SVC with rbf kernel in the future, the hyper-parameter C will be set at 1.5.

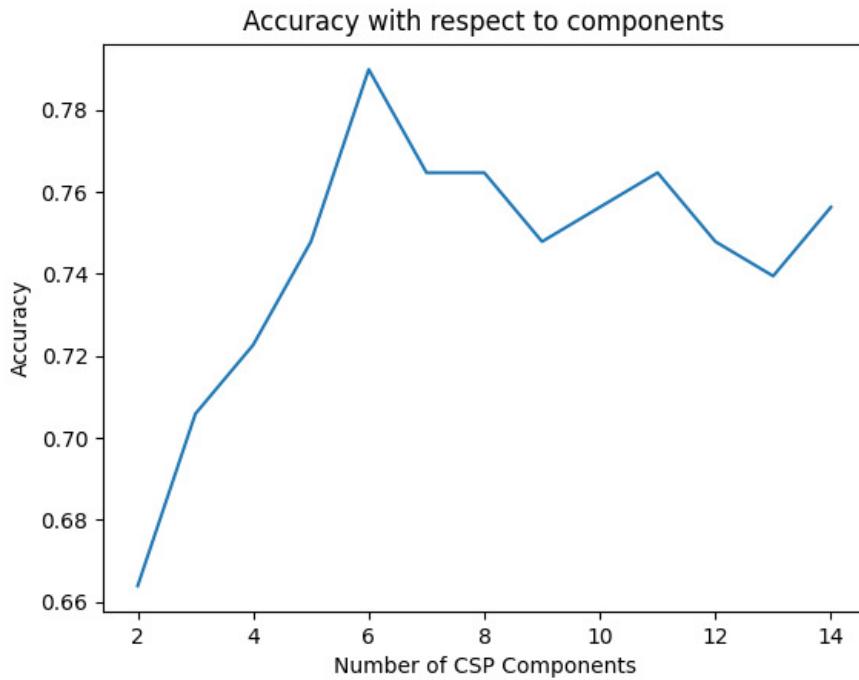


Figure 6.26: Diagram of the CSP components with regards to the accuracy score of the SCV with RBF kernel.

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. We can clearly see a peak of accuracy at 6 components. When implementing the SVC with RBF kernel in the future, 6 components will be used by the CSP filter.

### 6.2.3 SVC with linear kernel

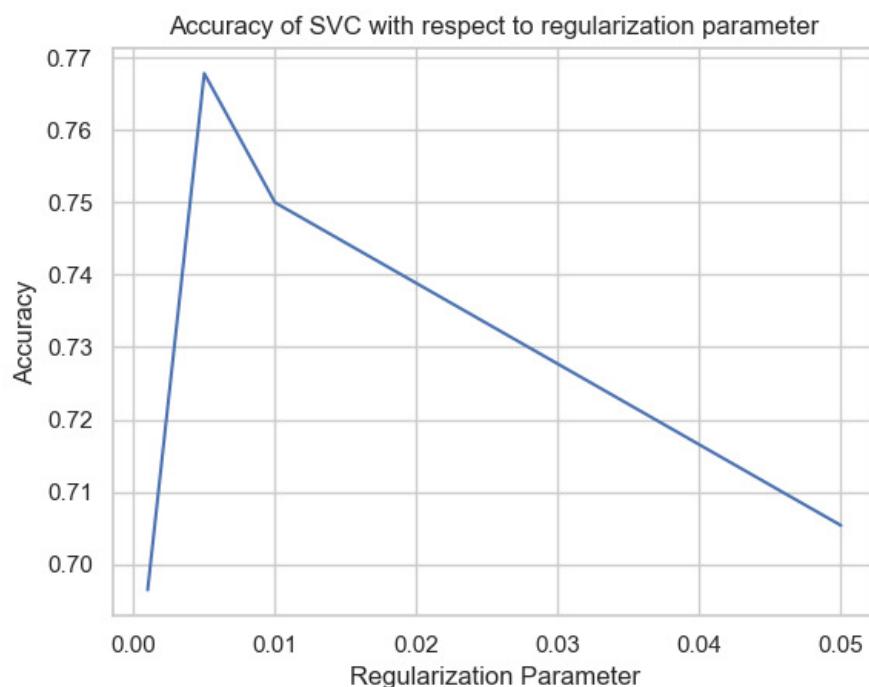


Figure 6.27: Diagram of the regularization parameter C with regards to the model's accuracy score when using Linear kernel.

The above graph depicts the relation between the regularization parameter C of the SVC model and the accuracy of its predictions. We can see a peak of accuracy at the 0.005 mark. When implementing the SVC with linear kernel in the future, the hyper-parameter C will be set at 0.005.

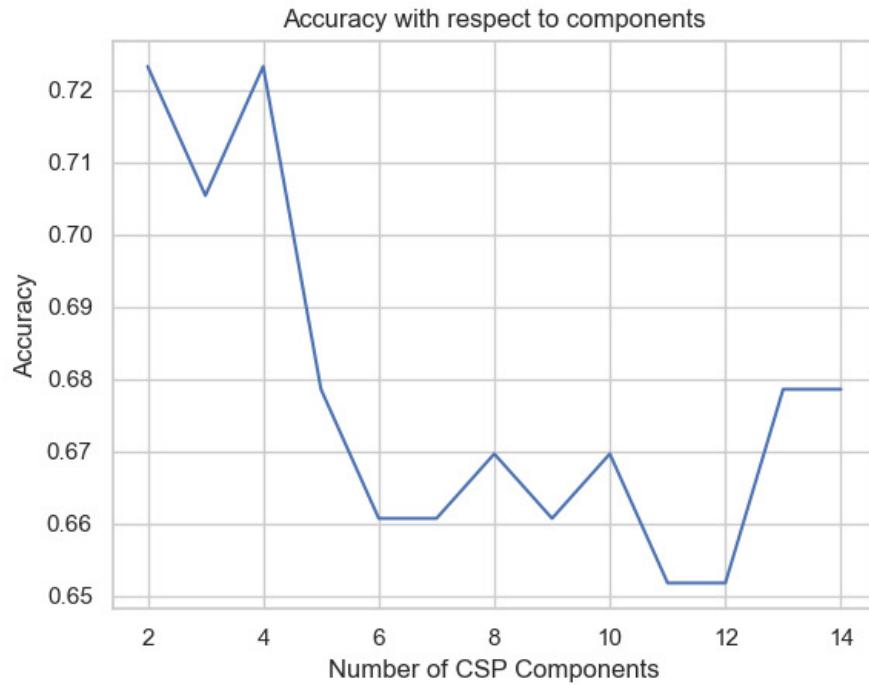


Figure 6.28: Diagram of the CSP components with regards to the accuracy score of the SCV with polynomial kernel.

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. We can see a peak of accuracy at 4 components making the linear approach the one that uses the least components from CSP filtering.. When implementing the SVC with linear kernel in the future, 4 components were used by the CSP filter.

## 6.2.4 Decision Tree

The decision tree was trained on the average power of the components created by the CSP algorithm. A very interesting feature of decision trees is their interpretability. As we analyzed in a previous chapter, decision trees create rules in order to split the data into sub sets with the minimum entropy possible. We used sci-kit learn to train and visualize the decision tree.

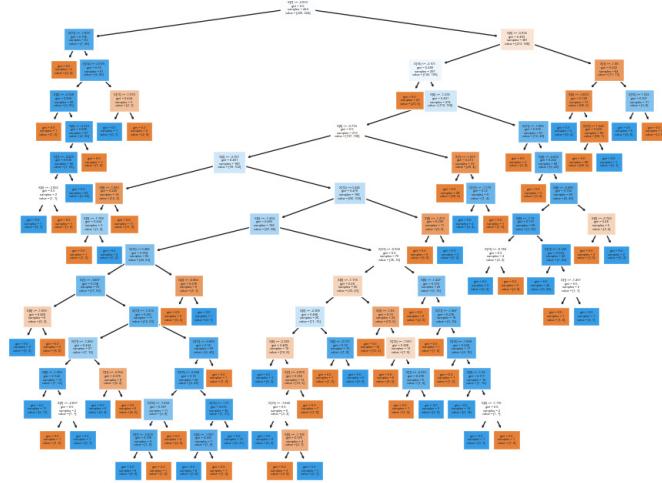


Figure 6.29: The decision tree created by training on 80% of the data

We can see that the resulting tree is a deep tree, having up to 15 branching points. Each branch includes the rule that is used to split the data, the samples remaining to be split with the number of samples belonging to each class and the gini score.

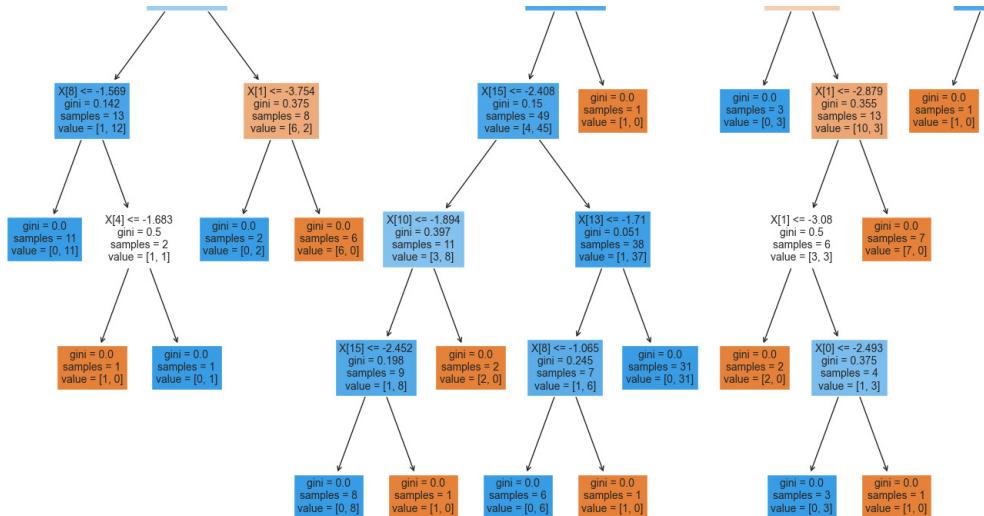


Figure 6.30: Same decision tree, zoomed in order to

In figure 6.30, we can see a small subtree of the initial decision tree in order to examine an example of the tree's structure. In the top right and down branch we can see a rule created by the decision tree. The tree splits the data based on the average power of the first component provided by the CSP filtering. If that component, denoted as  $X[0]$ , has a value less than -2.493. The samples to be split, are 4, one being in the first category while 3 belonging to the second. The gini index at that point of the tree is 0.375. After the splitting of the data, we can see that the resulting subsets contain samples from only one category, splitting the data perfectly.

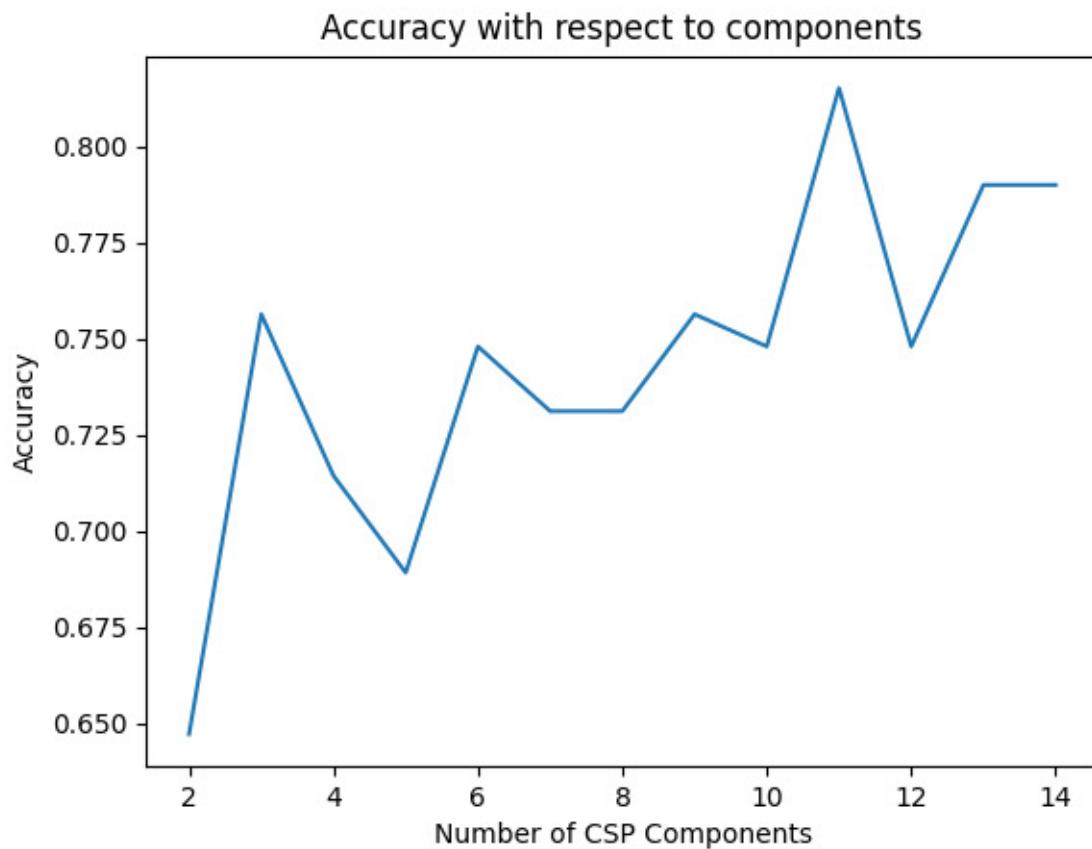


Figure 6.31: Diagram of the CSP components with regards to the decision tree's accuracy score

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. We can see a peak of accuracy at 11 components. When implementing the Decision Tree algorithm 11 components will be used by the CSP filter.

### 6.2.5 Random Forset Classifier

The Random Forest Classifier is an ensemble method using multiple decision trees, trained on a smaller and randomly selected sample of the training dataset. We used the sci-kit learn library's ensemble sub library to implement our model. First we found the optimal number of estimators for the model and then the optimal number of CSP components to be used.

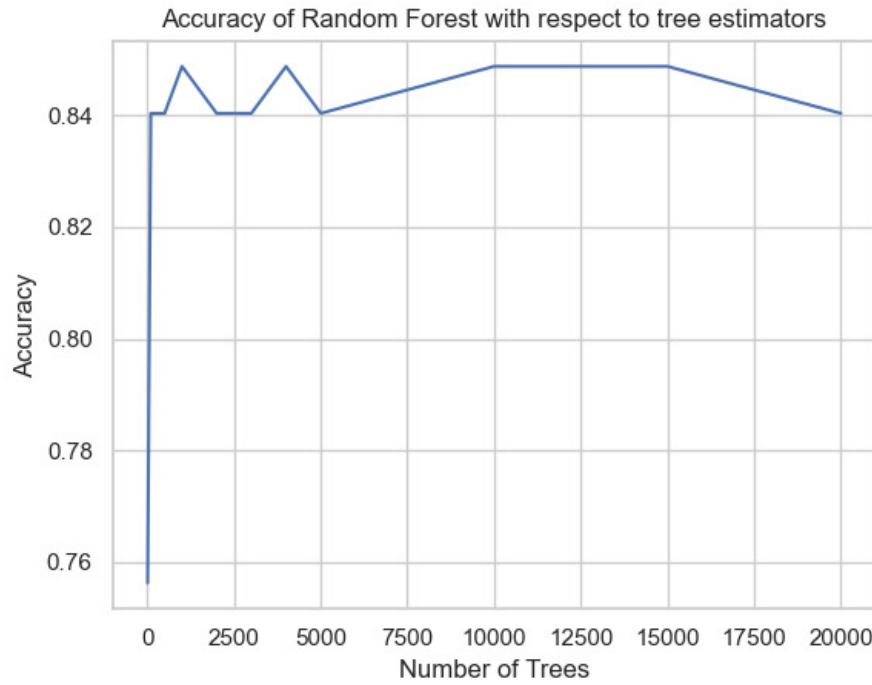


Figure 6.32: Diagram of the number of estimators (decision trees) with regards to the model's accuracy score.

The above graph depicts the relation between the number of decision tree estimators of the Random Forest algorithm with the accuracy of its predictions. We can see a peak of accuracy at 1000 and 3000 components. When implementing the Random Forest Algorithm in the future, the number of estimators will be set at 1000.

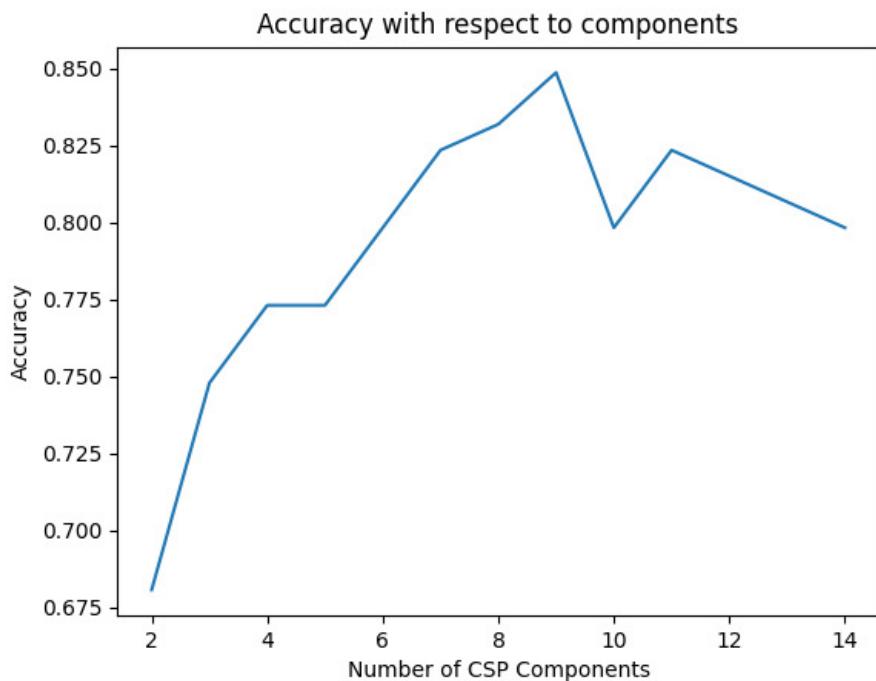


Figure 6.33: Diagram of the CSP components with regards to the Random Forest Classifier accuracy score

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. We can see a peak of accuracy at 9 components. When implementing the Random Forest algorithm 9 components will be used by the CSP filter.

### 6.2.6 Logistic Regression

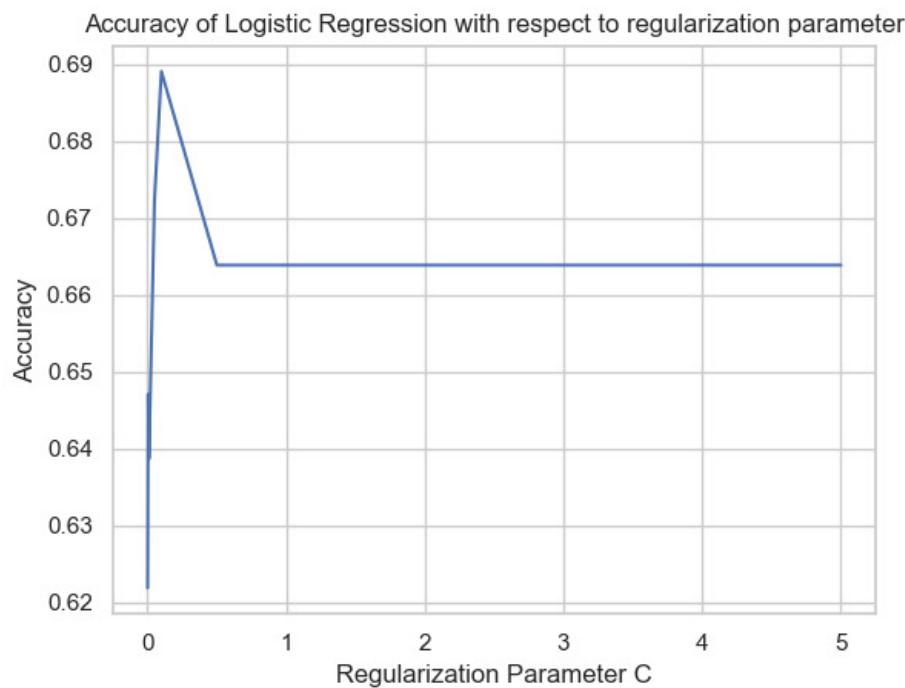


Figure 6.34: Diagram of the regularization parameter C with regards to the Logistic Regression's accuracy score.

The above graph depicts the relation between the value of the regularization parameter C of the Logistic Regression algorithm with the accuracy of its predictions. We can see a peak of accuracy at a value of 0.1. When implementing the Logistic Regression in the future, the value of C will be set at 0.1.

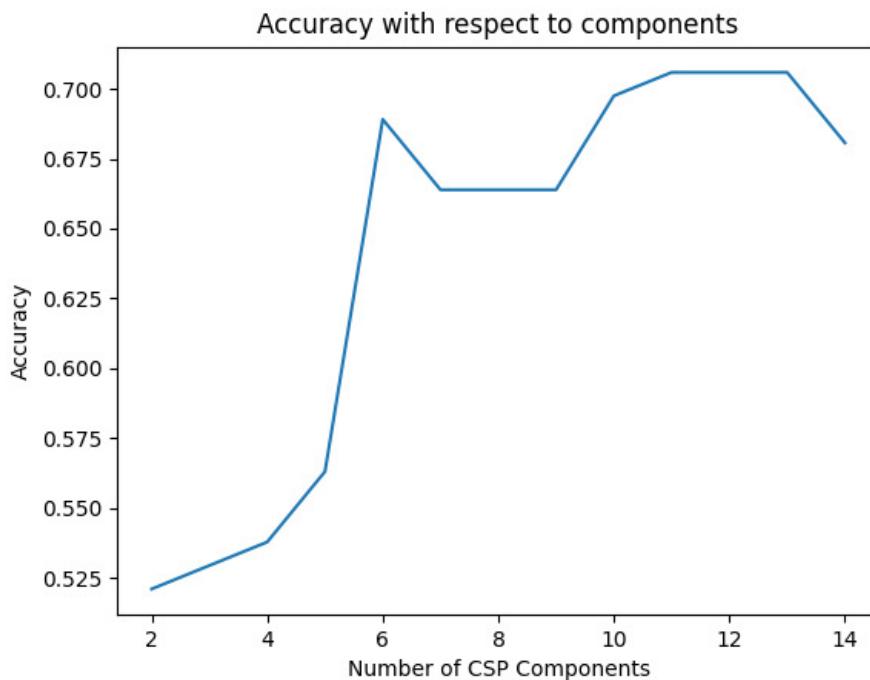


Figure 6.35: Diagram of the CSP components with regards to the Logistic Regression's accuracy score

The above graph depicts the relation between the number of components calculated by the CSP process and the accuracy of the model's predictions. The results are very similar with figure 6.28 . We can see a peak of accuracy at 13 components, with a notable high score at 6 components. When implementing the Logistic Regression Algorithm in the future, 13 components were used by the CSP filter.

### 6.2.7 Classification scores

Following the fine tuning process of each model we tried to fit the data into the models using the best hyper-parameter configuration possible. The following table 6.2 contains the results from each model.

Model	Accuracy	Macro F1 average	Weighted F1 average
SVC polynomial kernel	0.81	0.80	0.81
SVC rbf kernel	0.79	0.76	0.79
SVC linear kernel	0.76	0.75	0.76
Decision Tree	0.8	0.79	0.8
Random Forest	0.85	0.84	0.85
Logistic Regression	0.71	0.67	0.71

Table 6.2: Accuracy, macro F1 average and Weighted F1 average score of each model

We can see that the best model is the Random Forest model with 85% accuracy. The worst model we tried was Logistic Regression which seemed to not be able to approximate the underlying mathematical functions easily. An interesting thing to note is that the macro average is very close to the weighted average meaning that the two classes are equally well predicted. The next table 6.3 presents the Precision and Recall scores for the two classes for each model.

Model	Crash Precision	Crash Recall	Evasion Precision	Evasion Recall
SVC polynomial kernel	0.86	0.84	0.74	0.76
SVC rbf kernel	0.73	0.8	0.79	0.78
SVC linear kernel	0.75	0.73	0.70	0.79
Decision Tree	0.85	0.79	0.77	0.89
Random Forest	0.80	0.93	0.83	0.81
Logistic Regression	0.78	0.79	0.6	0.53

Table 6.3: Precision and Recall scores of both crash and evasion cases of each model

From the above tables, we can see that the Random Forest Classifier is the best fit for this data. It is also evident that there exists a strong correlation between the patterns extracted from the CSP algorithm with the outcome of the events. In order to further investigate, we will visualize some of the patterns. The following image shows the 9 components used in the training of the Random Forest model. Note that some channels have been removed due to obvious artefacts that hindered the display of the useful data.

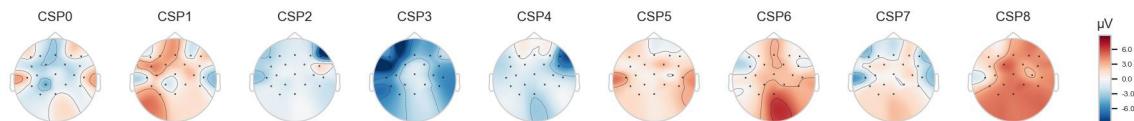


Figure 6.36: Patterns extracted by the CSP algorithm

From the figure above we can see that there is general activity in the central regions of the brain that signify a difference between the crash and evasion cases. Similarly to the EEG analysis, we can say that focusing on the distraction task can be observed as activity at the front and middle of the brain, and especially in the frontal lobe. We can also see that some patterns have significant positive activity while others significant negative activity. We assume that these patterns are extracted from the difference in activity between the two cases.

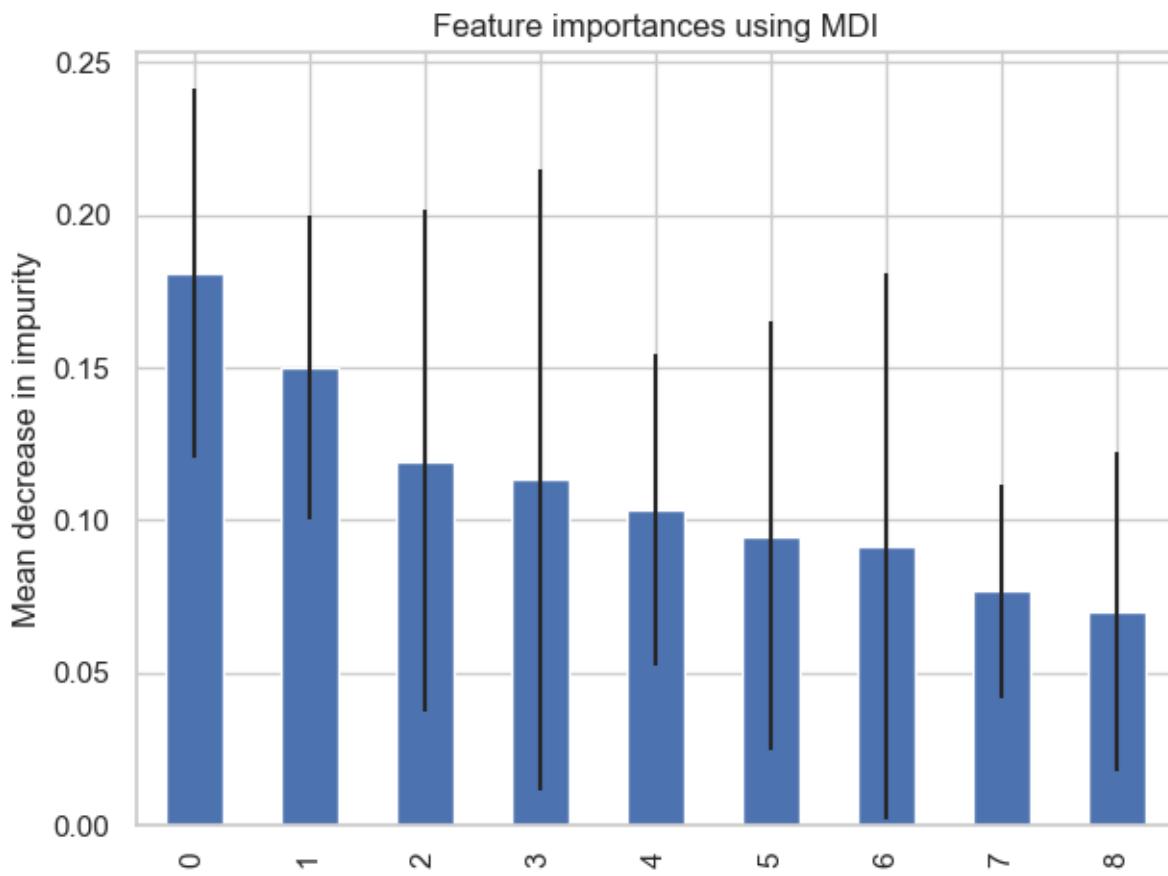


Figure 6.37: Mean Decrease in Impurity of each feature

From figure 6.37 we can see that the order that the patterns are presented in figure 6.36 is the order of importance for the model as well. The Mean Decrease in Impurity, or MDI, counts the times a feature is used to split a node, weighted by the number of samples it splits.

Based on these results we conclude that the classification task was successfully addressed. We showed that there is a way to correlate brain activity before the obstacle creation events with the outcome of the event. According to our goals that were stated initially, we can be optimistic about the creation of a system that will be able to tell when a driver is distracted to a point at which he will not be able to complete a takeover request completely.

## 6.3 Regression Results

In this section we will go over the methods we used to estimate the participant's reaction times. Predicting the exact time seems like a very unlikely task so we should consider our findings in the reaction time analysis. Unfortunately, we could not find significant difference between crash and evasion cases and a difference between notification modalities seem to have very narrow margins. With these findings in mind, we should consider this problem as a hard one, setting our goals accordingly. If we could show that it is possible to estimate the reaction time with a small mean error and a model that can explain a portion of the data's variance.

In order to achieve that, we used the average power of the SPoC filtered signals. Three different models were used for this task:

1. Support Vector Regression (SVR)
2. Random Forest Regression
3. Ridge Regression

### 6.3.1 SVR

The first model considered was the SVR that provided some interesting results in the classification analysis. Considering the performance of the different kernels in the classification analysis, we decided to train a SVR model using only the polynomial kernel and fine tune it.

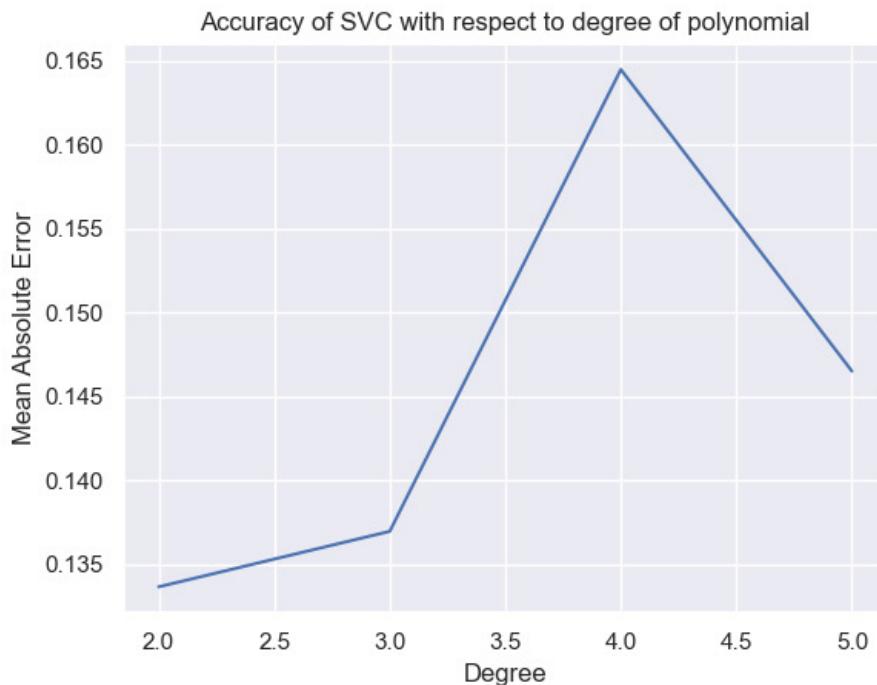


Figure 6.38: Diagram of the Mean Absolute Error with respect to the degree of the SVR kernel's polynomial

The figure above shows the mean absolute error of the validation set in relation to the degree of the polynomial kernel. We can see that the smallest error is observed at a degree of 2, meaning that the data does not have a higher order polynomial relation.

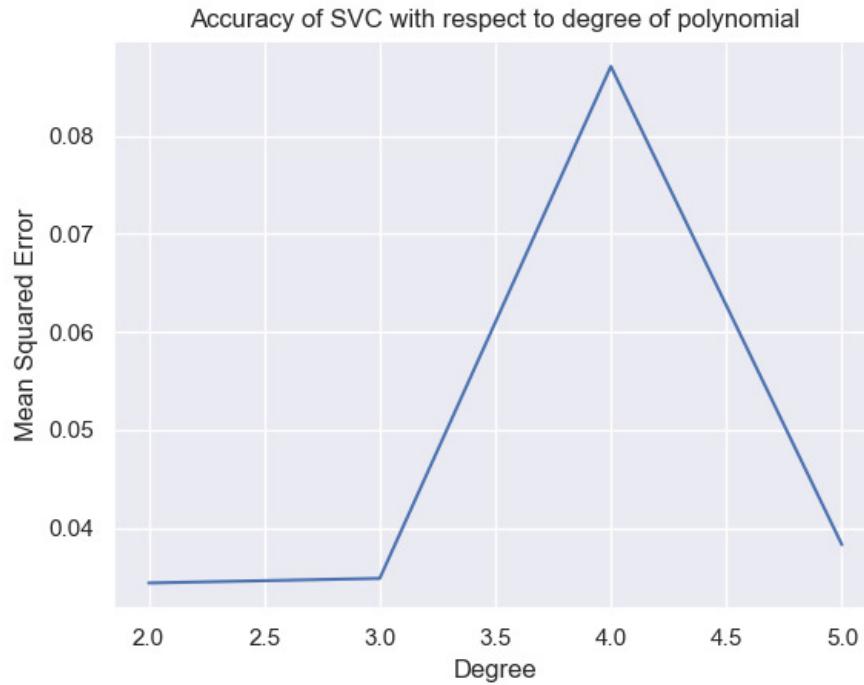


Figure 6.39: Diagram of the Mean Squared Error with respect to the degree of the SVR kernel's polynomial

The figure above shows the mean squared error of the validation set in relation to the degree of the polynomial kernel. Similarly to the mean absolute error diagram we can see that the smallest error is observed at a degree of 2, meaning that the data does not have a higher order polynomial relation.

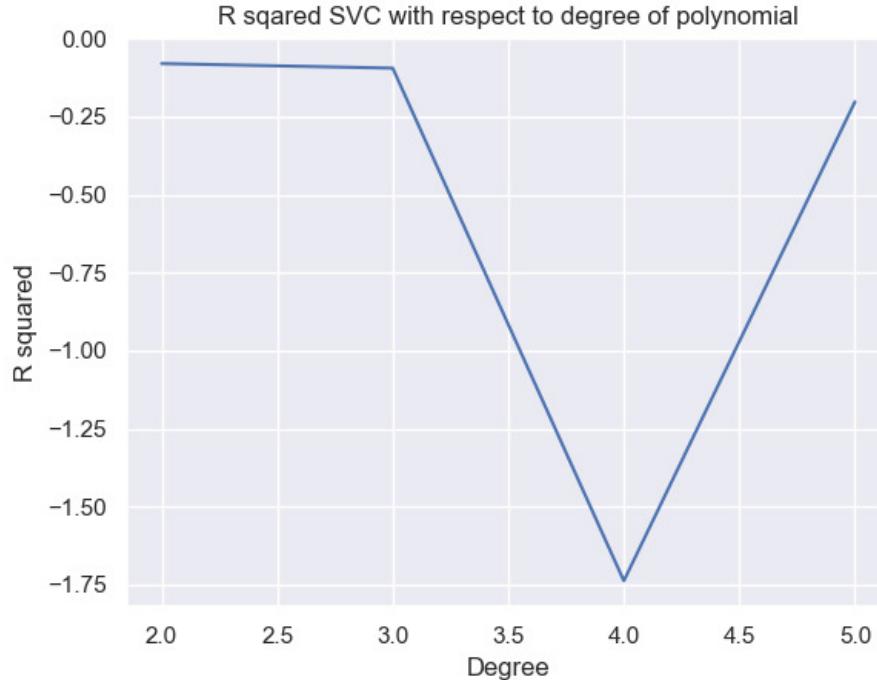


Figure 6.40: Diagram of the R squared metric with respect to the degree of the SVR kernel's polynomial

The figure above shows the R squared metric of the validation set in relation to the degree of the

polynomial kernel. We can see that the highest R squared score is observed at a degree of 2, meaning that the data does not have a higher order polynomial relation. Even further, the R squared metric stays below zero. In the regression metrics section, we mentioned that the R squared metric can have negative values if the model performs worse than a straight line estimator. This concludes that the data do not have a polynomial relation whatsoever.

### 6.3.2 Random Forest Regression

A more sophisticated and complex non-linear relation can be uncovered by the Random Forest Regression which uses an ensemble method to introduce its non linearity. Similarly to all the previous sections we will fine tune the model to see if the maximum R squared metric can over-pass the value of 0 giving hints of a non linear underlying function, governing the data.

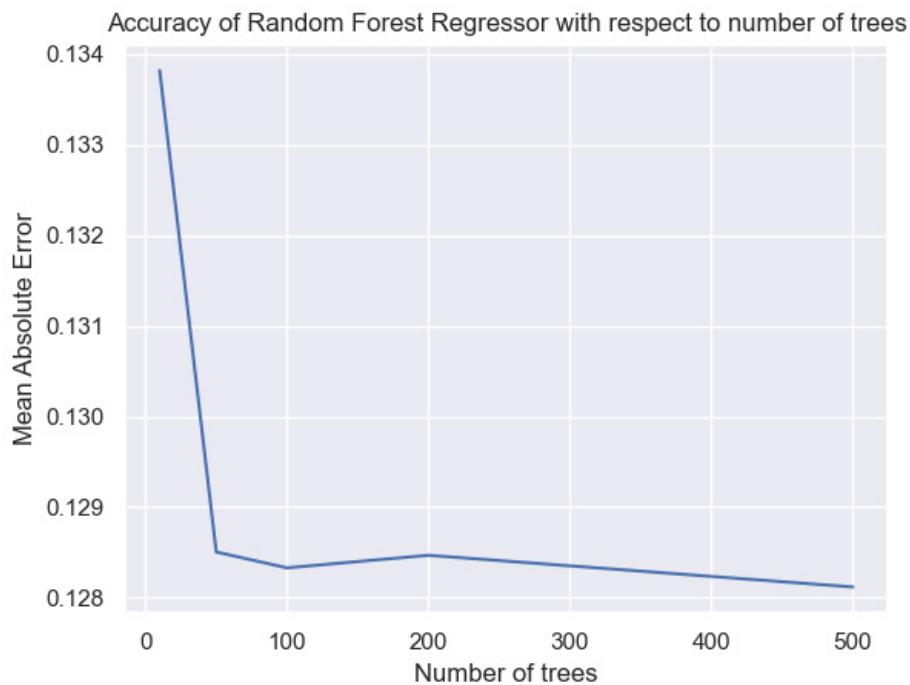


Figure 6.41: Diagram of the Mean Absolute Error with respect to the degree of the number of estimators used

The figure above shows the mean absolute error of the validation set in relation to the number of decision tree regressor estimators used in our model. Similarly to the mean absolute error diagram we can see that the smallest error is observed a number of estimators equal to 500. We can also see that the largest difference between number of estimators is observed at the 10 to 50 step. The 100 estimators seem to be a good choice in order to minimize the time of training required while keeping the mean absolute error score low.

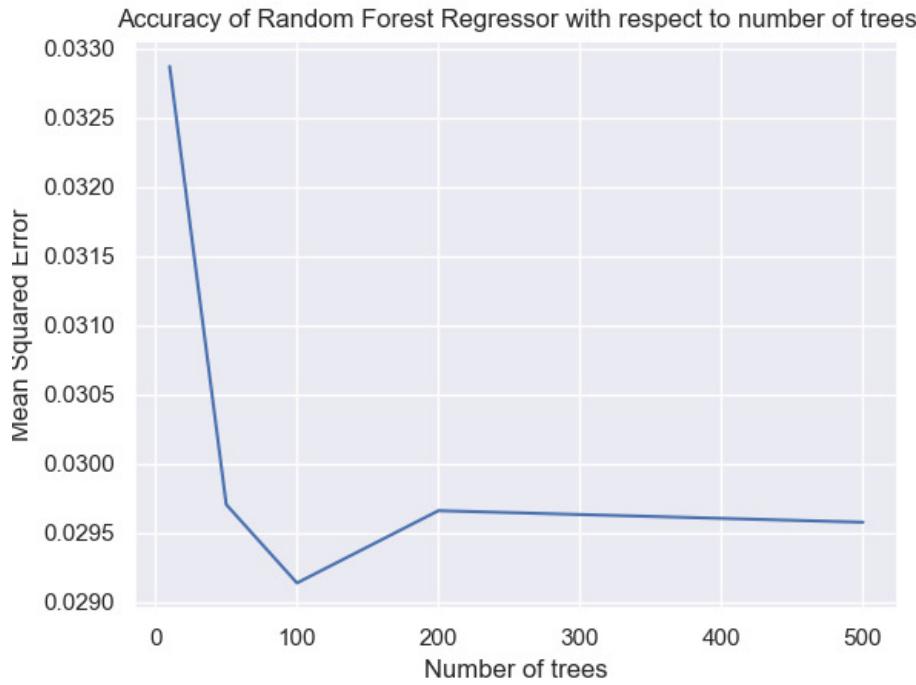


Figure 6.42: Diagram of the Mean Squared Error with respect to the degree of the number of estimators used

The figure above shows the mean squared error of the validation set in relation to the number of decision tree regressor estimators used in our model. The Mean Squared Error values indicate the same thing as the Mean Absolute Error diagram. Specifically, the best number of estimators is 100.

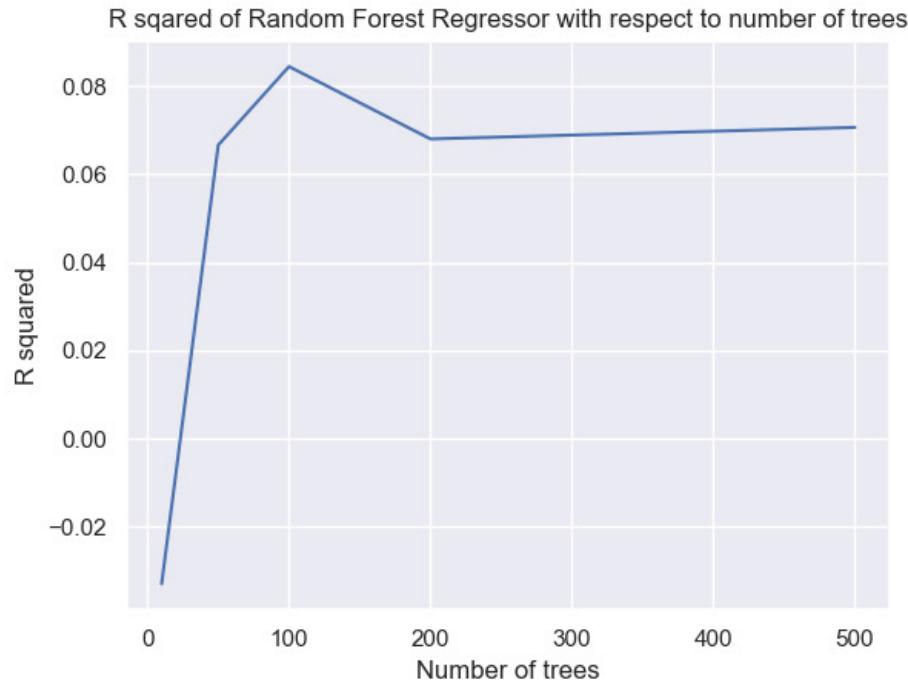


Figure 6.43: Diagram of the R squared metric with respect to the degree of the number of estimators used

The figure above shows the R squared metric of the validation set in relation to the number of

decision tree regressor estimators used in our model. We can see that the metric is positive this time but at a very low mark of just 0.08. This means that our model can not explain most of the variance in the model.

### 6.3.3 Ridge Regression

The last model we tried was the Ridge Regression model. This model is a robust regression model that is especially useful when we have to deal with a small amount of features and training data.

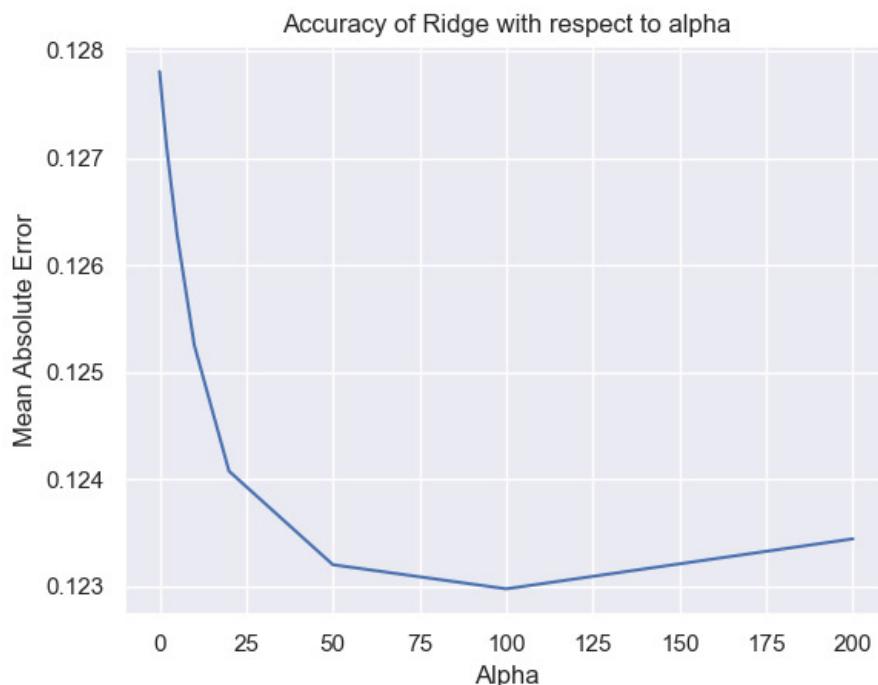


Figure 6.44: Diagram of the Mean Absolute Error with respect to the regularization parameter alpha of the model

The figure above shows the mean absolute error of the validation set in relation to the value of the regularization parameter alpha used in our model. We can see that the larger the regularization parameter the lower the Mean Absolute Error metric. The best MAE value is given for a regularization parameter equal to 100.

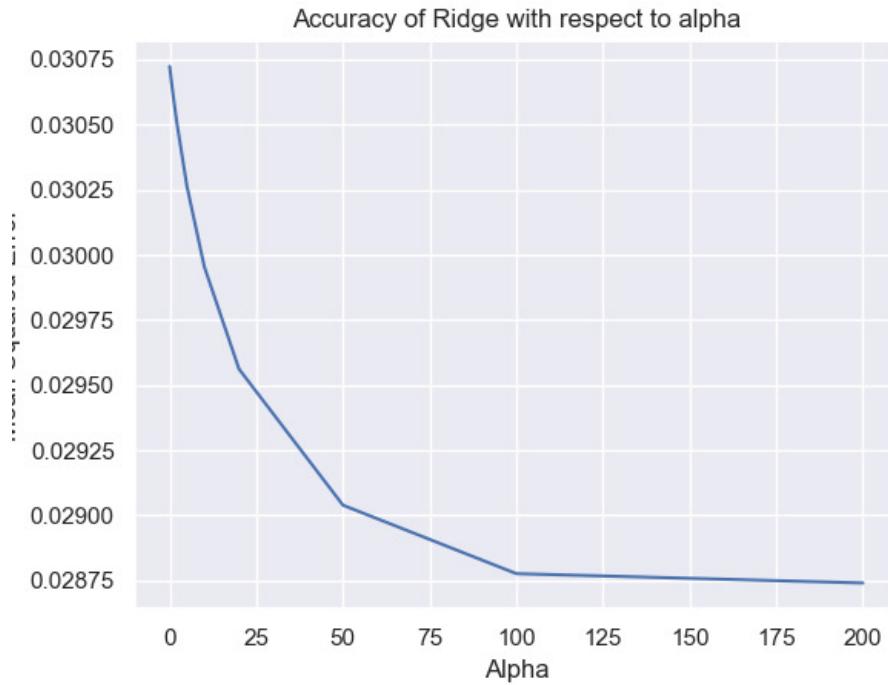


Figure 6.45: Diagram of the Mean Squared Error with respect to the regularization parameter alpha of the model

The figure above shows the mean squared error of the validation set in relation to the value of the regularization parameter alpha used in our model. We can see that the larger the regularization parameter the lower the Mean Squared Error metric.

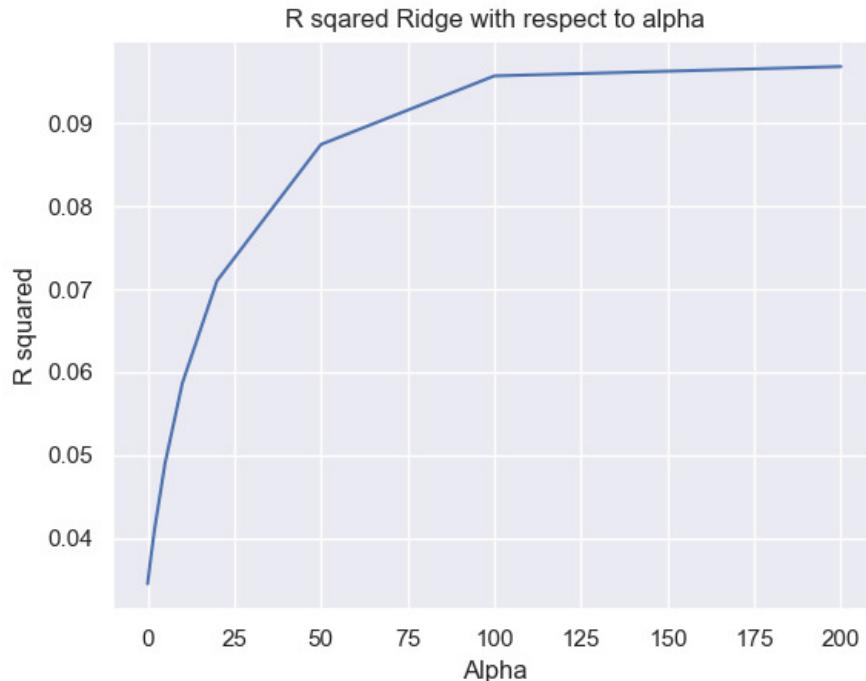


Figure 6.46: Diagram of the  $R^2$  metric with respect to the regularization parameter alpha of the model

The figure above shows the  $R^2$  metric of the validation set in relation to the value of the

regularization parameter alpha used in our model. We can see that the metric is positive this time but at a very low mark of just below 0.1. This means that our model can not explain most of the variance in the model.

### 6.3.4 Regression scores

Following the fine tuning process of each model we tried to fit the data into the models using the best hyper-parameter configuration possible. The following table 6.4 contains the results from each model.

Model	Mean Absolute Error	Mean Squared Error	$R^2$
SVC polynomial kernel	0.1345	0.03	-0.08
Random Forest Regression	0.1284	0.029	0.08
Ridge Regression	0.123	0.029	0.095

Table 6.4: Precision and Recall scores of both crash and evasion cases of each model

We can see that the results are underwhelming. However, the mean absolute error is not very high. It is worth while to see whether or not the data itself has an underlying structure. In figure 6.47 we can see all the patterns the SPoC algorithm extracted. There are a few patterns that look like artefacts because they only appear over one electrode, for example patterns 2 and 7. Others, like pattern 13 and 23, convey a difference in potential between the front and back of the brain. On the same note, patterns like 17 indicate a hemispherical difference in potential.

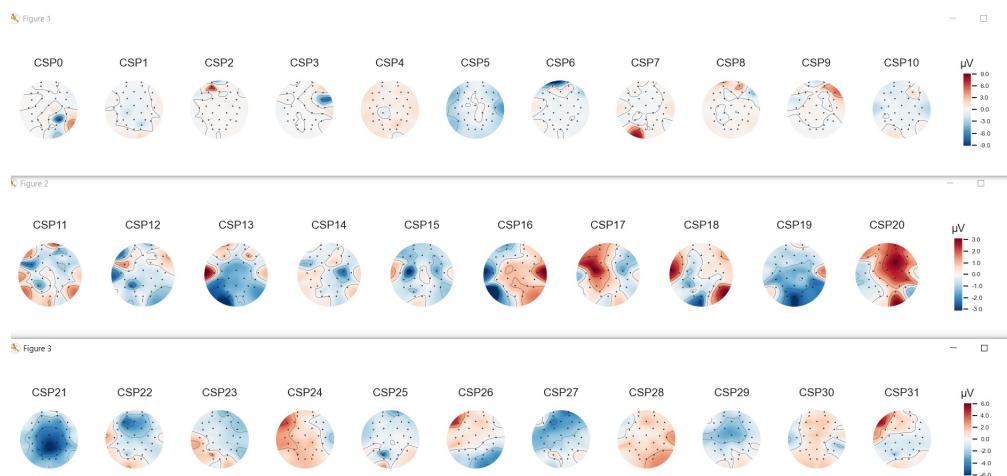


Figure 6.47: Patterns extracted by the SPoC algorithm

Following the previous analysis, we examine 16 of these patterns to see if there exists a linear or non-linear correlation of the patterns to the reaction time.

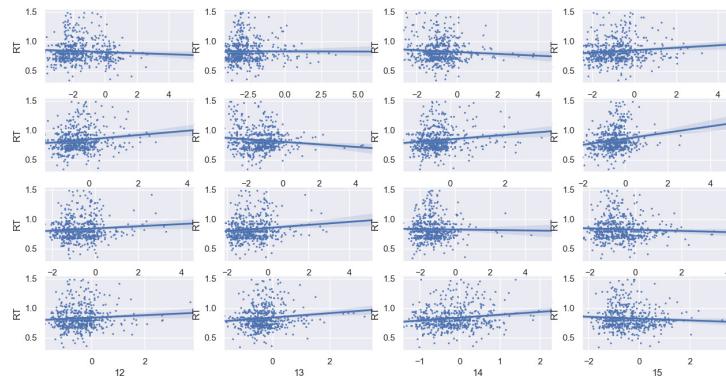


Figure 6.48: Average power of SPoC components correlated to Reaction Time. A line is fitted through every feature to show the best linear fit possible in the data.

We can see an obvious trend among most of these patterns. However, the variance of the samples is large. Most of the testing samples are concentrated in a central ellipsoid with high deviation from its major axis. We can also see that a linear fit is most appropriate since there does not seem to be a non linear correlation between the features and the reaction time.

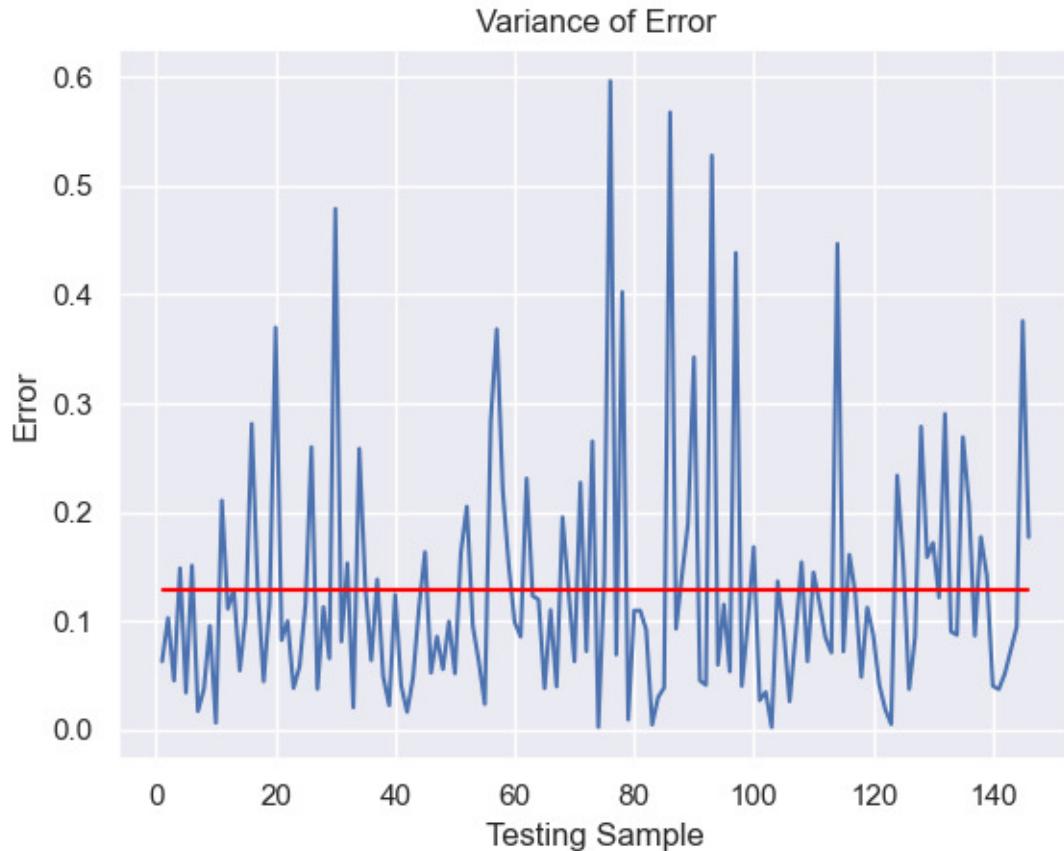


Figure 6.49: Error of the prediction of each testing sample

From figure 6.49 we can see the results of trying to predict a dataset with high variance. The red line indicates the mean absolute error. We can see that there are a lot of predictions very close to the target variable but a sizeable subsample of predictions has a significant error.

# 7. Limitations and Future Work

## 7.1 Uncertainties about participant's attention

One major oversight of the study was the lack of proof regarding the actual attention of the participants. All of the participants were asked questions at the end of the experiment, most of them provided very accurate roundups of the texts they read. But the truth is that we can not be certain regarding their attention at every moment.

A likely scenario would be that the participants started the experiment and as they got tired they would put less and less effort into reading. That means that some of the trials were not conducted under the same conditions as the others. That hypothesis could be detrimental to the processing of the EEG recordings. If we could discern between events in which the participants were not reading, but were actually concentrated on the road, we would have had a more dynamic dataset to work with. Road concentration would be a great default state for the experiment and it could have produced more interesting and even more accurate results for this thesis.

In future work, reorganizing this experiment, we could add a couple of mechanics that would ensure we know exactly where the attention of the participants is called upon. The first mechanism would be an eye tracking system. The eye tracker would track the movement of the eyes of the participant. Looking directly outside of the text for a prolonged period of time would obviously signify us that the participant is paying more attention to the road or the surrounding area, than the actual distraction task. Yet, in case the participants were looking directly at the text, it would not be so simple to conclude that they were paying attention to its contents. They could easily be absent minded at that moment and just mechanistically go over the text, without focusing on understanding it, which could alter the localization of the brain's activity. Combining the eye tracking mechanism with interactive questions during the test could provide us the certainty we need regarding the participant's attention. Instead of asking the participants questions at the end of the experiment, asking them right after they finish a section of the text would ensure us that, with the aid of the eye tracking mechanism, they focused on the text for the last couple of minutes.

## 7.2 Reflex response hypothesis and alternative distractions

During the brain's activity analysis we observed some interesting alterations of the results with respect to our expectations. We mentioned in the Neurological Background chapter that we expected a localized activation of brain function in the Supramarginal and Angular Gyri, which are related to reading. We also mentioned the closeness of the centers that correlate with movement planning and decision making. Considering the widely spread activity we observed from the CSP and SPoC filters and the EEG statistical analysis, we can say that the real distraction might not have been the reading task.

Many participants reported that they had trouble using the Leap Motion Hand Tracking Controller. We observed cases where the hands of the participants disappeared from the simulation, or stuck at a certain spot or generally were not responsive to the actual movement of the participant's hands. During a follow-up questioning, many participants reported that they felt that they crashed more times

when they tried to adjust the reading system rather than when they were focusing their attention in the texts. This leads us to conclude that a dual distraction might have taken place. One important distraction of reading and a heavier one, that of trying to control a user interface. The findings of the filter and statistical analysis are not conclusive on the amount each distraction played a role.

In general, we believe that this did not affect the attention of the participants, it might even have helped to avoid the participants focusing on the road. However, creating two different localized reactions might significantly hurt our model's accuracy. If the participants crashed in both distraction cases with different frequency, we might have different impact from the patterns created by CSP and SPoC. In future work, creating a more user friendly UI, or a distraction task that does not require the constant feedback from the user, like always presenting the text at a specific surface of the car.

Another significant obstacle in the process is the reflex response hypothesis. One of the most unexpected results of this study, was the lack of statistical difference between the crash and evasion reaction times. In order for the reaction time to not play a significant role in this process, we can consider the response from the participants, as requiring less thought and planning. Rather, it can be attributed like a reflex to recurring external stimuli, that brought out the same reaction in order to avoid the obstacle. To further support this hypothesis, the obstacle was created always at the same distance from the car and the track to avoid the obstacle was always to the left lane and back to the right lane. A wider road with more ways to evade the obstacle could be more cognitively demanding from the subject, while combining it with random obstacle distance and car speed would reassure us that the participants had to make a decision at every event.

### 7.3 Furthering the collection of data

Regardless of the aforementioned problems we encountered, the results, especially the classification results, were very promising. We consider the main goal of this study, meaning the proof that there exists a foundation for takeover request effectiveness prediction, was successful. This opens the way for more sensors to be applied to the problem.

First of all, from a behavioural standpoint, collecting data regarding the steering angle and the pedal use could be beneficial to making more educated guesses about the outcome of a takeover request. Additionally, the track of the car could also provide a more sophisticated way to classify the outcome of an event. From a biological sensor standpoint, studying the movement and structural variations of the eyes using an eye tracking system could help us build a system that can actually be used in real life scenarios. Other sensors like respiration tracking, heart rate and temperature measurements could also help make this system even more accurate.

### 7.4 Alternative feature extraction methods

While the CSP and SPoC filters gave us great and sufficient results respectively, we are far from exhausting all our options. Adding statistical information, like the standard deviation of the signals could prove beneficial, as we saw in the EEG statistical analysis. Other methods of feature extraction could also be very useful. Firstly, a pairwise study of the electrodes could provide some more precise source localization data. Complex methods like convolutional neural networks and other deep learning methods, that have been employed more and more in the last years, could increase the accuracy of our models by automating the feature extraction process. Last but not least, trying some other models that specialize in time series machine learning applications, like composition and interval-based regression techniques for the regression problem and shaplet based time series classifiers for the classification task, could significantly improve both the accuracy of our estimations and the explainability of our models.

# Bibliography

- [1] NHTSA. Preliminary statement of policy concerning automated vehicles. [http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated\\_Vehicles\\_Policy.pdf](http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf)., 2013.
- [2] Walter Morales-Alvarez, Oscar Sipele, Régis Léberon, Hadj Hamma Tadjine, and Cristina Olaverri-Monreal. Automated driving: A literature review of the take over request in conditional automation. *Electronics*, 9(12):2087, December 2020.
- [3] Kateřina Bucsuházy, Eva Matuchová, Robert Zůvala, Pavlína Moravcová, Martina Kostíková, and Roman Mikulec. Human factors contributing to the road traffic accident occurrence. *Transportation Research Procedia*, 45:555–561, 2020.
- [4] Vinayak V. Dixit, Sai Chand, and Divya J. Nair. Autonomous vehicles: Disengagements, accidents and reaction times. *PLOS ONE*, 11(12):e0168054, December 2016.
- [5] Francesca M. Favarò, Nazanin Nader, Sky O. Eurich, Michelle Tripp, and Naresh Varadaraju. Examining accident reports involving autonomous vehicles in california. *PLOS ONE*, 12(9):e0184952, September 2017.
- [6] Sol Hee Yoon, Young Woo Kim, and Yong Gu Ji. The effects of takeover request modalities on highly automated car control transitions. *Accident Analysis & Prevention*, 123:150–158, February 2019.
- [7] Na Du, Feng Zhou, Elizabeth M. Pulver, Dawn M. Tilbury, Lionel P. Robert, Anuj K. Pradhan, and X. Jessie Yang. Predicting driver takeover performance in conditionally automated driving. *Accident Analysis & Prevention*, 148:105748, December 2020.
- [8] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):596–614, June 2011.
- [9] L. Angell, J. Auflick, P.A. Austria, D. Kochhar, L. Tijerina, W. Biever, T. Diptiman, J. Hogsett, and S. Kiger. Driver workload metrics task 2 final report. [https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/driver\\_workload\\_metrics\\_final\\_report.pdf](https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/driver_workload_metrics_final_report.pdf), 2006.
- [10] Joanne L. Harbluk, Y. Ian Noy, Patricia L. Trbovich, and Moshe Eizenman. An on-road assessment of cognitive distraction: Impacts on drivers' visual behavior and braking performance. *Accident Analysis & Prevention*, 39(2):372–379, March 2007.
- [11] ESA M. RANTANEN and JOSEPH H. GOLDBERG. The effect of mental workload on the visual field size and shape. *Ergonomics*, 42(6):816–834, June 1999.
- [12] Masahiro Miyaji, Haruki Kawanaka, and Koji Oguri. Driver's cognitive distraction detection using physiological features by the adaboost. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, October 2009.

- [13] Makoto Itoh. Individual differences in effects of secondary cognitive activity during driving on temperature at the nose tip. In 2009 International Conference on Mechatronics and Automation. IEEE, August 2009.
- [14] Avinash Wesley, Dvijesh Shastri, and Ioannis Pavlidis. A novel method to monitor driver's distractions. In CHI '10 Extended Abstracts on Human Factors in Computing Systems. ACM, April 2010.
- [15] Chris Berka, Daniel J Levendowski, Michelle N Lumicao, Alan Yau, Gene Davis, Vladimir T Zivkovic, Richard E Olmstead, Patrice D Tremoulet, and Patrick L Craven. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. Aviat. Space Environ. Med., 78(5 Suppl):B231–44, May 2007.
- [16] Thomas A. Ranney, Transportation Research Center, Inc. Driver distraction: A review of the current state-of-knowledge.
- [17] Huiping Zhou, Makoto Itoh, and Toshiyuki Inagaki. Influence of cognitively distracting activity on driver's eye movement during preparation of changing lanes. In 2008 SICE Annual Conference. IEEE, August 2008.
- [18] Azim Eskandarian, Riaz Sayed, Pierre Delaigue, Jeremy Blum, and Ali Mortazavi . Advanced driver fatigue research.
- [19] Birgitta Thorslund, Anna Anund, Åsa Forsman, Susanne Gustafsson, and Gunnar Soerensen. Electrooculogram analysis and development of a system for defining stages of drowsiness: Master's thesis project in biomedical engineering, reprint from linkoeping university, dept. biomedical engineering, liu-imt-ex-351, linkoeping 2003. 2004.
- [20] J WU and T CHEN. Development of a drowsiness warning system based on the fuzzy logic images analysis. Expert Systems with Applications, 34(2):1556–1561, February 2008.
- [21] Thomas A. Dingus, H.Lenora Hardee, and Walter W. Wierwille. Development of models for on-board detection of driver impairment. Accident Analysis & Prevention, 19(4):271–283, August 1987.
- [22] Zuojin Li, Shengbo Li, Renjie Li, Bo Cheng, and Jinliang Shi. Online detection of driver fatigue using steering wheel angles for real driving conditions. Sensors, 17(3):495, March 2017.
- [23] Guosheng Yang, Yingzi Lin, and Prabir Bhattacharya. A driver fatigue recognition model based on information fusion and dynamic bayesian network. Information Sciences, 180(10):1942–1954, May 2010.
- [24] Chern-Pin Chua, Gary McDarby, and Conor Heneghan. Combined electrocardiogram and photoplethysmogram measurements as an indicator of objective sleepiness. Physiological Measurement, 29(8):857–868, July 2008.
- [25] I.G. Damousis and D. Tzovaras. Fuzzy fusion of eyelid activity indicators for hypovigilance-related accident prediction. IEEE Transactions on Intelligent Transportation Systems, 9(3):491–500, September 2008.
- [26] Venkatesh Balasubramanian and K. Adalarasu. EMG-based analysis of change in muscle activity during simulated driving. Journal of Bodywork and Movement Therapies, 11(2):151–158, April 2007.

- [27] J.A. Healey and R.W. Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, June 2005.
- [28] MN Farid, M Kopf, Heiner Bubb, and Ali El Essaili. Methods to develop a driver observation system used in an active safety system. *VDI BERICHTE*, 1960:639, 01 2006.
- [29] Yan-Jiong Zhong, Li-Ping Du, Kan Zhang, and Xiang-Hong Sun. Localized energy study for analyzing driver fatigue state based on wavelet analysis. In *2007 International Conference on Wavelet Analysis and Pattern Recognition*. IEEE, 2007.
- [30] S. Furugori, N. Yoshizawa, C. Iname, and Y. Miura. Estimation of driver fatigue by pressure distribution on seat in long term driving. 26:053–058, 01 2005.
- [31] T. Wakita, K. Ozawa, C. Miyajima, K. Igarashi, K. Itou, K. Takeda, and F. Itakura. Driver identification using driving behavior signals. In *Proceedings. 2005 IEEE Intelligent Transportation Systems*, 2005. IEEE.
- [32] Rajinda Senaratne, David Hardy, Bill Vanderaa, and Saman Halgamuge. Driver fatigue detection by fusing multiple cues. In *Advances in Neural Networks – ISNN 2007*, pages 801–809. Springer Berlin Heidelberg.
- [33] Luis M. Bergasa, Jose M. Buenaposada, Jesus Nuevo, Pedro Jimenez, and Luis Baumela. Analysing driver's attention level using computer vision. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, October 2008.
- [34] K Hyoki, M Shigeta, Y Nishikawa, Y Kawamuro, and K Soda. The correlation between changes in EEG and eye movement (EOG) accompanied by lowered arousal level. *Rinsho Byori*, 41(7):813–818, July 1993.
- [35] Sadegh Arefnezhad, James Hamet, Arno Eichberger, Matthias Frühwirth, Anja Ischebeck, Ioana Victoria Koglbauer, Maximilian Moser, and Ali Yousefi. Driver drowsiness estimation using EEG signals with a dynamical encoder–decoder modeling framework. *Scientific Reports*, 12(1), February 2022.
- [36] Koichi Fujiwara, Erika Abe, Keisuke Kamata, Chikao Nakayama, Yoko Suzuki, Toshitaka Yamakawa, Toshihiro Hiraoka, Manabu Kano, Yukiyoshi Sumi, Fumi Masuda, Masahiro Matsuo, and Hiroshi Kadotani. Heart rate variability-based driver drowsiness detection and its validation with EEG. *IEEE Transactions on Biomedical Engineering*, 66(6):1769–1778, June 2019.
- [37] Mehmet Akin, Muhammed B. Kurt, Necmettin Sezgin, and Muhittin Bayram. Estimating vigilance level by using EEG and EMG signals. *Neural Computing and Applications*, 17(3):227–236, May 2007.
- [38] Boon-Giin Lee, Boon-Leng Lee, and Wan-Young Chung. Mobile healthcare for automatic driving sleep-onset detection using wavelet-based EEG and respiration signals. *Sensors*, 14(10):17915–17936, September 2014.
- [39] Hanna Damasio. *Human brain anatomy in computerized images*. Oxford university press, 2005.
- [40] Ilaria Decimo, Guido Fumagalli, Valeria Berton, Mauro Krampera, and Francesco Bifari. Meninges: from protective membrane to stem cell niche. *Am. J. Stem Cells*, 1(2):92–105, May 2012.
- [41] Suzana Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Front. Hum. Neurosci.*, 3:31, November 2009.

- [42] Suzana Herculano-Houzel and Roberto Lent. Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *J. Neurosci.*, 25(10):2518–2521, March 2005.
- [43] Paul C Letourneau. Nerve cell shape. In *Cell shape: determinants, regulation, and regulatory role*, pages 247–289. Academic Press San Diego, 1989.
- [44] A. L. HODGKIN and A. F. HUXLEY. Action potentials recorded from inside a nerve fibre. *Nature*, 144(3651):710–711, October 1939.
- [45] Eric R Kandel et al. Nerve cells and behavior. *Principles of neural science*, 3:18–32, 1991.
- [46] James H Schwartz. The transport of substances in nerve cells. *Scientific American*, 242(4):152–171, 1980.
- [47] Palle Juul-Jensen and Richard F Mayer. Threshold stimulation for nerve conduction studies in man. *Archives of neurology*, 15(4):410–419, 1966.
- [48] Ronald MacGregor. *Neural modeling: electrical signal processing in the nervous system*. Springer Science & Business Media, 2012.
- [49] Hai yan He and Hollis T Cline. What is excitation/inhibition and how is it regulated? a case of the elephant and the wisemen. *Journal of Experimental Neuroscience*, 13:117906951985937, January 2019.
- [50] E.R. Kandel and W.A. Spencer. Excitation and inhibition of single pyramidal cells during hippocampal seizure. *Experimental Neurology*, 4(2):162–179, August 1961.
- [51] John Lisman. Excitation, inhibition, local oscillations, or large-scale loops: what causes the symptoms of schizophrenia? *Current Opinion in Neurobiology*, 22(3):537–544, June 2012.
- [52] Vikaas S Sohal and John LR Rubenstein. Excitation-inhibition balance as a framework for investigating mechanisms in neuropsychiatric disorders. *Molecular psychiatry*, 24(9):1248–1257, 2019.
- [53] E.J. Fine. Caton, richard. In *Encyclopedia of the Neurological Sciences*, page 612. Elsevier, 2014.
- [54] Mario Tudor, Lorainne Tudor, and Katarina Ivana Tudor. Hans berger (1873-1941)—the history of electroencephalography. *Acta Med. Croatica*, 59(4):307–313, 2005.
- [55] HERBERT JASPER, BERNARD PERTUISSET, and HERMAN FLANIGIN. Eeg and cortical electrograms in patients with temporal lobe seizures. *AMA Archives of Neurology & Psychiatry*, 65(3):272–290, 1951.
- [56] D Taussig, A Montavont, and J Isnard. Invasive eeg explorations. *Neurophysiologie Clinique/Clinical Neurophysiology*, 45(1):113–119, 2015.
- [57] Kaveh Hassani and Won-Sook Lee. An experimental study on semi-invasive acupuncture-based eeg signal acquisition. In *The 3rd International Winter Conference on Brain-Computer Interface*, pages 1–4. IEEE, 2015.
- [58] Piotr Olejniczak. Neurophysiologic basis of eeg. *Journal of clinical neurophysiology*, 23(3):186–189, 2006.

- [59] Erwin Habibzadeh Tonekabony Shad, Marta Molinas, and Trond Ytterdal. Impedance and noise of passive and active dry eeg electrodes: a review. *IEEE Sensors Journal*, 20(24):14565–14577, 2020.
- [60] Gaetano Gargiulo, Rafael A Calvo, Paolo Bifulco, Mario Cesarelli, Craig Jin, Armin Mohamed, and André van Schaik. A new eeg recording system for passive dry electrodes. *Clinical Neurophysiology*, 121(5):686–693, 2010.
- [61] Ashwati Krishnan, Ritesh Kumar, Praveen Venkatesh, Shawn Kelly, and Pulkit Grover. Low-cost carbon fiber-based conductive silicone sponge eeg electrodes. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1287–1290. IEEE, 2018.
- [62] Borís Burle, Laure Spieser, Clémence Roger, Laurence Casini, Thierry Hasbroucq, and Franck Vidal. Spatial and temporal resolutions of EEG: Is it really black and white? a scalp current density view. *International Journal of Psychophysiology*, 97(3):210–220, September 2015.
- [63] P.L. Nunez, R.B. Silberstein, P.J. Cadusch, R.S. Wijesinghe, A.F. Westdorp, and R. Srinivasan. A theoretical and experimental study of high resolution EEG based on surface laplacians and cortical imaging. *Electroencephalography and Clinical Neurophysiology*, 90(1):40–57, January 1994.
- [64] Michal Teplan et al. Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11, 2002.
- [65] Andrew Morley, Lizzie Hill, and AG Kaditis. 10-20 system eeg placement. *European Respiratory Society, European Respiratory Society*, 2016.
- [66] Robert Oostenveld and Peter Praamstra. The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology*, 112(4):713–719, April 2001.
- [67] Saeid Sanei and Jonathon A Chambers. *EEG signal processing*. John Wiley & Sons, 2013.
- [68] Christopher J. Davis, James M. Clinton, Kathryn A. Jewett, Mark R. Zielinski, and James M. Krueger. Delta wave power: An independent sleep phenotype or epiphenomenon? *Journal of Clinical Sleep Medicine*, 7(5 Suppl), October 2011.
- [69] Daniel L. Schacter. EEG theta waves and psychological phenomena: A review and analysis. *Biological Psychology*, 5(1):47–82, March 1977.
- [70] Gregor Thut and Carlo Miniussi. New insights into rhythmic brain activity from TMS–EEG studies. *Trends in Cognitive Sciences*, 13(4):182–189, April 2009.
- [71] Bjørg Elisabeth Kilavik, Manuel Zaepffel, Andrea Brovelli, William A. MacKay, and Alexa Riehle. The ups and downs of beta oscillations in sensorimotor cortex. *Experimental Neurology*, 245:15–26, July 2013.
- [72] C HERRMANN and T DEMIRALP. Human EEG gamma oscillations in neuropsychiatric disorders. *Clinical Neurophysiology*, 116(12):2719–2733, December 2005.
- [73] Shiang Hu, Dezhong Yao, and Pedro A. Valdes-Sosa. Unified bayesian estimator of EEG reference at infinity: rREST (regularized reference electrode standardization technique). *Frontiers in Neuroscience*, 12, May 2018.
- [74] Mona Sazgar and Michael G Young. Eeg artifacts. In *Absolute epilepsy and EEG rotation review*, pages 149–162. Springer, 2019.

- [75] Gabriele Gratton. Dealing with artifacts: The EOG contamination of the event-related brain potential. *Behavior Research Methods, Instruments, & Computers*, 30(1):44–53, March 1998.
- [76] Agustina Garces, Eric Laciár, Hector Patiño, and Max Valentiniuzzi. Artifact removal from eeg signals using adaptive filters in cascade. *Journal of Physics: Conference Series*, 90:012081, 12 2007.
- [77] Michael I. Posner, Steven E. Petersen, Peter T. Fox, and Marcus E. Raichle. Localization of cognitive operations in the human brain. *Science*, 240(4859):1627–1631, June 1988.
- [78] Serge Weis, Michael Sonnberger, Andreas Dunzinger, Eva Voglmayr, Martin Aichholzer, Raimund Kleiser, and Peter Strasser. Localization of brain function. In *Imaging Brain Diseases*, pages 401–423. Springer Vienna, 2019.
- [79] Ching-Chang Kuo, Phan Luu, Kyle K. Morgan, Mark Dow, Colin Davey, Jasmine Song, Allen D. Malony, and Don M. Tucker. Localizing movement-related primary sensorimotor cortices with multi-band EEG frequency changes and functional MRI. *PLoS ONE*, 9(11):e112103, November 2014.
- [80] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013.
- [81] A design of a low-pass FIR filter using hamming window functions in matlab. *Computer Engineering and Intelligent Systems*, February 2020.
- [82] Nasser Samadzadehaghdam, Bahador Makkabadi, Sadegh Masjoodi, Mohammad Mohammadi, and Fahimeh Mohagheghian. A new linearly constrained minimum variance beamformer for reconstructing EEG sparse sources. *International Journal of Imaging Systems and Technology*, 29(4):686–700, June 2019.
- [83] J. Gross, J. Kujala, M. Hämäläinen, L. Timmermann, A. Schnitzler, and R. Salmelin. Dynamic imaging of coherent sources: Studying neural interactions in the human brain. *Proceedings of the National Academy of Sciences*, 98(2):694–699, January 2001.
- [84] Denis A. Engemann and Alexandre Gramfort. Automated model selection in covariance estimation and spatial whitening of MEG and EEG signals. *NeuroImage*, 108:328–342, March 2015.
- [85] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (PCA). *Computers & Geosciences*, 19(3):303–342, March 1993.
- [86] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. An introduction to independent component analysis. *Journal of Chemometrics*, 14(3):123–149, 2000.
- [87] Pierre Ablin, Jean-Francois Cardoso, and Alexandre Gramfort. Faster independent component analysis by preconditioning with hessian approximations. *IEEE Transactions on Signal Processing*, 66(15):4040–4049, August 2018.
- [88] Sven Dähne, Frank C. Meinecke, Stefan Haufe, Johannes Höhne, Michael Tangermann, Klaus-Robert Müller, and Vadim V. Nikulin. SPoC: A novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters. *NeuroImage*, 86:111–122, February 2014.

- [89] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.
- [90] Leo Breiman. *Machine Learning*, 45(1):5–32, 2001.



