Universidade do Minho
Escola de Engenharia
Departamento de Informática

Raimundo Nonato Barros Neto

Decoding human movement intentions and postural reactions through brain signals

January 2023

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Raimundo Nonato Barros Neto

Decoding human movement intentions and postural reactions through brain signals

Master dissertation
Integrated Master's in Informatics Engineering

Dissertation supervised by
Cristina Manuela Peixoto dos Santos
Luís Ricardo Monteiro Jacinto
Nuno Miguel Ferrete Ribeiro

January 2023

ACKNOWLEDGEMENTS

All the work done over the last year would not have been possible without the assistance, understanding, and support of many people who deserve to be here recognized.

First and foremost, I want to express my gratitude to my supervisor, Prof. Cristina Santos for her availability, knowledge and support that were important for the elaboration of this work.

To my co-supervisor, Prof. Luís Jacinto, for his availability and guidance, which were important for the preparation of this work, mainly in the area of Electroencphalogram.

I'd also like to express my gratitude to my co-supervisor, Nuno Ribeiro, for his encouragement and understanding at difficult times, especially to be always willing to help and support me to achieve my best.

I'm also grateful to everyone in the BiRDLab for the help during the run of the experimental protocol and data acquisition, and mainly to Luís Martins, to share his knowledge regarding the BiRDLab python framework.

I want to express my gratitude to my family, even living far away, they always are present sharing all their love and assistance, not only during my dissertation but throughout my entire life.

For all my friends from Brazil, who lives in Portugal and Portuguese whom I had the pleasure to meet throughout this journey, my sincerely grateful for all your support and motivational words.

Finally, but certainly not least, my wife, Thaís, especially deserves a heartfelt thank you for her support and patience throughout the past year.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity.

I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

Falls are one of the most common causes of injuries in the elderly population. As a result, treatment costs have also increased. Recent efforts to restore lower limb function in these populations have seen an increase in the use of wearable robotic systems, however, fall prevention measures in these systems require early detection of loss of balance to be effective. In short, the development of technologies, such as a brain-computer interface, that is capable of recognizing situations at risk of falling based on the loss of balance caused by several factors, is essential. Previous studies have investigated whether kinematic variables contain information about an impending fall, but few have examined the potential of using electroencephalography (EEG) as a predictor of falling and how the brain responds to prevent a fall. Perceived disturbances of balance are always accompanied by a specific cortical activation, called disturbance-evoked potential (PEP).

In this study, the recognition of daily activities (walking, lifting, crouching, going up and down stairs) was also part of the initial objective, however, due to the challenges encountered, the object of study of the present work was focused on the recognition and binary classification of the presence of loss of balance (PEPs) in brain signals. Thus, this dissertation intends to take the first steps toward the decoding of brain activity in response to imbalanced events. Initially, to acquire the data, an experimental protocol was designed, so that the participants, using EEG, were submitted to gliding-like perturbations while walking on the treadmill. Two healthy subjects were exposed to a glide-like perturbation, and these perturbations occurred interspersed over a period lasting 30 to 60 seconds. Each subject performed 2 experiments, that is, perturbations provoked while the individual walked on the treadmill: i) at a speed of 1.6 km/h and ii) at a speed of 2.5 km/h.

Based on the approached methods, the perturbation evoked potential (PEP) components were found between 70-155 ms after the onset of the external perturbation. To decode pre-processed EGG data, four (4) artificial neural networks were tested and different network architecture parameters and electrode layouts were compared. Overall, the convolutional neural network trained to predict EEG balance disturbances had a far superior classification performance than the other architectures, whose mean accuracy was 91.51 $\pm$ 2.91%, using a short window length of 200 ms. The electrode layout composed of 5 channels (Fz, C3, Cz, C4, and Pz) presented the shortest execution time to train the model, whose average value was 196 $\pm$ 44.24ms. In addition, it was possible to verify that the use of a single electrode (Cz) obtained satisfactory precision results (86.47 +/- 0.03%). These discoveries may contribute to the development of a system capable of detecting equilibrium disturbances in real-time.

KEYWORDS Brain-computer interface, Electroencephalogram, Falls recognition, Pertubation-Evoked Potential Deep learning.

RESUMO

As quedas são uma das causas mais comuns de lesões na população idosa. Como resultado, custos com o tratamento têm também aumentado. Esforços recentes para restaurar a função dos membros inferiores nessas populações viram um aumento no uso de sistemas robóticos vestíveis, no entanto, as medidas de prevenção de quedas nesses sistemas exigem a detecção precoce da perda de equilíbrio para serem eficazes. Em suma, é fundamental o desenvolvimento de tecnologias, tal como interface cérebro-computador, que seja capaz de reconhecer situações risco de queda com base na perda de equilíbrio ocasionada por diversos fatores. Estudos anteriores investigaram se as variáveis cinemáticas continham informações sobre uma queda iminente, mas poucos examinaram o potencial do uso da eletroencefalografia (EEG) como um sinal de previsão de queda e como o cérebro responde para evitar uma queda. As perturbações do equilíbrio percebidas são sempre acompanhadas por uma ativação cortical específica, chamada de potencial evocado por perturbação (PEP).

Neste estudo, também fazia parte do objetivo inicial o reconhecimento das atividades diárias (andar, levantar, agachar, subir e descer escada), porém, devido aos desafios encontrados, o objeto de estudo do presente trabalho foi focado no reconhecimento e classificação binária da presença de perda de equilíbrio (PEPs) em sinais cerebrais. Assim, esta dissertação pretende dar os primeiros passos em direção a decodificação da atividade cerebral em resposta a eventos de desequilíbrio. Inicialmente, para adquirir os dados, um protocolo experimental foi delineado, de forma que os participantes, usando EEG, fossem submetidos a perturbações do tipo deslizamento enquanto andavam sobre a esteira. Dois sujeitos saudáveis foram expostos a perturbação do tipo deslizamento, e estas perturbações ocorreram intercaladamente em um período de duração de 30 a 60 segundos. Cada sujeitou realizou 2 experimentos, ou seja, perturbações provocadas enquanto o indivíduo andava na passadeira: i) a uma velocidade de 1.6 km/h e ii) a uma velocidade de 2.5 km/h.

Com base nos métodos abordados os componentes de potencial evocado de perturbação (PEP) foram encontrados entre 70-155 ms após o início da perturbação externa. Para decodificar dados pré-processados do EGG, quatro (4) redes neurais artificiais foram testados e diferentes parâmetros da arquitetura de rede e layouts de eletrodos foram comparados. No geral, a rede neural convolucional treinada para prever as perturbações do equilíbrio do EEG teve um alcance desempenho de classificação superior as demais arquiteturas, cuja precisão média foi de 91,51 $\pm$ 2,91%, usando um comprimento de janela curto de 200 ms. O layout de eletrodo composto por 5 canais (Fz, C3, Cz, C4 e Pz) apresentou o menor tempo de execução para treinar o modelo, cujo valor médio foi 196 $\pm$ 44.24ms. Além disso, foi possível verificar que a utilização de um único eletrodo (Cz) obteve resultados de precisão satisfatórios (86.47 +/- 0.03%). Essas descobertas podem contribuir para o desenvolvimento de um sistema capaz de detectar perturbações do equilíbrio em tempo real.

PALAVRAS-CHAVE Interface cérebro-computador, Eletroencefalograma, Reconhecimento de queda, Potencial Evocado de Perturbação, Aprendizagem profunda.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACC Anterior Cingulate Cortex. 43
ADL Activities of Daily Living. 7, 9, 10, 29, 30, 41, 49, 51, 56, 81
AI Artificial Intelligence. 30, 31, 47, 48, 50, 51, 61, 79
AM Amplitude Modulated. 45
ANN Artificial Neural Networks. 9, 26, 65
ASR Artifact Subspace Reconstruction. 23, 24, 39

BCI Brain Computer Interface. 6, 8, 9, 10, 12, 13, 14, 15, 16, 18, 23, 25, 26, 28, 29, 30, 31, 41, 43, 44, 45, 48, 50
BF Biceps Femoris. 37
BiRDLab Biomedical Robotic Devices Lab. 6, 50, 51, 52, 53

CAR Common Average Reference. 23, 24, 38, 56, 73
CCA Canonical Correlation Analysis. 44
CLDA Closed-Loop Decoder Adaptation. 41
CMEMS Center for Micro-Electro-Mechanical Systems. 6
CNN Convolutional Neural Network. 9, 26, 42, 43, 46, 63, 73, 76, 77, 80, 81

DL Deep Learning. 6, 10, 11, 26, 30, 50, 62, 67, 73, 74

ECG Electrocardiographic. 13, 16, 18
EEG Electroencephalogram. 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 23, 24, 25, 26, 28, 29, 30, 31, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 55, 56, 57, 60, 61, 62, 77, 78, 80
EKG Electrocardiographic. 13
EMG Electromyography. 13, 16, 18, 23, 37, 80, 81
EOG Electrooculogram. 13, 16, 36, 37, 41
ERD Event-Related Desynchronization. 39, 42, 43
ERP Event-Related Potential. 8, 17, 77, 78, 79
ERS Event-Related Synchronization. 39

FDA Food and Drug Administration. 7
FIR Finite Impulse Response. 37
fMRI functional Magnetic Resonance Imaging. 8, 14
FN False Negative. 69
fNIRS functional Near-Infrared Spectroscopy. 8
FP False Positive. 69

GL Gastrocnemius Lateralis. 37
GM Gastroc Medialis. 37
GND Ground. 36
GRU Gated Recurrent Units. 9, 27, 63, 73, 74, 78, 80, 81

hBCI hybrid Brain Computer Interface. 13
HMI Human-Machine Interaction. 8

IC Independent Components. 24, 38, 39, 72, 74, 77, 78, 80
ICA Independent Component Analysis. 23, 24, 38, 39, 56, 74, 81
IPL Inferior Parietal Lobe. 43

LOB Loss of Balance. 53, 61
LSTM Long Short Term Memory. 9, 27, 42, 63, 78, 80, 81

MCC Matthews correlation coefficient. 71, 79
ME Motor Exection. 41
MEG Magnetoencephalography. 14
MI Motor Imagery. 16, 28
ML Machine Learning. 30
MLM Mode Labeling Method. 64
MMP Movement Monitoring Potential. 41
MRCP Movement Related Cortical Potentials. 41, 42, 46

N True Negative. 69

P Positive. 69
pBCI Passive Brain Computer Interface. 29
PCA Principal Component Analysis. 24
PEP Perturbation-Evoked Potential. 7, 11, 17, 18, 19, 21, 22, 23, 49, 56, 57, 58, 60
PET Positron Emission Tomography. 8, 14
PPC Posterior Parietal Cortex. 43

REF Reference. 36
RELICA Reliable Independent Component Analysis. 39
RF Rectus Femoris. 37
RNN Recurrent Neural Network. 26, 27
RP Readiness Potential. 41

SNR Signal-to-Noise Ratio. 13, 15
SSAEP Steady-State Auditory Evoked Potential. 16
SSEP Steady-State Evoked Potential. 16

SSSEP Steady-State Somatosensory Evoked Potential. 16
SSVEP Steady-State Visually Evoked Potential. 16, 28
STD Standard Deviation. 24
STG Superior Temporal Gyrus. 43

TA Tibialis Anterior. 37
TN True Negative. 69
TP True Positive. 69

UKF Unscented Kalman Filter. 41, 43

VL Vastus Lateralis. 37
VM Vastus Medialis. 37

WHO World Health Organization. 6

1

INTRODUCTION

This dissertation presents the work developed during my Masters's Degree in Informatic Engineering, at the Biomedical Robotic Devices Lab (BiRDLab) included in the Center for Micro-Electro-Mechanical Systems (CMEMS), a research center of the Department of Industrial Electronics in the University of Minho.

The purpose of this dissertation was to investigate the decoding of human brain signals in response to slip-like perturbations. The data that supports this work was acquired with electroencephalography (EEG) recordings of subjects who were submitted to slip-like perturbations while walking on a treadmill at speed of 1.6km/h and 2.5km/h. The work developed represents a step towards a Brain-Computer Interface (BCI) device useful to promptly detect gait disturbances, possibly giving more time margin for fall prevention mechanisms to act and be more effective. This strategy aims to help seniors improve their quality of life, through a BCI that reads brain signals and translates them into commands that will be relayed to the device when it detects a fall event.

The project was divided into 3 main phases, namely: **i)** research and design of EEG data acquisition protocols during balance loss experiments in a laboratory setting; **ii)** research and development of data pre-processing framework using EEGLAB to remove artifacts from the data and extract relevant features; and **iii)** research, development, and validation of AI-based Deep Learning (DL) models for the detection of balance loss using EEG data collected and preprocessed from experiments.

1.1 MOTIVATION

The aging process has a wide-reaching impact on the human body and brain. It can affect various aspects of daily life, such as the development of new tissues, the maintenance of mental health, and the activities of daily living. Older adults may face difficulties in communicating, focusing, remembering, speaking, moving, or maintaining balance. Thus, the elder may find it challenging to interact with family members, use stairs securely, remember new information, or drive safely due to these impairments [30].

The World Health Organization (WHO) [50] estimates that over the course of a year, at least one-third of the elderly population suffers a fall, provoked by cognitive, physical, and sensory deficits which arise with aging. Accidental falls are one of the leading causes of injuries and the second leading cause of death related to unintentional injuries.

The negative psychological and social repercussions of falls include not fulfilling daily tasks due to constant threat by the unpredictability of fall risk events and practicing fewer physical activities, which leads to a lower

quality of life [2]. As a result, there is an urgent need to create fall prevention and prediction technologies in order to reduce the individual and societal burden of falls.

Conventional therapies such as exercising, managing medications, having vision checked, and making home safer homes have been successful in fall reduction and prevention, however many individuals with severe illness or injury remain unable to participate in activities of daily living (ADL) or complete standard care protocols [35]. Due to the need to improve the recovery of the locomotion impaired, robotic-assisted devices such as prostheses, electrical stimulators, and exoskeletons have been developed. Additionally, similar types of devices can be also applied for fall prevention purposes [7].

Powered exoskeletons are categorized by the U.S. Food and Drug Administration (FDA) as Class 2 medical devices subject to additional regulations. They are commonly utilized in rehabilitation applications since they may offer active support for sitting, standing, and walking. Exoskeletons can be used for rehabilitation as well as to lower the risk of falling and/or help avoid falls [35]. However, falls while wearing exoskeletons pose a substantial risk while using these technologies. Current exoskeletons approved by the FDA employ a variety of tactics to prevent falls and are intended for use in conjunction with a trained partner. The efficiency of these approaches has not been researched and is currently unknown [35].

Near-fall detection could provide new opportunities to identify older people at high risk of falling before a fall occurs. Near falls are defined as trips, slips, and missteps and involve a loss of balance that does not result in a fall because corrective action is taken to recover balance. Near falls occur more frequently than actual falls, and older people who frequently experience near falls are at increased risk of future falls [20]. Therefore, for patients to have a natural and effective recovery, they must fully control the devices they use, such as exoskeletons, prostheses, orthoses, and robotic canes.

1.2 PROBLEM STATEMENT

It is suggested that human bipedal balance control is the combined activity of dispersed brain regions. However, while there is evidence linking the cortical activity of specific brain regions, such as the motor cortex, to reactive balance regulation, less is known about the functional interplay of additional cortical regions [49].

Electroencephalography signals recorded from the scalp reflect synchronous activation of cortical neurons [48]. During active walking, robot-assisted passive walking, and walking with interactive feedback in a virtual environment, EEG recordings revealed activation of electrocortical sources distributed across different cortical regions including the anterior cingulate, prefrontal cortex, pre-motor cortex, supplementary motor area, posterior parietal, and sensorimotor cortex [49].

Recent research shows that brain involvement occurs during balance reactions induced by whole-body postural disturbances. An external disturbance evokes an evoked potential in humans, known as the perturbation-evoked potential (PEP). PEP are broadly dispersed throughout the fronto-centro-parietal regions, with the greatest amplitude occurring at the FCz/Cz electrodes [48]. Specifically, PEP can be used to study sensory and cortical control of postural responses.

BCI based on EEG technology has been employed in this context to develop portable synchronous and asynchronous control and communication. Noninvasive EEG-based BCI can be classified as "evoked" (reactive BCI), "spontaneous" (active BCI), or passive BCI. An evoked BCI takes use of a major feature of the EEG, the so-called evoked potential, which displays the brain's instantaneous automatic reactions to some external stimuli. On the other hand, spontaneous BCI is based on the investigation of EEG events linked with various elements of brain activity connected to mental tasks performed at the user's discretion [30]. Finally, the concept of passive BCI has recently been developed in the BCI area, in which the system takes action to perform a command based on the random brain activity of the user while the participant lacks voluntary control over the system [41]. The use of EEG signal may increase the applicability of individualized brain-controlled rehabilitation equipment. Individually tailored parameters for describing signals are generated from many training trials in the development of EEG-based BCI, which are then utilized to perform online signal decoding [25].

As a result, EEG might reveal the cortical activations associated with loss of balance or balance regulation. Furthermore, when compared to other neuroimaging techniques such as Positron Emission Tomography (PET), functional Magnetic Resonance Imaging (fMRI), or functional Near-Infrared Spectroscopy (fNIRS), EEG provides superior temporal resolution (1 ms), allowing researchers to precisely time-lock neural activity relative to perturbation onset or subsequent movements in order to extract perturbation-related or response-related cortical activations [48].

One disadvantage of utilizing EEG is that it measures the total of electrical potentials from several sources of cerebral activity that correlate to the distinct sensory, motor, or cognitive events. Nevertheless, it is possible to extract Event-Related Potentials (ERP) associated with specific occurrences by averaging across EEG epochs [48].

The ability to predict human movement intention is critical for successful gait restoration. Brain waves are captured, processed, and interpreted to operate an assistive device in a BCI-based rehabilitation system. It is vital for a successful assistive system to detect movement intention as early as possible to allow the system adequate time to adjust to the individual's needs [20].

Assistive, adaptive, and rehabilitative BCI applications for older adults and elderly patients should be created to help with household duties, strengthen connections with family members, and improve cognitive and motor abilities. Many BCI applications have been developed in the last decade to assist the elderly keeping them healthy, with high quality of life, and a feeling of well-being [30].

The main goal of research in the field of Brain-computer interfaces is to gain control of a computer or machine purely by utilizing users' thoughts. While there has been a major emphasis on utilizing BCI to operate assistive devices, BCI can improve Human-Machine Interaction (HMI) [10]. Thus, an effective algorithm for detecting near falls based on ata can contribute to the development of fall detection systems by improving their performance, time response, and quality in terms of accuracy.

1.3 DISSERTATION GOALS AND RESEARCH QUESTIONS

The present work aims to decode events of imbalance in human gait from the collection of brain signals through the use of the EEG. A protocol for data collection was designed based on good practices found in the scientific literature. The data were collected with different subjects, wearing an EEG cap with 16 channels and submitted to slip-like perturbations while walking on a treadmill in a controlled environment. EEG data were processed to create relevant features, through feature engineering, which were then used to detect loss of balance during walking on a treadmill through the implementation of artificial intelligence models. Because it was necessary to identify the most used pre-processing and artifact removal methods and algorithms, as well as identify the most relevant brain rhythms. Furthermore, the performance of different Artificial Neural Network (ANN) models, such as Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Gated Recurrent Units (GRU), among others, were evaluated and compared in order to find a benchmark model that can identify the perturbations, being robust enough to be implemented and integrated into different robotic platforms.

The present dissertation focused on pursuing the following towards the outlined solution:

- **Goal 1**: Gather information about human brain signals in response to ADL activities and loss of balance, by performing a state-of-the-art review, and understanding the most effective and significant paradigms, techniques, and devices. This will be presented in Chapter 2. Key Performance Indicators (KPIs): i) identify BCI technologies applied in the real world; ii) EEG signals related to BCI application; iii) EEG variables (namely latency, peak amplitude, and time to peak) regarding brain signals in response to slip events evidenced in literature [48];

- **Goal 2**: Gather knowledge about methods used currently for decoding ADL activities, from data acquisition to Artificial Intelligence models by performing a literature review. This method will be described in Chapter 3. KPIs: i) systematic review for article published; ii) data pre-processed tools used to extract relevant information regarding ADLs; iii) Artificial Neural Network architecture applied to decode brain signal.

- **Goal 3**: Delineate protocols to obtain data from multiple subjects in the laboratory, while performing slip-like perturbations wearing the EEG. This was then used to train and evaluate the best AI-based model. This topic will be addressed in Chapter 4. KPIs: i) definition of an experimental protocol for data acquisition.

- **Goal 4**: Data analysis to identify patterns in the EEG signals and channels most related to the loss of balance caused by slip-like perturbations in different subjects walking on treadmill at different speeds. - Chapter 5. KPIs: i) identify EEG channels where a pattern can be found; ii) detection of signals components in response to slip-like perturbations which positive potential (P1) that peaks around 30–90 ms after perturbation onset and a negative potential (N1) that peaks around 90–160 ms [48].

- **Goal 5**: Exploit ANN models capable of recognizing slip-perturbation from a dataset with data collected during trials performed in a laboratory setting. Chapter 6 shows the obtained results. KPIs: i) obtain an average accuracy higher than 87.6% $\pm$ 4.2%, and F-Score higher than 87.2% $\pm$ 5 [35]; ii) obtain accuracy above 80% for minimal electrode layout (1 channel) [10].

Consequently, in this dissertation the following Research Questions are expected to be answered:

- **RQ 1**: What are the most relevant signals and features that can help recognize the loss of balance provoked by slip-like perturbations? The answer is included in Chapter 5.
- **RQ 2**: What are the most relevant channels to decode the slip-like perturbations during walking? The answer is included in Chapters 5 and 3.
- **RQ 3**: What is the best Deep Learning model to implement for the detection of slip-like perturbation during walking? The answer is included in Chapter 3.

1.4 SCIENTIFIC CONTRIBUTIONS

The main contributions of this dissertation to the current state-of-the-art in this field are:

- A systematic literature review of brain signals decoding related to daily activities and loss of balance provoked by slip-like perturbations.
- Validation of data pre-processing methodologies described in the literature for slip-perturbation during walking on a treadmill.
- Creation of a segmented and feature-selected dataset from the raw data collected along slip-perturbation experiments performed in the laboratory, which allows more robust and validated training and testing of AI-based models.
- Creation of Deep Learning classification algorithms capable of detecting loss of balance provoked by slip-like perturbations.
- Selection through computational performance analysis, of the most relevant features and DL models to be tested in a real-time application.

1.5 DISSERTATION OUTLINE

The remainder of this dissertation is organized in 6 chapters:

Chapter 2 contains a review of the literature regarding brain-computer interfaces, which devices are used for data acquisition with a focus on EEG signal and its paradigms. In addition, in this chapter data pre-processing techniques and the AI-based models applied for the decoding brain signal in general BCI applications are presented.

Chapter 3 includes a start-of-the-art review about decoding human brain signals strategies to identify different ADL and induced loss of balance in a static position. This analysis will include the experimental protocol previously developed to collect the data, steps related to the data pre-processing, classification applying Deep Learning algorithms, and discussion about the results obtained from the review.

In Chapter 4, the project conceptual design of the dissertation is introduced and shows the dissertation's main phases.

In Chapter 5, the methodology used to obtain the PEP signals involved in the trials developed at EEGLAB will be presented. This Chapter also shows the results which will be addressed in the next chapter.

Chapter 6 demonstrates the process of developing and evaluating DL-based models for slip-like perturbations, as well as selecting the classifier that best fits the loss balance recognition.

Finally, Chapter 7 concludes the dissertation, presenting a short analysis of the work developed as well as the answers to the RQs. In addition, it includes the directions for future research improvements to be developed.

2

BRAIN COMPUTER INTERFACES AND DECODING BRAIN SIGNALS

The study of the neurophysiological basis underlying many motor function abnormalities, such as difficulty regulating limbs or movement limits, has piqued the public's attention in recent decades. This research has made substantial contributions to various domains, including neurorehabilitation and the method known as the brain-computer interface. BCI are instruments that allow the translation of brain electrical activity into commands that operate equipment such as computers, wheelchairs, prostheses, orthoses, or exoskeletons. According to Mercado et al. [27], a BCI is defined by three key elements: i) a method for acquiring brain neural signals; ii) a computer algorithm for decoding the collected signals; and iii) the device to be controlled. BCIs' primary purpose is to understand the user's intents in order to regulate assistive devices. BCI advancements have demonstrated the capability of deciphering motion intents or trajectories from EEG data, which is beneficial for neurological gait rehabilitation. The description of the hardware and characteristics of the brain signals related to the processing of data acquired from the EEG for BCI applications will be addressed in Section 2.1. The components necessary to build a BCI system which translates brain signals into commands to control devices will be addressed in Section 2.2. In the end, Section 2.3 presents a brief discussion regarding BCI applications and the importance of decoding brain signals that comprehensively characterize the response to slip-like perturbations.

2.1 EEG-BASED BRAIN COMPUTER INTERFACES

Brain-computer interface system typically consists of five main sequential steps: brain activity measurement, preprocessing, feature extraction, classification, and translation into a command. Figure 1 shows a typical block diagram illustrating the various phases of EEG signal processing for BCI. The brain activity of the user is recorded using several types of EEG sensors during the brain activity acquisition phase. The raw EEG data contains a number of artifacts due to muscular or ocular movement, which is removed during the pre-processing step. The goal of feature extraction is to describe the signals using a few important values known as features. Frequently, the selection of key features is also examined at this step. In the classification step, the retrieved characteristics are classified using various machine learning and deep learning techniques. Finally, the categorized outputs are converted into device commands in order to create a real-world BCI application.

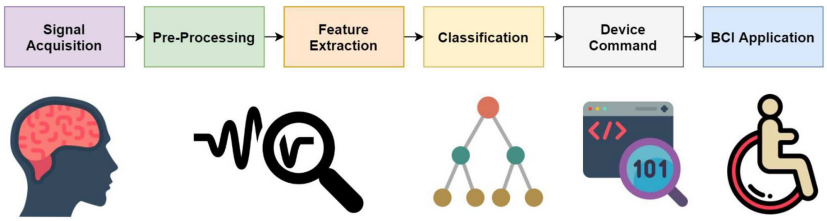


Figure 1: General architecture of a brain-computer interface.Taken from [26]

2.1.1 *BCI Features and Resources*

As previously stated, BCI entails essentially converting human brain activity into external action by delivering neural commands to external equipment. Although BCI is most commonly used to assist handicapped persons with motor system diseases, it may also be a highly beneficial tool for enhancing the quality of life of healthy people, particularly the elderly. Assistive, adaptive, and rehabilitative BCI applications for older persons and elderly patients should be created to help with household duties, strengthen connections with family members, and improve cognitive and motor abilities.

According to Palmer et al., [30] there are several essential considerations to consider while developing a ready-to-use BCI solution. The major difficulties include low classification accuracy, a limited number of degrees of freedom, and a lengthy training period to learn how to use a BCI flawlessly. As a result, researchers have been attempting to improve BCIs by creating a hybrid BCI (hBCI) that integrates at least two BCI modalities. The hBCI combines various methodologies to make use of the benefits of numerous BCI modalities. It can also be a mix of brain activity and nonbrain activity, and a variety of different psychological signals have been demonstrated to be a potential choice for hBCI development. As a result, the input signals can be a combination of EEG with eye movements (Electrooculogram - EOG), muscular activity (Electromyography, EMG), or heart signal (Electrocardiogram, ECG or EKG) [30].

Furthermore, a closed-loop BCI system that uses visual and proprioceptive feedback with real-time modulation and communication can be used not only for interacting with the external environment but also as a biofeedback platform to improve the cognitive abilities of elderly patients and provide better therapeutic effects. This closed-loop interaction between the participant's brain reactions and the stimuli is hypothesized to produce cerebral plasticity and so aid recovery [30].

The creation of less invasive or non-invasive devices is one of the most difficult tasks in BCI technology. Using non-invasive equipment can significantly lower both the overall cost of the surgical procedure and the patient's physical damage. Non-invasive approaches, on the other hand, might result in weaker signals and a poor signal-to-noise ratio (SNR), as well as reduced source inference accuracy and spatial resolution. These disadvantages can be mitigated in part by using modern technologies like deep learning to interpret and extract more meaningful source information from the EEG signal [30].

2.1.2 *Electroencephalography*

Researchers have demonstrated the feasibility of controlling exoskeletons and neuroprosthetic devices using non-invasive EEG, which collects and records brain electrical activity from electrodes placed on the scalp with amplitudes in the microvolt range in real time [25, 12]. Because of the distance and impedance of bone and skin between the electrodes and the cerebral cortex, the EEG cannot accurately detect single-neuron action potentials. Instead, the EEG picks up local current flows on groups of active neurons within the cerebral cortex.

Other imaging techniques, such as fMRI, magnetoencephalography (MEG), and PET, are extremely expensive, technically difficult, and not easily portable. Thus, EEG is still regarded as the only practical and realistic non-invasive BCI method at present [25].

EEG-based BCIs have been mentioned in several recent reviews used to control assistive devices, for upper or lower limbs, because EEG signals have shown the involvement of the sensorimotor cortex in the control of human gait and also revealed differences in electrocortical activities between uphill and level ground walking in humans, as well as active versus passive walking in a robotic device [27, 38].

According to Hasan et al. [16] due to the fact EEG data may be used to reflect the brain's motor activity during locomotion in real time, EEG-based gait studies have a high potential for predicting future movement intentions. In Luu et al. [46] studies EEG recordings during walking exhibit different characteristics that 1) differ from calm standing EEG and 2) are gait phase dependent. These findings suggest that EEG-based BCI that predicts user movement intention might be developed.

2.1.2.1 *EEG Scheme and Hardware for BCI Application*

According to Rashid et al. [26] the cerebral cortex and subcortical regions are the two primary components of the human brain. The subcortical areas manage fundamental and vital activities such as body temperature, breathing, heart rate, and emotional reactions such as reflexes, fear, learning, and memory. The cerebral cortex, also known as the cerebrum, on the other hand, governs sensory and motor processing as well as higher-level activities like language processing, pattern recognition, thinking, and planning. The cerebral cortex is divided into two hemispheres, each of which is divided into four lobes: the parietal, occipital, frontal, and temporal lobes.

The parietal lobe controls several functions, including spelling, objects, manipulation, perception, and spatial awareness. The temporal lobe, on the other hand, is responsible for language, memory, facial recognition, and emotion generation. The frontal lobe is responsible for organizing, social skills, planning, flexible thinking, problem-solving, conscious movement, attention, and emotional and behavioral control. The occipital lobe is involved in the interpretation of visual inputs.

Measurements from all electrodes need a correct EEG electrode placement to ensure the proper location of electrodes in relation to cortical areas so that they can be reliably and precisely maintained from individual to individual. Thus, the International 10/20 system is a widely accepted way of indicating the position of electrodes on the scalp. The system is reliant on the link between the electrode placement and the underlying cerebral cortex region. The numerals 10 and 20 denote that the distances between neighboring electrodes are either 10% or 20% of the entire front-back or right-left distance of the skull, respectively. The lobe is represented by a letter at

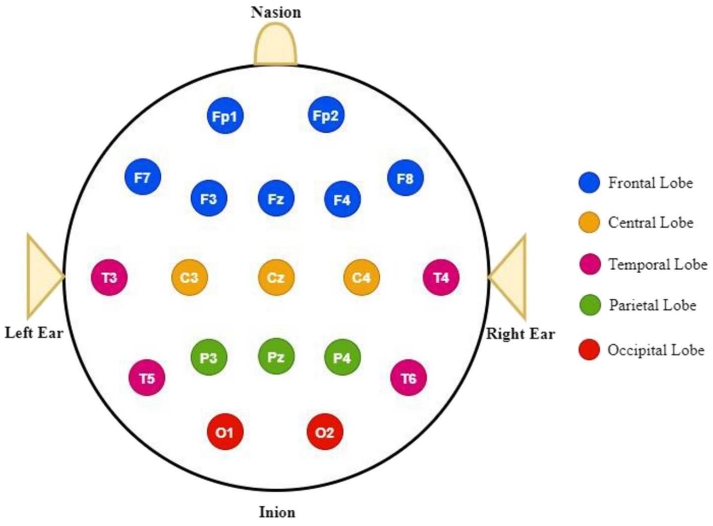


Figure 2: Example of Electrode Placement Scheme. Taken from [26]

each position, whereas the hemisphere location is represented by a number. The frontal, parietal, temporal, and occipital lobes of the 10/20 system are represented by the letters F, P, T, and O, respectively as shown in Figure 2. The central lobe is not included; the letter C is just used for identification. The letter z (zero) indicates that an electrode is positioned on the midline. Even numbers (2, 4, 6, 8) are used to represent right hemisphere electrode placements, whereas odd numbers (1, 3, 5, 7) represent left hemisphere electrode positions.

EEG signals can be acquired in two ways: wirelessly or wired. EEG signal measurements are typically achieved with a number of electrodes ranging from 1 to around 256. These electrodes are typically held in place by an elastic cover. A conductive gel or paste is routinely used to improve contact between the electrodes and the skin. As a result, placing the electrodes on the head is often arduous and time-consuming. Nonetheless, the use of dry electrodes that do not require conductive gels or pastes has been demonstrated [26].

Wireless BCI systems have received adequate attention, owing mostly to their capacity to circumvent some constraints found in wired systems, such as the use of cables between electrodes and the acquisition component, which is sometimes difficult and time-consuming. The fact that a wireless EEG headgear is noninvasive is one of its advantages. Furthermore, it does not obstruct the user's movement. Nevertheless, its main disadvantage is in relation to the range of the signal between the transmitter and the receiver, because depending on the environment and activity that is carried out, the signal may suffer a loss due to the material and thickness of the walls.

2.1.2.2 *Noise and Artifacts*

A low SNR, which is a commonly used measure of fidelity in physical systems (it is defined as the ratio of a signal's squared amplitude or variance relative to the variance of the noise), and different noise sources are among the most difficult challenges in EEG-based BCI application studies. Unwanted signals in the primary transmission are referred to as noise, artifacts, or interference. EEG artifacts can come from two sources: external or environmental factors and physiological ones. AC power lines, lights, and a wide range of electronic devices

are examples of external noise sources (from computers, displays, and TVs to wireless routers, notebooks, and mobile phones, amongst others). Physiological noise is caused by a variety of physiological activities such as movement, other bioelectrical potentials, or changes in skin resistance. Electrooculographic activity (EOG, eye), Electrocardiographic activity (ECG, heart), scalp-recorded Electromyographic activity (EMG, muscle), ballistocardiography activity (heart-related pulsatile motion), and breathing are the most common physiological noises.

2.1.3 *EEG Control Signals*

2.1.3.1 *Motor Imagery*

Motor imagery (MI) can be defined as the mental simulation or rehearsal of an action without any movements and motor output [15]. The alpha and beta brain waves are the most significant frequency ranges for motor imagery. Left and right hand MI evoked activity is detected on the C3 and C4 regions of the brain, respectively, whereas foot movement imagery is generated by Cz. Due to the close proximity of the relevant brain areas, left and right foot motions are difficult to differentiate in EEG. The cortical regions of the left and right hands, the tongue, and the foot are vast and distinct. Thus, previous studies have shown that BCI applications may control the movement of those bodily limbs through imagination.

According to Yokoyama et al. [15] neural decoding of basic MI information (e.g. differentiation of brain states between 'walk' and 'rest' and between 'left leg' and 'right leg') was employed in gait rehabilitation research using MI-based BCI. Upper-limb movement studies, on the other hand, have been successful in deciphering more comprehensive and functional MI information, such as hand postures and movement direction.

2.1.3.2 *Steady-State Evoked Potentials (SSEP)*

When a person observes a periodic stimulus, such as a flashing visual or an amplitude-modulated sound, SSEP emerges. The stimulation frequency or harmonics are equal to the EEG signal frequencies, which is an essential feature of SSEP. The stimulation of a set frequency causes SSEP by producing EEG activity at the same frequency as the stimulus. SSEP can be further classified based on visual, auditory, and somatosensory stimulation as Steady-State Visually Evoked Potentials (SSVEP), Steady-State Auditory Evoked Potentials (SSAEP), and Steady-State Somatosensory Evoked Potentials (SSSEP) respectively [26].

2.1.4 *P300*

The EEG signal in the brain is composed of multiple signals with varying qualities, each of which correlates to a particular mental activity. One of the signals, P300, has been employed in a variety of BCI applications. When the user is exposed to an unusual stimulus in a sequence of common stimuli, a positive wave known as P300 occurs roughly 300 milliseconds after the commencement of the target/rare stimulus. It is triggered when a person

detects a single "target" stimulus within a steady stream of "non-target" inputs. P300 can be generated when the person focuses on a specific stimulus while concurrently being exposed to multiple stimuli.

2.1.4.1 *Event-Related Potential (ERP)*

Event-Related Potentials are very small voltages deflections generated in brain areas in response to certain events or stimuli. They are EEG alterations that are time-locked to the sensory, motor, or cognitive events and provide a safe and noninvasive technique for studying psychophysiological correlates of mental processes. A wide range of sensory, cognitive, and motor activities can trigger event-related potentials. They are assumed to represent the total of postsynaptic potentials generated when a large number of similarly oriented cortical neurons (thousands or millions) fire in synchrony while processing information [42].

Human ERPs are classified into two types. The early waves, or components peaking within the first 100 milliseconds following input, are referred to as 'sensory' or 'exogenous' since they are heavily influenced by the physical properties of the stimulus. In contrast, ERPs generated in later parts indicate how the person analyzes the stimuli and are referred to as 'cognitive' or 'endogenous' ERPs because they investigate information processing. The waveforms are classified based on latency and amplitude.

Because of its significant potential in understanding and analyzing gait-related brain rhythms and Event-Related Potentials, Brain-Computer Interfaces or Brain-Machine Interfaces have been more intensively researched in the field of gait rehabilitation in recent years [16].

2.1.5 *Perturbation Evoked Potential (PEP)*

Physical balance disturbances cause brain reactions accompanied by a specific cortical activation, the so-called PEP [10]. Perturbation Evoked Potentials are ERPs elicited by predicted or unexpected external whole-body disturbances (mechanical perturbations to stability). PEPs are multicomponent cortical responses that are comprised of an initial small positive wave (P1) followed by a large negative potential (N1) and a succeeding positive (P2) and negative (N2) waves (4). The PEP differs from other ERPs in that it is evoked by multimodal stimuli (visual, vestibular, somatosensory) caused by whole-body perturbations and has a relatively large N1 amplitude compared to evoked responses to other stimuli [49].

PEPs are extensively dispersed throughout the frontal, central, and parietal regions, however, the peak amplitude of PEP N1 is greatest over frontocentral locations independent of stimulus or task circumstances. As a result, the minimal recording montage to express the PEP N1 should comprise one electrode at either Cz or FCz as shown in Figure 3.

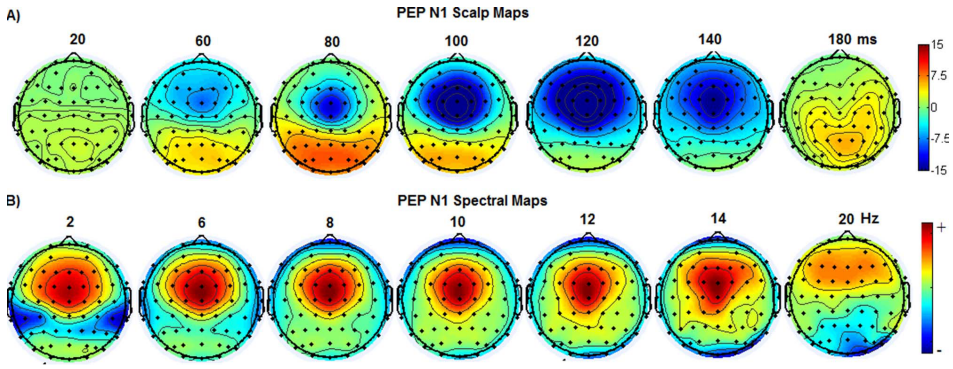


Figure 3: ERP scalp maps showing the topography of perturbation-evoked N1. Taken from [48]

PEPs can be contaminated by movement artifacts owing to whole-body perturbations, in addition to other neural and non-neural artifacts such as eye blinks, eye movements, EMG, ECG, and line noise artifacts, which are of special concern. It is worth noting, however, that perturbed head motions do not appear to mask the early components of PEPs. This is most likely owing to the timing of events, with the perturbed head movement beginning after the early components of the PEPs [48].

2.1.5.1 *Components*

The first component is a brief positive wave that lasts between 30 and 90 milliseconds. This is followed by a negative peak at 90-160 ms and a last, late response (P2 and N2) at 200-400 ms. These PEPs are frequently identified by averaging event-related waveforms from several trials. If PEP could be identified in a single trial and used in a BCI system, balance disturbances could be detected far sooner in a fall detection system based on EEG. This would provide considerably more time to implement preventative actions [35].

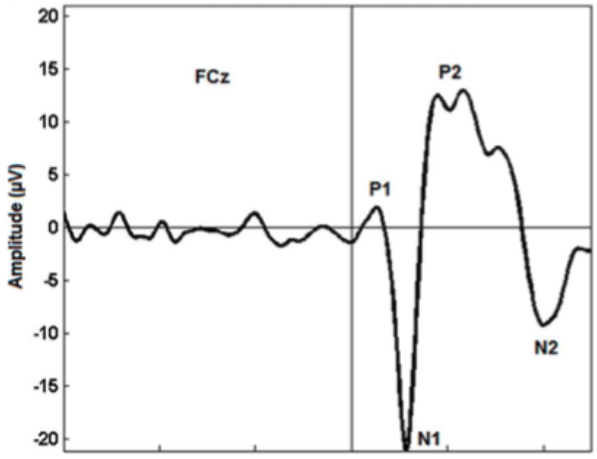


Figure 4: Waveform and terminology of Perturbation-evoked Potentials from a single normal subject. Taken from [48]

The most commonly researched properties of PEP components are the onset latency, peak latency, and peak amplitude. The peak amplitude of the PEP components is measured with regard to a pre-perturbation baseline period or as the voltage difference between the component peak and the perturbation onset. In Varghese et al. [48] studies, the latencies of the PEP components are assessed in relation to perturbation onset (i.e., time = 0 (Figure 5)). The peak latencies of the various PEP components vary between studies in the healthy adult population due to changes in perturbation release timing, the metric employed to quantify perturbation initiation or the time point picked by different researchers as the start of timing for the PEPs. For example, perturbations delivered at various stages of the gait cycle may result in variable PEP latencies. Furthermore, the peak amplitude of the PEP components varies with the kind of perturbation utilized and the starting acceleration of the perturbation in the healthy adult population.

2.1.5.2 *PEP P1*

The first component of the PEP is the P1. Varghese et al. [48] paper summarizes that the peak amplitude of the P1 is small, usually ranging between 0.2 and 7 µV. However, mean amplitudes as great as 12.7 µV were reported for the P1 evoked by sudden platform toe-up (ankle perturbations) tilt, measured from the P1 onset to the peak of P1.

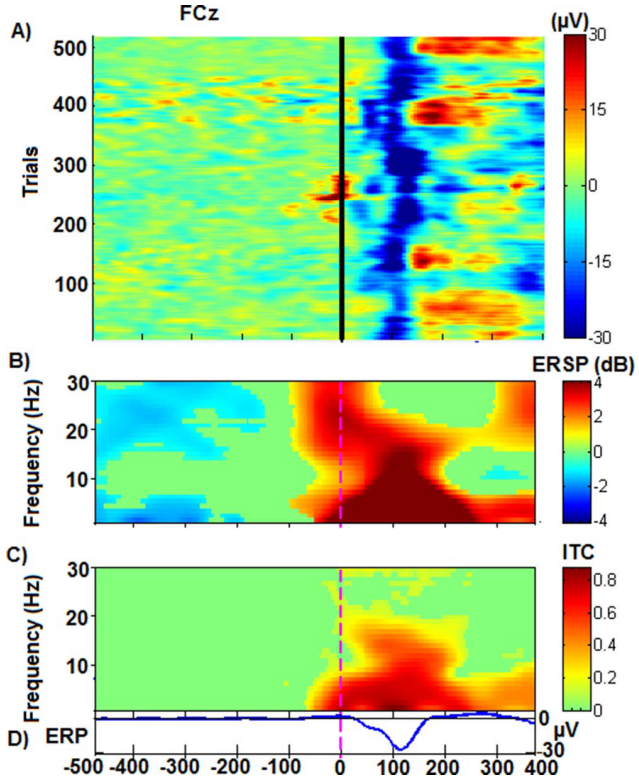
Because of its modest amplitude and/or unpredictability in the mechanical initiation of disturbance relative to the trigger, the P1 is not always discernible from background EEG signals. When the disturbance is quick, the P1 is easily detected; but, when the perturbation is sluggish, it might be indistinguishable from the baseline EEG activity [10].

Evidence from previous works [10, 48] indicates that P1 is the first unspecific cortical response to perturbations and that P1 features are associated with various parameters such as age, stance width, and height. The same studies indicate that the P1 is most likely generated by somatosensory input. P1 latency is controlled by the subject's starting postural state (stance or gait), height, age, and diseases, whereas P1 amplitude is determined by the beginning stance width (narrow or wide). It was discovered that when the perturbation is created during stance and gait circumstances, the latency of the P1 peak is considerably delayed in gait situations. With increasing stance width, the amplitude of the P1 rises. Varghese et al. [48] reported the impact of the base of support by inducing sudden platform toe-up tilts with three different initial stance widths (0 cm, 3 cm, and 30 cm) and discovered that the amplitude of the P1 increased with a wider stance.

2.1.5.3 *PEP N1*

The PEP N1 is broadly distributed in the frontal, central, and parietal regions. The amplitude of N1 varies from 0.8 to 80 µV depending on the variables varied in the tests and is greatest at either the FCz or CZ electrode [6, 35].

Unlike other PEP components, the N1 is reproducible across trials and subjects, regardless of the direction of perturbation or the many parameters that impact the latency and amplitude of N1. Even with various types of perturbations (stance perturbations versus sitting perturbations) and balancing reactions, the PEP N1 occurs bilaterally with no significant changes in magnitude across left and right hemispheres (upper limb vs lower limb) [48].



(A) ERP-image plot of single-trial EEG epochs across all subjects. (B) Event-related spectral perturbation (ERSP) and (C) inter-trial coherence (ITC) plots at the FCz electrode site. The vertical dashed line at time = 0 represents the perturbation onset. The bottom trace (D) shows the perturbation-evoked N1 averaged across all subjects.

Figure 5: ERP the scalp maps, time series, and time–frequency plots of PEP N1 averaged from a single-subject EEG data. Taken from [48]

It has been proposed in Varghese et al. [48] that the N1 is associated with error detection and is independent of sensory and motor processes linked with compensatory postural responses. They discovered that the amplitudes of the N1 were smaller during predictable perturbations compared to unexpected situations. A substantial N1 potential emerged, however, when an unpredictable perturbation was applied following a sequence of predictable perturbations.

After observing similar N1 peak latency and amplitude for stance and seated perturbations, the authors [10, 48] proposed that the N1 response is driven by the perturbation and is independent of specific sensory inputs that code the perturbation characteristics and motor elements of the balance reactions.

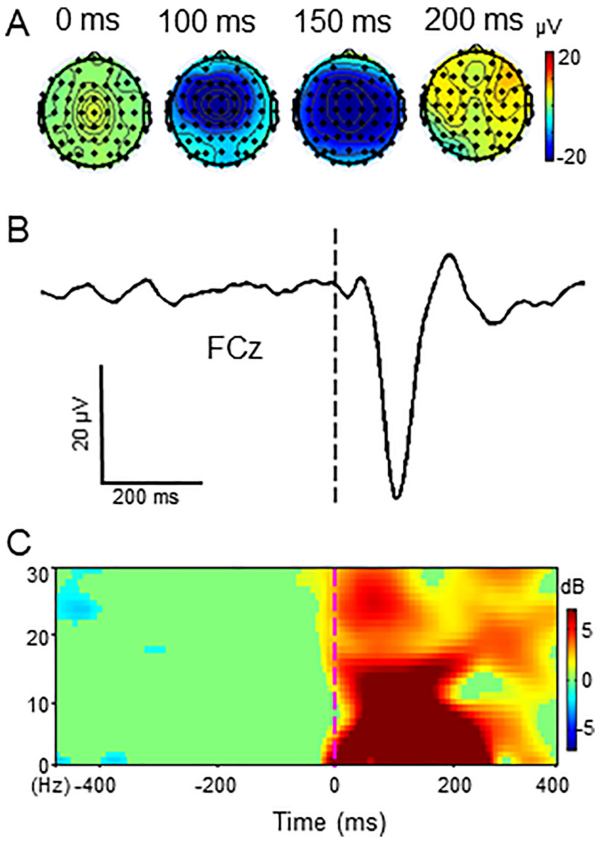
Despite these differences in the role of the N1, it is acceptable to assume that the N1 plays a role in compensating postural responses. However, due to the lack of information on previous reports, further studies are needed to understand the significance of the N1, especially whether it plays a role in detecting postural instability and/or generating postural responses.

The PEP N1 is the outcome of averaging across many trials. As such, it is simply a summary measure of cortical activations linked to balancing regulation, and averaging may result in the loss of significant information about the underlying neuronal assemblies involved in producing the PEP N1.

The participation of distinct frequency bands during the PEP N1 has been linked to sensorimotor and cognitive processes connected with reactive balance management. Furthermore, concurrent theta, alpha, and beta frequency phase synchronization during this phase of the response to balance perturbation demonstrates the participation of many concurrent cognitive processes in balance regulation, and that their combined activity leads to the emergence of the PEP N1 [48].

The PEP N1 is the most extensively researched component of the PEP. Several environmental factors impact the PEP N1 latency, amplitude, and psychological factors such as initial postural state (stance or gait), learning effects or adaptation, perturbation amplitude and duration, the direction of perturbation (forward or backward), perturbation mode (unilateral or bilateral), mechanical triggering mode of perturbation (self-induced or externally induced), the base of support (stance width), concurrent cognitive task (dual tasking) and postural threat.

As mentioned by Varghese et al. [48] PEPs related to both stance and gait perturbations using treadmill accelerations indicated delayed latencies and lower amplitudes in the PEP N1 during the walking test compared to the standing condition. The authors hypothesized that the lower amplitude of the PEP N1 explains neurons' failure to react to perturbation stimuli caused by active movement during gait. In contrast to the differences in amplitudes and latencies between stance and gait, no difference in N1 amplitude and latency was observed between seated perturbations that elicited upper limb compensatory postural responses and stance perturbations that elicited lower limb compensatory responses. Also, the authors found that changes in sensory qualities (sensory from sitting vs stance perturbations) and motor responses (upper limb versus lower limb) required to recover balance have no effect on the spatiotemporal aspects of the PEP N1.



ERP-scalp maps (A) and ERP time series (B) of PEP N1 at the FCz electrode site averaged from a single normal subject. Time = 0 represents the perturbation onset. (C) Event-related spectral perturbation plot at the FCz electrode from a single normal subject.

Figure 6: ERP plots. Taken from [49]

2.1.5.4 *PEP P2 and N2*

The PEP N1 is followed by the P2 and N2 (known as late PEPs). In contrast to the PEP N1, the P2 and N2 are not identified in all PEP research. Varghese et al. [48] cite that early PEP experiments, in particular, were unable to uncover any long latency components (longer than 180 ms).

One possible explanation is that these lengthy delay components are associated with task-specific behavior and their timing is not fixed, depending on the task [48]. As a result, late PEP are analyzed in a wider time window after the N1 response, typically 200-400 ms after the start of the perturbation [10]. The N2 amplitude fluctuates between 27 and 28 μV and is greatest at the CZ electrode. The P2 peak amplitude spans between 22 and 44 μV and is greatest at the Cz or CPz electrodes. The starting stance width influences the magnitude of the P2 [48].

As a result, the authors [48] hypothesized that late PEP could indicate cognitive processes linked with task demands or cognitive processing of the balance disturbance rather than sensorimotor processing involved with balance response regulation. The same authors discovered that a concurrent cognitive task performed during

the stance perturbations had no effect on the amplitude of the late PEP and argued that they indicate a sort of orienting response that does not demand the attentional resources required in a dual-task paradigm.

The amplitude of P2 but not the latency is also modified by the initial postural position (stance or sitting), with stance perturbations producing much greater P2 amplitudes than seated perturbations. This discrepancy was attributed to differences in the latter phases of balance control between standing and sitting circumstances [10].

To summarize, late PEP are unlikely to represent cortical sensorimotor processing related to postural response regulation, and if attention does explain variation in late PEP, it is more likely driven by attention changes toward new perturbation events.

2.2 DECODING

As mentioned in the previous subsection, to build a BCI system, five components are generally needed: signal acquisition during a specific experimental paradigm, preprocessing, feature extraction, classification (detection), translation of the classification result to commands (BCI applications), which one of them will be detailed on next subsections.

For quick and accurate processing and analysis of brain data, researchers have developed many open-source software packages and toolboxes such as BCI20001, EEGLab, FieldTrip, and Brainstorm. These software packages are based on advanced signal and image processing methods and artificial intelligence programs for performing sensor or source-level analyses.

Prediction of human movement intention is highly significant for successful gait rehabilitation. In a BCI-based rehabilitation system, the brain waves are extracted, processed, and translated to control an assistive device. For an effective assistive system, it is critical to detect the movement intention as early as possible to provide the system with enough time to adapt to the requirement of the individual [16].

2.2.1 *Data Preprocessing and Feature Extraction*

2.2.1.1 *Data Preprocessing*

Pre-processing is a time-consuming procedure that is used to eliminate any undesirable components present in the EEG signal. Good preprocessing improves signal quality, which leads to improved feature separation and classification performance. The basic approaches to reducing artifacts in the observed EEG are simply low, high, and bandpass filters. However, these are only effective when the signal's frequency bands do not overlap. When there is spectral overlap and artifacts are recorded with the EEG, various artifact removal techniques such as adaptive filtering, Artifact Subspace Reconstruction ASR, Common Average Reference CAR, and Independent Component Analysis ICA decomposition. Besides, in some cases to increase computational efficiency and sync EEG frequency with other devices, for example, EMG, the signal is downsampled.

As mentioned in the previous paragraph for data preprocessing, firstly, a series of filters are applied to the signal to extract accurate information. The raw signal is collected from the electrode, and stored in a circular

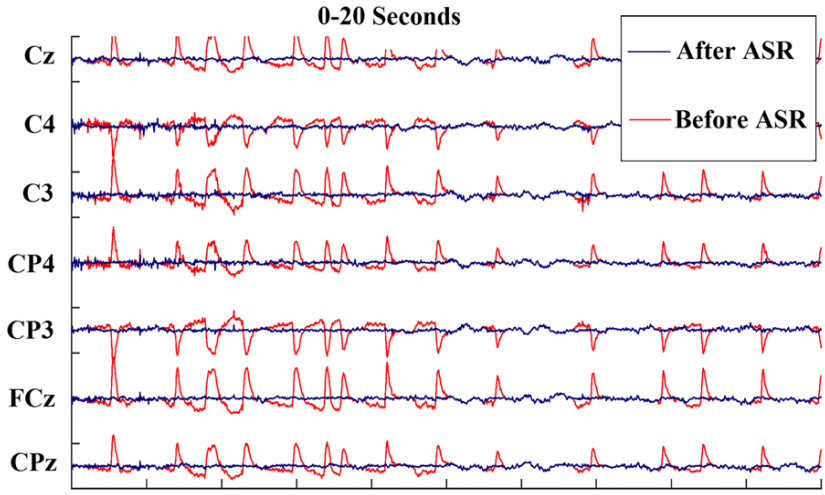


Figure 7: Performance of ASR in removing eye movement artifacts and unnatural high amplitude noise.

buffer for easier data stream handling. The notch filters, type of filter, can be applied to the raw signal to diminish power line noise and its harmonic interference at 50Hz. The filtered signal is then filtered with a 2nd or 4th-order Butterworth filter can be the low-pass, high-pass, or band-pass filter to extract the information [12]. If needed the filters can be combined to obtain better signals.

In addition, an approach to remove artifacts is to perform amplitude threshold rejection removing all trials with an amplitude that exceeds a specific pre-defined voltage. The rejection threshold can be adjusted based on the value of the standard deviation (STD) [10]. Other approaches can be used in process of removing unwanted artifacts from EEG signal and depending on the application they can be carried out in two steps. Artifact Subspace Reconstruction is a non-stationary method that uses sliding window principal components analysis (PCA) to remove unusual large-amplitude noise or artifacts. The usage of ASR increases data stationarity and makes the data suitable for independent component analysis operation. In [16] studies, ASR can be used for two purposes: bad channel rejection (flat channels, channels with a large amount of noise may be removed based on their standard deviation, and channels, which are poorly correlated with other channels), and removal of short-time high-amplitude artifacts in continuous data. The value is chosen in such a way that it was small enough to remove activities from artifacts and eye-related components and large enough to retain signals from brain-related components.

According to Varghese et al. [49], ICA is performed on epoched EEG data to remove eye blinks, eye movements, whole-body movements, muscle artifacts, heart signals, and line noise artifacts. Identification and rejection of artifact-independent components (IC) is based on the visual inspection of topographic maps, power spectra, and time domain activity of each IC [49].

Lastly, the common average reference which is spatial filtering can be applied to the raw input signals to improve the signal-to-noise ratio. For every sample time, CAR subtracts the mean value of all electrodes, which minimizes the uncorrelated random noise with a zero mean through the averaging process [25].

2.2.1.2 *Feature Extraction*

After the line noise and artifact removal phase, the most discriminative and non-redundant information within the EEG is extracted through different feature extraction techniques. Time-domain, frequency domain, time-frequency domain, and spatial domain are the popular types of feature extraction techniques in EEG-based BCIs.

The features can be extracted from all channels or from specific channels depending on preprocessing results. In Jochumsen and Niazi [23], the template matching feature, an average of the epochs for the specific channel was used, and the cross-correlation is calculated with each epoch. The feature was the cross-correlation with zero time lag. The time-domain features were the mean amplitude of 0.5-second non-overlapping windows (i.e.: four features were extracted from each channel). The frequency-domain features were power spectral density estimated for the entire epoch in 1 Hz bins from 6–30 Hz. A 1-second Hamming window with a 0.5-second overlap was used.

For time-series applied data segmentation to divide the data into time intervals [16] where the signals are segmented in epochs locked to the onset of each class [32]. Using a sliding window with shift and overlapping is possible to extract the signal amplitude in specific channels or components that will be used as input to the classifier [51]. Besides, windows of different time lengths can be chosen to correspond to a specific class.

If the number of features is too high compared to the number of samples, there is a high chance of having redundant and noisy features in the feature set. That is why a feature selection method is necessary to remove redundant features. In Hasan et al. [16] work, the absolute value of the standardized u-statistic of a two-sample unpaired Wilcoxon test (also known as the Mann-Whitney test) was chosen to be the criterion to select distinctive and informative features. To further reduce the number of features, the average of the absolute values of the cross-correlation coefficient between the candidate feature and all previously selected features were calculated, and features that were highly correlated with the features already picked were less likely to be included in the output list. This procedure ensured the formation of a reduced and more distinctive set of features for successful classification.

Dimensionality reduction techniques improve the performance of classifiers, as they reduce the number of features to mitigate the curse of dimensionality. These techniques can be categorized into feature selection (e.g. F-score) selects a subset of original features and feature extraction which derives new features from original features. The first category does not change the nature of the features, while in the second category, mathematical transformations are used to reduce the number of features [37].

2.2.2 *Classification - AI Algorithms*

To operate a BCI system, the individual must generate distinct brain activity patterns that the system can recognize and convert into commands. It should be noted that either regression or classification techniques might be used to achieve the stated goal. However, the use of classification algorithms is the most prevalent strategy as the degrees of freedom of most BCI systems are low and output commands are reduced to just a few options (classes). The classification step's design requires selecting one or more classification algorithms from a large number of options.

The capability to achieve reliable automated categorization of EEG data is a critical step toward making EEG more practical in many applications and less reliant on skilled specialists. It is worth mentioning that, despite major progress in traditional BCI systems over the last two decades, the biggest challenge still confronts significant problems in EEG categorization. Recently, the availability of large EEG data sets has led to the use of Deep Learning (DL) architectures, particularly to extract relevant information from signals that were previously impossible to obtain using conventional approaches, and has demonstrated success in addressing the aforementioned challenges.

2.2.2.1 *Convolutional Neural Network*

Convolutional Neural Networks (CNN) is a class of artificial neural networks (ANNs) that specializes in spatial information exploration. CNNs have seen a lot of successful applications in many different domains such as reaching human-level performance in image recognition problems as well as different natural language processing tasks Motivated by the success of these CNN architectures in these various domains, researchers have started adopting them for time series analysis [17].

CNN has at least one convolutional layer, which uses a convolution operator to map the input to output [43]. A CNN's three major layers are convolution, pooling, and fully connected as shown in Fig. 8. On a dataset of huge registers of various classifications, CNNs outperform the prior highest-computing techniques. CNNs are feed-forward neural networks that may be used for image processing, pattern recognition, and classification. The convolution layer filters the input data, such as kernels with trainable parameters, creating the next layer in the network also called the feature map. The feature map is then down-sampled using the pooling layer to reduce the dimension and consequently the computational complexity and overfitting. These settings enabled the learning of many network features while keeping the number of traceable parameters low. After several alternating convolution and pooling operations, the final feature map is unrolled to a fully connected hidden layer to generate the output [34].

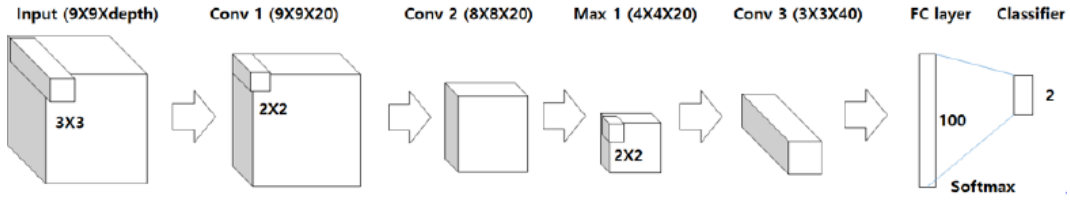


Figure 8: Example of CNN Architecture. Taken from [38]

2.2.2.2 *Recurrent Neural Network*

Recurrent Neural Network (RNN) architecture is a sophisticated Deep Learning classification algorithm that is designed exclusively for sequential data. This type of DL architecture can assess the entire logical sequence of the incoming data. These logical sequences are content-rich and have a complicated temporal interaction with one another. The basic principle of RNN is that the current network's hidden state retains the prior input information,

which is then used for the next current network. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are two popular RNN designs that has received a lot of attention and success [43].

2.2.2.3 *Long Short-Term Memory*

Long Short Term Memory is a more advanced variant of the standard RNN in which three gates (forget gate, update gate, and output gate) are added to regulate the information to store and pass while avoiding the vanishing gradient problem that is generally associated with training a conventional RNN. Each gate in the cell is in charge of a distinct duty. The forgetting gate's function is to remove undesired information from the prior state and output from the top hidden layer. The update gate updates new components to the state, while the cell filters the current state and finds desired and unwanted information to guarantee that the output gate selects the critical information generated by it. The data to be output is selected by the output layer, which is handled by the filtered input and cellular state, i.e. the output layer decides which information will be output and processed by the cell state.

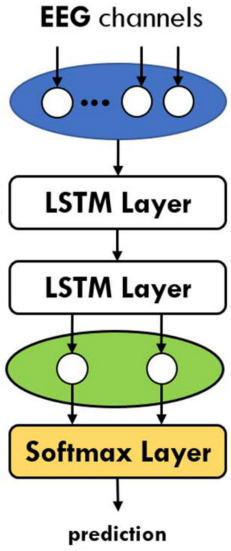


Figure 9: Example of LSTM Architecture. Adapted from [40]

2.2.2.4 *Gated Recurrent Unit*

The Gated Recurrent Unit (GRU) is another RNN enhancement, and similar to LSTM, GRU uses gating mechanisms to control and manage the flow of information between cells in the neural network. GRU contains two gates (a reset gate and an update gate) that manage how information is retained and sent between nodes. The update gate functions similarly to an LSTM's forget and input gates, it determines what information to discard and what fresh information to include, and another gate used to select how much past information to forget is the reset gate. Although prior empirical assessments have not revealed a clear winner between GRU and LSTM, it is thought that GRU may be a superior model when dealing with fewer data points to generalize on, due to the fewer parameters in contrast to LSTM [43].

2.2.3 *BCI Applications for Activity Daily Life*

BCI technology has clinical and non-clinical uses in a variety of fields, including medicine, entertainment, education, and psychology, to address a variety of health conditions, including cognitive deficiencies, poor processing speed, impaired memory, and motor capacity decrease in the elderly. These concerns can reduce the quality of life for the elderly and have a negative impact on their mental health. Many BCI applications, as shown in Fig. 10 have been created in the last decade to assist elderly persons in maintaining a healthy, decent quality of life and feeling of well-being.

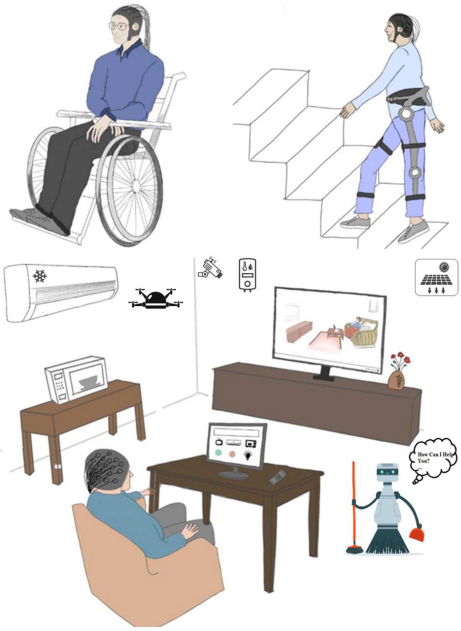
2.2.3.1 *Wheelchair Control*

One of the primary goals of a BCI wheelchair is to improve the quality of life and autonomy of patients with motor neuron illnesses such as amyotrophic lateral sclerosis. This breakthrough technology enables handicapped individuals to drive a wheelchair using their brain activity, offering them autonomy while traveling through an experimental environment. BCI wheelchairs can employ various forms of EEG control signals such as MI, P300, SSVEP, and hybrid [30].

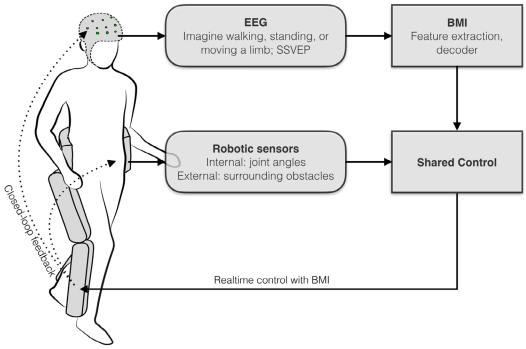
2.2.3.2 *Gait Recovery*

Exoskeleton robots help patients in a bottom-up strategy, in which the exoskeleton acts on their limbs (bottom) to create reactions in the brain region (top). Bottom-up therapy aided by a robot has yet to show considerable improvement over therapist-assisted rehabilitation.

The use of BCI with lower limb exoskeleton robots has increased due to its use in rehabilitation. According to research [38], to compensate for the downside of current robot-assisted technology, it is desirable to shift from the bottom-up approaches to the top-down approach in the aspect of enhancing neuroplasticity in rehabilitation. By completing the loop, gait rehabilitation parameter feedback produced from bio-signals can have a favorable influence on patients' neuroplasticity, which means that walking supported by a robot stimulates the patient's brain. In this regard, EEG is useful for recognizing the patient's thoughts during gait therapy.



(a) BCI Applications. Taken from [30]



(b) Exoskeleton. Taken from [18]

Figure 10: Examples of BCI Applications

Robotic training is being used to reestablish gait patterns. Conventional rehabilitation methods utilizing robotic technology aid in sensorimotor repair by assisting and encouraging participants to do specified activities. Exoskeletons are a potential task-oriented tool for restoring a more physiological walking pattern in severely impaired persons and recovering natural alternating activations of lower limb muscles [33].

2.2.4 *Prevent Fall Risk*

In order to develop a BCI control of the robot-assisted gait device, fundamental research aiming at detecting the precise active role of the motor cortex during the gait cycle has to be done [11]. Passive Brain-Computer Interface (pBCI) do not provide active control to users; moreover, they monitor their state of mind and detect changes in the state of mind of users. Due to their ability to detect changes in the user's mental state, pBCIs are a promising tool for detecting perceived balance perturbation [10].

So far, only a few studies have investigated the neural correlates of human walking, principally due to both the inherent experimental difficulty of measuring EEG signals in the ambulatory context and the challenging goal of balance control in walk rehabilitation tasks and loss of balance during a perturbation.

2.3 DISCUSSION

This chapter provided an introduction to current topics related to brain-computer interfaces and the decoding of brain signals for ADLs and slip-like perturbations. The analysis carried out considered EEG data that allowed

a comprehensive characterization of the response to this type of perturbation. In terms of the brain signal, understanding the role of brain regions (channels) related to gait and the responses of these channels in response in identifying the occurrence of loss of balance due to perturbation during gait is very important. The analysis of EEG data, presenting pre-processing techniques, ML and DL algorithms applied in BCI devices, corroborates the importance of developing and improving this technology. In addition, with regard to slip-like perturbations, it was possible to identify the type of signal to look for and what are the characteristics of the signal when there is a loss of balance (waveform, amplitude, and moment).

As it was possible to verify throughout this literature review, the combination of EEG data for ADL and loss of balance provides different complementary information for the development of this work, taking into account the parameters addressed in the literature review. Furthermore, the ranking of the various AI-based models for classifying disturbance and non-disturbance identifying those that most influence the landslide outcome should also be considered in the next tasks of this dissertation. Finally, the computational performance demanded in the construction of AI models must be studied and analyzed due to its relevance in a real-time scenario.

3

ACTIVITIES OF DAILY LIVING AND RESPONSE TO BALANCE LOSS: LITERATURE REVIEW

Despite the progress of devices and gadgets to get a natural and effective recovery, the subject must fully control the devices they use. In this context, Brain Computer Interface has been introduced to decode motor or cognitive intentions from recordings of brain activity and translate these predictions into commands for computer programs or robotic devices [28]. This part of the BCI which deciphers the user's motor intent from recorded brain activity is typically referred to as a neural decoder. BCI technologies, based on motion decoders from brain recordings through non–invasive technologies such as scalp EEG signals appealing to ensure wider applicability of personalized brain–controlled rehabilitation devices. The electrodes placed on the scalp capture the signals that have an amplitude in the range of microvolts in real-time. In the development of EEG-based BCIs, individually tuned parameters for characterizing signals are computed from several training trials, which are used to perform online signal decoding [25].

The main goal of this study is to review the literature to understand the process of decoding different modes of locomotion (walking, climbing/descending stairs, ascending/descending on a ramp) which are classified as Activities of Daily Living, and develop a solution for imbalance events of human gait through the use of EEG which function is capture and register the brain signals [40, 4]. A protocol for data collection will be implemented based on good practices found in the scientific literature [4]. It is intended that the group of volunteers with determined healthy subjects use at least a 16-channel EEG. These sensor data will be collected with different subjects walking in a controlled environment. The data of EEG must be processed to create significant features, through feature engineering, which will be used to detect the subject's locomotion intentions through the implementation of artificial intelligence models [26]. It will be necessary to identify tasks and algorithms for pre-processing and removal of the most commonly used artifacts, such as the Independent Component Analysis and the Artifacts Subspace Reconstruction, as well as identifying the most relevant brain rhythms. Therefore, it is intended to evaluate the classification of AI-Based models, and compare them in order to have an optimized benchmark model for real-time use with low computational cost. It is expected that the best model is able to predict the occurrence of loss of balance.

3.1 METHODS

3.1.1 *Research Strategy*

We have conducted a systematic review. This type of review is a systematic way of exploring existing literature. It starts with choosing a set of keywords, along with Boolean operators, to try and extract only the most pertinent papers in the literature. Since our focus is to review papers that decoded modes of locomotion capturing the signals from EEG and use intelligent machine learning algorithms for gait analysis in the lower limb, the choice of keywords should reflect the topic. They should be general enough not to miss applicable literature and encompass the varying techniques, sensors, devices, and terminology researchers use in their research but restrictive to eliminate irrelevant research. The aforementioned keywords have been used to extract papers available in three databases: WEB OF SCIENCE, PUBMED, and SCOPUS. : ("EGG" OR "Electroencephalogram") AND ("ADL" OR "Walking" OR "Climb" OR "Descend" OR "Sit-to-Stand" OR "Stand-to-Sit") AND ("BCI") AND ("Decoding") AND ("Classification") AND ("Machine Learning" OR "Deep Learning").

3.1.2 *Inclusion and Exclusion Criteria*

In addition to the keywords, inclusion, and exclusion criteria have been used to further filter the results. Results were limited to journal and conference papers only. There were a few studies where the full-text paper was not available or inaccessible by us, and hence was not included. The total number of papers identified with this inclusion and exclusion criteria was 151 papers. Afterward, duplicate papers available on databases have been removed, as good papers were published before 2015, reducing the number of papers to 98. These papers have been analyzed based on abstract only, manually removing less relevant or irrelevant papers (i.e. the scope of these papers have no significance to our review). Some reasons that resulted in the exclusion of papers included, upper limb rather than lower limb robotics, motor imagery instead of gait analysis, and the absence of electroencephalography. Results were further limited to research articles excluding conference papers. 16 out of the 54 full-text articles that we assessed for eligibility were included in this review. The full text of these papers has been reviewed focusing on the parameters the researchers are considering, the intelligent algorithms they used, the sensing modalities, the types of subjects they have tested/trained their algorithms on, and the overall performance of their systems. This process is visually illustrated in Figure 11.

3.2 EXPERIMENTAL PROTOCOL AND SETUP

3.2.1 *Experimental Procedure*

Regarding the group of 16 articles gathered in this literature search, there was clear evidence that walking was more prevalent to decode gait. From the whole group of studies, 12 conducted walking, stop and stand movement, 5 of 12 papers conducted sit-to-stand, stand-to-sit movement, 2 of 12 conducted step up, side step, and back

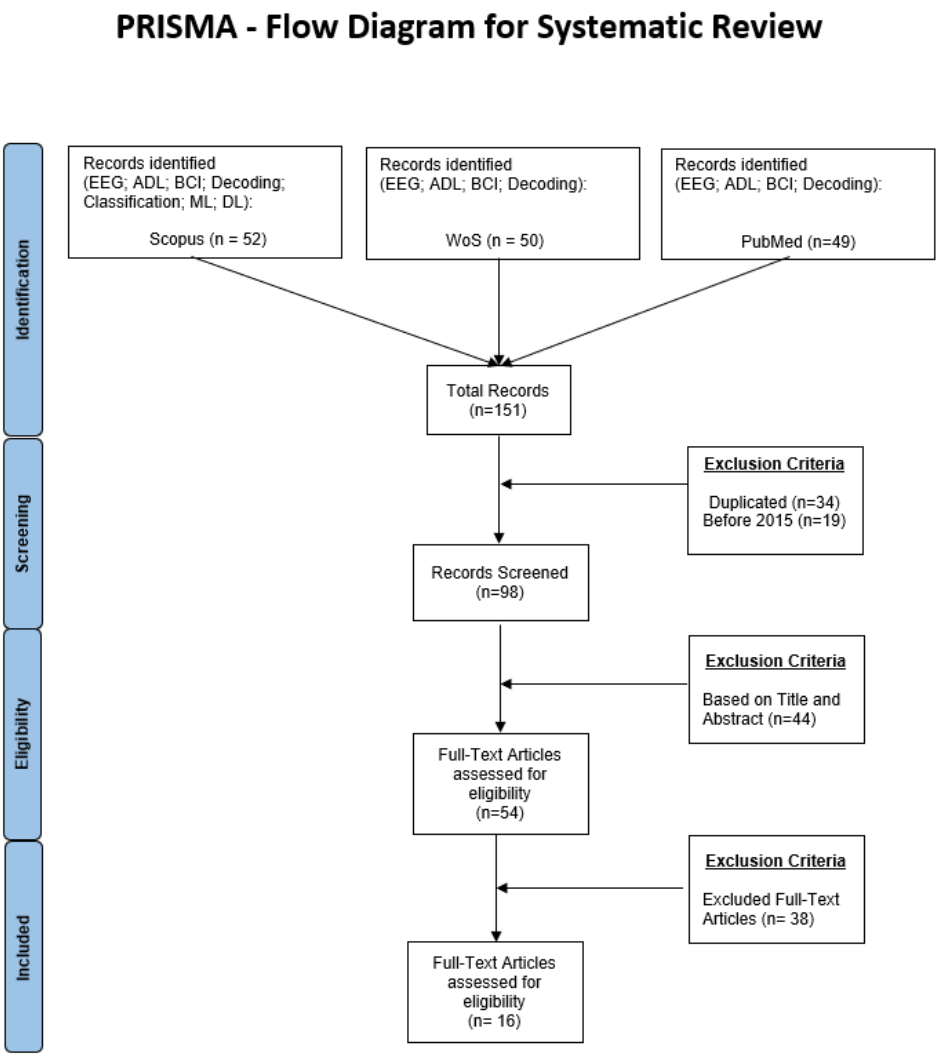


Figure 11: Flowchart on the methodology of article selection.

step movement, and only one conducted turn left and turn right in its experimental protocol. Table 1 depicts the 16 studies that conducted gait decoding. All experimental protocols and informed consent were obtained from all participants. In 13 papers all individuals with no history of neurological disorder participated in their study, 2 papers conducted their experiments with stroke patients as well and only one paper conducted its experiment with an amputee subject.

Gait decoding was delivered in 16 studies, seven papers conduct these studies on treadmills with a minimum speed of 1.5 km/h and a maximum speed of 3.5km/h. In 3 papers each session began with the subject standing quietly in an upright posture for 2 minutes (mins). The treadmill's speed was slowly increased to 1.6 kilometers per hour (km/h) in [46] experiment and the subject walked normally at this gait speed for 5 mins. At the beginning part of the experiment where subjects finished the baseline period, subjects were instructed to walk on a treadmill at a fixed slow speed of 1.5 kilometers per hour (km/h) [43]. During the task, subjects were instructed to walk

steadily for 15 mins where the decoder is calibrated for the next BCI control [47]. Chisari et al. [39] and [40] experiment with the participants walking on a treadmill at two distinct velocities, 2.5 km/h and 3.5 km/h. For each walking velocity, two acquisitions of 10 minutes each were performed. Healthy participants in [11] study attended one experimental session in which they were asked to either walk in a passive or in an active mode at a speed of 1.5Km/h wearing an exoskeleton, as soon the treadmill started, participants walked for 49s, after which the treadmill was stopped. The treadmill had a delay of at least 7s to come up to a stable speed and 7s to slow down and stop completely.

Table 1: Experimental Protocol - Procedure

Article	Participants	Age (avg)	Task	Treadmill Speed (km/h)	Trials	Time per trial (min)
[43], [47]	8 Healthy	24.5	sit-to-stand; stand-to-sit	1.6	3	20
[32]	8 Healthy	24.5	walking	1.6	3	20
[39], [40]	11 Healthy	30	walking	2.5 / 3.5	2 for each speed	10 for each trial
[22]	10 Healthy	25	resting; walking intention; walking		50	0.5
[11]	10 Healthy	32	walking; stand	1.5	7 active walking; 7 passive walking	1
[23]	13 Healthy	24	stand-to-sit; sit-to-stand; walking; step up; side step; back step;		50 for each task	180
[37]	27 Healthy	24	walking			
[46]	5 Healthy	24.5	walking	1.6		
[45]	5 Healthy	8	stand-to-sit; sit-to-stand; walking; stop;	1.6	20	
[51]	1 Healthy; 1 Non-Healthy		walking; turn left; turn right; stop		20	
[38]	3 Healthy; 8 Non-Healthy		walking		9	1
[12]	4 Healthy	23	stand-to-sit; sit-to-stand		78	2
[27]	20 Healthy	22	step forward; step up; step back		10	4
[16]	7 Healthy; 2 Non-Healthy	32.6	walking; stop		145	175

Six of these 16 manuscripts performed gait movements during overground locomotion. The experiment paradigm in [22] consisted of three states: a resting state of approximately 10 s, an intention state, and 10 s exoskeleton walking. Prasad et al. [51] is composed of 2 tasks, the first task was a four-class, single-session task in which the subjects performed different movements (i.e.: walking forward, turning right, turning left and stopping,

following the marked path on the ground, and the second task, subjects only executed walking and stop motions according to audible beep instructions).

Eguren et al. [45] perform four different types of locomotion tasks (i.e.: sitting and standing, start and stop overground walking). In the sitting and standing task, the participants completed 20 sit-to-stand and stand-to-sit transitions which were initiated by a visual cue placed in front of the subjects. The session began with the participant standing quietly in an upright posture for 15 seconds. After the quiet standing period, the participant began the self-initiated sit-to-stand and stand-to-sit transitions. The waiting period between the transitions was approximately 10 seconds. The participants were verbally informed when 20 transitions were completed. At the start and stop of the overground walking session, the participants completed 20 walk-to-stand and stand-to-walk transitions. The session began with the participants standing still for 15 seconds. After the stationary standing period, the participant held the standing position for a period ranging randomly from 10 to 15 seconds before the stand-to-walk transition (indicated by a green light). The participants walked on a 10-meter walkway and were instructed to stop walking at the onset of a red light. The walk-to-stop transition completed a single trial and the participant returned to the starting position. The process continued until 20 trials were completed.

Jochumsen et al. [23] performed 6 tasks which were: (1) stand-to-sit; the subject was standing in front of a chair (height of seat: 45 cm) and had to sit on that, (2) sit-to-stand; the subject was sitting on the chair and had to stand up, (3) walking; the subject had to walk three strides (starting with the right leg), (4) step up; the subject had to step up to a plateau (height: 16 cm) starting with the right foot, (5) side step; the subject took one step to the right side, and (6) back step; the subject took one step back starting with the right foot. Each run consisted of six blocks where each block was 10 movements of the same movement type, (i.e., after each run 10 movement trials were performed of each movement type). After each run, a resting recording of two minutes was performed (i.e., five recordings in total) while the subjects were standing relaxed and focused on a point on the wall four meters away. A clock was counting down to three seconds, and the subject had to initiate the movement task at this point. The movement trials were separated by 15 s. The subjects were instructed not to blink or do any facial movements during the 3-second countdown and while the movement task was performed. The experiment was performed in an electrically shielded room, and it lasted approximately three hours.

The single trial performed in [38] experiment is composed of trial start, auditory cue, start walking, stop walking, and trial end. It took about 1 minute for each trial. Each subject was asked to walk into the room after the cue sign. They walked at their natural pace while looking at the marked dot to minimize ocular and head movement artifacts. Tan Abdullah Al-Mamun et al. [37] the subjects had to perform free walking along an approximately 21-meter corridor.

Lastly, in the experiment [32] for sit-to-stand and stand-to-sit, a video stimulus that lasted for 4s to 5s and showed either the sit-to-stand or stand-to-sit video task, was presented to guide the participants to avoid the ambiguity of the instructions. The protocol began with a sitting posture, followed by 5 repeated trials of sit-to-stand and stand-to-sit tasks alternatively.

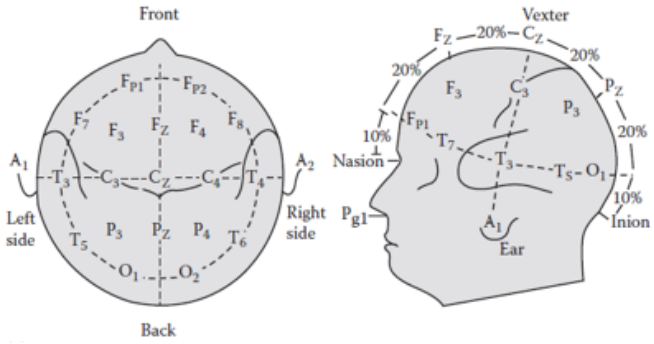


Figure 12: The 10/20international system of electrode positions for EEG. Taken from [29]

3.2.2 *EEG Setup*

According to the articles reviewed in this study, all the EEG data were collected and labeled in accordance with the extended 10-20 international system (see an example in Fig. 12). Under the 10/20 system, the skull is divided into six areas from nasion to inion with interval rates of 10%, 20%, 20%, 20%, 20%, and 10% (Fp: frontopolar, F: frontal, C: central, P: parietal, and O: occipital, respectively), and also divided into the same ratios from left to right preauricular points (T3: temporal, C3: central, Cz, C4, and T5, respectively).

Four (4) papers describe the use of a 64-channel active EEG electrode system in which 4 channels were used as EOG sensors to capture and remove eye-related artifacts using adaptive filtering algorithms. The sampling frequency in 3 of 4 papers mentioned was set to 100 Hz [43, 47, 46]. Ground (GND) and reference (REF) channels were placed on the left and right earlobe (A1 and A2), respectively. T7 and T8 channels were moved to FCz and AFz, respectively. FT9, FT10, TP9, and TP10 were used as EOG to capture eye blinks and eye movements. This modification aims to improve the decoding accuracy based on two main reasons: 1) GND and REF channels in the standard setup were close to the motor cortex and 2) EOG sensors were required in the artifact removal algorithm [43, 46]. Eguren et al. [45] study a 64-channel was used to record wirelessly at 1000 Hz from the face and scalp. Channels TP9, PO9, PO10, and TP10 were removed from the cap and used for EOG to capture blinks and eye movements; however, these data were excluded from all analyses in this experiment.

EEG data were recorded with a custom signal pre-amplifying active electrode cap and a 64- channel EEG amplifier with a sampling rate of 2048 Hz/channel (bandwidth DC - 1024 Hz), however EEG was resampled at 1024 Hz before further preprocessing [39, 40]. Jochumsen et al. [23] also recorded using a 64-channel sampled with 1200 Hz whereas [51] the sampling frequency was 100Hz.

Nienhuis et al. [11] electrical signals from 62 electrodes were recorded at a 500Hz sampling rate, while Jeong et al. [22] and Park et al. [38] used 32-channel wireless EEG data system, however in the first paper the authors digitized the frequency at a sampling of 1000 Hz and second paper at 500Hz. In addition, in contrast with the rest of the articles presented in this study, [51] EEG (64 channels) was recorded by combining two 32-channel amplifiers (sampled at 100 Hz).

From whole papers, only [32] obtained EEG signals using 11 passive electrodes with a sampling rate set to 1200 Hz, and the reference and ground electrodes placed on the left and the right earlobes, respectively. EOG signals were acquired from 2 passive electrodes positioned under and next to the outer canthus of the right eye. The system in this paper and in other 5 papers was set up to record EMG signals simultaneously throughout the experiment which is often used to modulate the gait pattern and some application to identify the onset of the movement. EMG electrodes were placed on Tibialis Anterior (TA), Vastus Medialis (VM) and Biceps Femoris (BF) of each leg [40, 39], Rectus Femoris (RF), TA, and Gastrocnemius Lateralis (GL) of two lower limbs [32], Gastroc Medialis (GM), Semitendinosus and Vastus Lateralis (VL) [11], two bipolar Ag/AgCl electrodes on the TA and BF muscles of the right leg (these muscles are known as fast activation muscles for walking)[22] and in [16] study The EMG channel was placed at the mid-belly of right leg TA muscle with the reference electrode placed on the bony surface of the right knee. TA was chosen because it is one of the muscles which activates the earliest during a gait cycle.

Table 2 shows EEG details from 16 studies.

Table 2: Experimental Protocol - Sensor Detail

Article	EEG Channels	EEG Frequency	Additional Sensors
[43], [47], [46]	64	100Hz	EOG
[39], [40]	64	2048Hz	EMG
[23]	64	1200Hz	
[45]	64	1000Hz	EOG
[37]	62	1000Hz	
[11]	62	500Hz	EMG
[22]	32	1000Hz	EMG
[38]	32	500Hz	
[51]	64	100Hz	
[32]	11	1200Hz	EOG and EMG
[12]	8	250Hz	EMG
[27]	19	200Hz	
[16]	8	500Hz	EMG

3.3 EEG DATA ACQUISITION AND PROCESSING

The EEG signals, typically ranging in the amplitude of microvolts, are captured from active electrodes and amplified through an EEG amplifier and digitized. Initially, the signals were then filtered using the following filter types: **i)** 4th order Butterworth in 5 papers, **ii)** 2nd order Butterworth filter in 4 papers, **iii)** Finite Impulse Response (FIR) in 2 papers, and **iv)** Zero phase 24th Chebyshev type II in 1 paper. In addition, different classes of the filter were applied: **i)** highpass filter from 0.1 Hz to 10 Hz was used, based on [40, 16] the value of 1 Hz is applied to eliminate DC power supply bias from the signal and according to [22] value of 10 Hz is applied to remove delta (1-4 Hz), theta (4-8 Hz) band from the signals; **ii)** bandpass filter from 0.1 Hz to 50 Hz (average) was applied in the majority of the papers, which frequency range is the most suited for gait decoding, allowing to EEG spectrum

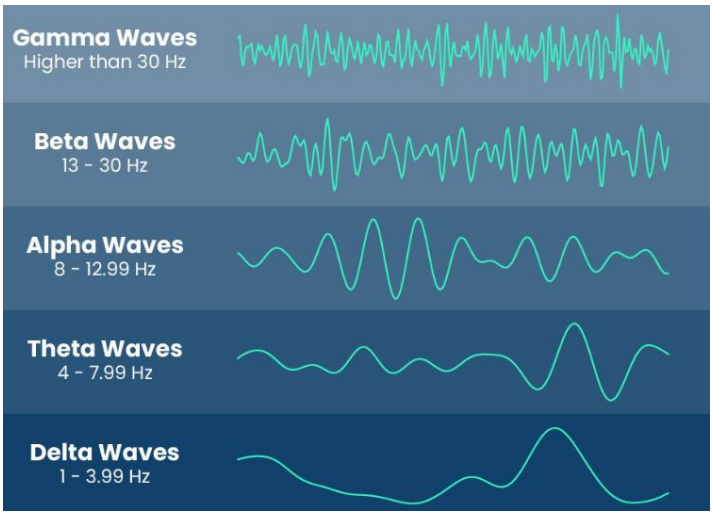


Figure 13: Brain Waves Frequencies. Adapted from [13]

be divided, as shown in Fig. 13 into delta (1-4 Hz), theta (4-8 Hz), mu (8-12 Hz), beta (12-30 Hz) and gamma (30-50 Hz) bands [37].

In Prasad et al. [51] the acquired data were filtered in the 0.1–2Hz range using a second-order Butterworth filter, and standardized by channel by subtracting the mean and dividing by the standard deviation (z-score) [46, 45, 51]. Despite taking as many experimental precautions to reduce interference of movement artifacts as possible, EEG data during walking may generally contain movement and other artifacts, either physiological or non-physiological. In particular, gait-locked artifacts overlap in time and frequency with brain activity [21].

As a consequence, when decoding gait activity from EEG signals special care should be taken to avoid exploiting task-related artifacts instead of neural correlates to decode the task itself. To minimize this risk, several preprocessing procedures have been proposed in the literature to reject movement artifacts, most of them based on independent component analysis (ICA) [24]. ICA was employed to remove the independent components (IC) (Figure 14) representing artifacts by visual inspection [38] to project the data from the scalp channels domain to the IC domain. A custom 50 Hz comb notch filter with no real poles has been used to remove the power line interference [39].

Bad channels, indicated as a standard deviation greater than 1000 µV or kurtosis of more than five standard deviations from the mean can be rejected [47], in another hand an automatic rejection of bad channels can also be performed on EEGLAB toolbox in Matlab [3] based on the probability and Kurtosis statistics of the distribution of the entire signal in each channel using a threshold of 90% [39].

After that, a common average reference (CAR) spatial filtering on the remaining signals is applied to improve the signal-to-noise ratio. A CAR filter is applied to the chunk of EEG data inside each window and the band (1-8 Hz) was extracted with a 4th-order zero-lag Butterworth filter [40]. For every sample time, CAR subtracts the mean value of all electrodes, which minimizes the uncorrelated random noise with a zero mean through the averaging process [25]. Epochs containing high- amplitude (above 100 µV) and irregular artifacts were manually removed from the data by visual inspection [39, 40], however, epochs were rejected if they exceeded $\pm$ 150 µV [23].

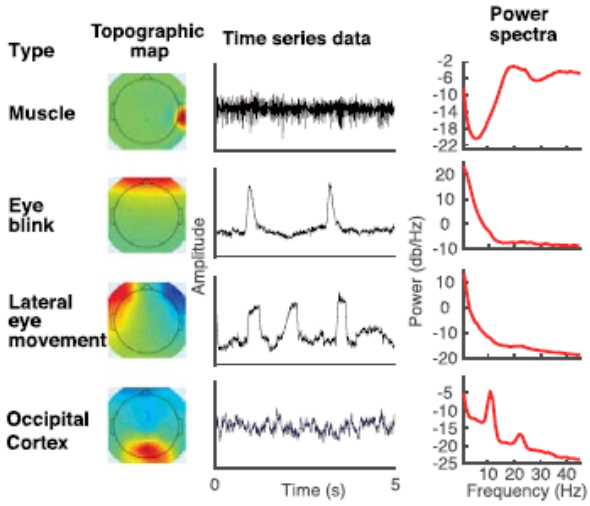


Figure 14: Examples of artifact and brain ICs.Taken from [15]

An artifact subspace reconstruction (ASR) was applied to remove high amplitude artifacts (e.g. eye blinks, muscle burst) [45]. ASR applies principal component analysis to the EEG data in sliding windows and identifies channels that significantly deviate from the baseline data containing minimal movement artifact [47]. Prasad et al. [51] compared classification accuracies with and without ASR, to assess the potential effects of motion artifacts.

Next, Reliable Independent Component Analysis (RELICA) was applied to ensure that classification performance was not affected by movement-related artifacts. The extracted ICs were then clustered across RELICA repetitions for each subject according to their mutual similarity into a number of clusters equal to the number of EEG channels. Within the RELICA framework, a quality index can be associated with each IC based on the compactness of the cluster to which it was associated [39]. The ICA weights were applied to the EEG signals coming from Step I, projecting the data into the domain of the independent component. Components that belong to stereotypical artifacts (e.g. neck muscles, eye movement) were rejected by back projecting the EEG signals to the original domain using only the components related to brain activity [40].

A summary of some procedures discussed above can be seen in Figure 15, where single-subject EEG data were first preprocessed (green box) by going through two EEG preprocessing stages, namely Step I (cyan box) and Step II (orange box). The first is more conservative and includes line noise, bad channels, and noisy epochs removal. The second applies a more aggressive rejection of artifacts to maximize the reliability of extracted ICs (blue box). ICs are applied to data processing according to the first preprocessing step.

Studies from the literature have identified neural correlates of gait in different frequency bands. The most common approach is based on Event-Related Desynchronization (ERD)/Event-Related Synchronization (ERS), which was calculated by normalizing the power in the frequency of interest from the active and passive walking by the corresponding baseline condition [11] in mu (8-13 Hz) and beta (14- 20 Hz) known as sensory-motor rhythms [39].

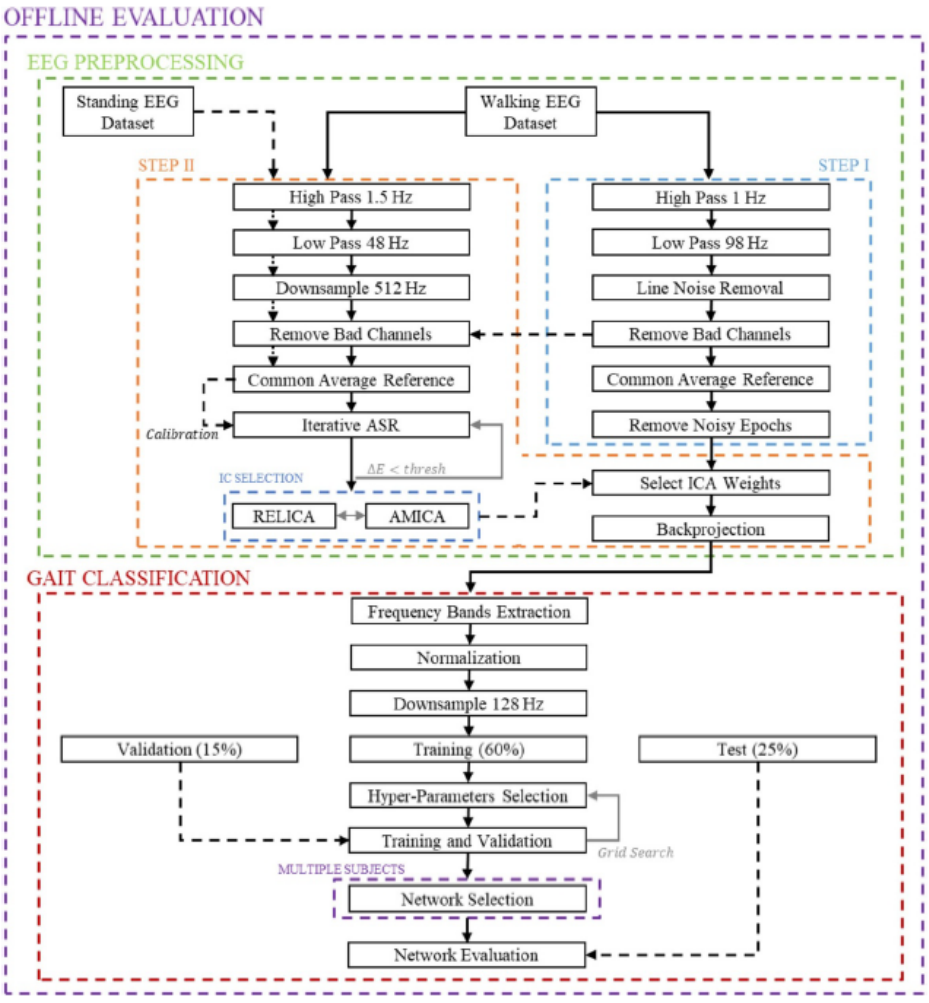


Figure 15: Schematic workflow of the offline EEG processing and classification.Taken from [39]

Other works exploited Movement Related Cortical Potentials (MRCP). An MRCP comprises two main components: a readiness potential (RP) and a movement monitoring potential (MMP). The RP is a negative cortical potential, which begins approximately 1 2 s before the onset of a voluntary movement activated over the pre-supplementary motor area (pre-SMA) or the contralateral primary motor cortex (M1). An MMP is a slow positive deflection associated with the outcome of the motor process after the execution of the movement intention. Therefore, owing to its characteristics such as potential spontaneity and early detection of user intentions, an MRCP decodes the process of movement preparation/execution based on a single trial basis [22]. in delta (<3 Hz) or theta (4–8 Hz) [39]. The MRCP signals were extracted from the EEG signals recorded during the Motor Execution (ME). The EEG signals were then high-pass filtered at 0.05 Hz (2nd order non-causal Butterworth filter). The notch filter frequency rate was defined at 50 Hz to filter out the electrical noises. Next, the EEG signals were down-sampled from 1200 to 250 Hz [32].

Besides the procedures mentioned above, two papers included other methods in their real-time operations: (1) Ravindran et al. [43] used an H-infinity algorithm to specifically remove eye blinks, eye motions, amplitude drifts, and recording biases simultaneously. The parameters of the H-infinity algorithms were kept the same as the real-time decoding. Peripheral channels were removed as they typically contain many artifactual components, (2) whereas Nakagame et al. [46] using data recorded from EOG channels, applied H-infinity filter to remove ocular artifacts and signal drifts which may be the major source ofEEG contamination. EOG and peripheral channels were then removed resulting in 50 channels being retained for designing and validating the Unscented Kalman Filter (UKF) decoder [46] [43]. An UKF was implemented as a neural decoder and its parameters were updated during BCI operations by using a closed-loop decoder adaptation (CLDA) to improve the performance in real-time [47].

3.4 OUTCOMES

The results of the majority of papers that used Machine Learning and Deep Learning approaches to train and validate the models to decode EEG signals to classify the types of ADLs will be presented in the next sections with their respective results. The table below summarizes the percentage of data split and the models' accuracy obtained from the literature review.

Table 3: Dataset Split (Train, Validation, and Test) & Accuracy

Article	Train	Validation	Test	Accuracy
[43]	80%	20%	10%	90%
[39]	80%	20%	10%	90%
[40]	60%	15%	25%	80%
[22]	50%	shared with train	50%	86% multiple channels and 81% Cz Channel
[46]	4 segment of data		1 segment of data	43%
[32]	14 trials		1 trial	82.73%
[11]	90%		10%	89.9%
[38]	7 trials		1 trial	83.4%

[37]				CV = 77.8%
[12]	75		25	83.3%
[27]	70		30	86%
[16]				85.65%

3.4.1 *Gait Deconding from EEG with LSTM*

Chisari et al. [39] adopted an extensive offline preprocessing stage as shown in Fig. 15 to guarantee that the optimization of network hyper-parameters, as well as model performance, were not, or minimally, influenced by artifacts time-locked with stepping frequency.

Regarding cortical correlates of gait, different frequency bands of the EEG spectrum (δ band (1-4 Hz), δ and θ bands (1-8 Hz), from δ to μ and low β bands (1-16 Hz), only μ and low β bands (8- 16 Hz), high β band (24-40 Hz)) were extracted as neurophysiological features to decode walking activity.

All the bands including low frequencies (1-3, 1- 8, 1-16 Hz) are characterized by the presence of a MRCP prior to each swing phase. In contrast, the ERDS ranges (8-16, 24-40 Hz) present a desynchronization—smaller absolute signal amplitude—during the swing phase with respect to the stance phase [39].

During the offline evaluation, the effect of LSTM hyper-parameters selection we examined on the decoding performance. Performance was evaluated in terms of accuracy and cross-entropy loss of the network trained on 60% of the whole dataset (training set) in the 1-16 Hz frequency band, since it is the widest band considered in this study, and used to predict 15% of the whole dataset (validation set). To evaluate the performance with different frequency bands, we considered a network with 2 LSTM layers, 250 LSTM units in the first layer and 100 LSTM units in the second layer, trained with constant learning rate of $10-3$ (none learning rate schedule), achieving on average $92.8\pm3.1\%$ of accuracy and 0.14 ± 0.12 of loss in validation [39].

Thus, the three classifiers were trained for each subject with the EEG signals filtered in the 1-8 Hz frequency band, and their performances were tested on the 25% of the completely unseen data in the test set.

Chisari et al. [39] proposed in their study a deep learning-based classifier for decoding of gait events from scalp EEG signals, which their method relies on the memory capabilities of a specific type of recurrent network, namely LSTM, to learn time dependencies within the data. Indeed, the brain activity generated prior to the swing phase of each leg overlaps in time with the stance phase of the contro-lateral leg (Wagner et al 2014). This fact may prevent clear discrimination of the dynamics of each leg, as shown by the ineffective performance of memory-less classifiers in the decoding of right and left gait events separately with respect to the decoding of the two legs together.

3.4.2 *Gait Intention Using Spatio-Spectral CNN*

This study was conducted entirely online. The accuracy is the average of all test sets from k-fold cross-validation. The subaverage is the average accuracy of each subject group and the total average is the average of all subjects. The classification accuracy using spatio-spectral CNN model was 83.4% on gait/stand state recognition. In another

hand, the accuracy for the gait and stand intention recognition reached 77.3% and 77.7%. The findings suggest that selecting the data's characteristics is critical while building a CNN architecture. Park and Park [38] used only 19 EEG channels from the participants, which was insufficient to reflect spatial features. A deep learning model with improved spatial and temporal features of EEG data should increase performance.

3.4.3 *Real-Time EGG-based BCI*

The paper results analyzed below were performed using online preprocessing and offline decoding combination. Nakagome et al. [43] aimed to improve the decoding accuracy and robustness for the lower limb decoding using EEG from algorithm perspectives. Accurate lower limb decoding is important for controlling exoskeletons and neuroprosthetics for better usability and systems.

Unscented Kalman Filter UKF showed its superior performance in early convergence with a smaller number of samples and when evaluated from the r-value perspective. On the other hand, UKF showed its vulnerability when evaluated from the R2 score and also when a channel is perturbed.

UKF decoders tend to perform better with fewer samples, but with larger tap sizes, other algorithms may outperform UKF decoders. This might be one of the baselines used to determine the number of samples to be utilized in real-time decoding.

It was not previously known how sampling frequency affects performance. This study looked at this issue using a sample-by-sample decoding approach and discovered that performance may be improved. Although data can be recorded at the highest sampling frequency, the delta band passed features can technically reduce the sample size to 20Hz if there is a sufficient frequency range for reconstruction.

With this approach, [43] could technically also increase the future prediction time from 1 ms (when 100 Hz) to 5 ms (in 20 Hz) with the same decoding scheme. We also showed that the performance could actually improve.

In particular, Luu et al. [32] found sustained α/μ suppression in the Posterior Parietal Cortex (PPC), and Inferior Parietal Lobe (IPL) regions and significant decreases (ERD) in the β band, indicating increased cortical involvement in the walking task. Besides, the results also revealed a substantial increase in cortical activity in the low frequency (Δ bands) in the Anterior Cingulate Cortex (ACC) region, demonstrating the potential benefits of using closed BCI, which employs cortical networks involved in error monitoring and motor learning. In addition, β suppression was observed in the ACC, PPC, and IPL, as well as low γ in the and superior temporal gyrus STG. These findings suggest that the closed-loop BCI system encourages spontaneous control of human gait.

The system may also help to better understand cortical dynamics during walking with a closed-loop BCI system.

3.4.4 *Decoding Sensorimotor Rhythms*

Classification performances differentiating walking from baseline for both healthy participants and stroke patients were above 93% and 89%, respectively [11]. The classifier weights revealed that the key brain signals contributing to this performance were ERD in the mu rhythm, which was more bilaterally distributed, and ERD in the beta and low gamma bands, which were more centro-medially situated, as previously described. In comparison to

the baseline, brain signals in the beta band centro-medially situated (Cz electrode) were substantially different between passive walking and active walking (with an exoskeleton). The mu-band effect appeared to be more lateralized over the hand parts, whereas beta/gamma appeared to be more medially concentrated over the foot areas. This might be a result of implicit hand activity or volume conduction effects causing central mu-cancellation. It is commented in article [11] that some studies have measured cerebral activity during actual gait, showing that walking increases cerebral activity bilaterally in the medial primary sensorimotor cortices, the supplementary motor area, and the prefrontal cortex.

A canonical correlation analysis (CCA) has shorter computational time and can be used experimentally, and because of that CCA can be easily implemented during online BCI.

Although the main goal of work is not to develop a BCI for gait rehabilitation, it is relevant to understand a collect all information about the procedure needed first to assess the possibility of decoding walking intention from brain signals during lower limb rehabilitation by a robot, because maybe it will contribute in cases and experiments where the subjects are non-healthy.

When stroke patients are immersed in a robot-gait training system, [11] offline classification findings reveal that it is possible to discriminate between resting and walking with high accuracy. García-Cossio et al.[11] demonstrated in the previous study, that the online implementation of this approach (i.e., using a logistic regression classifier to distinguish walking intention from resting) can be used to control a treadmill in a binary mode (on/off) using EEG signals from healthy volunteers, with high accuracy rates. These findings demonstrated the viability of establishing a BCI for gait rehabilitation.

3.4.5 *Decoding Movement-Related Cortical Potentials*

Jeong et al. [22] performed single-channel and multiple-channel decoding. As the data reveal, the performance of the multi and single-channel techniques differed very little (less than 5%). This suggests that for a BCI, it may be possible to detect movements with only a single EEG channel; hence, assistive technology for giving afferent input can be activated. Furthermore, it was proven that the Cz channel outperformed other neighboring channels in terms of discriminant performance. Their experimental findings are connected to the lower-limb movement, which is one of the explanations for the Cz channel activation in this investigation (i.e., walking). Besides, each electrode's increased cortical density during a body movement is reflected across the motor cortex according to the body region. The lower limb is intimately associated with the motor cortex's core position near the Cz channel. The single-channel BCI technique with optimum sites results in the utilization of a minimal number of EEG feature vectors for a final choice, possibly reducing computing time for model training and real-time situations.

Jochumsen and Niazi [23] identified the movements that could be classified with respect to the idle activity with accuracies in the range of 80%–90%. For the movement type discrimination, 54% of the movements were correctly classified with the stand-to-sit and sit-to-stand movements being the most discriminable movement types. The results indicate that it is feasible to identify and categorize movement intentions linked with functional motions, which might be utilized to construct a BCI that introduces task diversity in neurorehabilitation.

The detection of movement intents was measured by classification between movement intentions and idle activities. The epochs were recovered with prior knowledge of when the movements happened; consequently, in a real-time BCI decoder, this information will be unavailable, and fewer movements may be successfully recognized. The detection algorithm may be tweaked to be less cautious, but this increases the risk of false positives. One possible solution is to build a cue-based BCI system in which the detector is only active during specific time frames [23].

3.4.6 *Neural Decoding of Gait in Developing Children*

At least 6 of 13 articles in this analysis, showed that meaningful changes in EEG during walking occur at low frequencies (< 10 Hz) [43, 47, 45, 37, 51, 46]. EEG in the delta band (0.1 – 3 Hz) was used for neural decoding walking; thus, the EEG features correspond to time-domain amplitude modulated (AM) potentials in the delta band [43, 47, 45]. EEG signals in this band have been demonstrated to carry relevant information on human gait. Slow cortical potentials in the delta band EEG have also been used with a linear decoder to interpret human gait during treadmill walking, and the results were equivalent to invasive BCI techniques. Furthermore, these studies have revealed that a real-time closed-loop BCI for human treadmill walking from an EEG signal in the delta band has high decoding accuracy.

Because EEG signals carry not only efferent motor control during human walking but also multisensory afferent feedback, they need to be observed carefully. It is a difficult task to extract relevant neural features from scalp EEG data for the interpretation of human gait. To deal with this, [45] employed a causal model that predicts current kinematics based on EEG signals before the current motor command. Moreover, feedback signals include delays related to the present motor command, being afferent feedback modified in future EEG signals that are not included as inputs in the designed causal model.

Finally, brain signals are very dynamic, and it is crucial in the development of a robust neural decoder. Typically, neural decoders are taught offline by fitting neural signals to real motions. This technique, on the other hand, ignores participants' brain dynamics while moving from open-loop to closed-loop BCI, which often results in lower online performance.

3.5 DISCUSSION

The literature review was thoroughly evaluated for studies including BCIs for directing lower-limb robotic systems. The experiments frequently entailed categorizing discrete state directives such as walking, stopping, turning, standing, and sitting. It was observed that just a few EEG denoising techniques were used, or that they were not thoroughly tested. For neural classification, many neural features and decoders were employed. Overall, the systems' performance is promising, but it is distant from practical applications due to the small sample pools used, potential safety hazards, and other difficulties. Moreover, in all papers, there were no records regarding experiments with loss of balance in the elderly, which is the main subject of this study. Summarizing the results obtained from the literature review allows inferring that:

(1) Deep learning-based model for gait decoding enables to the exploitation of non-invasive brain activity recordings to detect walking events (i.e. swing and stance phases) in healthy people. Overall, Tortora et al. [39] outcomes suggested the use of a memory-based deep learning classifier as a possibly more effective input to brain-controlled rehabilitation or restoration devices when used in real-time and with the ability to control each leg individually.

(2) Park et al. [38] developed a spatio-spectral CNN model with a gait state identification accuracy of 83.4% We also achieved 77.3% and 77.7% accuracy in gait and stand intention identification, respectively. Based on the results, this study was able to decipher the gait intention of individuals with subacute or chronic strokes as well as healthy volunteers.

(3) 4 papers [47, 43, 46, 45] showed that motion artifacts are negligible in delta band frequencies at the gait speeds used in this study (i.e., 1.6km/h).

(4) Walking intention decoded from cortical patterns formed in the sensorimotor strip during robot-assisted gait training in both healthy volunteers and stroke patients with minor lower limb disability could be revealed in [11]. The modulation of low gamma activity in the central midline regions was shown to be linked with the stages of the gait cycle in healthy volunteers but not in stroke patients.

(5) Lower extremity functional motions generate MRCPs that are evident in the EEG. The MRCP with the greatest amplitude was induced by the stand-to-sit exercise, while the MRCP with the least amplitude was evoked by walking. It was demonstrated that motions could be recognized in idle activity and that individual movement tasks could be categorized [51]. Although a direct comparison with the prior studies is difficult due to varied experimental procedures and methodology, MRCP detection obtained a high grand-average decoding performance of 86% for all patients in a single session[22].

4

DECODING LOWER LIMB LOSS OF BALANCE FROM BRAIN SIGNALS: DESIGN CONCEPTION

Currently, as demonstrated by the literature review described in Chapter 3, very few studies direct their efforts towards loss of balance caused by slip-like perturbation or any others. Additionally, although some studies have already shown reasonable results to identify loss of balance, the data collected is generally from participants that stayed stand on a balance platform, and then the platform introduced balance perturbations. Thus, there is a need to obtain EEG data of slip-like perturbations during walking on the treadmill and analyze the brain signals response to these events.

This chapter discusses the conceptual design of this dissertation. The analysis of the state-of-the-art presented in the previous chapters, in addition to gathering a set of essential theoretical knowledge for the practical chapters presented next, also allowed for the identification of some gaps in the scientific literature regarding the analysis of brain signals in response to slip-like perturbations and their importance in developing AI-based algorithms to reduce the risk of falling. As a result, the answer offered in this dissertation is split into two parts: **i)** experimental data analysis of experimental data for studying how brain signals react to slip-like perturbations.; and **ii)** using AI-based models, identify perturbation and non-perturbation events considering the collected data and resulting findings.

4.1 INTRODUCTORY INSIGHT

WHO reports emphasize the high global incidence of falls while also warning that this incidence may rise in the coming years due to the world's aging population. In addition to economic, social, and physical concerns, falls in the elderly produce a long-term sensation of anxiety that prevents them from doing everyday duties, resulting in decreased autonomy and physical ability, and therefore increasing the risk of future falls. As a result, different technical solutions to this challenge are required.

Therefore, this dissertation aims to contribute to the study of the brain signals in response to slip-like perturbations through the analysis of EEG experimental data, previously collected in – University of Minho. The experimental data analysis performed in this dissertation will allow for a general and complete understanding of this brain signal response, as well as the identification of signal behavior for loss of balance, which may be considered in the development of technological solutions that act on this problem.

Obtaining quantifiable data from the examination of slip-like perturbations that can be utilized as inputs in the process of training AI-based algorithms would increase the promising outcomes related to BCI development for fall prevention purposes. Additionally, the work developed has not only the objective to select the best fit AI-Based algorithm for falls prevention but rather to gather information of interest for the design of fall prevention devices allowing to produce of scientific outputs not only for the development of this dissertation but also for other works under development in the same field. Independently of this, it is also an objective to present a comparison among Deep Learning algorithms in comparison to computer performance.

4.2 RESEARCH HYPOTHESIS

The investigation activities that were developed within the scope of this dissertation are based on the following technological hypotheses.

- The analysis of the subject's EEG data pre-processed signals can allow the identification of the event of balance loss events in relevant EEG channels during slip-like perturbation [48, 10, 10]. (Goal 4; Chapter 5)
- This project also considers the ability of AI-Based models in recognizing the loss of balance events during slip-like perturbations and the existing hypothesis that brain signals decoding has the potential to prevent slip-related falls [35]. (Goal 5; Chapter 3)
- The feasibility of using a single EEG channel (Cz) to decode the brain signal referring to slip-like perturbations, so that the accuracy of the model is greater than 85% [10]. (Goal 5; Chapter 3)

4.3 PROJECT CONCEPTUAL DESIGN

In order to address the previous issues in the field of elderly fall prevention, this dissertation aims to provide quantitative information for the development of anti-fall BCI decoding. To this end, the approach presented in this dissertation is divided into two main points: i) Brain signal analysis of slip-induced falls previously collected at Birdlab – University of Minho and ii) Classifying slip-like perturbation events based on quantitative data obtained from the previous point. The strategy that is presented in this chapter follows the type of architecture that is usually followed by the literature, and it is summarized in Figure 16.

The first block of the proposed tool (Data Acquisition and Processing) represents the proposed data acquisition through an experimental protocol and processing techniques to clean and normalize the data that were used to train the AI models. The second block presents the deep learning models that were built to classify the data extracted from the brain signals. Thus, based on the results obtained from the classification models, disturbance perturbation was identified and the results are intended to contribute to the development of BCI Devices.

Due to the fact that the signal originates from perturbation events, several artifacts from muscle and postural movements can alter the data. As a result, numerous methodologies were used during experimental trials to complement this analysis and make changes based on the signal origin.

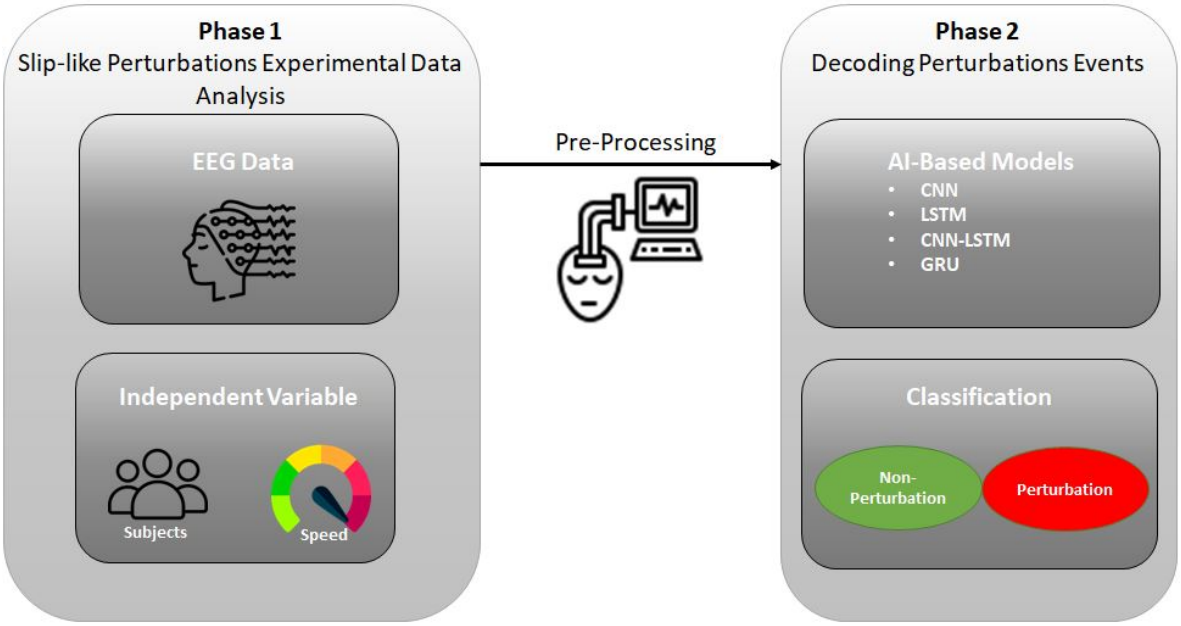


Figure 16: Schematic of the Project Phases.

Four (4) artificial neural networks were trained and tested, and their results were discussed. Thus, the classification for perturbation and non-perturbation focuses on the human response to slip-like events during walking on a treadmill at 1.6km/h and 2.5km/h.

4.3.1 *Slip-like perturbations experimental data analysis*

For the development of phase 1, it was essential the collection of slip-like perturbations' experimental data was carried out prior to the beginning of this dissertation. From the data collection protocol, it is possible to analyze several EEG data. Additionally, the initial framework using EEGLAB was created to run the pre-process tools over EEG raw data. Firstly, the study of decoding EEG signals for ADLs contributes to understanding which techniques and tools must be applied to preprocess data, however, the majority of literature removes the artifacts from the signals, which implies the removal of some time windows regarding these artifacts.

In turn, the analysis of independent variables such as gait speed, for example, can increase artifacts during walking as soon as the speed increases and their influence on the recovery process. Through this analysis will be possible to understand the brain signal response to risky scenarios with variable conditions, making it challenging to find solutions for fall prevention. Outline the EEGLAB framework, and tools to segment the data must be applied to obtain the PEP waveform from EEG signals. The obtained quantitative data as outputs of this analysis are also objective to be considered during the analysis in order to facilitate classification for loss of balance events addressed in 4.3.2. EEG data analysis for slip-like perturbations will be presented in Chapter 5.

4.3.2 *Decoding perturbations events*

Since the development of BCI is one of the currently growing approaches to fall prevention [30], the second major goal of this dissertation is to classify slip-like perturbations events during walking on the treadmill so that can contribute to the development of a device for fall prevention. A framework script in Python developed by BiRDLab was customized to work with EEG data collected from experimental trials. As previously mentioned, the classification for loss of balance events was based on the quantitative outputs from the previous addressed data analysis.

Before feeding the framework with the data, it is necessary to label the segmented data and to establish strategies to split the data, to compare different experimental setups, such as: **i)** same subject with different speed, **ii)** same speed with different subjects, and **iii)** all subjects and all speeds. The preceding described methods are ways to lessen overfitting on a dataset because of its size and the dearth of data on perturbations. Additionally, regarding this section, decoding brain signals using AI-based models to prevent falls due to slip-induced perturbations will be presented in Chapter 6, which purpose is to perform comparative analysis and identify the best hyperparameters and the best fit artificial neural network architecture considering the possibility of decoding brain signals from EEG data, besides the model has good computer performance because perhaps it will contribute for a real-time solution.

4.4 OUTCOMES

The aging of the world's population leads to an increase in the prevalence of neurological illnesses such as dementia, Parkinson's disease, and cerebrovascular accidents. In addition to other implications, the aforementioned disorders cause people to be less mobile, increasing the likelihood of falls. Aside from the economic ramifications for the world's healthcare systems, these incidents result in post-fall injuries. For all these reasons and considering the need to develop alternative strategies for fall prevention, this dissertation intends, based on a brain signal analysis, to gather a set of quantitative information, considering EEG data, for training the AI-Based models to identify the slip-like perturbation with the purpose of get the natural human response to these events classifying as perturbation and non-perturbation.

Two major phases were selected and presented in this Chapter: **(i)** slip-induced experimental data analysis which design is based on good practices to collect EEG data and preprocessing which is applied to remove the artifacts and select relevant features, as mentioned on previous Chapter, allowing to detect loss of balance during walking on a treadmill will be addressed in Chapter 5; and **ii)** decoding perturbations events for fall prevention robotic devices which AI framework, developed by BiRDLab team, was adapted from this project, and selection of the deep learning architecture was based on the literature review presented in Chapter 1 and 2, where DL-based models and their computer performance will be evaluated and discussed in Chapter 6.

4.5 DISCUSSION

This dissertation aims to gather a set of information useful to the development of wearable robotic devices to fall prevention, based on a comprehensive study of the brain signal response to slip perturbations. In this chapter, an architecture was idealized based on a literature review of collected and processed data for ADLs and its gaps related to running experimental protocols for loss of balance.

The literature reviews presented in Chapter 3, in addition to allowing the understanding and variables of interest to be analyzed during the study of experimental data from slip-like perturbations, also allowed for the identification of some gaps in the literature, which will be filled as much as possible during the analysis of experimental data collected at BiRDLab.

The experimental data analysis' outcome will be a fundamental step in identifying the loss of balance signature in EEG the signals. Additionally, the EEG data analysis performed will provide a more comprehensive understanding of the brain signals response to slip-like perturbations considering different subjects and speeds of gait, thus filling the gaps found in the review of the existing literature [35, 10, 31], which results for loss of balance were obtained from experiments where the subjects stood on a platform and postural reactions were applied in two horizontal translations (forward and backward) and two vertical translation (toes up and toes down). This outcome will be achieved in Chapter 5.

Finally, the classification for perturbation and non-perturbation using different Artificial Neural Networks Architectures for fall prevention was also a gap determined after the analysis of the existing literature [10, 31], which the present dissertation intends to overcome, besides to compare the AI-based models in the face to computer performance to evaluate their effectiveness, which outcomes will be discussed in Chapter 6.

In this Chapter research hypotheses that were considered to support the investigation described were also addressed as well as the outcomes expected to achieve in this dissertation.

5

INDUCED LOSS BALANCE DATA ANALYSIS

The purpose of this chapter is to present a comprehensive analysis of brain response to slip-like perturbations considering EEG data in different walking speeds and considering variable slip perturbations. This chapter will also address the experimental protocol used for data collection as well as how the data were processed. The primary output of this chapter is quantitative data that may be used to help determine the target requirements for balance loss decoding by analyzing cerebral cortex signals of the response to slips. The methods to set up and run an experimental protocol, data pre-processing, and recognition of brain reactions caused by balance disturbances as well as the data segmentation and data labeling will be addressed in Section 5.2. The main results related to Perturbation Evoked Potential will be analyzed in Section 5.3. In the end, Section 5.4 presents a discussion regarding results from the pipeline deployed in the EEGLAB framework (Matlab).

5.1 INTRODUCTORY INSIGHT

As previously mentioned, the development of balance loss decoding requires, as an initial step the analysis of the natural brain signals response of humans to slip events, in order to gather information capable of determining the characteristics of actuation capable of mimicking this response. Therefore, an experimental protocol was designed and conducted by BiRDLab team, to collect brain signals from the cerebral cortex and study these signals in order to identify and obtain the variables of interest in slip-like perturbation responses analysis.

In this analysis, a pipeline was deployed in EEGLAB, whose goal is the data cleansing and normalization by applying pre-processing techniques to extract relevant features and evaluating their importance for the effectiveness of a successful brain signal response after a slip perturbation based on literature review (Chapter 3) and later be used in the decoding and classification to predict slip-like perturbations, which one will be presented on next Chapter.

5.2 METHODS

The research covered in this chapter sought to understand the brain signals' reactions to slip-like perturbations by collecting EEG data and processing it through data cleaving and normalization processes. The methods applied to obtain a comprehensive analysis will be addressed in this Section sorted by the following topics: **i)** the

experimental setup to data acquisition through EEG system inside a controlled environment (Subsection 5.2.1); **ii)** pre-processing the raw data applying filter and techniques to improve signal-to-noise ratio (Subsection 5.2.2).

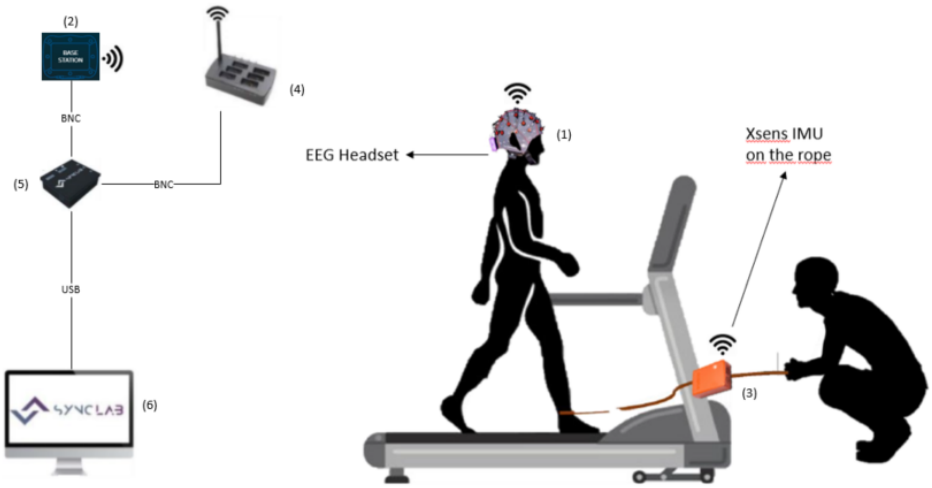
5.2.1 *Experimental Procedures*

In order to study the brain signals in response to slip-like perturbations, experiments were conducted at BiRDLab, University of Minho. Initially, in order to proceed with the experiment, the subjects had to present: i) healthy locomotion; ii) total postural balance; iii) over 18 years old; and iv) body mass less than 135 kg. Individuals who: i) had any disease (mainly neurological) or deficit that affected locomotion were excluded; and ii) have recently undergone surgical procedures that affect mobility. Based on the prerequisites three healthy male participants (age: 26 $\pm$ 5; height: 1.80m $\pm$ 3.85cm; weight: 71 $\pm$ 3.3 kg) were selected for the experimental protocol. All the subjects that participated in these protocols presented the right dominance. All participants provided written informed consent and voluntarily accepted to participate in the experimental trials. Each participant performed the qualitative assessment of the preferred foot by completing the Waterloo Footedness Questionnaire.

To provide data to better understand the brain signals response due to balance loss events and record the instant of the slip perturbations, two sensor systems were used in this protocol. Wireless EEG Headset (g.NAUTILUS PRO), 16 channels (active wet electrodes) to record brain activity in a controlled environment; and IMUs from Xsens MVN Awinda which is composed of 17 IMUs, however, only one was used to record acceleration caused when the rope is pulled to mimic a slip-like perturbation event when the subject is walking on the treadmill. The data from the brain activities were collected through EEG system at 500Hz, and 1 IMU at 100Hz was used to record the acceleration caused by the action of pulling the rope.

Subjects used a safety harness device during the experiments to avoid falls in the event of an irreversible Loss of Balance (LOB). This method consisted of a vest that was rope-attached to a structure in the ceiling. The length of the rope was modified to ensure that there was at least 15 cm between the knees and the treadmill belt. This step was carried out by instructing individuals to elevate their feet, resulting in the application of complete body weight to the harness system.

To achieve synchronized data gathering from sensors, Sync Lab Desktop for Windows OS, created by BiRDLab, was utilized to synchronously start and stop data collection from previously listed sensors. The Desktop program sends electrical trigger signals. The former is delivered via Syncbox, a previously designed hardware interface that links to the Xsens and EEG devices via direct USB connection. The following diagram describes the experimental setup that was used to obtain data.



1) EEG headset. 2) EGG base station to capture the signals from the headset and send them to SyncLab. (3) Xsens IMU to register the acceleration regarding pulling rope. (4) Xsens base station, which establishes communication between the Xsens IMU and Synclab. (5) Sync Box to synchronize the base stations. (6) PC with SyncLab to collect and process the sync data;

Figure 17: Experimental setup for balance loss data collection.

Subjects were instructed to manage unexpected slip-like disturbances while walking on the treadmill, and they were not informed about the protocol in order to avoid any prior bias in their brain activity response. A concealed rope was hooked up to the subject's ankle at heel striking time, and a second person manually marked the perturbations to be utilized later to adjust the registers with Xsens IMU data. Because the rope was constantly linked to one of the subjects' feet during all of the trials, the participants did not know whether a disturbance would occur or not. The experimental protocol was applied to volunteers in two (2) different treadmill velocities (1.6 km/h and 2.5 km/h). The volunteers had to wear the EEG cap while walking on the treadmill, and the protocol was performed for each speed per subject.

- 1 treadmill inclination (flat – 0º)
- time per trial: 20 - 30 minutes
- 2 situations (perturbation and non-perturbation)

According to a literate review of Chapter 3, the volunteers had to walk on the treadmill for 20 to 30 minutes for each speed. For each experimental session (1.6 km/h and 2.5 km/h) where perturbations were delivered, the operator applied a minimum of 30 perturbations at random moments during walking. Both phases (Non-perturbation and perturbation) occurred intercalated within a duration period of the 30s to 60s.

5.2.2 *Data Pre-processing*

The events (slip-like perturbations) were automatically recorded by Xsens whose magnitude of the acceleration vector was used to indicate the occurrence of provoked perturbation for both speeds. In order to detect any spikes that represent wrong situations such as dropping the rope, or a rope hitting the treadmill, as a solution, it was suggested to use the synclab to record the occurrence of events manually. To synchronize these events, a script was created in Matlab to allow this adjustment and the validation of the records before starting the pre-processing of the data. The result of 1 trial is shown in Fig.18.

After the data acquisition, the pre-processing of EEG data was carried out using custom-made scripts (Appendix D) in EEGLAB toolbox [3] running in MATLAB. The EEG dataset has 19 rows (**row 1**: time (2 ms for each register); **rows 2 - 17**: EEG channels; **row 18**: null; **row 19**: events related to perturbation), and each column represents the values from brain signals obtained by the electrodes, thus, the number of columns is directly related to the duration of the trial.

Firstly, the irrelevant attributes were deleted (rows 1 and 18), and the registers in row 19 were imported as events. Afterward, as described before, it is necessary to check the time recorded in the dataset with the time recorded by Xsens and then adjust the time when the perturbation event occurred. It is important to note that the EEG system frequency is 500Hz and Xsens is 100Hz, and this may affect the register of the event time in milliseconds. Furthermore, the information about channel locations needs to be imported to be used when plotting the graphics and analyzing the channels' significance.

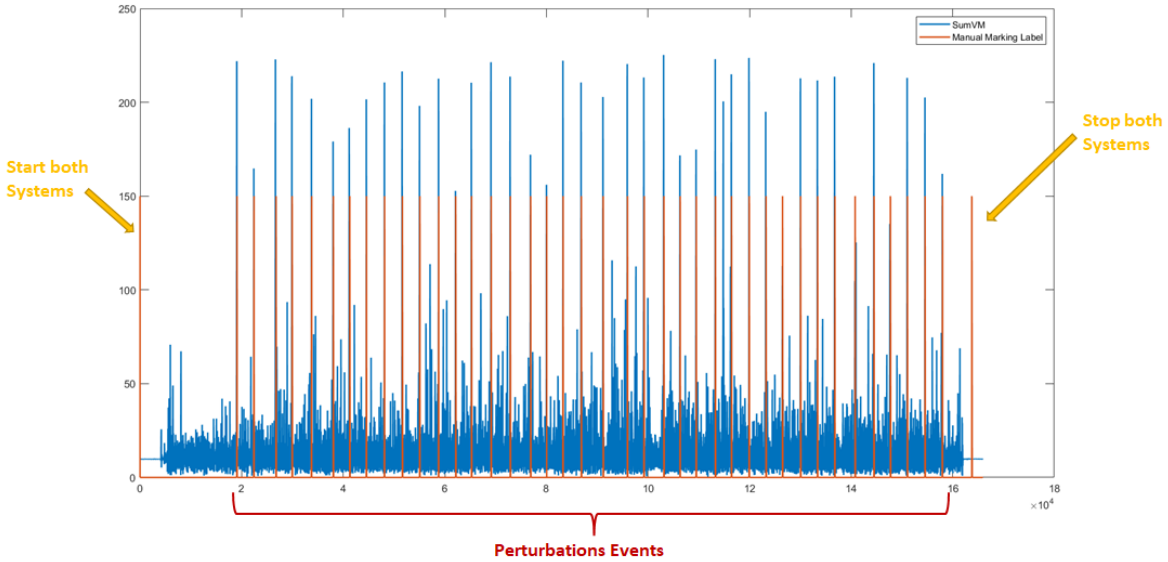


Figure 18: Comparison between the manual label and Xsens IMU acceleration vector magnitude to perform label correction (1 Trial - 40 Perturbations)

As EEG signal is highly prone to noise and also is non-stationary, EEG recordings can be understood as a mixture of independent cerebral and non-cerebral sources called artifacts (e.g., ocular, muscular, etc.). These

artifacts are considered disturbances in brain electrical signals, thus a pre-processing procedure to remove them from the recorded EEGs must be implemented to allow the extraction of valuable event-related information. The pre-processing steps used to process EEG signal are summarized in Fig.19 and the images of the results of the signals obtained after each step of the pre-processing are shown in Appendix A.

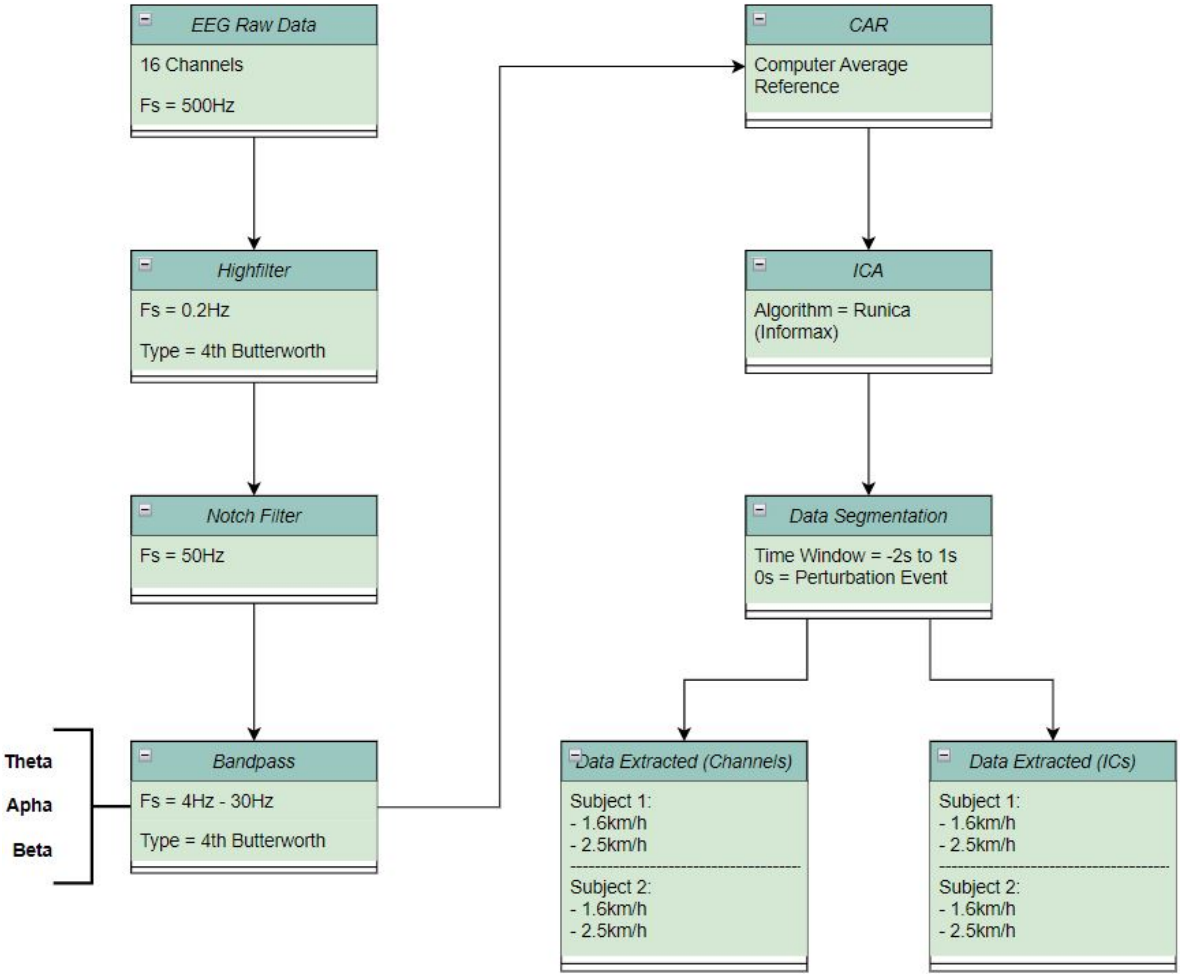


Figure 19: Pre-processing Framework Pipeline

Initially, based on papers related to ADLs, in order to cut off the low-frequency drift the data were high-pass filtered at 0.3 Hz using a 4th-order Butterworth filter, and line noise was attenuated at 50 Hz by using a notch filter. However, analyzing the data after these steps, it was not possible to obtain PEP patterns. Thus, a band-pass filter between 4Hz and 30Hz was applied to the data removing low frequencies (applied in motor imagery or stable movements) and frequencies above 30Hz avoiding the muscular artifacts that belong to the natural walking and postural recovery after a balance loss [35].

The processed EEG signal was re-referenced using a common average as a part of the pre-processing process. CAR is commonly used in EEG, where it is necessary to identify small signal sources in very noisy recordings [1]. The re-referenced EEG signal was then ready for ICA operation, in which independent components representing

independent EEG source signals were extracted using Infomax [6]. According to papers [31, 49, 35, 48, 10] components centered around CZ and distributed over frontal, central, and parietal areas (FCz, C3, C4, and CPz), have a strong relation to the detection of balance loss. As shown in Appendix A Fig.26 the EEG headset does not have frontocentral and centroparietal electrodes, and due to the fact of electrodes are fixed, it was decided to test four (4) approaches: **i)** use all channels; **ii)** remove all channels except Fz, C3, Cz, C4, Pz, which ones are closer of central area, **iii)** Fz and Cz, because during some trials was observed the Fz channel obtained similar responses related to provoked perturbation as presented in literature [31, 49, 35, 48, 10], and **iv** Cz was used.

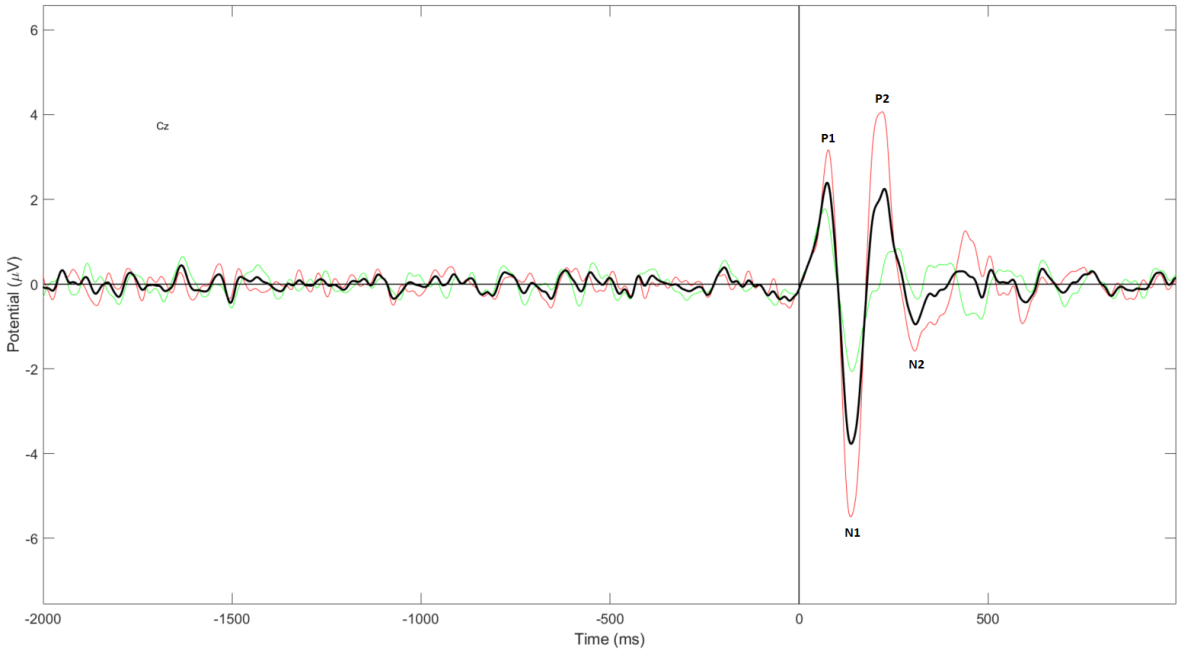
To study the event-related EEG dynamics of continuously recorded data, the data epochs time-locked to events of interest were extracted (for example, data epochs time-locked to onsets of one class of experimental stimuli). Thus, the EEG signals were segmented into epochs time-locked to perturbation onset (epoch range: -2s to +1s), wherein 0 ms is included in the window, corresponding to the first sample at the perturbation onset. In addition, due to the poor quality of the data obtained, one subject was discarded. Thus, two remaining male subjects were kept to analyze data.

5.3 OUTCOMES

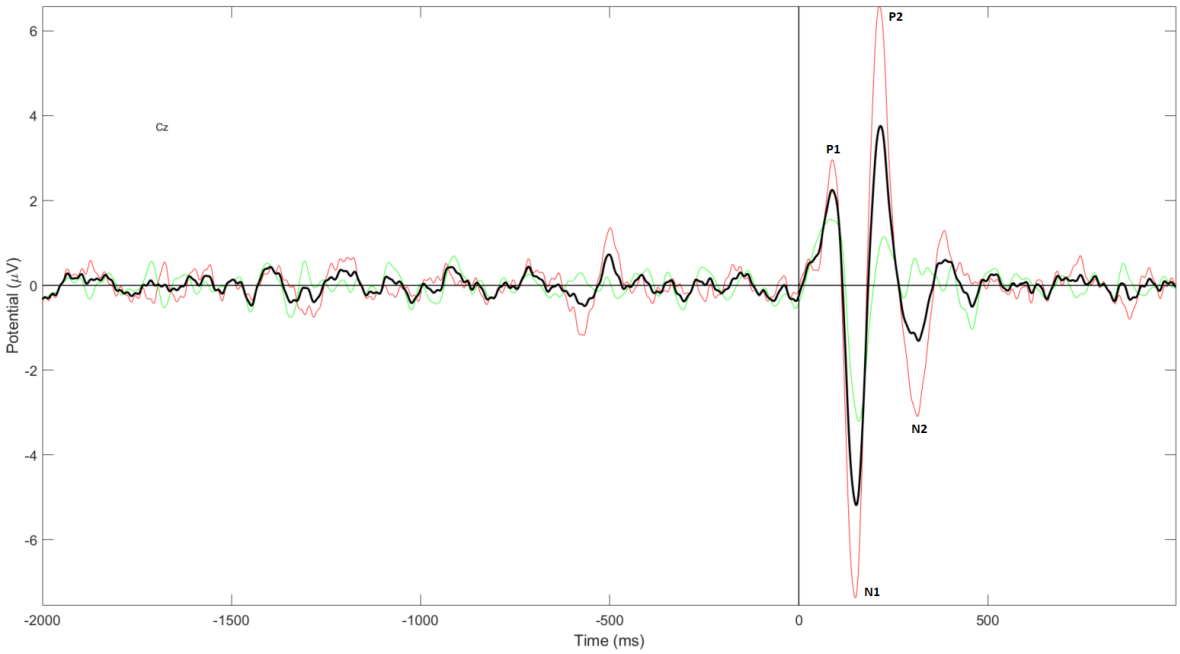
During the occurrence of a slip-like perturbation, the grand average shows (Fig. 20) a strong negative shift (N1 component of PEPs) starting shortly after perturbation onset and peaking 138 $\pm$ 3 ms after perturbation onset with an average amplitude of –3.77 $\pm$ 1.72 µV for treadmill speed = 1.6km/h, and peaked on average 150 $\pm$ 5 ms after perturbation onset with an amplitude of –5.16 $\pm$ 2.08 µV for speed = 2.5km/h. The negative peak is followed by a strong positive rebound, the P2 component of PEPs, that peaked on average 226 $\pm$ 21 ms after perturbation onset with an amplitude of 2.25 $\pm$ 1.61 µV for treadmill speed = 1.6km/h and peaking 216 $\pm$ 5 ms after perturbation onset with an average amplitude of 3.76 $\pm$ 2.72 µV for speed = 2.5km/h.

Furthermore, the P1 component was also detectable in the grand average. P1 started shortly after perturbation and peaking 74 $\pm$ 4ms after perturbation onset with an average amplitude of 2.35 $\pm$ 0.7µV for treadmill speed = 1.6km/h and peaked on average 88 $\pm$ 3 ms after perturbation onset with an amplitude of 2.25 $\pm$ 0.7 µV for speed = 2.5km/h. On the other hand, the negative peak (N2) which succeeds the later positive component (P2) was not as clearly detected as other components, however, it was still possible to analyze its grand average value and it showed peaking at 308 $\pm$ 4 ms after perturbation onset with an average amplitude of –0.94 $\pm$ 0.6 µV for treadmill speed = 1.6km/h, and peaked on average 318 $\pm$ 17 ms after perturbation onset with an amplitude of –1.3 $\pm$ 1.39 µV for speed = 2.5km/h. At perturbation onset (0 ms) on average, no visible derivation from the baseline is found.

The graphics of grand mean PEP latencies and PEP magnitudes for a representative remaining electrodes (Fz, C3, C4 e Pz) are shown in Appendix A from Fig.32 to Fig. 39. One observation is Fz channel produces strong PEP responses as compared to parietal and central-left and central-right cortices. Moreover, it was analyzed during the development of this work that C3 and C4 responses are undetectable in subject 2 at both speeds.



(a) Grand Average PEP for Speed = 1.6km/h

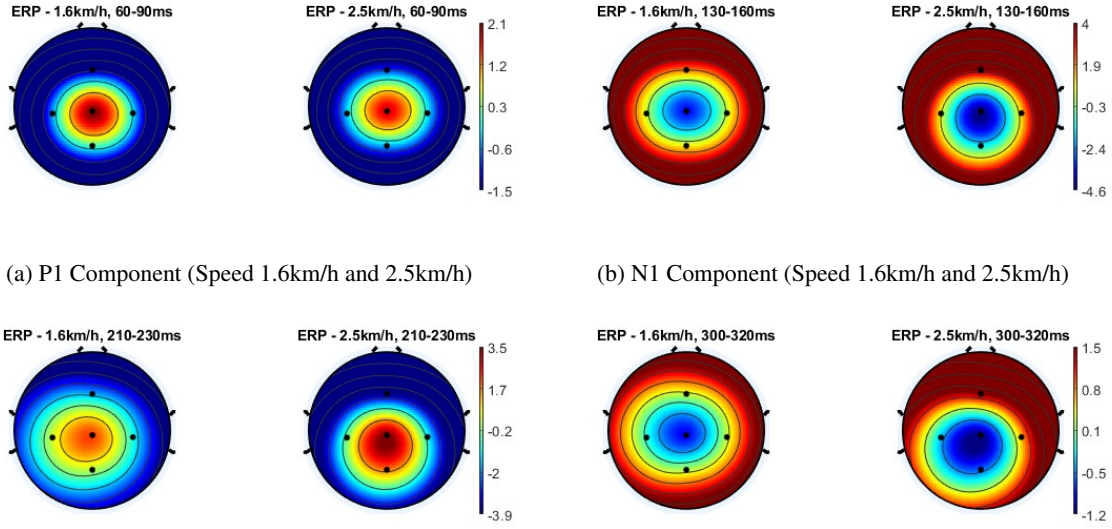


(b) Grand Average PEP for Speed = 2.5km/h

Figure 20: Grand average of perturbation trials at channel Cz. Black lines show the average for each speed. **(a):** Average of 70 trials for treadmill speed = 1.6km/h; **(b):** Average of 69 trials for treadmill speed = 2.5km/h. Time = 0 represents the perturbation onset

As depicted in Fig. 21, the grand average on a topographical level for time-points: t = 60-80 ms (P1 peak); t = 130-160 ms (N1 peak) and t = 300-320 ms (P2 peak) and t = 210-230 ms (N2 peak). The peak of the N1 component was centered around Cz and distributed for speeds 1.6km/h and 2.5km/h. Also, the peak of all PEP

components is centered around Cz, although, for P2 and N2 peak is highlighted the presence of Pz channel as well for speed = 2.5km/h. Otherwise, the peak magnitude in N1, P2, and N2 components over the C4 channel has opposite directions (signal) in comparison to the Cz channel.



(a) P1 Component (Speed 1.6km/h and 2.5km/h)

(b) N1 Component (Speed 1.6km/h and 2.5km/h)

(c) P2 Component (Speed 1.6km/h and 2.5km/h)

(d) N2 Component (Speed 1.6km/h and 2.5km/h)

Figure 21: Topographical plots show the scalp distribution of the grand average in PEP responses

5.4 DISCUSSION

We were able to elicit PEPs using the paradigm and experimental setup described in the method section of this work. Our findings were in agreement with [48, 35, 10, 6]. We observed an average amplitude of -3.77 µV (speed = 1.6km/h) and -5.16 µV (speed = 2.5km/h) for the N1 component of PEPs elicited during the two tested protocols. On average, the N1 component peaked with a latency of 138 $\pm$ 3 ms and 150 $\pm$ 5 ms relative to perturbation onset. The N1 event was spread around frontal, and central areas with a maximal peak at Cz. These findings related to latency are in accordance with results published in previous studies, however, related to amplitude they are in discordance. According to Ditz et al. [10] study, they found higher amplitudes (-28.3 ± 14.5 µV), and Ozdemir et al. [31] found amplitudes of -16 ± 3 µV (Young) and -11 ± 2 µV (Elderly).

The late components of PEPs (P2 and N2), can be mainly found from 200 ms and 300 ms after perturbation onset [48]. As shown in Fig. 20, we were able, through grand average, to identify those components, however, for subject 1 (green line) the component N2 cannot be distinguishable. On average, the P2 peaked at 226 $\pm$ 21 ms (speed = 1.6km/h) after perturbation onset with an amplitude of 2.25 $\pm$ 1.61 µV and peaked at 216 $\pm$ 5 ms (speed = 2.5km/h) after perturbation onset with an amplitude of 3.76 $\pm$ 2.72 µV This finding agrees with the result regarding waveform and latency of previously conducted studies involving postural seated perturbations [10] and balance platform which introduces perturbation [35, 6]. In contrast, Ditz et al. study found higher amplitudes in P2

(16.7 $\pm$ 5.3 µV). The observed differences between PEP amplitudes between our study and previous reports could be explained by the difference in task design between our study and [10].

A decreasing amplitude of signal obtained from the grand average could be justified due to the postural reaction to recover the posture after the evoked perturbation. In the previous studies, the P2 amplitude was higher in passive trials, i.e. trials that do not evoke a compensatory balance reaction. This disagreement between their and our results could occur due to different types of perturbation. In our experiment, participants were exposed to whole-body perturbations when they are walking on a treadmill, while the PEPs in previous studies' experiments were elicited using base translation or seated on the cockpit. According to Ozdemir et al. [31], lower N1 amplitude can be observed when attentional resources are limited by dual-tasking which led to greater postural instability.

The P1 component of PEPs was distinguishable from background EEG for all of the recorded participants. This finding is inconsistent with P1 responses observed in previously conducted studies' perturbation tasks. Due to the experimental setup that was run in this study which general results were lower than in comparison with the results in the literature, even the small amplitude in P1 could result in separating this early PEP component from background EEG. Since the perturbation was manually induced, there were small differences in the perturbation stimulus between the trials. These variations, together with the low amplitude, might have resulted in the absence of a P1 response in PEPs of single participants due to the averaging of trials.

In a recent review, Varghese et al. [48] suggested that P1 is the earliest non-specific cortical response to a perturbation. They contended that the P1 is unrelated to the context of the balancing perturbation task and lacks information on the predictability of the perturbation or whether the perturbations are internally or externally caused. Unlike P1, N1 potential has been shown to not just be influenced by afferent signals. Instead, it is also influenced by the predictability and difficulty of the balance task [31], as well as the presence of competing for cognitive tasks [48]. Typically, EEG data are trial-averaged to improve the signal-to-noise ratio from event-related potentials. After confirming that PEP components were preserved, it was determined that perturbations could also be observed during trials while wearing EEG. This is a crucially important step toward the decode brain signal in response to the detection of perturbation.

In summary, by analyzing the grand mean of all channels per speed, it is possible to distinguish the presence of N1 magnitude in comparison to other PEP components, mainly in comparison to the walking stage (non-perturbation). Moreover, the Cz results differ from the rest of the electrodes, being constant in all trials (1.6km/h and 2.5 km/h). These observations led to speculation that there might exist a balance control network that has a status for the walking stage (the baseline period) and activation during reactive balance control events (PEP N1). Underlying EEG correlates show differences between the perturbation condition and the rest condition which correspond to the time window used for feature extraction of the best-performing classifiers [10].

6

DECODING LOWER LIMB SLIP-LIKE PERTURBATION

This chapter aims to offer baseline specifications for the future development of a brain-computer interface to prevent slip-related falls, considering brain signals related to perturbation events. The target specifications defined in this chapter for the deep learning architectures were obtained from the slip-induced experimental data and the methods used to pre-process and classify the data from the EEGLAB addressed in Chapter 5. For this reason, the target specifications primarily intend to decode brain signals in response to slip-like perturbations.

Thus, the methods to segment and label data as well as the architecture and parameters regarding deep learning models will be addressed in Section 6.2. Furthermore, the main results obtained from the training of the different Artificial Neural Networks architectures will be analyzed in Section 6.3. Finally, Section 6.4 presents a discussion regarding the results obtained, highlighting the positives aspect as well as recognizing the limitations.

6.1 INTRODUCTORY INSIGHT

The current chapter shows the AI-based models used to detect the presence of perturbation in EEG trials which will be part of the development of fall prevention wearable robotic devices. This requires the decoding of brain signals in response of humans to slip events, in order to determine if there were a loss of balance during walking. Based on previously collected fall data and the specification for fall slip-like perturbations allowed feeding the AI-based model to classify the presence of the disturbance in the brain signal, consequently, the system can take control of the patient if it is needed to prevent falling.

Studies examining perturbations with an EEG paradigm, as well as the temporal relationship between signal modalities, are rare. More importantly, based on the literature review, no previous studies [45, 39, 23, 32, 47]have evaluated the influence of balance perturbations on EEG during walking, and mainly, few of them [35, 10, 6] developed AI-based model to identify when a balance loss occurs.

6.2 METHODS

As a result, and taking into consideration the drawbacks observed in the literature, the methods used to validate and recognize the LOB provoked by perturbation (study material of this work) presented in Chapter 3 will be discussed in this Section by the following methods: **i)** labeling the data in two classes (perturbation and non-

perturbation) considering the data segmentation mentioned in the literature review in Chapter 3, and convert the data into sub-datasets (Subsection 6.2.1); **ii)** application of Deep Learning-based models on the datasets and parameters used for training the models (Subsection 6.2.2); **iii)** strategies applied to split data into training, validation, and test dataset (Subsection 6.2.3); and **iv)** the metrics used to quantify the developed models quality (Subsection 6.2.4).

6.2.1 *Data Labeling*

The step that followed the data segmentation was the labeling, i.e., the process in which each acquired sample is assigned a "label" that identifies it as part of perturbation or non-perturbation events. This process was carried out manually, where, according to Ravindran et al. [35], the **Class 1** (non-perturbation) was composed of individual trial windows during the baseline period (1200 – 500 ms) prior to the onset of perturbation and **Class 2** consisted of EEG segments between 0 ms until +500 ms post perturbation onset.

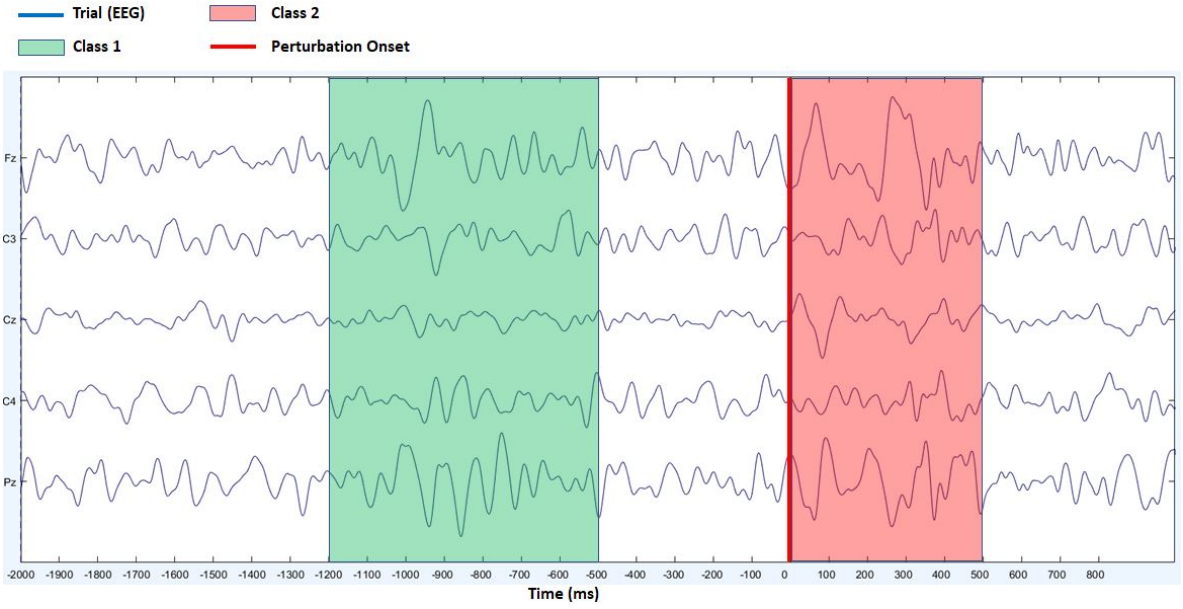


Figure 22: Sample of raw data from EEG trial demonstrating how the classes are defined.

Python code was written to label the signals present in each file, allowing easy clicks to label all samples. Labeling processes based on sliding time regarding class 1 and class 2 allowed to the extraction of data sets with relevant data which ones will be used to train the DL-based models. These extracted datasets were combined and filtered by subjects and speeds. Combining the data sets by the subjects and grouping them according to the speed and the filters used, allowed us to have a total of 10 datasets that were later used as a data source in the classification step through deep learning models. The table below summarizes the available datasets:

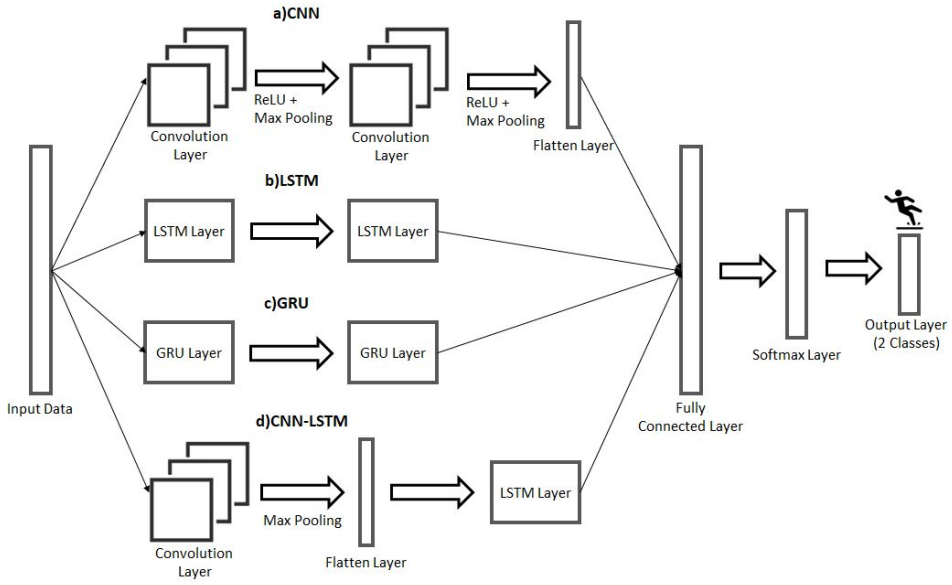
Table 4: New datasets extracted from data segmentation

Dataset	Subjects	Treadmill Speed	Channels/ICA	Nº Perturbation
DATASET_16_CAR_CH	s1 and s2	1.6km/h	Channels	70
DATASET_25_CAR_CH	s1 and s2	2.5km/h	Channels	69
DATASET_s1_CAR_CH	s1	1.6km/h and 2.5km/h	Channels	60
DATASET_s2_CAR_CH	s2	1.6km/h and 2.5km/h	Channels	79
DATASET_all_CAR_CH	s1 and s2	1.6km/h and 2.5km/h	Channels	139
DATASET_16_ICA_CH	s1 and s2	1.6km/h	ICA	70
DATASET_25_ICA_CH	s1 and s2	2.5km/h	ICA	69
DATASET_s1_ICA_CH	s1	1.6km/h and 2.5km/h	ICA	60
DATASET_s2_ICA_CH	s2	1.6km/h and 2.5km/h	ICA	79
DATASET_all_ICA_CH	s1 and s2	1.6km/h and 2.5km/h	ICA	139

6.2.2 *Classification Models and Architectures Parameters*

The fast development and advancement of deep learning techniques have made it possible to achieve promising results for the analysis of gait for the control or the potential control of active lower limb exoskeletons and orthoses [34, 43]. Using the literature analysis carried out in Chapter 3, three distinct neural network architectures were chosen and adapted to the problem of this dissertation. Thus, architectures of Convolutional Neural Networks (CNNs) [35], Gated Recurrent Unit (GRU) [43] and Long Short-Term Memory Neural Networks (LSTMs) [39, 14, 15] were built. In addition, even not mentioned at the state-of-the-art, a CNN-LSTM architecture that involves layers of CNN for feature extraction on input data combined with LSTMs to support sequence prediction was employed and tested during the development of this thesis [5]. These architectures are described in the next paragraph and shown in Fig. 23.

Regarding the extraction of features from the data obtained from the previous Section. The extraction was accomplished using the sliding window approach, the most often used method for data segmentation and feature extraction, in which a signal is divided into numerous equal-sized windows (subgroups) from which distinct characteristics may be estimated. The size of the sliding window represents the resolution of activity recognition. By using a large window size, a stream of activity data may be processed as one to analyze long-term activity patterns, but if several activities are packed in one window, it fails to identify balance loss events. When dealing with perturbations that occur in a relatively short period of time, a smaller window size may be more appropriate.



The input shown represents a single feature window. a) The CNN detects correlations among the many features presented. b) The LSTM network recognizes important temporal aspects. c) The GRU is able to effectively retain long-term dependencies in sequential data d) The hybrid CNN-LSTM retrieves temporal patterns from the CNN convolutional layer utilizing convolutional features. Following the fully connected layer (the layer where all the inputs from one layer are connected to every activation unit of the next layer), the last layer employs the softmax activation function (rather than ReLU) to get probabilities of the input belonging to a certain class (classification).

Figure 23: Deep Learning architectures.

According to the literature reviewed in Chapter 3, the most commonly used sliding windows have a size equating to roughly 200 milliseconds, with a 50% overlap between subsequent windows. Overlap can be utilized to increase the relevance of processed findings when processing data *versus* time. Depending on how it is used, overlap may aid in two ways: it can provide extra detail on how a quantity is changing over time or it can provide a smoothing effect, making the data prediction clear. In addition, increasing overlap can provide a smoothing effect and help make trends in data easier to observe, however, it results in more calculations per unit of time. This effect can be used to gain additional granularity on how a quantity in the data is changing with respect to time by increasing the number of calculations in a given measurement length. Within the scope of this research, a 200ms window was chosen to execute the comparison analysis, equating to a window containing 100 samples, with a 50% and 80% overlap between two successive windows of 50 samples, corresponding to a window "advance" of 200 ms. Despite research demonstrating that overlap can have a large influence on classification performance and computational cost for real-time applications, the proportion of overlap between subsequent windows was preserved at 50% in this dissertation for all experiments undertaken.

As mentioned on Subsection 6.2.1 the labeling of dataset signals was done at the sample level, windows constructed by the sliding window approach required to be tagged as well. The window labeling was done automatically using the Mode Labeling Method (MLM), where the label of a particular window is the mode of all labels in that window.

During all of the operations, test specifications such as the loss function used, the number of Epochs, the optimizer employed, the number of hidden layers, and the Learning Rate were kept constant for all the architectures.

According to Ishaani and Chase [19] the hyperparameters are values that can control the process of learning and based on their studies the description on the main hyperparameters are listed below:

a The batch size indicates how many samples will be considered for tuning before the internal model parameters (weights and biases) are updated.

b The number of epochs indicates how many times the learning algorithm will run on the training dataset. The number of epochs dictates the amount of time it takes for the whole set of input training data to be transmitted through the neural network while training takes place. Multiple epochs during training will guarantee that the neural network weights change. When too few or too many epochs are employed, underfitting or overfitting can occur.

c For tuning, the optimization algorithm is considered. While the algorithm trains, the network's weights, and biases may vary, influencing the overall model performance. The output of the loss function is large if network predictions based on the model are bad. An optimizer is in charge of closing the gap between updating model parameters and calculating the loss function.

d During the training phase, the dropout strategy selects neurons to disregard. When these neurons' contributions are removed, their weights are no longer used. Dropout is based on the premise that as a neural network trains, the weights of the neurons are altered in relation to the neurons. This can also result in nearby neurons being specialized, which can lead to an overfitted model.

e The number of neurons in a hidden layer may be described as the number of units reflecting the number of neurons in a layer. A unit is in charge of receiving input from all of the nodes in the layer below. The number of neurons in a layer determines the total capacity of the network.

f Cross-entropy loss, also known as log loss, assesses the effectiveness of a classification model whose output is a probability value between 0 and 1. As the predicted probability diverges from the actual label, the cross-entropy loss increases.

Although ANN can work with raw data as input, some studies refer to the option of using previously extracted features as inputs [35, 10, 39]. Thus, in order to conduct a comparison research on how both neural networks react with the same sort of data, the four feature subsets were employed as inputs in the training and testing of the constructed architectures. During all of the operations, according to Ravindran et al. [35], test specifications such as the loss function used, the number of Epochs, the optimizer employed, the number of hidden layers, dropout, and the Learning Rate were kept constant for all the architectures, in contrast, different sizes of Batch was used to train the models[35, 39]. The Table below provides a summary of all these characteristics and respective values. The use of neural networks in this study was accomplished by implementing the underlying architecture without resorting to internal layer optimizations.

Table 5: Specifications for the use of the Deep Learning models

Hyperparameters	Value
Epoch Number	100
Batch Size	16, 32, 64 and 128
Hidden Layers	128
Dropout	0.5
Optimizer	Adam
Learning Rate	0.0005
Loss Function	Cross Entropy Loss

6.2.3 *Model Building and Evaluation*

The first step taken for the model's building and evaluation in the comparative analysis was to select the proper dataset and split it into a train set (training the model), a validation set (evaluate the model during the training process), and test set (evaluate the final model accuracy before deployment). For this division, due to the amount of data and samples obtained during the development of this project, each dataset used in this study has only two distinct subjects, subject 1 with 30 disturbances recorded at a speed of 1.6km/h and 2.5 km/h, and subject 2 with 40 disturbances recorded at a speed of 1.6km/h and 39 disturbances at a speed of 2.5km/h. Furthermore, as no cross-validation techniques were applied, the created feature windows were divided by the following approaches:

- Approach 1: Speed of 1.6km/h (Total samples = 70):
  - 40 samples of subject 2 and 2 samples of subject 1 were used as training data (total of 60%)
  - 14 samples of subject 1 were used as validation data (total of 20%)
  - 14 samples of subject 1 were used as test data (total of 20%)
- Approach 2: Speed of 2.5m/h (Total samples = 69):
  - 40 samples of subject 2 and 1 sample of subject 1 were used as training data (total of 60%)
  - 14 samples of subject 1 were used as validation data (total of 20%)
  - 14 samples of subject 1 were used as test data (total of 20%)
- Approach 3: Subject 1 (Total samples = 60):
  - 42 samples were randomly selected as training data (total of 70%)
  - 9 samples were randomly selected from remaining data as validation data (total of 15%)
  - The last 9 samples were used as test data (total of 15%)
- Approach 4: Subject 4 (Total samples = 79):
  - 55 samples were randomly selected as training data (total of 70%)
  - 12 samples were randomly selected from the remaining data as validation data (total of 15%)

  - The last 12 samples were used as test data (total of 15%)

- Approach 5: Subject 4 (Total samples = 139):
  - 97 samples were randomly selected as training data (total of 70%)
  - 21 samples were randomly selected from remaining data as validation data (total of 15%)
  - The last 21 samples were used as test data (total of 15%)

These splitting methodologies are not ideal, however, it is expected that, in approaches 1 and 2, all windows of the certain subjects are placed entirely either in the training data group or in the test data group, causing no activity to be only trained or only tested. In addition, with the separation of data by subject, although there is a probability of overfitting for the training data, we can still independently test the generalization of the model (or increase confidence in the obtained generalization value). In approaches 3 and 4, as the number of samples for speeds 1.6km/h and 2.5km/h are practically the same, and in the combined dataset all subsets of data are aggregated, the dataset was divided randomly. These same methodologies were also applied to the dataset that has independent components.

In order to understand deep-learning models the classification of slip-like perturbations was carried out using (4) four Neural Networks architectures mentioned in Section 6.2.2 and presented in Fig. 23. Furthermore, to analyze which would be the most suitable models for the classification of slip-like perturbations, we performed several training and validation using different combinations between 5 datasets (Table 4) and DL-based models. In addition, for each dataset was used 4 subsets of features as input to train and evaluate these deep learning models (a subset of features is summarized in the Table below). The performance of neural networks with the best set of features was analyzed and from the performance, results obtained an initial analysis was made for the best classification model for balance loss recognition.

Table 6: Subset of Features per Dataset and Deep Learning-Based Model

Channels Subsets			
Participants	**Treadmill Speed**	**Channels**	**DL Model**
S1	1.6km/h 2.5km/h	All Fz, C3, Cz, C4, Pz Fz, Cz Cz	GRU CNN-LSTM CNN LSTM
S2	1.6km/h 2.5km/h	All Fz, C3, Cz, C4, Pz Fz, Cz Cz	GRU CNN-LSTM CNN LSTM
S1 S2	1.6km/h	All Fz, C3, Cz, C4, Pz Fz, Cz Cz	GRU CNN-LSTM CNN LSTM
S1	2.5km/h	All Fz, C3, Cz, C4, Pz	GRU CNN-LSTM

S2		Fz, Cz Cz	CNN LSTM
All	All	All Fz, C3, Cz, C4, Pz Fz, Cz Cz	GRU CNN-LSTM CNN LSTM
Independent Components Subsets			
Participants	**Treadmill Speed**	**ICs**	**DL Model**
S1	1.6km/h 2.5km/h	All	GRU CNN-LSTM CNN LSTM
S2	1.6km/h 2.5km/h	All	GRU CNN-LSTM CNN LSTM
S1 S2	1.6km/h	All	GRU CNN-LSTM CNN LSTM
S1 S2	2.5km/h	All	GRU CNN-LSTM CNN LSTM
All	All	All	GRU CNN-LSTM CNN LSTM

6.2.4 *Model Evaluation Metrics*

Evaluating a model is the most important task in building an effective AI-based model. Its metrics are used to measure the quality of the model. As a binary classification task, the models used during the development of this work can only produce two results (perturbation or non-perturbation). Because supervised deep learning approach was applied, the models follow the following steps: **i)** fit a model on training data; **ii)** test the model on testing data. Once the model's predictions are obtained from the test data, a comparison between the test data and the true values (the correct labels) is performed. To measure the effectiveness of the classification model, the generated classification outcomes may be compared with the actual values of the provided observation. To extract the metrics a Confusion Matrix is utilized to describe these results.

The confusion matrix is a table that summarizes the classification model's performance in predicting instances from different classes i.e., it is an organized way of mapping the predictions to the original classes to which the data belong. The confusion matrix has two axes: one for the expected label and one for the actual label.

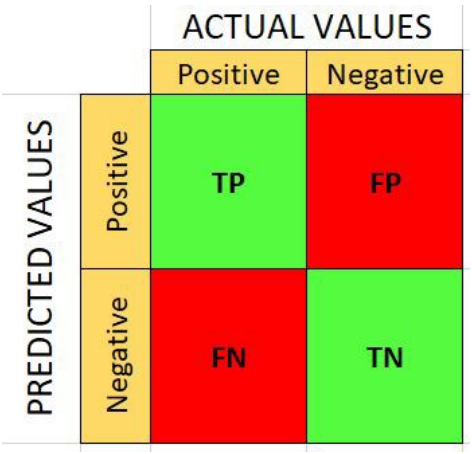


Figure 24: Confusion Matrix Example for Binary Classes. Taken from [44]

- **True Positives (TP):** the actual value is Positive (P) and predicted is also Positive i.e., the model correctly predicts the positive class.
- **True negatives (TN):** when the actual value is Negative (N) and prediction is also Negative i.e., the model correctly predicts the negative class.
- **False positives (FP):** When the actual is negative but prediction is Positive. Also known as the Type 1 error i.e., the model gives the wrong prediction of the negative class (predicted-positive, actual-negative).
- **False negatives (FN):** When the actual is Positive but the prediction is Negative. Also known as the Type 2 error i.e., the model wrongly predicts the positive class (predicted-negative, actual-positive).

The confusion matrix delivers insight not only into the errors being made by the classifier but more importantly the types of errors that are being made. In addition, the confusion matrix not only allows the calculation of the accuracy of a classifier, be it the global or the class-wise accuracy, but also helps compute other important metrics that evaluate ML and DL models. In general, we can get the following quantitative evaluation metrics from this binary class confusion matrix:

6.2.4.1 *Accuracy*

Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0. Based on equation 1 accuracy is calculated as the total number of two correct predictions (TP + TN) divided by the total number of a dataset (P + N).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Accuracy is widely used to evaluate model performance; nevertheless, there are a few drawbacks considered before relying on accuracy excessively. One of these disadvantages is that the model must categorize observations based on an imbalanced dataset in which one class, either true or false, is more prevalent than the other.

Error costs of positives and negatives are usually different. For instance, one wants to avoid false negatives more than false positives or vice versa. Other basic measures, such as sensitivity and specificity, are more informative than accuracy and error rate in such cases.

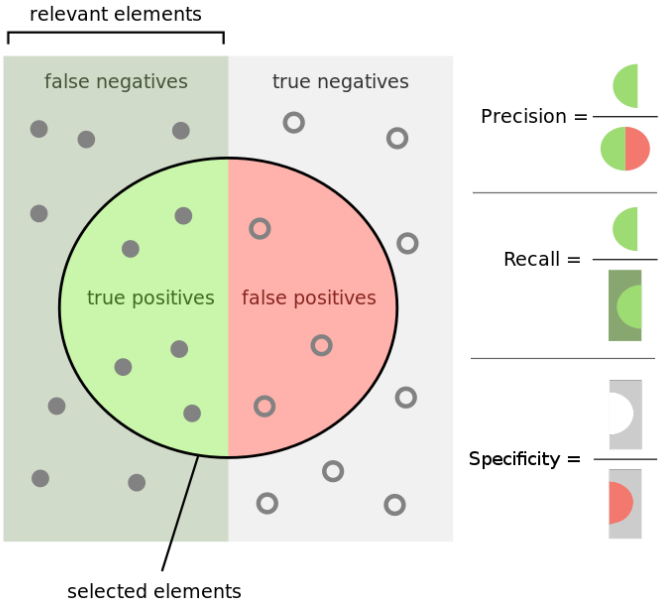


Figure 25: Illustration of Metrics extracted from Confusion Matrix. Adapted from [36]

6.2.4.2 *Recall (Sensitivity)*

The recall or Sensitivity metric calculates how many of the actual positive instances are able to correctly predict. It is the ratio of true positives to the total number of actual positive values in the data.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

6.2.4.3 *Specificity*

Specificity is defined as the algorithm's or model's ability to predict a genuine negative of each accessible category. It is also known simply as the true negative rate in the literature. Formally, the equation below may be used to compute it.

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

6.2.4.4 *Precision*

Precision metrics inform how correct, or precise, the model's positive predictions are. In definition, precision is defined as the actual correct prediction divided by the total number of positive predictions as presented in equation 4.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Precision is an important metric when False Positives are more important than False Negatives

6.2.4.5 *F1 Score*

The F1 score is calculated as a weighted average of precision and recall. As presented in equation 4 and 2, there is a false positive and false negative in accuracy and recall, therefore it takes both into account. F1 score is frequently more helpful than accuracy, especially if the class distribution is unequal. Accuracy works best when the cost of false positives and false negatives is comparable. If the cost of false positives and false negatives varies significantly, it is preferable to include both Precision and Recall.

$$F1Score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \tag{5}$$

6.2.4.6 *Matthews Correlation Coefficient*

Scientific researchers can use a variety of statistical tools to analyze binary classifications and their confusion matrices, depending on the purpose of the investigation. Accuracy and F1 score based using confusion matrices were (and continue to be) among the most often used metrics in binary classification problems. However, when applied to unbalanced datasets, these statistical techniques might possibly provide overoptimistic inflated outcomes [8].

Instead, the Matthews correlation coefficient (MCC) is a more reliable statistical rate that produces a high score only if the prediction performed well in all four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally to the size of positive and negative elements in the dataset. MCC, in fact, calculates the correlation between the real classes and the predicted labels. (worst value = -1; best value = +1) [9].

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (FP + FN) \times (TN + FP) \times (TN + FN)}} \tag{6}$$

6.3 RESULTS

In the following subsections, the main results obtained in the evaluation process of the different Deep Learning architectures explained in the previous Section will be exposed in Section 6.3.1. Furthermore, the choice of

the best subset of parameters, considering the recognition of slip-like perturbations based on the evaluation metrics, as well as the analysis regarding runtime to train and validate the model, will be depicted in Section 6.3.2. Furthermore, the comparison of the best model results between the channel subset and ICs subset will be presented in Section 6.3.2.

6.3.1 *Deep Learning Architecture*

As mentioned in Sections 6.2.2 and 6.2.3, four architectures of Deep Neural Networks were applied to decode and classify the slip perturbation events. In order to make a comparative analysis of the performance of the Deep Learning models, the inputs for the training of these architectures are the combination of 3 attributes from datasets presented in Table 6. These combinations resulted in 128 test outcomes for channel subsets and 34 test outcomes for the ICs subset to be analyzed.

Table 7: Classification Metrics Results for Channels' Dataset

Speed at 1.6 km/h						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	88.10%	86.91%	74.08%	87.58%	87.12%	88.10%
CNN-LSTM	86.61%	84.99%	71.40%	90.33%	84.21%	86.61%
GRU	84.82%	81.96%	67.38%	86.95%	82.05%	84.82%
LSTM	84.82%	83.08%	66.97%	87.66%	82.19%	84.82%
Speed at 2.5 km/h						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	92.86%	92.26%	85.48%	94.19%	91.37%	92.86%
CNN-LSTM	90.48%	88.06%	78.65%	92.43%	87.36%	90.48%
GRU	92.14%	91.42%	84.05%	93.74%	90.41%	92.14%
LSTM	91.07%	90.16%	80.83%	91.78%	89.39%	91.07%
Subject 1						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	95.31%	94.39%	89.68%	96.44%	95.65%	95.31%
CNN-LSTM	96.35%	96.03%	92.18%	96.54%	95.66%	96.35%
GRU	90.63%	90.13%	81.05%	93.48%	89.67%	90.63%
LSTM	88.54%	87.49%	75.20%	59.43%	86.94%	88.54%
Subject 2						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	93.18%	91.88%	80.74%	91.49%	92.78%	93.18%
CNN-LSTM	92.80%	91.44%	83.46%	91.62%	92.47%	92.80%
GRU	95.08%	92.82%	87.08%	93.19%	94.05%	95.08%
LSTM	93.94%	91.89%	83.28%	91.85%	92.19%	93.94%
Combined Dataset (Subjects and Speeds)						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	88.10%	86.48%	73.87%	91.03%	86.86%	88.10%
CNN-LSTM	87.14%	85.29%	73.84%	90.08%	85.75%	87.14%
GRU	86.19%	84.24%	72.80%	90.33%	83.43%	86.19%

LSTM	85.52%	83.45%	69.58%	88.30%	81.84%	85.52%

The final results for channel subsets, obtained for each of the tests performed as well as the graphics of the remaining results regarding the evaluation metrics are shown in Appendix B, from Fig.40 to Fig.89. Through direct observation of metrics results in Table 7, the best performance results in the subsets of features sort of by **speed (1.6km/h and 2.5km/h)** and CAR filter were obtained for the CNN model in both subsets when used 5 channels (Fz, C3, Cz, C4, Pz) as the input (Speed: 2.6km/h, Accuracy: 92.86%, Recall: 91.37%, Specificity: 92.86%, Precision: 94,19%, F1-Score: 92.26%, MCC: 85.48%). In addition, other tests were performed to evaluate the model performance sort of by subjects 1 and 2 as well, and the best performance results regarding DL-Model were different between the subsets. For **Subject 1** the CNN-LSTM model was the best and it used all 16 channels as the input (Accuracy: 96.35%, Recall: 95.65%, Specificity: 96.35%, Precision: 96.54%, F1-Score: 96.03%, MCC: 92.18%). Otherwise, for **Subject 2** the GRU model was the best and it also used all 16 channels as the input (Accuracy: 95.08%, Recall: 91.93%, Specificity: 83.22%, Precision: 87.80%, F1-Score: 88.62%, MCC: NaN). Complementary to the tests mentioned previously, an evaluation was performed with **combined dataset** which includes all speeds and subjects to be used as training input and the best performance result was obtained for CNN model when it used all channels as input (Accuracy: 88.10%, Recall: 86.86%, Specificity: 88.0%, Precision: 91.03%, F1-Score: 86.48%, MCC: NaN).

From the tests carried out with the artificial neural networks, besides training the model and running the tests with a different number of channels, as shown in Appendix B in Figs. 44, 54, 64, 74 and 84, it can be seen that the best results regarding performance evaluation metrics are obtained using the following hyperparameters: **a) Batch Size:** 16 (Subject 1 and combined data), 32 (2.5km/h), 64 (Subject 1), 128 (1.6km/h); **b) Overlap:** 50 (2.5km/h and combined data), 80 (1.6km/h, Subject 1 and Subject 2).

Table 8: Classification Metrics Results for ICs' Dataset

Speed at 1.6 km/h						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	76.79%	75.49%	51.97%	76.45%	75.97%	76.79%
CNN-LSTM	79.29%	78.21%	57.37%	79.07%	78.35%	79.29%
GRU	80.00%	78.36%	57.80%	80.46%	77.75%	80.00%
LSTM	79.46%	76.15%	54.59%	78.38%	76.39%	79.46%
Speed at 2.5 km/h						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	85.71%	84.38%	69.77%	85.57%	84.86%	85.71%
CNN-LSTM	83.63%	82.52%	66.95%	86.14%	83.29%	83.63%
GRU	78.27%	75.86%	53.72%	82.81%	75.15%	78.27%
LSTM	79.46%	76.15%	55.37%	80.70%	75.20%	79.46%
Subject 1						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	88.54%	86.87%	75.28%	90.11%	85.34%	88.54%
CNN-LSTM	90.10%	87.60%	75.95%	89.88%	86.69%	90.10%
GRU	89.06%	85.01%	71.56%	88.76%	83.85%	89.06%

LSTM	88.54%	87.18%	74.94%	88.75%	86.23%	88.54%
Subject 2						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	91.25%	90.40%	83.49%	93.15%	91.67%	91.25%
CNN-LSTM	94.79%	94.46%	89.91%	94.70%	95.27%	94.79%
GRU	92.71%	91.99%	84.21%	92.86%	91.36%	92.71%
LSTM	92.71%	92.05%	84.27%	92.58%	91.70%	92.71%
Combined Dataset (Subjects and Speeds)						
Model	ACC	F1 Score	MCC	Precision	Recall	Specification
CNN	88.10%	84.33%	70.01%	84.97%	85.07%	88.10%
CNN-LSTM	89.09%	88.56%	77.78%	88.22%	89.57%	89.09%
GRU	87.10%	84.93%	72.64%	8.90%	84.10%	87.10%
LSTM	88.49%	86.59%	74.84%	89.83%	85.93%	88.49%

Regarding ICs subsets, the final results obtained for each of the tests performed as well as the graphics of the remaining results regarding the evaluation metrics are shown in Appendix C, from Fig.90 to Fig.119. Opposite to channel subsets, which have 4 combinations for channels (16 channels, 5 channels, 2 channels, and 1 channel), the ICs subsets were training with 16 independent components only. Through direct observation of the results in Table 8, the best performance results in the subsets of features sort of by **speed (1.6km/h and 2.5km/h)** and ICA filter were obtained for the **GRU model at speed of 1.6km/h** as the input (Accuracy: 80.00%, Recall: 77.75%, Specificity: 80.00%, Precision: 80.10%, F1-Score: 78.36%, MCC: 57.80%) and the **CNN model at speed of 2.5km/h** as the input (Accuracy: 85.71%, Recall: 84.86%, Specificity: 85.71%, Precision: 84.97%, F1-Score: 84.38%, MCC: 69.77%). In addition, other tests were performed to evaluate the model performance sort of by subjects 1 and 2 as well, and the best performance results regarding DL-Model were different between the subsets. For **Subject 1** the **CNN-LSTM model** was the best (Accuracy: 90.10%, Recall: 81.09%, Specificity: 90.10%, Precision: 80.40%, F1-Score: 79.87%, MCC: NaN). For **Subject 2** the **CNN-LSTM model** was the best (Accuracy: 94.79%, Recall: 95.27%, Specificity: 94.79%, Precision: 94.70%, F1-Score: 94.46%, MCC: 89.91%). Complementary to the tests mentioned previously, an evaluation was performed with **combined dataset** which includes all speeds and subjects to be used as training input and the best performance result was obtained for **CNN-LSTM model** was the best (Accuracy: 89.09%, Recall: 89.57%, Specificity: 89.09%, Precision: 88.22%, F1-Score: 88.56%, MCC: 77.78%).

From the tests carried out with the artificial neural networks, as shown in Appendix C in Figs. 91, 97, 103, 109 and 115, it can be seen that the best results regarding performance evaluation metrics are obtained using the following hyperparameters: **a) Batch Size:** 16 (Subject 1 and 2.5km/h), 32 (Subject 2), 128 (Subject 2 and combined data); **b) Overlap:** 50 (1.6km/h and 2.5km/h), 80 (Subject 1, Subject 2, and combined data).

6.3.2 *Models and Parameters Evaluation*

Furthermore, to the value of metrics presented above, a comparative analysis of Deep Learning models related to the frequency (number of times) of the model performed above accuracy of 85%, as well as the inputs used and hyperparameters tested, it was possible to obtain the values to evaluate their relevance to decode brain signal to

predict loss of balance provoked by slip-like perturbations. The final results can be observed in AppendixB through the graphics Figs. 45, 55, 65, 75 and 85 and AppendixC through the graphics Figs. 92, 98, 104, 110 and 116

Table 9: Number of results from Channels data (Accuracy ⩾ Target)

Dataset	DL Models		EEG Channels		Hyperparameters			
	Model	ACC⩾85%	Channels	ACC⩾85%	Batch	ACC⩾85%	Overlap	ACC⩾85%
1.6 km/h	CNN	1	16	2	16	1	50	1
	CNN-LSTM	3	5	1	32	-		
	GRU	-	2	-	64	-	80	3
	LSTM	-	1	1	128	3		
Dataset	**DL Models**		**EEG Channels**		**Hyperparameters**			
	Model	**ACC⩾85%**	**Channels**	**ACC⩾85%**	**Batch**	**ACC⩾85%**	**Overlap**	**ACC⩾85%**
2.5 km/h	CNN	21	16	14	16	18	50	36
	CNN-LSTM	18	5	32	32	19		
	GRU	17	2	3	64	18	80	40
	LSTM	20	1	27	128	21		
Dataset	**DL Models**		**EEG Channels**		**Hyperparameters**			
	Model	**ACC⩾90%**	**Channels**	**ACC⩾90%**	**Batch**	**ACC⩾90%**	**Overlap**	**ACC⩾90%**
S1	CNN	10	16	13	16	4	50	6
	CNN-LSTM	8	5	8	32	7		
	GRU	3	2	-	64	6	80	15
	LSTM	-	1	-	128	4		
Dataset	**DL Models**		**EEG Channels**		**Hyperparameters**			
	Model	**ACC⩾90%**	**Channels**	**ACC⩾90%**	**Batch**	**ACC⩾90%**	**Overlap**	**ACC⩾90%**
S2	CNN	4	16	13	16	9	50	8
	CNN-LSTM	5	5	8	32	6		
	GRU	12	2	2	64	6	80	17
	LSTM	4	1	-	128	4		
Dataset	**DL Models**		**EEG Channels**		**Hyperparameters**			
	Model	**ACC⩾85%**	**Channels**	**ACC⩾85%**	**Batch**	**ACC⩾85%**	**Overlap**	**ACC⩾85%**
All	CNN	7	16	9	16	6	50	12
	CNN-LSTM	7	5	10	32	5		
	GRU	3	2	-	64	3	80	7
	LSTM	2	1	-	128	5		

The results attained in this study for channel subsets, for frequency of accuracy above 85%, obtained for each test performed, are depicted in Table 9. It should be noted that the **combined subdataset** obtained approximately 15% of the test's accuracy above 85%, from this 15% as CNN model (36.84%) well as CNN-LSTM (36.84%) performed above 85%, and the best parameter used as input was 5 channels (52.63%). In addition, regarding hyperparameters, 16 of Batch Size (31.58%) and Overlap of 50 (63.16%) performed above 85%. The results obtained related to **speed of 1.6km/h**, only 3% of results had accuracy above 85%. The CNN-LSTM model represented 75% of these results, and 16 channels were the best parameter used as input (50%). Both hyperparameters, 128 of Batch Size and Overlap of 80 obtained 75% of the results with accuracy above 85%. Regarding **speed at 2.5km/h** greater than 59% or results performed above 85%. It can be noted that all models, batch sizes, and overlaps have results almost equally distributed. Otherwise, regarding channels, it can be noted that the best input was 5 channels (42.1%) of results with accuracy above 85%. In contrast to subsets results, it can be observed in Appendix B through the graphics Figs. 75 and 85 that subsets filtered by Subject 1 or Subject

2 obtained results above 90%. As they are shown in Table 9 the results obtained related to **subject 1** obtained approximately 16% of the test's accuracy above 90%, from this 15% the CNN model (47.62%) performed above 90%, and the best parameter used as input was 16 channels (61.9%). In addition, regarding hyperparameters, 32 of Batch Size (33.33%) and Overlap of 80 (71.43%) performed above 90%. For **subject 2** the result was 19.53% which test accuracy was above 90%. The GRU model represented 48% of these results, and 16 channels were the best parameter used as input (52%). In addition, regarding hyperparameters, 128 of Batch Size (36%) and Overlap of 80 (68%) performed above 85%.

A direct comparison through Table 11 between the best results obtained (regardless of the architecture and execution time) in the tests performed with the dataset with ERP of the EEG channels and the ERP of the ICs, where the approach with 16 channels was considered for analysis, since the training and testing the models with dataset ICs was done with 16 ICs only, showing that the subsets of the dataset of channels obtained the best performance with consistent results above 88% and accuracy superior to the results obtained by the dataset of ICs. Another comparative analysis was also carried out, directly, where the architecture of the models that obtained greater accuracy in a dataset was analyzed, with their respective architecture in another dataset and vice versa. Through direct observation of the Table 11, it was possible to verify that only for the approach with a subset of data related to Subject 2 and Combined, the same architectural topology in the IC dataset was superior.

Table 10: Number of results from ICs data (Accuracy $\geqslant$ Target)

Dataset	**DL Models**		**Hyperparameters**			
	Model	**ACC$\geqslant$85%**	**Batch**	**ACC$\geqslant$85%**	**Overlap**	**ACC$\geqslant$85%**
1.6 km/h	CNN	-	16	-	50	1
	CNN-LSTM	-	32	-		
	GRU	1	64	-	80	-
	LSTM	-	128	1		
Dataset	**DL Models**		**Hyperparameters**			
	Model	**ACC$\geqslant$80%**	**Batch**	**ACC$\geqslant$80%**	**Overlap**	**ACC$\geqslant$80%**
2.5 km/h	CNN	6	16	3	50	7
	CNN-LSTM	7	32	3		
	GRU	-	64	3	80	6
	LSTM	-	128	4		
Dataset	**DL Models**		**Hyperparameters**			
	Model	**ACC$\geqslant$90%**	**Batch**	**ACC$\geqslant$90%**	**Overlap**	**ACC$\geqslant$90%**
S1	CNN	-	16	1	50	-
	CNN-LSTM	1	32	-		
	GRU	-	64	-	80	1
	LSTM	-	128	-		
Dataset	**DL Models**		**Hyperparameters**			
	Model	**ACC$\geqslant$90%**	**Batch**	**ACC$\geqslant$90%**	**Overlap**	**ACC$\geqslant$90%**
S2	CNN	3	16	4	50	3
	CNN-LSTM	5	32	2		
	GRU	1	64	1	80	8
	LSTM	2	128	4		
Dataset	**DL Models**		**Hyperparameters**			

	Model	ACC⩾85%	Batch	ACC⩾85%	Overlap	ACC⩾85%
All	CNN	1	16	4	50	3
	CNN-LSTM	4	32	5		
	GRU	3	64	2	80	10
	LSTM	5	128	2		

In order to compare, the results for ICs subsets are depicted in Table 10. These datasets, as mentioned in the previous Section, were trained with 16 ICs. Through direct observation, it should be noted that the **combined subdataset** obtained approximately 40.63% of the test's accuracy above 85%. The best model was LSTM model (38.46%), and the best hyperparameters used to train the models 32 of Batch Size (38.46%) and Overlap of 80 (79.62%) performed above 85%. In contrast to channel subsets, the tests accuracy-related at **speed 1.6km/h** and **2.5km/h** were less than 85%. Regarding **speed at 1.6km/h**, only one model and hyperparameters obtained results above 80%, they were GRU model, and 128 Batch Size and Overlap of 50. Related to **speed at 2.5 km/h** greater than 40% or results performed above 80%. The CNN-LSTM model represented 53.85% of these results, and the following hyperparameters, 128 of Batch Size (30.77%) and Overlap of 50 (53.85%) performed above 80%. Similar to channel subsets, the test's accuracy related to Subject 1 and Subject 2 obtained results above 90%. Nevertheless, for **Subject 1**, only one model and hyperparameters obtained results above 90%, they were the CNN-LSTM model, and 16 Batch Size and Overlap of 50, while the **Subject 2** presented the results above 34% of the tests accuracy above 90%, which one obtained CNN-LSTM model (45.45%) as the greatest results. Moreover, regarding hyperparameters, both 16 and 128 of Batch Size presented the same results (36.36%) and Overlap of 80 (72.73%) performed above 90%.

A direct comparison through Table 11 between the best results obtained (regardless of the architecture and execution time) in the tests performed with the dataset with ERP of the EEG channels and the ERP of the ICs, where the approach with 16 channels was considered for analysis since the training and testing of the models with dataset ICs was done with 16 ICs only, showing that the subsets of the dataset of channels obtained the best performance with consistent results above 88% and accuracy superior to the results obtained by the dataset of ICs. Another comparative analysis was also carried out, directly, where the architecture of the models that obtained greater accuracy in a dataset was analyzed, with their respective architecture in another dataset and vice versa. Through direct observation of Table 11, it was possible to verify that only for the approach with a subset of data related to Subject 2 and Combined, the same architectural topology in the IC dataset was superior.

Table 11: Comparative table between the data set of Channels and ICs that obtained better results in their respective network architectures of the model

Channels Subsets					
Filtered By:	**Model**	**Batch Size**	**Overlap**	**Runtime (sec)**	**Accuracy**
1.6km/h	CNN	128	80	143	88.10%
	GRU	128	50	102	69.29%
2.5km/h	CNN	128	50	103	90.71%
	CNN	16	50	172	88.57%
S1	CNN-LSTM	64	80	194	96.35%
	CNN-LSTM	16	80	388	95.31%

S2	GRU	16	80	366	95.78%
	CNN-LSTM	32	80	352	92.80%
Combined	CNN	16	50	277	88.10%
	CNN-LSTM	128	80	273	85.71%
Independent Components Subsets					
Filtered By:	**Model**	**Batch Size**	**Overlap**	**Runtime (sec)**	**Accuracy**
1.6km/h	GRU	128	50	104	80.00%
	CNN	128	80	154	73.51%
2.5km/h	CNN	16	50	176	85.71%
	CNN	128	50	90	80.71%
S1	CNN-LSTM	16	80	390	90.10%
	CNN-LSTM	64	80	191	89.58%
S2	CNN-LSTM	32	80	280	94.74%
	GRU	16	80	295	89.58%
Combined	CNN-LSTM	128	80	280	89.09%
	CNN	16	50	289	80.00%

6.4 DISCUSSION

In this master's dissertation, python's framework to decode the brain signal and recognize the loss of balance provoked by slip-like perturbations from several AI-based classification models and feature selection models was built in order to find the combination that presents better classification results for this type of perturbation event. The performance of the different combinations was evaluated from the results presented in Section 6.3, using the following parameters: **i)** Performance evaluation metrics, **ii)**Subset of parameters used, and **iii)** Training and classification times of the models.

Deep Learning-based classification was established with a combination of different classifiers and parameter selection and proved to be an accurate strategy, as based on the results presented in the previous chapter (Table 9 and Table 10), at the level of accuracy, all models that were trained performed above of 75%, and some of these obtained results greater than 95%. This process allowed the most relevant parameters and hyperparameters to be found for the binary classification (perturbation event and non-perturbation), either by data extracted from which the ERP of the channels were processed or from the ERP of independent components (Appendix B and Appendix C respectively), which should aid in a time-effective and low computation strategy for real-time near fall recognition.

The success achieved regarding the performance of the classification models based on accuracy, through the approaches described in Section 6.2.3, shows the robustness of the applied process. An important point that must be observed is in relation to the other metrics, the F1-Score presented similar results, with its average value 2% lower than the accuracy. However, the MCC presented lower results compared to the other metrics. This discrepancy in the results can be justified by the size of the dataset and the absence of techniques such as cross-validation, the model cannot generalize the data and is adjusted very close to the training dataset, generating overfitting in the data. Another factor is in relation to the balance of the data, as the segmented data

were labeled as non-disturbance in a time window of 700 ms and disturbance in a time window of 500 ms, this time window difference can contribute to dataset imbalance, and consequently in the training of the model, as the model will be trained with a larger time window whose data are related to the non-perturbation period. Thus, it will better classify the data when there is no slip-like perturbation during the walk.

Regarding the analysis made of the artificial neural network architectures, it is noted, through observation of the graphs in Appendix B and C and in the tables of Section 6.3.2, that the maximum results of all models were similar to each other in all approaches in the data set grouped by EEG channels, except for the approach related to subset Subject 1, where it was observed that CNN-LSTM obtained a maximum accuracy of 96.35% while LSTM obtained the worst result (88.54%). Nonetheless, even so, it is a good result compared to the results presented in the literature. Similar to the dataset grouped by EEG channels, the dataset grouped by ICs also presented models whose results were close to each other, except for the approach of the data subset Speed 2.5km/h, where it was observed that CNN obtained a maximum accuracy of 85.71% while LSTM had the worst result (78.27%).

Even though the results presented are similar to each other, we can also verify that in all approaches (10) both in the set grouped by EGG Channels and by data set grouped by ICs, the CNN model was the most ranked in 1st place with maximum accuracy. It is generally the most used model to solve problems involving image inputs and in this process, the implemented CNN architecture received as input 1D matrices of features extracted from sequential data. This method obtained good results in terms of performance metrics, since the convolution processes applied to extract characteristics from the arrays used, apparently, were carried out in the most efficient way, as stated in the literature [35]. Despite the performance results in all architectures being satisfactory, due to the low volume of data used in this work, the architectures of artificial neural networks must continue to be tested with more samples of data, different parameters and inputs, such as EEG data and EMG together, because several studies claim to obtain positive results in the recognition of activities with the use of this type of neural networks [39, 12, 16].

Furthermore, to the maximum values of the model classification metrics that were presented in the previous paragraph, it is possible to obtain relevant insights by analyzing the frequency of the results of models whose accuracy was greater than 85%, or even approaches whose value was greater than 90%. Thus, it was possible to observe through the tables of Section 6.3.1 that the CNN-LSTM model obtained mostly superior results than the other model, in terms of times that the model reached accuracy above the mentioned values. Based on its architecture, described in Section 6.2.2, models in CNN-LSTM are expected to have such results, however, a large computational effort is required to achieve these results.

Through direct observation of the graphs in Appendix B in Figs. 46 56 66 76 86 for EEG Channels datasets and Appendix C in Figs. 93 99 105 111 117 for ICs datasets it is noticed in general that execution times are generally higher in the CNN-LSTM architecture, followed by LSTM, GRU and finally CNN in both datasets (Channels and ICs). These results are coherent because due to the architecture of the CNN-LSTM model, it can support very long input sequences that can be read as blocks or subsequences by the CNN model and later gathered by the LSTM model. Applying this hybrid model, the samples are divided into additional subsequences, where the CNN model will interpret each subsequence and the LSTM will gather their interpretations. Thus, this process should require a higher computational training effort than the other models, and this can be interpreted through

the aforementioned graphs. It is also observed that the other LSTM and GRU models, in most tests, have very close execution times, which is justified by the similarity in the respective architectures. And finally, from a general analysis, it is possible to analyze that the CNN model has a shorter execution time, with an average difference of the 30s, which can corroborate the choice of model.

The comparative analysis between the best architectures from channels subsets and ICs subsets, from results shown in Table 10 and described in Section 6.3, the differences among results may indicate that the data does not need to be preprocessed using ICA algorithms, or that applying ICA to the signal data, in addition to reducing artifacts, can reduce overfitting in the training process, however, to reach a conclusion about the results, more samples need to be collected and tests performed.

In summary, related to the topics discussed above and through the graphics in Appendix B C combine with the metrics results presented in Section 6.3, the deep neural network models could, in general, predict the presence of slip-like perturbation signals through 5 channels with 32 of batch size and overlap of 50. Furthermore, to support this decision, was also important to analyze the runtime for each model and its parameters, and the models using 5 channels as input spent a maximum of 180 seconds to train the models.

Thus, this initial study of the behavior of neural networks for decoding data from brain signals, showed that, despite the positive results, due to the low volume of data and tools that were not applied in the training and validation stage of the models, such as cross-validation in search of finding the best combination for dataset division and grid search to optimize the choice of hyperparameters, may have contributed to optimistic results, therefore, more parameters and approaches need to be tested, mainly referring to the loss of balance caused by slip-like perturbation or other types of near fall perturbations. This need is due to the fact that most literature has carried out studies of the brain response related to ADLs, while the literature [35] was the only one that addressed the issue of loss of balance, however, the study itself analyzed the perturbation applied through a platform and not walking on a treadmill.

This fact raises the need to carry out several studies and some changes in the future, in order to obtain satisfactory performance results. The architectures used must be tested and validated in other scenarios, such as walking on a treadmill at normalized speed in accordance with the principle of dynamic similarity (v = $\sqrt{Fr * g * L}$, where Fr the Froude number (0.15); g is the gravity acceleration ($9.81 m/s^2$); L is the leg length from the prominence of the greater trochanter external surface to the lateral malleolus), applying other types of perturbations and carry out tests for decoding daily activities in order to obtain results to be comparable with the studies currently carried out in the literature. Furthermore, studies should be carried out to better understand the influence that different inputs and parameters have on the performance of these artificial neural networks. Finally, and as mentioned in most of the works analyzed in the literature, new tests must be performed in conjunction with other sensors, for example, EMG, and simulate real-time decoding.

7

CONCLUSION

This dissertation presents the process of developing brain signal decoding in response to slip-like perturbation. The current state-of-art concern that falls among the elderly is a severe danger that can result in death or non-fatal effects, as well as a significant sociological and economic impact on families and society. Furthermore, even if a fall does not result in an injury, people can develop a fear of falling, which is a big issue that affects both their mental and physical health.

Slipping appears to be the most common cause of falls. Slip-related falls typically have even worse effects than being held responsible for a major share of hip fractures in the elderly, highlighting the need to address this issue as soon as possible. As a result, effective solutions to the aforementioned problem are vital, and any effort to reduce or avoid a fall can save many lives. As a result, the development of wearable robotic systems for preventing slip-like falls arises as a solution to this problem. The fundamental function of these devices is to identify a loss of balance scenario and generate an actuation capable of restoring a proper biomechanical posture to allow the patient to resume a steady walking stride after a slip-like disturbance.

To intervene in this problem, the study of brain signals in response to slip-like perturbations appears to be a common preliminary step of the previous approach. Thus, the first step of the work was developing theoretical knowledge in the fields of EEG signals, BCI applications, and ANN, including the research for the most current studies related to decoding brain signals to ADLs. Furthermore, the extent of the literature included guidance and manuals related to frameworks, which assisted the understanding of the concepts, and guidelines from these tools.

During the process of developing the requirements applied to the case study (slip-like perturbation), the absence of detailed and specific guidance for creating those requirements was confirmed as a considerable difficulty for the process, mainly during the data analysis phase, which led to another state-of-the-art related to the brain signal response to the loss of balance. A new study was conducted in order to perceive balance perturbations in brain signals which are always accompanied by a specific cortical activation (PEP). However, it is aggravated due to the reduced amount of papers related to ANN architecture to decode those signals.

Considering the previously referred limitations found in the scientific literature, the present dissertation seeks to, firstly, present a comprehensive analysis of the brain signal response to slip-like perturbations previously collected at BirdLab. Through the brain signal analysis performed in Chapter 3 it was possible to elicit PEPs, and the findings were in agreement with related papers. When the perturbation is delivered, it results in peaks in the N1 component. In addition, it was observed the maximum peak in central areas, mainly at Cz.

Regarding the deep learning results attained in this dissertation for slip-like perturbation recognition, a comparative analysis was performed by using different deep learning models and architecture parameters, which allows the most relevant combination to classify non-perturbation and perturbation events. The best classification algorithm for the deep learning model was the CNN classifier. The performance obtained for this algorithm was: Accuracy > 88.09%; F1-Score > 86.48%; MCC > 74.07%; Precision > 87.58%; Recall > 86.52%; and Specificity > 88.09%; Furthermore, the training runtime was same for all models, except to CNN-LSTM which runtime spent was higher than other models. In general, all deep learning outcomes for the same classification problem were good (accuracy > 84%), and their potential was demonstrated, indicating that they could be a good option, however, it is important to collect more data.

The results from this study will be useful to design future studies for testing on those populations. Overall, the findings suggest that the EEG signals contain short-latency neural information related to an incoming fall, which may be useful for developing brain-machine interface systems for fall prevention.

In short, the investigation work developed in the scope of this dissertation enabled for answering the RQs outlined in Chapter 1.

- **RQ 1**: What are the most relevant signals and features that can help recognize the loss of balance provoked by slip-like perturbations?

  This RQ is addressed in Chapter 5. This study identified the presence of PEPs components in brain signals in response to slip-like perturbation and how changes in PEPs are related to changes in the actual dynamic postural response. For example, whether there is any difference in the onset, latency, or amplitude of ERPs that are correlated to particular elements of a balanced response. Compared to ERPs in other modalities such as visual, auditory, and memory-related ERPs, PEPs are the least explored ERPs even though they possess many diagnostic and research applications to assist in understanding the cortical control of balance [48]. It was found that all the components of the PEP were preserved and that the latency of the P1 and N1 waves preceded. This suggests the P1 and N1 components are viable signals for fall prediction. Compared to P1, N1 is a significantly larger component distributed across the central, frontal, and parietal channels at a latency of 100–150 ms, thus it can consider relevant features that can help recognize the loss of balance. Furthermore, evidence of the potential involvement of the cortex during reactive balance control in humans has come from event-related potentials (ERPs) that are time-locked to periods of instability. An unexpected postural perturbation evokes a large negative potential (termed as the perturbation-evoked potential N1 (PEP N1) that peaks 100–200 ms after the perturbation onset and is widely distributed over frontal, central, and parietal

- **RQ 2**: What are the most relevant channels to decode the slip-like perturbations during walking?

  Since the component is distributed over frontal, central, and parietal areas with the peak amplitude located at Cz, the removal of channels that are far away from Cz does not change the classification accuracy and suggests that no additional discriminating information can be found in these channels [10]. The most relevant channels that can decode the slip-like perturbations during walking, based on analysis of brain signals in response to loss of balance events in Chapter 5 and the results of performance metrics related to

DL-based models in Chapter 6, were the Fz, C3, Cz, C4, and Pz (5 channels). This hypothesis is supported by the high accuracies of well above 85% (peak) and the minimum runtime spent to train the models as presented in Appendix B.

In addition, the minimal layout that uses only the Cz electrode obtained high accuracies of well above 80% (peak) as listed below, however, as per results regarding runtime in Appendix B this layout spent much time in comparison to 5 channels and 2 layouts is still a much-discussed topic and without a definitive answer

- **Speed at 1.6 km/h**: Model = CNN-LSTM, ACC = 85.00%, F1S = 82.10%, MCC = 71.40%, PREC = 90.03%, REC = 82.38% and SPEC = 85.00%;
- **Speed at 2.5 km/h**: Model = CNN, ACC = 91.07%, F1S = 90.40%, MCC = 81.06%, PREC = 90.34%, REC = 90.73% and SPEC = 91.07%;
- **Subject 1**: Model = CNN, ACC = 88.02%, F1S = 86.37%, MCC = 73.96%, PREC = 86.88%, REC = 87.13% and SPEC = 88.02%;
- **Subject 2**: Model = CNN-LSTM, ACC = 86.36%, F1S = 84.77%, MCC = 72.85%, PREC = 89.59%, REC = 83.55% and SPEC = 86.36%;
- **Combined**: Model = LSTM, ACC = 81.90%, F1S = 79.76%, MCC = 63.05%, PREC = 84.55%, REC = 78.94% and SPEC = 81.90%.

- **RQ 3**: What is the best Deep Learning model to implement for the detection of slip-like perturbation during walking?

  From the work results presented in Chapter 6, the best Deep Learning algorithm for slip-like perturbation events recognition was the CNN architecture described in Chapters 2 and 6. When testing the trained model with unseen data it was achieved the following results related to five (5) data subsets, which were study objects of this dissertation:

  - **Speed at 1.6 km/h**: ACC = 88.10%, F1S = 86.91%, MCC = 74.08%, PREC = 87.58%, REC = 87.12% and SPEC = 88.10%;
  - **Speed at 2.5 km/h**: ACC = 92.86%, F1S = 92.26%, MCC = 85.48%, PREC = 94.19%, REC = 91.37% and SPEC = 92.86%;
  - **Subject 1**: ACC = 95.31%, F1S = 94.39%, MCC = 89.68%, PREC = 96.44%, REC = 95.65% and SPEC = 95.31%;
  - **Subject 2**: ACC = 93.18%, F1S = 91.88%, MCC = 80.74%, PREC = 91.49%, REC = 92.78% and SPEC = 93.18%;
  - **Combined**: ACC = 88.10%, F1S = 86.48%, MCC = 73.87%, PREC = 91.03%, REC = 86.86% and SPEC = 88.10%.

  In addition, through direct observation of the graphs in Appendix B in Figures 46, 56, 66, 76 and 86, the mean, standard deviation and maximum of the runtime spent to train the ANN architectures, the CNN model also presented the best performance spending less time to be trained, as described below:

- **Speed at 1.6 km/h**: AVG = 163.75s, SD = 59.54s and MAX = 315s;
- **Speed at 2.5 km/h**: AVG = 162.56s, SD = 61.43s and MAX = 311s;
- **Subject 1**: AVG = 151.53s, SD = 58.96s and MAX = 292s;
- **Subject 2**: AVG = 176.88s, SD = 75.11s and MAX = 351s;
- **Combined**: AVG = 261.59s, SD = 134.40s and MAX = 578s.

7.1 FUTURE WORK

In summary, significant components in PEPs were recognized as early as 70-155 ms after the onset of a mechanical external disturbance, and the perturbations were decoded using a CNN model from a single channel. Furthermore, it has been proved that the model was driven mostly by important components in the PEP to infer predictions, rather than by artifacts. However, it should be noted that these results were obtained using data from able-bodied young adults, and only the data collected from two participants were used in the process of analyzing brain signals in response to slip-like perturbations, as well as training and validating the artificial intelligence model. Future research into the repeatability and reproducibility of the findings in more susceptible groups, such as those over 65 and people with other gait disorders, such as chronic stroke, is needed. In addition, due to the fact PEP frequency analysis has only been performed in healthy people, future studies should analysis at the frequency properties of PEPs in older individuals and patient groups to see if abnormal frequency modulations in a specific frequency band contribute to the abnormalities identified in the time domain [48].

Regarding the experimental protocol, as initially the recognition of ADLs was the object of study of this dissertation, the new experimental protocol needs to be developed to carry out data collection experiments for the different daily activities listed in the literature review. Complementarily, as a suggestion for future research, it recommends that the steps of the EEGLAB framework used in the pre-processing phase be applied and tested with the data acquired from this new experimental protocol, whose aim is to validate whether the methods used for the recognition of the PEP also apply to the recognition of ADLs.

As mentioned in this dissertation, non-invasive hBCI technologies have been studied and applied for attempting to improve classification accuracy. In addition, Electroencephalography (EEG) uses fast temporal resolution and is most widely utilized in combination with other brain/non-brain signal acquisition modalities. Thus, the suggestion for future work is to combine EEG and EMG signals in hBCI to detect muscular movement which acts as an artifact in EEG signals, resulting in the false detection of brain signals, and analyze EMG activation patterns during slip-like perturbations to verify the latency related to postural recovery after perturbation events.

Due to the fact cross-validation was not applied to the dataset and the size of it is smaller in comparison with the data related to the papers in Chapters 2 and 3. As a result, it is envisaged that the results produced would be suboptimal, thus, attempts to enhance the design should be incorporated in future works in order to discover how deep learning approaches operate best in the context of identifying slip-like perturbations. Furthermore, the use of cross-validated grid search to identify the optimal hyperparameters for ANN architecture, demands future studies to investigate this method in order to obtain the optimal model to decode the brain signal.

Although other online BCI systems do not suffer severely from the use of a window-based classification technique, a BCI that is designed to detect loss of balance control operates under strong time constraints [10]. Future studies should address the delay imposed by the window-based categorization approach and determine if the duration of that delay is problematic in real-world scenarios.

Overall, the findings suggest that the EEG signals contain short-latency neural information related to an incoming fall, which may be useful for developing brain-machine interface systems for fall prevention. Despite comparable results with the literature, continued development of the tested neural networks and the usage of raw inertial data as input to the networks should be processed to be implemented. Ablation studies should be conducted to determine the impact of the various stages of each architecture on fall event recognition.

BIBLIOGRAPHY

[1] Ludwig Kip A, Miriani Rachel M, Langhals Nicholas B, Joseph Michael D, Anderson David J, and Kipke Daryl R. Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *The Journal of Neurophysiology*, 101, 2009.

[2] N. Van Dijk T. Van der Hooft A. C. Scheffer, M. J. Schuurmans and S. E. de Rooij. Fear of falling: Measurement strategy, prevalence, risk factors and consequences among older persons. *Age and Ageing*, 37(1):19–24, 2008.

[3] S. Makeig A. Delorme. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of Neuroscience Methods*, 2003.

[4] L. W. Forrester A. Presacco and title = "Decoding Intra-Limb and Inter-Limb Kinematics During Treadmill Walking From Scalp Electroencephalographic (EEG) Signals J. L. Contreras-Vidal". *IEEE Xplore*, 20(2):1–3, 2012.

[5] Ali Agga, Ahmed Abbou, Moussa Labbadi, Yassine El Houm, and Imane Hammou Ou Ali. Cnn-lstm: An efficient hybrid deep learning architecture for predicting short-term photovoltaic power production. *Electric Power Systems Research*, 208:107908, 2022.

[6] Didier Allexandre, Armand Hoxha, H. Vikram Shenoy, Soha Saleh, S. Easter Selvan, and Guang H. Yue. Altered cortical and postural response to balance perturbation in traumatic brain injury – an eeg pilot study. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1543–1546, 2019.

[7] A. Singla E. Singla M. Isaksson B. S. Rupal, S. Rafique and G. S. Virk. Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications. *International Journal of Advanced Robotic Systems*, pages 1–2, 2017.

[8] Chicco Davide and Jurman Giuseppe. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 2020.

[9] Chicco Davide, Tötsch Niklas, and Jurman Giuseppe. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BMC Genomics*, 41, 2021.

[10] Jonas C Ditz, Andreas Schwarz, and Gernot R Müller-Putz. Perturbation-evoked potentials can be classified from single-trial EEG. *Journal of Neural Engineering*, 17(3):036008, may 2020.

[11] B. Nienhuis J. Duysens P. Desain N. Keijsers J. Farquhar E. García-Cossio1, M. Severens. Decoding sensorimotor rhythms during robotic-assisted treadmill walking for brain computer interface (bci) applications. *Plos One*, 2016.

[12] ZiYing Fang, Neo Darren, Jue Wang, YiCheng Zheng, and Lam Yvonne Y. H. An integrated fall prevention system with single-channel eeg and emg sensor. *2021 4th International Conference on Circuits, Systems and Simulation (ICCSS)*, pages 183–189, 2021.

[13] Sleep Foundation. Alpha waves and sleep. Accessed: 01.04.2023.

[14] Van Houdt Greg, Mosquera Carlos, and Nápoles Gonzalo. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020.

[15] K. Watanabe H. Yokoyama, N. Kaneko and K. Nakazawa. Neural decoding of gait phases during motor imagery and improvement of the decoding accuracy by concurrent action observation. *Journal of Neural Engineering*, 2021.

[16] S. M. Shafiul Hasan, Masudur R. Siddiquee, Roozbeh Atri, Rodrigo Ramon, J. Sebastian Marquez, and Ou Bai. Prediction of gait intention from pre-movement eeg signals: a feasibility study. *Journal of NeuroEngineering and Rehabilitation*, 17, 2020.

[17] Ismail Fawaz Hassan, Forestier Germain, Weber Jonathan, Idoumghar Lhassane, and Muller Pierre-Alain. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33, 07 2019.

[18] Yongtian He, David Eguren, José M Azorín, Robert G Grossman, Trieu Phat Luu, and Jose L Contreras-Vidal. Brain–machine interfaces for controlling lower-limb powered robotic systems. *Journal of Neural Engineering*, 15(2):021004, 2018.

[19] Priyadarshini Ishaani and Cotton Chase a. A novel lstm–cnn–grid search-based deep neural network for sentiment analysis. *The Journal of Supercomputing*, 17, 2021.

[20] Pang Ivan, Okubo Yoshiro, Sturnieks Daina, Lord Stephen R., and Brodie Matthew A. Detection of near falls using wearable devices: A systematic reviews. *Journal of Geriatric Physical Therapy*, 42:48–56, 2019.

[21] K. L. Snyder J. E. Kline, H. J. Huang and D. P. Ferris. Isolating gait-related movement artifacts in electroencephalography during human walking. *Journal of Neuro Engineering*, 2015.

[22] C. Guan J. H. Jeong, N. S. Kwak and S. W. Lee. Decoding movement-related cortical potentials based on subject-dependent and section-wise spectral filtering. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(3), 2020.

[23] M. Jochumsen and I. K. Niazi. Detection and classification of single-trial movement-related cortical potentials associated with functional lower limb movements. *Journal of Neural Engineering*, 2020.

[24] H. J. Huang K. L. Snyder, J. E. Kline and D. P. Ferris. Independent component analysis of gait-related movement artifact recorded using eeg electrodes during treadmill walking. *Frontiers in Neuroscience*, 2015.

[25] L. Perroud R. Chavarriaga Millán J. del R. K. Lee, D. Liu. A brain-controlled exoskeleton with cascaded event-related desynchronization classifiers. *Robotics and Autonomous Systems*, 90:15–23, 2017.

[26] A. P. P. A Majeed R. M. Musa A. F. Ab. Nasir B. S. Bari M. Rashid, N. Sulaiman and S. Khatun. Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review. *Frontiers in Neurorobotics*, 2020.

[27] Luis Mercado, Lucero Alvarado, Griselda Quiroz-Compean, Rebeca Romo-Vazquez, Hugo Vélez-Pérez, M.A. Platas-Garza, Andrés A. González-Garrido, J.E. Gómez-Correa, J. Alejandro Morales, Angel Rodriguez-Liñan, Luis Torres-Treviño, and José M. Azorín. Decoding the torque of lower limb joints from eeg recordings of pre-gait movements using a machine learning scheme. *Neurocomputing*, 446:118–129, 2021.

[28] José del R. Millán. The human-computer connection : an overview of brain-computer interfaces. *Metode Science Studies Journal-Annual Review*, 9:135, 2018.

[29] Teli Mohammad. Dimensionality reduction and classification of time embedded eeg signals. 01 2023.

[30] Belkacem Abdelkader Nasreddine, Jamil Nuraini, Palmer Jason A., Ouhbi Sofia, and Chen Chao. Brain computer interfaces for improving the quality of life of older adults and elderly patients. *Frontiers in Neuroscience*, 14, 2020.

[31] Recep A. Ozdemir, Jose L. Contreras-Vidal, and William H. Paloski. Cortical control of upright stance in elderly. *Mechanisms of Ageing and Development*, 169:19–31, 2018.

[32] N. Mingchinda P. Leelaarporn N. Kunaseth S. Tammajarung P. Manoonpong S. C. Mukhopadhyay R. Chaisaen, P. Autthasan and T. Wilaiprasitporn. Decoding eeg rhythms during action observation, motor imagery, and execution for standing and sitting. *IEEE Sensors Journal*, 20, 2020.

[33] O. Lennon E. Formaggio N. Sutaj G. Dazzi C. Venturin I. Bonini R. Ortner H. A. C. Bazo L. Tonin S. Tortora S. Masiero R. Di Marco, M.Rubega and A. Del Felice. Experimental protocol to assess neuromuscular plasticity induced by an exoskeleton training session. *MPDI*, 2021.

[34] M. K. Al-Hares R. Kolaghassi and K. Sirlantzis. Systematic review of intelligent algorithms in gait analysis and prediction for lower limb robotic systems. *IEEE Access*, 9, 2021.

[35] Akshay Sujatha Ravindran, Christopher A Malaya, Isaac John, Gerard E Francisco, Charles Layne, and Jose L Contreras-Vidal. Decoding neural activity preceding balance loss during standing with a lower-limb exoskeleton using an interpretable deep learning model. *Journal of Neural Engineering*, 19(3):036015, may 2022.

[36] Abhijeet Rojatkar. Precision, recall, sensitivity and specificity. Accessed: 24.10.2022.

[37] K. C. Tan Abdullah Al-Mamun N. Thakor A. Bezerianos S. K. Goh, H. A. Abbass and J. Li. Spatio–spectral representation learning for electroencephalographic gait-pattern classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29(9), 2018.

[38] and; H. Kim S. Park; F. C. Park; J. Cho. Eeg-based gait state and gait intention recognition using spatio-spectral convolutional neural network. *7th International Winter Conference on Brain-Computer Interface (BCI)*, 2019.

[39] C. Chisari S. Micera S. Tortora, S. Ghidoni and F. Artoni. Deep learning-based bci for gait decoding from eeg with lstm recurrent neural network. *Journal of Neural Engineering*, 2020.

[40] C. Chisari S. Micera E. Menegatti S. Tortora, L. Tonin and F. Artoni. Hybrid human-machine interface for gait decoding through bayesian fusion of eeg and emg classifiers. *Frontiers in Neurorobotics*, pages 1–3, 2020.

[41] Jalilpour Shayan and Müller-Putz Gernot. Toward passive bci: asynchronous decoding of neural responses to direction- and angle-specific perturbations during a simulated cockpit scenario. *Scientific Reports*, 12(6802), 2022.

[42] Sur Shravani and Sinha V. K. Event-related potential: An overview. *Industry Psychiatry Journal*, 18:70–73, 2009.

[43] Y. He A. S. Ravindran S.Nakagome, T.P. Luu and J. L. Contreras-Vidal. An empirical comparison of neural networks and machine learning algorithms for eeg gait decoding. *Nature*, 2020.

[44] Anuganti Suresh. What is a confusion matrix? Accessed: 24.10.2022.

[45] D. Eguren M. Cestari T. P. Luu, Member and J. L. Contreras-Vidal. Eeg-based neural decoding of gait in developing children. *International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[46] S. Nakagame J. Gorges K. Nathan T. P. Luu, Y. He and J. L. Contreras-Vidal. Unscented kalman filter for neural decoding of human treadmill walking from non-invasive electroencephalography. *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016.

[47] Y. He T. P. Luu, S. Nakagome and J. L. Contreras-Vidal. Real-time eeg-based brain-computer interface to a virtual avatar enhances cortical involvement in human treadmill walking. *Nature*, 2017.

[48] Jessy Parokaran Varghese, Robert E. McIlroy, and Michael Barnett-Cowan. Perturbation-evoked potentials: Significance and application in balance control research. *Neuroscience & Biobehavioral Reviews*, 83:267–280, 2017.

[49] Jessy Parokaran Varghese, William R. Staines, and William E. McIlroy. Activity in functional cortical networks temporally associated with postural instability. *Neuroscience*, 401:43–58, 2019.

[50] WHO. Who global report on falls prevention in older age. Accessed: 01.02.2022.

[51] S. Prasad A. Kilicarslan Y. Zhang, 2 and J. L. Contreras-Vidal. Multiple kernel based region importance learning for neural classification of gait states from eeg signals. *Frontiers in Neuroscience*, 2017.

Appendices

A

OUTCOMES FROM PRE-PROCESSING STEPS

A.1 COMPONENTS OF WIRELESS EEG SYSTEM

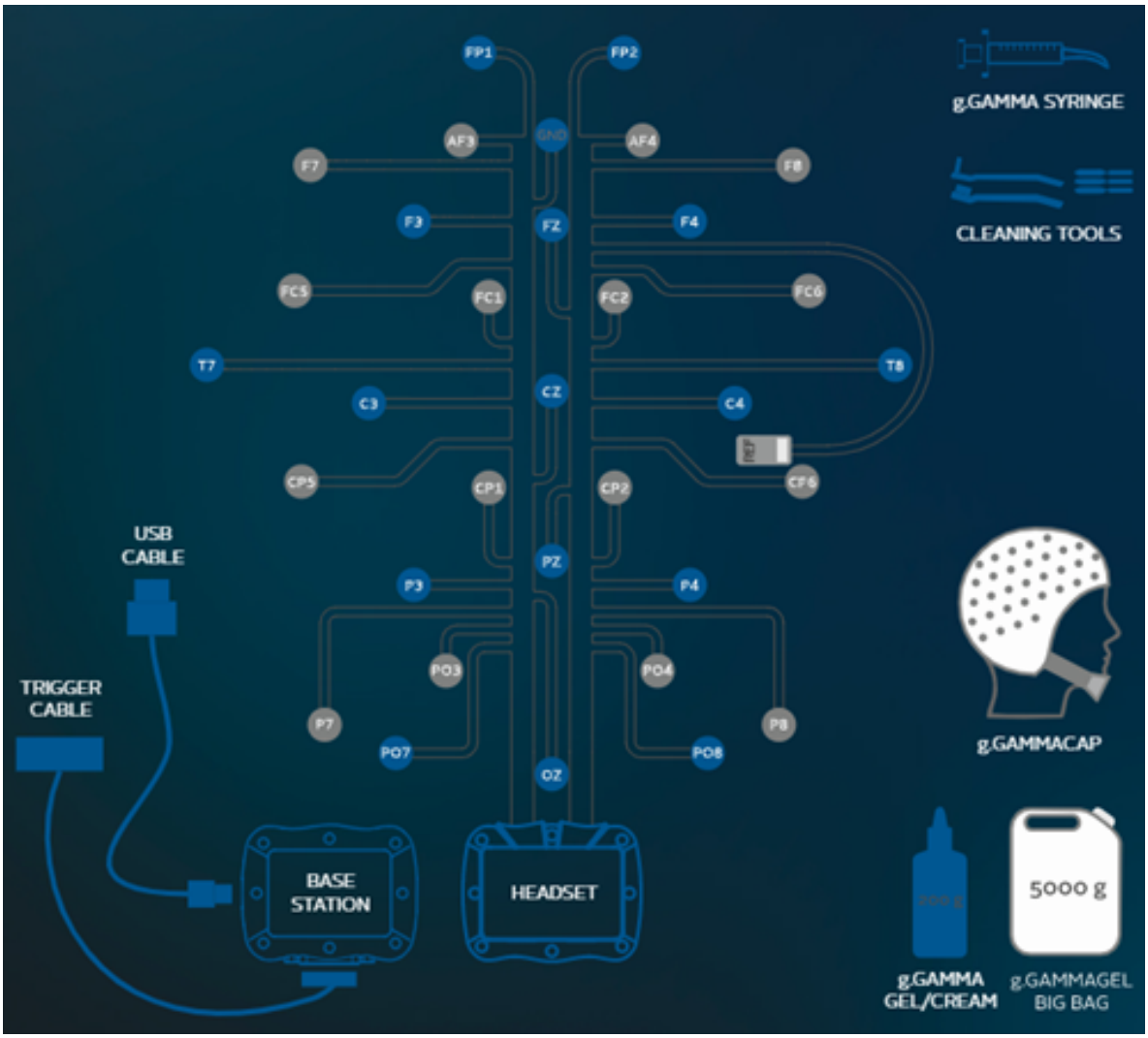


Figure 26: EEG g.NAUTILUS PRO - Components

A.2 PLOTS FROM DATA PRE-PROCESSING STEPS

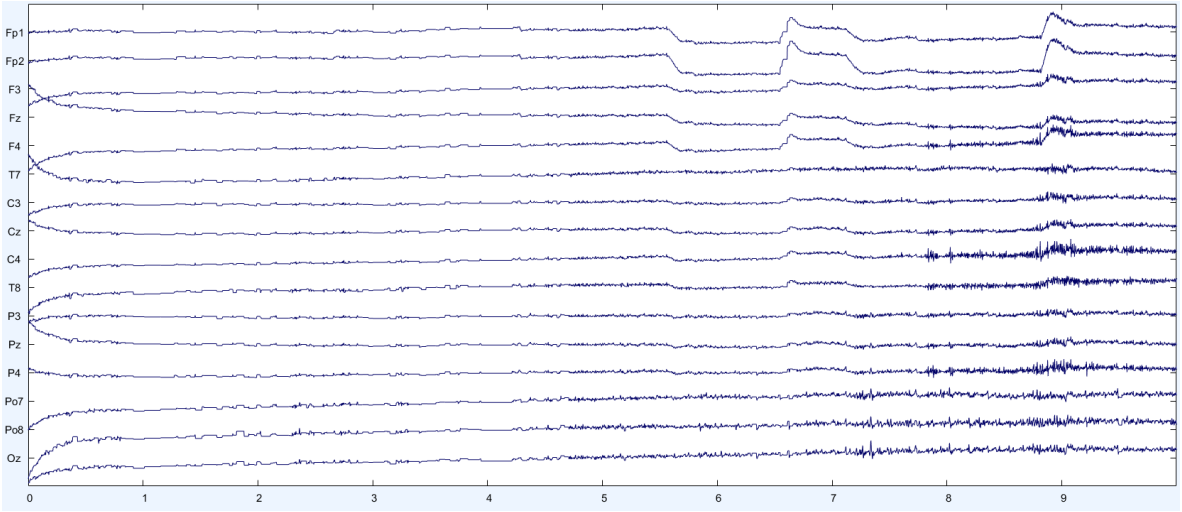


Figure 27: EEG Raw Data

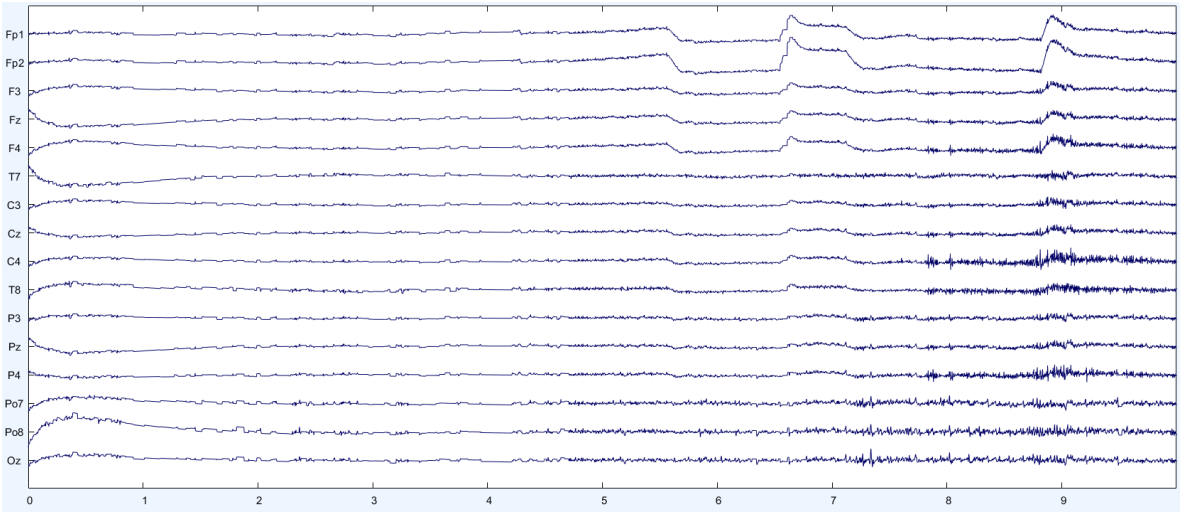


Figure 28: Data high filtered at 0.2 Hz using 4th Butterworth

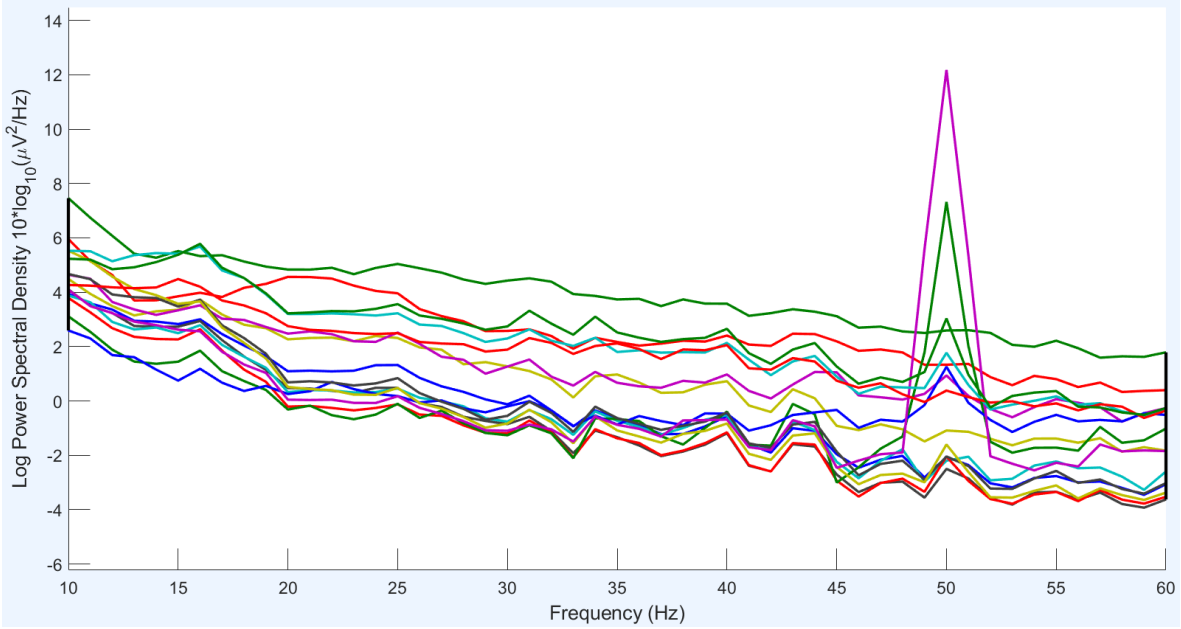


Figure 29: Line Noise

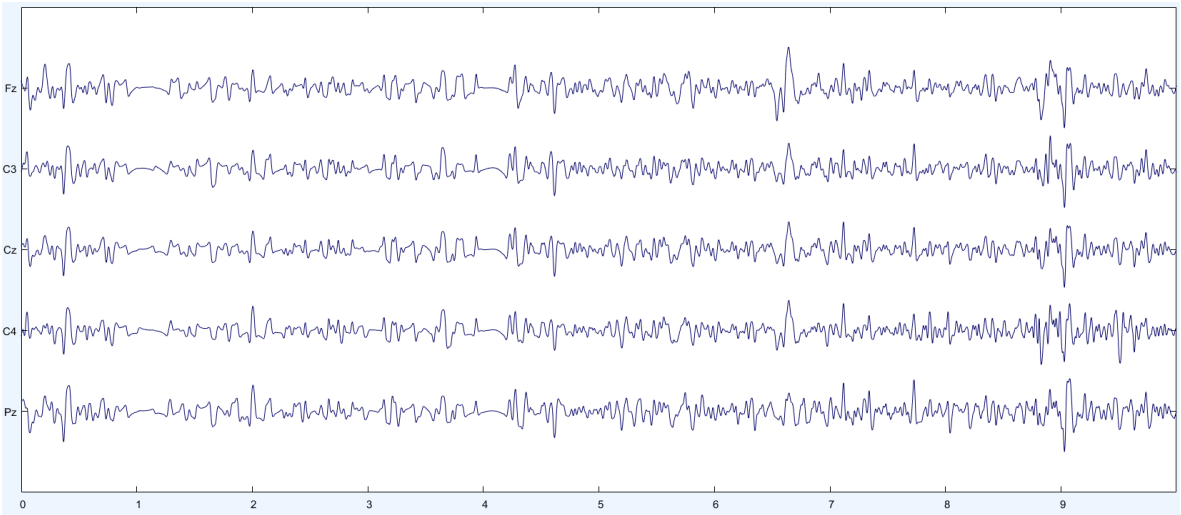


Figure 30: Data filtered by bandpass between 4 Hz and 30 Hz

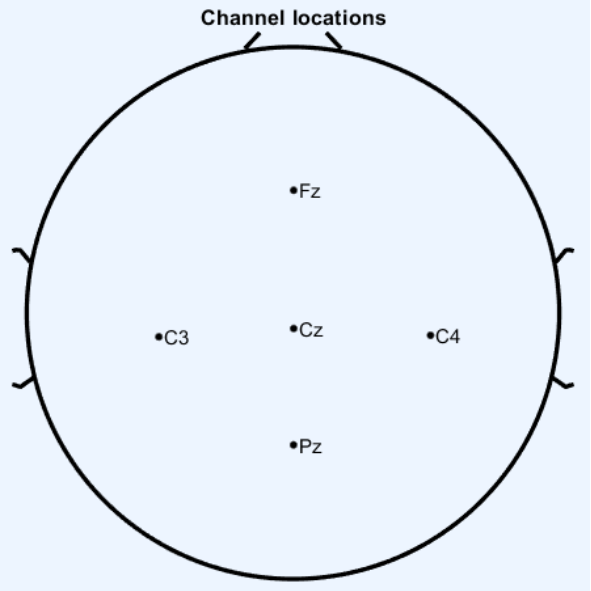


Figure 31: Channel Locations (Fz, C3, Cz, C4, and Pz)

A.3 PLOTS FROM GRAND AVERAGE OF PERTURBATIONS OVER FZ, C3, C4 AND PZ ELECTRODES

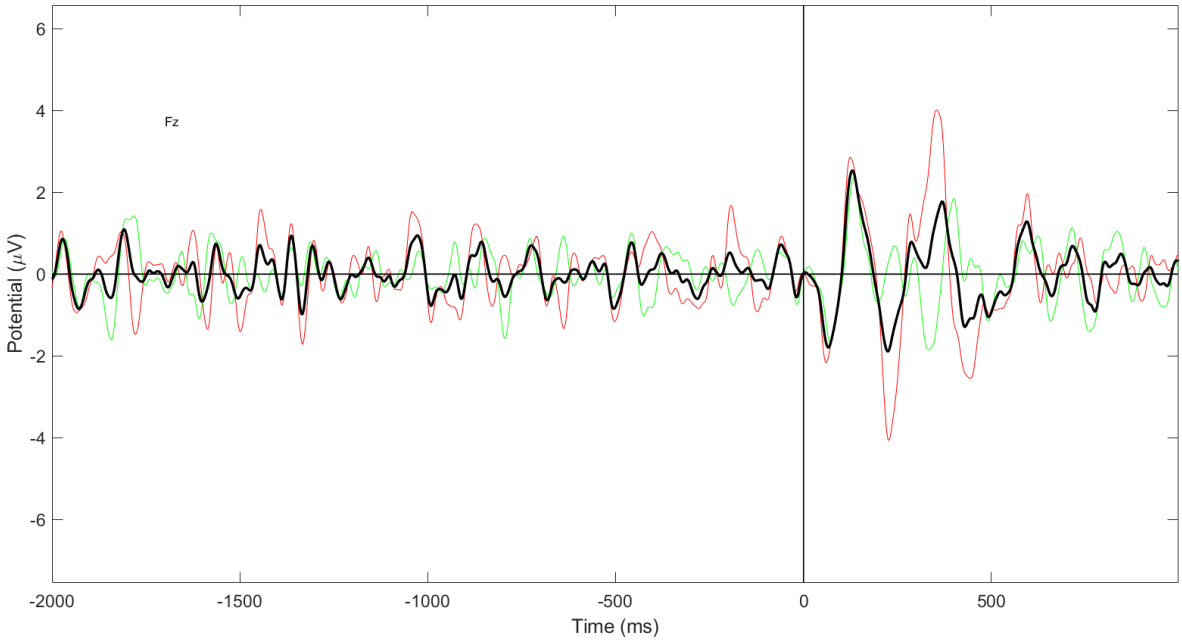


Figure 32: Fz - Grand Average PEP for (Speed = 1.6km/h)

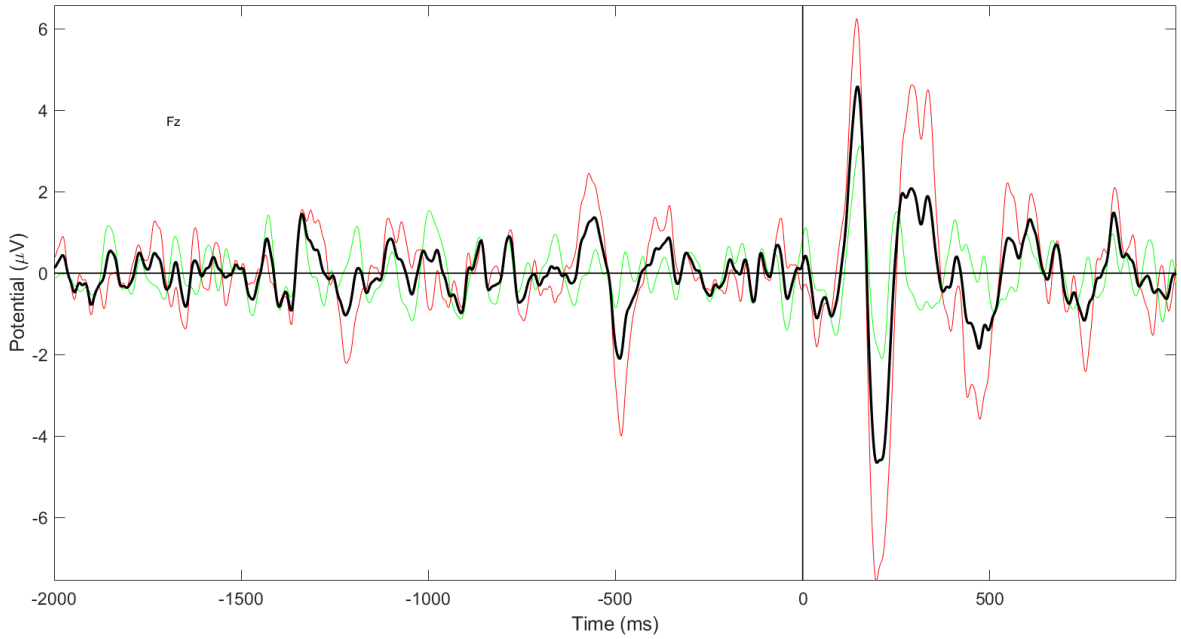


Figure 33: Fz - Grand Average PEP for (Speed = 2.5km/h)

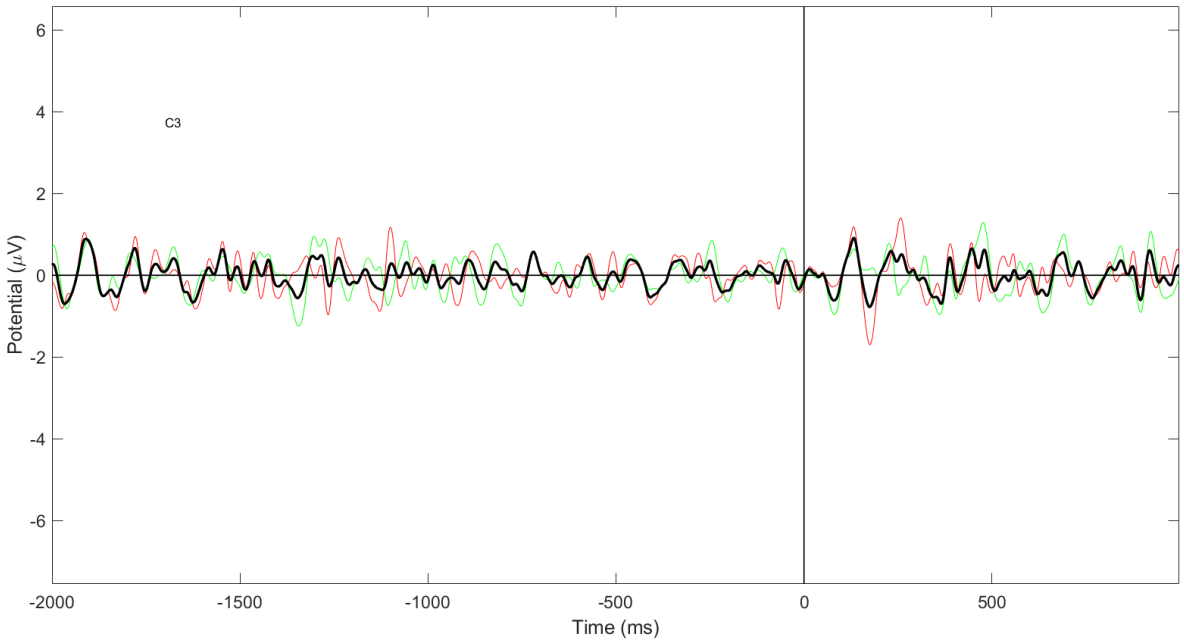


Figure 34: C3 - Grand Average PEP for (Speed = 1.6km/h)

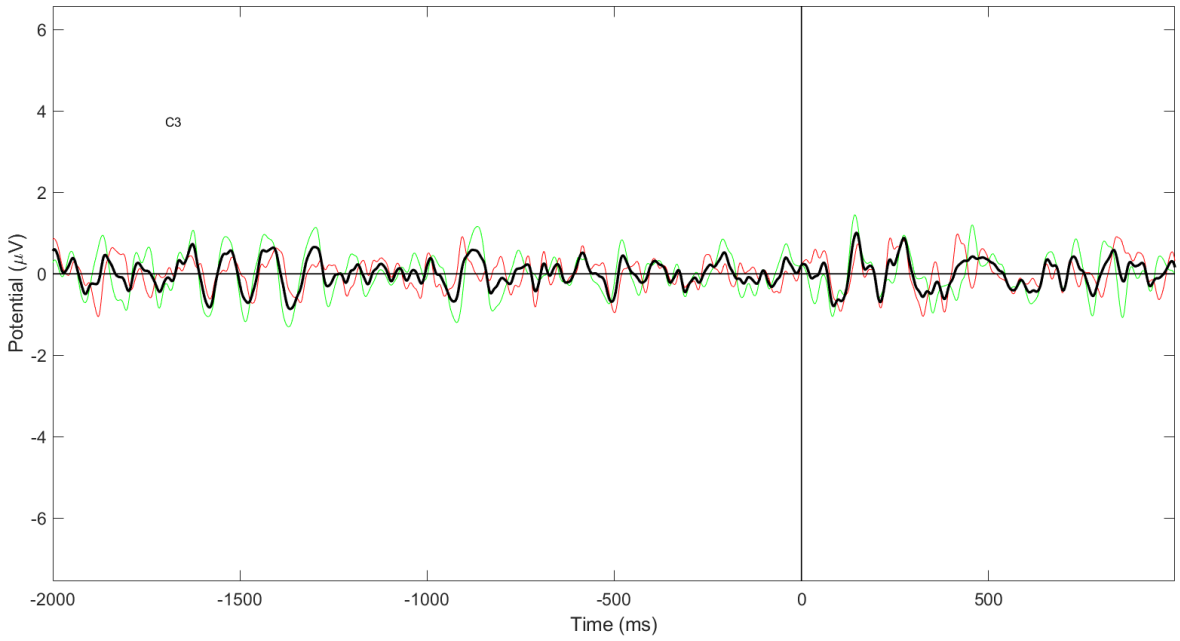


Figure 35: C3 - Grand Average PEP for (Speed = 2.5km/h)

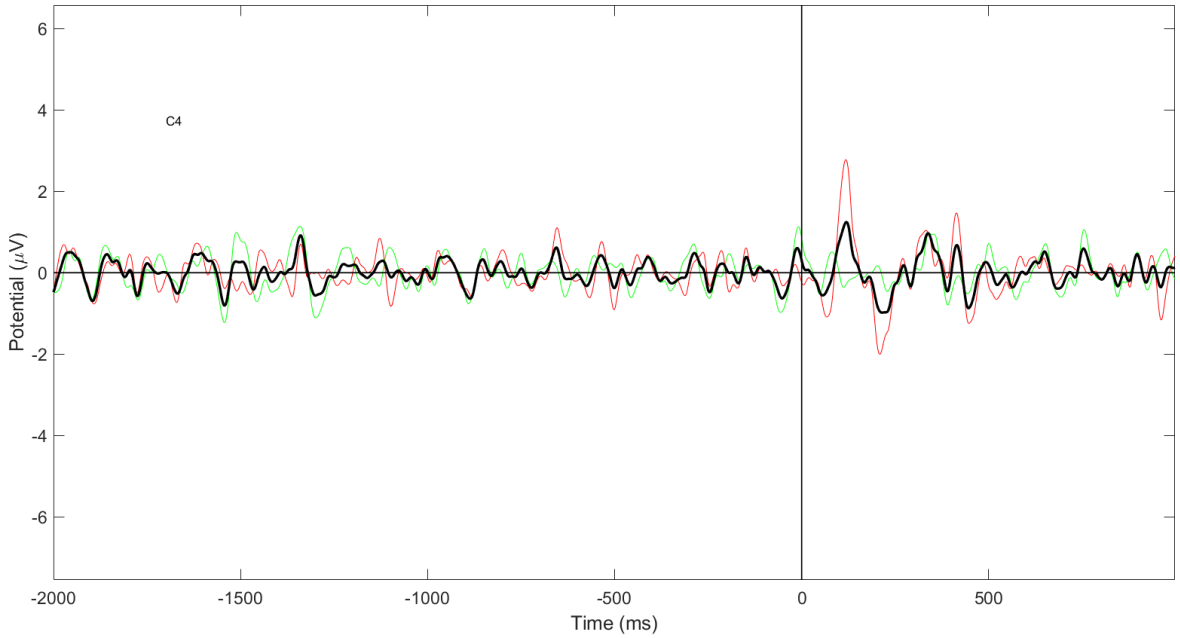


Figure 36: C4 - Grand Average PEP for (Speed = 1.6km/h)

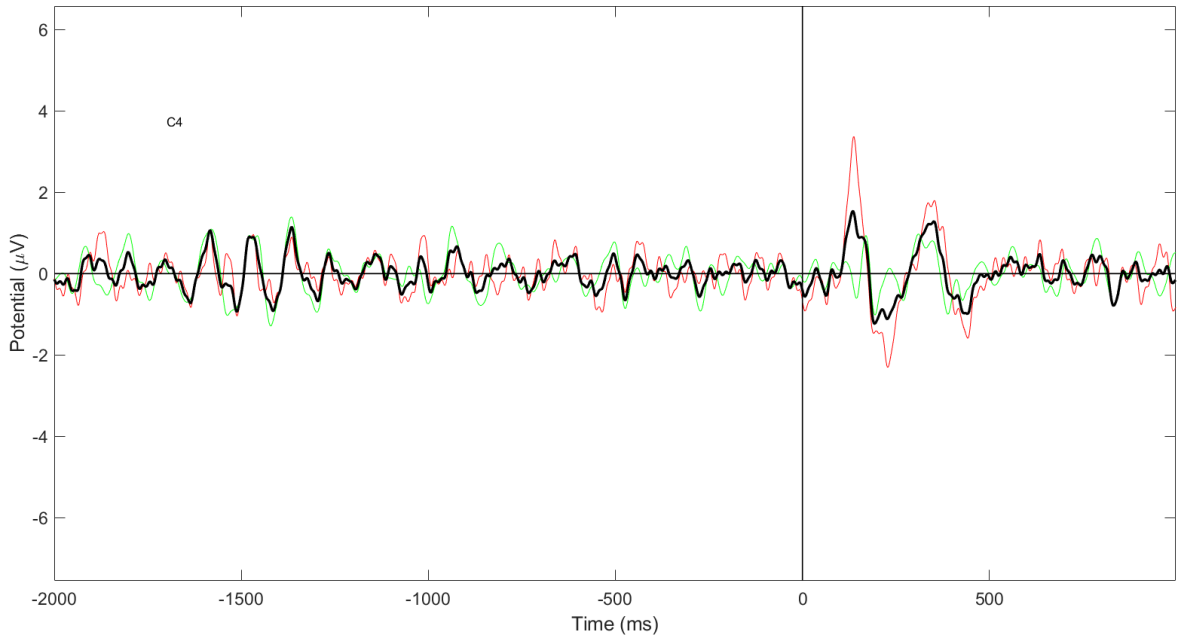


Figure 37: C4 - Grand Average PEP for (Speed = 2.5km/h)

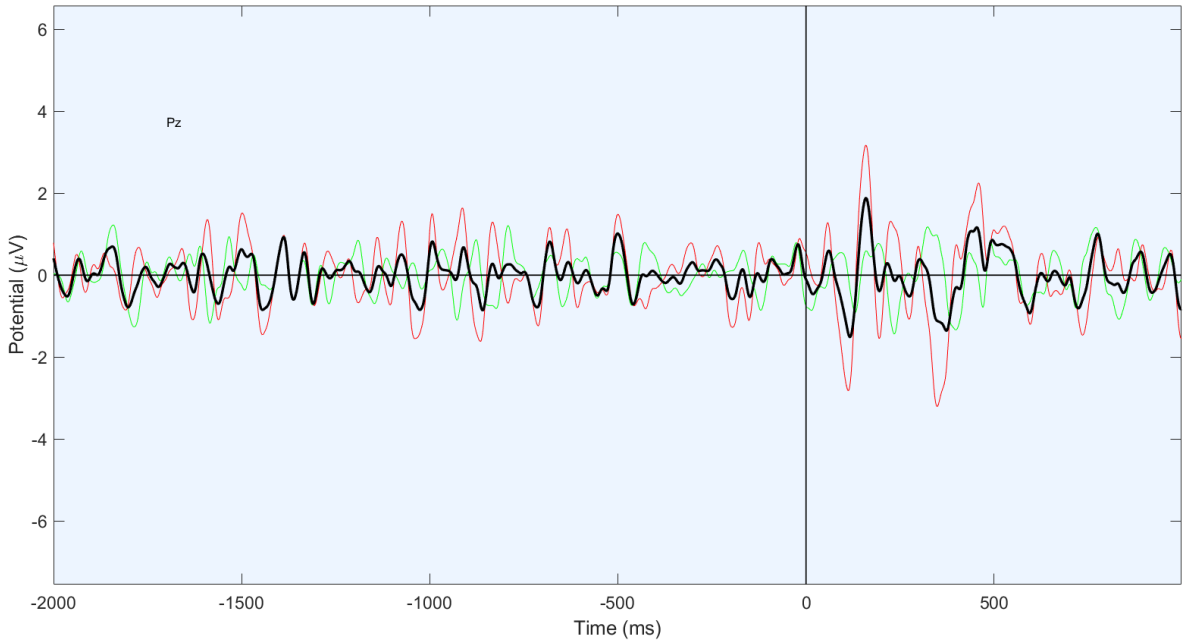


Figure 38: Pz - Grand Average PEP for (Speed = 1.6km/h)

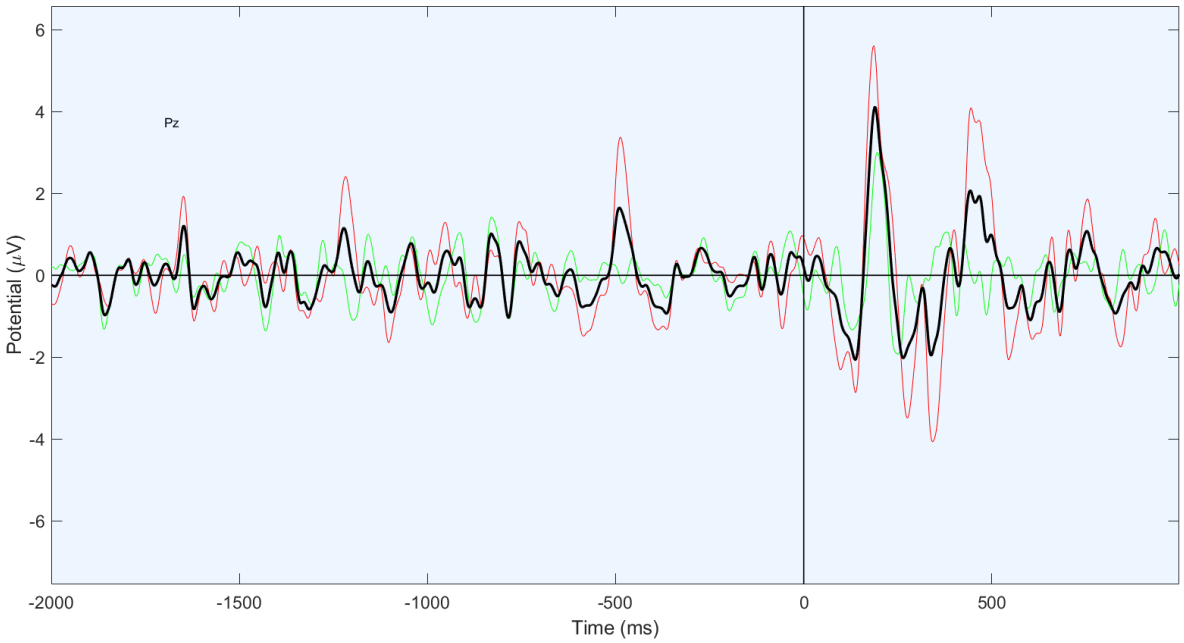


Figure 39: Pz - Grand Average PEP for (Speed = 2.5km/h)

A.4 SUBDATASETS FROM EEGLAB

Table 12: Dataset extracted from EEGLAB

Dataset from EEGLAB	Month	Treadmill Speed	Channels/ICA	Epochs Qty
Data_16_ago_CAR_Channels_ERP	August	1.6km	Channels	30
Data_25_ago_CAR_Channels_ERP	August	2.5km	Channels	40
Data_16_set_CAR_Channels_ERP	September	1.6km	Channels	30
Data_25_set_CAR_Channels_ERP	September	2.5km	Channels	39
Data_16_ago_CAR_ICA_ERP	August	1.6km	ICA	30
Data_25_ago_CAR_ICA_ERP	August	2.5km	ICA	40
Data_16_set_CAR_ICA_ERP	September	1.6km	ICA	30
Data_25_set_CAR_ICA_ERP	September	2.5km	ICA	39

B

SLIP-LIKE PERTURBATION CLASSIFICATION METRICS FOR DEEP LEARNING MODELS - CHANNELS

B.1 OUTCOMES OF ENTIRE DATASET (NO FILTER):

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
all	Cz	cnn	Batch16	80	578	79.17%	69.21%	NaN	72.39%	75.78%	79.17%
all	Cz	cnn	Batch16	50	276	80.00%	76.96%	NaN	83.87%	77.52%	80.00%
all	Cz	cnn	Batch32	80	340	80.16%	73.46%	NaN	76.66%	74.99%	80.16%
all	Cz	cnn	Batch32	50	182	80.48%	77.80%	59.97%	83.34%	77.19%	80.48%
all	Cz	cnn	Batch64	80	242	79.96%	75.43%	57.62%	83.99%	74.67%	79.96%
all	Cz	cnn	Batch64	50	143	80.00%	77.30%	58.47%	82.44%	76.43%	80.00%
all	Cz	cnn	Batch128	80	192	79.96%	75.52%	57.95%	85.03%	74.17%	79.96%
all	Cz	cnn	Batch128	50	119	81.43%	79.30%	61.41%	83.48%	78.17%	81.43%
all	2Ch	cnn	Batch16	80	558	79.96%	72.97%	NaN	76.47%	81.78%	79.96%
all	2Ch	cnn	Batch16	50	274	81.43%	79.79%	NaN	82.84%	81.11%	81.43%
all	2Ch	cnn	Batch32	80	334	78.77%	76.61%	58.58%	79.57%	79.65%	78.77%
all	2Ch	cnn	Batch32	50	179	83.81%	82.62%	67.03%	84.47%	82.75%	83.81%
all	2Ch	cnn	Batch64	80	244	76.98%	75.96%	55.29%	76.98%	78.39%	76.98%
all	2Ch	cnn	Batch64	50	142	83.81%	82.31%	66.40%	84.59%	82.02%	83.81%
all	2Ch	cnn	Batch128	80	192	80.56%	79.41%	58.94%	79.21%	79.73%	80.56%
all	2Ch	cnn	Batch128	50	119	82.86%	81.73%	64.05%	82.85%	81.23%	82.86%
all	5Ch	cnn	Batch16	80	559	81.94%	75.61%	NaN	76.89%	82.15%	81.94%
all	5Ch	cnn	Batch16	50	274	87.14%	85.90%	NaN	86.99%	86.73%	87.14%
all	5Ch	cnn	Batch32	80	336	81.15%	76.85%	NaN	77.18%	80.18%	81.15%
all	5Ch	cnn	Batch32	50	180	87.14%	85.93%	73.87%	88.34%	85.76%	87.14%
all	5Ch	cnn	Batch64	80	244	83.13%	81.10%	63.80%	83.20%	80.81%	83.13%
all	5Ch	cnn	Batch64	50	143	86.19%	84.75%	71.57%	87.77%	84.04%	86.19%
all	5Ch	cnn	Batch128	80	204	82.94%	81.28%	63.39%	83.13%	80.34%	82.94%
all	5Ch	cnn	Batch128	50	121	83.33%	80.83%	67.30%	88.49%	79.49%	83.33%
all	16CH	cnn	Batch16	80	573	85.52%	77.68%	NaN	77.06%	85.36%	85.52%
all	16CH	cnn	Batch16	50	277	88.10%	86.48%	NaN	91.03%	86.86%	88.10%
all	16CH	cnn	Batch32	80	382	84.92%	80.79%	NaN	79.99%	83.33%	84.92%
all	16CH	cnn	Batch32	50	198	86.19%	83.92%	71.89%	89.01%	83.63%	86.19%
all	16CH	cnn	Batch64	80	272	84.72%	82.33%	68.05%	87.05%	81.54%	84.72%
all	16CH	cnn	Batch64	50	158	80.95%	77.31%	61.98%	86.76%	76.45%	80.95%
all	16CH	cnn	Batch128	80	209	85.32%	83.03%	68.35%	87.10%	81.58%	85.32%
all	16CH	cnn	Batch128	50	127	81.90%	79.23%	63.36%	86.00%	77.90%	81.90%

Figure 40: All subjects and speeds - Table filter by CNN model.

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
all	Cz	lstm	Batch16	80	593	81.75%	72.45%	NaN	73.84%	78.84%	81.75%
all	Cz	lstm	Batch16	50	283	80.00%	77.02%	NaN	82.12%	77.49%	80.00%
all	Cz	lstm	Batch32	80	359	80.16%	74.25%	NaN	75.42%	77.11%	80.16%
all	Cz	lstm	Batch32	50	190	81.43%	79.15%	62.40%	84.69%	78.23%	81.43%
all	Cz	lstm	Batch64	80	240	79.76%	76.74%	56.85%	80.74%	76.63%	79.76%
all	Cz	lstm	Batch64	50	141	81.90%	79.76%	63.05%	84.55%	78.94%	81.90%
all	Cz	lstm	Batch128	80	182	80.16%	77.83%	56.70%	79.78%	77.00%	80.16%
all	Cz	lstm	Batch128	50	113	77.62%	74.72%	53.59%	80.11%	73.91%	77.62%
all	2Ch	lstm	Batch16	80	585	81.35%	71.50%	NaN	73.12%	77.47%	81.35%
all	2Ch	lstm	Batch16	50	282	82.86%	81.25%	NaN	83.83%	81.71%	82.86%
all	2Ch	lstm	Batch32	80	356	81.94%	77.20%	NaN	79.68%	78.37%	81.94%
all	2Ch	lstm	Batch32	50	188	80.48%	78.63%	59.65%	81.37%	78.56%	80.48%
all	2Ch	lstm	Batch64	80	239	79.56%	76.90%	55.70%	78.65%	77.19%	79.56%
all	2Ch	lstm	Batch64	50	139	82.86%	81.16%	64.21%	83.66%	80.78%	82.86%
all	2Ch	lstm	Batch128	80	182	80.95%	78.85%	58.62%	80.55%	78.15%	80.95%
all	2Ch	lstm	Batch128	50	116	83.81%	82.34%	66.63%	85.32%	81.47%	83.81%
all	5Ch	lstm	Batch16	80	580	80.75%	72.72%	NaN	74.15%	77.24%	80.75%
all	5Ch	lstm	Batch16	50	284	83.81%	81.12%	NaN	87.41%	81.78%	83.81%
all	5Ch	lstm	Batch32	80	357	83.53%	79.23%	NaN	82.54%	80.93%	83.53%
all	5Ch	lstm	Batch32	50	188	82.86%	79.98%	65.97%	87.31%	79.81%	82.86%
all	5Ch	lstm	Batch64	80	238	85.12%	82.35%	69.18%	88.30%	81.65%	85.12%
all	5Ch	lstm	Batch64	50	142	81.43%	78.81%	61.54%	83.74%	78.28%	81.43%
all	5Ch	lstm	Batch128	80	196	85.52%	83.45%	69.58%	88.05%	81.84%	85.52%
all	5Ch	lstm	Batch128	50	117	83.33%	81.32%	66.12%	86.42%	80.02%	83.33%
all	16CH	lstm	Batch16	80	592	81.35%	68.41%	NaN	69.86%	71.33%	81.35%
all	16CH	lstm	Batch16	50	286	82.38%	79.28%	NaN	84.37%	79.40%	82.38%
all	16CH	lstm	Batch32	80	423	84.72%	78.97%	NaN	80.78%	79.61%	84.72%
all	16CH	lstm	Batch32	50	216	80.95%	78.16%	60.67%	83.88%	77.31%	80.95%
all	16CH	lstm	Batch64	80	271	84.72%	82.07%	67.70%	86.78%	81.31%	84.72%
all	16CH	lstm	Batch64	50	154	83.33%	81.34%	66.12%	86.39%	80.13%	83.33%
all	16CH	lstm	Batch128	80	200	82.14%	80.10%	60.90%	81.74%	79.22%	82.14%
all	16CH	lstm	Batch128	50	124	80.48%	78.67%	58.87%	81.22%	77.76%	80.48%

Figure 41: All subjects and speeds - Table filter by LSTM model.

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
all	Cz	gru	Batch16	80	590	80.16%	70.10%	NaN	73.19%	74.61%	80.16%
all	Cz	gru	Batch16	50	283	79.05%	75.97%	NaN	81.68%	77.27%	79.05%
all	Cz	gru	Batch32	80	354	80.36%	74.44%	NaN	75.34%	77.21%	80.36%
all	Cz	gru	Batch32	50	186	78.57%	75.08%	55.59%	82.02%	74.30%	78.57%
all	Cz	gru	Batch64	80	236	79.56%	76.12%	56.36%	81.09%	75.86%	79.56%
all	Cz	gru	Batch64	50	139	79.05%	75.80%	56.65%	82.25%	74.98%	79.05%
all	Cz	gru	Batch128	80	176	79.76%	76.56%	56.14%	81.19%	75.35%	79.76%
all	Cz	gru	Batch128	50	110	77.62%	73.70%	54.57%	82.62%	72.83%	77.62%
all	2Ch	gru	Batch16	80	583	79.17%	67.02%	NaN	70.03%	72.08%	79.17%
all	2Ch	gru	Batch16	50	280	81.90%	79.83%	NaN	83.59%	80.54%	81.90%
all	2Ch	gru	Batch32	80	352	80.16%	75.14%	NaN	77.91%	77.28%	80.16%
all	2Ch	gru	Batch32	50	186	80.95%	78.88%	60.90%	82.61%	78.62%	80.95%
all	2Ch	gru	Batch64	80	235	81.35%	77.98%	59.57%	82.38%	77.65%	81.35%
all	2Ch	gru	Batch64	50	137	82.86%	80.59%	64.28%	84.70%	79.92%	82.86%
all	2Ch	gru	Batch128	80	176	80.36%	77.37%	57.40%	81.38%	76.30%	80.36%
all	2Ch	gru	Batch128	50	111	80.00%	76.98%	59.40%	84.19%	75.80%	80.00%
all	5Ch	gru	Batch16	80	580	83.53%	72.55%	NaN	73.46%	74.99%	83.53%
all	5Ch	gru	Batch16	50	277	80.00%	77.53%	NaN	82.27%	77.05%	80.00%
all	5Ch	gru	Batch32	80	352	84.72%	79.74%	NaN	82.09%	80.85%	84.72%
all	5Ch	gru	Batch32	50	186	86.19%	84.24%	72.80%	90.33%	83.02%	86.19%
all	5Ch	gru	Batch64	80	234	83.73%	80.68%	66.04%	86.93%	79.90%	83.73%
all	5Ch	gru	Batch64	50	137	80.95%	77.90%	61.92%	85.96%	76.78%	80.95%
all	5Ch	gru	Batch128	80	192	85.91%	83.80%	70.54%	88.81%	82.15%	85.91%
all	5Ch	gru	Batch128	50	115	81.90%	79.18%	63.96%	86.79%	77.81%	81.90%
all	16CH	gru	Batch16	80	581	84.52%	75.60%	NaN	76.63%	78.77%	84.52%
all	16CH	gru	Batch16	50	284	83.81%	81.19%	NaN	86.63%	80.63%	83.81%
all	16CH	gru	Batch32	80	417	84.92%	80.40%	NaN	80.06%	82.06%	84.92%
all	16CH	gru	Batch32	50	212	82.86%	81.00%	64.53%	84.61%	80.17%	82.86%
all	16CH	gru	Batch64	80	269	84.33%	81.51%	66.17%	85.47%	81.29%	84.33%
all	16CH	gru	Batch64	50	151	86.19%	84.14%	71.80%	88.85%	83.43%	86.19%
all	16CH	gru	Batch128	80	195	83.93%	81.48%	64.96%	84.98%	80.26%	83.93%
all	16CH	gru	Batch128	50	121	83.33%	81.98%	64.97%	83.88%	81.14%	83.33%

Figure 42: All subjects and speeds - Table filter by GRU model.

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
all	Cz	cnn-lstm	Batch16	80	816	81.35%	73.43%	NaN	74.87%	81.27%	81.35%
all	Cz	cnn-lstm	Batch16	50	380	80.00%	75.65%	NaN	79.50%	76.48%	80.00%
all	Cz	cnn-lstm	Batch32	80	483	79.76%	74.41%	NaN	74.25%	77.93%	79.76%
all	Cz	cnn-lstm	Batch32	50	244	81.90%	79.18%	63.06%	85.44%	78.28%	81.90%
all	Cz	cnn-lstm	Batch64	80	328	79.17%	76.32%	55.77%	79.57%	76.54%	79.17%
all	Cz	cnn-lstm	Batch64	50	180	79.05%	75.81%	57.30%	83.03%	74.91%	79.05%
all	Cz	cnn-lstm	Batch128	80	257	79.37%	76.15%	54.75%	79.93%	75.08%	79.37%
all	Cz	cnn-lstm	Batch128	50	143	76.19%	70.46%	53.74%	85.80%	70.19%	76.19%
all	2Ch	cnn-lstm	Batch16	80	804	79.17%	72.25%	NaN	76.42%	80.19%	79.17%
all	2Ch	cnn-lstm	Batch16	50	372	79.52%	78.61%	NaN	80.31%	80.35%	79.52%
all	2Ch	cnn-lstm	Batch32	80	477	78.57%	76.26%	57.64%	78.35%	79.50%	78.57%
all	2Ch	cnn-lstm	Batch32	50	242	81.43%	80.16%	62.35%	81.91%	80.64%	81.43%
all	2Ch	cnn-lstm	Batch64	80	329	79.96%	78.78%	60.04%	79.30%	80.82%	79.96%
all	2Ch	cnn-lstm	Batch64	50	179	82.38%	80.83%	63.12%	83.03%	80.21%	82.38%
all	2Ch	cnn-lstm	Batch128	80	255	78.57%	77.04%	54.40%	77.27%	77.14%	78.57%
all	2Ch	cnn-lstm	Batch128	50	143	83.81%	82.74%	66.07%	83.88%	82.23%	83.81%
all	5Ch	cnn-lstm	Batch16	80	795	79.96%	74.05%	NaN	77.16%	82.63%	79.96%
all	5Ch	cnn-lstm	Batch16	50	373	85.24%	83.07%	NaN	88.27%	82.95%	85.24%
all	5Ch	cnn-lstm	Batch32	80	477	84.52%	80.41%	NaN	82.01%	82.67%	84.52%
all	5Ch	cnn-lstm	Batch32	50	241	85.71%	84.00%	71.07%	88.45%	82.97%	85.71%
all	5Ch	cnn-lstm	Batch64	80	330	79.76%	78.40%	58.55%	78.77%	79.84%	79.76%
all	5Ch	cnn-lstm	Batch64	50	179	82.86%	80.93%	65.36%	85.66%	80.10%	82.86%
all	5Ch	cnn-lstm	Batch128	80	268	83.93%	82.27%	65.64%	84.36%	81.42%	83.93%
all	5Ch	cnn-lstm	Batch128	50	149	86.19%	84.49%	72.74%	90.08%	83.04%	86.19%
all	16CH	cnn-lstm	Batch16	80	808	86.11%	77.61%	NaN	77.39%	82.53%	86.11%
all	16CH	cnn-lstm	Batch16	50	377	86.67%	84.90%	NaN	88.76%	85.75%	86.67%
all	16CH	cnn-lstm	Batch32	80	566	84.13%	79.78%	61.95%	78.80%	82.47%	84.13%
all	16CH	cnn-lstm	Batch32	50	278	87.14%	85.29%	73.84%	89.41%	84.97%	87.14%
all	16CH	cnn-lstm	Batch64	80	365	84.52%	82.77%	66.66%	83.83%	82.94%	84.52%
all	16CH	cnn-lstm	Batch64	50	199	83.81%	81.91%	66.64%	85.68%	81.31%	83.81%
all	16CH	cnn-lstm	Batch128	80	273	85.71%	84.49%	69.14%	84.85%	84.29%	85.71%
all	16CH	cnn-lstm	Batch128	50	159	83.33%	80.81%	66.97%	88.27%	79.31%	83.33%

Figure 43: All subjects and speeds - Table filter by CNN-LSTM model.

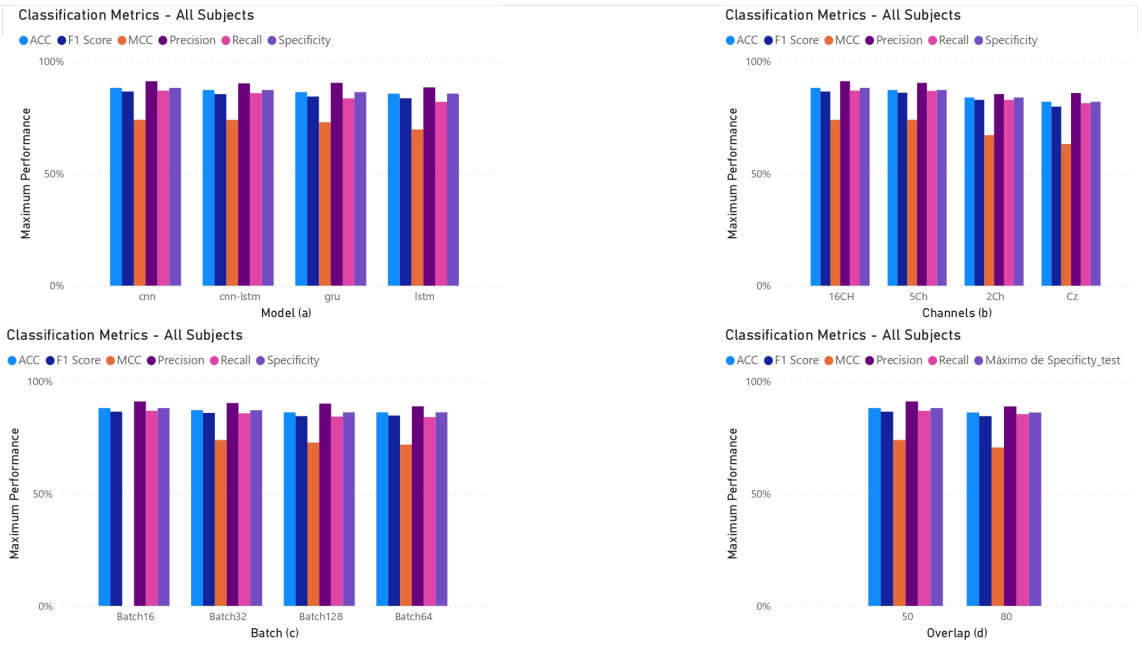


Figure 44: All subjects and speeds - Classification Metrics Graphics sort of by (a) Models, (b) Channels, (c) Batch and (d) Overlap

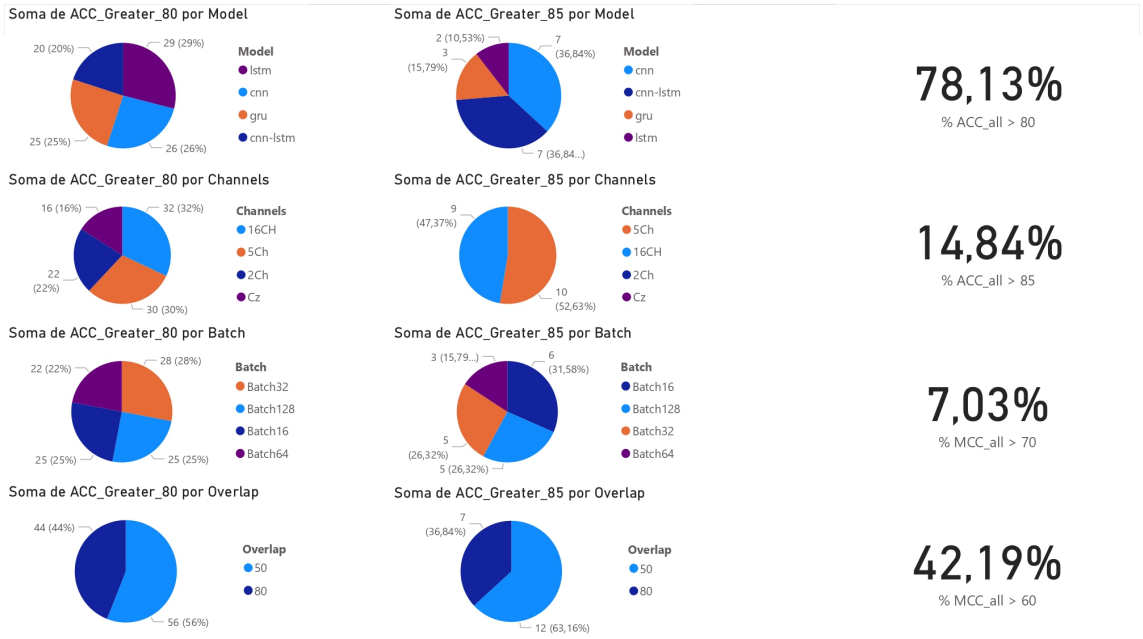


Figure 45: All subjects and speeds - Accuracy greater 80% and 85% sort of by Model, Channels, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%.

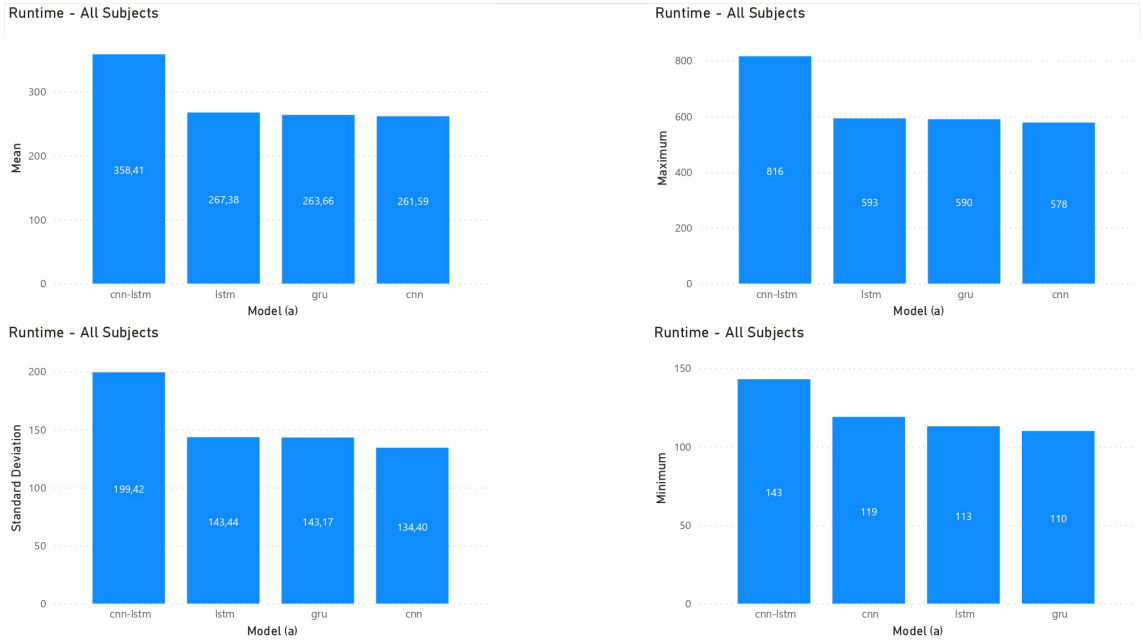


Figure 46: All subjects and speeds - Runtime sort of by Model.

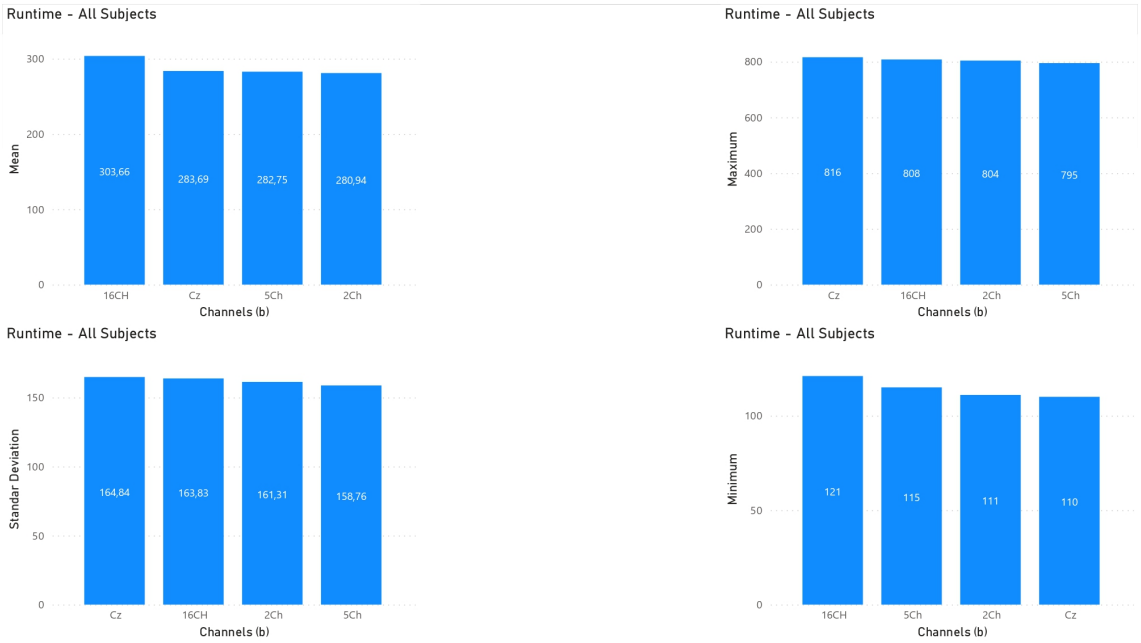


Figure 47: All subjects and speeds - Runtime sort of by Channel.

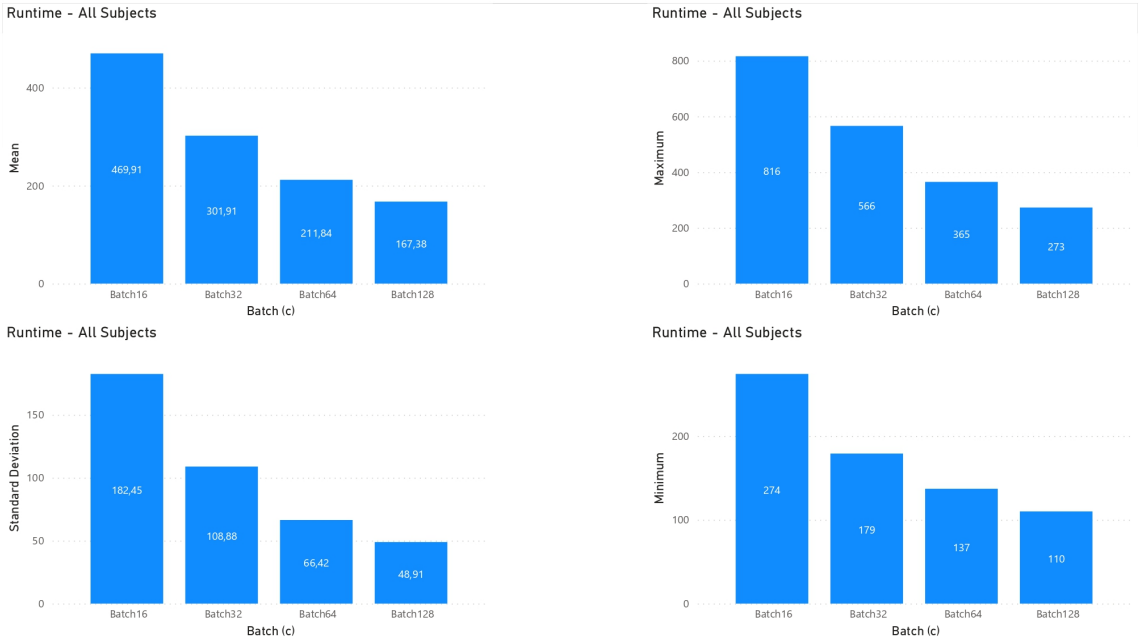


Figure 48: All subjects and speeds - Runtime sort of by Batch Size.

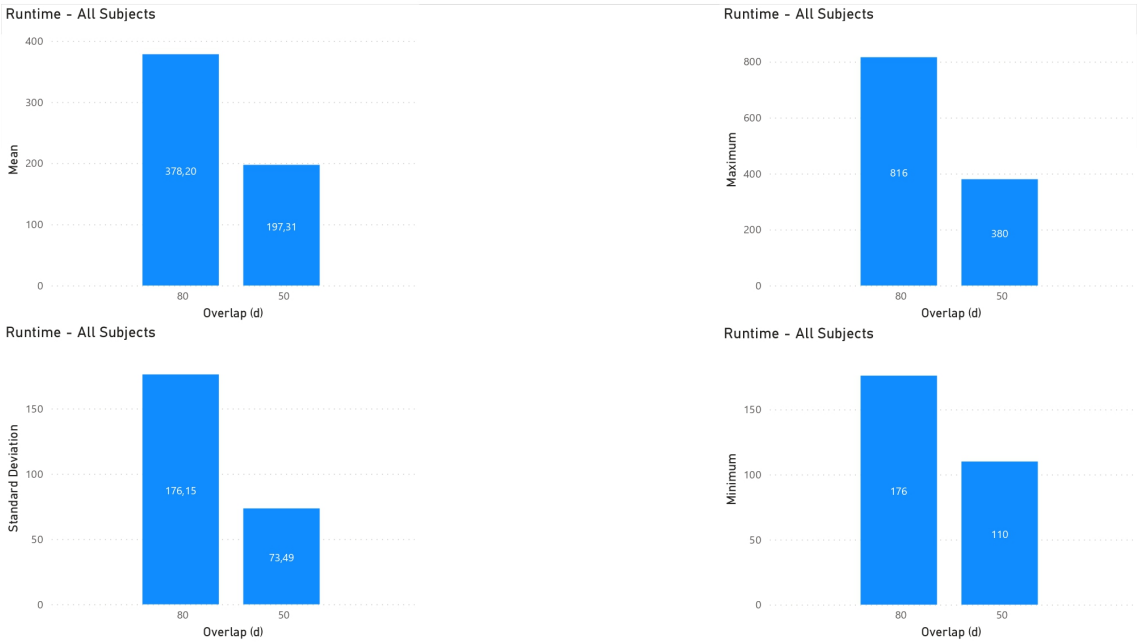


Figure 49: All subjects and speeds - Runtime sort of by Overlap.

B.2 OUTCOMES FILTERED SPEED = 1.6KM/H

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
1.6KM	Cz	cnn	Batch16	80	292	74.11%	68.43%	48.60%	74.36%	76.25%	74.11%
1.6KM	Cz	cnn	Batch16	50	166	82.14%	79.07%	65.21%	86.52%	79.52%	82.14%
1.6KM	Cz	cnn	Batch32	80	193	77.08%	74.57%	50.89%	75.99%	75.00%	77.08%
1.6KM	Cz	cnn	Batch32	50	125	77.14%	71.16%	53.43%	84.37%	71.23%	77.14%
1.6KM	Cz	cnn	Batch64	80	151	77.38%	73.92%	48.69%	75.38%	73.37%	77.38%
1.6KM	Cz	cnn	Batch64	50	109	75.71%	69.38%	50.89%	83.67%	69.45%	75.71%
1.6KM	Cz	cnn	Batch128	80	129	80.36%	76.88%	56.08%	80.59%	75.74%	80.36%
1.6KM	Cz	cnn	Batch128	50	100	73.57%	65.54%	47.67%	84.77%	66.57%	73.57%
1.6KM	2Ch	cnn	Batch16	80	291	72.32%	66.95%	44.61%	71.27%	75.14%	72.32%
1.6KM	2Ch	cnn	Batch16	50	166	75.71%	74.65%	51.84%	75.38%	76.57%	75.71%
1.6KM	2Ch	cnn	Batch32	80	197	78.57%	77.33%	56.68%	77.45%	79.30%	78.57%
1.6KM	2Ch	cnn	Batch32	50	125	72.14%	71.89%	46.98%	73.20%	73.79%	72.14%
1.6KM	2Ch	cnn	Batch64	80	151	77.38%	76.41%	54.71%	76.58%	78.19%	77.38%
1.6KM	2Ch	cnn	Batch64	50	109	72.14%	71.27%	42.93%	71.26%	71.68%	72.14%
1.6KM	2Ch	cnn	Batch128	80	130	79.17%	76.71%	53.97%	77.89%	76.11%	79.17%
1.6KM	2Ch	cnn	Batch128	50	101	75.00%	73.95%	48.05%	73.94%	74.11%	75.00%
1.6KM	5Ch	cnn	Batch16	80	292	81.85%	74.24%	53.39%	76.62%	76.53%	81.85%
1.6KM	5Ch	cnn	Batch16	50	168	73.57%	71.99%	46.40%	73.18%	73.29%	73.57%
1.6KM	5Ch	cnn	Batch32	80	193	74.70%	73.42%	52.34%	75.55%	76.88%	74.70%
1.6KM	5Ch	cnn	Batch32	50	125	65.00%	64.11%	36.96%	68.67%	68.32%	65.00%
1.6KM	5Ch	cnn	Batch64	80	151	80.65%	78.79%	58.23%	78.86%	79.41%	80.65%
1.6KM	5Ch	cnn	Batch64	50	110	77.14%	76.57%	53.81%	76.48%	77.34%	77.14%
1.6KM	5Ch	cnn	Batch128	80	134	78.87%	78.03%	57.15%	77.83%	79.35%	78.87%
1.6KM	5Ch	cnn	Batch128	50	107	80.71%	79.91%	59.97%	79.86%	80.11%	80.71%
1.6KM	16CH	cnn	Batch16	80	315	83.63%	78.71%	63.98%	81.50%	83.29%	83.63%
1.6KM	16CH	cnn	Batch16	50	193	77.86%	76.20%	55.32%	77.30%	78.10%	77.86%
1.6KM	16CH	cnn	Batch32	80	223	84.23%	83.45%	71.53%	84.59%	87.12%	84.23%
1.6KM	16CH	cnn	Batch32	50	148	77.86%	76.98%	54.84%	77.28%	77.57%	77.86%
1.6KM	16CH	cnn	Batch64	80	175	84.82%	84.18%	69.82%	83.99%	85.89%	84.82%
1.6KM	16CH	cnn	Batch64	50	123	83.57%	81.80%	65.30%	84.53%	80.94%	83.57%
1.6KM	16CH	cnn	Batch128	80	143	88.10%	86.91%	74.08%	87.58%	86.52%	88.10%
1.6KM	16CH	cnn	Batch128	50	105	84.29%	83.21%	66.80%	84.10%	82.71%	84.29%

Figure 50: Table filter by CNN model (Speed = 1.6km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
1.6KM	Cz	lstm	Batch16	80	306	81.85%	74.15%	NaN	74.91%	77.66%	81.85%
1.6KM	Cz	lstm	Batch16	50	169	83.57%	80.38%	66.31%	86.45%	80.95%	83.57%
1.6KM	Cz	lstm	Batch32	80	202	81.55%	80.06%	62.92%	81.33%	81.65%	81.55%
1.6KM	Cz	lstm	Batch32	50	129	81.43%	78.10%	62.30%	86.16%	76.95%	81.43%
1.6KM	Cz	lstm	Batch64	80	150	82.44%	80.60%	62.99%	82.22%	80.98%	82.44%
1.6KM	Cz	lstm	Batch64	50	108	80.71%	77.79%	60.43%	84.61%	76.55%	80.71%
1.6KM	Cz	lstm	Batch128	80	123	84.82%	83.08%	66.97%	84.84%	82.19%	84.82%
1.6KM	Cz	lstm	Batch128	50	99	79.29%	75.30%	58.28%	85.20%	74.21%	79.29%
1.6KM	2Ch	lstm	Batch16	80	308	78.57%	69.92%	NaN	72.44%	75.47%	78.57%
1.6KM	2Ch	lstm	Batch16	50	170	72.14%	70.02%	42.30%	71.33%	71.00%	72.14%
1.6KM	2Ch	lstm	Batch32	80	204	72.62%	69.75%	40.93%	70.38%	70.63%	72.62%
1.6KM	2Ch	lstm	Batch32	50	130	77.14%	75.21%	51.38%	76.74%	74.72%	77.14%
1.6KM	2Ch	lstm	Batch64	80	149	81.55%	79.31%	59.61%	80.52%	79.17%	81.55%
1.6KM	2Ch	lstm	Batch64	50	109	72.86%	71.95%	44.22%	71.89%	72.33%	72.86%
1.6KM	2Ch	lstm	Batch128	80	125	75.89%	74.59%	50.27%	74.88%	75.39%	75.89%
1.6KM	2Ch	lstm	Batch128	50	99	80.71%	79.47%	59.21%	80.09%	79.12%	80.71%
1.6KM	5Ch	lstm	Batch16	80	305	81.25%	68.88%	NaN	73.11%	70.22%	81.25%
1.6KM	5Ch	lstm	Batch16	50	172	77.14%	73.69%	51.07%	77.71%	73.81%	77.14%
1.6KM	5Ch	lstm	Batch32	80	215	78.27%	75.30%	53.99%	77.78%	76.32%	78.27%
1.6KM	5Ch	lstm	Batch32	50	129	75.71%	73.31%	48.26%	75.60%	72.78%	75.71%
1.6KM	5Ch	lstm	Batch64	80	149	79.76%	77.99%	57.09%	78.70%	78.47%	79.76%
1.6KM	5Ch	lstm	Batch64	50	108	79.29%	76.91%	56.44%	80.66%	76.03%	79.29%
1.6KM	5Ch	lstm	Batch128	80	130	81.85%	79.97%	60.17%	80.59%	79.60%	81.85%
1.6KM	5Ch	lstm	Batch128	50	102	81.43%	80.12%	61.44%	81.72%	79.78%	81.43%
1.6KM	16CH	lstm	Batch16	80	366	79.46%	65.06%	NaN	71.05%	68.69%	79.46%
1.6KM	16CH	lstm	Batch16	50	203	73.57%	67.66%	40.88%	72.20%	68.81%	73.57%
1.6KM	16CH	lstm	Batch32	80	256	80.65%	75.02%	60.08%	87.66%	74.61%	80.65%
1.6KM	16CH	lstm	Batch32	50	158	72.86%	68.41%	39.90%	71.80%	68.34%	72.86%
1.6KM	16CH	lstm	Batch64	80	175	75.89%	66.37%	45.92%	82.25%	66.97%	75.89%
1.6KM	16CH	lstm	Batch64	50	123	77.86%	75.94%	54.21%	78.69%	75.72%	77.86%
1.6KM	16CH	lstm	Batch128	80	141	84.82%	82.24%	66.95%	86.57%	80.68%	84.82%
1.6KM	16CH	lstm	Batch128	50	105	71.43%	68.11%	38.42%	70.77%	67.77%	71.43%

Figure 51: Table filter by LSTM model (Speed = 1.6km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
1.6KM	Cz	gru	Batch16	80	303	81.25%	73.47%	NaN	75.59%	78.77%	81.25%
1.6KM	Cz	gru	Batch16	50	170	82.86%	80.95%	66.66%	84.94%	82.05%	82.86%
1.6KM	Cz	gru	Batch32	80	202	83.33%	80.99%	65.11%	84.18%	81.20%	83.33%
1.6KM	Cz	gru	Batch32	50	129	80.00%	77.53%	58.47%	82.57%	76.33%	80.00%
1.6KM	Cz	gru	Batch64	80	147	84.82%	81.96%	67.38%	86.95%	80.91%	84.82%
1.6KM	Cz	gru	Batch64	50	108	81.43%	78.29%	61.98%	85.68%	77.19%	81.43%
1.6KM	Cz	gru	Batch128	80	121	82.44%	79.14%	62.37%	85.48%	77.47%	82.44%
1.6KM	Cz	gru	Batch128	50	97	78.57%	74.24%	56.82%	84.80%	73.30%	78.57%
1.6KM	2Ch	gru	Batch16	80	306	78.27%	69.61%	43.92%	72.16%	72.43%	78.27%
1.6KM	2Ch	gru	Batch16	50	170	80.71%	78.06%	58.22%	80.05%	78.29%	80.71%
1.6KM	2Ch	gru	Batch32	80	201	82.14%	80.18%	61.45%	81.12%	80.35%	82.14%
1.6KM	2Ch	gru	Batch32	50	130	80.00%	78.83%	57.87%	79.11%	78.76%	80.00%
1.6KM	2Ch	gru	Batch64	80	148	81.25%	78.14%	57.86%	80.39%	77.67%	81.25%
1.6KM	2Ch	gru	Batch64	50	106	77.14%	74.51%	50.86%	77.00%	74.03%	77.14%
1.6KM	2Ch	gru	Batch128	80	121	79.46%	77.77%	55.55%	77.90%	77.65%	79.46%
1.6KM	2Ch	gru	Batch128	50	97	81.43%	79.10%	61.61%	84.14%	77.81%	81.43%
1.6KM	5Ch	gru	Batch16	80	324	82.14%	69.29%	NaN	70.72%	71.26%	82.14%
1.6KM	5Ch	gru	Batch16	50	170	77.86%	74.77%	52.28%	77.59%	75.00%	77.86%
1.6KM	5Ch	gru	Batch32	80	202	80.95%	78.73%	59.90%	80.57%	79.50%	80.95%
1.6KM	5Ch	gru	Batch32	50	130	80.00%	77.81%	58.45%	81.83%	76.85%	80.00%
1.6KM	5Ch	gru	Batch64	80	148	80.65%	77.93%	57.74%	80.70%	77.19%	80.65%
1.6KM	5Ch	gru	Batch64	50	107	81.43%	79.39%	61.72%	83.45%	78.49%	81.43%
1.6KM	5Ch	gru	Batch128	80	127	81.55%	79.99%	60.10%	80.45%	79.65%	81.55%
1.6KM	5Ch	gru	Batch128	50	101	80.71%	79.40%	59.48%	80.51%	79.00%	80.71%
1.6KM	16CH	gru	Batch16	80	344	82.74%	68.11%	NaN	71.50%	70.90%	82.74%
1.6KM	16CH	gru	Batch16	50	199	76.43%	72.84%	50.45%	78.05%	73.43%	76.43%
1.6KM	16CH	gru	Batch32	80	257	81.25%	76.56%	59.30%	84.54%	75.60%	81.25%
1.6KM	16CH	gru	Batch32	50	154	81.43%	79.55%	60.92%	82.22%	78.82%	81.43%
1.6KM	16CH	gru	Batch64	80	174	81.85%	77.17%	57.81%	82.26%	76.28%	81.85%
1.6KM	16CH	gru	Batch64	50	121	81.43%	79.35%	61.84%	83.78%	78.35%	81.43%
1.6KM	16CH	gru	Batch128	80	143	80.06%	75.36%	56.02%	82.71%	74.14%	80.06%
1.6KM	16CH	gru	Batch128	50	102	69.29%	65.52%	33.42%	68.19%	65.35%	69.29%

Figure 52: Table filter by GRU model (Speed = 1.6km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
1.6KM	Cz	cnn-lstm	Batch16	80	396	79.17%	71.01%	NaN	71.88%	78.00%	79.17%
1.6KM	Cz	cnn-lstm	Batch16	50	210	85.00%	82.10%	71.40%	90.33%	82.38%	85.00%
1.6KM	Cz	cnn-lstm	Batch32	80	253	78.57%	77.13%	56.87%	78.04%	78.93%	78.57%
1.6KM	Cz	cnn-lstm	Batch32	50	152	78.57%	72.95%	57.32%	87.04%	72.62%	78.57%
1.6KM	Cz	cnn-lstm	Batch64	80	227	80.36%	76.85%	54.98%	79.12%	75.96%	80.36%
1.6KM	Cz	cnn-lstm	Batch64	50	126	71.43%	62.07%	43.30%	83.86%	64.01%	71.43%
1.6KM	Cz	cnn-lstm	Batch128	80	171	81.85%	78.23%	60.10%	83.81%	76.77%	81.85%
1.6KM	Cz	cnn-lstm	Batch128	50	113	70.00%	59.81%	40.56%	83.31%	62.40%	70.00%
1.6KM	2Ch	cnn-lstm	Batch16	80	392	63.69%	59.68%	38.57%	70.56%	69.43%	63.69%
1.6KM	2Ch	cnn-lstm	Batch16	50	230	76.43%	75.60%	55.02%	77.01%	78.14%	76.43%
1.6KM	2Ch	cnn-lstm	Batch32	80	254	68.15%	67.31%	45.48%	72.32%	73.34%	68.15%
1.6KM	2Ch	cnn-lstm	Batch32	50	151	72.86%	72.09%	45.37%	72.56%	72.83%	72.86%
1.6KM	2Ch	cnn-lstm	Batch64	80	189	74.40%	73.44%	49.59%	74.35%	75.28%	74.40%
1.6KM	2Ch	cnn-lstm	Batch64	50	125	75.71%	74.03%	48.47%	74.74%	73.74%	75.71%
1.6KM	2Ch	cnn-lstm	Batch128	80	154	77.68%	76.46%	53.27%	76.15%	77.13%	77.68%
1.6KM	2Ch	cnn-lstm	Batch128	50	111	80.71%	79.51%	59.11%	79.86%	79.25%	80.71%
1.6KM	5Ch	cnn-lstm	Batch16	80	415	81.25%	74.59%	NaN	76.73%	78.43%	81.25%
1.6KM	5Ch	cnn-lstm	Batch16	50	214	75.71%	73.73%	50.21%	75.23%	75.05%	75.71%
1.6KM	5Ch	cnn-lstm	Batch32	80	263	78.57%	76.82%	56.47%	78.80%	77.86%	78.57%
1.6KM	5Ch	cnn-lstm	Batch32	50	170	78.57%	77.46%	56.08%	78.81%	77.33%	78.57%
1.6KM	5Ch	cnn-lstm	Batch64	80	189	80.65%	79.43%	59.22%	79.62%	79.61%	80.65%
1.6KM	5Ch	cnn-lstm	Batch64	50	126	82.14%	80.01%	63.39%	84.34%	79.40%	82.14%
1.6KM	5Ch	cnn-lstm	Batch128	80	164	85.71%	83.77%	68.56%	85.95%	82.71%	85.71%
1.6KM	5Ch	cnn-lstm	Batch128	50	120	78.57%	77.66%	56.36%	78.32%	78.09%	78.57%
1.6KM	16CH	cnn-lstm	Batch16	80	477	80.95%	74.50%	54.76%	77.53%	77.96%	80.95%
1.6KM	16CH	cnn-lstm	Batch16	50	271	76.43%	75.31%	55.83%	77.58%	78.33%	76.43%
1.6KM	16CH	cnn-lstm	Batch32	80	316	80.95%	80.20%	64.96%	81.10%	83.99%	80.95%
1.6KM	16CH	cnn-lstm	Batch32	50	191	77.14%	75.79%	52.97%	76.98%	76.07%	77.14%
1.6KM	16CH	cnn-lstm	Batch64	80	230	83.33%	82.50%	66.05%	82.06%	84.04%	83.33%
1.6KM	16CH	cnn-lstm	Batch64	50	141	74.29%	73.43%	48.16%	74.12%	74.05%	74.29%
1.6KM	16CH	cnn-lstm	Batch128	80	171	86.61%	84.99%	70.48%	86.33%	84.21%	86.61%
1.6KM	16CH	cnn-lstm	Batch128	50	136	74.29%	74.16%	51.44%	75.30%	76.15%	74.29%

Figure 53: Table filter by CNN-LSTM model (Speed = 1.6km/h).

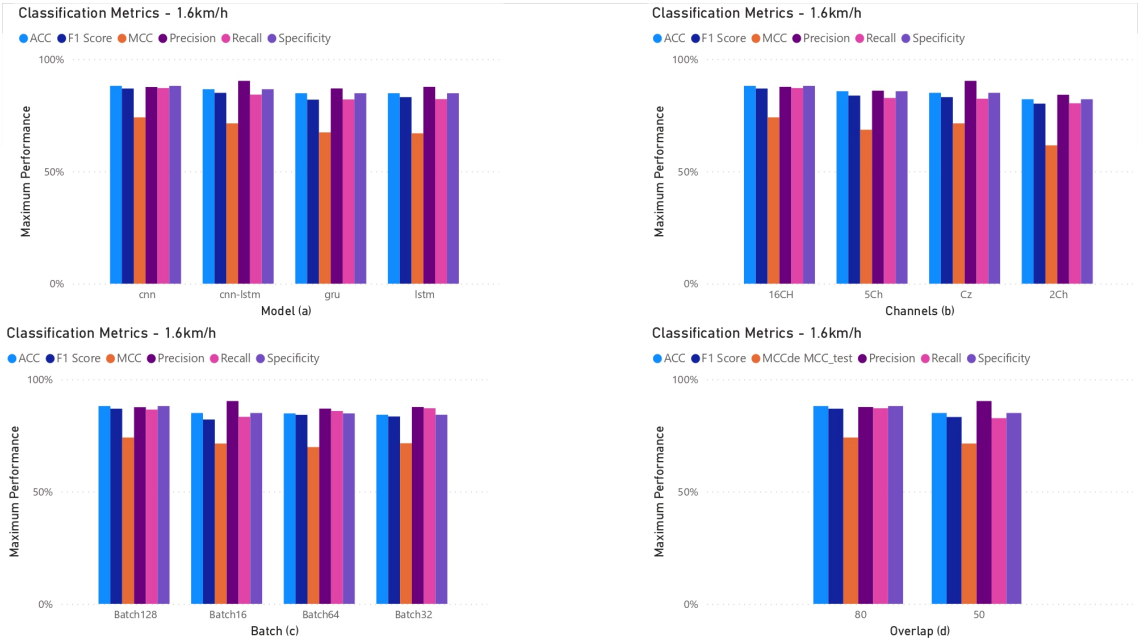


Figure 54: Classification Metrics Graphics sort of by (a) Models, (b) Channels, (c) Batch and (d) Overlap. Dataset filtered by speed = 1.6km/h.

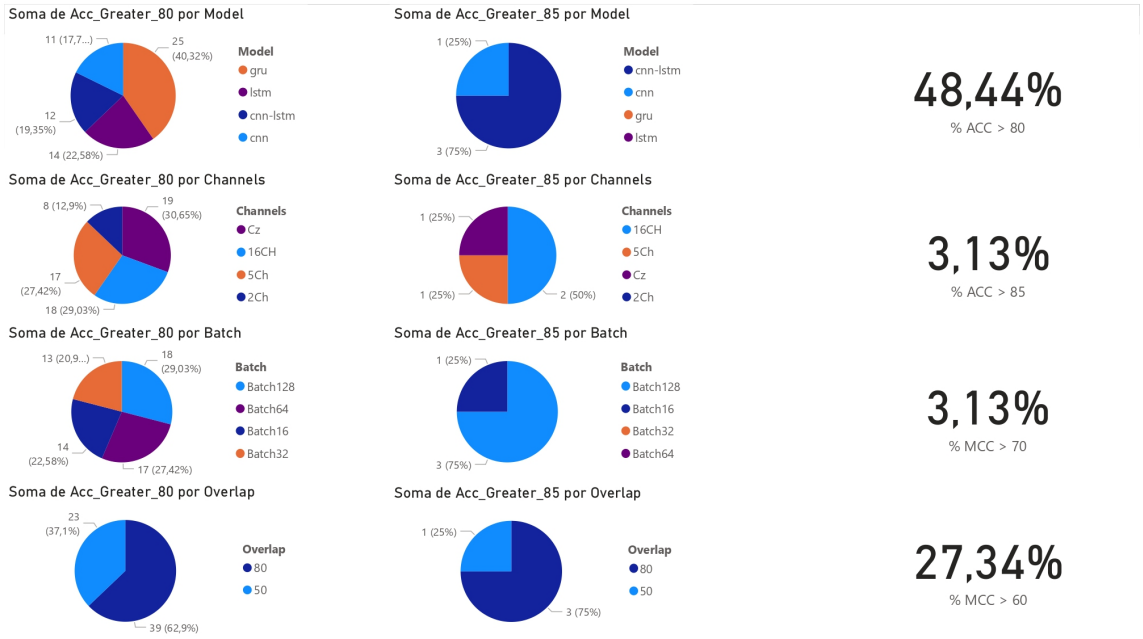


Figure 55: Accuracy greater 80% and 85% sort of by Model, Channels, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by speed = 1.6km/h.

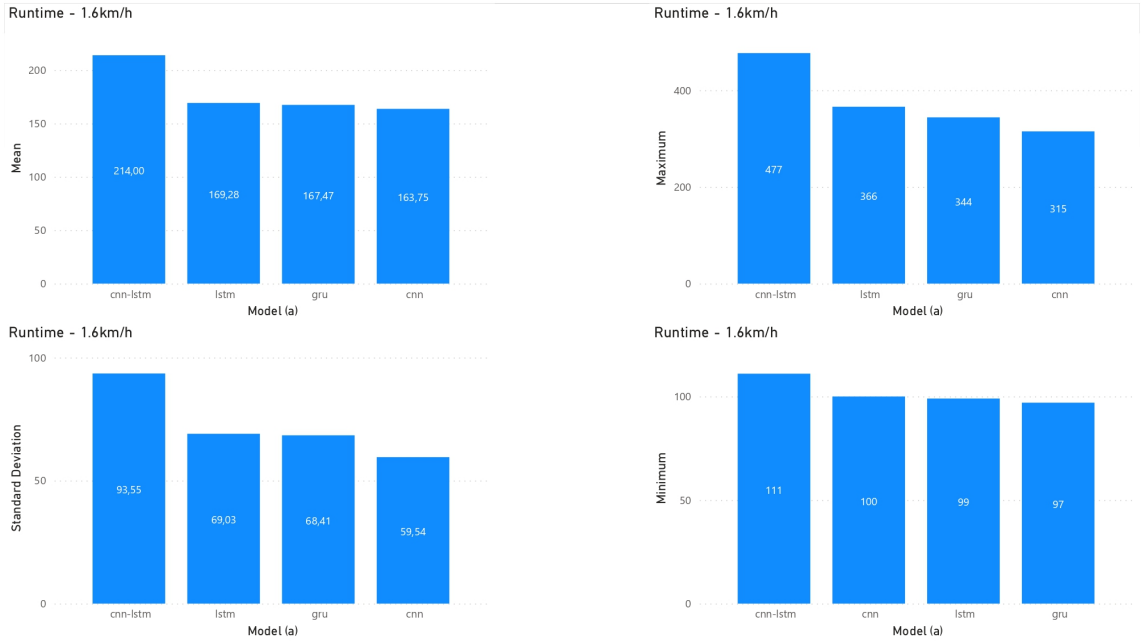


Figure 56: Runtime sort of by Model. Dataset filtered by speed = 1.6km/h.

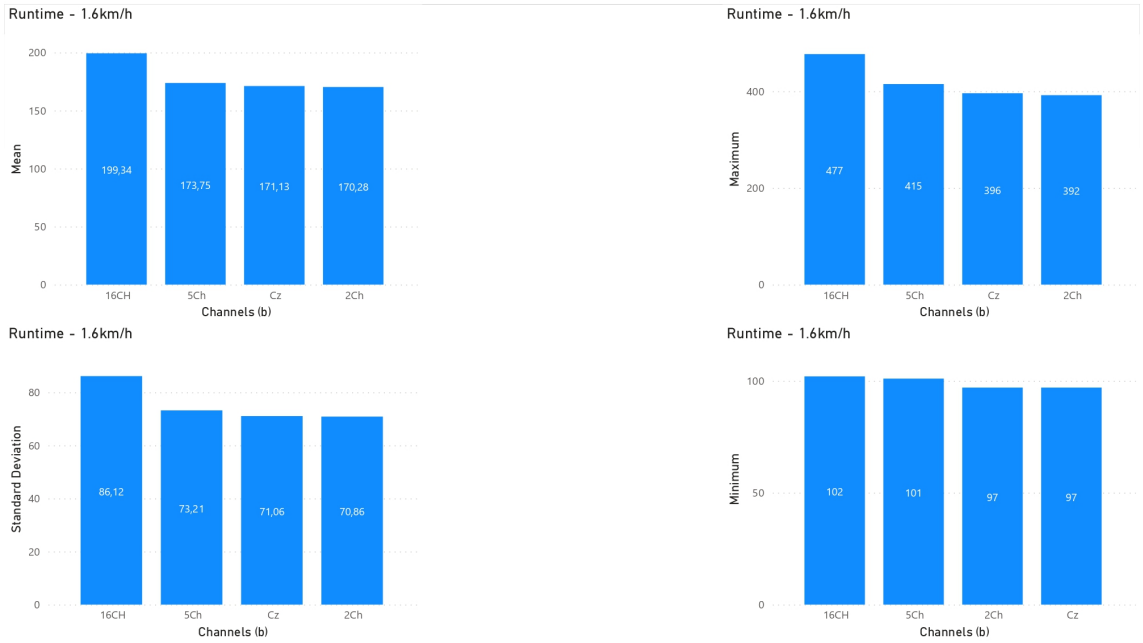


Figure 57: Runtime sort of by Channels. Dataset filtered by speed = 1.6km/h.

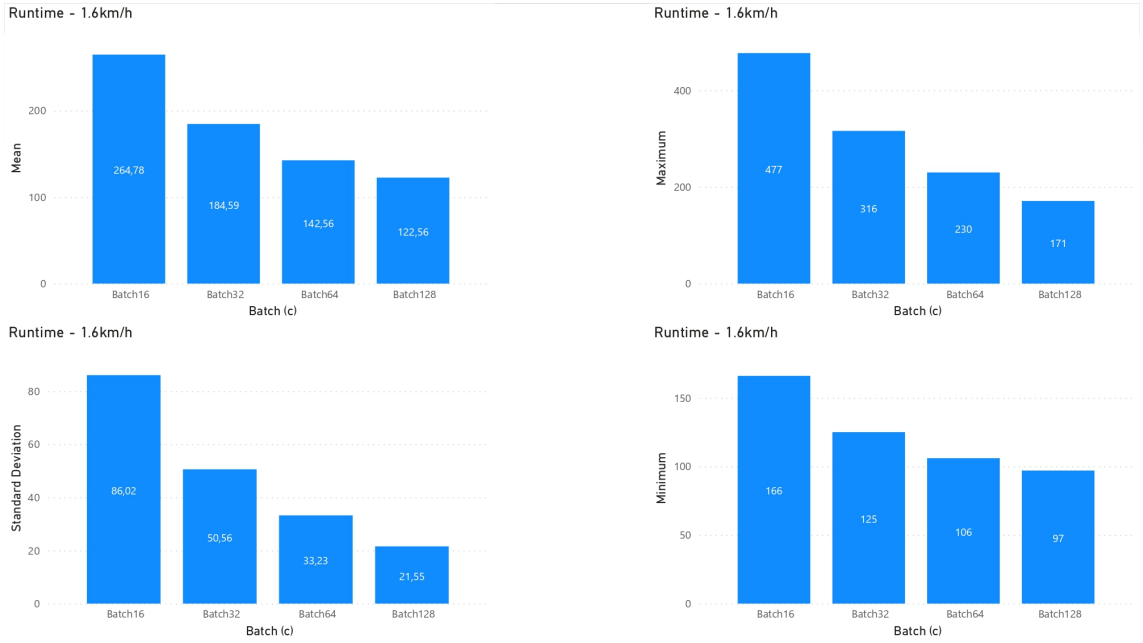


Figure 58: Runtime sort of by Batch. Dataset filtered by speed = 1.6km/h.

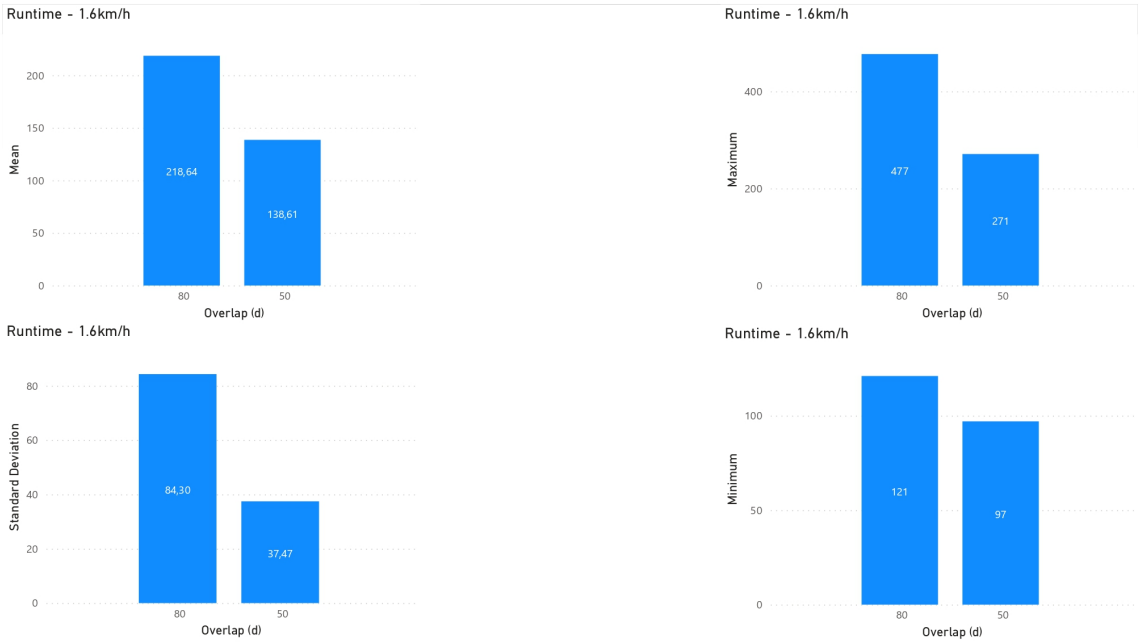


Figure 59: Runtime sort of by Overlap. Dataset filtered by speed = 1.6km/h.

B.3 OUTCOMES FILTERED BY SPEED = 2.5KM/H

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
2.5KM	Cz	cnn	Batch16	80	296	88.99%	81.43%	NaN	83.41%	81.95%	88.99%
2.5KM	Cz	cnn	Batch16	50	169	85.71%	82.62%	70.77%	89.64%	82.33%	85.71%
2.5KM	Cz	cnn	Batch32	80	198	88.69%	86.39%	74.06%	88.23%	85.92%	88.69%
2.5KM	Cz	cnn	Batch32	50	130	85.71%	83.70%	70.58%	88.53%	82.50%	85.71%
2.5KM	Cz	cnn	Batch64	80	154	90.48%	89.75%	79.83%	90.56%	89.29%	90.48%
2.5KM	Cz	cnn	Batch64	50	111	85.00%	82.88%	68.33%	86.92%	81.66%	85.00%
2.5KM	Cz	cnn	Batch128	80	129	91.07%	90.41%	81.06%	90.34%	90.73%	91.07%
2.5KM	Cz	cnn	Batch128	50	97	82.86%	79.47%	66.23%	89.24%	78.06%	82.86%
2.5KM	2Ch	cnn	Batch16	80	297	81.85%	76.30%	NaN	80.17%	80.59%	81.85%
2.5KM	2Ch	cnn	Batch16	50	170	85.00%	83.34%	69.22%	86.22%	83.33%	85.00%
2.5KM	2Ch	cnn	Batch32	80	199	83.33%	81.47%	65.26%	82.84%	82.64%	83.33%
2.5KM	2Ch	cnn	Batch32	50	127	84.29%	82.94%	66.33%	83.68%	82.68%	84.29%
2.5KM	2Ch	cnn	Batch64	80	152	83.93%	82.59%	65.88%	82.55%	83.36%	83.93%
2.5KM	2Ch	cnn	Batch64	50	109	82.14%	80.04%	61.32%	82.14%	79.27%	82.14%
2.5KM	2Ch	cnn	Batch128	80	130	82.14%	80.16%	60.81%	81.43%	79.41%	82.14%
2.5KM	2Ch	cnn	Batch128	50	101	85.00%	83.79%	68.09%	85.03%	83.10%	85.00%
2.5KM	5Ch	cnn	Batch16	80	314	89.58%	80.26%	NaN	80.54%	83.33%	89.58%
2.5KM	5Ch	cnn	Batch16	50	170	90.71%	90.00%	81.71%	91.68%	90.14%	90.71%
2.5KM	5Ch	cnn	Batch32	80	195	87.80%	86.40%	73.41%	86.21%	87.24%	87.80%
2.5KM	5Ch	cnn	Batch32	50	128	92.86%	92.26%	85.48%	94.19%	91.37%	92.86%
2.5KM	5Ch	cnn	Batch64	80	151	89.88%	88.14%	78.49%	91.94%	86.83%	89.88%
2.5KM	5Ch	cnn	Batch64	50	109	91.43%	90.62%	82.85%	93.73%	89.25%	91.43%
2.5KM	5Ch	cnn	Batch128	80	132	90.48%	89.10%	79.45%	91.95%	87.65%	90.48%
2.5KM	5Ch	cnn	Batch128	50	101	91.43%	90.62%	82.84%	93.74%	89.23%	91.43%
2.5KM	16CH	cnn	Batch16	80	305	81.25%	73.88%	NaN	76.64%	76.23%	81.25%
2.5KM	16CH	cnn	Batch16	50	172	88.57%	86.67%	77.13%	91.55%	86.10%	88.57%
2.5KM	16CH	cnn	Batch32	80	214	83.33%	80.52%	62.69%	82.53%	80.33%	83.33%
2.5KM	16CH	cnn	Batch32	50	134	86.43%	84.82%	72.97%	89.70%	83.54%	86.43%
2.5KM	16CH	cnn	Batch64	80	162	84.82%	82.76%	68.96%	87.32%	81.95%	84.82%
2.5KM	16CH	cnn	Batch64	50	112	74.29%	66.90%	49.36%	85.04%	67.69%	74.29%
2.5KM	16CH	cnn	Batch128	80	131	85.71%	83.45%	69.76%	88.36%	81.73%	85.71%
2.5KM	16CH	cnn	Batch128	50	103	90.71%	89.88%	81.94%	93.38%	88.71%	90.71%

Figure 60: Table filter by CNN model (Speed = 2.5km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
2.5KM	Cz	lstm	Batch16	80	313	86.61%	81.88%	NaN	84.35%	82.13%	86.61%
2.5KM	Cz	lstm	Batch16	50	173	86.43%	84.49%	73.77%	90.31%	83.95%	86.43%
2.5KM	Cz	lstm	Batch32	80	208	88.10%	85.22%	72.36%	88.07%	84.59%	88.10%
2.5KM	Cz	lstm	Batch32	50	133	87.86%	86.27%	75.12%	90.47%	84.98%	87.86%
2.5KM	Cz	lstm	Batch64	80	154	87.80%	85.93%	73.53%	89.06%	84.76%	87.80%
2.5KM	Cz	lstm	Batch64	50	108	85.00%	82.84%	68.84%	87.65%	81.51%	85.00%
2.5KM	Cz	lstm	Batch128	80	126	89.29%	87.84%	76.85%	90.54%	86.44%	89.29%
2.5KM	Cz	lstm	Batch128	50	99	87.14%	85.90%	73.29%	88.68%	84.73%	87.14%
2.5KM	2Ch	lstm	Batch16	80	324	76.49%	69.59%	47.37%	75.04%	72.08%	76.49%
2.5KM	2Ch	lstm	Batch16	50	174	80.71%	76.85%	54.80%	78.05%	76.86%	80.71%
2.5KM	2Ch	lstm	Batch32	80	208	78.57%	76.51%	55.29%	77.77%	77.68%	78.57%
2.5KM	2Ch	lstm	Batch32	50	131	82.14%	80.72%	61.94%	81.25%	80.71%	82.14%
2.5KM	2Ch	lstm	Batch64	80	152	80.06%	78.67%	58.74%	79.24%	79.55%	80.06%
2.5KM	2Ch	lstm	Batch64	50	109	80.71%	79.61%	59.41%	79.96%	79.46%	80.71%
2.5KM	2Ch	lstm	Batch128	80	127	81.55%	80.34%	61.50%	81.03%	80.50%	81.55%
2.5KM	2Ch	lstm	Batch128	50	96	82.86%	81.68%	63.73%	82.53%	81.21%	82.86%
2.5KM	5Ch	lstm	Batch16	80	310	86.61%	77.56%	NaN	79.58%	78.49%	86.61%
2.5KM	5Ch	lstm	Batch16	50	174	88.57%	87.19%	76.60%	89.67%	87.29%	88.57%
2.5KM	5Ch	lstm	Batch32	80	207	90.18%	88.21%	78.25%	90.92%	87.70%	90.18%
2.5KM	5Ch	lstm	Batch32	50	131	88.57%	87.48%	76.32%	89.80%	86.67%	88.57%
2.5KM	5Ch	lstm	Batch64	80	150	89.88%	88.88%	78.82%	90.51%	88.41%	89.88%
2.5KM	5Ch	lstm	Batch64	50	109	88.57%	87.87%	76.07%	88.63%	87.46%	88.57%
2.5KM	5Ch	lstm	Batch128	80	127	91.07%	90.16%	80.83%	91.62%	89.25%	91.07%
2.5KM	5Ch	lstm	Batch128	50	99	90.00%	89.24%	79.17%	90.82%	88.38%	90.00%
2.5KM	16CH	lstm	Batch16	80	312	83.63%	75.26%	NaN	77.77%	79.12%	83.63%
2.5KM	16CH	lstm	Batch16	50	178	80.00%	77.83%	58.22%	80.60%	78.00%	80.00%
2.5KM	16CH	lstm	Batch32	80	232	89.58%	88.62%	79.70%	90.39%	89.39%	89.58%
2.5KM	16CH	lstm	Batch32	50	139	90.00%	89.00%	79.50%	91.78%	87.88%	90.00%
2.5KM	16CH	lstm	Batch64	80	162	85.12%	83.67%	67.85%	83.88%	83.99%	85.12%
2.5KM	16CH	lstm	Batch64	50	113	77.14%	74.34%	50.14%	76.27%	73.96%	77.14%
2.5KM	16CH	lstm	Batch128	80	130	89.29%	88.22%	76.82%	89.49%	87.37%	89.29%
2.5KM	16CH	lstm	Batch128	50	102	82.86%	82.34%	65.01%	82.04%	82.98%	82.86%

Figure 61: Table filter by LSTM model (Speed = 2.5km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
2.5KM	Cz	gru	Batch16	80	308	87.50%	76.11%	NaN	78.69%	78.66%	87.50%
2.5KM	Cz	gru	Batch16	50	169	87.14%	85.01%	74.80%	90.98%	84.43%	87.14%
2.5KM	Cz	gru	Batch32	80	204	88.10%	85.47%	73.34%	89.50%	84.29%	88.10%
2.5KM	Cz	gru	Batch32	50	129	88.57%	87.04%	76.44%	90.96%	85.80%	88.57%
2.5KM	Cz	gru	Batch64	80	149	87.20%	83.94%	71.88%	90.28%	82.68%	87.20%
2.5KM	Cz	gru	Batch64	50	109	85.00%	82.68%	69.48%	88.75%	81.21%	85.00%
2.5KM	Cz	gru	Batch128	80	125	86.01%	82.98%	70.14%	89.61%	81.20%	86.01%
2.5KM	Cz	gru	Batch128	50	99	85.71%	83.80%	71.20%	89.34%	82.24%	85.71%
2.5KM	2Ch	gru	Batch16	80	308	78.87%	71.19%	NaN	75.65%	75.50%	78.87%
2.5KM	2Ch	gru	Batch16	50	172	80.00%	77.68%	57.46%	79.72%	77.86%	80.00%
2.5KM	2Ch	gru	Batch32	80	205	81.25%	79.26%	61.17%	81.16%	80.23%	81.25%
2.5KM	2Ch	gru	Batch32	50	130	80.71%	78.81%	59.08%	80.76%	78.48%	80.71%
2.5KM	2Ch	gru	Batch64	80	149	83.33%	81.65%	64.64%	82.98%	81.75%	83.33%
2.5KM	2Ch	gru	Batch64	50	109	84.29%	83.24%	66.83%	83.99%	82.86%	84.29%
2.5KM	2Ch	gru	Batch128	80	124	80.06%	78.01%	56.38%	78.81%	77.59%	80.06%
2.5KM	2Ch	gru	Batch128	50	97	84.29%	83.35%	66.90%	83.86%	83.04%	84.29%
2.5KM	5Ch	gru	Batch16	80	308	89.29%	77.29%	NaN	78.50%	77.90%	89.29%
2.5KM	5Ch	gru	Batch16	50	171	88.57%	86.72%	75.54%	89.09%	86.57%	88.57%
2.5KM	5Ch	gru	Batch32	80	205	91.07%	89.48%	80.84%	92.17%	88.93%	91.07%
2.5KM	5Ch	gru	Batch32	50	132	92.14%	91.42%	84.05%	93.74%	90.41%	92.14%
2.5KM	5Ch	gru	Batch64	80	149	88.69%	86.80%	75.69%	90.60%	85.37%	88.69%
2.5KM	5Ch	gru	Batch64	50	109	90.00%	89.34%	79.08%	90.36%	88.73%	90.00%
2.5KM	5Ch	gru	Batch128	80	125	90.77%	89.63%	79.97%	91.58%	88.46%	90.77%
2.5KM	5Ch	gru	Batch128	50	99	90.71%	89.88%	81.08%	92.53%	88.64%	90.71%
2.5KM	16CH	gru	Batch16	80	311	84.23%	73.31%	NaN	74.54%	75.98%	84.23%
2.5KM	16CH	gru	Batch16	50	174	80.00%	77.98%	59.29%	80.82%	78.95%	80.00%
2.5KM	16CH	gru	Batch32	80	238	84.82%	81.43%	64.72%	83.40%	81.51%	84.82%
2.5KM	16CH	gru	Batch32	50	138	83.57%	82.83%	66.08%	82.77%	83.31%	83.57%
2.5KM	16CH	gru	Batch64	80	160	83.04%	80.15%	63.06%	83.31%	80.05%	83.04%
2.5KM	16CH	gru	Batch64	50	111	83.57%	83.01%	66.53%	82.81%	83.73%	83.57%
2.5KM	16CH	gru	Batch128	80	125	83.93%	83.25%	67.56%	82.95%	84.64%	83.93%
2.5KM	16CH	gru	Batch128	50	99	87.86%	87.20%	74.41%	87.33%	87.08%	87.86%

Figure 62: Table filter by GRU model (Speed = 2.5km/h).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
2.5KM	Cz	cnn-lstm	Batch16	80	439	90.48%	76.65%	NaN	77.81%	78.97%	90.48%
2.5KM	Cz	cnn-lstm	Batch16	50	216	80.00%	73.92%	NaN	80.15%	74.81%	80.00%
2.5KM	Cz	cnn-lstm	Batch32	80	298	88.39%	86.18%	74.42%	89.53%	85.18%	88.39%
2.5KM	Cz	cnn-lstm	Batch32	50	156	83.57%	80.72%	65.20%	85.81%	79.88%	83.57%
2.5KM	Cz	cnn-lstm	Batch64	80	194	88.10%	86.23%	74.64%	90.48%	84.52%	88.10%
2.5KM	Cz	cnn-lstm	Batch64	50	128	79.29%	74.70%	56.42%	83.28%	74.11%	79.29%
2.5KM	Cz	cnn-lstm	Batch128	80	173	86.90%	85.21%	71.15%	87.01%	84.20%	86.90%
2.5KM	Cz	cnn-lstm	Batch128	50	113	79.29%	75.33%	57.62%	84.22%	74.34%	79.29%
2.5KM	2Ch	cnn-lstm	Batch16	80	406	84.23%	78.17%	NaN	80.89%	82.33%	84.23%
2.5KM	2Ch	cnn-lstm	Batch16	50	223	81.43%	78.45%	60.53%	83.20%	78.05%	81.43%
2.5KM	2Ch	cnn-lstm	Batch32	80	277	80.06%	78.82%	62.35%	80.80%	81.64%	80.06%
2.5KM	2Ch	cnn-lstm	Batch32	50	152	83.57%	82.27%	65.27%	83.28%	82.05%	83.57%
2.5KM	2Ch	cnn-lstm	Batch64	80	205	83.33%	82.40%	66.80%	82.81%	84.06%	83.33%
2.5KM	2Ch	cnn-lstm	Batch64	50	133	79.29%	77.22%	55.55%	79.06%	76.54%	79.29%
2.5KM	2Ch	cnn-lstm	Batch128	80	162	86.31%	85.62%	71.85%	85.33%	86.55%	86.31%
2.5KM	2Ch	cnn-lstm	Batch128	50	126	81.43%	80.15%	60.66%	80.97%	79.71%	81.43%
2.5KM	5Ch	cnn-lstm	Batch16	80	400	85.42%	78.41%	NaN	80.37%	83.43%	85.42%
2.5KM	5Ch	cnn-lstm	Batch16	50	210	88.57%	87.29%	75.93%	88.79%	87.19%	88.57%
2.5KM	5Ch	cnn-lstm	Batch32	80	258	87.50%	85.60%	71.95%	85.96%	86.09%	87.50%
2.5KM	5Ch	cnn-lstm	Batch32	50	152	88.57%	87.72%	76.06%	88.92%	87.19%	88.57%
2.5KM	5Ch	cnn-lstm	Batch64	80	196	86.90%	85.78%	72.08%	85.53%	86.57%	86.90%
2.5KM	5Ch	cnn-lstm	Batch64	50	125	89.29%	88.06%	78.65%	92.43%	86.47%	89.29%
2.5KM	5Ch	cnn-lstm	Batch128	80	162	86.90%	85.76%	71.60%	85.89%	85.71%	86.90%
2.5KM	5Ch	cnn-lstm	Batch128	50	129	88.57%	87.70%	76.07%	89.22%	86.88%	88.57%
2.5KM	16CH	cnn-lstm	Batch16	80	408	88.69%	84.17%	NaN	85.85%	86.87%	88.69%
2.5KM	16CH	cnn-lstm	Batch16	50	226	82.14%	79.75%	63.01%	83.56%	79.67%	82.14%
2.5KM	16CH	cnn-lstm	Batch32	80	288	82.44%	79.40%	62.18%	82.75%	79.72%	82.44%
2.5KM	16CH	cnn-lstm	Batch32	50	173	86.43%	85.33%	71.77%	87.28%	84.57%	86.43%
2.5KM	16CH	cnn-lstm	Batch64	80	215	86.31%	85.23%	71.26%	85.63%	85.66%	86.31%
2.5KM	16CH	cnn-lstm	Batch64	50	157	87.14%	86.72%	73.81%	86.47%	87.36%	87.14%
2.5KM	16CH	cnn-lstm	Batch128	80	161	86.01%	84.57%	70.03%	86.28%	83.81%	86.01%
2.5KM	16CH	cnn-lstm	Batch128	50	130	79.29%	75.12%	59.81%	87.14%	74.09%	79.29%

Figure 63: Table filter by CNN-LSTM model (Speed = 2.5km/h).

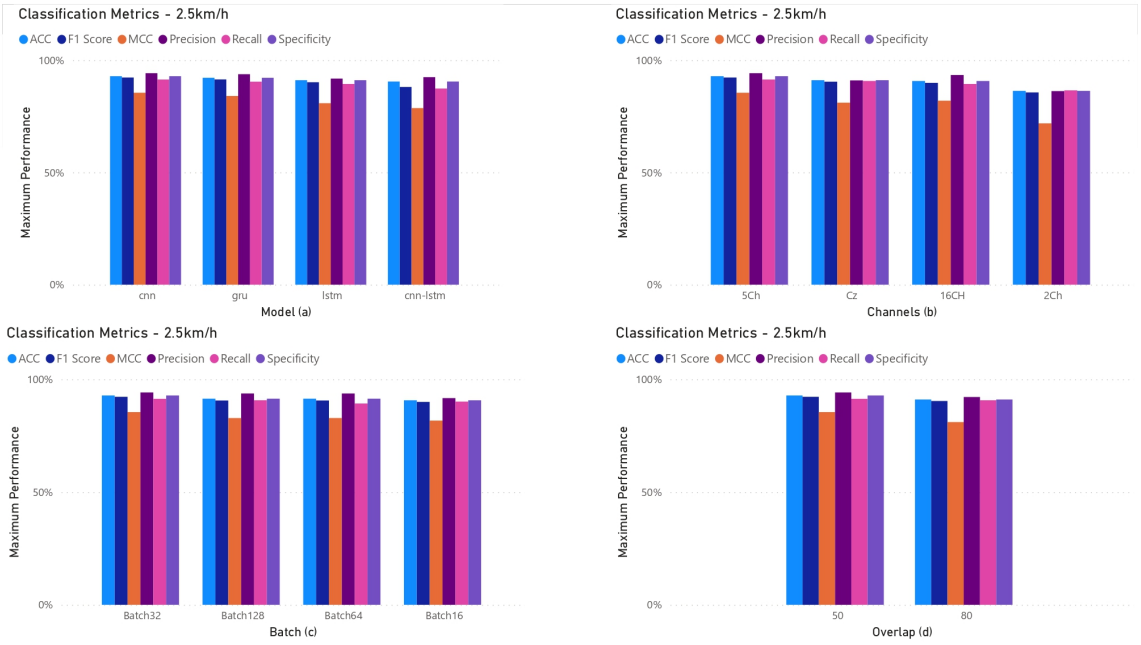


Figure 64: Classification Metrics Graphics sort of by (a) Models, (b) Channels, (c) Batch and (d) Overlap. Dataset filtered by speed = 2.5km/h.

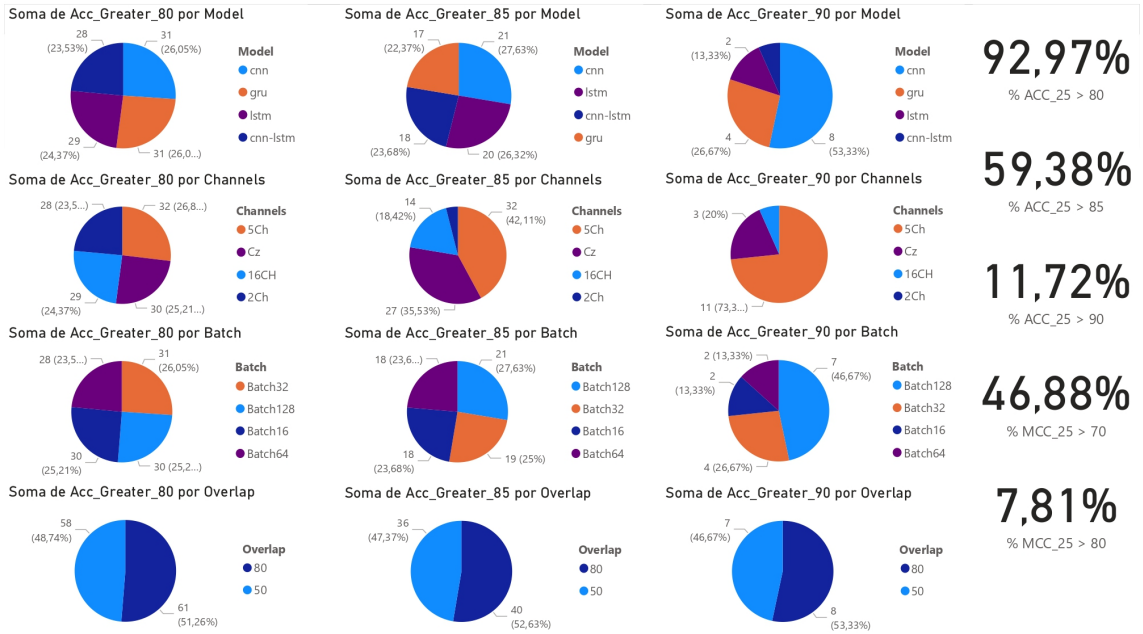


Figure 65: Accuracy greater 80% and 85% sort of by Model, Channels, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by speed = 2.5km/h.

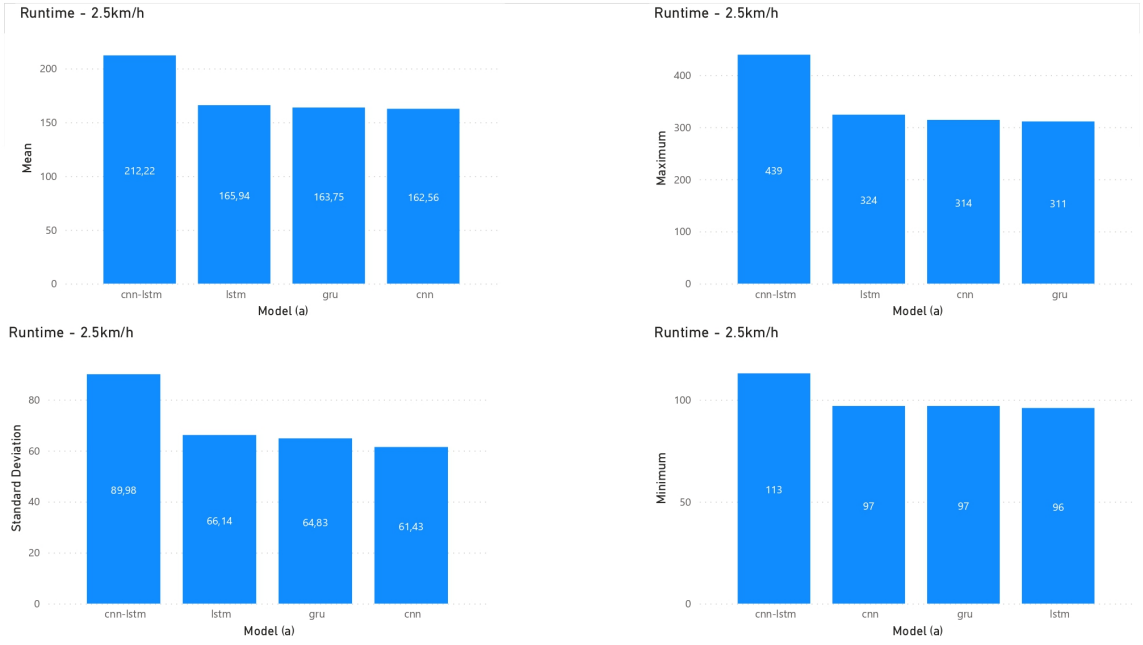


Figure 66: Runtime sort of by Model. Dataset filtered by speed = 2.5km/h.

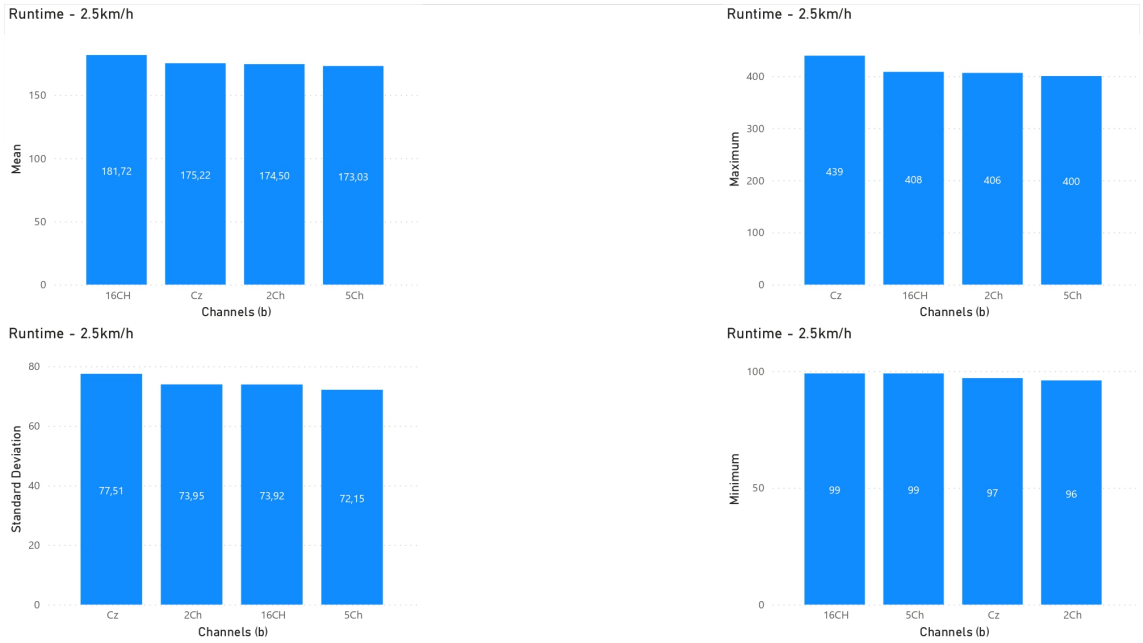


Figure 67: Runtime sort of by Channels. Dataset filtered by speed = 2.5km/h.

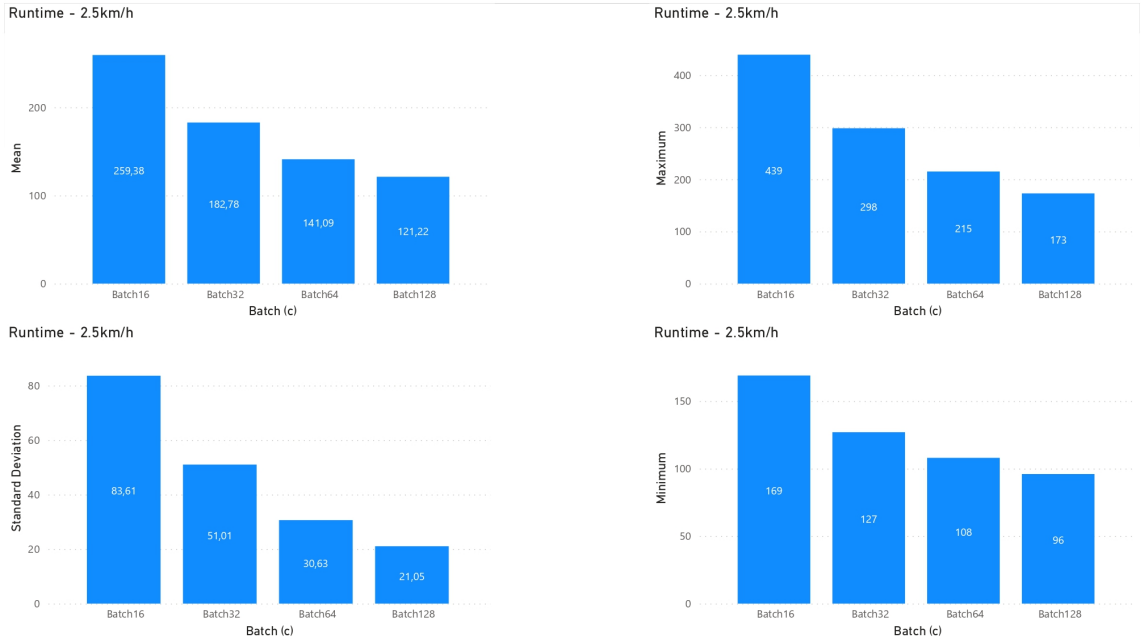


Figure 68: Runtime sort of by Batch. Dataset filtered by speed = 2.5km/h.

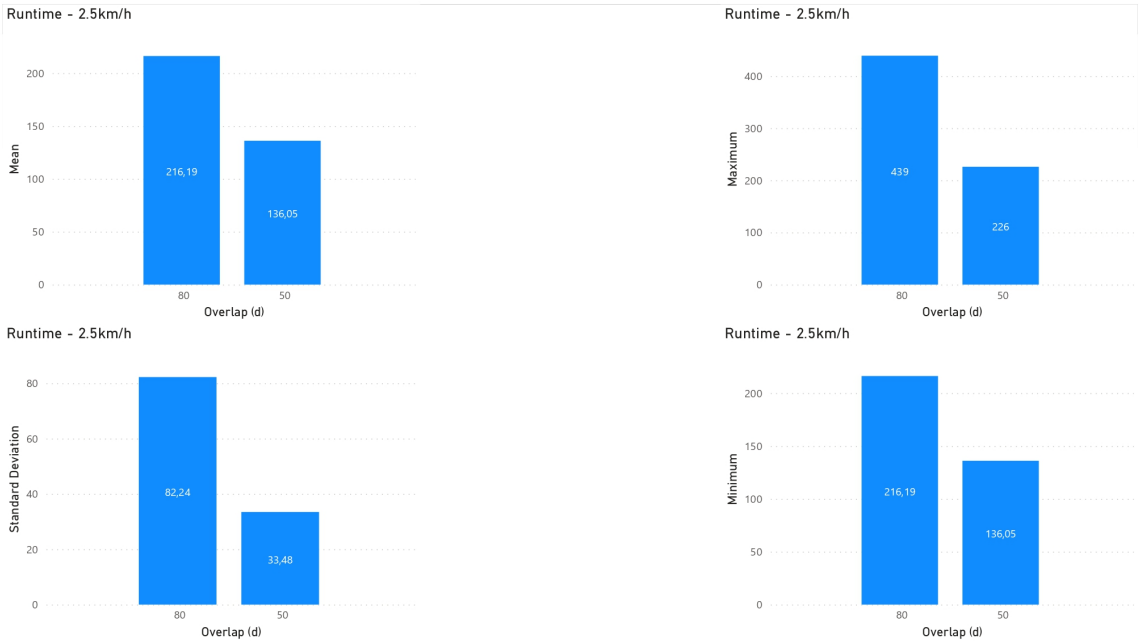


Figure 69: Runtime sort of by Overlap. Dataset filtered by speed = 2.5km/h.

B.4 OUTCOMES FILTERED BY SUBJECT 1

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s1	Cz	cnn	Batch16	80	292	85.94%	76.77%	NaN	76.93%	80.19%	85.94%
s1	Cz	cnn	Batch16	50	159	82.50%	79.95%	62.21%	82.67%	79.67%	82.50%
s1	Cz	cnn	Batch32	80	188	88.02%	86.37%	73.96%	86.88%	87.13%	88.02%
s1	Cz	cnn	Batch32	50	120	83.75%	82.72%	66.10%	83.61%	82.50%	83.75%
s1	Cz	cnn	Batch64	80	143	86.46%	85.26%	71.75%	87.60%	84.24%	86.46%
s1	Cz	cnn	Batch64	50	102	86.25%	85.12%	71.17%	86.70%	84.50%	86.25%
s1	Cz	cnn	Batch128	80	126	85.94%	84.52%	70.02%	86.88%	83.24%	85.94%
s1	Cz	cnn	Batch128	50	94	83.75%	82.78%	65.82%	83.54%	82.29%	83.75%
s1	2Ch	cnn	Batch16	80	285	85.94%	79.16%	61.74%	79.25%	83.69%	85.94%
s1	2Ch	cnn	Batch16	50	159	81.25%	80.25%	62.19%	81.06%	81.25%	81.25%
s1	2Ch	cnn	Batch32	80	184	89.06%	87.80%	76.10%	87.71%	88.41%	89.06%
s1	2Ch	cnn	Batch32	50	117	81.25%	80.64%	61.80%	80.64%	81.17%	81.25%
s1	2Ch	cnn	Batch64	80	140	84.90%	84.32%	69.61%	83.69%	85.98%	84.90%
s1	2Ch	cnn	Batch64	50	100	80.00%	78.83%	57.67%	78.83%	78.83%	80.00%
s1	2Ch	cnn	Batch128	80	119	86.46%	85.71%	71.45%	85.83%	85.62%	86.46%
s1	2Ch	cnn	Batch128	50	92	82.50%	81.77%	63.54%	81.77%	81.77%	82.50%
s1	5Ch	cnn	Batch16	80	285	80.21%	73.48%	56.87%	78.35%	80.68%	80.21%
s1	5Ch	cnn	Batch16	50	160	88.75%	88.08%	76.97%	88.36%	88.67%	88.75%
s1	5Ch	cnn	Batch32	80	184	91.67%	90.62%	82.17%	90.46%	91.80%	91.67%
s1	5Ch	cnn	Batch32	50	117	91.25%	90.75%	81.64%	90.89%	90.75%	91.25%
s1	5Ch	cnn	Batch64	80	141	89.58%	89.03%	79.24%	88.51%	90.80%	89.58%
s1	5Ch	cnn	Batch64	50	100	93.75%	93.20%	87.27%	95.25%	92.08%	93.75%
s1	5Ch	cnn	Batch128	80	122	92.19%	91.50%	83.27%	92.21%	91.08%	92.19%
s1	5Ch	cnn	Batch128	50	94	90.00%	89.33%	79.17%	90.66%	88.54%	90.00%
s1	16CH	cnn	Batch16	80	285	95.31%	94.25%	89.52%	94.15%	95.65%	95.31%
s1	16CH	cnn	Batch16	50	160	92.50%	91.95%	84.95%	93.81%	91.25%	92.50%
s1	16CH	cnn	Batch32	80	198	91.67%	90.86%	82.57%	91.31%	91.28%	91.67%
s1	16CH	cnn	Batch32	50	121	95.00%	94.39%	89.68%	96.44%	93.33%	95.00%
s1	16CH	cnn	Batch64	80	149	91.67%	91.14%	82.93%	90.44%	92.54%	91.67%
s1	16CH	cnn	Batch64	50	102	88.75%	87.39%	75.81%	89.57%	86.33%	88.75%
s1	16CH	cnn	Batch128	80	119	91.15%	90.11%	80.97%	92.17%	88.87%	91.15%
s1	16CH	cnn	Batch128	50	92	86.25%	84.48%	73.07%	90.68%	82.81%	86.25%

Figure 70: Table filter by CNN model (Subject 1).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s1	Cz	lstm	Batch16	80	305	83.33%	72.56%	NaN	73.22%	73.12%	83.33%
s1	Cz	lstm	Batch16	50	168	80.00%	79.33%	61.35%	80.81%	80.67%	80.00%
s1	Cz	lstm	Batch32	80	194	84.38%	82.24%	65.92%	83.54%	82.48%	84.38%
s1	Cz	lstm	Batch32	50	121	81.25%	80.31%	61.25%	80.76%	80.50%	81.25%
s1	Cz	lstm	Batch64	80	144	86.98%	86.07%	72.83%	86.47%	86.39%	86.98%
s1	Cz	lstm	Batch64	50	102	80.00%	78.92%	59.13%	79.92%	79.25%	80.00%
s1	Cz	lstm	Batch128	80	120	85.94%	84.84%	69.96%	85.64%	84.34%	85.94%
s1	Cz	lstm	Batch128	50	92	80.00%	78.67%	57.77%	79.67%	78.13%	80.00%
s1	2Ch	lstm	Batch16	80	297	86.98%	73.63%	NaN	74.28%	73.93%	86.98%
s1	2Ch	lstm	Batch16	50	164	77.50%	75.33%	52.27%	76.98%	75.42%	77.50%
s1	2Ch	lstm	Batch32	80	192	86.46%	83.48%	67.93%	84.71%	83.29%	86.46%
s1	2Ch	lstm	Batch32	50	124	77.50%	75.50%	51.79%	76.30%	75.50%	77.50%
s1	2Ch	lstm	Batch64	80	141	83.33%	82.30%	64.83%	82.76%	82.08%	83.33%
s1	2Ch	lstm	Batch64	50	99	77.50%	75.30%	51.70%	76.91%	74.83%	77.50%
s1	2Ch	lstm	Batch128	80	115	82.81%	81.27%	62.96%	82.42%	80.57%	82.81%
s1	2Ch	lstm	Batch128	50	90	76.25%	74.49%	49.64%	75.72%	73.96%	76.25%
s1	5Ch	lstm	Batch16	80	294	85.94%	72.27%	NaN	75.02%	72.90%	85.94%
s1	5Ch	lstm	Batch16	50	164	85.00%	83.56%	69.63%	85.88%	83.83%	85.00%
s1	5Ch	lstm	Batch32	80	195	85.42%	82.73%	68.02%	85.80%	82.53%	85.42%
s1	5Ch	lstm	Batch32	50	121	81.25%	79.21%	61.40%	83.39%	78.25%	81.25%
s1	5Ch	lstm	Batch64	80	142	87.50%	85.71%	73.70%	89.43%	84.46%	87.50%
s1	5Ch	lstm	Batch64	50	97	81.25%	80.51%	61.89%	80.74%	81.17%	81.25%
s1	5Ch	lstm	Batch128	80	118	88.54%	87.49%	75.20%	88.28%	86.94%	88.54%
s1	5Ch	lstm	Batch128	50	92	85.00%	84.00%	68.47%	85.16%	83.33%	85.00%
s1	16CH	lstm	Batch16	80	302	85.42%	73.34%	NaN	75.89%	74.45%	85.42%
s1	16CH	lstm	Batch16	50	166	86.25%	84.89%	72.34%	88.11%	84.42%	86.25%
s1	16CH	lstm	Batch32	80	216	85.94%	83.94%	69.52%	85.78%	83.79%	85.94%
s1	16CH	lstm	Batch32	50	129	85.00%	83.71%	68.86%	86.13%	82.83%	85.00%
s1	16CH	lstm	Batch64	80	153	86.46%	85.31%	71.44%	87.33%	84.18%	86.46%
s1	16CH	lstm	Batch64	50	103	87.50%	86.41%	73.32%	87.59%	85.75%	87.50%
s1	16CH	lstm	Batch128	80	123	86.98%	85.34%	72.06%	88.31%	83.94%	86.98%
s1	16CH	lstm	Batch128	50	90	86.25%	85.43%	71.12%	86.24%	84.90%	86.25%

Figure 71: Table filter by LSTM model (Subject 1).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s1	Cz	gru	Batch16	80	303	85.42%	78.30%	57.42%	78.54%	78.80%	85.42%
s1	Cz	gru	Batch16	50	166	76.25%	74.85%	52.12%	76.95%	75.25%	76.25%
s1	Cz	gru	Batch32	80	198	84.90%	81.73%	64.69%	83.22%	81.62%	84.90%
s1	Cz	gru	Batch32	50	121	80.00%	78.59%	58.07%	79.94%	78.17%	80.00%
s1	Cz	gru	Batch64	80	140	84.38%	83.03%	66.77%	84.45%	82.36%	84.38%
s1	Cz	gru	Batch64	50	100	78.75%	77.45%	56.15%	78.61%	77.58%	78.75%
s1	Cz	gru	Batch128	80	121	83.85%	82.45%	65.39%	83.90%	81.54%	83.85%
s1	Cz	gru	Batch128	50	87	80.00%	78.67%	57.77%	79.67%	78.13%	80.00%
s1	2Ch	gru	Batch16	80	295	85.42%	74.50%	NaN	75.45%	75.40%	85.42%
s1	2Ch	gru	Batch16	50	164	77.50%	75.27%	53.18%	78.08%	75.42%	77.50%
s1	2Ch	gru	Batch32	80	190	85.94%	83.56%	68.20%	85.03%	83.22%	85.94%
s1	2Ch	gru	Batch32	50	120	76.25%	73.22%	47.96%	74.84%	73.17%	76.25%
s1	2Ch	gru	Batch64	80	140	88.02%	86.91%	74.84%	89.24%	85.69%	88.02%
s1	2Ch	gru	Batch64	50	100	77.50%	75.30%	51.70%	76.91%	74.83%	77.50%
s1	2Ch	gru	Batch128	80	112	84.90%	82.71%	67.67%	86.90%	81.06%	84.90%
s1	2Ch	gru	Batch128	50	90	77.50%	76.00%	52.43%	76.92%	75.52%	77.50%
s1	5Ch	gru	Batch16	80	296	86.46%	78.23%	NaN	81.96%	79.67%	86.46%
s1	5Ch	gru	Batch16	50	164	86.25%	84.38%	70.67%	86.76%	84.00%	86.25%
s1	5Ch	gru	Batch32	80	190	89.58%	87.52%	77.66%	91.28%	86.69%	89.58%
s1	5Ch	gru	Batch32	50	121	86.25%	85.25%	71.92%	87.06%	84.92%	86.25%
s1	5Ch	gru	Batch64	80	139	90.10%	88.95%	79.84%	92.47%	87.54%	90.10%
s1	5Ch	gru	Batch64	50	99	90.00%	88.94%	79.10%	91.45%	87.75%	90.00%
s1	5Ch	gru	Batch128	80	115	90.63%	89.44%	81.05%	93.48%	87.78%	90.63%
s1	5Ch	gru	Batch128	50	89	88.75%	87.53%	77.80%	92.11%	85.94%	88.75%
s1	16CH	gru	Batch16	80	297	88.54%	77.37%	NaN	77.99%	77.88%	88.54%
s1	16CH	gru	Batch16	50	164	86.25%	84.99%	71.84%	87.95%	84.08%	86.25%
s1	16CH	gru	Batch32	80	214	88.02%	86.85%	74.53%	87.46%	87.13%	88.02%
s1	16CH	gru	Batch32	50	127	85.00%	84.04%	69.42%	85.55%	83.92%	85.00%
s1	16CH	gru	Batch64	80	149	90.63%	90.13%	80.36%	90.71%	89.67%	90.63%
s1	16CH	gru	Batch64	50	104	85.00%	83.95%	68.09%	84.34%	83.75%	85.00%
s1	16CH	gru	Batch128	80	121	89.58%	88.76%	77.61%	89.24%	88.37%	89.58%
s1	16CH	gru	Batch128	50	90	87.50%	86.67%	73.82%	87.91%	85.94%	87.50%

Figure 72: Table filter by GRU model (Subject 1).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s1	Cz	cnn-lstm	Batch16	80	412	82.81%	68.12%	NaN	69.49%	69.00%	82.81%
s1	Cz	cnn-lstm	Batch16	50	205	81.25%	78.50%	61.05%	82.37%	78.83%	81.25%
s1	Cz	cnn-lstm	Batch32	80	258	85.42%	82.62%	67.33%	85.51%	81.94%	85.42%
s1	Cz	cnn-lstm	Batch32	50	145	81.25%	78.39%	60.62%	83.35%	77.58%	81.25%
s1	Cz	cnn-lstm	Batch64	80	181	85.42%	83.99%	69.15%	86.21%	83.02%	85.42%
s1	Cz	cnn-lstm	Batch64	50	122	81.25%	79.10%	60.86%	82.78%	78.25%	81.25%
s1	Cz	cnn-lstm	Batch128	80	148	84.90%	82.63%	68.45%	87.88%	80.92%	84.90%
s1	Cz	cnn-lstm	Batch128	50	105	75.00%	68.65%	51.45%	85.29%	68.75%	75.00%
s1	2Ch	cnn-lstm	Batch16	80	389	83.33%	71.01%	NaN	71.31%	72.40%	83.33%
s1	2Ch	cnn-lstm	Batch16	50	200	81.25%	80.62%	63.66%	81.69%	82.08%	81.25%
s1	2Ch	cnn-lstm	Batch32	80	246	84.90%	81.69%	64.81%	83.69%	81.26%	84.90%
s1	2Ch	cnn-lstm	Batch32	50	148	78.75%	77.63%	55.53%	77.95%	77.58%	78.75%
s1	2Ch	cnn-lstm	Batch64	80	180	84.38%	83.31%	67.21%	84.78%	82.48%	84.38%
s1	2Ch	cnn-lstm	Batch64	50	116	78.75%	77.29%	55.18%	78.02%	77.17%	78.75%
s1	2Ch	cnn-lstm	Batch128	80	144	84.38%	82.99%	66.62%	84.76%	81.94%	84.38%
s1	2Ch	cnn-lstm	Batch128	50	103	76.25%	73.20%	49.87%	77.76%	72.40%	76.25%
s1	5Ch	cnn-lstm	Batch16	80	387	86.98%	78.00%	61.03%	79.82%	82.53%	86.98%
s1	5Ch	cnn-lstm	Batch16	50	203	83.75%	83.17%	68.36%	83.21%	85.25%	83.75%
s1	5Ch	cnn-lstm	Batch32	80	244	90.10%	89.13%	79.55%	88.76%	90.87%	90.10%
s1	5Ch	cnn-lstm	Batch32	50	143	86.25%	84.48%	73.11%	90.61%	82.92%	86.25%
s1	5Ch	cnn-lstm	Batch64	80	179	90.63%	89.83%	80.71%	92.03%	88.76%	90.63%
s1	5Ch	cnn-lstm	Batch64	50	116	88.75%	87.61%	78.09%	92.06%	86.25%	88.75%
s1	5Ch	cnn-lstm	Batch128	80	147	89.06%	88.10%	76.72%	89.43%	87.32%	89.06%
s1	5Ch	cnn-lstm	Batch128	50	105	85.00%	83.23%	69.72%	88.24%	81.77%	85.00%
s1	16CH	cnn-lstm	Batch16	80	388	95.31%	90.87%	83.29%	90.85%	93.06%	95.31%
s1	16CH	cnn-lstm	Batch16	50	202	91.25%	90.67%	82.81%	92.92%	90.00%	91.25%
s1	16CH	cnn-lstm	Batch32	80	276	94.79%	94.12%	88.75%	95.00%	93.82%	94.79%
s1	16CH	cnn-lstm	Batch32	50	154	91.25%	90.75%	81.64%	90.89%	90.75%	91.25%
s1	16CH	cnn-lstm	Batch64	80	194	96.35%	96.03%	92.18%	96.54%	95.66%	96.35%
s1	16CH	cnn-lstm	Batch64	50	120	86.25%	84.18%	72.63%	90.65%	82.50%	86.25%
s1	16CH	cnn-lstm	Batch128	80	150	93.75%	93.31%	86.77%	93.52%	93.25%	93.75%
s1	16CH	cnn-lstm	Batch128	50	120	88.75%	87.53%	77.80%	92.11%	85.94%	88.75%

Figure 73: Table filter by CNN-LSTM model (Subject 1).

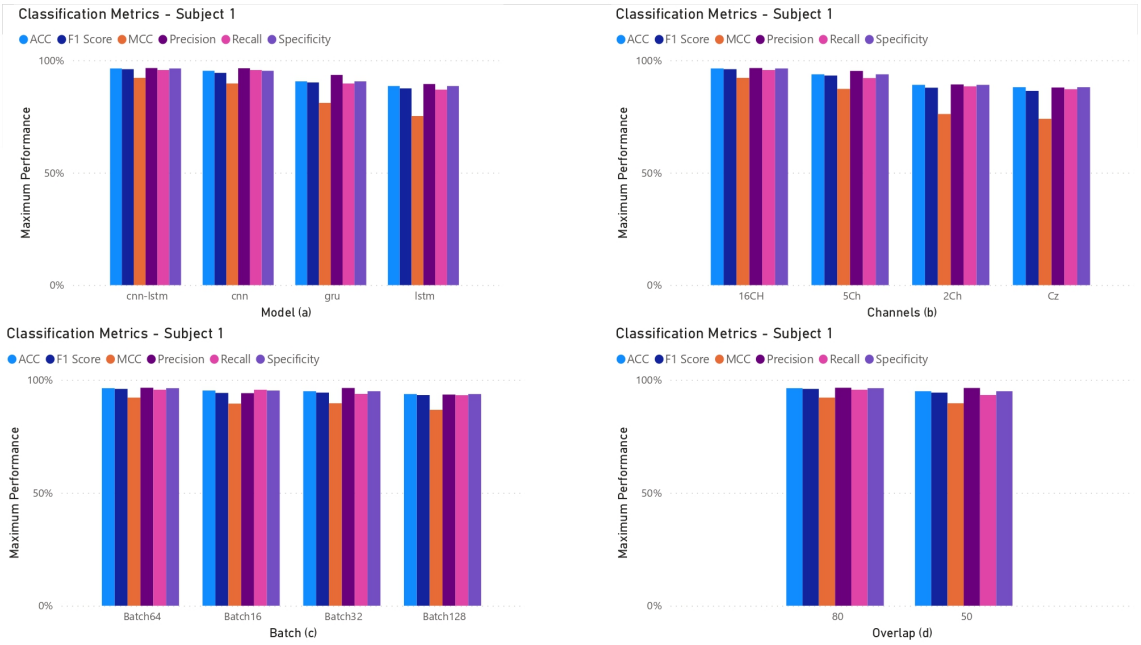


Figure 74: Classification Metrics Graphics sort of by (a) Models, (b) Channels, (c) Batch and (d) Overlap. Dataset filtered by Subject 1.

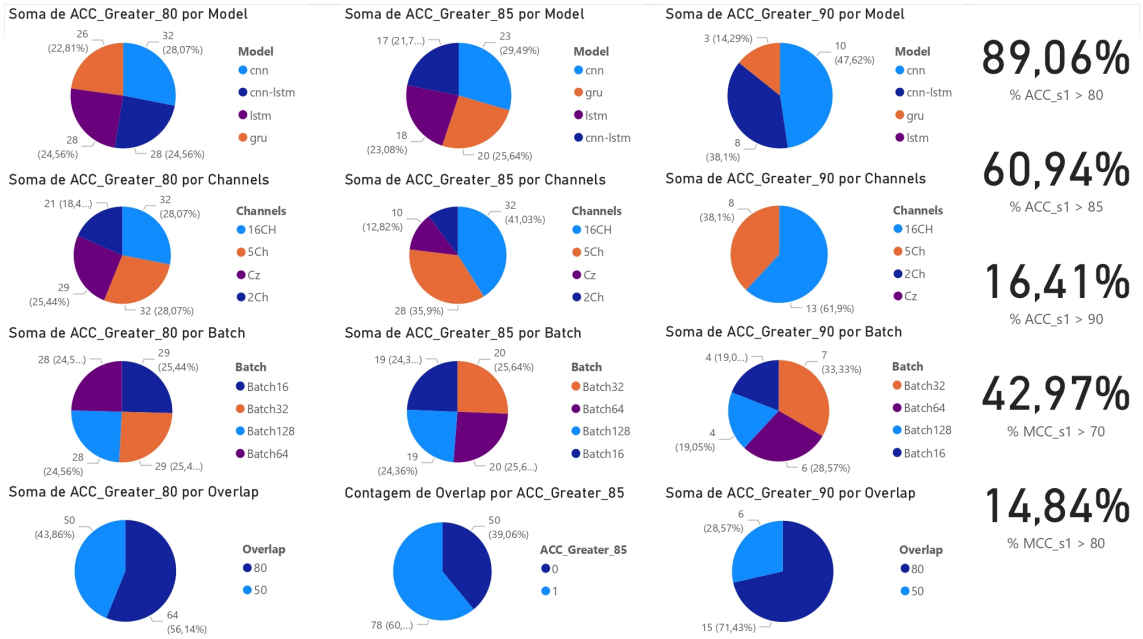


Figure 75: Accuracy greater 80% and 85% sort of by Model, Channels, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by subject 1.

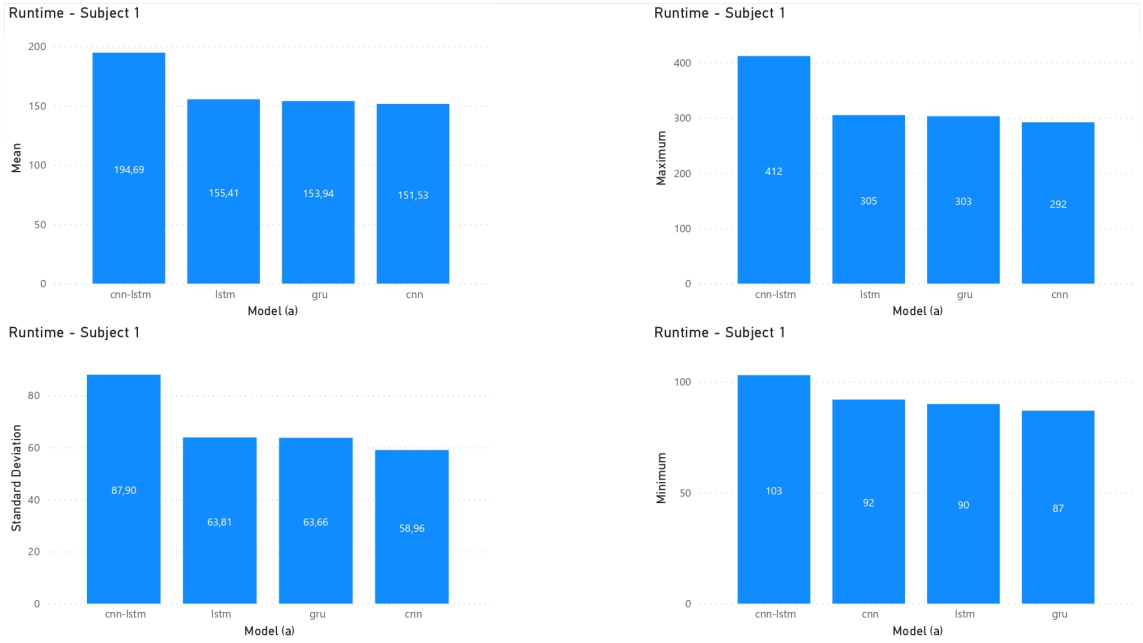


Figure 76: Runtime sort of by Model. Dataset filtered by Subject 1.

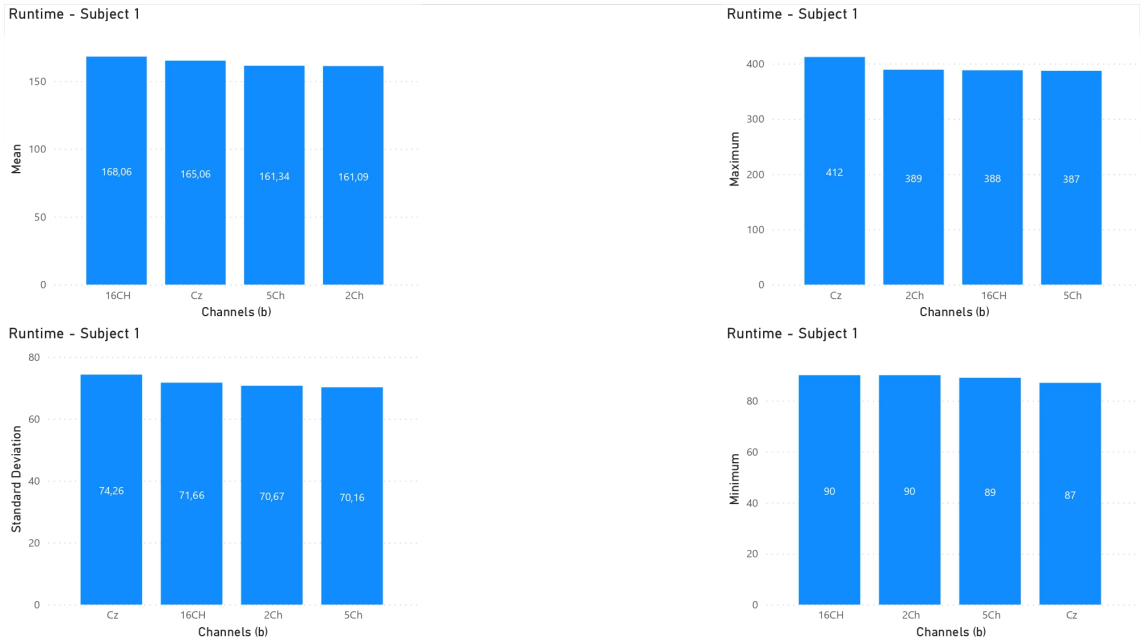


Figure 77: Runtime sort of by Channels. Dataset filtered by Subject 1.

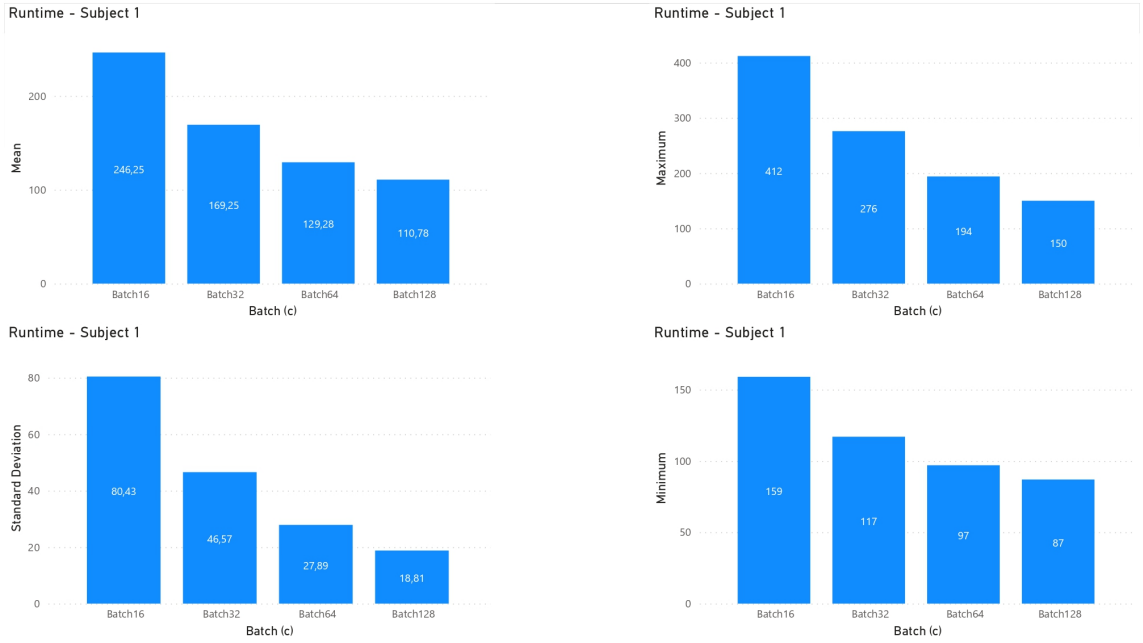


Figure 78: Runtime sort of by Batch. Dataset filtered by Subject 1.

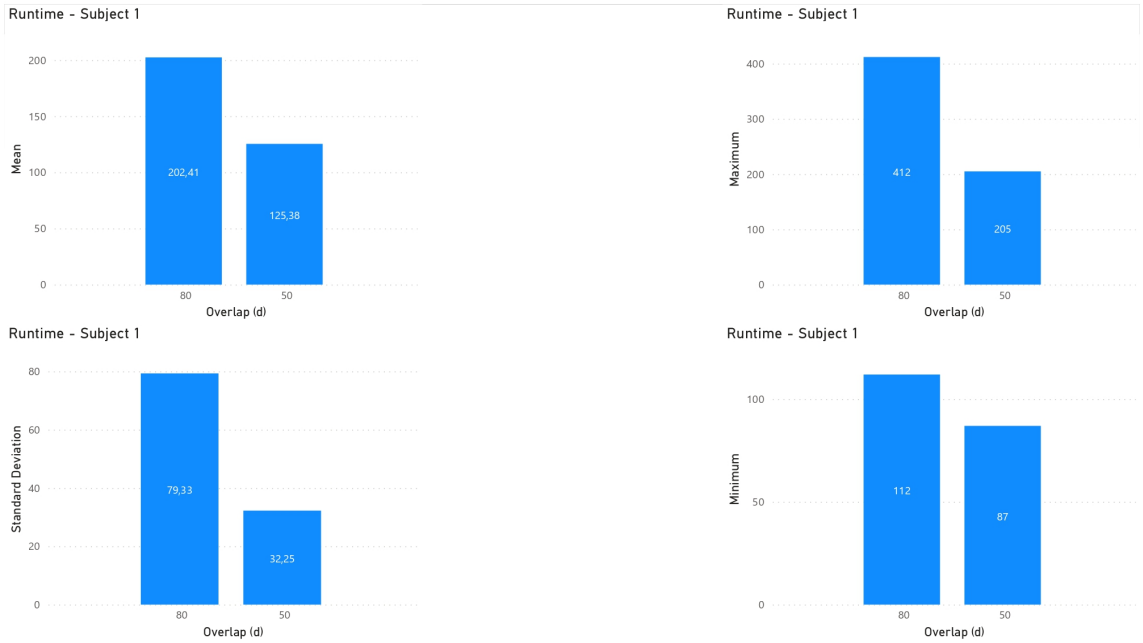


Figure 79: Runtime sort of by Overlap. Dataset filtered by Subject 1.

B.5 OUTCOMES FILTERED BY SUBJECT 2

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s2	Cz	cnn	Batch16	80	343	81.44%	74.90%	NaN	78.03%	79.45%	72.35%
s2	Cz	cnn	Batch16	50	186	80.00%	76.25%	63.08%	86.60%	78.03%	78.29%
s2	Cz	cnn	Batch32	80	222	82.58%	79.87%	NaN	81.23%	80.92%	73.80%
s2	Cz	cnn	Batch32	50	131	80.91%	77.70%	62.35%	85.96%	77.11%	76.60%
s2	Cz	cnn	Batch64	80	164	80.30%	76.99%	NaN	81.84%	76.32%	78.64%
s2	Cz	cnn	Batch64	50	110	80.91%	78.19%	61.61%	84.86%	77.27%	79.11%
s2	Cz	cnn	Batch128	80	140	82.58%	79.95%	NaN	82.10%	78.70%	77.41%
s2	Cz	cnn	Batch128	50	98	81.82%	79.35%	63.05%	85.45%	78.03%	81.83%
s2	2Ch	cnn	Batch16	80	344	93.18%	89.22%	NaN	89.42%	92.55%	74.64%
s2	2Ch	cnn	Batch16	50	186	85.45%	82.48%	69.74%	87.17%	82.88%	80.63%
s2	2Ch	cnn	Batch32	80	224	92.05%	91.27%	NaN	91.32%	92.11%	80.71%
s2	2Ch	cnn	Batch32	50	131	79.09%	75.33%	59.69%	85.77%	75.11%	78.61%
s2	2Ch	cnn	Batch64	80	165	92.42%	91.88%	NaN	91.49%	92.78%	78.12%
s2	2Ch	cnn	Batch64	50	110	81.82%	79.08%	64.16%	86.92%	77.91%	81.25%
s2	2Ch	cnn	Batch128	80	139	86.36%	84.65%	NaN	87.78%	83.32%	78.06%
s2	2Ch	cnn	Batch128	50	98	84.55%	82.57%	69.07%	88.40%	81.06%	79.46%
s2	5Ch	cnn	Batch16	80	344	83.71%	76.02%	NaN	79.14%	79.60%	81.21%
s2	5Ch	cnn	Batch16	50	186	90.00%	89.15%	80.74%	90.96%	89.85%	83.59%
s2	5Ch	cnn	Batch32	80	220	84.09%	82.93%	NaN	85.18%	85.53%	83.57%
s2	5Ch	cnn	Batch32	50	130	90.00%	89.35%	80.19%	91.36%	88.89%	83.00%
s2	5Ch	cnn	Batch64	80	165	86.74%	85.58%	NaN	86.57%	87.47%	84.74%
s2	5Ch	cnn	Batch64	50	108	89.09%	88.33%	78.08%	90.26%	87.91%	82.03%
s2	5Ch	cnn	Batch128	80	145	83.71%	82.10%	NaN	82.73%	82.49%	84.59%
s2	5Ch	cnn	Batch128	50	100	89.09%	88.21%	77.51%	90.47%	87.12%	81.88%
s2	16CH	cnn	Batch16	80	351	91.29%	84.67%	NaN	84.70%	88.78%	79.51%
s2	16CH	cnn	Batch16	50	190	83.64%	82.94%	71.17%	84.51%	86.83%	84.08%
s2	16CH	cnn	Batch32	80	249	87.88%	85.65%	NaN	87.24%	86.82%	82.05%
s2	16CH	cnn	Batch32	50	141	89.09%	88.81%	79.00%	88.79%	90.23%	82.00%
s2	16CH	cnn	Batch64	80	182	89.77%	86.89%	NaN	87.71%	87.24%	82.41%
s2	16CH	cnn	Batch64	50	113	85.45%	84.34%	69.84%	86.34%	83.56%	74.82%
s2	16CH	cnn	Batch128	80	145	89.02%	87.57%	NaN	88.46%	87.05%	83.68%
s2	16CH	cnn	Batch128	50	100	87.27%	86.25%	73.56%	88.40%	85.23%	82.07%

Figure 80: Table filter by CNN model (Subject 2).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s2	Cz	lstm	Batch16	80	365	80.30%	71.75%	NaN	73.04%	75.42%	72.70%
s2	Cz	lstm	Batch16	50	192	82.73%	79.88%	66.04%	86.41%	80.14%	72.59%
s2	Cz	lstm	Batch32	80	232	84.09%	81.88%	NaN	85.68%	82.30%	77.33%
s2	Cz	lstm	Batch32	50	137	84.55%	82.37%	69.65%	88.63%	81.55%	80.80%
s2	Cz	lstm	Batch64	80	165	81.82%	79.15%	NaN	83.83%	77.99%	77.21%
s2	Cz	lstm	Batch64	50	106	83.64%	81.45%	67.65%	87.96%	80.17%	75.19%
s2	Cz	lstm	Batch128	80	134	83.33%	80.75%	NaN	85.94%	78.79%	77.53%
s2	Cz	lstm	Batch128	50	96	84.55%	82.57%	69.07%	88.40%	81.06%	80.44%
s2	2Ch	lstm	Batch16	80	362	88.26%	78.59%	NaN	80.43%	82.45%	68.78%
s2	2Ch	lstm	Batch16	50	192	87.27%	84.74%	71.92%	86.93%	85.13%	77.66%
s2	2Ch	lstm	Batch32	80	232	83.71%	81.16%	NaN	81.33%	81.65%	74.60%
s2	2Ch	lstm	Batch32	50	136	87.27%	86.48%	75.83%	88.90%	87.05%	78.05%
s2	2Ch	lstm	Batch64	80	163	85.23%	82.96%	NaN	83.52%	82.73%	71.26%
s2	2Ch	lstm	Batch64	50	108	88.18%	87.41%	76.28%	89.19%	87.18%	82.75%
s2	2Ch	lstm	Batch128	80	134	86.36%	84.62%	NaN	86.76%	83.39%	77.06%
s2	2Ch	lstm	Batch128	50	96	88.18%	87.30%	75.41%	89.10%	86.36%	81.95%
s2	5Ch	lstm	Batch16	80	366	90.15%	82.68%	NaN	84.06%	84.24%	79.63%
s2	5Ch	lstm	Batch16	50	190	84.55%	83.11%	68.63%	84.44%	84.28%	82.20%
s2	5Ch	lstm	Batch32	80	232	87.12%	84.95%	NaN	87.29%	86.13%	80.95%
s2	5Ch	lstm	Batch32	50	137	82.73%	81.54%	63.61%	82.64%	81.02%	81.59%
s2	5Ch	lstm	Batch64	80	166	87.88%	85.42%	NaN	88.12%	84.58%	84.77%
s2	5Ch	lstm	Batch64	50	108	86.36%	85.63%	71.86%	86.31%	85.57%	78.93%
s2	5Ch	lstm	Batch128	80	142	89.39%	87.11%	NaN	89.07%	85.97%	82.50%
s2	5Ch	lstm	Batch128	50	96	89.09%	88.55%	77.16%	88.90%	88.26%	78.06%
s2	16CH	lstm	Batch16	80	369	92.05%	86.29%	NaN	86.25%	89.21%	76.75%
s2	16CH	lstm	Batch16	50	193	89.09%	87.85%	79.13%	89.57%	89.92%	81.84%
s2	16CH	lstm	Batch32	80	271	86.74%	83.80%	NaN	85.34%	84.34%	76.22%
s2	16CH	lstm	Batch32	50	149	90.00%	89.33%	79.43%	90.50%	88.95%	81.60%
s2	16CH	lstm	Batch64	80	184	93.94%	91.89%	NaN	91.85%	92.19%	79.49%
s2	16CH	lstm	Batch64	50	111	86.36%	85.70%	71.44%	85.64%	85.80%	76.18%
s2	16CH	lstm	Batch128	80	143	88.64%	86.95%	NaN	88.85%	85.80%	82.54%
s2	16CH	lstm	Batch128	50	98	91.82%	91.57%	83.28%	91.24%	92.05%	75.89%

Figure 81: Table filter by LSTM model (Subject 2).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s2	Cz	gru	Batch16	80	359	79.55%	69.97%	NaN	73.36%	75.30%	74.50%
s2	Cz	gru	Batch16	50	189	81.82%	78.18%	64.27%	86.28%	78.93%	78.37%
s2	Cz	gru	Batch32	80	232	78.79%	75.52%	NaN	83.15%	76.52%	79.63%
s2	Cz	gru	Batch32	50	135	80.91%	78.18%	62.43%	85.46%	77.60%	81.83%
s2	Cz	gru	Batch64	80	161	79.17%	75.36%	NaN	83.59%	75.18%	76.35%
s2	Cz	gru	Batch64	50	105	82.73%	80.26%	66.05%	87.53%	79.12%	78.27%
s2	Cz	gru	Batch128	80	131	79.17%	74.81%	NaN	83.87%	72.80%	75.91%
s2	Cz	gru	Batch128	50	94	82.73%	80.52%	64.86%	86.06%	79.17%	80.07%
s2	2Ch	gru	Batch16	80	362	85.61%	81.53%	NaN	83.02%	85.96%	73.93%
s2	2Ch	gru	Batch16	50	190	85.45%	83.45%	69.92%	86.52%	83.55%	79.81%
s2	2Ch	gru	Batch32	80	230	87.50%	85.11%	NaN	88.15%	83.77%	75.82%
s2	2Ch	gru	Batch32	50	136	89.09%	88.27%	78.36%	90.78%	87.67%	84.60%
s2	2Ch	gru	Batch64	80	162	87.50%	85.77%	NaN	88.06%	84.61%	81.78%
s2	2Ch	gru	Batch64	50	107	90.00%	89.26%	79.62%	91.21%	88.47%	80.44%
s2	2Ch	gru	Batch128	80	129	87.88%	86.03%	NaN	88.85%	84.54%	81.48%
s2	2Ch	gru	Batch128	50	93	86.36%	85.01%	72.08%	88.53%	83.71%	79.81%
s2	5Ch	gru	Batch16	80	360	91.67%	82.61%	NaN	83.40%	83.04%	74.83%
s2	5Ch	gru	Batch16	50	190	91.82%	91.22%	83.96%	91.98%	92.05%	80.40%
s2	5Ch	gru	Batch32	80	228	90.53%	87.52%	NaN	89.92%	87.77%	84.17%
s2	5Ch	gru	Batch32	50	135	87.27%	86.51%	73.83%	87.08%	86.77%	81.46%
s2	5Ch	gru	Batch64	80	166	89.77%	86.99%	NaN	90.87%	85.48%	82.81%
s2	5Ch	gru	Batch64	50	105	92.73%	92.36%	85.48%	93.09%	92.42%	81.22%
s2	5Ch	gru	Batch128	80	138	90.53%	89.08%	NaN	91.21%	88.01%	83.48%
s2	5Ch	gru	Batch128	50	95	90.00%	89.25%	79.34%	91.13%	88.26%	84.52%
s2	16CH	gru	Batch16	80	366	95.08%	88.62%	NaN	87.80%	91.93%	83.22%
s2	16CH	gru	Batch16	50	191	93.64%	92.82%	87.08%	93.19%	94.05%	76.43%
s2	16CH	gru	Batch32	80	271	91.67%	89.77%	NaN	91.54%	89.42%	80.17%
s2	16CH	gru	Batch32	50	147	91.82%	91.49%	84.43%	92.05%	92.42%	80.41%
s2	16CH	gru	Batch64	80	179	91.67%	89.30%	NaN	90.49%	88.87%	81.40%
s2	16CH	gru	Batch64	50	110	92.73%	92.44%	85.14%	92.09%	93.06%	80.94%
s2	16CH	gru	Batch128	80	143	91.67%	89.63%	NaN	90.45%	89.01%	85.71%
s2	16CH	gru	Batch128	50	96	89.09%	88.86%	78.26%	88.50%	89.77%	82.86%

Figure 82: Table filter by GRU model (Subject 2).

Filter_by	Channels	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s2	Cz	cnn-lstm	Batch16	80	481	84.09%	80.47%	NaN	83.16%	84.38%	77.87%
s2	Cz	cnn-lstm	Batch16	50	244	86.36%	85.07%	73.56%	87.17%	86.73%	80.11%
s2	Cz	cnn-lstm	Batch32	80	303	82.58%	79.96%	NaN	84.12%	80.96%	75.24%
s2	Cz	cnn-lstm	Batch32	50	168	86.36%	84.77%	72.85%	89.59%	83.55%	85.32%
s2	Cz	cnn-lstm	Batch64	80	214	81.82%	79.13%	NaN	82.00%	79.30%	76.65%
s2	Cz	cnn-lstm	Batch64	50	130	82.73%	80.17%	66.33%	87.74%	79.29%	84.32%
s2	Cz	cnn-lstm	Batch128	80	174	82.20%	79.16%	NaN	83.54%	77.31%	77.82%
s2	Cz	cnn-lstm	Batch128	50	114	72.73%	64.88%	46.77%	84.38%	65.91%	68.65%
s2	2Ch	cnn-lstm	Batch16	80	483	88.64%	85.01%	NaN	86.37%	90.37%	76.80%
s2	2Ch	cnn-lstm	Batch16	50	242	84.55%	81.21%	69.74%	89.95%	80.95%	79.08%
s2	2Ch	cnn-lstm	Batch32	80	300	90.15%	88.21%	NaN	88.38%	88.82%	82.02%
s2	2Ch	cnn-lstm	Batch32	50	167	86.36%	84.77%	72.85%	89.59%	83.55%	77.22%
s2	2Ch	cnn-lstm	Batch64	80	215	88.64%	86.65%	NaN	87.56%	86.55%	81.46%
s2	2Ch	cnn-lstm	Batch64	50	129	83.64%	81.60%	67.67%	87.32%	80.82%	77.38%
s2	2Ch	cnn-lstm	Batch128	80	174	87.88%	86.29%	NaN	88.35%	85.11%	80.73%
s2	2Ch	cnn-lstm	Batch128	50	113	86.36%	84.83%	72.60%	89.53%	83.33%	77.87%
s2	5Ch	cnn-lstm	Batch16	80	485	85.61%	77.30%	NaN	79.67%	81.47%	77.77%
s2	5Ch	cnn-lstm	Batch16	50	242	90.91%	90.54%	83.46%	91.07%	92.47%	81.53%
s2	5Ch	cnn-lstm	Batch32	80	299	77.27%	76.73%	NaN	79.77%	81.92%	76.07%
s2	5Ch	cnn-lstm	Batch32	50	166	89.09%	88.70%	78.98%	89.39%	89.61%	85.50%
s2	5Ch	cnn-lstm	Batch64	80	213	85.61%	83.73%	NaN	85.79%	84.21%	82.77%
s2	5Ch	cnn-lstm	Batch64	50	129	89.09%	88.82%	79.25%	89.45%	89.83%	78.39%
s2	5Ch	cnn-lstm	Batch128	80	182	89.02%	86.89%	NaN	87.95%	86.40%	82.11%
s2	5Ch	cnn-lstm	Batch128	50	116	91.82%	91.44%	82.90%	91.62%	91.29%	82.37%
s2	16CH	cnn-lstm	Batch16	80	497	88.26%	83.08%	NaN	84.71%	88.99%	75.50%
s2	16CH	cnn-lstm	Batch16	50	248	89.09%	88.06%	77.91%	87.69%	90.36%	85.47%
s2	16CH	cnn-lstm	Batch32	80	352	92.80%	90.91%	NaN	90.60%	92.27%	82.96%
s2	16CH	cnn-lstm	Batch32	50	188	88.18%	87.78%	77.94%	88.68%	89.30%	74.98%
s2	16CH	cnn-lstm	Batch64	80	236	92.05%	89.75%	NaN	89.73%	90.20%	86.46%
s2	16CH	cnn-lstm	Batch64	50	137	88.18%	87.44%	75.41%	88.84%	86.62%	76.42%
s2	16CH	cnn-lstm	Batch128	80	184	89.02%	87.57%	NaN	87.37%	88.11%	79.98%
s2	16CH	cnn-lstm	Batch128	50	120	88.18%	87.30%	75.41%	89.10%	86.36%	74.60%

Figure 83: Table filter by CNN-LSTM model (Subject 2).

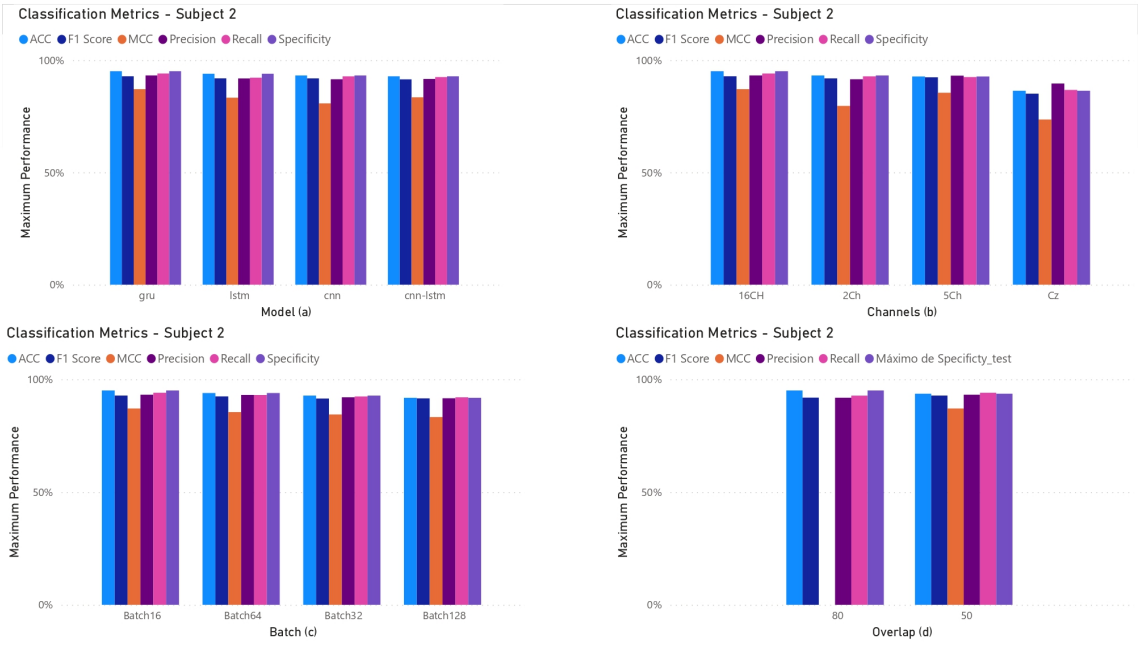


Figure 84: Classification Metrics Graphics sort of by (a) Models, (b) Channels, (c) Batch and (d) Overlap. Dataset filtered by Subject 2.

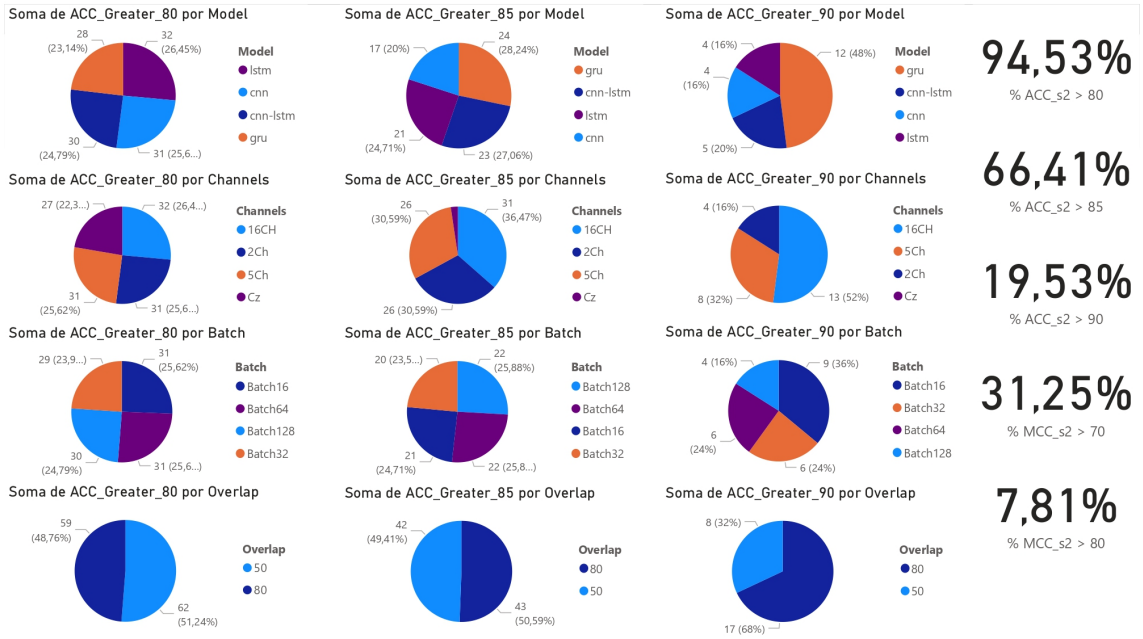


Figure 85: Accuracy greater 80% and 85% sort of by Model, Channels, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by subject 2.

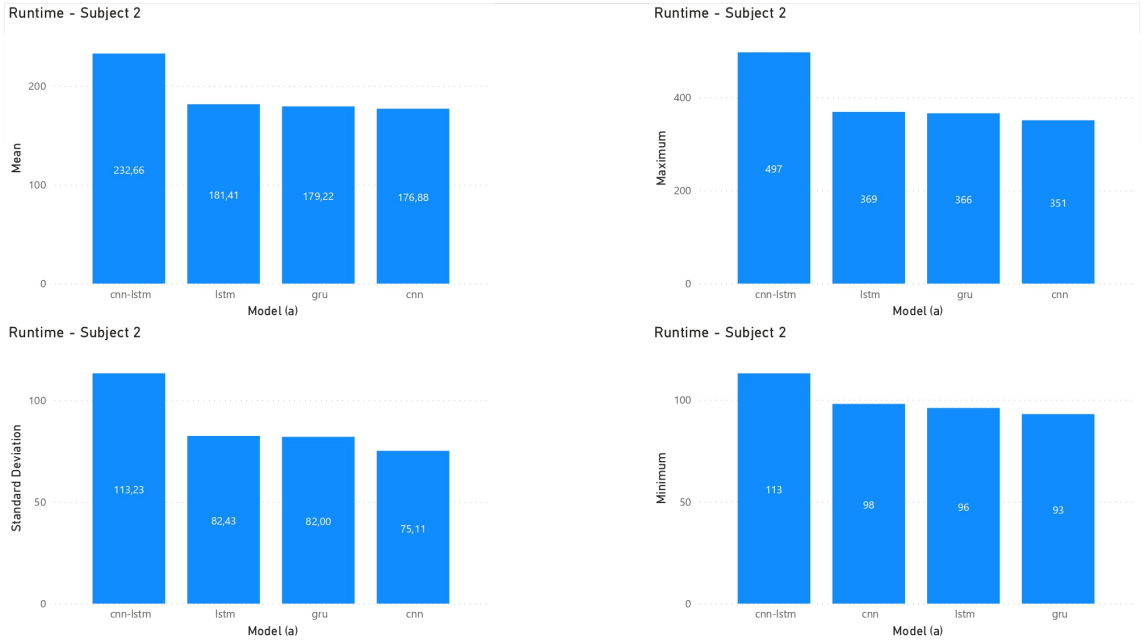


Figure 86: Runtime sort of by Model. Dataset filtered by Subject 2.

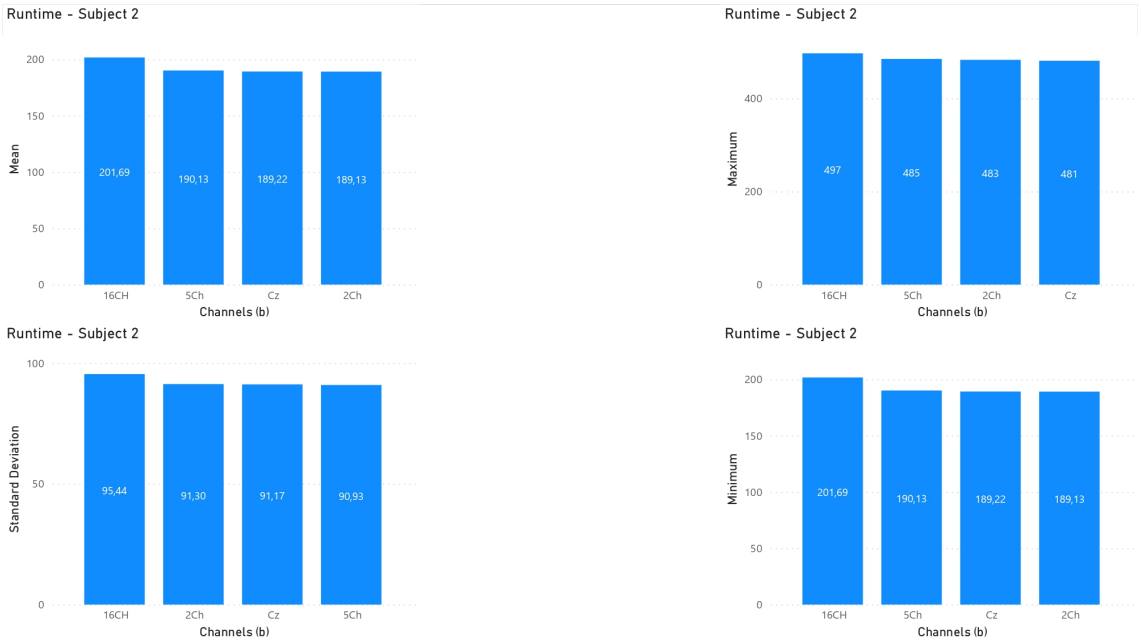


Figure 87: Runtime sort of by Channels. Dataset filtered by Subject 2.

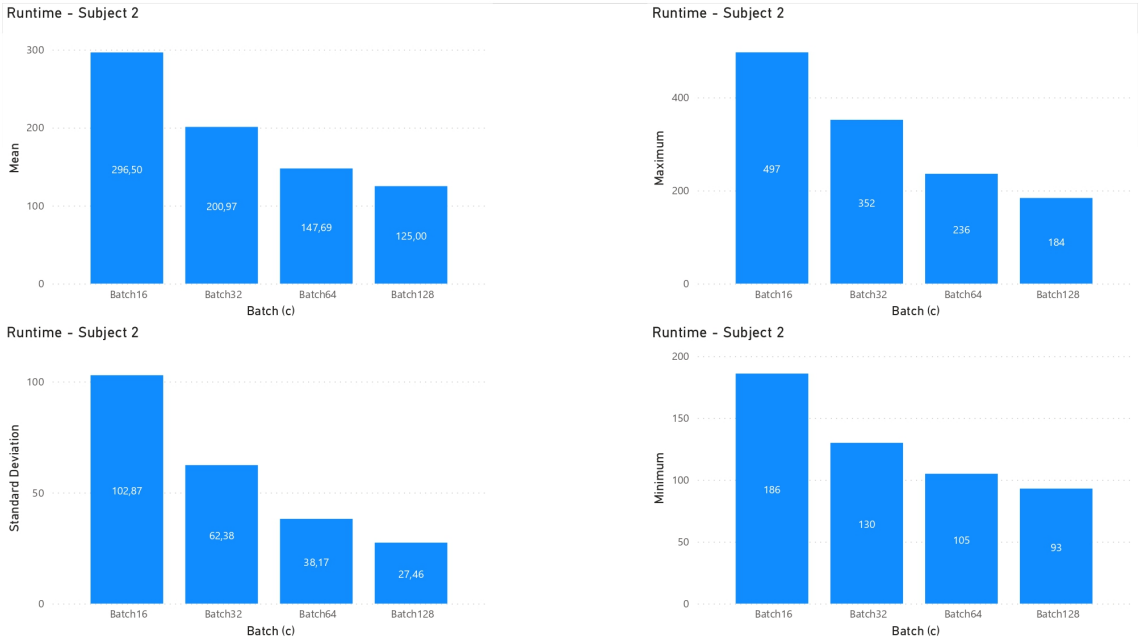


Figure 88: Runtime sort of by Batch. Dataset filtered by Subject 2.

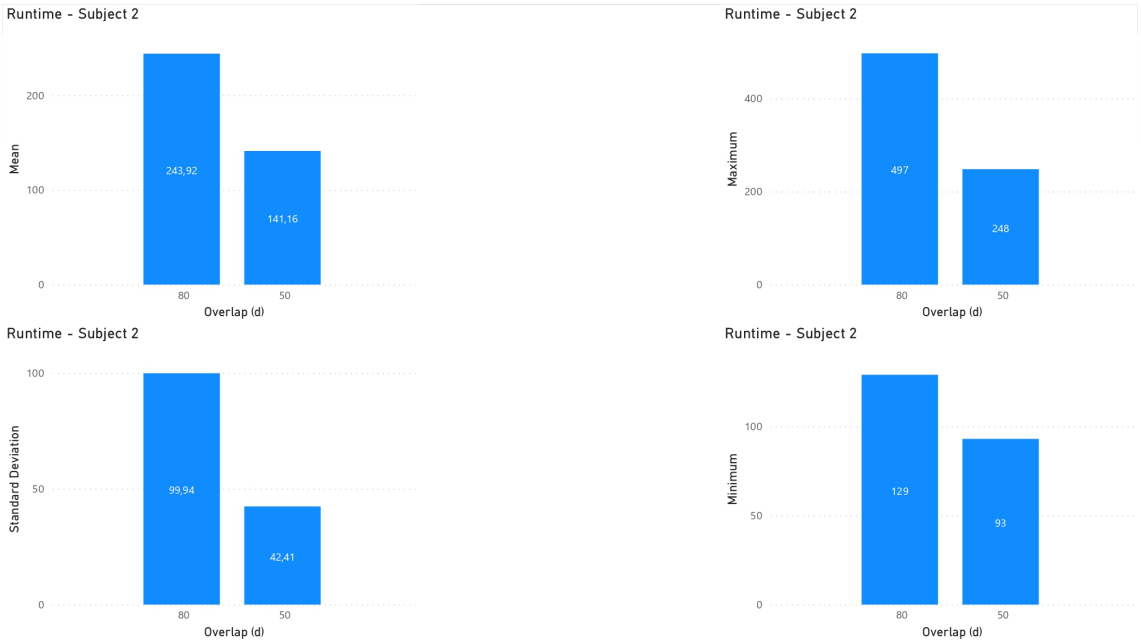


Figure 89: Runtime sort of by Overlap. Dataset filtered by Subject 2.

C

SLIP-LIKE PERTURBATION CLASSIFICATION METRICS FOR DEEP LEARNING MODELS - ICS

C.1 OUTCOMES OF ENTIRE DATASET (NO FILTER):

Filter_by	ICs	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
all	16	gru	Batch16	80	596	86.31%	73.39%	NaN	74.20%	74.68%	86.31%
all	16	lstm	Batch16	80	605	88.10%	78.72%	NaN	79.19%	81.09%	88.10%
all	16	cnn	Batch16	80	586	88.10%	78.94%	NaN	79.55%	82.94%	88.10%
all	16	cnn-lstm	Batch16	80	855	83.73%	77.08%	NaN	79.21%	79.26%	83.73%
all	16	gru	Batch16	50	292	81.90%	78.64%	NaN	85.08%	78.70%	81.90%
all	16	lstm	Batch16	50	296	82.86%	80.35%	NaN	85.63%	80.46%	82.86%
all	16	cnn	Batch16	50	289	80.00%	79.36%	NaN	80.62%	80.95%	80.00%
all	16	cnn-lstm	Batch16	50	385	85.24%	83.95%	NaN	85.37%	85.16%	85.24%
all	16	gru	Batch32	80	433	86.90%	82.46%	NaN	85.32%	82.54%	86.90%
all	16	lstm	Batch32	80	432	88.49%	86.38%	74.17%	88.36%	85.93%	88.49%
all	16	cnn	Batch32	80	391	82.14%	81.24%	66.60%	83.29%	83.40%	82.14%
all	16	cnn-lstm	Batch32	80	573	87.50%	86.14%	74.05%	87.34%	86.87%	87.50%
all	16	gru	Batch32	50	220	83.33%	80.64%	67.05%	87.79%	80.08%	83.33%
all	16	lstm	Batch32	50	224	87.62%	86.39%	74.84%	89.46%	85.57%	87.62%
all	16	cnn	Batch32	50	220	81.43%	80.08%	61.48%	82.03%	79.54%	81.43%
all	16	cnn-lstm	Batch32	50	311	86.19%	85.18%	72.00%	87.05%	85.13%	86.19%
all	16	gru	Batch64	80	294	87.10%	84.93%	72.64%	88.90%	84.10%	87.10%
all	16	lstm	Batch64	80	294	86.31%	83.71%	71.77%	89.83%	82.58%	86.31%
all	16	cnn	Batch64	80	285	79.96%	79.57%	63.31%	81.44%	81.88%	79.96%
all	16	cnn-lstm	Batch64	80	378	84.13%	83.69%	70.04%	84.67%	85.39%	84.13%
all	16	gru	Batch64	50	155	82.86%	81.01%	64.91%	84.81%	80.54%	82.86%
all	16	lstm	Batch64	50	155	83.81%	83.04%	67.10%	84.14%	83.01%	83.81%
all	16	cnn	Batch64	50	159	84.76%	84.33%	70.01%	84.97%	85.07%	84.76%
all	16	cnn-lstm	Batch64	50	200	81.43%	81.19%	64.99%	82.27%	82.75%	81.43%
all	16	gru	Batch128	80	200	83.93%	81.03%	64.61%	84.96%	79.95%	83.93%
all	16	lstm	Batch128	80	204	88.29%	86.59%	74.72%	89.38%	85.50%	88.29%
all	16	cnn	Batch128	80	216	80.16%	79.72%	61.42%	79.88%	81.57%	80.16%
all	16	cnn-lstm	Batch128	80	280	89.09%	88.56%	77.78%	88.22%	89.57%	89.09%
all	16	gru	Batch128	50	126	79.52%	76.08%	58.72%	84.38%	75.12%	79.52%
all	16	lstm	Batch128	50	123	84.29%	82.98%	66.95%	84.92%	82.09%	84.29%
all	16	cnn	Batch128	50	128	80.48%	78.93%	58.82%	80.60%	78.27%	80.48%
all	16	cnn-lstm	Batch128	50	196	80.00%	79.89%	62.72%	80.81%	81.92%	80.00%

Figure 90: Classification metrics results of all subjects and speeds - Entire Dataset

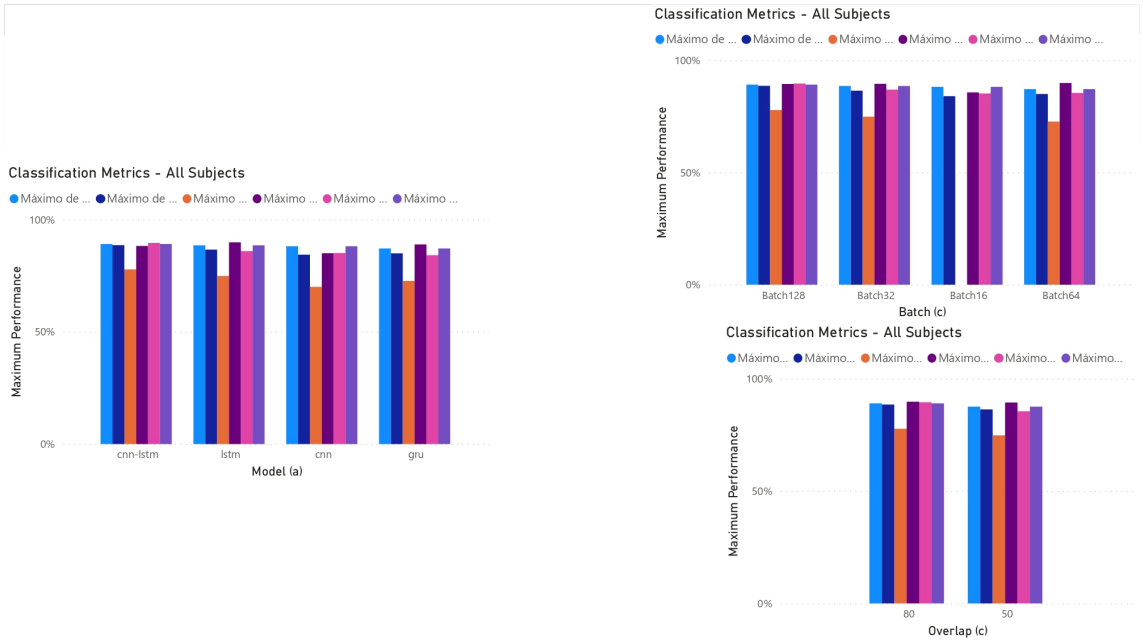


Figure 91: All subjects and speeds - Classification Metrics Graphics sort of by (a) Models, (b) Batch and (c) Overlap

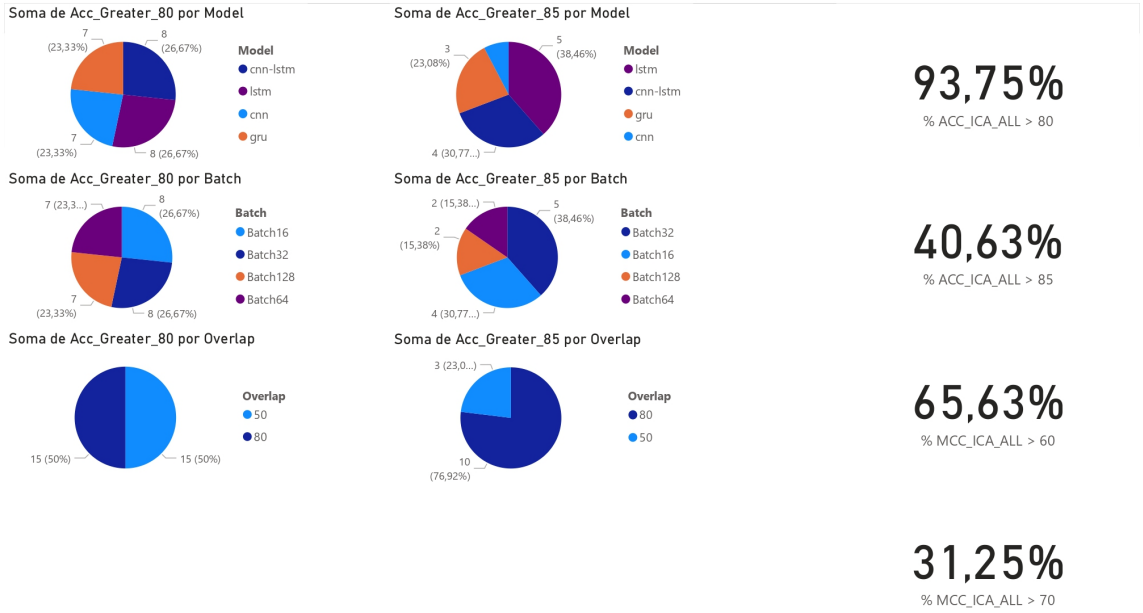


Figure 92: All subjects and speeds - Accuracy greater 80% and 85% sort of by Model, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%.

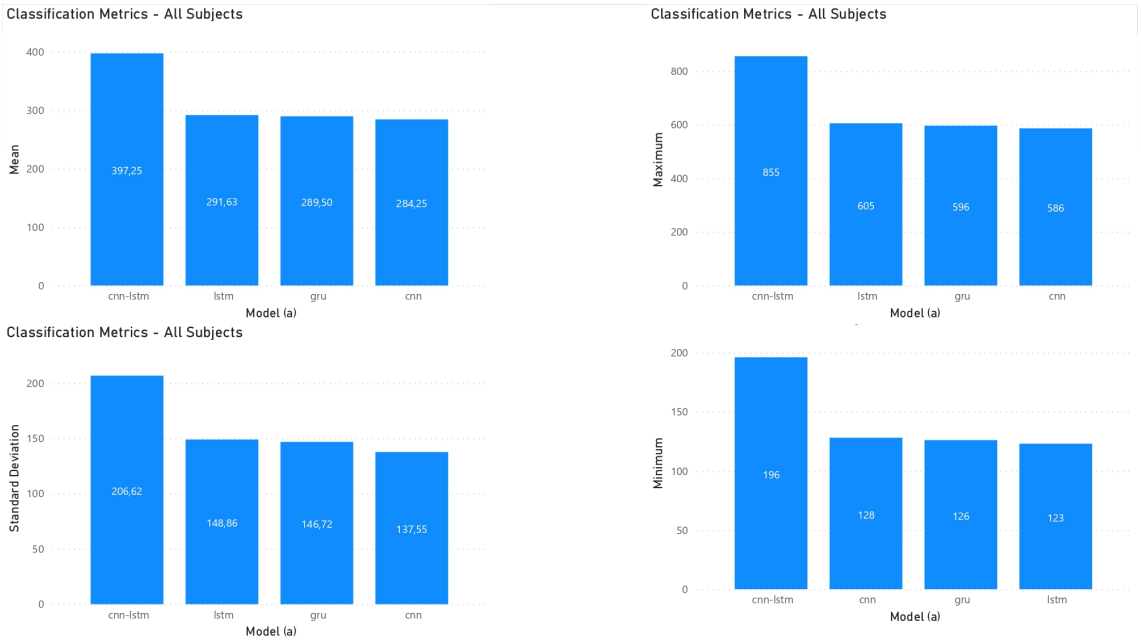


Figure 93: All subjects and speeds - Runtime sort of by Model.

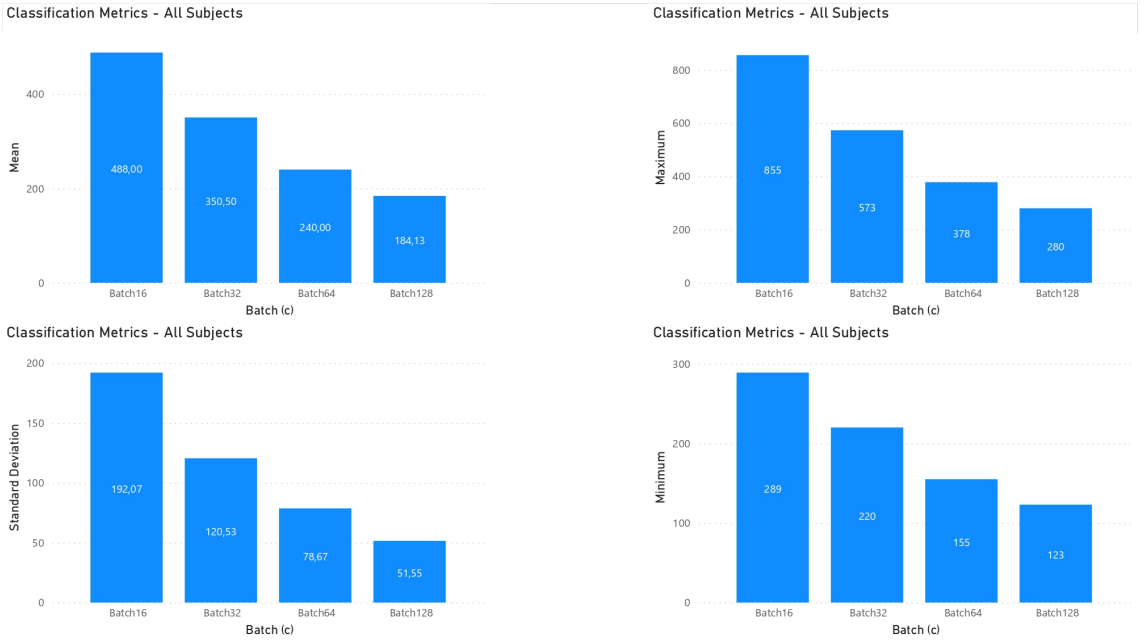


Figure 94: All subjects and speeds - Runtime sort of by Batch.

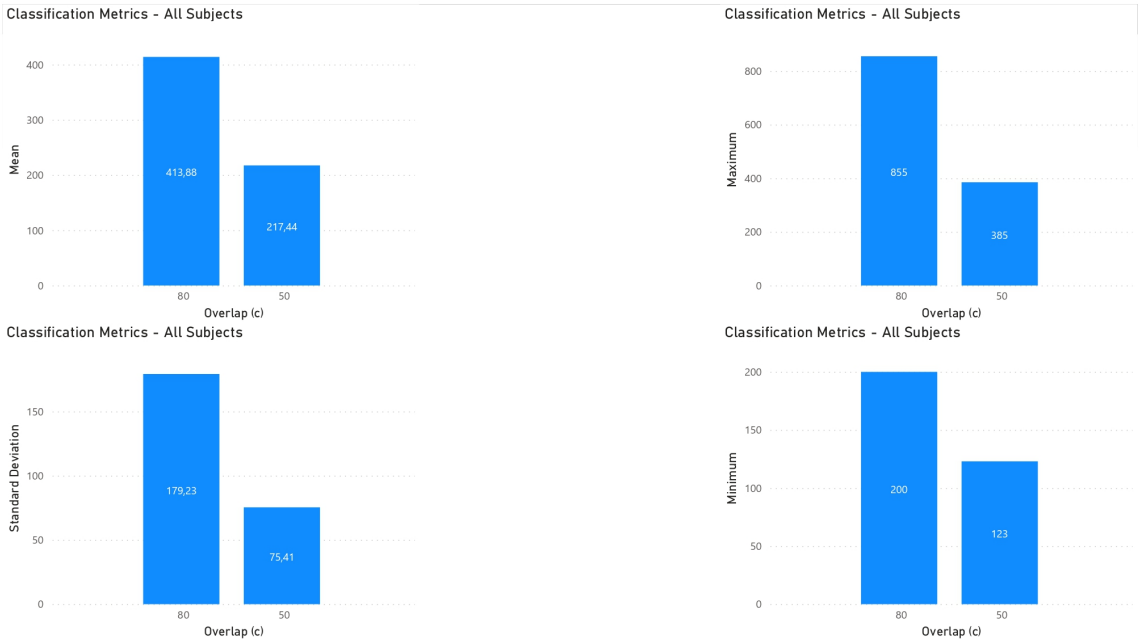


Figure 95: All subjects and speeds - Runtime sort of by Overlap.

C.2 OUTCOMES FILTERED SPEED = 1.6KM/H

Filter_by	ICs	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
1.6km	16	gru	Batch16	80	426	79.76%	71.79%	NaN	74.70%	74.00%	79.76%
1.6km	16	lstm	Batch16	80	449	79.46%	64.58%	NaN	69.41%	66.88%	79.46%
1.6km	16	cnn	Batch16	80	465	76.79%	70.35%	NaN	72.40%	73.56%	76.79%
1.6km	16	cnn-lstm	Batch16	80	564	74.11%	69.56%	45.60%	72.11%	74.21%	74.11%
1.6km	16	gru	Batch16	50	208	78.57%	75.21%	54.07%	77.48%	76.52%	78.57%
1.6km	16	lstm	Batch16	50	225	75.00%	72.78%	50.49%	76.44%	74.19%	75.00%
1.6km	16	cnn	Batch16	50	216	72.14%	70.13%	46.71%	75.02%	72.10%	72.14%
1.6km	16	cnn-lstm	Batch16	50	304	70.71%	68.82%	42.88%	72.84%	70.29%	70.71%
1.6km	16	gru	Batch32	80	292	77.68%	71.61%	NaN	76.28%	72.90%	77.68%
1.6km	16	lstm	Batch32	80	308	78.27%	71.93%	50.01%	77.10%	73.14%	78.27%
1.6km	16	cnn	Batch32	80	317	72.32%	69.24%	42.93%	71.86%	71.25%	72.32%
1.6km	16	cnn-lstm	Batch32	80	345	75.89%	70.93%	45.53%	73.26%	72.32%	75.89%
1.6km	16	gru	Batch32	50	155	77.86%	76.32%	54.71%	78.52%	76.34%	77.86%
1.6km	16	lstm	Batch32	50	157	75.00%	72.71%	48.92%	76.70%	72.52%	75.00%
1.6km	16	cnn	Batch32	50	153	75.71%	74.95%	51.97%	76.45%	75.57%	75.71%
1.6km	16	cnn-lstm	Batch32	50	188	75.00%	73.93%	50.82%	76.41%	74.54%	75.00%
1.6km	16	gru	Batch64	80	201	77.68%	73.34%	50.43%	78.08%	72.70%	77.68%
1.6km	16	lstm	Batch64	80	194	78.57%	76.15%	54.59%	78.38%	76.39%	78.57%
1.6km	16	cnn	Batch64	80	190	75.00%	72.80%	46.61%	73.85%	72.80%	75.00%
1.6km	16	cnn-lstm	Batch64	80	227	78.87%	75.93%	52.96%	77.00%	75.99%	78.87%
1.6km	16	gru	Batch64	50	124	79.29%	77.13%	56.82%	80.46%	76.52%	79.29%
1.6km	16	lstm	Batch64	50	120	73.57%	71.04%	43.96%	73.39%	70.67%	73.57%
1.6km	16	cnn	Batch64	50	120	76.43%	75.49%	51.92%	75.97%	75.97%	76.43%
1.6km	16	cnn-lstm	Batch64	50	140	73.57%	71.70%	45.03%	73.18%	71.93%	73.57%
1.6km	16	gru	Batch128	80	139	75.60%	69.85%	45.65%	77.40%	69.12%	75.60%
1.6km	16	lstm	Batch128	80	141	76.19%	73.19%	48.02%	75.83%	72.39%	76.19%
1.6km	16	cnn	Batch128	80	154	73.51%	72.45%	45.83%	72.45%	73.40%	73.51%
1.6km	16	cnn-lstm	Batch128	80	171	77.08%	73.24%	49.89%	78.24%	72.15%	77.08%
1.6km	16	gru	Batch128	50	104	80.00%	78.36%	57.80%	80.10%	77.75%	80.00%
1.6km	16	lstm	Batch128	50	102	76.43%	74.21%	50.35%	76.75%	73.71%	76.43%
1.6km	16	cnn	Batch128	50	103	75.71%	74.19%	49.41%	75.36%	74.11%	75.71%
1.6km	16	cnn-lstm	Batch128	50	121	79.29%	78.21%	57.37%	79.07%	78.35%	79.29%

Figure 96: Classification metrics results of dataset filtered by 1.6km/h (Speed)

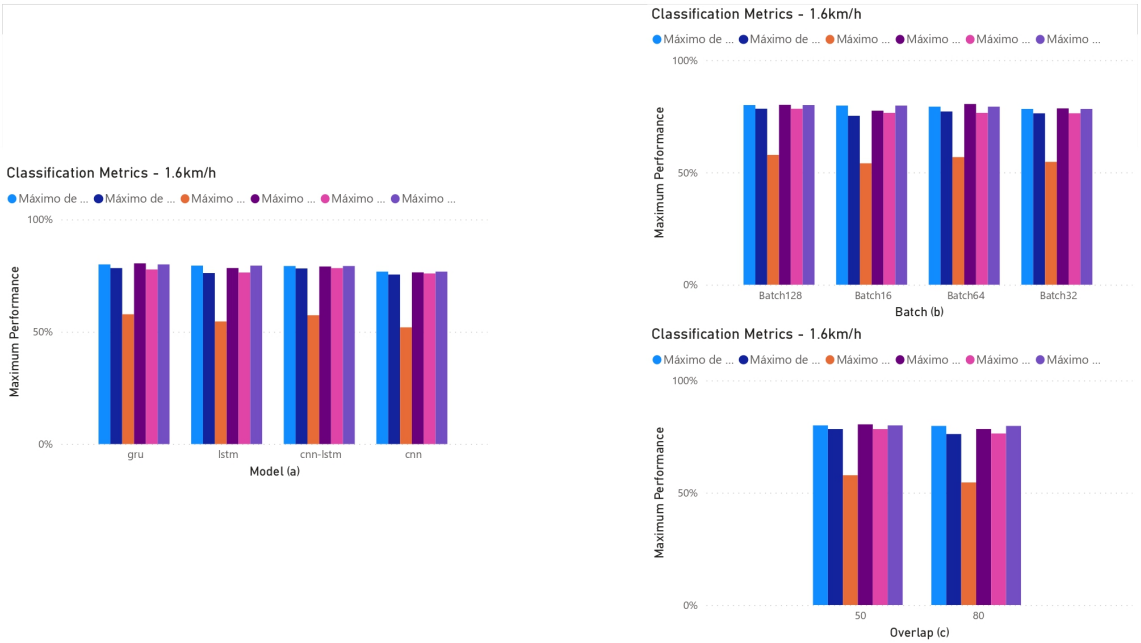


Figure 97: Classification Metrics Graphics sort of by (a) Models, (b) Batch and (c) Overlap. Dataset filtered by speed = 1.6km/h.

Figure 98: Accuracy greater 80% and 85% sort of by Model, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by speed = 1.6km/h.

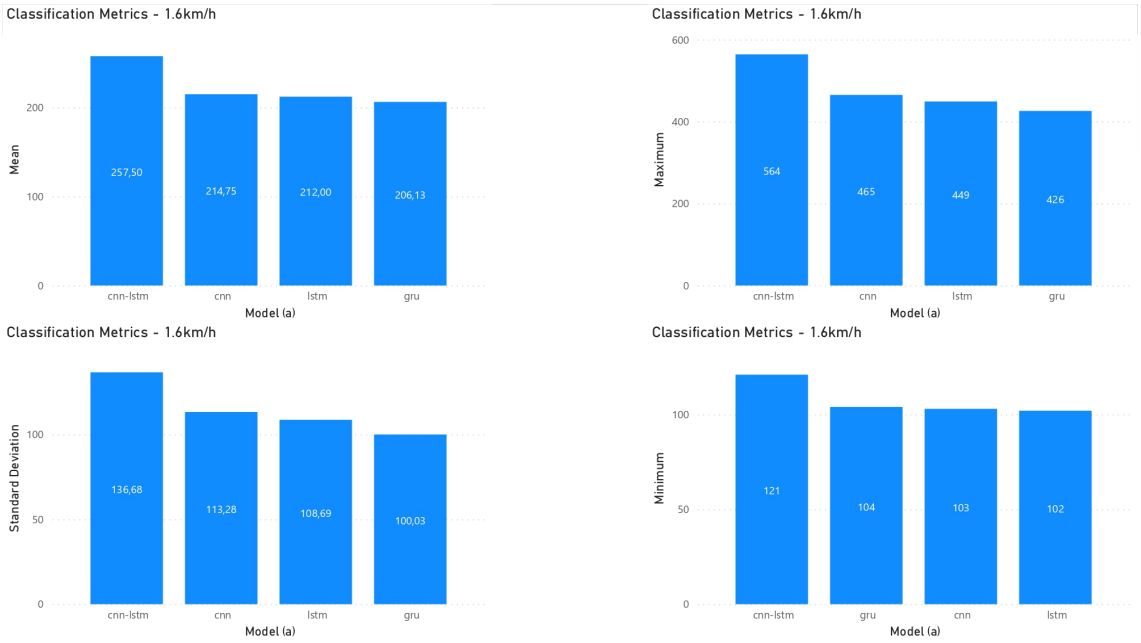


Figure 99: Runtime sort of by Model. Dataset filtered by speed = 1.6km/h.

Figure 100: Runtime sort of by Batch. Dataset filtered by speed = 1.6km/h.

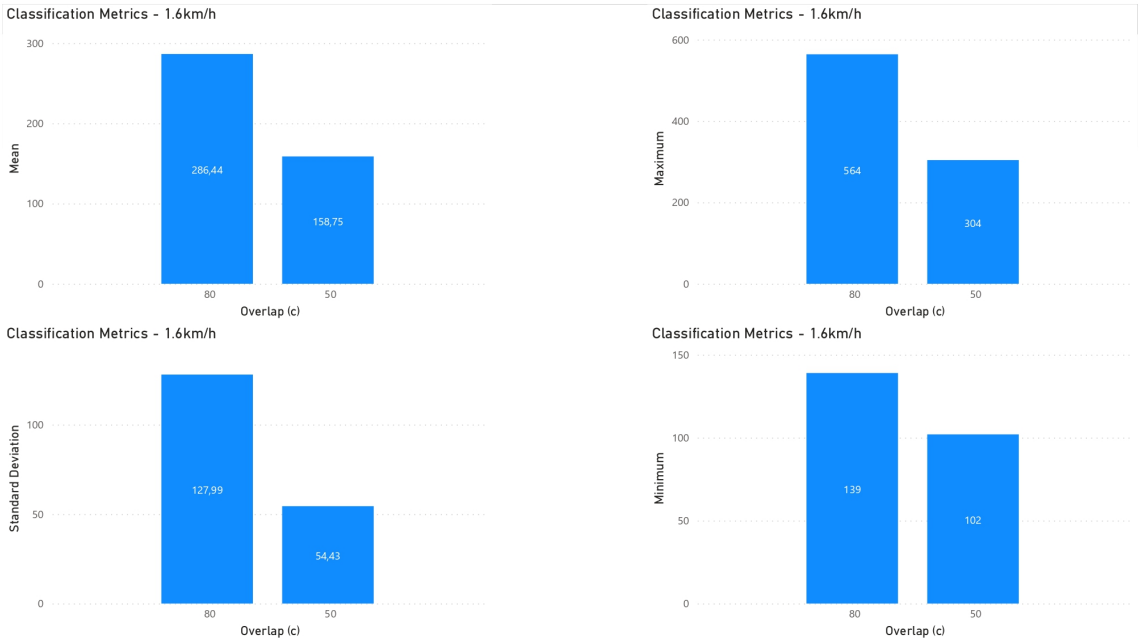


Figure 101: Runtime sort of by Overlap. Dataset filtered by speed = 1.6km/h.

C.3 OUTCOMES FILTERED BY SPEED = 2.5KM/H

Filter_by	ICs	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
2.5km	16	gru	Batch16	80	330	75.60%	61.46%	NaN	67.75%	65.44%	75.60%
2.5km	16	lstm	Batch16	80	336	78.57%	66.53%	NaN	70.16%	71.20%	78.57%
2.5km	16	cnn	Batch16	80	318	77.08%	72.25%	51.05%	76.88%	73.86%	77.08%
2.5km	16	cnn-lstm	Batch16	80	435	83.33%	75.43%	55.02%	78.57%	76.25%	83.33%
2.5km	16	gru	Batch16	50	180	72.14%	68.52%	40.68%	72.20%	68.71%	72.14%
2.5km	16	lstm	Batch16	50	180	75.71%	70.99%	50.31%	80.01%	71.76%	75.71%
2.5km	16	cnn	Batch16	50	176	85.71%	84.38%	69.77%	84.97%	84.86%	85.71%
2.5km	16	cnn-lstm	Batch16	50	222	80.71%	79.20%	61.53%	82.03%	79.90%	80.71%
2.5km	16	gru	Batch32	80	237	77.98%	73.90%	51.91%	78.17%	74.25%	77.98%
2.5km	16	lstm	Batch32	80	237	71.43%	59.43%	NaN	66.10%	63.14%	71.43%
2.5km	16	cnn	Batch32	80	226	74.11%	72.86%	53.08%	75.97%	77.62%	74.11%
2.5km	16	cnn-lstm	Batch32	80	295	83.63%	81.72%	66.95%	84.13%	83.29%	83.63%
2.5km	16	gru	Batch32	50	131	75.71%	72.07%	49.97%	79.37%	71.33%	75.71%
2.5km	16	lstm	Batch32	50	135	69.29%	63.81%	33.75%	70.18%	64.35%	69.29%
2.5km	16	cnn	Batch32	50	127	82.86%	81.11%	64.47%	84.43%	80.28%	82.86%
2.5km	16	cnn-lstm	Batch32	50	173	80.71%	79.34%	60.20%	81.43%	78.96%	80.71%
2.5km	16	gru	Batch64	80	183	77.98%	75.54%	52.87%	78.14%	74.96%	77.98%
2.5km	16	lstm	Batch64	80	179	72.92%	64.92%	36.75%	69.11%	66.89%	72.92%
2.5km	16	cnn	Batch64	80	171	82.14%	81.41%	65.01%	82.12%	82.95%	82.14%
2.5km	16	cnn-lstm	Batch64	80	226	81.25%	77.42%	62.17%	86.14%	77.44%	81.25%
2.5km	16	gru	Batch64	50	113	76.43%	71.36%	52.31%	82.81%	70.98%	76.43%
2.5km	16	lstm	Batch64	50	107	70.71%	64.50%	37.17%	73.23%	65.15%	70.71%
2.5km	16	cnn	Batch64	50	107	80.71%	79.54%	59.69%	80.41%	79.30%	80.71%
2.5km	16	cnn-lstm	Batch64	50	123	70.00%	69.16%	38.64%	69.02%	69.62%	70.00%
2.5km	16	gru	Batch128	80	136	78.27%	75.86%	53.72%	78.84%	75.15%	78.27%
2.5km	16	lstm	Batch128	80	129	79.46%	76.15%	55.37%	80.70%	75.20%	79.46%
2.5km	16	cnn	Batch128	80	135	84.23%	82.28%	66.56%	85.57%	81.23%	84.23%
2.5km	16	cnn-lstm	Batch128	80	161	81.55%	80.70%	63.26%	81.69%	81.62%	81.55%
2.5km	16	gru	Batch128	50	91	73.57%	69.63%	43.28%	74.41%	69.20%	73.57%
2.5km	16	lstm	Batch128	50	90	74.29%	70.27%	45.05%	75.67%	69.78%	74.29%
2.5km	16	cnn	Batch128	50	90	80.71%	78.91%	58.97%	80.89%	78.14%	80.71%
2.5km	16	cnn-lstm	Batch128	50	106	83.57%	82.52%	65.31%	83.20%	82.13%	83.57%

Figure 102: Classification metrics results of dataset filtered by 2.5km/h (Speed)

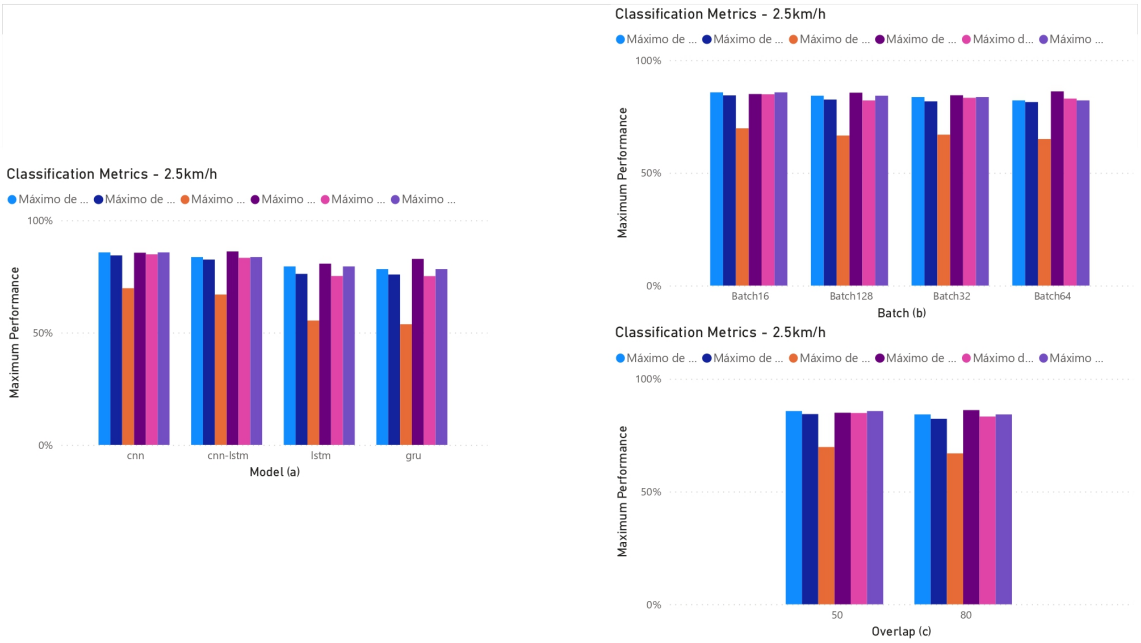


Figure 103: Classification Metrics Graphics sort of by (a) Models, (b) Batch and (c) Overlap. Dataset filtered by speed = 2.5km/h.

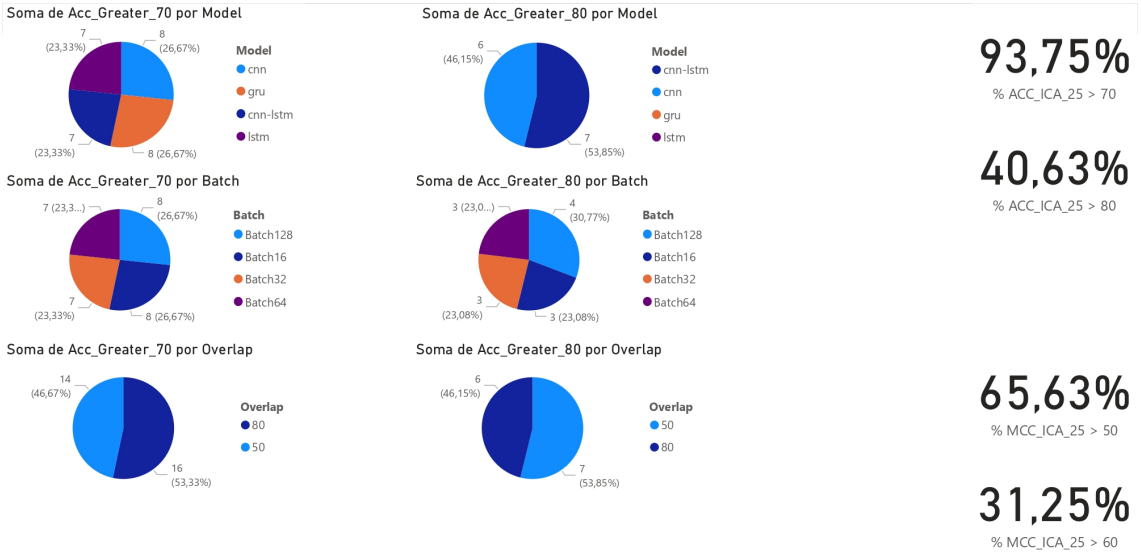


Figure 104: Accuracy greater 80% and 85% sort of by Model, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by speed = 2.5km/h.

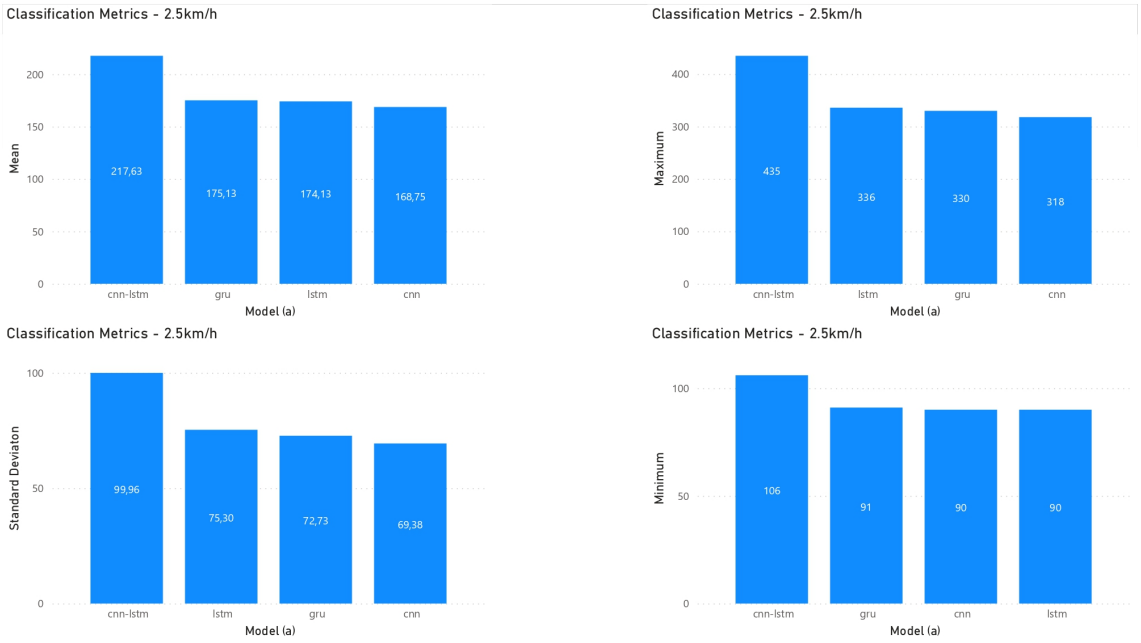


Figure 105: Runtime sort of by Model. Dataset filtered by speed = 2.5km/h.

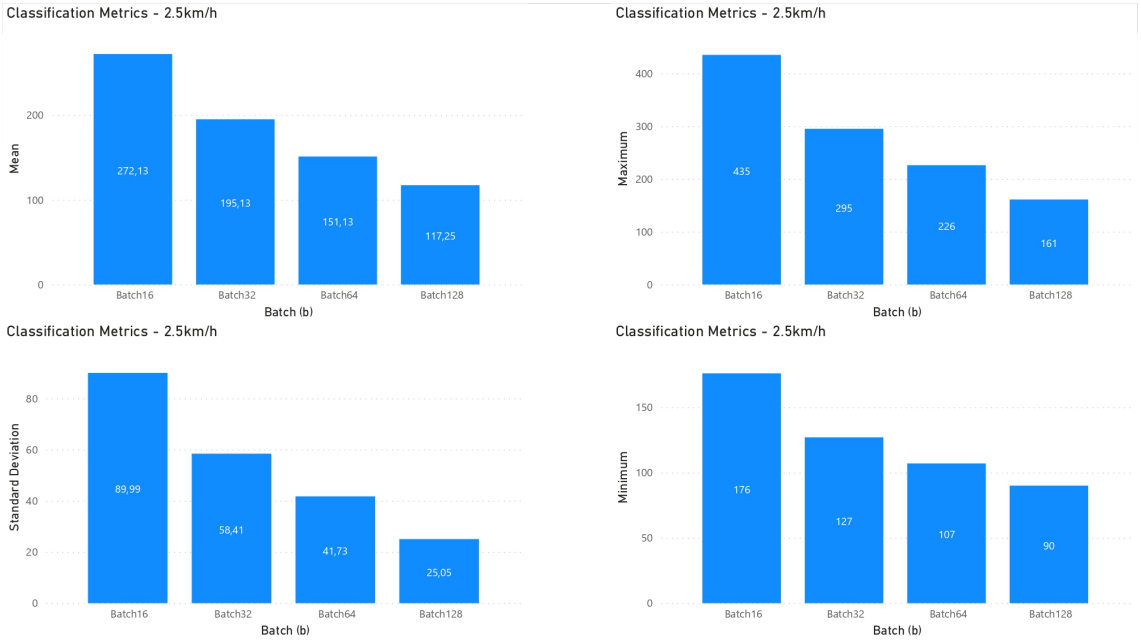


Figure 106: Runtime sort of by Batch. Dataset filtered by speed = 2.5km/h.

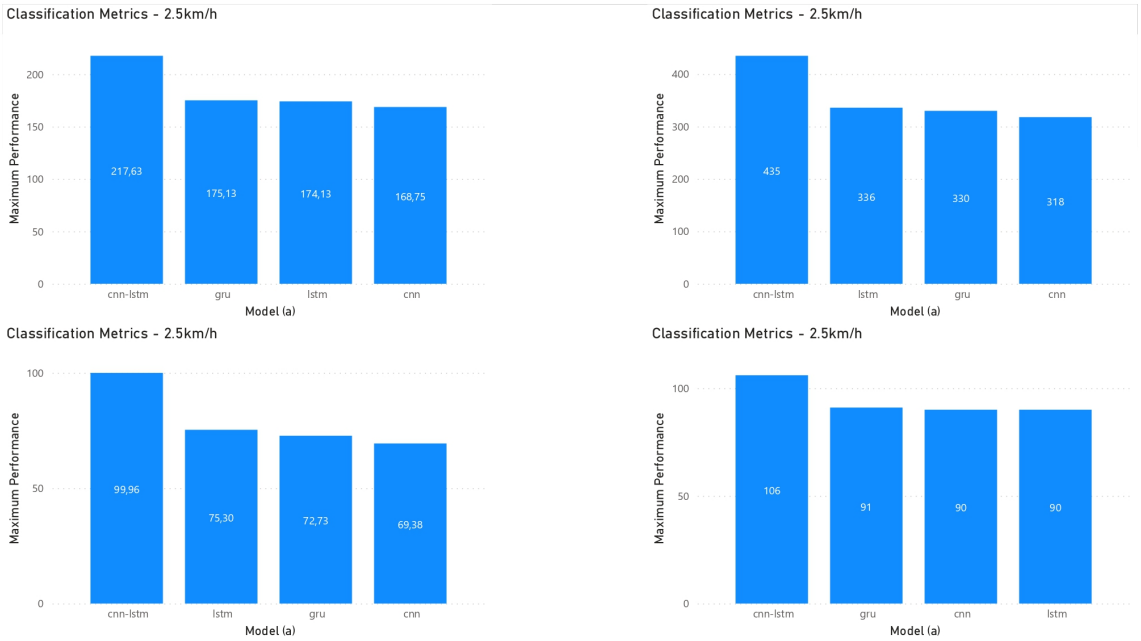


Figure 107: Runtime sort of by Overlap. Dataset filtered by speed = 2.5km/h.

C.4 OUTCOMES FILTERED BY SUBJECT 1

Filter_by	ICs	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
s1	16	gru	Batch16	80	296	89.06%	77.52%	NaN	79.65%	78.76%	89.06%
s1	16	lstm	Batch16	80	300	82.29%	69.44%	NaN	73.66%	70.46%	82.29%
s1	16	cnn	Batch16	80	283	83.85%	74.32%	NaN	77.38%	76.63%	83.85%
s1	16	cnn-lstm	Batch16	80	390	90.10%	79.47%	NaN	80.40%	81.09%	90.10%
s1	16	gru	Batch16	50	161	81.25%	78.12%	61.87%	85.09%	77.58%	81.25%
s1	16	lstm	Batch16	50	166	78.75%	75.79%	54.79%	79.49%	75.50%	78.75%
s1	16	cnn	Batch16	50	160	83.75%	83.03%	68.08%	83.99%	84.17%	83.75%
s1	16	cnn-lstm	Batch16	50	204	86.25%	84.29%	71.95%	88.82%	83.67%	86.25%
s1	16	gru	Batch32	80	214	80.73%	73.44%	55.15%	84.66%	73.75%	80.73%
s1	16	lstm	Batch32	80	216	82.29%	77.14%	58.14%	81.50%	77.02%	82.29%
s1	16	cnn	Batch32	80	214	82.29%	81.26%	65.04%	82.02%	83.15%	82.29%
s1	16	cnn-lstm	Batch32	80	278	86.46%	84.78%	70.26%	85.41%	84.92%	86.46%
s1	16	gru	Batch32	50	137	76.25%	71.83%	52.13%	81.69%	72.00%	76.25%
s1	16	lstm	Batch32	50	129	82.50%	79.90%	64.66%	86.61%	78.58%	82.50%
s1	16	cnn	Batch32	50	125	83.75%	82.84%	68.27%	84.48%	83.83%	83.75%
s1	16	cnn-lstm	Batch32	50	155	83.75%	82.43%	65.31%	83.25%	82.08%	83.75%
s1	16	gru	Batch64	80	149	85.42%	81.53%	67.56%	88.76%	80.15%	85.42%
s1	16	lstm	Batch64	80	152	83.85%	80.44%	62.92%	83.86%	79.25%	83.85%
s1	16	cnn	Batch64	80	152	84.90%	83.42%	67.23%	83.91%	83.33%	84.90%
s1	16	cnn-lstm	Batch64	80	191	89.58%	87.60%	75.95%	89.33%	86.69%	89.58%
s1	16	gru	Batch64	50	101	83.75%	82.05%	65.65%	84.57%	81.17%	83.75%
s1	16	lstm	Batch64	50	100	86.25%	85.03%	71.97%	87.82%	84.25%	86.25%
s1	16	cnn	Batch64	50	104	85.00%	84.13%	68.83%	84.67%	84.17%	85.00%
s1	16	cnn-lstm	Batch64	50	119	83.75%	82.91%	66.43%	83.27%	83.17%	83.75%
s1	16	gru	Batch128	80	116	85.42%	83.15%	68.16%	86.66%	81.70%	85.42%
s1	16	lstm	Batch128	80	119	88.54%	87.18%	74.94%	88.75%	86.23%	88.54%
s1	16	cnn	Batch128	80	121	88.54%	86.87%	75.28%	90.11%	85.34%	88.54%
s1	16	cnn-lstm	Batch128	80	147	85.42%	84.48%	69.49%	84.67%	84.84%	85.42%
s1	16	gru	Batch128	50	90	86.25%	85.01%	71.56%	87.82%	83.85%	86.25%
s1	16	lstm	Batch128	50	90	82.50%	80.77%	63.47%	83.93%	79.69%	82.50%
s1	16	cnn	Batch128	50	103	85.00%	84.00%	68.47%	85.16%	83.33%	85.00%
s1	16	cnn-lstm	Batch128	50	110	87.50%	86.26%	74.61%	89.88%	84.90%	87.50%

Figure 108: Classification metrics results of dataset filtered by Subject 1

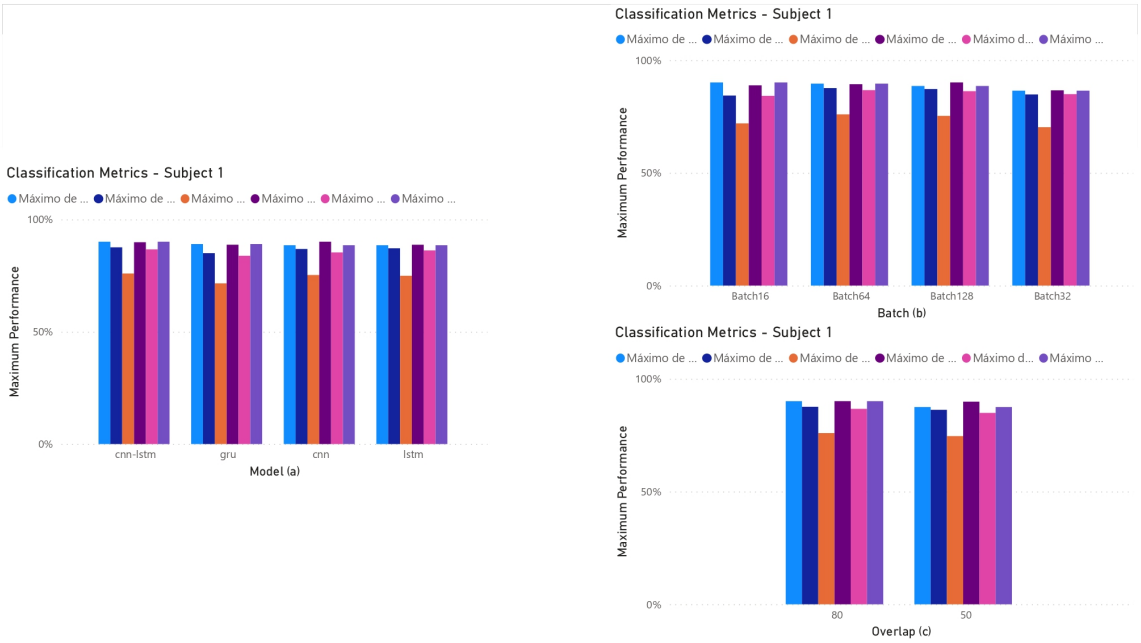


Figure 109: Classification Metrics Graphics sort of by (a) Models, (b) Batch and (c) Overlap. Dataset filtered by Subject 1.

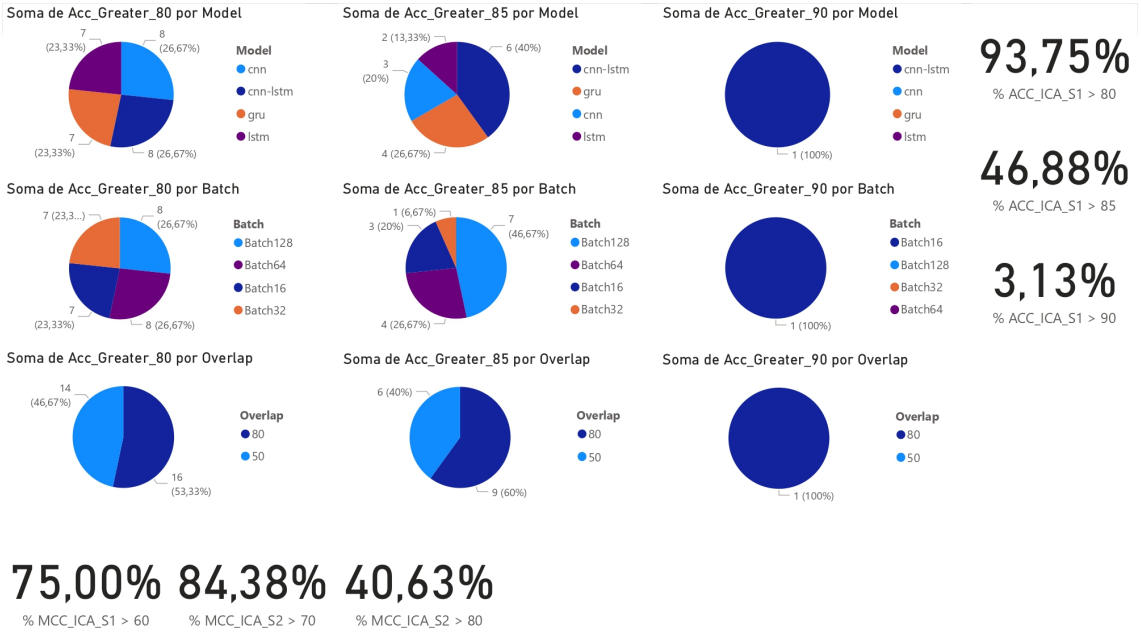


Figure 110: Accuracy greater 80% and 85% sort of by Model, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by subject 1.

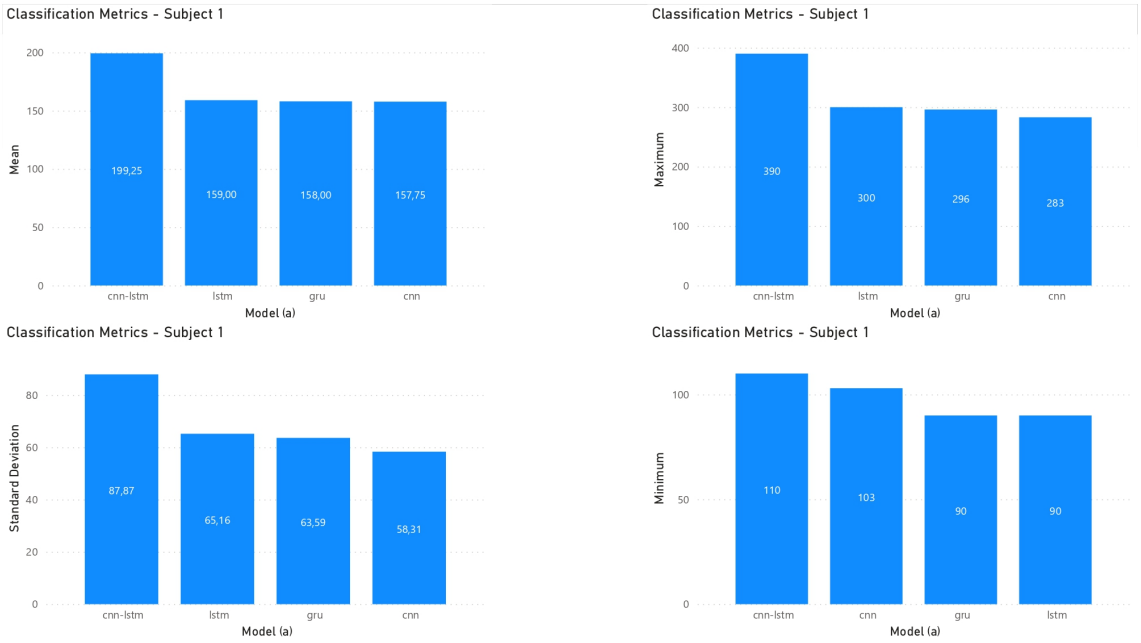


Figure 111: Runtime sort of by Model. Dataset filtered by Subject 1.

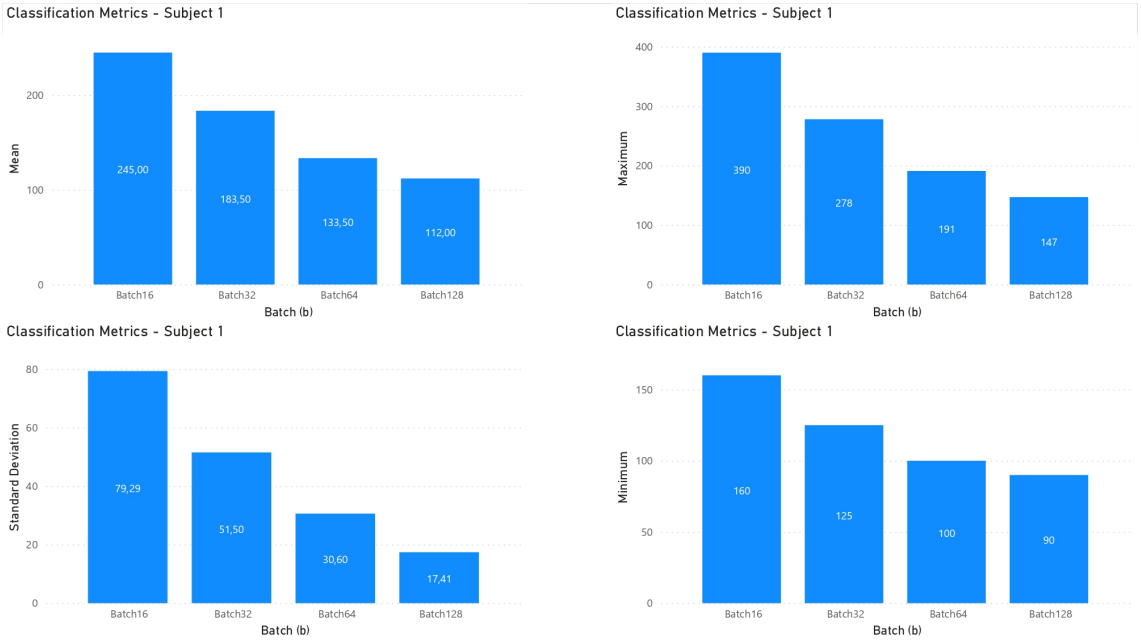


Figure 112: Runtime sort of by Batch. Dataset filtered by Subject 1.

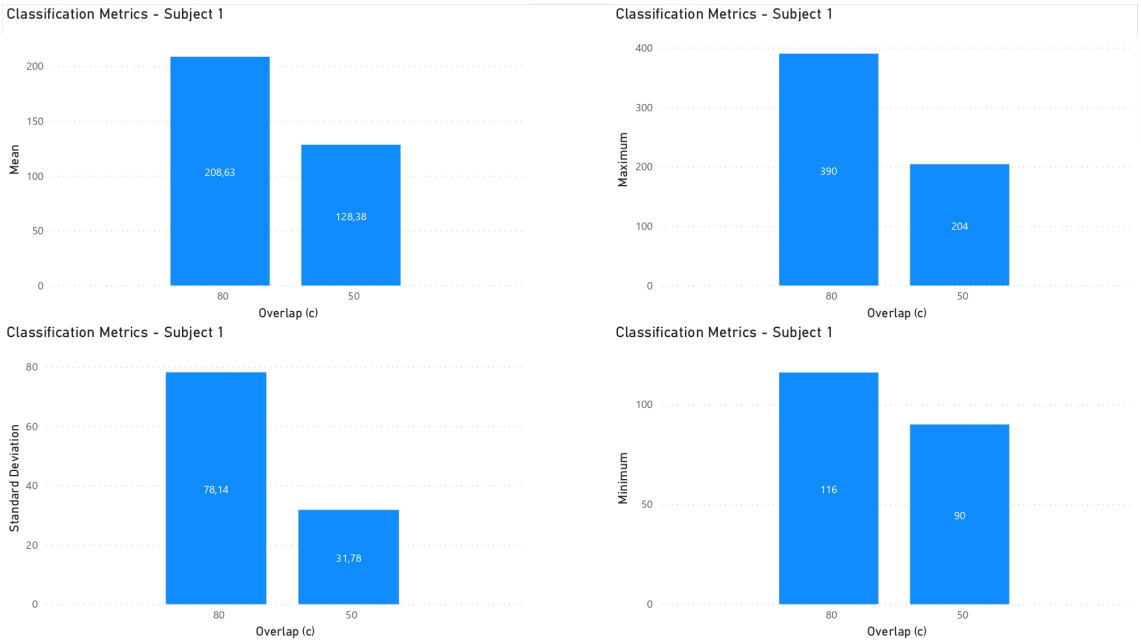


Figure 113: Runtime sort of by Overlap. Dataset filtered by Subject 1.

C.5 OUTCOMES FILTERED BY SUBJECT 2

Filter_by	ICs	Model	Batch	Overlap	Runtime	accuracy_test	F1_test	MCC_test	Prec_test	Recall_test	Specificty_test
≤2	16	gru	Batch16	80	295	89.58%	84.13%	NaN	86.32%	87.37%	89.58%
≤2	16	lstm	Batch16	80	298	91.15%	86.12%	NaN	87.54%	88.15%	91.15%
≤2	16	cnn	Batch16	80	288	90.10%	88.94%	81.77%	90.81%	91.65%	90.10%
≤2	16	cnn-lstm	Batch16	80	393	92.71%	91.52%	86.96%	93.33%	94.34%	92.71%
≤2	16	gru	Batch16	50	164	90.00%	88.64%	80.07%	91.67%	88.75%	90.00%
≤2	16	lstm	Batch16	50	164	88.75%	87.27%	77.85%	90.34%	87.92%	88.75%
≤2	16	cnn	Batch16	50	161	91.25%	90.40%	83.49%	92.05%	91.67%	91.25%
≤2	16	cnn-lstm	Batch16	50	206	83.75%	83.21%	69.34%	83.72%	85.75%	83.75%
≤2	16	gru	Batch32	80	219	88.02%	86.63%	77.14%	90.44%	87.00%	88.02%
≤2	16	lstm	Batch32	80	220	88.54%	87.24%	77.86%	89.79%	88.36%	88.54%
≤2	16	cnn	Batch32	80	202	87.50%	86.55%	76.09%	87.66%	88.54%	87.50%
≤2	16	cnn-lstm	Batch32	80	280	94.79%	94.46%	89.91%	94.70%	95.27%	94.79%
≤2	16	gru	Batch32	50	130	90.00%	88.99%	79.65%	91.43%	88.42%	90.00%
≤2	16	lstm	Batch32	50	128	86.25%	84.79%	71.54%	87.20%	84.50%	86.25%
≤2	16	cnn	Batch32	50	123	90.00%	89.09%	80.60%	92.85%	87.92%	90.00%
≤2	16	cnn-lstm	Batch32	50	157	91.25%	90.26%	82.12%	92.89%	89.42%	91.25%
≤2	16	gru	Batch64	80	151	89.06%	87.80%	77.60%	90.24%	87.54%	89.06%
≤2	16	lstm	Batch64	80	153	89.58%	88.71%	78.24%	89.29%	89.00%	89.58%
≤2	16	cnn	Batch64	80	153	89.58%	88.28%	78.87%	92.10%	86.96%	89.58%
≤2	16	cnn-lstm	Batch64	80	192	89.06%	88.63%	80.32%	89.69%	90.73%	89.06%
≤2	16	gru	Batch64	50	101	90.00%	89.14%	78.44%	89.78%	88.67%	90.00%
≤2	16	lstm	Batch64	50	104	87.50%	86.38%	73.98%	88.55%	85.50%	87.50%
≤2	16	cnn	Batch64	50	103	86.25%	84.77%	72.35%	88.96%	83.58%	86.25%
≤2	16	cnn-lstm	Batch64	50	123	91.25%	90.40%	81.18%	91.54%	89.67%	91.25%
≤2	16	gru	Batch128	80	118	92.71%	91.99%	84.21%	92.86%	91.36%	92.71%
≤2	16	lstm	Batch128	80	117	92.71%	92.05%	84.27%	92.58%	91.70%	92.71%
≤2	16	cnn	Batch128	80	122	91.15%	90.10%	81.63%	93.15%	88.61%	91.15%
≤2	16	cnn-lstm	Batch128	80	147	91.15%	90.53%	81.11%	90.38%	90.73%	91.15%
≤2	16	gru	Batch128	50	91	85.00%	84.00%	68.47%	85.16%	83.33%	85.00%
≤2	16	lstm	Batch128	50	92	86.25%	85.43%	71.12%	86.24%	84.90%	86.25%
≤2	16	cnn	Batch128	50	92	82.50%	80.06%	64.82%	86.67%	78.65%	82.50%
≤2	16	cnn-lstm	Batch128	50	110	88.75%	88.44%	77.15%	88.10%	89.06%	88.75%

Figure 114: Classification metrics results of dataset filtered by Subject 2

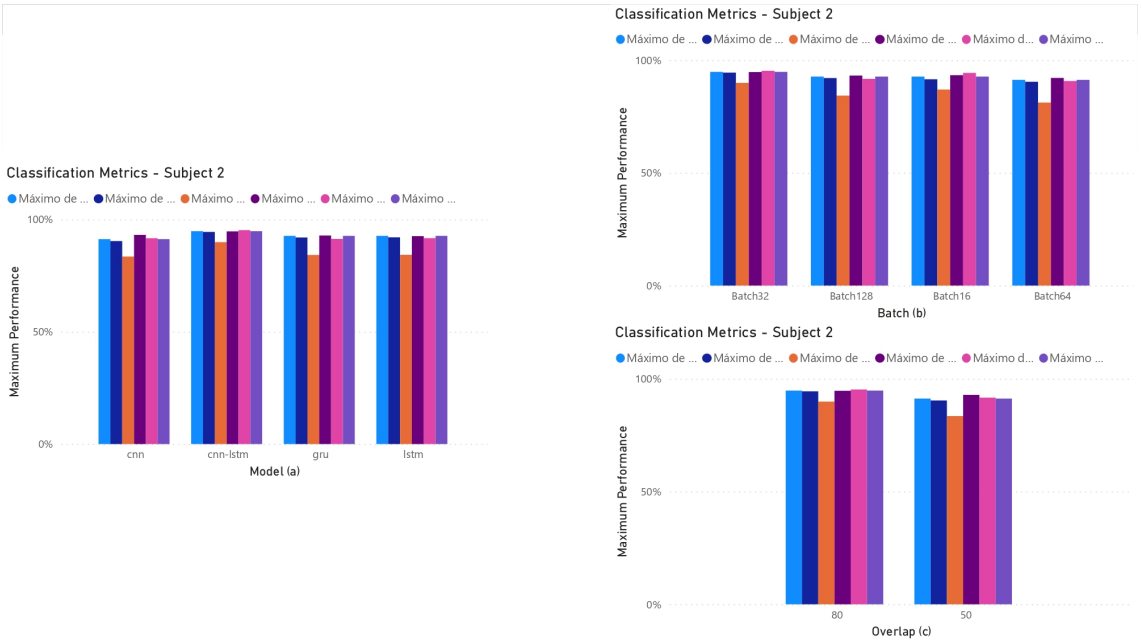


Figure 115: Classification Metrics Graphics sort of by (a) Models, (b) Batch and (c) Overlap. Dataset filtered by Subject 2.

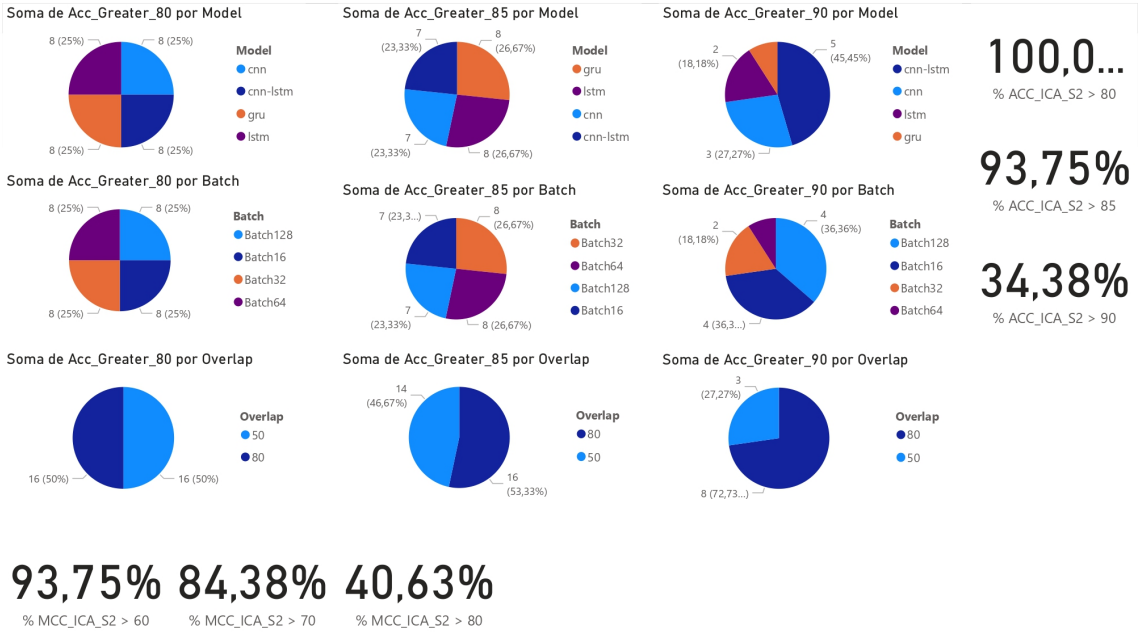


Figure 116: Accuracy greater 80% and 85% sort of by Model, Batch and Overlap. Overall accuracy greater 80% and 85%. Overall MCC greater 70% and 80%. Dataset filtered by subject 2.

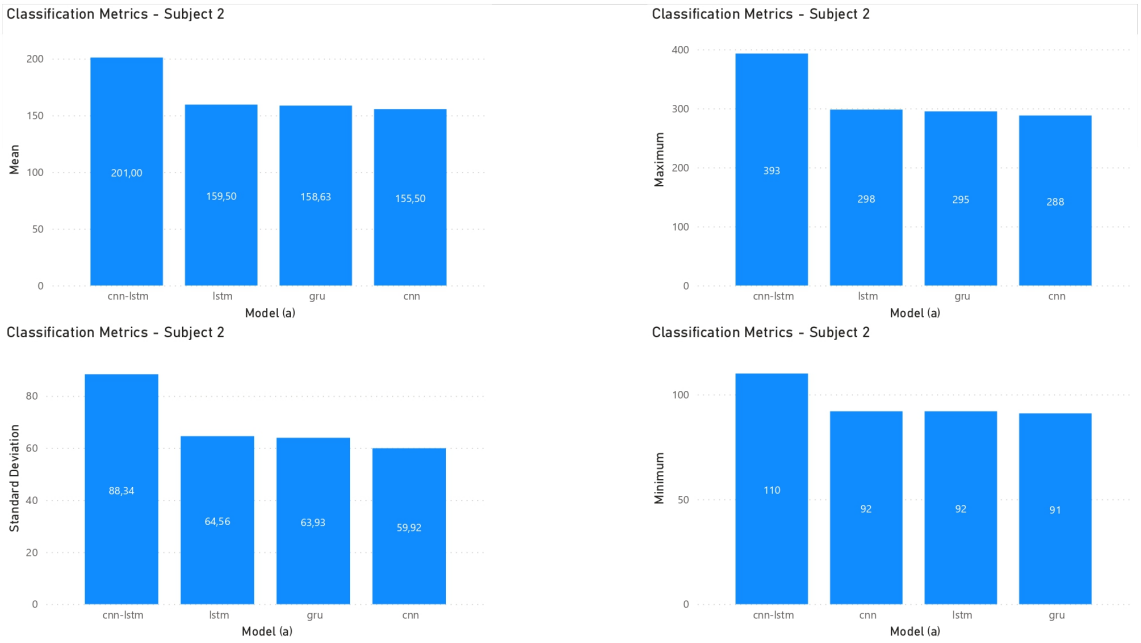


Figure 117: Runtime sort of by Model. Dataset filtered by Subject 2.

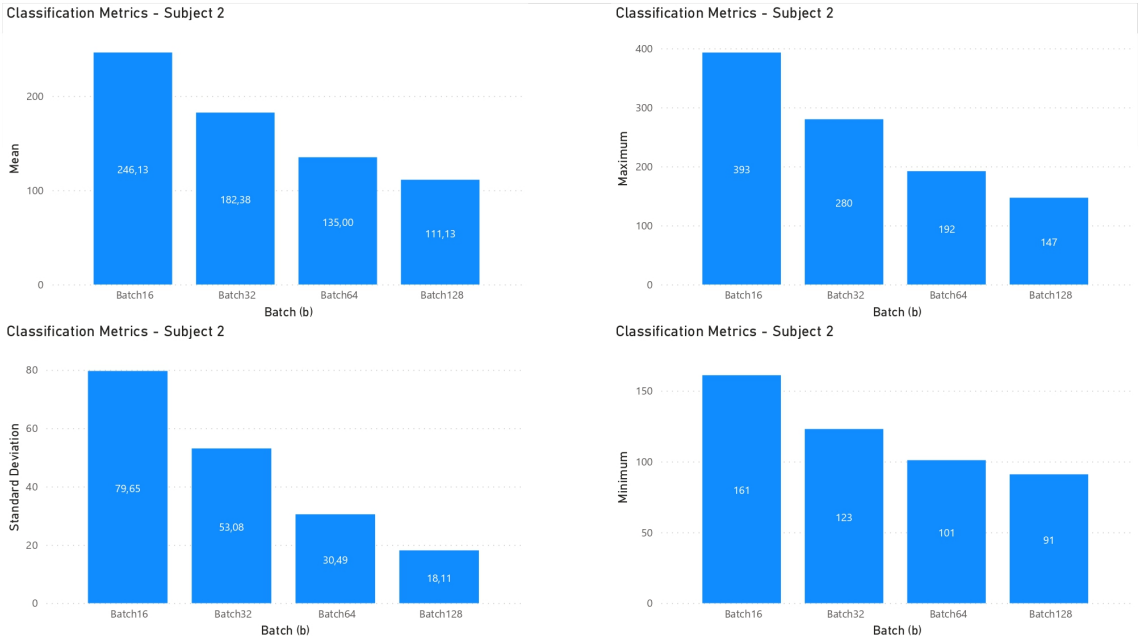


Figure 118: Runtime sort of by Batch. Dataset filtered by Subject 2.

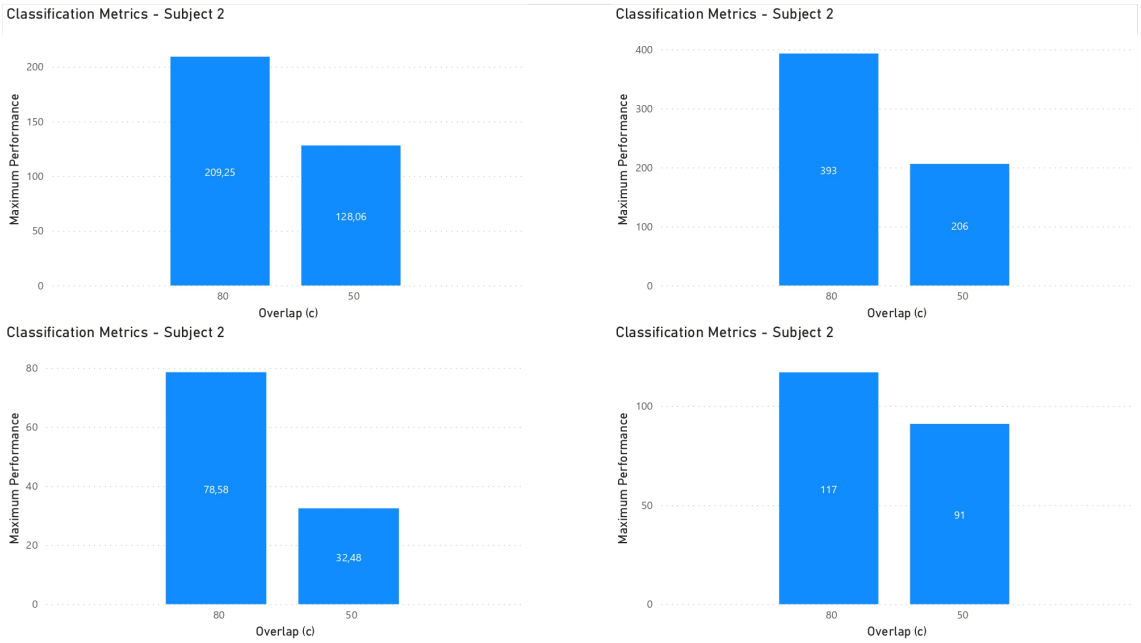
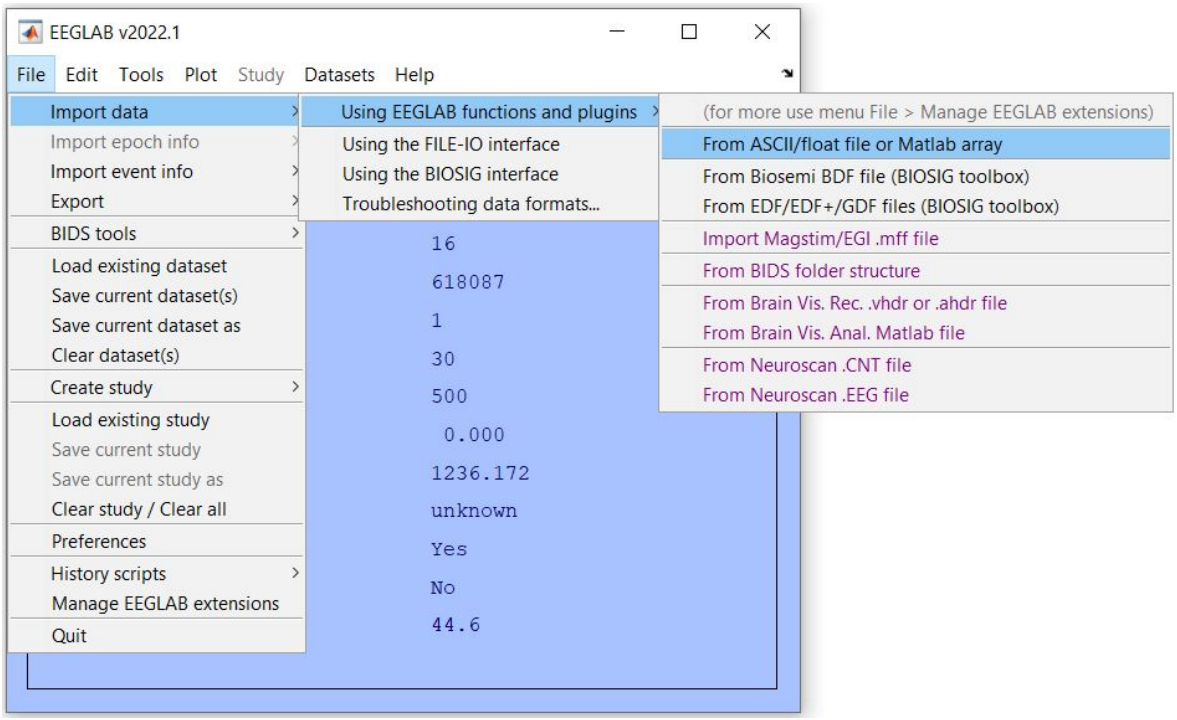


Figure 119: Runtime sort of by Overlap. Dataset filtered by Subject 2.

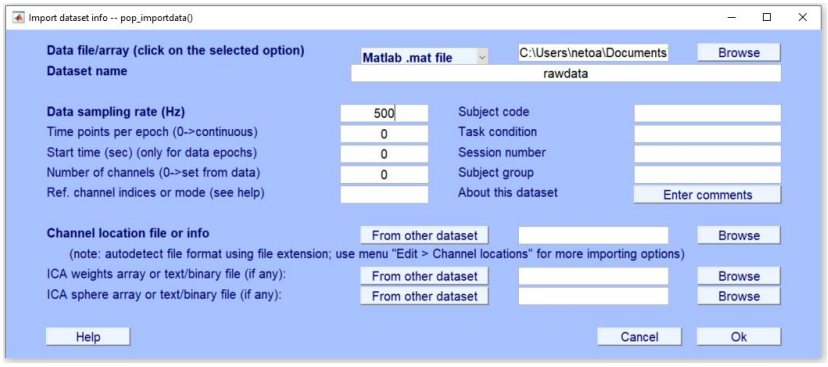
D

EEGLAB BASIC USER GUIDE



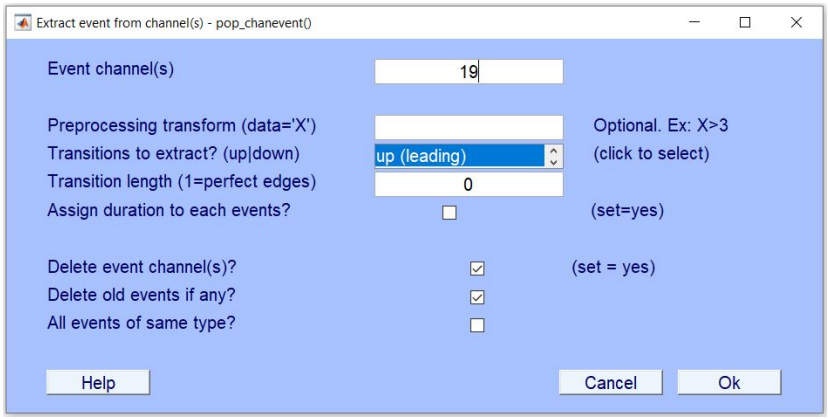
Type "EEGLAB" on *Command Window* in Matlab. The EEGLAB windows will pop-up, and then click on *Import Data* to upload the dataset.

Figure 120: Import data



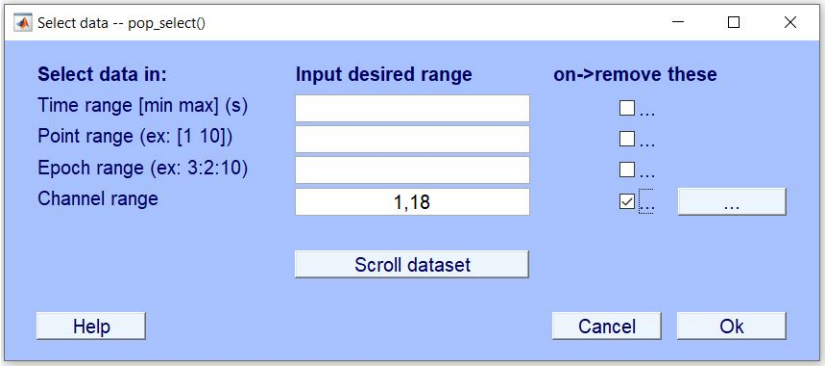
In data file, select the file format. Click on *Browse* to select the file in PC folders. Type the name for the dataset and then type the data sammpling rate in Hertz (Hz).

Figure 121: Import data info



To import event info, the user can run one of the following commands. Pop-up will open a new window and then type the row's number from dataset on *Event channels* field.
Pop-Up GUI: EEG = pop_chanevent(EEG);
or
Command: EEG = pop_chanevent(EEG, 19, 'edge', 'leading', 'edgelen', 0);

Figure 122: Import Event Info



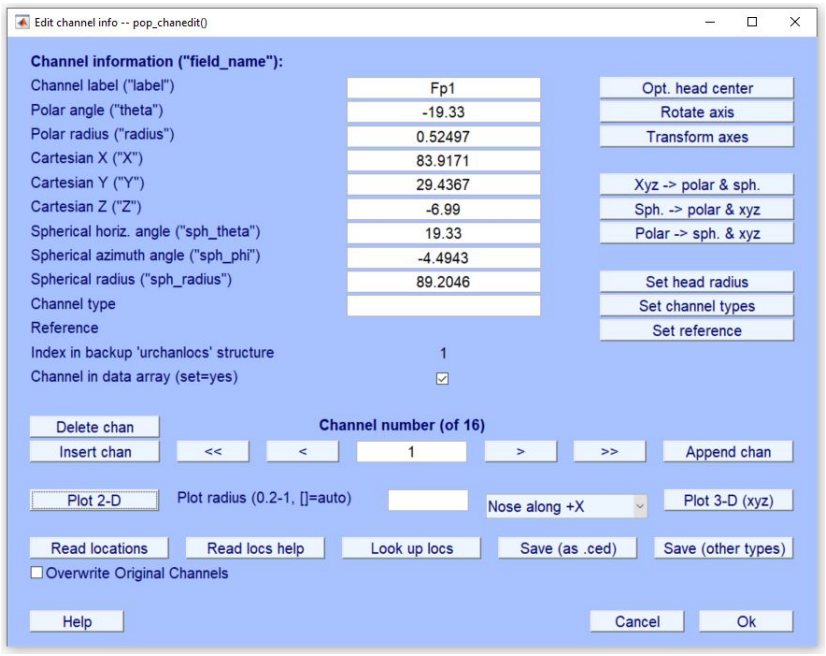
To import delete a channel, the user can run one of the following commands. Pop-up will open a new window. Type the row's number which will de deleted from dataset on *Channel range* field and then check the box regarding *remove these*

Pop-Up GUI: EEG = pop_select(EEG);

or

Command: EEG = pop_select(EEG, 'nochannel', [1,18]);

Figure 123: Delete Rows - (Row 1 and 18)



To import channel locations, create a variable which one will be used to save the channels' name and the run the command to save them on data info.

Create a variable: chanlocs = struct('labels', 'Fp1', 'Fp2', 'F3', 'Fz', 'F4', 'T7', 'C3', 'Cz', 'C4', 'T8', 'P3', 'Pz', 'P4', 'Po7', 'Po8', 'Oz');

Run the Command: EEG.chanlocs = pop_chanedit(chanlocs);

Figure 124: Insert channel locations

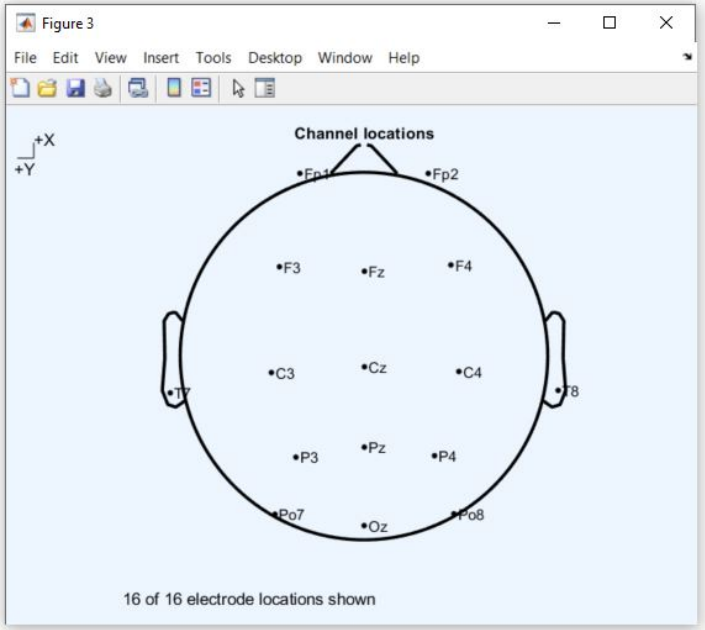


Figure 125: Plot - Channel locations (16 channels)

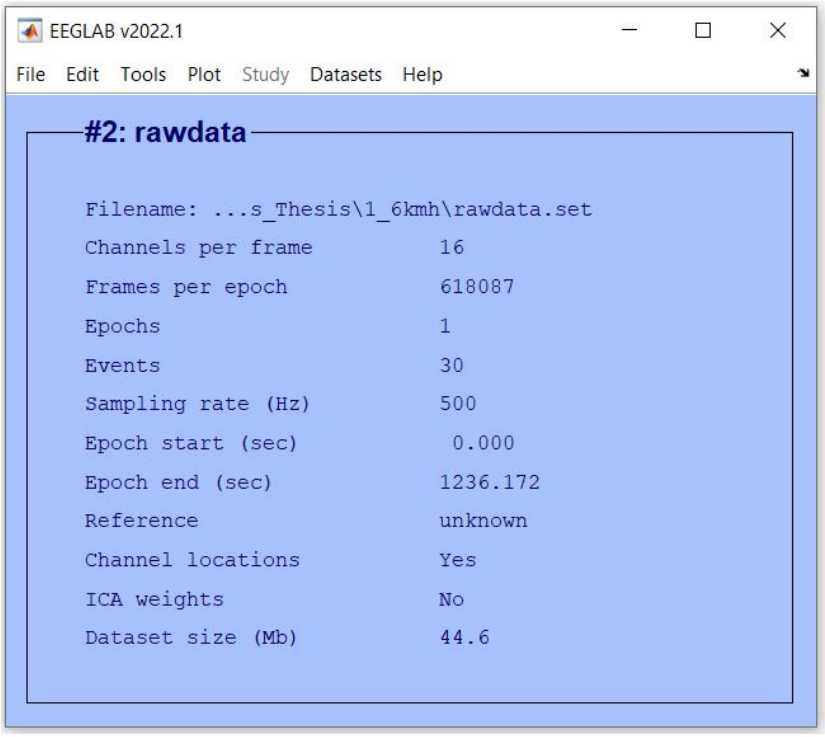
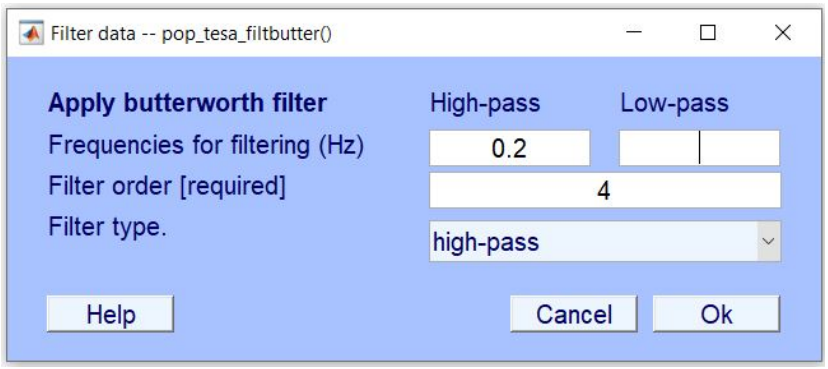
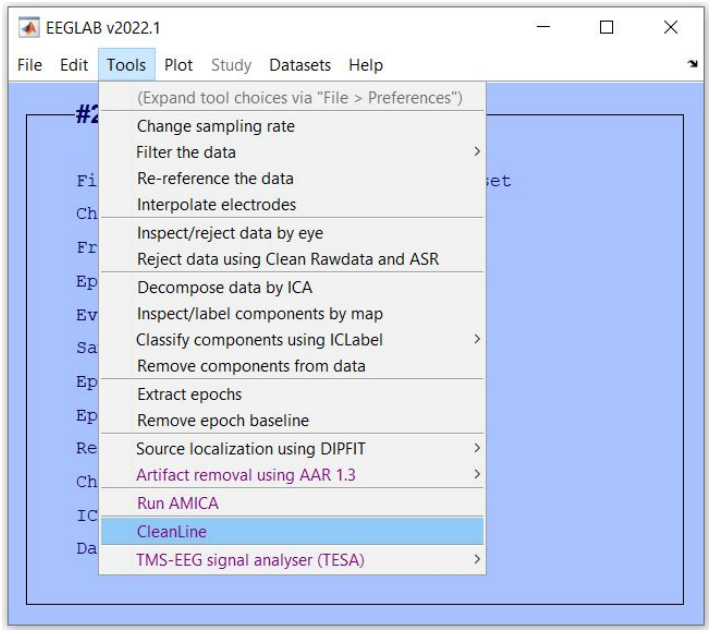


Figure 126: Status overview



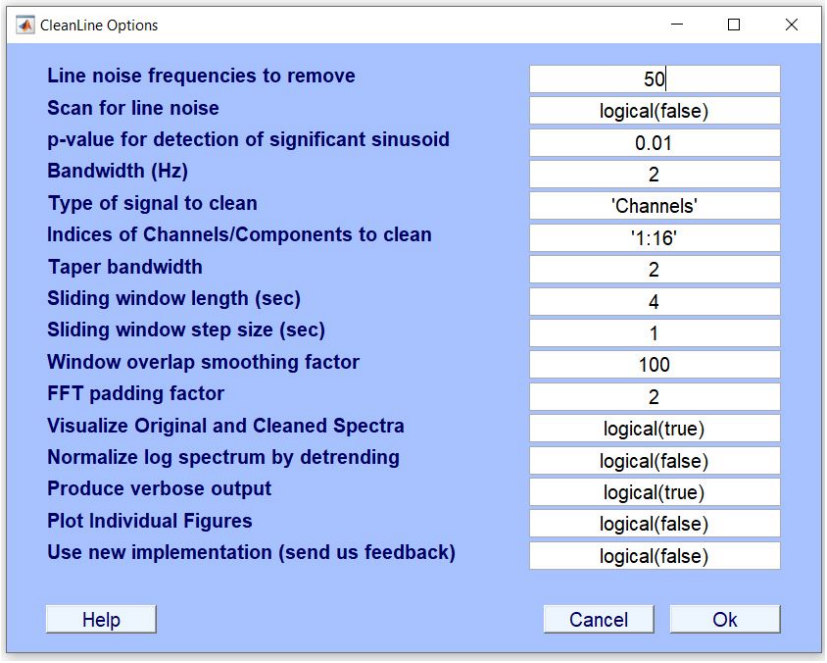
Apply high-pass filter using a fourth-order zero-phase Butterworth filter at 0.02Hz Pop-up will open a new window. On *Frequencies for filtering (Hz)* field fill it with 0.2 (High-pass) and EMPTY (Low-pass); on *File order* type 4 (fourth-order); and then select the filter type (high-pass).
Pop-Up GUI: EEG = pop_tesa_filtbutter(EEG);
or
Command: EEG = pop_tesa_filtbutter(EEG, 0.2, [], 4, 'highpass');

Figure 127: Filter data: High-pass



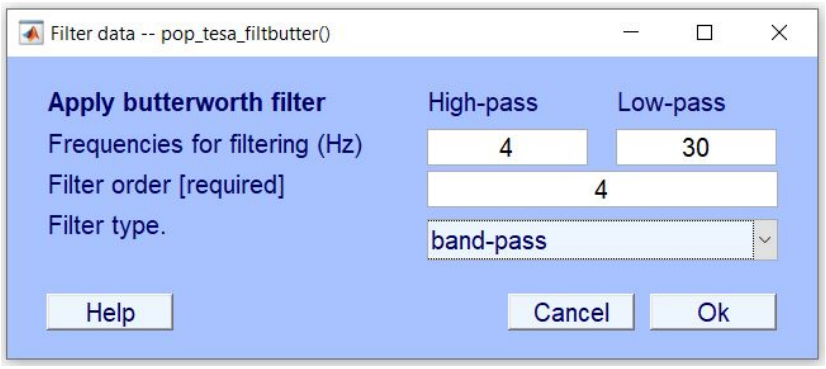
Remove line noise. Click on Tools -> Cleanline.

Figure 128: CleanLine



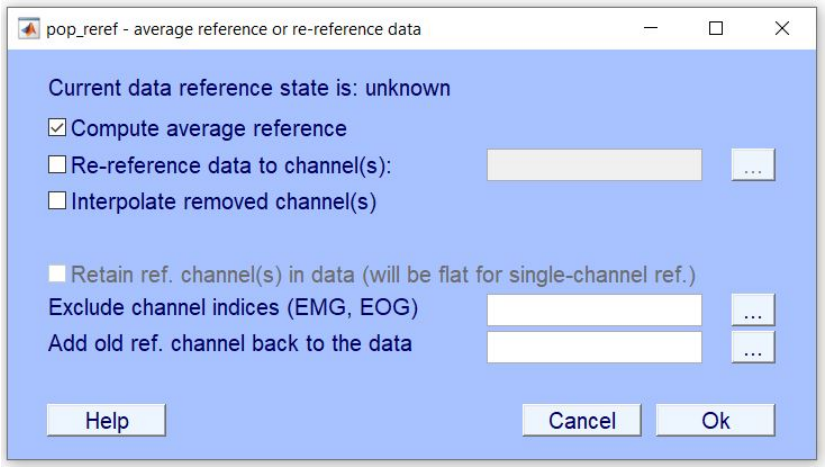
On *Line noise frequencies to remove* type "50" (50Hz)

Figure 129: CleanLine Options Window



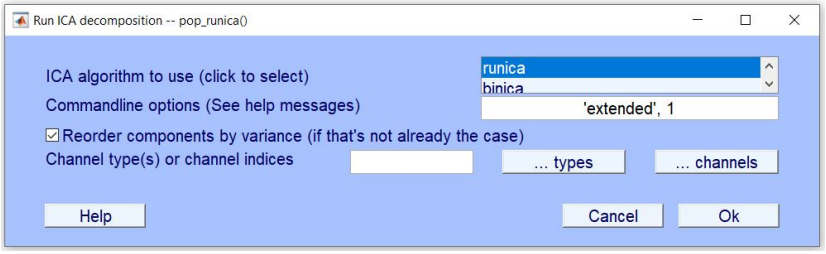
Apply band-pass filter using a fourth-order zero-phase Butterworth filter between 4Hz and 30Hz. Pop-up will open a new window. On *Frequencies for filtering (Hz)* field fill it with 4 (High-pass) and 30 (Low-pass); on *File order* type 4 (fourth-order); and then select the filter type (band-pass).
Pop-Up GUI: EEG = pop_tesa_filtbutter(EEG);
or
Command: EEG = pop_tesa_filtbutter(EEG, 4, 30, 4, 'bandpass');

Figure 130: Filter the data: Band-pass



Reference the channels to the common average. The user can run one of the following commands. Pop-up will open a new window and then check the box *Compute average reference*.
Pop-Up GUI: EEG = pop_reref(EEG);
or
Command: EEG = pop_reref(EEG, []);

Figure 131: Re-reference the data



Apply Independent Component Analysis (ICA) to decompose data into independent sources. Pop-up will open a new window and then select *runica* algorithm.
Pop-Up GUI: EEG = pop_runica(EEG);
or
Command: EEG = pop_runica(EEG, 'icatype', 'runica');

Figure 132: Decompose data by ICA

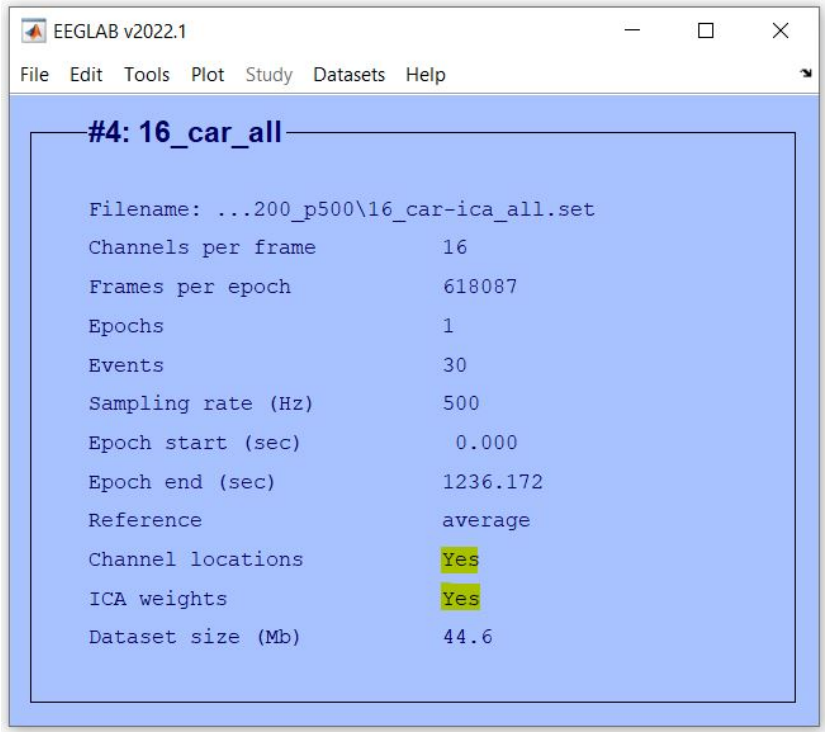
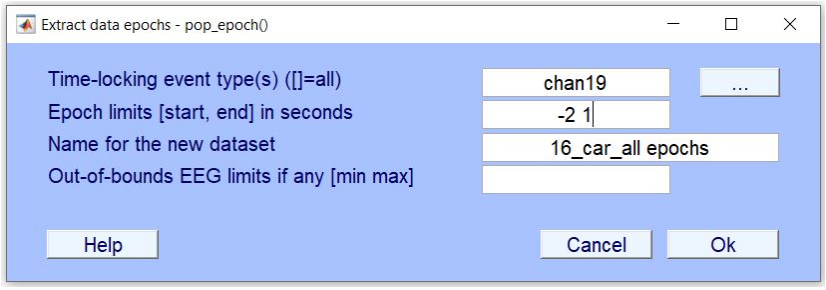


Figure 133: Status overview - Updated CAR and ICA



To open the Extract data epoch pop-up run the following command. Pop-up window will show up. Type the event name in *Time-locking event types*; type de *Epoch limits in seconds*, for exempla, -2 1; and the name the new dataset.
Pop-Up GUI: EEG = pop_epoch(EEG)

Figure 134: Extract data Epochs

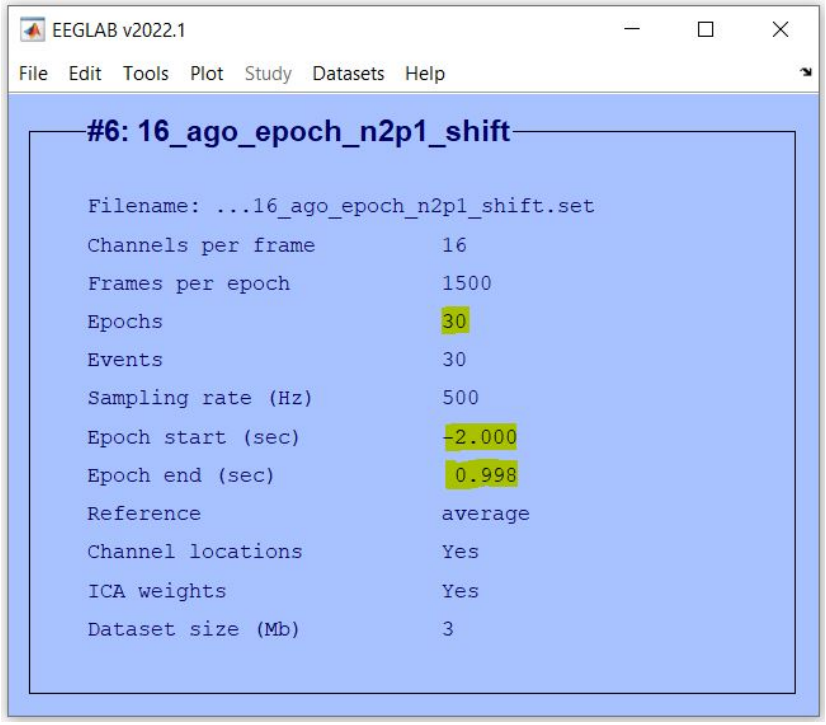
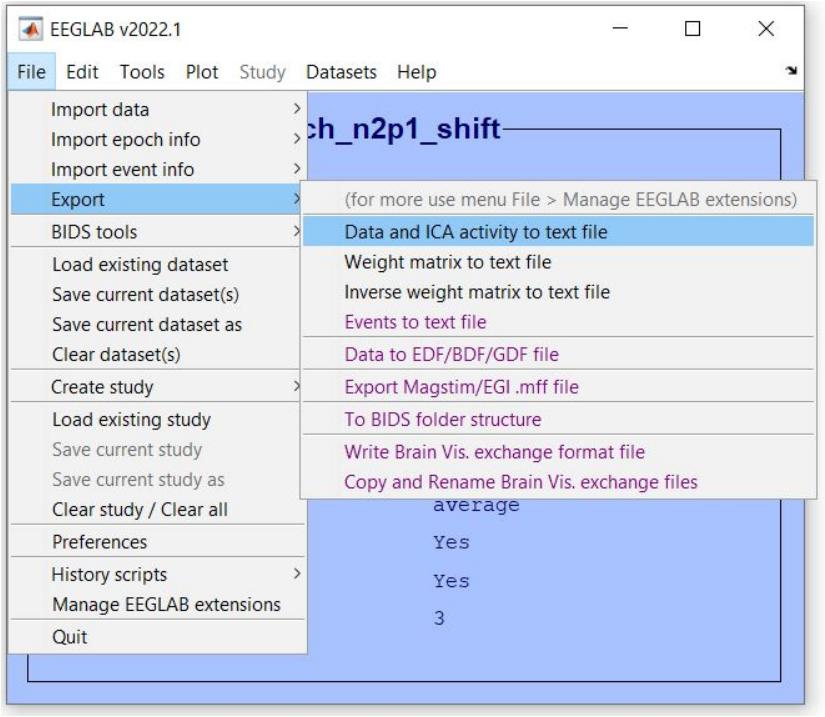
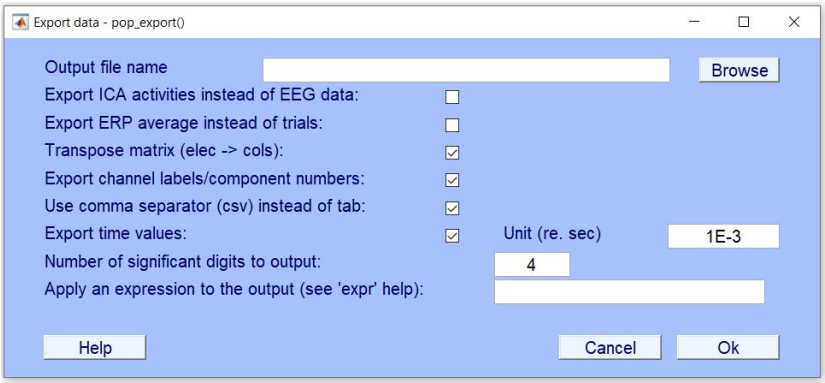


Figure 135: Status overview - Epoch



To export the pre-processed data click on *Files -> Export -> Data and ICA activity to text file*

Figure 136: Export data



Select the options that you prefer to export. Click on "Ok" and then name the file and save it.

Figure 137: Export data - Pop Up