1 2 90

UNIVERSIDADE Đ
COIMBRA

Vasco Guimarães Paiva

EEG Pre-Processing Using Deep Learning Networks: An Unichannel Approach

Thesis submitted to the Faculty of Science and Technology of the University of Coimbra for the degree of Master in Biomedical Engineering with specialization in Clinical Informatics and Bioinformatics, supervised by Prof. Dr. César Alexandre Domingues Teixeira and M. Sc. Fábio André da Costa Lopes.

September 2022

1290

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Vasco Guimarães Paiva

EEG Pre-Processing Using Deep Learning Networks: An Unichannel Approach

Thesis submitted to University of Coimbra for the degree of Master in Biomedical Engineering

Supervisors:
Prof. Dr. César Alexandre Domingues Teixeira (CISUC)
M. Sc. Fábio André da Costa Lopes (CISUC)

Coimbra, 2022

This work was developed in collaboration with:

CISUC - Center for Informatics and Systems of the University of Coimbra

This work was supported by the Portuguese Foundation for Science and Technology through projects CISUC (UID/CEC/00326/2020) and RECoD (PTDC/EEI-EEE/5788/2020).

Agradecimentos

A conclusão desta importante etapa não teria sido possível sem o apoio, auxílio e paciência de inúmeras pessoas ao longo dos últimos cinco anos. A todas elas gostaria de expressar o meu profundo agradecimento.

Ao Professor Doutor César Alexandre Domingues Teixeira e ao Mestre Fábio André da Costa Lopes, agradeço, antes de mais, a confiança que em mim depositaram. Um enorme obrigado por todos os vossos ensinamentos e, ainda, por toda a ajuda e mentoria, pela vossa dedicação e paciência e, acima de tudo, pela infindável disponibilidade que sempre demonstraram ao longo deste longo trabalho.

Aos amigos que Coimbra me deu, obrigado por todas as conversas, gargalhadas e momentos partilhados, por terem feito parte deste meu crescimento, tanto a nível pessoal como académico, e por tornarem inesquecíveis estes cinco anos na Cidade dos Estudantes. Ao vosso lado, vivi uma das mais importantes fases, que me ficará para sempre marcada em memórias que só nós recordaremos. Agradeço também aos meus amigos, em São João da Madeira, que embora muitas vezes longe, continuaram sempre dispostos a dar uma palavra de apoio e carinho.

À minha família, em especial à minha irmã e aos meus pais, por estarem sempre presentes, não só para celebrar as minhas conquistas como para ajudar em qualquer momento ou situação, em tudo o que fosse preciso e mais ainda. Um obrigado gigante por todos os sacrifícios por que passaram para me proporcionar esta oportunidade e por nunca deixarem de acreditar em mim, nem mesmo quando eu próprio não acredito. Tudo aquilo que atingi, devo-o a vocês.

Abstract

Electroencephalograms (EEGs) are non-linear and non-stationary biosignals that measure electric activity in the brain, and are very useful in the diagnosis and monitorization of different neurological disorders. In long term EEG acquisitions, the recordings are affected by different types of artifacts, which affect the quality of the signals and limit their use in clinical research. Although a variety of techniques have been developed to remove the noise from EEG signals, the majority of them offer only partial removal of noisy artifacts or are too computationally expensive to be used in real-time scenarios.

The automatic learning of complex characteristics and noise removal capabilities, as well as their capabilities to work in real-time over data streams, make Deep Learning Neural Networks (DLNNs) a promising alternative to EEG signal pre-processing. However, algorithms that are trained using multiple channels can only be applied to recordings that use the same exact channels. Thus, the aim of this study was the development of automatic single-channel (SC) Deep Learning (DL) architectures capable of removing noisy artifacts and reconstruct clean EEG signals.

Two approaches were followed in the construction and training of the architectures. The first approach utilized 10-minute segments from patients with epilepsy in the EPILEPSIAE database to train and test a simple Deep Convolutional Autoencoder (DCAE), which was compared to a previously trained multi-channel (MC) Deep Convolutional Neural Network (DCNN). The second approach used the same signals, but divided into 5-second windows. A DCAE with an incorporated BiLSTM layer (DCLSTMAE) was trained, along with two other SC architectures retrieved from literature - an One-Dimensional Residual Convolutional Neural Network (1D-ResCNN) and a DCNN. DCLSTMAE's performance was compared to the other trained SC models, as well as the results from the previous approach.

Results showed that the models from the two approaches behaved similarly, being capable of reducing the noise from different types of artifacts in a fast and automatic manner. They outperform the other SC models, obtaining smaller reconstruction errors and higher correlation to the target segments, but still present limitations in signal reconstruction when compared to MC algorithms. However, this research further cements the potential of automatic SC DL models to be used in EEG signal pre-processing without acquisition limitations and paves the way for future works to explore different models and training setups in order to address the limitations that were encountered.

Keywords: Electroencephalogram, Artifact Removal, Signal Reconstruction, Pre-processing, Deep Learning, Neural Networks.

Resumo

Electroencefalogramas (EEGs) são biossinais não lineares e não estacionários que medem a atividade elétrica no cérebro e são muito úteis no diagnóstico e monitorização de diferentes distúrbios neurológicos. Nas aquisições de EEG de longa duração, os sinais são afetados por diferentes tipos de artefactos, que contaminam a qualidade dos sinais e limitam o seu uso em investigação clínica. Embora várias técnicas tenham já sido desenvolvidas para remover o ruído de sinais de EEG, a maioria garante apenas uma remoção parcial dos artefatos ruidosos, ou são demasiado computacionalmente pesadas para ser utilizadas em tempo real.

As capacidades de aprendizagem automática de características complexas e de remoção de ruído, bem como a a aplicação rápida, tornam as redes neuronais de Deep Learning (DL) uma alternativa promissora no pré-processamento de sinais de EEG. No entanto, os algoritmos multicanal, treinados utilizando vários canais, só podem ser aplicados a aquisições de EEG com exatamente os mesmos canais. Assim, o objetivo deste estudo foi o desenvolvimento de arquiteturas de DL automáticas e unicanal, capazes de remover artefatos ruidosos e reconstruir sinais de EEG limpos.

Duas abordagens foram seguidas na construção e treino de arquiteturas. Na primeira abordagem, foram usados segmentos de 10 minutos de doentes com epilepsia do repositório EPILEPSIAE no treino e teste de um simples Deep Convolutional Autoencoder (DCAE), que foi comparado a uma Deep Convolutional Neural Network (DCNN) multicanal treinada anteriormente com os mesmos dados. A segunda abordagem utilizou os mesmos sinais, mas divididos em janelas de 5 segundos. Foi treinado um Deep Convolutional Autoencoder ao qual foi incorporado uma camada LSTM Bidirecional (DCLSTMAE), juntamente com duas outras arquiteturas unicanal retiradas da literatura - uma One-Dimensional Residual Convolutional Neural Network (1D-ResCNN) e uma DCNN. O desempenho do DCLSTAME foi comparado com o dos outros modelos unicanal treinados, bem como os resultados da abordagem anterior.

Os resultados mostraram que os modelos das duas abordagens apresentam comportamentos semelhantes, conseguindo reduzir os níveis de ruído de diferentes tipos de artefactos de forma rápida e automática. O seu desempenho é superior aos outros modelos unicanal, obtendo erros de reconstrução menores e níveis de correlação maiores com os segmentos alvo, mas ainda apresentam algumas limitações quando comparados aos algoritmos multicanal. No entanto, esta pesquisa cimenta ainda mais o potencial dos modelos automáticos unicanal de DL para serem usados no pré-processamento de sinais EEG sem limitações na aquisição, e abre caminho para trabalhos futuros de exploração de diferentes modelos e configurações de treino, de modo a solucionar as limitações encontradas.

Palavras-chave: Electroencefalograma, Remoção de Artefactos, Reconstrução de Sinais, Pré-processamento, Deep Learning, Redes Neuronais.

Contents

List of Figures **xiii**

List of Tables **xvi**

List of Abbreviations **xvii**

1 Introduction **1**

1.1 Motivation 1

1.2 Context 1

1.3 Goals and Expected Contributions 3

1.4 Structure 4

2 Background Concepts **5**

2.1 Electroencephalogram 5

2.1.1 Scalp Electroencephalogram 5

2.1.2 Intracranial Electroencephalogram 6

2.1.3 Electroencephalogram Montages 8

2.2 Neural Activity 8

2.3 Artifacts 9

2.3.1 Environmental Artifacts 10

2.3.2 Experimental Artifacts 11

2.3.3 Physiological Artifacts 11

2.4 Deep Learning Architectures 12

2.4.1 Feed Forward Neural Networks 13

2.4.2 Convolutional Neural Networks 14

2.4.3 Recurrent Neural Networks 14

2.4.4 Autoencoders 15

2.5 Evaluation Metrics 18

3 Related Work **20**

3.1 Linear Regression 20

3.2 Filtering Methods 21

3.2.1 Adaptive Filtering 21

3.2.2 Wiener Filtering 22

3.2.3 Bayes Filtering 22

3.3 Empirical Model Decomposition 22

3.4 Wavelet Decomposition 23

3.5 Blind Source Separation 23

3.5.1 Independent Component Analysis 24

3.5.2 Canonical-Correlation Analysis 25

3.5.3 Morphological Component Analysis 25

3.5.4 Automatic Blind Source Separation 26

3.6 Deep Learning Neural Networks 27

4 Experimental Setup **33**

4.1 Dataset 33

4.2 Methods 34

4.2.1 First Approach 36

4.2.1.1 Grid Search 36

4.2.1.2 Training and Testing 39

4.2.2 Second Approach 40

4.2.2.1 Grid Search 40

4.2.2.2 Training and Testing 41

5 Results and Discussion **44**

5.1 First Approach 44

5.1.1 Grid Search 44

5.1.2 Model Testing 45

5.1.2.1 Evaluation Metrics 45

5.1.2.2 Artifact Removal and Signal Reconstruction 53

5.2 Second Approach 54

5.2.1 Grid Search 54

5.2.2 Model Testing 67

5.2.2.1 Evaluation Metrics 67

5.2.2.2 Artifact Removal and Signal Reconstruction 77

6 Conclusion **90**

Bibliography **92**

List of Figures

2.1 Synaptic potentials measured in an Electroencephalogram (EEG) . . 6
2.2 Systems for electrode placement in EEG acquisitions 7
2.3 Types of iEEG electrodes 7
2.4 Common EEG montages 9
2.5 Types of brain waves 10
2.6 Amplitudes and Power Spectral Densities (PSDs) of signals with different kinds of artifacts 12
2.7 Representation of a Feed Forward Neural Network (FFNN) 13
2.8 Representation of a Convolutional Neural Network (CNN) 15
2.9 Representation of Recurrent Neural Networks (RNNs) 16
2.10 Representation of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks 16
2.11 Representation of an Autoencoder (AE) architecture 17
2.12 Regularized AEs 18

4.1 Schematic representation of developed approaches. 37
4.2 Architectures developed during the first approach 38
4.3 Deep Learning (DL) architecture trained using the full training dataset in the first approach. 39
4.4 Architectures developed during the second approach 42
4.5 DL architecture trained using the full training dataset in the second approach. 43

5.1 Root Mean Squared Error (RMSE) and Relative Root Mean Squared Error (RRMSE) mean values for all models developed in the grid search. 47
5.2 Mean values for the architectures with 8 and 16 filters developed in the grid search. 48
5.3 Mean values for the architectures with 32 and 64 filters developed in the grid search. 49

5.4 First approach RMSE performance results 51
5.5 First approach RRMSE performance results 51
5.6 First approach Pearson Correlation Coefficient (PCC) performance results 52
5.7 First approach Signal-to-Noise Ratio Difference (SNR$_{Diff}$) performance results 52
5.8 Example EEG segment from the test set containing eye blink artifact. 55
5.9 Example EEG segment from the test set containing eye movement artifact. 56
5.10 Example EEG segment from the test set containing muscle artifact. . 57
5.11 Example EEG segment from the test set containing pulse artifact. . . 58
5.12 Example EEG segment from the test set containing cardiac artifact. . 59
5.13 RMSE values for architectures with 3 convolutional layers before the output layer of the latent space. 63
5.14 RRMSE values for architectures with 3 convolutional layers before the output layer of the latent space. 63
5.15 RMSE values for architectures with 4 convolutional layers before the output layer of the latent space. 64
5.16 RRMSE values for architectures with 4 convolutional layers before the output layer of the latent space. 64
5.17 RMSE values for architectures with 5 convolutional layers before the output layer of the latent space. 65
5.18 RRMSE values for architectures with 5 convolutional layers before the output layer of the latent space. 65
5.19 RMSE mean values for all architectures developed in the grid search. 66
5.20 RRMSE mean values for all architectures developed in the grid search. 66
5.21 Second approach RMSE performance results 69
5.22 Second approach RRMSE performance results 69
5.23 Second approach PCC performance results 70
5.24 Second approach SNR$_{Diff}$ performance results 70
5.25 RMSE performance results comparing both approaches 72
5.26 RRMSE performance results comparing both approaches 72
5.27 PCC performance results comparing both approaches 73
5.28 SNR$_{Diff}$ performance results comparing both approaches 73
5.29 Distributions for RMSE, RRMSE and PCC metrics for channel O2 segments 75

5.30 Zoomed in line graphs representing the different PCC and RMSE values for each 10-minute segment in the test dataset. 76
5.31 Example of segment with very different PCC values. 77
5.32 Example EEG segment from the test set containing eye blink artifact comparing the three architectures developed in the second approach. . 79
5.33 Example EEG segment from the test set containing eye blink artifact comparing the DCAE with an incorporated BiLSTM layer (DCLSTMAE) to the Deep Convolutional Autoencoder (DCAE) model from the first approach. 80
5.34 Example EEG segment from the test set containing eye movement artifact comparing the three architectures developed in the second approach. 81
5.35 Example EEG segment from the test set containing eye movement artifact comparing the DCLSTMAE to the DCAE model from the first approach. 82
5.36 Example EEG segment from the test set containing muscle activity comparing the three architectures developed in the second approach. . 83
5.37 Example EEG segment from the test set containing muscle activity comparing the DCLSTMAE to the DCAE model from the first approach. 84
5.38 Example EEG segment from the test set containing pulse artifact comparing the three architectures developed in the second approach. . 85
5.39 Example EEG segment from the test set containing pulse artifact comparing the DCLSTMAE to the DCAE model from the first approach. 86
5.40 Example EEG segment from the test set containing cardiac artifact comparing the three architectures developed in the second approach. . 87
5.41 Example EEG segment from the test set containing cardiac artifact comparing the DCLSTMAE to the DCAE model from the first approach. 88

List of Tables

3.1 EEG Pre-Processing Models using Automatic Blind Source Separation (BSS) . . . 27
3.2 EEG Pre-Processing Models using Neural Networks . . . 31
3.3 Performances of EEG Pre-Processing Models . . . 32
3.4 Comparison of the Advantages and Disadvantages of various Artifact Removal Methods . . . 32

4.1 EPILEPSIAE Database Recordings . . . 33
4.2 Patient's Data . . . 35
4.3 Model Training Hyper-Parameters . . . 40

5.1 RMSE results for Deep Convolutional Neural Networks (DCNNs) and DCAEs in first approach Grid Search . . . 46
5.2 RRMSE results for DCNNs and DCAEs in first approach Grid Search 46
5.3 First Approach Grid Search - Best Model Parameters . . . 50
5.4 RMSE results for DCAEs and DCLSTMAEs in second approach Grid Search . . . 60
5.5 RRMSE results for DCAEs and DCLSTMAEs in second approach Grid Search . . . 61
5.6 Second Approach Grid Search - Best Model Parameters . . . 67

List of Abbreviations

1D-CNN One-Dimensional Convolutional Neural Network.
1D-ResCNN One-Dimensional Residual Convolutional Neural Network.

ADJUST Automatic EEG artifact Detection based on the Joint Use of Spatial and Temporal feature.
AE Autoencoder.
AMICA Adaptative Mixture Independent Component Analysis.
ANC Adaptive Noise Cancellation.
ANN Artificial Neural Network.
ApEn Approximate Entropy.

BiLSTM Bidirectional LSTM.
BN Batch Normalization.
BSS Blind Source Separation.

CAE Convolutional Autoencoder.
CC Correlation Coefficient.
CCA Canonical-Correlation Analysis.
CNN Convolutional Neural Network.
CSED Cumulative Squared Euclidean Distance.

DAE Denoising Autoencoder.
DC Direct Current.
DCAE Deep Convolutional Autoencoder.
DCLSTMAE DCAE with an incorporated BiLSTM layer.
DCNN Deep Convolutional Neural Network.
DL Deep Learning.
DLNN Deep Learning Neural Network.

ECD Equivalent Current Dipole.
ECG Electrocardiogram.

ECoG Electrocorticogram.
EEG Electroencephalogram.
EEMD Ensemble Empirical Model Decomposition.
EMD Empirical Model Decomposition.
EMG Electromyography.
EOG Electrooculogram.

FASTER Fully Automated Statistical Thresholding for EEG artifact Rejection.
FFNN Feed Forward Neural Network.
FIR Finite Impulse Response.

GRU Gated Recurrent Unit.
GRU-MARSC Gated Recurrent Unit based Multi-type Artifact Removal algorithm for Single-Channel.

IC Independent Component.
ICA Independent Component Analysis.
iEEG intracranial Electroencephalogram.
IIR Infinite Impulse Response.
IMF Intrinsic Mode Function.
IS International System.

K-ICA Kurtosis ICA.

LDA Linear Discriminant Analysis.
LMS Least Mean Square.
LPM Linear Programming Machine.
LSTM Long Short-Term Memory.

MAE Mean Absolute Error.
MAPE Mean Absolute Percentage Error.
MARA Multiple Artifact Rejection Algorithm.
MC multi-channel.
MCA Morphological Component Analysis.
MI Mutual Information.
ML Machine Learning.
MMNN Multi-Module Neural Network.
MSE Mean Squared Error.

NN Neural Network.

PCA Principal Component Analysis.
PCC Pearson Correlation Coefficient.
PDF Probability Density Function.
PSD Power Spectrum Density.
PSNR Peak Signal-to-Noise Ratio.

ReLU Rectified Linear Unit.
REST Reference Electrode Standardisation Technique.
RLDA Regularized Linear Discriminant Analysis.
RLS Recursive Least Square.
RMS Root Mean Square.
RMSE Root Mean Squared Error.
RNN Recurrent Neural Network.
RRMSE Relative Root Mean Squared Error.

S-RRMSE Spectral Relative Root Mean Square Error.
SAE Sparse Autoencoder.
SASICA Semi-Automated Selection of Independent Components of the electroencephalogram for Artifact correction.
SC single-channel.
SDR Source-to-Distortion Ratio.
SM-EFO Spider Monkey-based Electric Fish Optimization.
SNR Signal-to-Noise Ratio.
SNR$_{Diff}$ Signal-to-Noise Ratio Difference.
SOBI Second-Order Blind Identification.
SSAE Stacked Sparse Autoencoder.
SuBAR Surrogates-based Artifact Removal.
SVM Support-Vector Machine.

T-RRMSE Temporal Relative Root Mean Square Error.
TDSEP Temporal Decorrelation source SEParation.

1

Introduction

In this chapter, the motivation of this thesis is presented. Section 1.1. briefly introduces the overall theme of this project, with its context being explained in Section 1.2. Section 1.3 describes the main goals and contributions of the thesis, and Section 1.4 presents the structure of the remaining of this document.

1.1 Motivation

Electroencephalogram (EEG) is a non-linear and non-stationary biosignal that measures the electrical activity generated by the brain. Electrodes measure the sum of the potentials originated by the synapses in different brain cells [1]. Nowadays, EEG recordings are of great utility in the diagnosis and monitorization of many neurological disorders, such as epilepsy, sleep disturbances, encephalopathies and other mental disorders [2–4].

Several artifacts may rise in EEG acquisitions, especially in scalp acquisitions and in the ones that occur during longer periods of several hours in which patients maintain daily activities such as moving, sleeping and talking. The overlap with the neural signals hampers the analysis and usefulness of the recording, and specialists often have difficulties identifying these artifacts. As such, the development of a model capable of rapidly and effectively process the EEG signal, identifying and removing the different types of artifacts while maintaining all the relevant brain information, will facilitate and expand the use of EEG recordings in the assessment of several pathologies [5].

1.2 Context

EEG signals were firstly discovered, recorded and analysed during the 1930s. Since then, the evolution of this technique has been a continuous process. Today, EEGs are being recorded using fully computerized systems, with enough memory for long-term recordings of several hours. They are also equipped with many signal processing tools and accurate measurement electrodes. The varied applications of EEG recordings include [6]:

- monitoring alertness, coma, and brain death;
- locating areas of damage following head injury, stroke, and tumour;
- testing afferent pathways (by evoked potentials);
- monitoring cognitive engagement (alpha rhythm);
- producing biofeedback situations;
- controlling anaesthesia depth (servo anaesthesia);
- investigating epilepsy, locating seizure origin and distinguishing epileptic seizures;
- testing drugs for reduction of convulsive effects;
- assisting in experimental cortical excision of epileptic focus;
- monitoring the brain development;
- investigating mental disturbances, sleep disorders and physiology;
- providing a hybrid data recording system together with other imaging modalities.

However, even nowadays, these recordings are still affected by different artifacts, with their origins varying from environmental, such as electrical interference, to experimental, usually related to the setup of the acquisition system, to physiological, such as ocular, muscular and cardiac activity. These artifacts contaminate the quality of EEG data, interfering with neural information and potentially imitating cognitive or pathologic activity, and therefore distorting the visual interpretation and diagnosis in clinical research [5].

Either by identifying morphological features or by examining frequency bands associated with different mental activities/conscious states, EEG signals allow for a better understanding of brain activities. As such, several signal processing techniques to aid clinicians in their EEG interpretation have been developed, each with their own advantages and limitations. These include linear regression, filtering methods (linear, adaptive, Weiner and Bayes filters), Empirical Model Decomposition (EMD), wavelets, and Blind Source Separation (BSS) methods, which are based on unsupervised algorithms that separate multi-channel (MC) EEG signals in a set of independent sources, such as Independent Component Analysis (ICA) [5].

Despite the benefits presented by these methods, most of these are only able to partially remove the artifacts, or are too computationally complex to be successfully used in real time scenarios, since they are not automatic (require the manual evaluation of a specialist to remove the noise). As such, Deep Learning Neural Networks (DLNNs) are promising algorithms in EEG signal cleaning, since they have shown to be able to automatically learn the complex characteristics of the signals, eliminate the artifacts, losing less valuable information and, once trained, are fast to apply, requiring less EEG channels than other methods [7–15]. Additionally, advances in long term EEG acquisition technology are allowing recordings using a lower number of electrodes (*i.e.*, sub-scalp EEGs). MC models are limited to EEG recordings with the exact number of channels with which they were trained, whereas single-channel (SC) models can be used regardless of the number of channels present in the acquisition system.

1.3 Goals and Expected Contributions

This project aims at advancing the state of the art of EEG signal processing by developing an automatic SC model using DLNNs that is able to clean and reconstruct noisy EEG signals. To this end, a set of goals is proposed:

- Development of a pipeline preparing a preliminar dataset and training SC Deep Learning (DL) approaches with different hyperparameters.
- Comparison of the trained SC model with previously trained DL approaches developed from MC data, to verify if the new models are able to clean EEG signals without losing information, making this approach usable in any EEG acquisition system.
- Reducing the sample size, by segmenting samples in smaller portions. Ten-minute segments make the samples too heavy for DL models with dozens of layers. A reduction of sample size would allow an increase in the complexity of the models, and consequently an enhancement in the ability to learn the different patterns of EEG signals, leading to better results.

1.4 Structure

After the introduction, this document is composed of five more chapters. **Chapter 2** introduces the main concepts used in this project, from EEG montages and characteristics to the more usual artifacts and a few DL architectures. In **Chapter 3**, the current state of the art regarding EEG signal processing techniques is reviewed, with an emphasis on DLNNs used to clean EEG signals. **Chapter 4** describes the data and methods used in the course of this study. Results are discussed in **Chapter 5** and conclusions are presented in **Chapter 6**.

2

Background Concepts

In this chapter, the relevant concepts for a better understanding of this thesis are introduced. Section 2.1 introduces the Electroencephalogram (EEG) biosignal. Section 2.2 describes the rhythmic waves and potentials available in neural activity. Section 2.3 presents various noisy artifacts that are usually present in EEG. Section 2.4 describes Deep Learning (DL) architectures, which will be used in this work, and Section 2.5 describe the metrics used to assess the performance of the developed models.

2.1 Electroencephalogram

EEG is a non-linear and non-stationary biosignal that measures the electrical activity generated by the brain. Specifically, EEG electrodes measure the sum of the potentials originated by the synapses of populations of neurons [1, 6]. EEG is commonly used in the diagnosis and monitorization of brain related pathologies that alter brain activity, such as brain tumors, encephalopathies, sleep disturbances, and epilepsy [2–4].

EEG signals may be classified according to the type of acquisition system into invaisve or non-invasive. In a non-invasive EEG, scalp EEG or simply EEG, the electrodes are placed on the scalp of the patient. In an invasive EEG or intracranial Electroencephalogram (iEEG), also referred to as Electrocorticogram (ECoG), the electrodes are placed directly onto the surface of the brain, during a surgical procedure [17].

2.1.1 Scalp Electroencephalogram

Scalp EEG is a non-invasive technique that allows a continuous monitorization of a patient for a determined period of time. Since it covers several areas of the

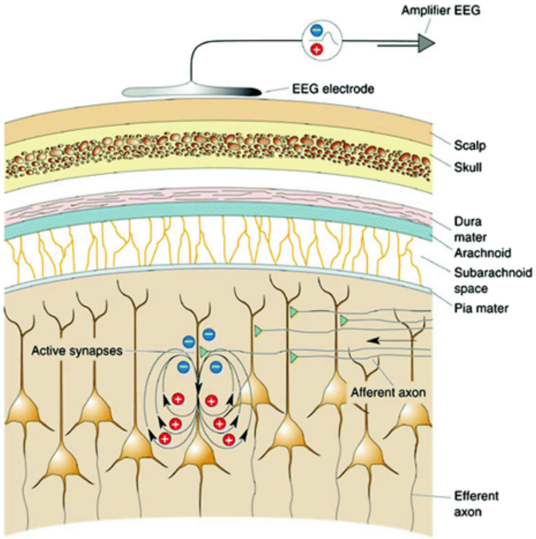


Figure 2.1: Synaptic potentials measured in an EEG. From [16].

scalp, it can also be used to study neural mechanisms that take place in different areas of the brain.

There are a few different types of architectures when it comes to placing the electrodes, with their names referring to the way the electrodes are distributed. The International 10-20 System [18], composed of up to 21 electrodes, is the most commonly used. The numbers 10-20 represent the way the electrodes are placed across the scalp, from Inion (Iz) to Nasion (Nz), at 10%, 20%, 20%, 20%, 20% and 10%. Following a similar logic, there is also the International 10-10 System, usually composed of of 64 to 85 electrodes, and the International 10-5 System, composed of up to 300 electrodes [18, 19]. These three architectures can be seen in Figure 2.2.

2.1.2 Intracranial Electroencephalogram

To acquire iEEGs, the electrodes are placed directly onto the patient's brain, and there is not an IS one can follow. As seen in Figure 2.3, the electrodes can be subdural electrodes [20], consisting of strips or grids placed on the surface of the brain, or depth electrodes, which are very thin and flexible, making them suitable to be implanted inside the cerebral cortex. Consequently, these electrodes are able to capture electrical activity from both superficial and deeper brain areas.

Comparing to scalp EEGs, iEEGs are acquired with lesser number of layers between the sensors and the signal sources [22]. Therefore, it presents a better Signal-to-Noise Ratio (SNR) and lesser noisy artifacts. However, being an invasive technique, it presents some risks such as hemorrhage, intracranial hematomas,

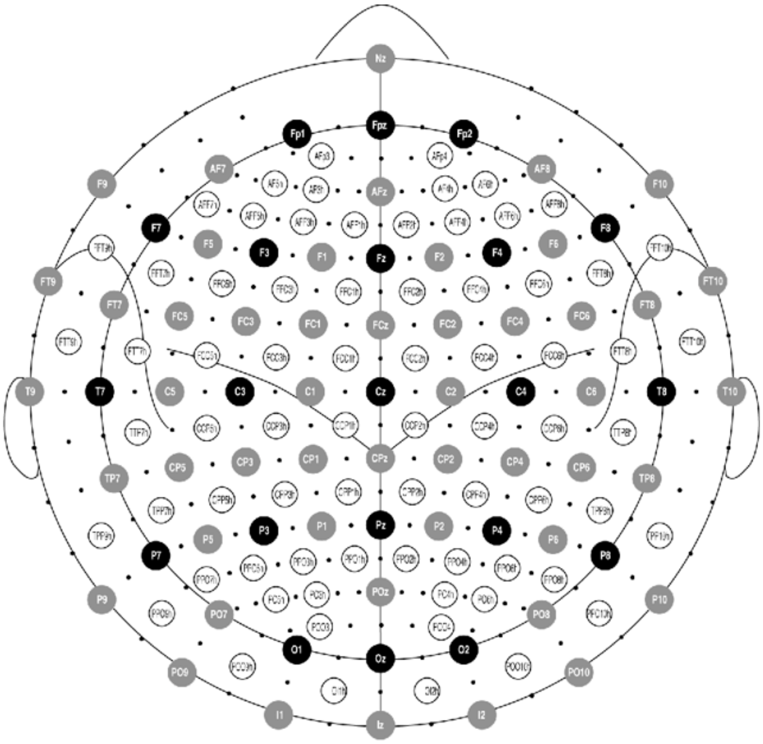


Figure 2.2: Three International System (IS) ways for electrodes' placement in EEG: International 10-20 System (electrodes in black), International 10-10 System (electrodes in black and grey), and International 10-5 System (all electrodes). From [19].

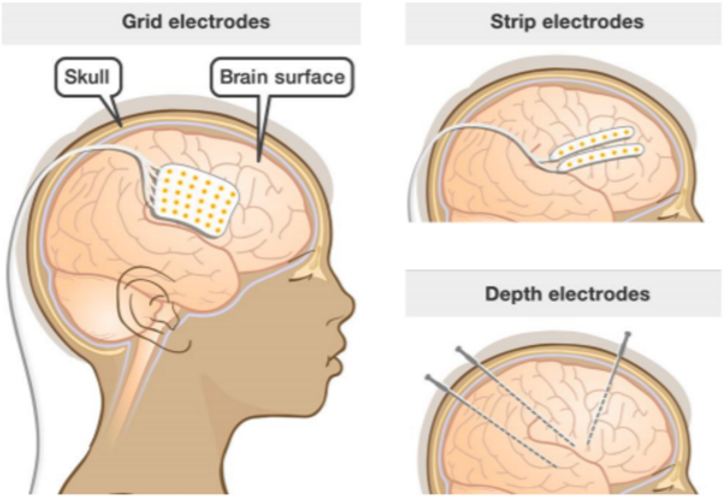


Figure 2.3: Different types of iEEG electrodes. From [21].

infection, and death [17, 22].

2.1.3 Electroencephalogram Montages

There are different ways to represent an EEG, in what is termed a montage or arrangement.

In bipolar montages [23], each channel usually results from the difference between two adjacent electrodes, in a specific order so that they are covering the entire scalp. In the bipolar parasagittal montage, channels can be created by connecting adjacent electrodes in two lines (anterior to posterior), with the midline electrodes also linked to each other, whereas the bipolar transverse montage links adjacent electrodes in a chain going from left to right.

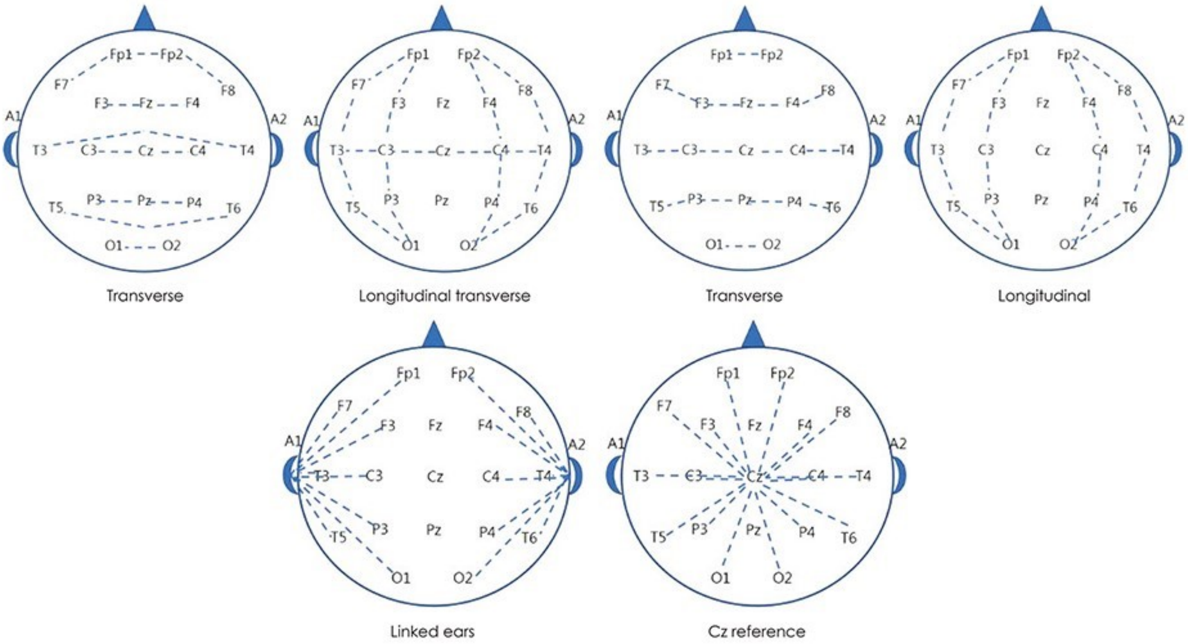
Monopolar or referential montages compare the signal of each of the electrodes to a common electrode, designated as reference. To avoid distortion of the signals, the ideal reference electrode would have no potential (null reference). Thus, central electrodes - between Fz and Nz - are usually used as reference [24], as is the difference between electrodes placed below or behind the ear lobes (mastoids) [25]. Several common montages are shown in Figure 2.4.

Other types of montages include the average reference montage, which compares the signal of an electrode to the average of all electrodes, usually excluding electrodes such as Fp1 and Fp2 for being susceptible to head and/or eye movements; the Laplacian montage [23], which compares the signal of an electrode to the average of the electrodes closest to it; and the Reference Electrode Standardisation Technique (REST), a method that transforms a scalp point or the average reference to an approximate zero reference, basing the transformation on the common origin of the two potentials before and after transformation, i.e. the actual neural sources or their equivalent sources [26, 27].

2.2 Neural Activity

EEG signals comprise two different types of activity: rhythmic and transient. Rhythmic activity [2, 29], caused by brain cells' excitability, is usually divided into frequency bands, which are represented in Figure 2.5:

- Delta waves (0.5 to 4 Hz): the slowest waves with the highest amplitudes, most predominant in children up to one year of age and in sleep stages 3 and 4. In adults, they occur most frequently in the frontal area of the brain, as opposed to posterior areas in children.



Transverse

Longitudinal transverse

Transverse

Longitudinal

Linked ears

Cz reference

Figure 2.4: Common EEG montages. From [28].

- Theta waves (4 to 8 Hz): usually found in children, during sleep, or in meditating adults.
- Alpha waves (8 to 13 Hz): occur more frequently in posterior areas of the brain, with higher amplitudes on the dominant hemisphere. They happen while relaxing and with eyes closed, stopping once the eyes are open or there is mental exercise. As such, they are predominant in relaxed adults.
- Beta waves (13 to 30 Hz): occur more frequently in the frontal area of the brain, being predominant when patients have their eyes open, or in states of high alert, anxiety, or focus.
- Gamma waves (above 30 Hz): are correlated with large scale brain network, usually in order to execute and achieve a certain motor or cognitive task.

Transient potentials are divided into normal (associated to sleep stages) and abnormal (which can represent interictal activity or an epileptic seizure).

2.3 Artifacts

As stated previously, scalp EEG recordings are acquired using non-invasive systems. Non-invasive systems, in addition to neural activity, usually capture data which are not related to data of interest, thus called noisy artifacts. These artifacts,

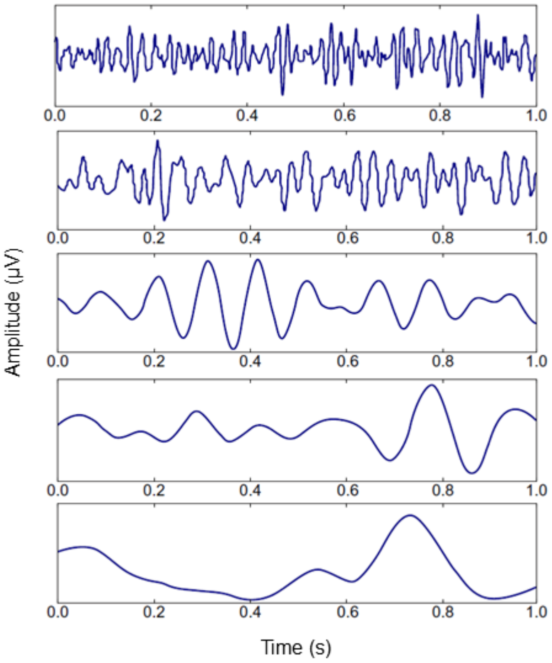


Figure 2.5: (From top to bottom) Gamma, Beta, Alpha, Theta, and Delta waves. Adapted from [29].

whether periodic or aperiodic, reduce the utility of the results and must be minimised when possible. They can be divided according to their source into environmental, experimental, and physiological [5].

2.3.1 Environmental Artifacts

Environmental artifacts originate from energy sources that surround the patient in their daily life, or directly from the acquisition system. These sources can create electrical, magnetic or electromagnetic interference, generating noisy signals in the form of, for example, power line interference with frequencies of 50-60 Hz [30, 31]. In addition, instrumental artifacts resulting from the different components of the acquisition system, are a type of environmental artifact that can be observed in the form of thermal noise (caused by changes in the temperature of the system's components), or pink noise (small variations of the condensed-matter materials) [32].

These artifacts can usually be eliminated almost completely, using digital filters, since they are confined to a small and fixed interval of frequencies [32]. Figure 2.6 shows signals affected by different artifacts, including power line noise, an environmental artifact.

2.3.2 Experimental Artifacts

Experimental artifacts are usually caused by human errors in the setup of the acquisition system, such as incorrect electrode placement [33], poor adhesion of the electrodes or a bad cleansing of the scalp, or by movements of the patient, which can cause the electrodes to change their position and consequently cause the occurrence of electrical impedance, leading to the distortion of the signal.

These artifacts are deemed uncontrollable and unpredictable. Although they can be reduced with a correct and careful setup of the acquisition system, it is almost impossible to completely avoid them, especially in uncontrolled environments. These artifacts cannot be eliminated using filtering techniques, since they do not have a characteristic interval of frequencies, regularly intersecting with the spectra of the signals of interest [30].

2.3.3 Physiological Artifacts

Physiological artifacts are caused by physiological processes happening in the patient's body [5], such as eye blinks, swallowing, chewing, cardiac activity, or even breathing.

Due to the proximity of the eyes to the brain, ocular related artifacts are the most prejudicial to EEG recordings. The change in potentials in the retina, caused by blinking or eye movement, is propagated and captured by the acquisition system (as seen in Figure 2.6), mostly in the frontal lobes, having higher amplitudes than the signals of interest [7]. However, these artifacts overlap with the frequencies of interest, preventing their removal using simple filtering techniques.

Muscle movement is one other source of physiological artifacts. These artifacts can originate from head muscles, especially when talking, chewing, and swallowing, being close to some of the electrodes. On account of their great diversity, these artifacts can encompass a big range of frequencies, overlapping mostly with beta waves (13 to 30 Hz). Since they can be caused by distinct groups of muscles, the signals they originate can have different shapes and amplitudes and are less repetitive than other artifacts, which hampers their removal [34]. Figure 2.6 shows examples of signals affected by muscular artifacts.

The electrical signal induced by cardiac activity can also constitute an EEG artifact. This artifact is usually repetitive and comprises a characteristic pattern which is visually easy to identify. However, normal pulse artifacts are non-stationary, meaning that even though they usually have a frequency of 1.2 Hz, that could change depending on the patient's cardiac activity [35]. Their amplitude is usually very low on the scalp, unless an electrode is placed over a blood vessel, which demonstrates how their positioning is fundamental in removing pulse artifacts.

Other less common physiological artifacts are the ones related to breathing and perspiration. Their frequency spectrum is very close to 0 Hz, and as such, they can be easily eliminated using a high pass filter.

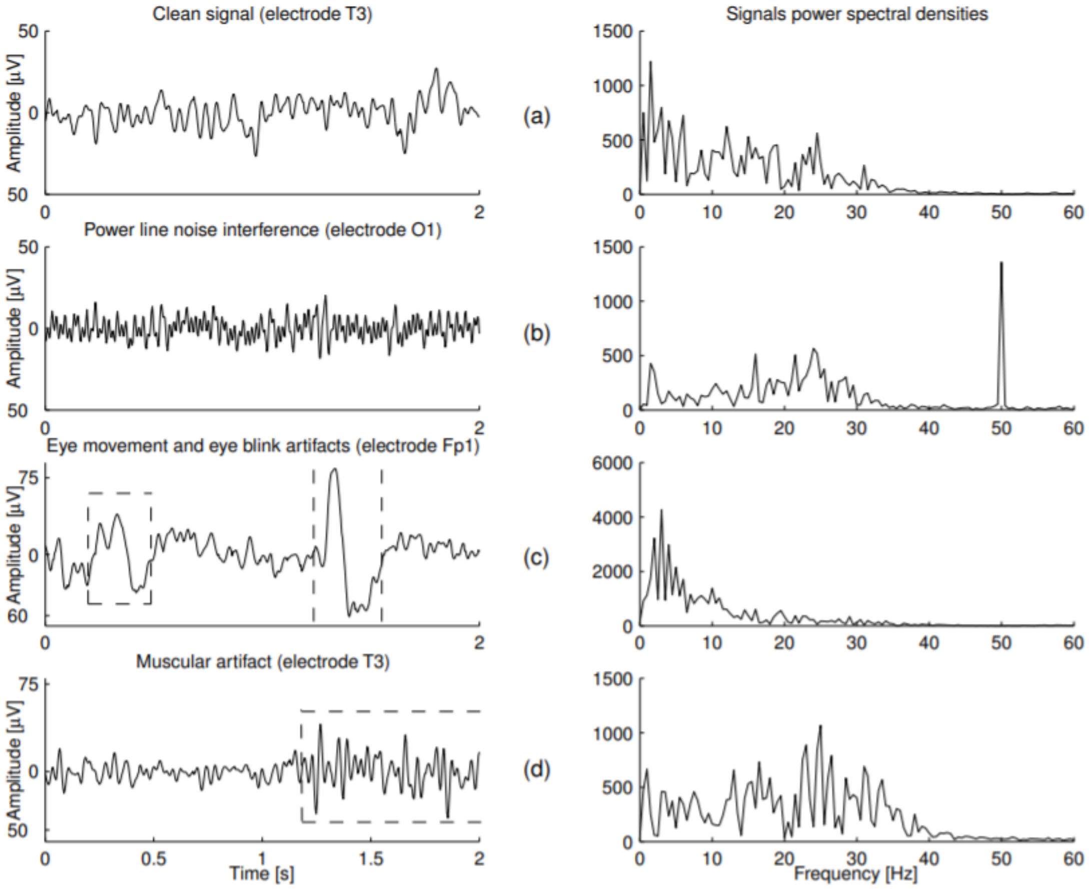


Figure 2.6: Amplitudes and Power Spectral Densities (PSDs) of signals with different kinds of artifacts: (a) a clean signal; (b) a signal with power line interference; (c) a signal with ocular artifacts; (d) a signal with muscular artifacts. From [36].

2.4 Deep Learning Architectures

DL architectures are based on Artificial Neural Networks (ANNs), which try to imitate the biological process of the human brain cortex. These architectures consist of advanced Machine Learning (ML) models which can automatically learn how to classify data without human support. This learning can be supervised, semisupervised or unsupervised. The increase of data, hardware, and software availability in the last decades has allowed a sudden raise in development and popularity of these models, which are already being used in image classification and bioinformatics [37]. There are several different types of DL architectures.

2.4.1 Feed Forward Neural Networks

Feed Forward Neural Networks (FFNNs) are the simplest deep learning architecture, mimicking the way our neurons are connected [37]. In each artificial layer of a FFNN, several different neurons, modulated by weights (dictating the strength of the response), are part of a sum function, which goes through an activation function and is then transmitted to the neurons in the following layer, thus imitating the synapse process in a human neuron. These activation functions can either be linear or non-linear (the latter are preferred since they allow models to learn more complex patterns).

Neural Networks (NNs) of this kind are usually made up of an input layer, multiple hidden layers of neurons, and an output layer (see Figure 2.7). The input layer transmits the initial data to the first hidden layer, in which all neurons transmit responses to the neurons of the second hidden layer (again, modulated by weights), passing through each hidden layer until the output layer is reached, gathering the responses, and producing a certain classification. The role of the hidden layers is to perform transformations to the data and try and learn, if possible, discriminative behavior within it.

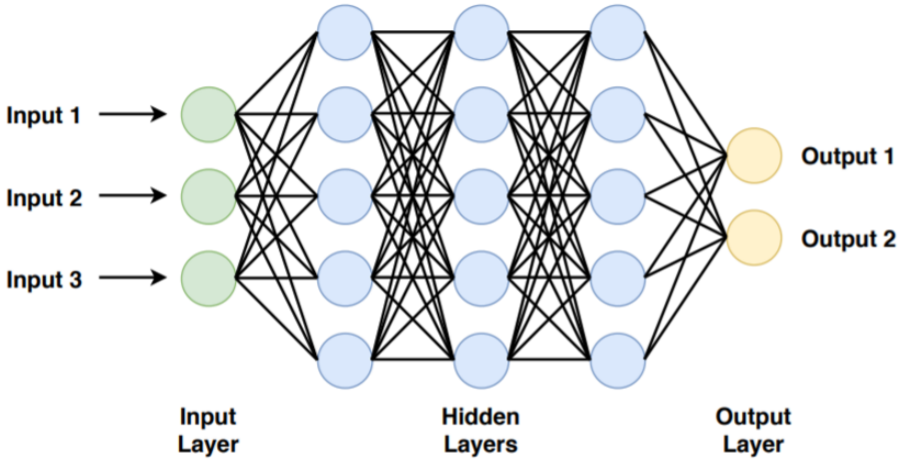


Figure 2.7: Representation of a FFNN, showcasing the different layers, with each circle representing a neuron. From [38].

2.4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) divide the input data into small subregions and analyse each of them individually, trying to learn characteristic patterns, before merging them. This process emulates the visual cortex, as is shown in Figure 2.8, and as such, is applied to complex data, such as 1D, 2D, or 3D data. These networks have:

- Convolutional layers: they create feature maps, by applying several filters that move through the input data at a certain rate (this rate is called the stride), transforming it, and passing every feature map through an activation function. To normalize the output, a Batch Normalization (BN) layer can be used [39]. The higher the stride and the lower the padding (kind of frame used to keep spatial information), the lower the number of values and the faster the computational time.
- Pooling layers: down-sampling layers with no weights which also go through the data at variable strides. They can be of different kinds, the most common being maximum (retrieves a feature map with the maximum values in each subregion) and average (retrieves a feature map with the average values in each subregion). Global pooling layers transmit only one value per feature map [38].
- Dense layers: also called fully connected layers, since all neurons are connected to all neurons from the previous and next layers (something that does not occur in convolutional or pooling layers). They are used at the end of the network to merge the information, reduce the number of features, and classify the data.

2.4.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) use information from either the past or the future to learn about the current data, by keeping an internal memory that enables them to create sequential rules. Simple RNNs have two main disadvantages: firstly, when training using gradient-based learning methods, the gradient may vanish; secondly, since the information from past samples does not go away, they gain a long-term dependency problem. Therefore, different types of RNNs have been developed to overcome these issues.

Long Short-Term Memory (LSTM) [41] is a more complex type of RNN, in which the memory, or cell state, is influenced by a forget gate (which modulates how much information in memory will be kept), an input gate (modulating how

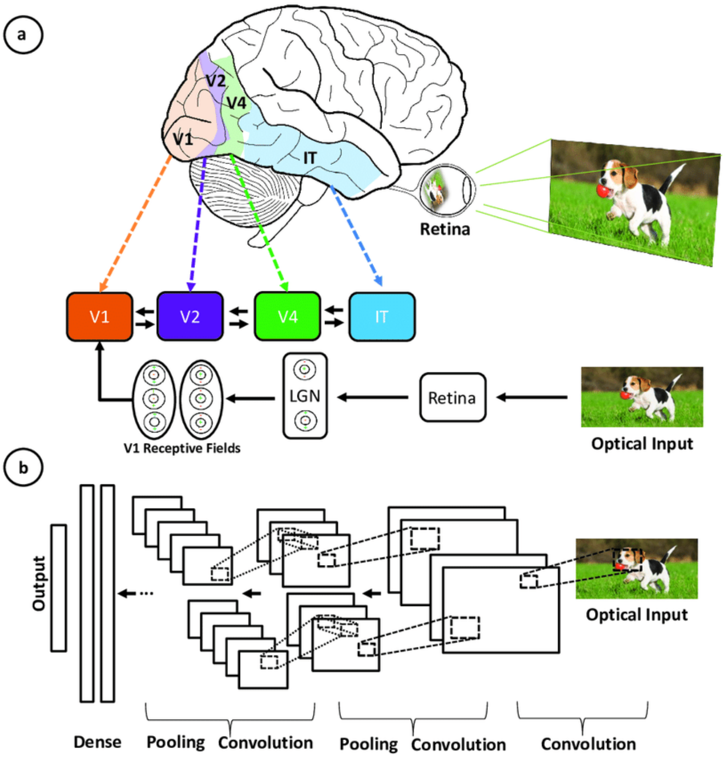


Figure 2.8: Visual cortex (a) vs. CNN (b). From [40].

much of the new information will be kept in the cell state) and an output gate (modulating how much information from the hidden and input states will be used to form the new hidden state). This means that LSTM networks can control how much of the information flows to the next time sample, at the cost of a higher complexity and computational time.

Gated Recurrent Unit (GRU) [42] is a simpler and faster version of LSTM networks, using only a reset gate to modulate the flow from past information and an update gate to modulate what new information should be added. RNNs can also use information from the future, when using bidirectional layers. LSTM and GRU architectures are represented in Figure 2.10.

2.4.4 Autoencoders

Autoencoders (AEs) are a specific type of neural networks with unsupervised learning that can be used to learn a compressed representation of data. An AE is composed of both an encoder and a decoder (see Figure 2.11). The encoder comprises a non-linear function that compresses the input into a lower-dimensional space (latent space), whereas the decoder attempts to recreate it to the original as closely as possible. Thus, the input and output must have the same number

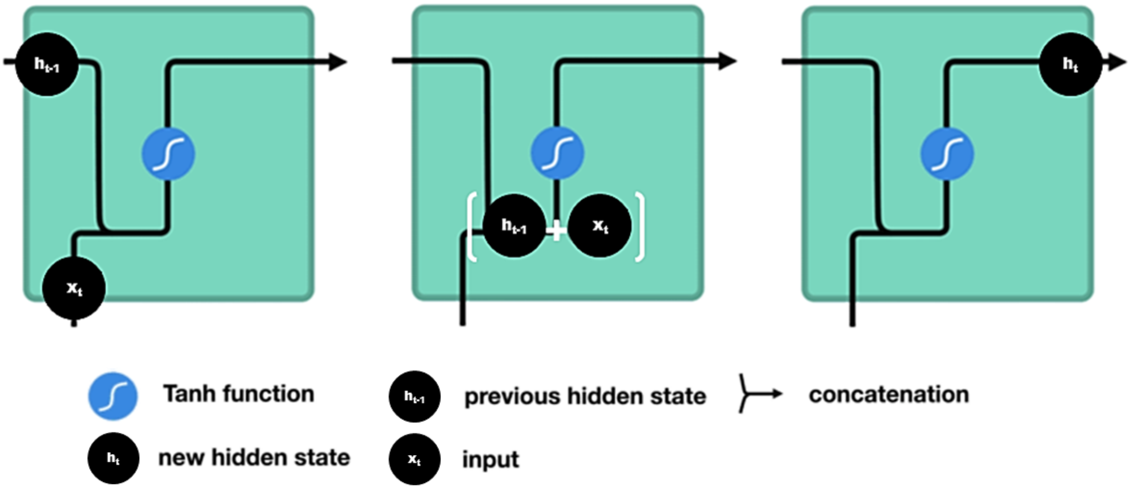


Figure 2.9: Representation of RNNs. From [43].

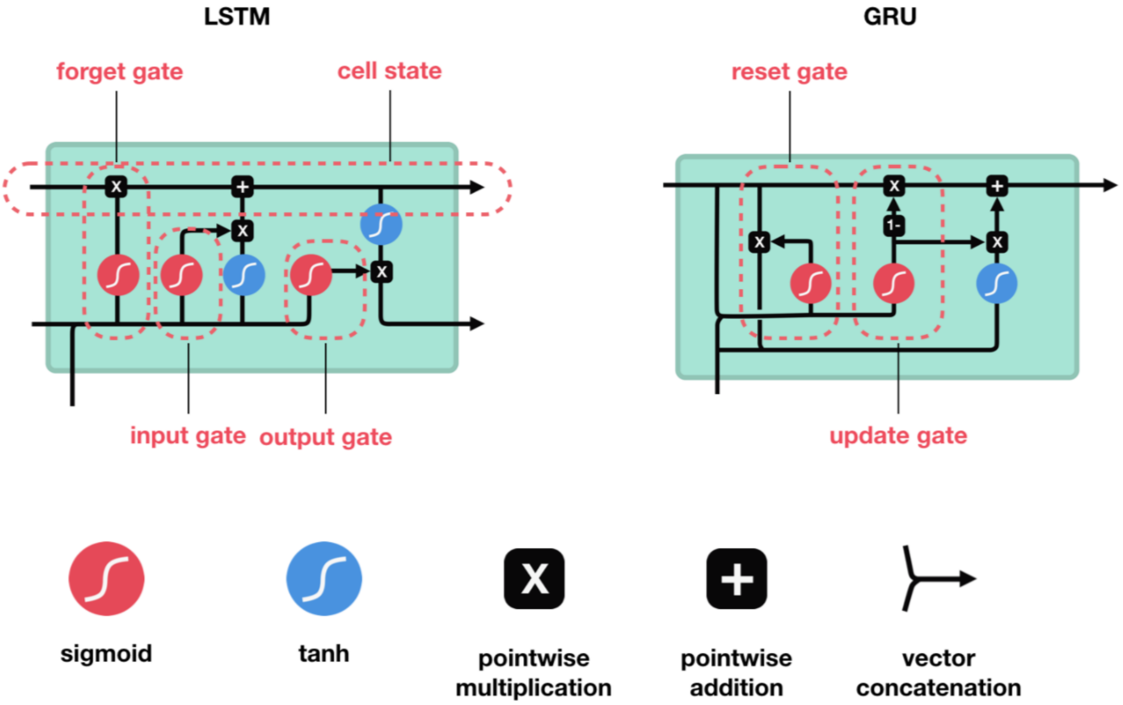


Figure 2.10: Representation of RNNs: LSTM and GRU. From [43].

of dimensions. The latent space is used to learn features which contain relevant information of the observed data [44]. Simple AEs are comprised of fully connected layers. However, there are other types of AEs, such as Convolutional Autoencoders (CAEs) and even AEs using LSTM layers.

Beyond basic AEs, there are regularized and variational AEs. The former use regularization methods to keep the AE from learning the identity function, thus

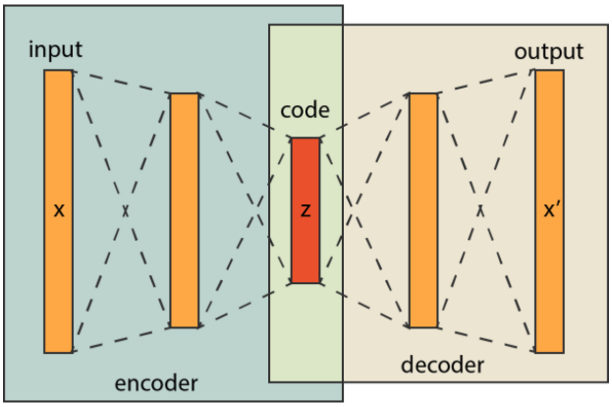
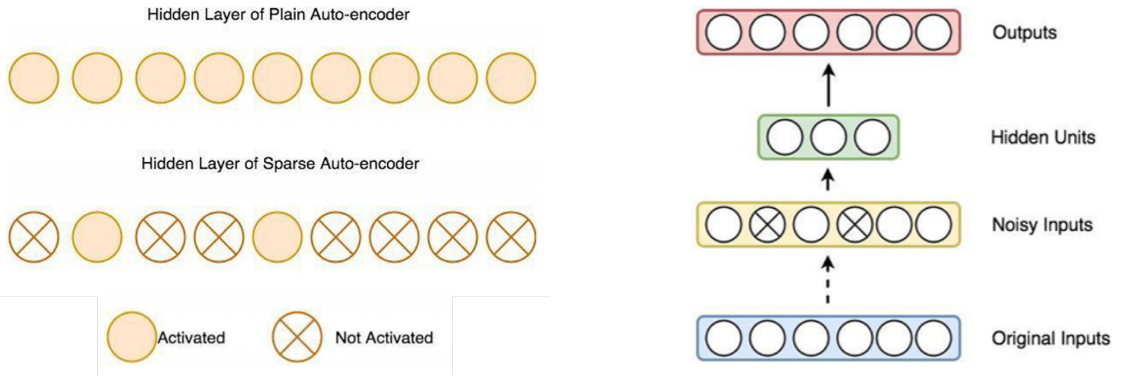


Figure 2.11: Representation of an AE architecture. From [44].

sometimes having a latent space that is larger than the input space. Some examples are:

- Sparse Autoencoders (SAEs): restrict the number of activated hidden neurons by introducing sparsity-constraints on the hidden units, so that only a fraction of the neurons presents nonzero values (active neurons). This regularization is therefore achieved by adding a penalty term to the loss function that forces the autoencoder to represent each input as a combination of a small number of neurons, making the extraction of relevant features more efficient.
- Denoising Autoencoders (DAEs): created by adding partially corrupt inputs (obtained, for instance, by setting values to zero for a small proportion of input nodes) or introducing a noise term. The goal is to recover the original noise free data while making sure that the AE does not simply copy the input to its output. Using noisy inputs helps to prevent overfitting and enables the construction of more robust feature representations.
- Contractive AEs: similar to DAEs but use a regulariser instead of corrupting the input data, making the model robust to small variations of the input.

Variational AEs are different from regularized AEs, since they aim to create samples out of the latent space instead of creating a latent space out of the data, to avoid overfitting. This is achieved by encoding each input as a normal distribution over the latent space. Then, a point from the latent space is sampled from that distribution and decoded, enabling the calculation of the reconstruction error which is then backpropagated through the network. This ensures a better organisation of the latent space, allowing a good generative process [45].


Hidden Layer of Plain Auto-encoder
Hidden Layer of Sparse Auto-encoder
Activated
Not Activated
Outputs
Hidden Units
Noisy Inputs
Original Inputs

(a) Sparse AE

(b) DAE

Figure 2.12: Regularized AEs. From [44].

2.5 Evaluation Metrics

The evaluation of an EEG signal cleaning model is made through statistical validation. Metrics normally used include the Root Mean Squared Error (RMSE), which is the standard deviation of the residuals (prediction errors), measuring how concentrated the data is around the line of best fit (Equation 2.1); Relative Root Mean Squared Error (RRMSE), which divides the RMSE by the Root Mean Square (RMS) of the target EEG segment (Equations 2.2 and 2.3), and is used to measure normalised reconstruction error; Pearson Correlation Coefficient (PCC), to test the linear correlation between the reconstructed signals and the clean target signals (Equation 2.4); and Signal-to-Noise Ratio Difference (SNR$_{Diff}$), which is the difference between input SNR - the difference between the original noisy segment and the respective clean target segment - and output SNR - the difference between the reconstructed segment and the clean target segment - used to quantify noise attenuation (Equations 2.5 and 2.6).

$$RMSE(x,y) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{N}} \tag{2.1}$$

$$RMS(x) = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{2.2}$$

$$RRMSE(x,y) = \frac{RMSE(x,y)}{RMS(y)} \tag{2.3}$$

$$PCC(x,y) = \frac{Covariance(x,y)}{\sigma_x \sigma_y} \tag{2.4}$$

$$SNR(x,y) = 10 \times log_{10} \frac{\sum_{i=1}^{N} y_i^2}{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{2.5}$$

$$SNR_{Diff} = SNR(\hat{y},y) - SNR(x,y) \tag{2.6}$$

In the equations above, the x represents original noisy segments, the y represents the clean target segments, the $\hat{y}$ the segments reconstructed by the model, and N is the number of samples.

3

Related Work

This chapter will review the state of the art related to pre-processing and consequent artifact removal in Electroencephalogram (EEG) recordings. Since completely removing segments from EEG signals affected by artifacts can cause the loss of important information, different methodologies have been developed in order to remove the noise from the signal while maintaining the maximum of relevant information. As such, Section 3.1 regards linear regression; Section 3.2 discusses filtering methods; Section 3.3 reviews Empirical Model Decomposition (EMD); Section 3.4 focuses on wavelets; Section 3.5 presents techniques based on Blind Source Separation (BSS); and Section 3.6 describes approaches developed to remove artifacts from EEG recordings using Neural Networks (NNs).

3.1 Linear Regression

Up until the last decade of the twentieth century, linear regression algorithms were the technique most often used for artifact removal in EEG recordings, due to their low computational complexity [46]. These algorithms need channels that can be used as a reference for the artifacts to be eliminated - e.g. Electrooculogram (EOG) or Electrocardiogram (ECG) - which are subtracted to the EEG signal after the artifact propagation coefficients have been calculated [47]. This constitutes one of the main disadvantages of this method, since it is only efficient if reference signals can be obtained (which only happens for some types of artifacts and always requires more equipment) [30]. One other drawback of linear regression algorithms is that, since EEG is a non-linear and non-stationary biosignal, the linear nature of the algorithm is not the most suitable for removing artifacts, possibly eliminating relevant information from the EEG recording. As a result, these methods have been replaced by others developed more recently [35, 48].

3.2 Filtering Methods

To remove EEG signal components in specific frequency bands, linear filters can be used. Linear filters can be divided into low-pass, high-pass, band-pass, band-stop, and notch filters. These can be either Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters, the latter being the ones more frequently used since they do not introduce a zero phase distortion [49].

An EEG signal is usually filtered using both low-pass and high-pass filters, which reduce the noisy data introduced by high frequencies and eliminate low-frequency components such as breathing-related artifacts and the Direct Current (DC) component, respectively. The type of filters and their respective cutoff frequencies are usually chosen by the authors of each study. The main disadvantage of linear filters is that they are not able to dissociate signals in the same frequency bands, thus being incapable to remove all types of artifacts [35].

More complex filtering methods can adapt their weights to produce a filtered signal with a better Signal-to-Noise Ratio (SNR) [35]. Adaptive, Wiener, and Bayes filters are the ones more commonly used.

3.2.1 Adaptive Filtering

In adaptive filtering, the EEG signal and the artifact to be removed are assumed to be uncorrelated. These filters estimate the artifact signal from a reference, removing this estimation from the EEG signal [35, 50]. The parameters of the filter are adjusted in accordance with the optimization algorithm used, meaning that both the choice of the reference signal and of the optimization algorithm are of extreme importance, with the latter determining the efficiency and computational complexity of the filter. Some of the algorithms more often applied are the Least Mean Square (LMS), which keeps on adjusting a coefficients vector aiming to reduce the mean square error [32], and the Recursive Least Square (RLS), which minimizes a weighted linear least squares cost function, having a higher computational cost but being more efficient than the previous [51].

These filters are easy to use since no calibration is required, and can be implemented online, but are limited in the sense that they depend on reference signals for the artifacts.

3.2.2 Wiener Filtering

Wiener filters are linear and time-invariant filters that result from a statistical approach, using the Power Spectrum Density (PSD), a representation of the distribution of signal frequency components which normalizes the amplitudes by the frequency resolution [52], to estimate a signal that is the closest possible to an artifact-free signal [32, 53]. Thus, these filters do not require a reference signal, overcoming the limitation of adaptive filters. However, Weiner filters must be calibrated before being used and cannot be used in real-time, because PSDs are not known *a priori* and can only be estimated once there is information about the system [32].

3.2.3 Bayes Filtering

Bayes filters utilize a probabilistic method based on the dynamic of a Marvok system, allowing them to be applied in real-time scenarios and without a reference signal [32, 54]. Due to their computational complexity, it is only possible to use methods that solely implement approximations of Bayes filters, such as Kalman and particle filters. Kalman filters are efficient and can accomplish a great ratio between computation time and memory usage [55], but need exact sensors and a detailed model of the process and measurements taken. If these are not available, particle filters are a viable option since they are flexible and relatively easy to implement [56].

3.3 Empirical Model Decomposition

EMD is a one-dimensional heuristic method suitable for non-linear and non-stationary signals. It divides the signal into Intrinsic Mode Functions (IMFs), which are calculated over multiple iterations until a convergence criterion is fulfilled [57, 58].

The main limitations of this method are its high computational complexity, hampering its use in real-time scenarios, and its high sensitivity to noise. An Ensemble Empirical Model Decomposition (EEMD) has already been developed. In EEMD, different white noise time series are added to the signal, allowing a better scale separation aptitude than the standard EMD, and thus presenting a higher level of robustness [59, 60].

3.4 Wavelet Decomposition

Wavelets are wave-like oscillations located in time, with two basic properties: scale or dilation, which is related to frequency and defines how "stretched" or "squished" a wavelet is, and location, which defines where the wavelet is positioned in time or space [61]. Wavelet transforms are generated by calculating the internal product of the signal to be cleaned and versions of a base function (a mother wavelet) with different values of the scale and location properties, resulting in several coefficients that represent the similarity between the signal and each wavelet transform [62, 63]. After this, a threshold is applied to these coefficients, and only the components that are not considered artifacts are kept and used in the reconstruction of the signal.

Since there is a wide variety of mother wavelets that can be chosen to best match the signal, this is a versatile tool that allows one to build different models that best suit each situation [35]. However, wavelet transforms are normally used alongside other techniques, due to their inability to completely remove artifacts that overlap the signal in the frequency spectrum, such as physiological artifacts.

3.5 Blind Source Separation

BSS methods are based on unsupervised algorithms that consider the recorded signal to be formed by a linear mixture of independent sources, following Equation 3.1.

$$X = AS + N \tag{3.1}$$

In this equation, the X represents the measured signals, the A the mixing matrix, the S the independent sources and the N the noise and error values [30, 64]. The main objective of BSS methods is to estimate, through a number of iterations, an unmixing matrix able to separate the different independent sources (assuming the noisy data is contained within these sources), following Equation 3.2.

$$\hat{S} = WX \tag{3.2}$$

Here, the $\hat{S}$ represents an approximation of the independent sources and the W the unmixing matrix [30, 64]. These techniques have the advantage of using only the information in the recorded data, thus not needing reference signals of the artifacts. However, the number of independent sources is never higher than the number of EEG channels available, meaning that, if the number of channels is small, the yielded results might be unsatisfactory.

Different BSS algorithms calculate the unmixing matrix using different statistical methods. These can be divided in four different classes: high-order statistic methods, second-order statistic methods, second-order statistic methods with non-stationary properties and methods that use temporal and spectral information to separate the sources [65]. In these algorithms, the data is assumed to be stationary, both temporally and spatially. As such, it is essential to divide the signal in quasi-stationary segments and to maintain the configuration of the system over the totality of the acquisition time. It is also assumed that each source is statistically independent from all the others. To ensure this, the signal can be previously submitted to Principal Component Analysis (PCA), an algorithm that uses an orthogonal transformation to convert the data in a number of linearly uncorrelated variables [35, 66].

3.5.1 Independent Component Analysis

Independent Component Analysis (ICA) is one of the most used methods of BSS in EEG signals. After an algorithm is used to calculate the unmixing matrix, signals from several EEG channels go through it, which returns the independent components. Then, through the observation of the frequency, time series, and topographic map of each independent component, one can identify which components represent artifacts, and remove them. The remaining components go through the inverse of the unmixing matrix, returning the EEG signal clear of artifacts [67].

Since the number of EEG channels must be equal to or higher than the number of independent components, ICA does not return satisfactory results when only a few EEG channels are used, and cannot be applied to only one EEG channel. Furthermore, its computational complexity limits its use in real-time contexts [32]. A few different algorithms can be used for the calculation of the unmixing matrix with the objective of trying to bypass previous limitations and obtain better results:

- Infomax ICA [68]: high-order statistic method that calculates the unmixing matrix through maximum likelihood estimation. It can be divided into two types: the restricted version, able to separate only leptokurtic sources (signals that are more concentrated on the mean, and thus more tapered than the normal distribution), such as physiological artifacts; and the extended version, able to separate both leptokurtic and platykurtic sources (less concentrated on the mean than the normal distribution), such as electrical artifacts.

- Adaptative Mixture Independent Component Analysis (AMICA) [69]: an enhanced version of Infomax ICA that computes the Probability Density Function (PDF) and spatial projection for each component, and can combine different ICA models to obtain a better approximation of each source.
- FastICA [70]: high-order statistic method, faster than the aforementioned in cases where all components are non-Gaussian sources, and that maximizes the negentropy (negative entropy) of the source distributions.
- Second-Order Blind Identification (SOBI) [71]: second-order statistic method that reduces the correlation between time-lagged versions of the sources in order to compute the unmixing matrix. It is ideal for real-time applications, due to its low computational time.
- Hybrid methods: they combine ICA with other techniques, such as wavelet and EMD based methods. Even though these methods improve the separation of Independent Components (ICs), by combining the advantages of linear and non-linear algorithms, and can be applied to single-channel (SC) EEGs, their high computational complexity makes them too slow to use in real-time contexts [32].

3.5.2 Canonical-Correlation Analysis

Canonical-Correlation Analysis (CCA) is a second-order statistic method that searches for uncorrelated components using a weaker statistic condition than the one used by ICA algorithms. However, CCA takes into account temporal correlations within each component. Just as in ICA, noise removal is done by zeroing the components that correspond to artifacts before the reconstruction of the signal. Unlike ICA, CCA is deterministic - always returns the exact same results if the same dataset is used [30, 72].

3.5.3 Morphological Component Analysis

Morphological Component Analysis (MCA) decomposes the signal in components with different morphological characteristics, with each component represented in a dictionary [73]. It can only be used to remove artifacts that are available in the used database, meaning that a reference signal of the artifact is required in order to be effective.

3.5.4 Automatic Blind Source Separation

The aforementioned BSS methods are not automatic. This means that the distinction between the sources representing neural activity and the ones representing artifacts must be done through manual observation. To further enable the use of BSS techniques in real-time scenarios, a few methods have been developed to attemp to make them autonomous [74–81].

A few studies have already been done using automatic BSS methods. Shoker *et al.* [74] combined technologies, using SOBI to separate the signals into different ICs and support Support-Vector Machines (SVMs) to classify the sources that correspond to artifacts and remove them. Mognon *et al.*, Nolan *et al.* and Chaumon *et al.* [75–77] developed Automatic EEG artifact Detection based on the Joint Use of Spatial and Temporal feature (ADJUST), Fully Automated Statistical Thresholding for EEG artifact Rejection (FASTER) and Semi-Automated Selection of Independent Components of the electroencephalogram for Artifact correction (SASICA) respectively, which divide the segments into ICs (with the Infomax algorithm) and then use threshold-based approaches to search for different artifact-specific spatial and temporal features, enabling the identification of ICs that represent different artifacts. Winkler *et al.* [78] constructed a model that uses Temporal Decorrelation source SEParation (TDSEP), an ICA method that takes temporal correlations into account, for separating the signal into ICs and the weight vector of a Linear Programming Machine (LPM) to select the best features from the ICs to be used by different classifiers - a LPM, a SVM and a Regularized Linear Discriminant Analysis (RLDA), obtaining viable performances using the latter with only 6 features. Radüntz *et al.* [79] developed a pipeline divided into 3 distinct phases: a pre-processing phase, where filtering and ICA were performed; a feature generation phase, where image processing algorithms were used to obtain feature images; and the classification phase, where different classifiers - Linear Discriminant Analysis (LDA), logistic regression, SVMs and Artificial Neural Networks (ANNs) - were trained using the feature images as input and classified them either as artifacts or EEG components, with ANNs outperforming the remaining classifiers. Tamburro *et al.* [80] formed IC-fingertips from 14 temporal, spatial, spectral, and statistical features of ICA-resulting sources that served as input for different SVM models, each able to automatically classify a different type of artifact, and reconstructed the EEG signals with the components that were not deemed artifacts. Finally, Pion-Tonachini *et al.* [81] trained classifiers using median PSD and autocorrelation fucntions, scalp topographies, single and bilaterally symmetric Equivalent Current Dipole (ECD) model fits, and features from [75–77], calling it the ICLabel classifier, which is freely available on MATLAB.

Most of these methods classify each component considering temporal, spatial, and spectral features. After this classification, the components classified as artifacts are removed, and the signal is reconstructed.

Table 3.1: EEG Pre-Processing Models using Automatic BSS

Authors	Technique	Databases	Artifacts
Shoker *et al.* [74]	SOBI-SVM	London King's College Hospital EEG Database	Eye Blinking
Mognon *et al.* [75]	Infomax ICA Threshold-based Model	Two Personal Databases	Eye Blinking Eye Movement Generic Discontinuities
Nolan *et al.* [76]	ICA Threshold-based Model	Personal Databases with simulated and real data (47 subjects)	Eye Blinking Eye Movement Muscular Artifacts Discontinuities Linear Trends White Noise
Chaumon *et al.* [77]	Extended Infomax ICA Threshold-based Model	Personal Databases	Eye Blinking Eye Movement Muscular Artifacts Bad Channel Rare Events Non Dipolar Components
Winkler *et al.* [78]	TDSEP-LDA	Personal Database (12 subjects)	Every type of artifact
Radüntz *et al.* [79]	ICA-ANN	Personal Database (57 subjects)	Eye Blinking Eye Movement Cardiac Pulse Muscular Artifacts Impedance
Tamburro *et al.* [80]	ICA-Nonlinear SVM	Personal Database (62 subjects)	Eye Blinking Eye Movement Cardiac Pulse Muscular Artifacts
Pion-Tonachini *et al.* [81]	ICA-wConvolutional Neural Network (CNN)	Personal Database	Every type of artifact

3.6 Deep Learning Neural Networks

In the last few years, several Deep Learning (DL)-based studies have been published with the goal of automatically removing artifacts from EEG signals.

A few studies focus mainly on using autoencoders to clean EEG signals. Yang *et al.* [7] used Stacked Sparse Autoencoders (SSAEs) to remove ocular artifacts, with an offline stage where they identify noiseless segments and feed them to the model, and an online stage, where the trained DL architecture is used to construct a clean EEG segment from a noisy one. The authors used PSD and the Root Mean Squared Error (RMSE) as performance metrics and compared the architecture to other four EEG signal processing methods: a shallow network, Sparse Autoencoder (SAE), ICA, Kurtosis ICA (K-ICA) and SOBI. They concluded that the deep neural network had a variety of advantages, such as not needing additional EOG recording as reference, being suitable for fewer number of electrodes, being fast and able to be applied in online systems, and having a good generalization ability. Leite *et al.* [8] proposed a method using deep Convolutional Autoencoders (CAEs) to remove eye blinking and jaw clenching noise from EEG signals, inspired by methods used in music and voice processing. They settled for a structure with 8 hidden layers, using convolutional, max pooling and upsampling layers. The used evaluation metrics were the Peak Signal-to-Noise Ratio (PSNR) and the Mean Squared Error (MSE). The results were compared to those of a Butterworth IIR passband filter, with cutoff frequencies of 4 Hz and 40 Hz. The authors confirmed that the NN always obtained a higher PSNR, having better efficiency than Butterworth filters. Ghosh *et al.* [9] proposed a novel combination of a SVM and an Autoencoder (AE) to automatically identify and remove eye blink artifacts in EEG signals. The signal within the window is firstly fed to the SVM classifier, which classifies it either as non-artifact and slides the window forward, or as artifact and feeds it to a trained AE. This AE corrects the specific window and returns a clean EEG signal. The authors used five metrics to evaluate the performances: RMSE, SNR, Correlation Coefficient (CC), Mean Absolute Error (MAE), and Mutual Information (MI) and compared it to wavelet and Adaptive Noise Cancellation (ANC) methods. The proposed method performed better than the others in every metric, showing that it could correct the EEG signal while preserving a good correlation between the EEG before and after correction.

Other studies focused on the development of Deep Learning Neural Networks (DLNNs). The paper by Sun *et al.* [10] proposes a new One-Dimensional Residual Convolutional Neural Network (1D-ResCNN) that can automatically learn the non-linear and discriminative features of noisy and clean EEG signals. Batch Normalization (BN) layers are used after the convolutional layers to speed up training, optimize results and provide faster convergence. The authors based their performance evaluation of the architecture on SNR and RMSE. They also used PSD, Approximate Entropy (ApEn) and autocorrelation function to measure the influence of the method on the non-linear characteristics of the EEG. They compared these metrics against ICA, FastICA, RLS filter, wavelet transforms and DLNNs. In regards to ECG and Electromyography (EMG) artifacts, the proposed model outperformed all other techniques in all evaluation metrics, performing similarly to DLNN in the presence of EOG artifacts. Lopes *et al.* [11] used long-term data acquired from patients with epilepsy in presurgical monitoring (previously inspected by two experts that removed independent sources corresponding to artifacts) to develop and train an EEG artifact removal model based on Deep Convolutional Neural Networks (DCNNs), with the NN consisting of convolutional layers with 32 to 128 filters and using the leaky Rectified Linear Unit (ReLU) function in all activation layers. The model was evaluated using RMSE, Relative Root Mean Squared Error (RRMSE), Pearson Correlation Coefficient (PCC) and Signal-to-Noise Ratio Difference (SNR_{Diff}), and it was compared to the 1D-ResCNN model and to an automatic ICA model based on extended Infomax ICA and Multiple Artifact Rejection Algorithm (MARA) classifier. The DCNN performed the best out of the tested approaches, and the authors also concluded that their model was faster than the other artifact removal models. This combination of minor reconstruction error with fast computation means the method may be used to remove artifacts from signals in real-time scenarios. However, this multi-channel (MC) approach presents a limitation, since it requires the EEG signals to have 19 channels. Jurczak *et al.* [12] developed a CNN with the objective of eliminating eye blink artifacts from EEG signals. The adopted architecture was composed of two convolutional layers, with 20 and 10 filters respectively, with ReLU as activation function. The authors used real EEG data that served as reference to generate artificial EEG signals with eye blink artifacts that were used to train and test the architecture. They compared this pre-processing approach to ICA and regression methods, using statistical coefficients that measure the correlation between the same channels in the clean and the noisy signal (C_{kk}) and between the reference electrode of the noisy signal and each electrode of the clean signal (C_{Fp1}), as well as Mean Absolute Percentage Error (MAPE), RMSE and *Skewness*. The CNN performed better in the removal of eye blink artifacts, especially in central and parietal electrodes, but also requires multiple channels of EEG signal. Mathe *et al.* [13] proposed a One-Dimensional Convolutional Neural Network (1D-CNN) that uses a new hybrid algorithm named Spider Monkey-based Electric Fish Optimization (SM-EFO) to tune the model parameters of the network and optimize the results. As evaluation metrics, the authors utilized Source-to-Distortion Ratio (SDR), Cumulative Squared Euclidean Distance (CSED), CC and MAE. They compared their approach to both SVM and DLNNs, achieving better results in all the aforementioned metrics. Zhang Z. *et al.* [14] developed a Multi-Module Neural Network (MMNN) using multiple parallel denoising modules, each constructed by 4 1D convolutional layers with ReLUs, a residual connection, and two full connected layers. They synthesized noisy EEG epochs using clean, EOG and EMG epochs to train the model, and used Temporal Relative Root Mean Square Error (T-RRMSE), Spectral Relative Root Mean Square Error (S-RRMSE), and CC in order to compare its performance to different DL architectures, as well as conventional models such as regression and ICA. The authors concluded that their model achieved higher signal reconstruction accuracy than the other methods, while also using less training data. Finally, Zhang W. *et al.* [15] constructed a Gated Recurrent Unit based Multi-type Artifact Removal algorithm for Single-Channel (GRU-MARSC), using an encoder-decoder approach to employ an adaptive artifact pattern concentration to separate the noise and clean EEG signals. They trained the model using semi-simulated data and applied it to 2000 real signal segments, comparing its CC and RRMSE to Surrogates-based Artifact Removal (SuBAR) (an approach based on wavelet analysis) and EEMD-CCA, concluding that their model is more effective and efficient at multi-type artifact removal than the compared approaches.

Tables 3.2 and 3.3 summarize the reviewed work regarding the use of NNs in EEG artifact removal. Some findings have shown that the use of NNs in EEG processing and artifact removal have both advantages and disadvantages: on the one hand, they are able to learn complex patterns and characteristics of the signals, using less channels than other methods and keeping the characteristics of the original signal, thus losing less information. They are also very fast to apply (once they are trained). On the other hand, the training phase of these networks can be quite time consuming, and their generalization capacity highly depends of a large amount of data.

In summary, Table 3.4 presents a comparison between the different artifact removal techniques aforementioned.

Table 3.2: EEG Pre-Processing Models using Neural Networks

Authors	Architecture	Databases	Artifacts
Yang *et al.* [7]	SSAEs	Personal Database (3 subjects) "Data Sets 1" for BCI Competition IV (7 subjects)	Ocular Artifacts
Leite *et al.* [8]	CAEs	Personal Database for noisy signals DEAP EEG Dataset for Emotion Analysis for clean signals (32 subjects)	Eye Blinking Jaw Clenching
Ghosh *et al.* [9]	Combination of SVMs and AEs	ABISSR Project Database (70 subjects)	Eye Blinking
Sun *et al.* [10]	1D-ResCNN	CHB-MIT Database (20 subjects)	Ocular Artifacts Muscular Artifacts Cardiac Artifacts
Lopes *et al.* [11]	DCNNs	EPILEPSIAE Database (25 subjects)	Every type of artifact
Jurczak *et al.* [12]	CNN	Combination of real EEG signals recorded during N-back task and artificially generated signals	Eye Blinking
Mathe *et al.* [13]	1D-CNN	CHB-MIT Database (22 subjects)	Ocular Artifacts Muscular Artifacts Cardiac Artifacts
Zhang Z. *et al.* [14]	MMNN	Clean and artifact epochs from the EEGdenoiseNet Database	Ocular Artifacts Myogenic Artifacts
Zhang W. *et al.* [15]	GRU-MARSC	Multiple databases for semi-simulated data [82–87] CAP Sleep Database for real data	Ocular Artifacts Muscular Artifacts Cardiac Artifacts Eye Blinking Motion Artifacts

Table 3.3: Performances of EEG Pre-Processing Models.

Authors	Architecture	Evaluation Metrics	Performance Results
Yang *et al.* [7]	SSAEs	RMSE	1.2838
Leite *et al.* [8]	CAEs	PSNR	7.810
Ghosh *et al.* [9]	Combination of SVMs and AEs	RMSE SNR CC MAE MI	0.024 1.562 0.892 0.0041 2.307
Sun *et al.* [10]	1D-ResCNN	SNR RMSE ApEn Autocorrelation	26.472 0.0680 0.61 0.9567
Lopes *et al.* [11]	DCNNs	RMSE RRMSE PCC SNR_{Diff}	4.83 0.52 0.86 8.81
Jurczak *et al.* [12]	CNN	C_{kk} C_{Fp1} MAPE RMSE *Skewness*	0.930 -0.027 0.805 2.935 0.037
Mathe *et al.* [13]	1D-CNN	SDR CSED CC MAE	3.4923 3049.9 0.74923 11.99
Zhang Z. *et al.* [14]	MMNN	T-RRMSE S-RRMSE CC	0.273 0.276 0.896
Zhang W. *et al.* [15]	GRU-MARSC	RMSE CC	0.5 0.88

Table 3.4: Comparison of the Advantages and Disadvantages of various Artifact Removal Methods

Approach	Fast	Autonomous	Suited for Real Time Scenarios	Removes artifacts with frequency spectrum overlapping neural signals	Does not require extra electrodes
Segment Rejection	✓	✓	✓		✓
Linear Regression	✓	✓	✓	✓	
Simple Filters	✓	✓	✓		✓
Adaptive Filters	✓	✓	✓	✓	
Weiner Filters	✓			✓	✓
Bayes Filters	✓		✓	✓	✓
EMD				✓	✓
Wavelets					✓
ICA				✓	✓
CCA				✓	✓
MCA				✓	
Neural Networks	✓	✓	✓	✓	✓

4

Experimental Setup

In this chapter, the experimental setup is discussed. In Section 4.1, a brief description of the dataset is provided. Section 4.2 introduces the techniques utilized along the duration of this study, with Sections 4.2.1 and 4.2.2 explaining the first and second approaches followed for the construction of the single-channel (SC) models, respectively.

4.1 Dataset

Electroencephalogram (EEG) signals used in this study were obtained from the EPILEPSIAE repository [88]. This database is composed of long-term epileptic EEG signals and seizure meta-data acquired during presurgical monitoring, recorded over several days and with different sampling rates (250 - 2500 Hz). Data were recorded at Hôpital de la Pitié-Salpêtrière, Paris (France), Universitätsklinikum Freiburg (Germany), and Centro Hospitalar e Universitário de Coimbra (Portugal). Presently, the database contains signals from 275 patients and includes both scalp EEG and intracranial Electroencephalogram (iEEG) recordings - Table 4.1.

Table 4.1: EPILEPSIAE Database Recordings

Type of Recording	Number of Patients
10-20 Scalp EEG	222
iEEG	49
Both	4

This study used 4.5 hours scalp EEG recordings with a sampling rate of 256 Hz and using the International 10-20 System, with 19 available channels - Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T7, T8, P7, P8, Fz, Cz and Pz. The data belongs to 13 male patients and 12 female patients, aged 39.6±16.8 years, and was recorded at Universitätsklinikum Freiburg.

These signals had been previously pre-processed [11, 89]. Firstly, a 0.5-100 Hz bandpass 4th-order Butterworth filter and a 50 Hz 2nd-order notch filter were applied to remove the Direct Current (DC) component, power line interference and high frequency noise. Experimental errors such as saturated segments, flatlines and atypical peaks were also removed. Afterwards, signals were then divided into multi-channel (MC) 10-minute segments (using zero padding in incomplete segments), and the spherical interpolation method [90] was used to fix experimental errors that persisted. These signals were subsequently divided into training and testing datasets (containing, respectively, segments belonging to 20 and 5 patients). They were then processed using extended-infomax Independent Component Analysis (ICA), which separated the segments of both datasets into Independent Components (ICs). These components were in turn inspected by two specialists and removed if deemed noisy. Finally, signals were reconstructed using only non-noisy ICs. As such, there were two versions of each segment: before the application of extended-infomax ICA and after the removal of noisy ICs. That is, the former will be considered as model inputs and the latter as the desired target output.

It was decided that segments where more than half of their ICs (10 or more) represented noise were to be removed from the training dataset entirely, since they would not allow a reliable reconstruction of an EEG segment. Since the data was comprised of MC segments (with 19 channels) and the aim of this work is to develop intelligent models able to automatically reconstruct SC EEG signals, the signals were separated, with each of the 19 channels of a segment constituting a new separate segment. The details of the training and test datasets can be observed in Table 4.2.

4.2 Methods

Two separate approaches were developed in the construction of an automatic SC model to clean and reconstruct EEG signals - one using 10-minute segments - Subsection 4.2.1 - and one using only 5-second segments - Subsection 4.2.2. A schematic representation of the steps in these approaches can be found in Figure 4.1.

All models were constructed, trained and tested in a computer with an AMD Ryzen 5 2600 CPU 3.4 GHz, 64 GB of RAM, NVIDIA RTX 2060 Super, and Linux Ubuntu 20.04 LTS, using Tensorflow 2.4.1 and Keras 2.4.3 from Spyder 3.8 in Anaconda Spyder 4. Available resources included 4 GPU NVIDIA Quadro P5000 with

Table 4.2: Patient's Data

Dataset	**Patient ID**	**Number of Segments**	**Total Number of Hours**
Training Dataset	8902	2831	9183.333 Hours (55100 segments)
	30802	4294	
	32702	2432	
	46702	2698	
	53402	2204	
	56402	2204	
	58602	2983	
	59102	2774	
	60002	3002	
	64702	2394	
	75202	4180	
	80702	2679	
	85202	2888	
	93402	2698	
	93902	2033	
	112802	2755	
	113902	608	
	114902	3781	
	123902	1444	
Test Dataset	402	3059	2881.667 Hours (17290 segments)
	16202	3876	
	23902	2945	
	50802	2793	
	55202	4617	

16 GB GDDR5 RAM and 5 GPU NVIDIA Quadro RTX 5000 with 16 GB GDDR6 RAM. Statistical validation was performed in Matlab R2020b.

4.2.1 First Approach

The initial approach utilized full 10-minute segments as input for the constructed models, with each segment having 153600 samples due to its sampling rate of 256 Hz. Initially, a grid search was performed to determine the best architecture, as well as some of its hyper-parameters, using a preliminary dataset. Once the results of the grid search were analyzed and the model and its parameters decided, the selected network was trained with the full training dataset, and its performance evaluated using the test dataset.

4.2.1.1 Grid Search

In order to perform an initial grid search, a preliminary dataset was constructed, by randomly selecting 10% of the training data. Two types of architectures, represented in Figure 4.2, were developed:

- Deep Convolutional Neural Network (DCNN), composed of three sets of three 1D convolutional layers, with increasing number of filters in each set to extract features from the signal, followed by leaky Rectified Linear Unit (ReLU) activation function. ReLU has a fast computation time and nonlinear characteristics, making it a good option for use as activation function, but has the potential to result in *dead neurons*, which will cause some network neurons to output zero values for various inputs. Leaky ReLU addresses this drawback by reducing the output of negative inputs instead of cancelling them [91]. The data is returned to its original dimensions with the final convolutional layer.
- Deep Convolutional Autoencoder (DCAE), that encoded the data with three 1D convolutional layers with a stride of 2 (all with the same number of filters), reducing dimensionality, and then decoded it with 1D convolutional layers with a stride of 1, followed by upsampling1D layers which are used to increase dimensionality by repeating the rows of its input. These reconstruct the signal as closely as they can to the same dimensions of the initial input.

Two hyper-parameters were inspected during the grid search:

- Filter's size in the convolutional layers: the symmetry surrounding the unit under study cannot be maintained by filters of an even size, which results in data aberrations across the layers, and size 1 filters are unable to analyze the

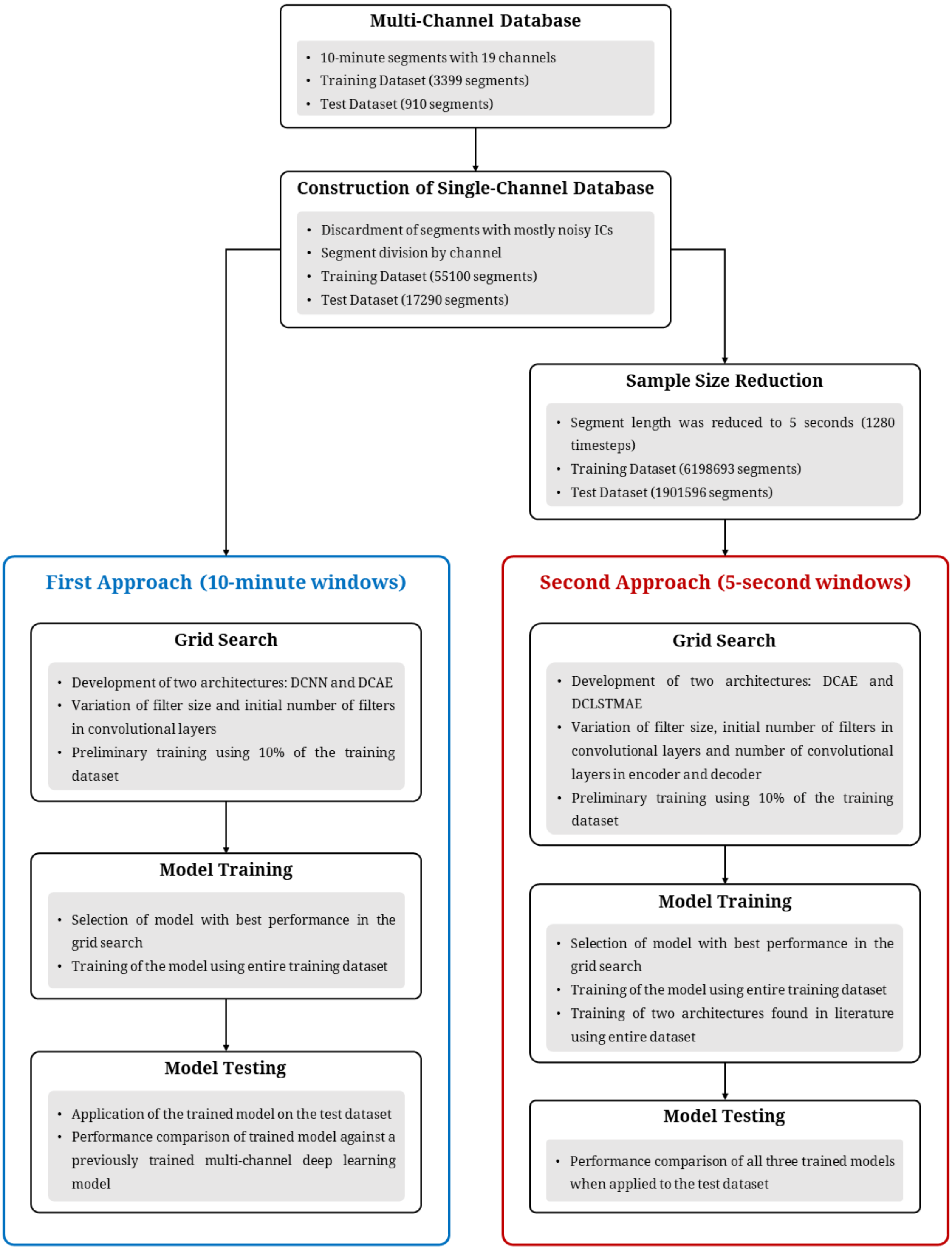

Multi-Channel Database
10-minute segments with 19 channels
Training Dataset (3399 segments)
Test Dataset (910 segments)
Construction of Single-Channel Database
Discardment of segments with mostly noisy ICs
Segment division by channel
Training Dataset (55100 segments)
Test Dataset (17290 segments)
Sample Size Reduction
Segment length was reduced to 5 seconds (1280 timesteps)
Training Dataset (6198693 segments)
Test Dataset (1901596 segments)
First Approach (10-minute windows)
Grid Search
Development of two architectures: DCNN and DCAE
Variation of filter size and initial number of filters in convolutional layers
Preliminary training using 10% of the training dataset
Model Training
Selection of model with best performance in the grid search
Training of the model using entire training dataset
Model Testing
Application of the trained model on the test dataset
Performance comparison of trained model against a previously trained multi-channel deep learning model
Second Approach (5-second windows)
Grid Search
Development of two architectures: DCAE and DCLSTMAE
Variation of filter size, initial number of filters in convolutional layers and number of convolutional layers in encoder and decoder
Preliminary training using 10% of the training dataset
Model Training
Selection of model with best performance in the grid search
Training of the model using entire training dataset
Training of two architectures found in literature using entire dataset
Model Testing
Performance comparison of all three trained models when applied to the test dataset

Figure 4.1: Schematic representation of developed approaches.

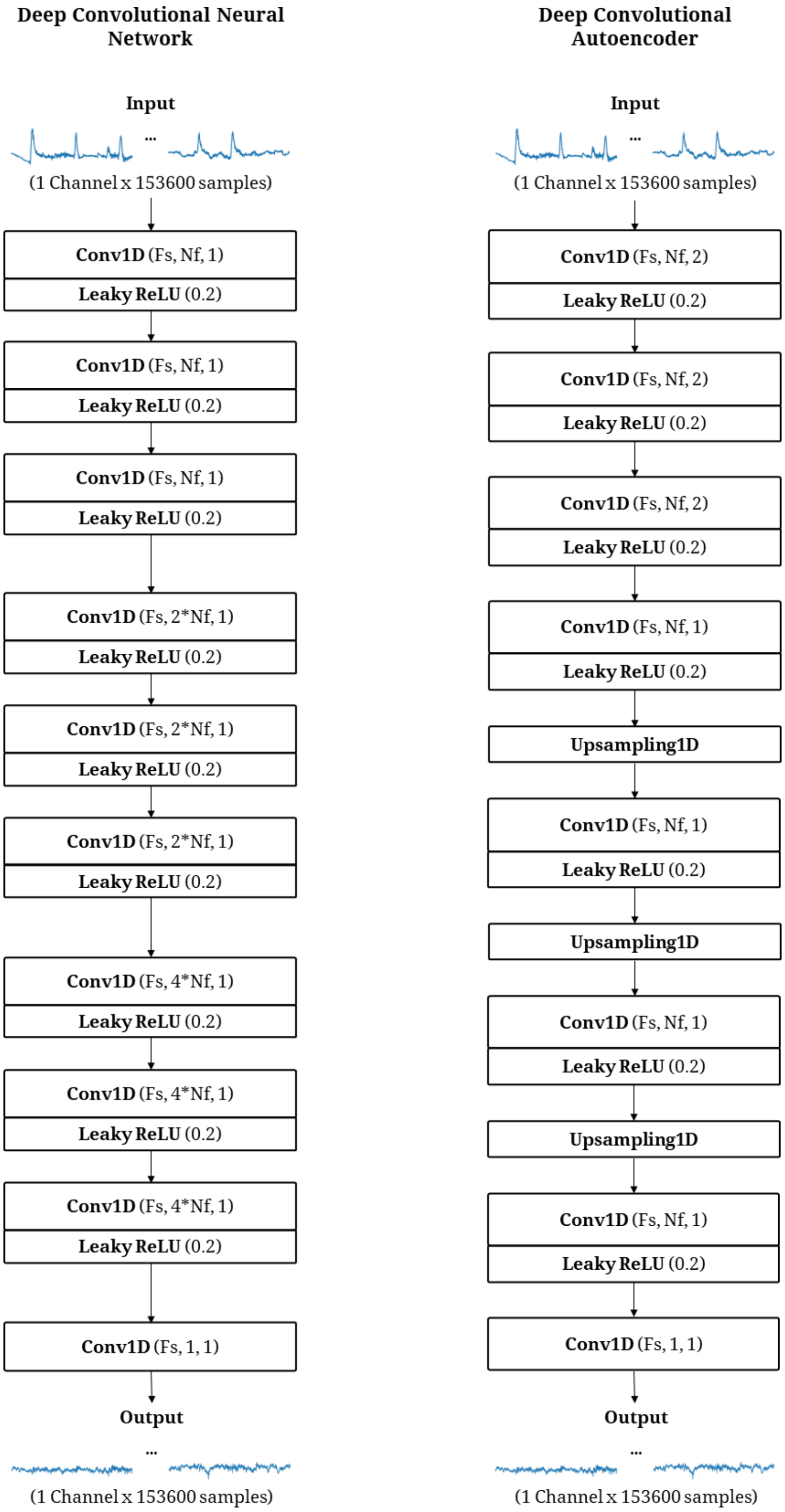


Figure 4.2: Architectures developed in the first approach. *Fs* refers to the selected filter size, *Nf* to the selected number of filters, and the third value is the stride of the layer. The 0.2 in Leaky ReLU layers is the value of α, a fixed parameter that represents the slope of the activation function [92].

values near the unit of analysis. As such, this value was varied between 3, 5 and 7 [93].

- Initial number of filters in the convolutional layers: a higher number of filters per layer increases the width of the model [94]. This value was varied between 8, 16, 32 and 64. Higher values were not used since the models would become too computationally complex giving the available hardware.

The 24 models (12 per architecture) were each trained 10 times using the preliminary training dataset, and the mean and the standard deviation of the Root Mean Squared Error (RMSE) and Relative Root Mean Squared Error (RRMSE) values of the validation group were analysed in Subsection 5.1.1.

4.2.1.2 Training and Testing

Analysis of grid search results showed that the DCAE with a filter size of 7 and 64 filters achieved the best performance, presented in Figure 4.3. Therefore, architecture was selected to be trained using the entire training dataset. Training hyper-parameters are specified in Table 4.3. Each model was trained 10 times, so that the training and validation subdatasets were randomly selected for each run, to ensure that performance results were not affected by chance.

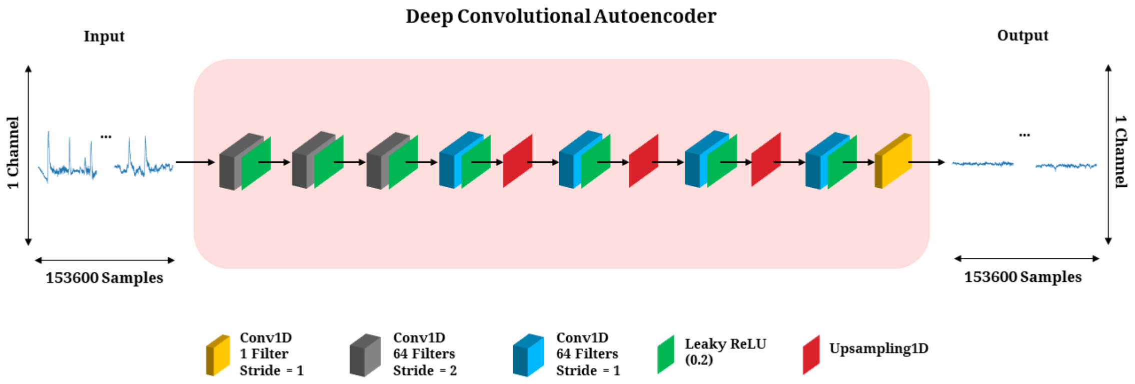

Deep Convolutional Autoencoder
Input
Output
1 Channel
1 Channel
...
...
153600 Samples
153600 Samples
Conv1D 1 Filter Stride = 1
Conv1D 64 Filters Stride = 2
Conv1D 64 Filters Stride = 1
Leaky ReLU (0.2)
Upsampling1D

Figure 4.3: Deep Learning (DL) architecture trained using the full training dataset in the first approach.

Once all 10 models had finished training, these were applied to the test dataset, and each model's performance was evaluated through RMSE, RRMSE, Pearson Correlation Coefficient (PCC) and Signal-to-Noise Ratio Difference (SNR$_{Diff}$). These metrics are further described in Section 2.5. The means, medians, standard-deviations, and the first and third quartiles of each evaluation metric were calculated, and these

Table 4.3: Model Training Hyper-Parameters

Hyper-Parameter	Value
Dataset Partition	Holdout Validation 70/30
Optimization Function	Adam
Learning Rate	3.0×10^{-4}
Loss Function	RRMSE
Epochs	500
Number of Runs	10

values were compared to values from the MC DL architecture that was developed by Lopes *et al.* [11], which had been previously trained and tested on the same data (before the segment division by channel). Results are presented in Subsection 5.1.2.

4.2.2 Second Approach

For the second approach, the 10-minute segments were divided into 5-second windows. This was done because the use of segments with that many timesteps make the training of the models an extremely computationally heavy procedure. The reduction of sample size allows the construction and training of models that are more complex in their layers, consequently expanding the ability to learn the different patterns and characteristics of EEG signals. Furthermore, seizure prediction and detection algorithms commonly use EEG signals in smaller windows (from 1 to 30 seconds).

To perform this reduction, the zero padding was removed so that there were no null 5-second segments. This sometimes led to the last window of a former 10-minute segment having a duration inferior to 5 seconds. In these cases, overlapping from the previous 5-second window was added in order for all segments to have an equal length of 5 seconds, corresponding to 1280 timesteps. This reduction was performed on both the training and test datasets, and was followed by a new grid search to find the best DL architecture and respective hyper-parameters, to be trained using the new training dataset and evaluated using the new test dataset.

4.2.2.1 Grid Search

To perform a new grid search, 10% of the training data was once again randomly selected to serve as a preliminary training dataset. Two more types of architectures, represented in Figure 4.4, were developed:

- Simple DCAE, similar to the ones constructed in the first approach.
- DCAE with an embedded Bidirectional LSTM (BiLSTM), which can use information from both the past and present since it adds one more Long Short-Term Memory (LSTM) layer that reverses the direction of information flow.

Three hyper-parameters were varied during the second grid search:

- Size of the filters in the convolutional layers, between 3, 5 and 7.
- Number of filters in the convolutional layers, between 64, 128 and 256.
- Number of layers before the output layer of the latent space, between 3, 4 and 5, to check how the increase or decrease in depth would affect the performance of the DL networks.

In total, 54 models were developed (27 per architecture). Just like in the first grid search, each model was trained 10 times with the preliminary training dataset and the validation group were examined - see Subsection 5.2.1.

4.2.2.2 Training and Testing

The best architecture, shown in Figure 4.5, was a DCAE with an incorporated BiLSTM layer (DCLSTMAE), using 256 filters of size 5, and 4 convolutional layers, which was trained using the entire training dataset, with the hyper-parameters of Table 4.3. Additionally, two more models, retrieved from literature, were trained with the same data and parameters: the DCNN developed by Lopes *et al.* and the One-Dimensional Residual Convolutional Neural Network (1D-ResCNN) developed by Sun *et al.* [10], with the former being adapted to SC use so that it could be trained using the training dataset.

All the trained models were then applied to the test dataset. The evaluation metrics were obtained by employing the same steps as in the first approach, and compared with the three types of architectures. This comparison, along with the comparison of our SC models from each approach, can be seen in Subsection 5.2.2.

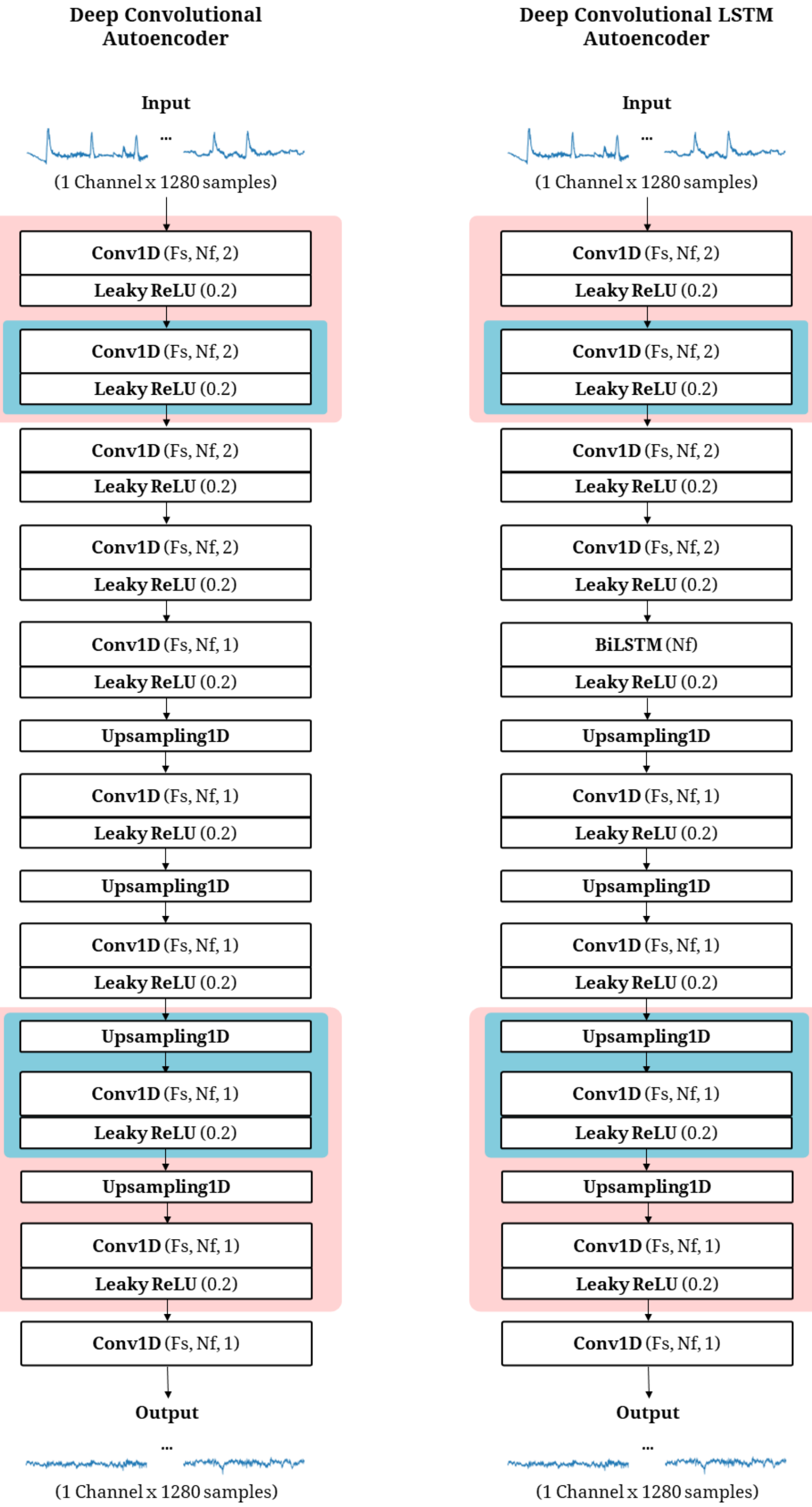


Figure 4.4: Architectures developed in the second approach, with *Fs* being the filter size, *Nf* the number of filters, and the third value the stride of the layer. Layers encapsulated in blue and red are added when the number of convolutional layers in the encoder and decoder is increased to 4 and 5, respectively. The 0.2 in Leaky ReLU layers is the value of α, a fixed parameter that represents the slope of the activation function [92].

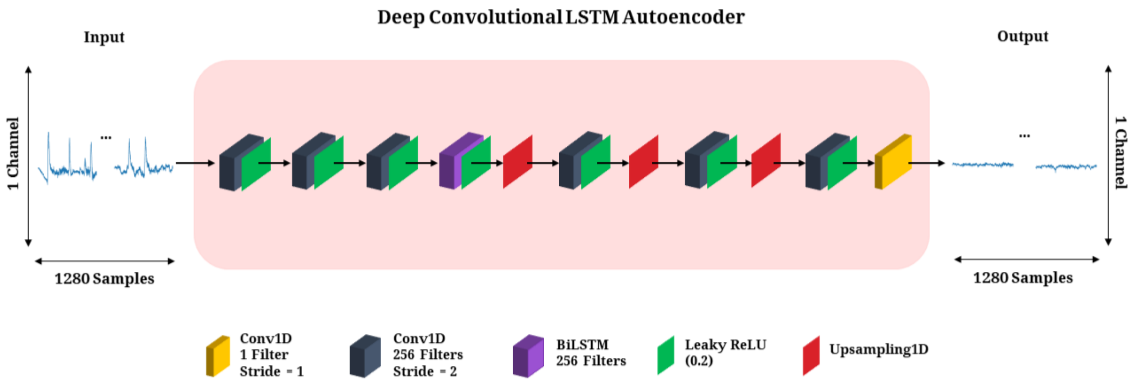


Figure 4.5: DL architecture trained using the full training dataset in the second approach.

5

Results and Discussion

This chapter describes the results of the study, divided by approach: Section 5.1 refers to the results of the first approach, where all constructed models received 10-minute Electroencephalogram (EEG) segments as input. Section 5.2 exhibits results of the second approach, after the reduction of the segments from 10 minutes to 5 seconds. Each of these sections begins by presenting the results of the respective grid searches and consequent architecture selection, and then discusses the performance of the fully trained models.

5.1 First Approach

This Section presents the results obtained in the first approach. Results of the grid search of hyper-parameters for Deep Convolutional Neural Networks (DCNNs) and Deep Convolutional Autoencoder (DCAEs) are discussed in Subsection 5.1.1. These are followed by a performance comparison in Subsection 5.1.2 between the best model found through the grid search trained with the entire training dataset, and a multi-channel (MC) DCNN model that had previously been trained with the same data [11].

5.1.1 Grid Search

Tables 5.1 and 5.2 detail the Root Mean Squared Error (RMSE) and Relative Root Mean Squared Error (RRMSE) means and standard deviations of the 10 runs of all the models developed during the grid search, trained with the preliminary training dataset. As seen in Figure 5.1, values tend to improve with higher number of filters as well as filter sizes. In Figure 5.2 the results show that, for a low number of filters (8 and 16), the DCNNs obtain better results when compared to DCAEs; however, the inverse happens when more than 32 or 64 filters are used - Figure 5.3. Both architectures show a decrease of RMSE and RRMSE values with the increase of both filter size and number of filters. The lowest values, presented in bold, both belong to the same architecture, the DCAE with the highest filter size and number of filters used, which is consistent with the notion that an increase in depth and width of a Deep Learning (DL) model expands its learning capabilities and leads to better results [94].

The model that achieved the best results, whose parameters are presented in Table 5.3, was then trained ten times using the full training dataset.

5.1.2 Model Testing

Once all ten runs had been completed, the performance of the trained models was analysed, and discussed in this subsection. Firstly, the calculated evaluation metrics are analysed and compared to values from a similarly trained MC model - Subsubsection 5.1.2.1. Then, to check on the model's ability to remove artifacts from EEG signals, we consider several examples of segments with different artifacts reconstructed both by our model and the MC architecture - Subsubsection 5.1.2.2.

5.1.2.1 Evaluation Metrics

Values for the evaluation metrics introduced in Section 2.5 were determined using the target segments and visualized in barplots, in order to compare the original EEG signals, the signals reconstructed by the developed single-channel (SC) architecture, and the previously trained MC model. The barplots compare the three RMSE, RRMSE and Pearson Correlation Coefficient (PCC) values for each channel, and the Signal-to-Noise Ratio Difference (SNR$_{Diff}$) values of the SC model and the MC model, also regarding each channel separately. For these graphics, medians and interquartile ranges were used as central tendency statistics in favor of means and standard deviations since experimental errors that persisted in the pre-processed segments gave rise to outliers which skewed the metrics' distributions.

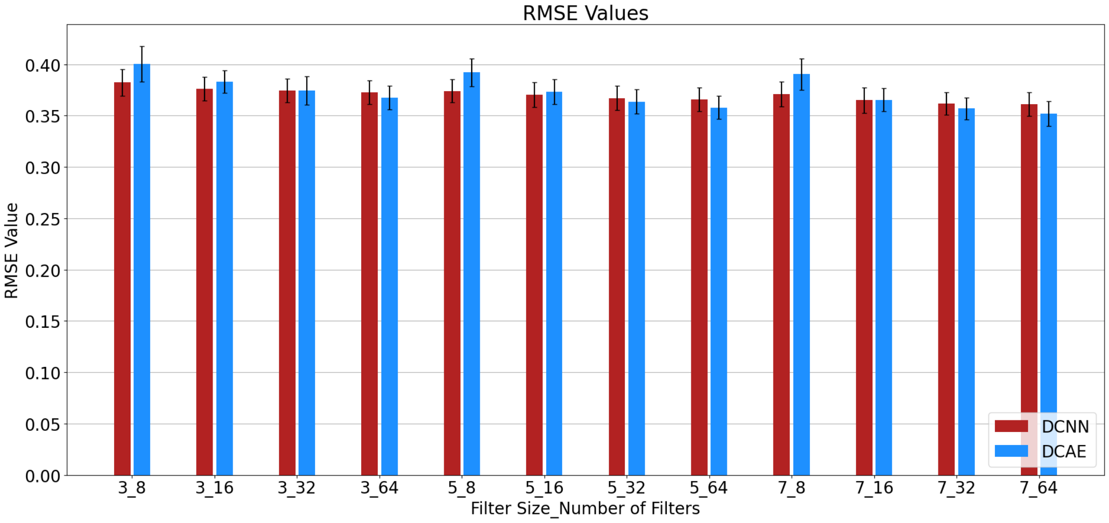
Figure 5.4 presents the RMSE values. Since these values dictate the errors in the reconstructed signals in regards to the target segments, the lower the value the better the ability of the model to remove artifacts in each channel. The highest reductions in RMSE values that our model achieved in comparison with the original segments are in channels Fp1, Fp2, F7 and F8. These are also the channels with the highest RMSE values in the original segments, since they are most heavily affected by ocular artifacts. The Fz channel is the only one where the RMSE value of the SC model slightly increases upon the original value, albeit being the channel with the lowest value overall.

Table 5.1: RMSE results for DCNNs and DCAEs in first approach Grid Search

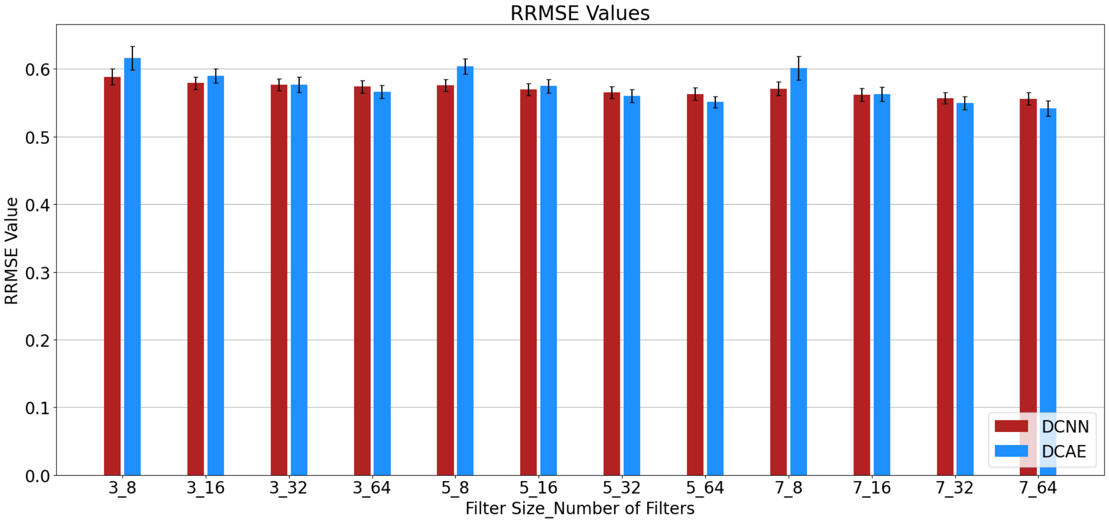
		DCNNs		DCAEs	
Filter Size	**Number of Filters**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
3	8	0.3823	0.0132	0.4004	0.0175
	16	0.3762	0.0113	0.3832	0.0111
	32	0.3747	0.0115	0.3745	0.0140
	64	0.3728	0.0116	0.3677	0.0114
5	8	0.3741	0.0112	0.3923	0.0135
	16	0.3703	0.0120	0.3733	0.0119
	32	0.3670	0.0117	0.3636	0.0117
	64	0.3658	0.0118	0.3579	0.0112
7	8	0.3708	0.0120	0.3903	0.0151
	16	0.3651	0.0124	0.3655	0.0112
	32	0.3616	0.0110	0.3569	0.0106
	64	0.3611	0.0116	**0.3520**	0.0123

Table 5.2: RRMSE results for DCNNs and DCAEs in first approach Grid Search

		DCNNs		DCAEs	
Filter Size	**Number of Filters**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
3	8	0.5885	0.0115	0.6163	0.0176
	16	0.5791	0.0093	0.5899	0.0106
	32	0.5769	0.0087	0.5764	0.0114
	64	0.5739	0.0092	0.5660	0.0096
5	8	0.5760	0.0085	0.6039	0.0114
	16	0.5700	0.0087	0.5747	0.0100
	32	0.5651	0.0086	0.5598	0.0096
	64	0.5632	0.0090	0.5510	0.0081
7	8	0.5709	0.0098	0.6009	0.0176
	16	0.5620	0.0097	0.5627	0.0103
	32	0.5567	0.0082	0.5495	0.0097
	64	0.5559	0.0092	**0.5418**	0.0115

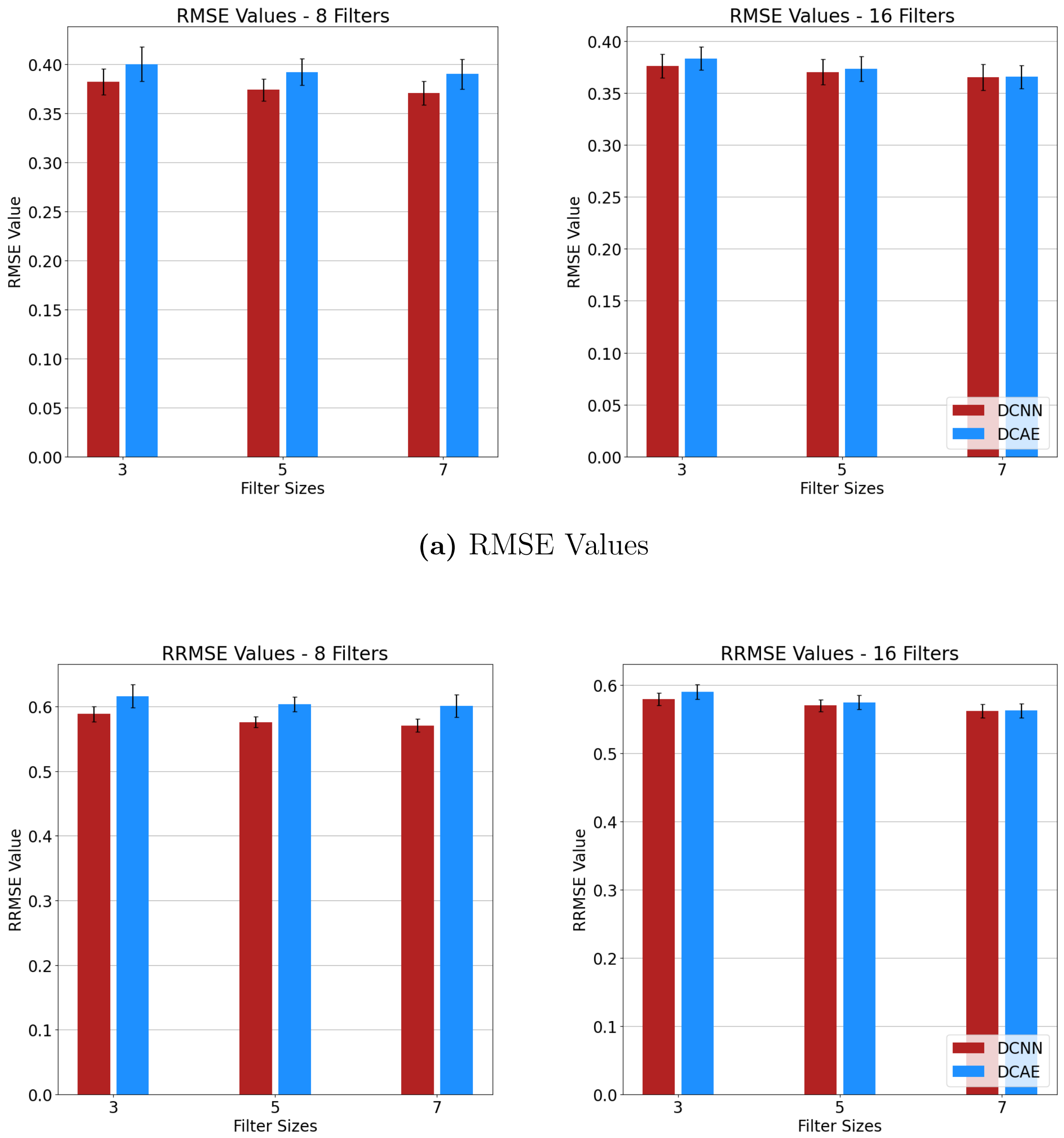


(a) RMSE Values



(b) RRMSE Values

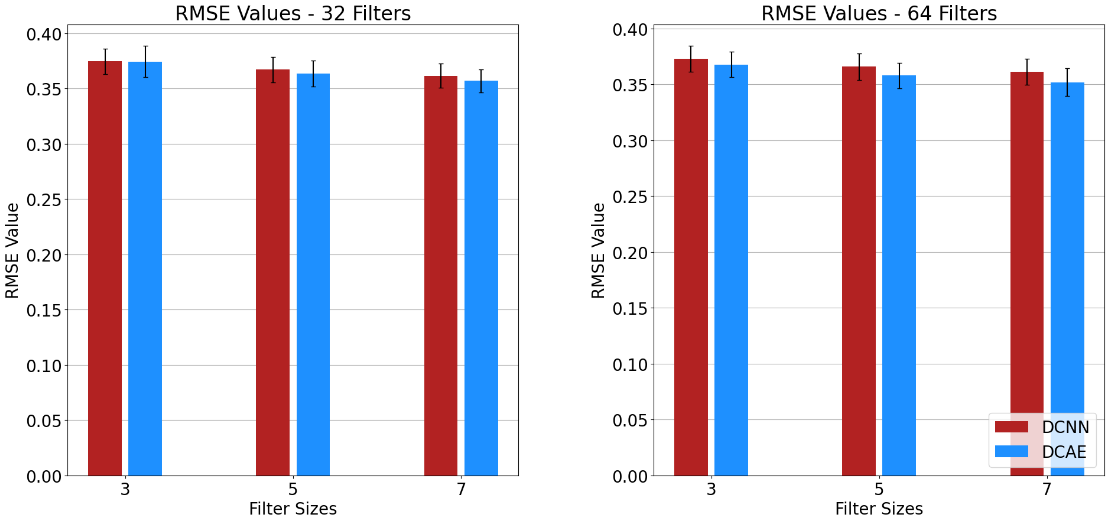
Figure 5.1: RMSE and RRMSE mean values for all models developed in the grid search.



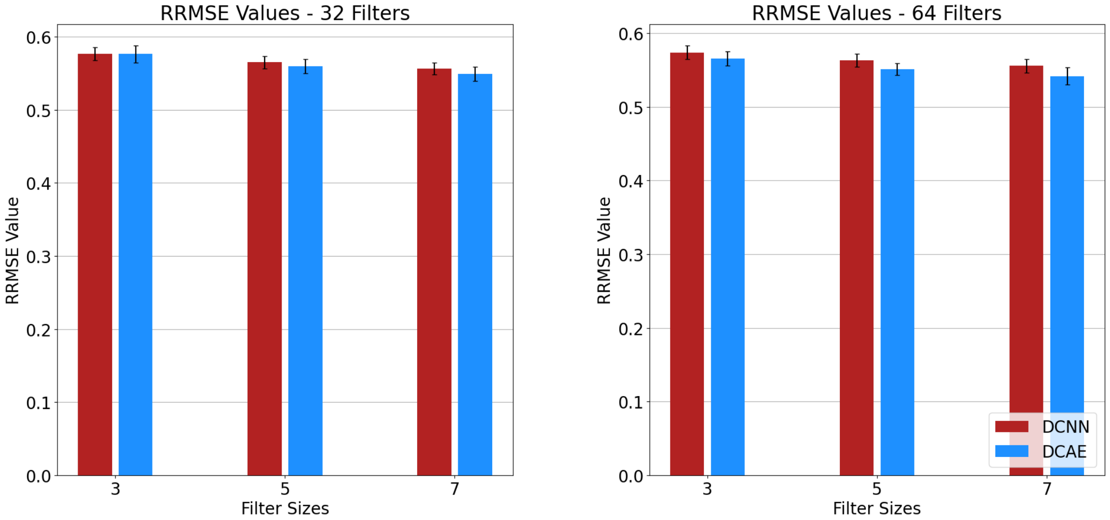
(a) RMSE Values

(b) RRMSE Values

Figure 5.2: Mean values for the architectures with 8 and 16 filters developed in the grid search.



(a) RMSE Values



(b) RRMSE Values

Figure 5.3: Mean values for the architectures with 32 and 64 filters developed in the grid search.

Table 5.3: First Approach Grid Search - Best Model Parameters

Type of Architecture	Filter Size	Number of Filters
DCAE	7	64

RRMSE values, representing the normalised errors, appear in Figure 5.5, and seem to generally follow the same trends as RMSE values. However, in the SC model, channels O1 and O2 present some of the lowest RRMSE values, even though their RMSE values are higher than in a lot of other channels. This indicates that channels O1 and O2 had expected higher root mean squared (RMS) values, meaning that the RMSE error value is less relevant in these channels. Comparing both artifact removal models, the MC architecture achieves better results in every channel, with the highest differences in channels O1 and O2, for both RMSE and RRMSE values.

PCC values are shown in Figure 5.6, showing the linear correlation between the original or reconstructed segments and the targets - a PCC value of 1 would be ideal since it meant the reconstructed segment was equal to its target. The developed DCAE SC architecture was able to achieve higher PCC values than the original segments, with channels Fp1, Fp2, F7 and F8 suffering the largest increases. The Fz channel, which has the highest PCC values among the original segments, is the only channel where SC model reconstructed segments show a decrease in this metric. In the majority of the channels, the upper quartile of the original segments is higher than in the SC model, and the lower quartile of this model is lower than the median of the original PCC value. Comparing the DCAE model to the MC architecture, the latter outperforms the former in every channel, with the highest differences appearing in the channels most heavily affected by ocular artifacts (Fp1, Fp2, F7 and F8).

As stated in Section 2.5, SNR$_{Diff}$ refers to the difference between the input and output Signal-to-Noise Ratios (SNRs), or the SNRs of the original noisy segment and the reconstructed segment (both in relation to the corresponding clean target segment). Figure 5.7 presents these values for each channel, with values of the SC model in blue, and of the MC model in orange. Although values for all channels are positive, meaning that the model is able to reconstruct signals with better SNR, 14 out of the 19 channels contain the zero value in their interquartile range. Channel Fz presents a median value that is very close to zero, with channels O1, O2, P7 and P8 also showing low values. Once more, the MC architecture is able to achieve better results in every channel, with the five aforementioned channels having the biggest gaps between the two performances.

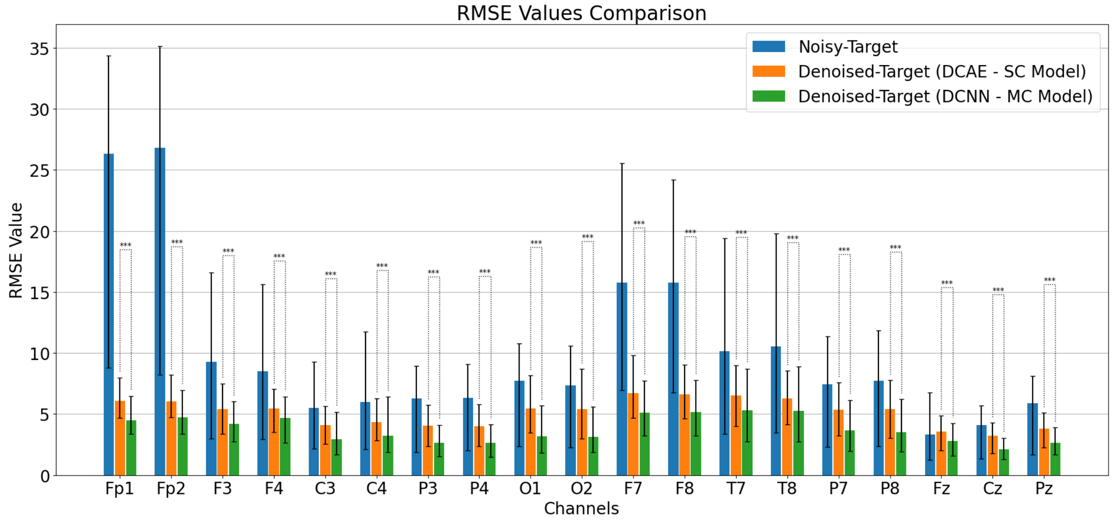


Figure 5.4: First approach RMSE performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

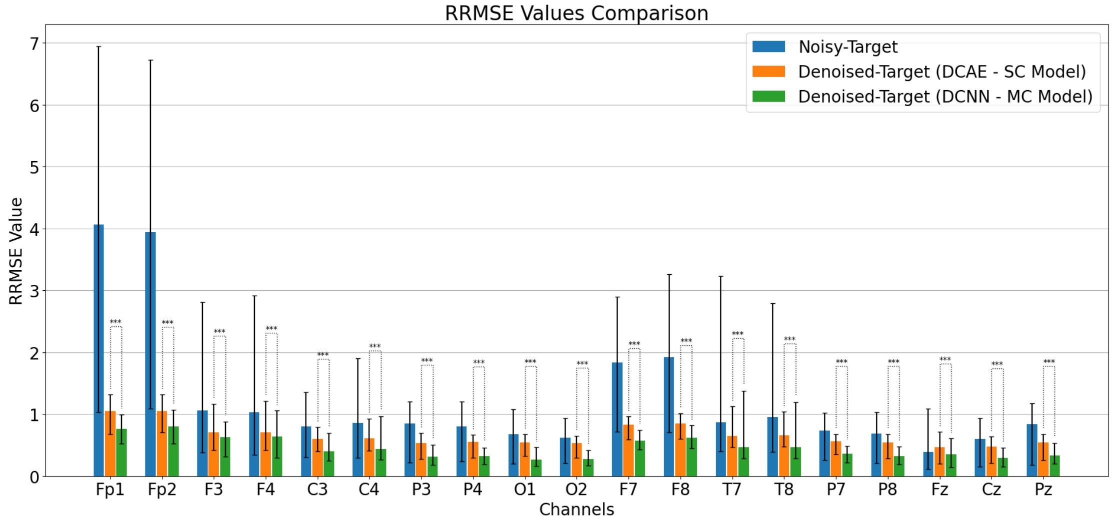


Figure 5.5: First approach RRMSE performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

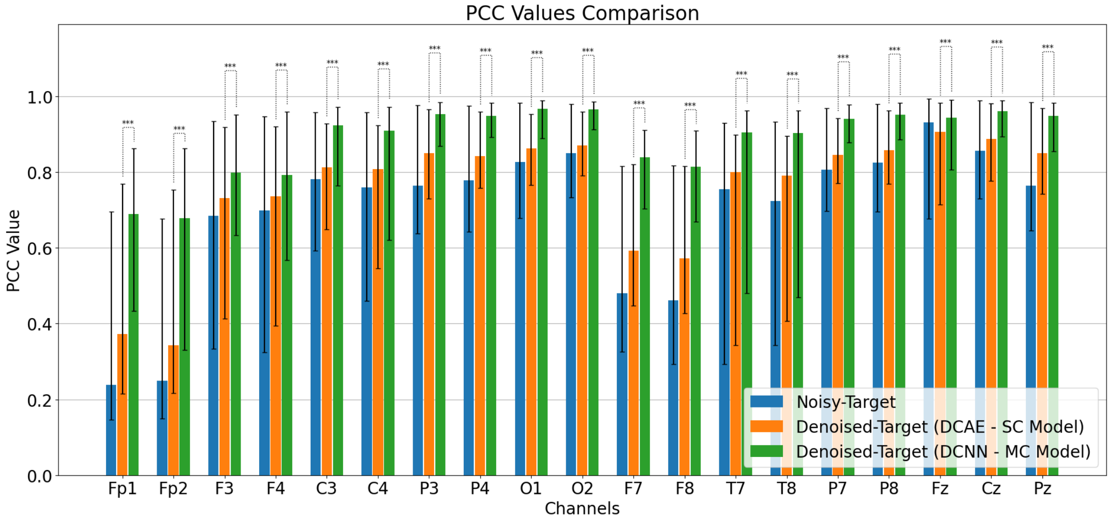


Figure 5.6: First approach PCC performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

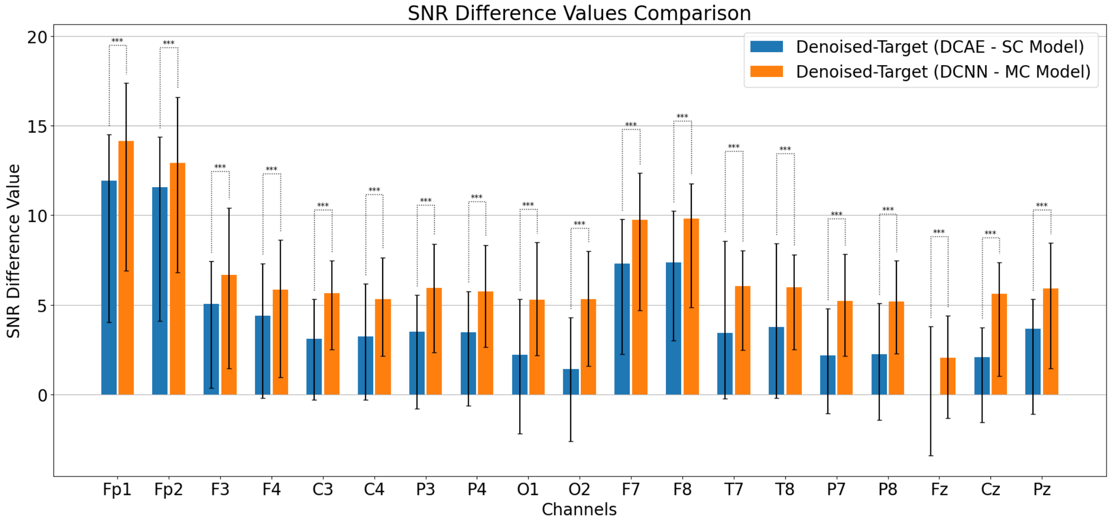


Figure 5.7: First approach SNR_{Diff} performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

Statistic validation was performed in order to check for significantly statistical differences between the results of the two compared models. Firstly, the Shapiro–Wilk [95] and Kolmogorov–Smirnov tests [96] were used to verify if the results followed a normal distribution. Since the p-value for both the SC model and the MC model results were inferior to 0.05, we reject the null hypothesis that the results followed a normal distribution, and used non-parametric tests to verify whether there were significant statistical differences. The Mann–Whitney U test was used [97], a test for independent samples whose null hypothesis is that the two samples come from the same population (i.e. that they both have the same median). We obtained a value of U equal to zero in every case, meaning that there are significant statistical differences between performances. Therefore, the performance of the MC approach is better than the performance of the SC model. This, along with every p-value being smaller than 0.05, allowed the rejection of the null hypothesis, confirming the existence of statistical differences between the results of the two models in every metric and for each channel. Asterisk symbols in Figures 5.4, 5.5, 5.6 and 5.7 refer to the significance levels of the test results - one symbol for a p-value smaller than 0.05, two for a p-value smaller than 0.01, and three for a p-value smaller than 0.001.

5.1.2.2 Artifact Removal and Signal Reconstruction

Examples of EEG channel segments with different types of artifacts were used to compare the artifact removal ability of the trained model, comparing the reconstructed signal to the original segment, the target segment and the signal cleaned by the DCNN MC architecture.

Figure 5.8 shows part of a EEG segment from the test dataset carrying an eye blink artifact, which is usually a high-amplitude low-frequency occurrence. We verify that although the developed model was able to remove the noise from the blink, it had some difficulties reconstructing the details of the target segment, a task that was better achieved by the MC architecture. The segment shown in Figure 5.9 includes an eye movement artifact. The SC architecture reduced the noise but, unlike the DCNN model, was unable to remove it completely, with a fraction of the artifact still present in the reconstructed signal.

The muscle activity present in the segment in Figure 5.10 was also largely attenuated by the model even if not entirely - it struggled to reconstruct the details of the target signal since residues of the artifacts remained, as is confirmed by the Power Spectrum Density (PSD), showing an overall higher density than the target segment across all frequencies.

Figure 5.11 displays pulse artifacts. In removing this type of artifact, the SC and MC models performed similarly, not being able to attenuate a big part of the noise, which is confirmed by both the time series and the PSD, where the density of both reconstructed signals, particularly in the higher frequencies, is still too high when compared to the target segment, with the MC model still performing slightly better.

The cardiac peaks in Figure 5.12 were not removed by either the SC or the MC model, both behaving similarly. This kind of artifact was not very common in the dataset, thus the models' limitations in classifying this artifact as noise.

These examples reflect the results of the evaluation metrics, with the SC model being able to largely reduce the noise in the signals, being however generally outperformed by the MC architecture.

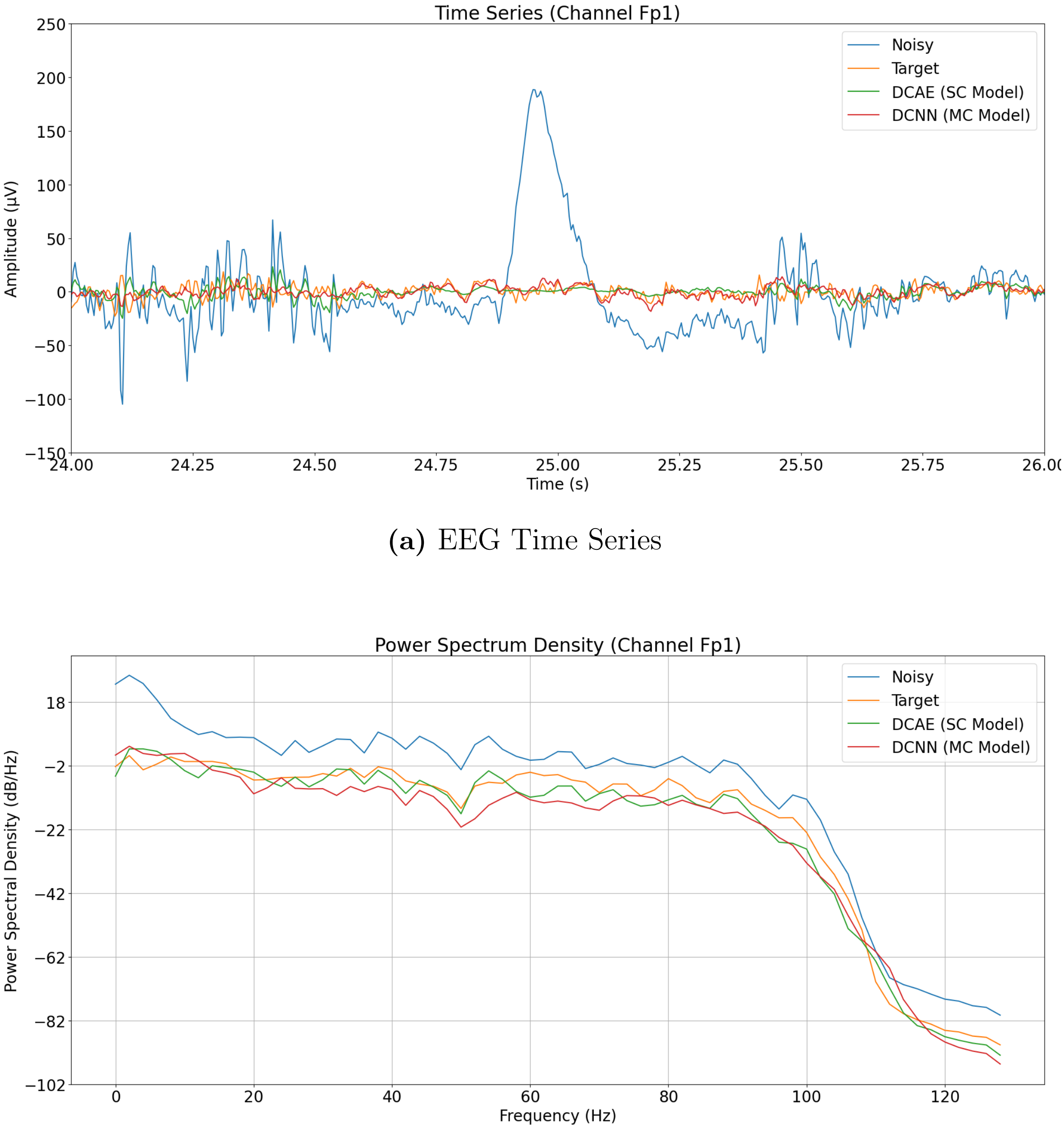
5.2 Second Approach

This section describes the results obtained in the second approach. Subsection 5.2.1 presents the outcomes of the grid search performed to find the best hyperparameters for both the DCAE and the DCAE with an incorporated BiLSTM layer (DCLSTMAE). Subsection 5.2.2 conducts a performance comparison between three models trained with 5-second segments (the architecture that achieved the best results in the grid search and two others retrieved from literature), as well as a comparison between the trained models for each of the two approaches.

5.2.1 Grid Search

The RMSE and RRMSE means and standard deviations for all the models created during the grid search, which used the preliminary training dataset, are shown in Tables 5.4 and 5.5, respectively.

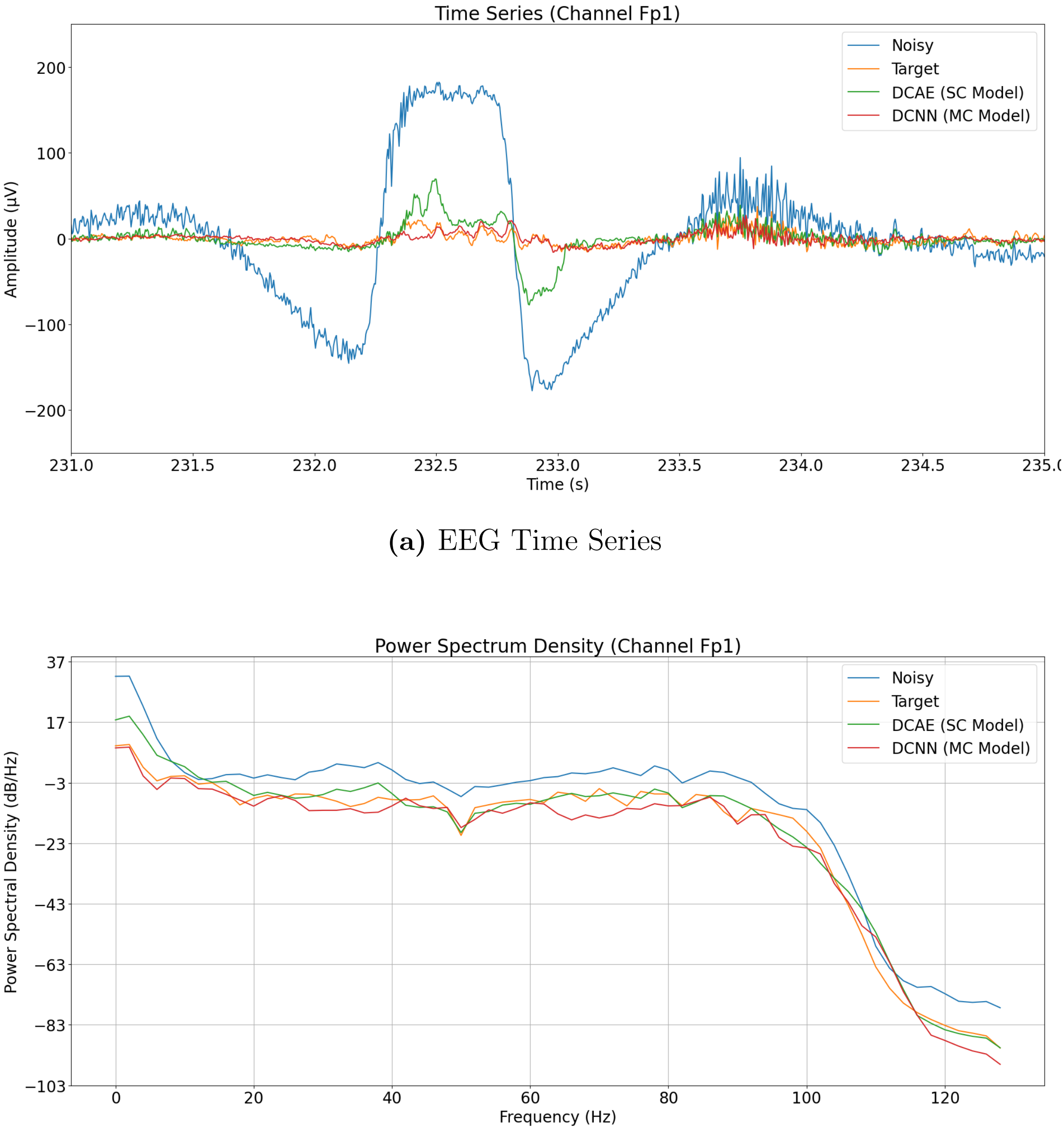
In simple DCAE architectures with 3 convolutional layers before the output layer of the latent space, RMSE and RRMSE values improve with the increase of the filter size, and with the increase in the number of filters from 64 to 128, but not always with the increase from 128 to 256 filters. As for the models with the Bidirectional LSTM (BiLSTM) layer, the decrease in values is more apparent with the rise of the number of filters. When the filter size is increased and the number of filters is left unchanged, the results deteriorate for architectures using 64 and 128



(a) EEG Time Series

(b) PSD

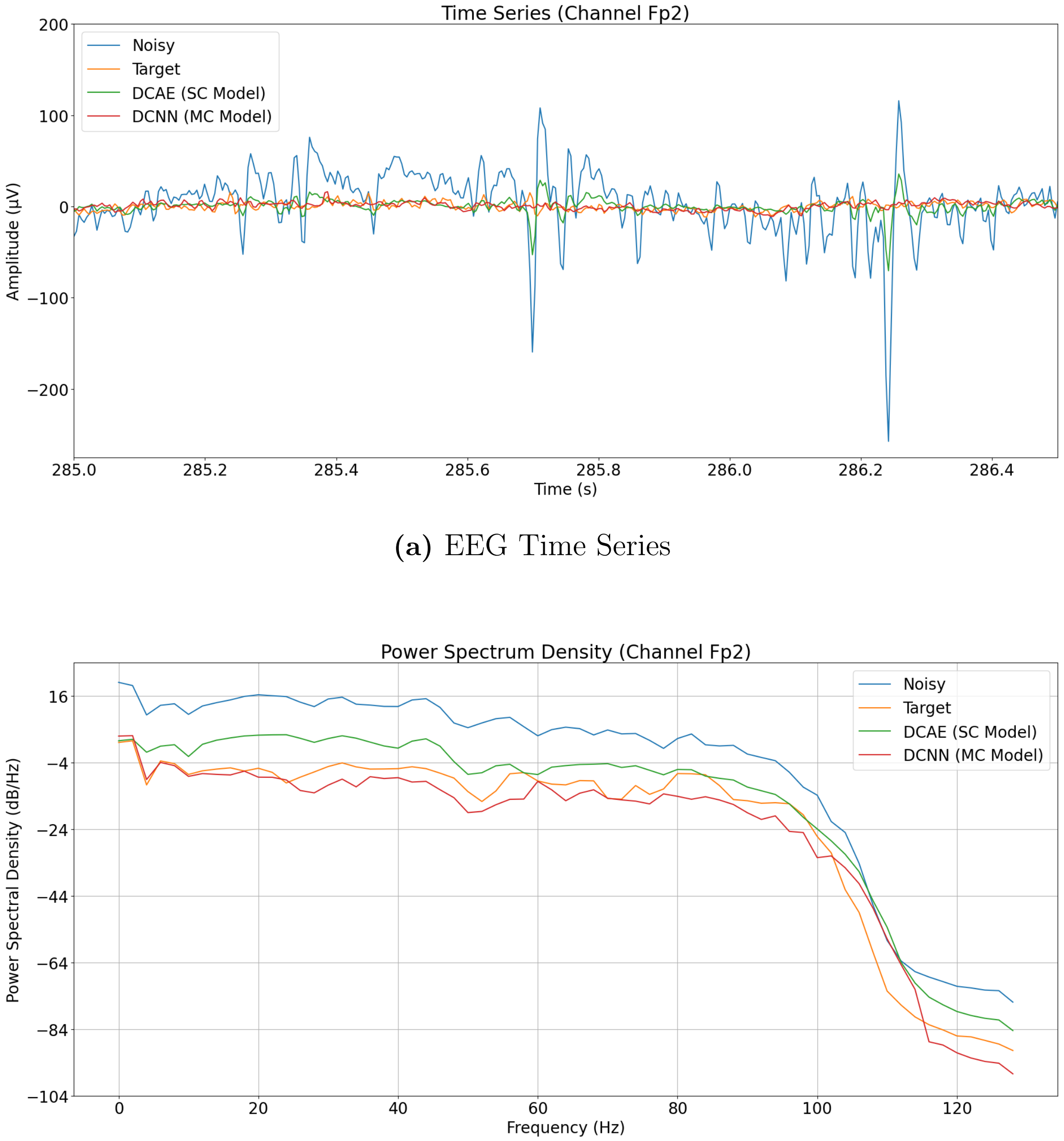
Figure 5.8: Example EEG segment from the test set containing eye blink artifact.



(a) EEG Time Series

(b) PSD

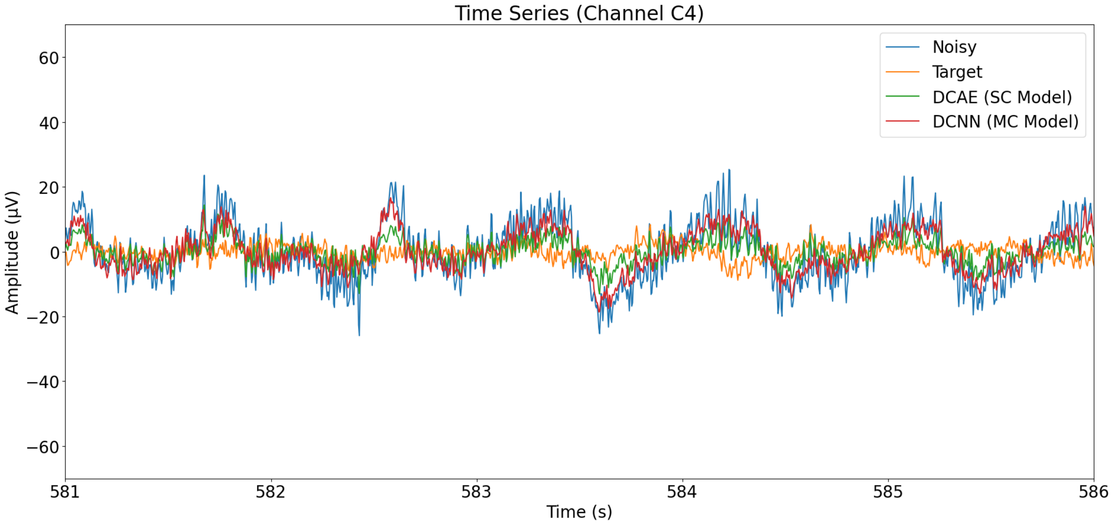
Figure 5.9: Example EEG segment from the test set containing eye movement artifact.



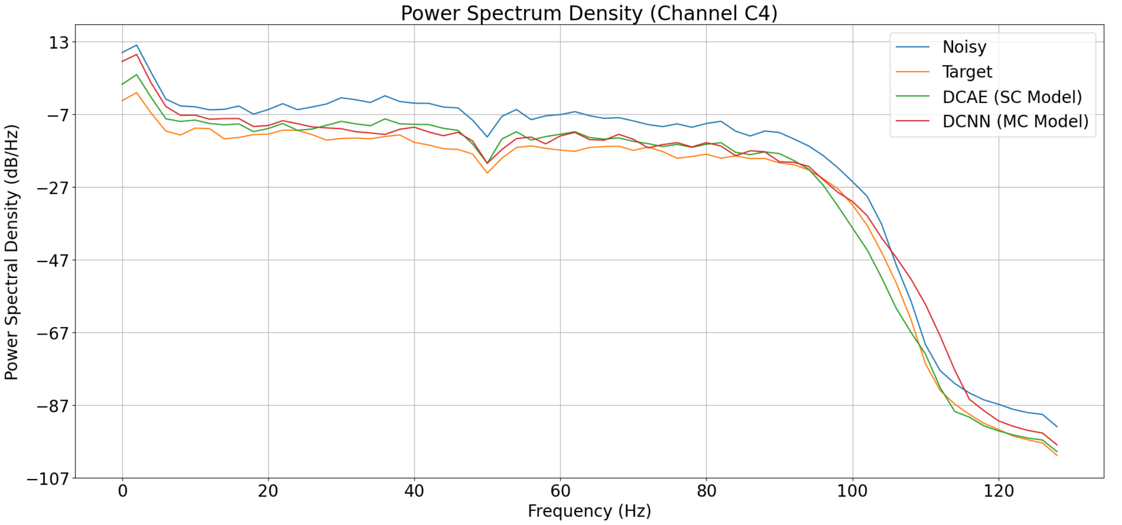
(a) EEG Time Series

(b) PSD

Figure 5.10: Example EEG segment from the test set containing muscle artifact.

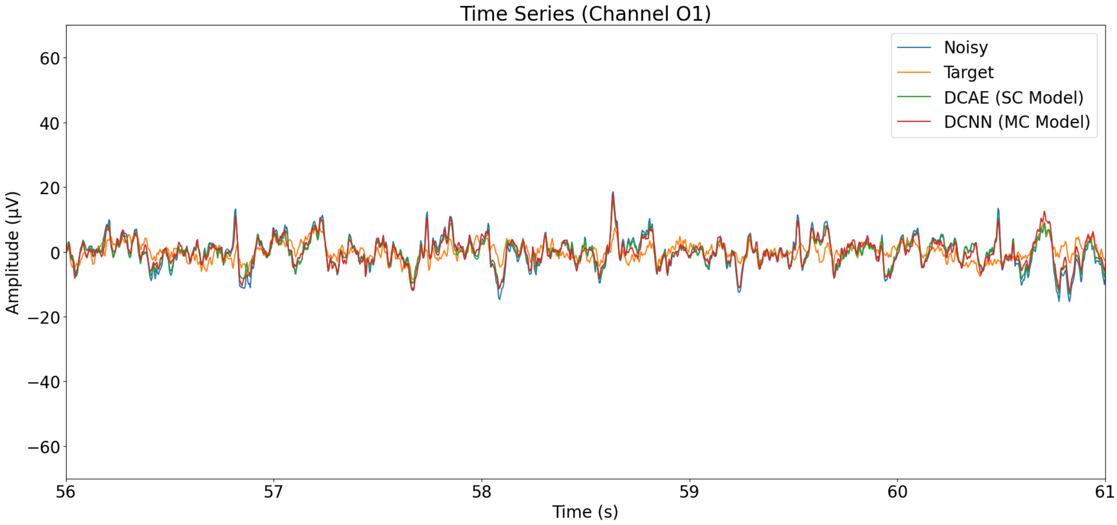


(a) EEG Time Series

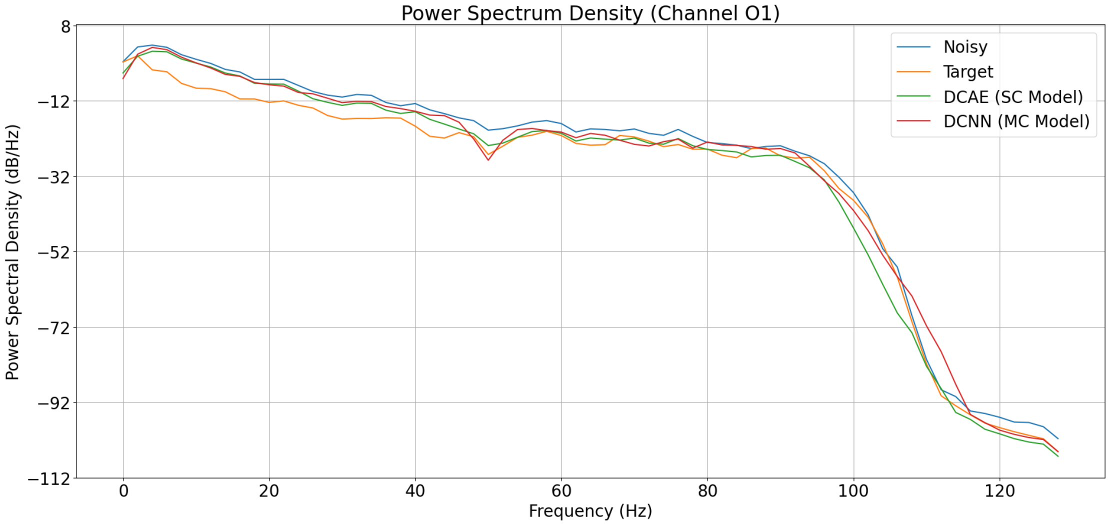


(b) PSD

Figure 5.11: Example EEG segment from the test set containing pulse artifact.



(a) EEG Time Series



(b) PSD

Figure 5.12: Example EEG segment from the test set containing cardiac artifact.

Table 5.4: RMSE results for DCAEs and DCLSTMAEs in second approach Grid Search

			DCAEs		DCLSTMAEs	
Filter Size	**Number of Filters**	**Number of Layers**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
3	64	3	0.3634	0.0035	0.3251	0.0042
		4	0.3572	0.0047	0.3224	0.0031
		5	0.3539	0.0057	0.3253	0.0035
	128	3	0.3548	0.0041	0.3197	0.0062
		4	0.3519	0.0039	0.3171	0.0038
		5	0.3447	0.0077	0.3197	0.0042
	256	3	0.3592	0.0037	0.3132	0.0041
		4	0.3508	0.0043	0.3134	0.0036
		5	0.3458	0.0045	0.3155	0.0029
5	64	3	0.3548	0.0041	0.3214	0.0041
		4	0.3504	0.0087	0.3196	0.0031
		5	0.3447	0.0077	0.3258	0.0069
	128	3	0.3533	0.0044	0.3163	0.0024
		4	0.3422	0.0043	0.3172	0.0053
		5	0.3358	0.0033	0.3172	0.0028
	256	3	0.3527	0.0040	0.3128	0.0047
		4	0.3428	0.0039	**0.3107**	0.0033
		5	0.3360	0.0038	0.3138	0.0040
7	64	3	0.3508	0.0041	0.3222	0.0063
		4	0.3433	0.0046	0.3182	0.0035
		5	0.3376	0.0046	0.3254	0.0047
	128	3	0.3479	0.0040	0.3169	0.0039
		4	0.3381	0.0044	0.3146	0.0044
		5	0.3386	0.0066	0.3170	0.0036
	256	3	0.3493	0.0044	0.3124	0.0035
		4	0.3415	0.0102	0.3120	0.0046
		5	0.3376	0.0069	0.3135	0.0022

Table 5.5: RRMSE results for DCAEs and DCLSTMAEs in second approach Grid Search

			DCAEs		**DCLSTMAEs**	
Filter Size	**Number of Filters**	**Number of Layers**	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
3	64	3	0.5840	0.0022	0.5225	0.0035
		4	0.5740	0.0038	0.5181	0.0034
		5	0.5687	0.0047	0.5228	0.0035
	128	3	0.5787	0.0021	0.5139	0.0109
		4	0.5656	0.0016	0.5096	0.0025
		5	0.5533	0.0025	0.5138	0.0045
	256	3	0.5722	0.0017	0.5034	0.0043
		4	0.5638	0.0028	0.5036	0.0044
		5	0.5557	0.0058	0.5071	0.0037
5	64	3	0.5702	0.0021	0.5166	0.0029
		4	0.5630	0.0114	0.5136	0.0023
		5	0.5539	0.0090	0.5236	0.0077
	128	3	0.5678	0.0027	0.5084	0.0025
		4	0.5499	0.0027	0.5098	0.0101
		5	0.5396	0.0042	0.5099	0.0038
	256	3	0.5667	0.0027	0.5026	0.0040
		4	0.5509	0.0047	**0.4994**	0.0031
		5	0.5401	0.0049	0.5043	0.0032
7	64	3	0.5637	0.0019	0.5177	0.0065
		4	0.5517	0.0052	0.5154	0.0019
		5	0.5425	0.0045	0.5230	0.0046
	128	3	0.5591	0.0025	0.5093	0.0048
		4	0.5434	0.0047	0.5056	0.0047
		5	0.5442	0.0063	0.5095	0.0033
	256	3	0.5614	0.0064	0.5021	0.0047
		4	0.5488	0.0165	0.5014	0.0046
		5	0.5426	0.0100	0.5039	0.0033

filters, but slightly improve with 256 filters - see Figures 5.13 and 5.14.

The same happens with architectures with 4 convolutional layers (Figures 5.15 and 5.16): in models without the BiLSTM layer, RMSE and RRMSE values get lower with the rise of the hyper-parameters, except when the number of filters is increased from 128 to 256. In models with the recurrent layer, a higher number of filters leads to better results, while a higher filter size can either improve or worsen the results, or even leave them unchanged.

When the number of convolutional layers is changed to 5 (Figures 5.17 and 5.18), simple DCAE models tend to, as before, show improvements with bigger filter sizes, but present inferior results when the number of filters is pushed from 128 to 256 (with filter sizes 3 and 5) and from 64 to 128 (with filter size 7). Models with the BiLSTM layer show however, better results with larger numbers of filters, while the size of said filters seems to hardly affect the RMSE and RRMSE values. Additionally, models with the BiLSTM layer consistently outperform those lacking it, for the same filter size and number of filters and layers.

In regards to the variation of the number of convolutional layers before the output layer of the latent space, the results in DCAE architectures with no recurrent layer are improved when the number of these layers is increased from 3 to 4 as well as from 4 to 5. When the BiLSTM layer is present, a boost from 3 to 4 convolutional layers leads to better results, but a further increase form 4 to 5 layers worsens them, which can be ssen in Figures 5.19 and 5.20.

According to the results, networks without the BiLSTM layer typically get better RMSE and RRMSE values with larger filter sizes. However, increasing the number of filters tends to have little to no effect or even impair the outcomes. The inverse occurs in networks with the BiLSTM layer: greater results are obtained with more filters, but increasing the size of the filters occasionally results in worse results. Regarding the quantity of convolutional layers, in architectures without the recurrent layer, the results get better as the number of layers rises; however, for models incorporating that layer, the models with the best RMSE and RRMSE values have 4 convolutional layers before the output layer of the latent space. Furthermore, for every hyper-parameter variation, the presence of the BiLSTM layer in the model leads to better results.

The architecture with the best RMSE and RRMSE mean values was the one chosen to be trained with the entire new training dataset. Its parameters can be observed in 5.6.

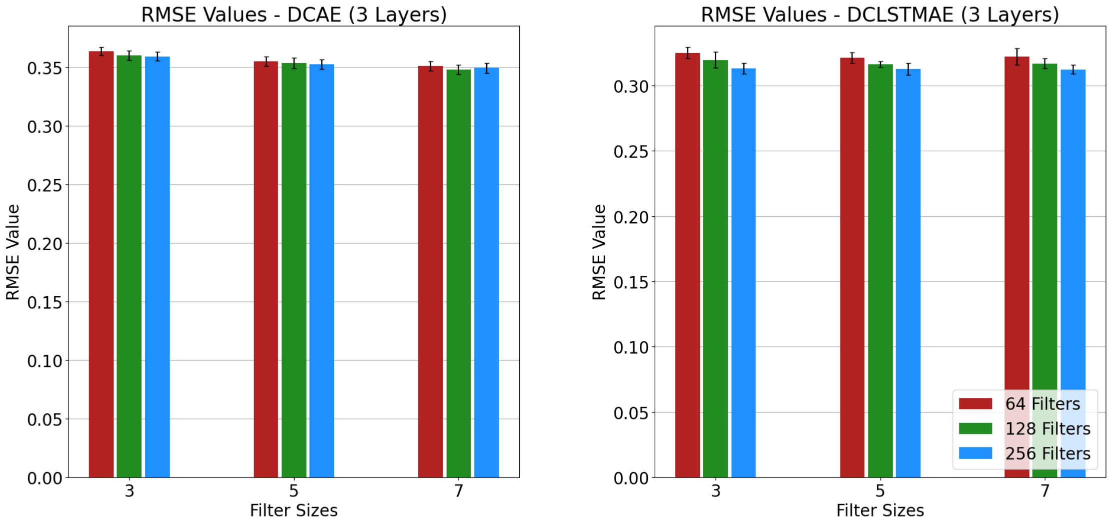


Figure 5.13: RMSE values for architectures with 3 convolutional layers before the output layer of the latent space.

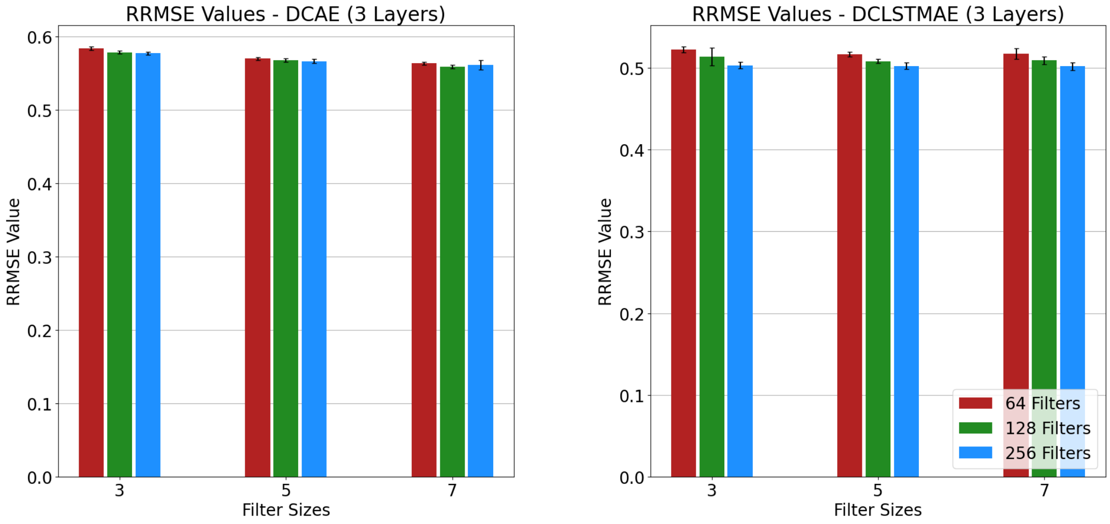


Figure 5.14: RRMSE values for architectures with 3 convolutional layers before the output layer of the latent space.

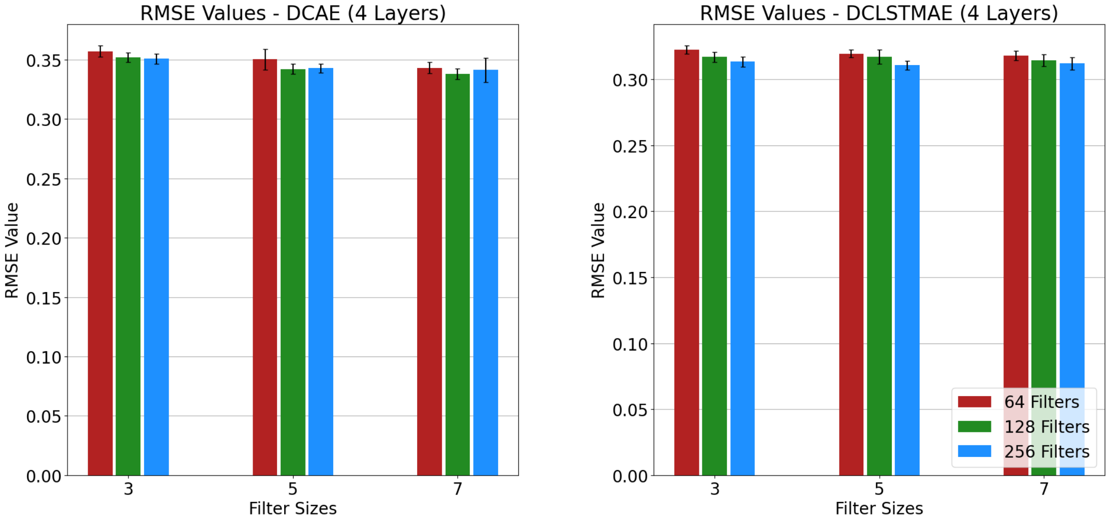


Figure 5.15: RMSE values for architectures with 4 convolutional layers before the output layer of the latent space.

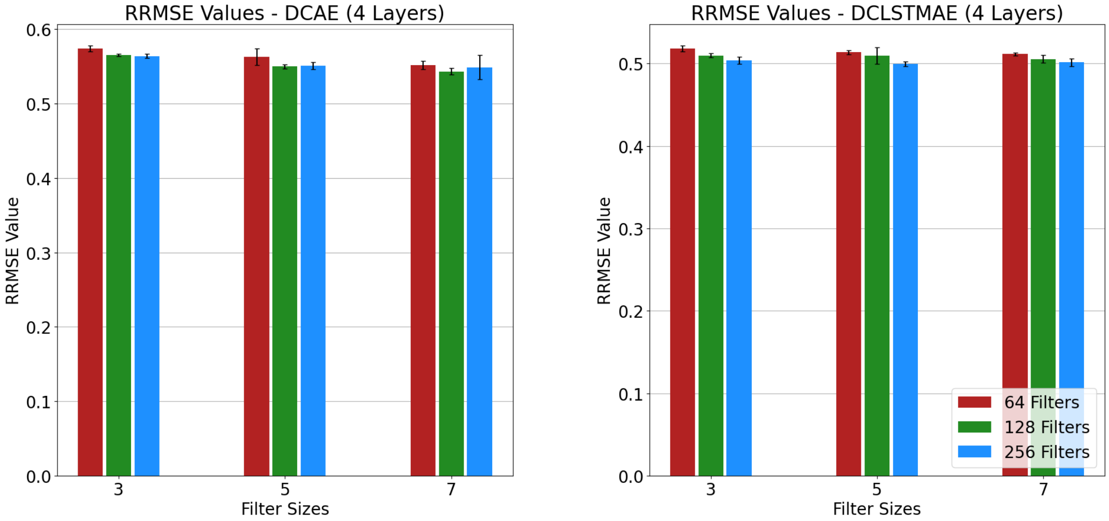


Figure 5.16: RRMSE values for architectures with 4 convolutional layers before the output layer of the latent space.

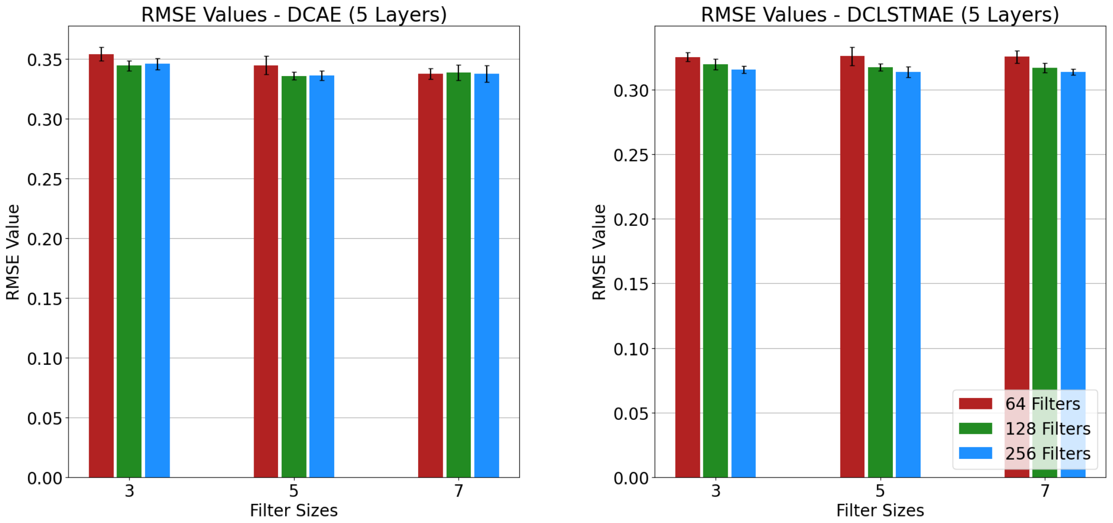


Figure 5.17: RMSE values for architectures with 5 convolutional layers before the output layer of the latent space.

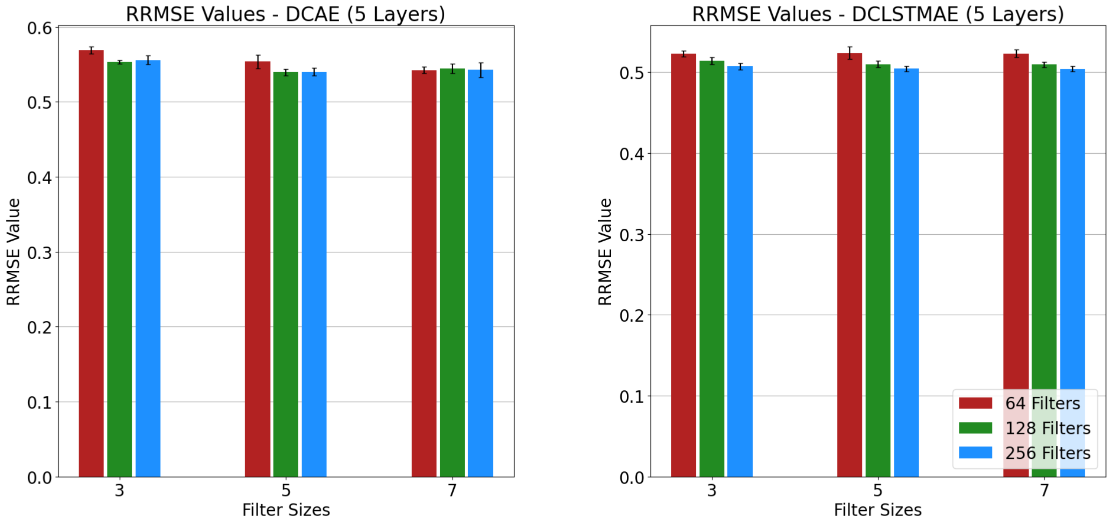


Figure 5.18: RRMSE values for architectures with 5 convolutional layers before the output layer of the latent space.

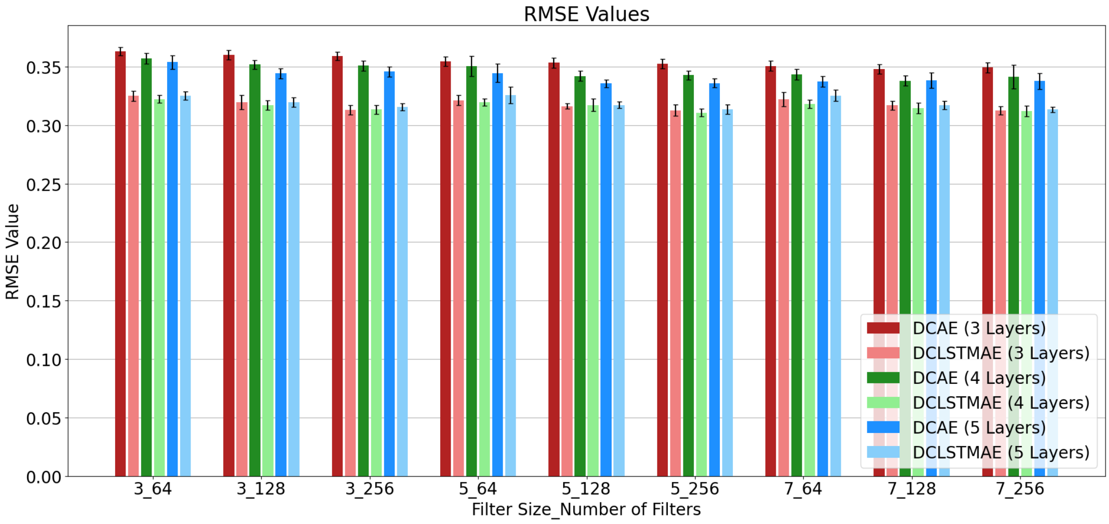


Figure 5.19: RMSE mean values for all architectures developed in the grid search.

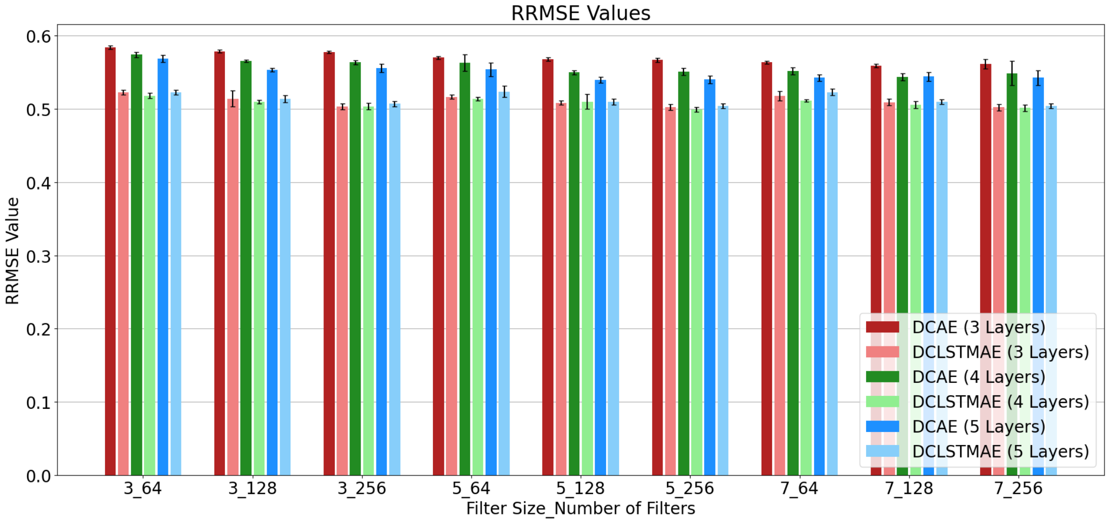


Figure 5.20: RRMSE mean values for all architectures developed in the grid search.

Table 5.6: Second Approach Grid Search - Best Model Parameters

Type of Architecture	Filter Size	Number of Filters	Convolutional Layers
DCLSTMAE	5	256	4

5.2.2 Model Testing

After training, the models were applied to the test dataset. We evaluated these using two methods: computing the evaluation metrics, which were compared and discussed in Subsubsection 5.2.2.1, and reconstructing various segments to visually evaluate the artifact removal capability of the architectures. These can be seen in Subsubsection 5.2.2.2.

5.2.2.1 Evaluation Metrics

The first method used for determining the evaluation metrics was by calculating their values for each 5-second window of the test dataset. Concerning the RMSE, RRMSE, and PCC metrics, barplots present the values of the original segments and of the signals reconstructed by each of the three trained models, for each channel - Figures 5.21, 5.22 and 5.23. Regarding the SNR$_{Diff}$, the barplot in Figure 5.24 shows the values of each channel regarding the three architectures. Once again, medians and interquartile ranges were used as central tendency statistics.

Regarding RMSE and RRMSE values, shown in Figures 5.21 and 5.22, our model (DCLSTMAE) achieved the lowest values, and thus the best results out of the three. Every channel presents lower error values compared to the ones of the original segments, apart from channels Fz and Cz, which incidentally are the channels with the lowest values overall. The DCNN model also lowers the RMSE and RRMSE in all channels apart from Fz and Cz, but its results are higher than the results from the DCLSTMAE model. The One-Dimensional Residual Convolutional Neural Network (1D-ResCNN) model performs the worst out of the three, only lowering the original error values in 8 out of the 19 channels. Although it normally presents the highest error value of the three architectures for each channel, in channels Fp1 and Fp2 its values are lower than the DCNN's (but still higher than the DCLSTMAE's).

PCC values in Figure 5.23 show that the DCLSTMAE model achieves higher values than the original segments in 12 of the 19 channels, with the biggest increases upon the original values in the Fp1 and Fp2 channels and the biggest reductions in the Fz and Cz channel, with multiple channels presenting values that do not differ greatly from the originals. The pattern of the DCNN is similar to the DCLSTMAE, although it achieves slightly worse results, being only able to improve upon the original values in channels Fp1 and Fp2. The 1D-ResCNN is unable to improve PCC values on any of the channels.

Figure 5.24 presents the SNR_{Diff} values. The DCLSTMAE and DCNN models achieve positive results for every channel except Fz and Cz, with the DCLSTMAE consistently achieving higher values. The 1D-ResCNN model shows positive values for 9 of the 19 channels. In channels Fp1, Fp2, F7 and F8, this architecture's results are higher than the DCNN but lower than the DCLSTMAE, and in the other 5 channels, are the lowest of the three models.

Statistic validation was once again performed, firstly between the DCLSTMAE model to both the 1D-ResCNN and the DCNN architectures. The p-values for the the Shapiro–Wilk and Kolmogorov–Smirnov tests across all metrics and channels were inferior to 0.05 in all models, meaning that non-parametric tests should be used. Since we were in the presence of three independent groups, the Kruskal-Wallis test [98], considered to be the non-parametric equivalent of the One-Way ANOVA, and the Dunn-Šidák test [99] were used to determine whether there were statistically significant differences between their medians. We the used the MATLAB tool *multcompare* to perform pairwise comparisons between the DCLSTMAE and the other two models. Significance levels of 0.05, 0.01 and 0.001 were used, and the results can be observed in the number of asterisk symbols of Figures 5.21, 5.22, 5.23, and 5.24. Statistical differences were found for every metric in every channel between our model and the other two.

Next, the results of the DCLSTMAE (the architecture with the best performance) were compared to the results of the architecture developed during the first approach. To this end, it was necessary to recalculate the metrics: instead of determining them for each 5-second segment prediction, these smaller segments were re-concatenated in the 10-minute segments used in the first approach, and the evaluation metrics were then determined for these longer signals.

Figures 5.25 and 5.26 compare RMSE and RRMSE values, and show that the DCLSTMAE achieves similar results to the DCAE developed in the first approach, with slight improvements in channels Fp1, Fp2, F7 and F8. As in the first approach architecture, channel Fz is the unique channel where the reconstructed error is higher than the original value. PCC values between the two models in Figure 5.27 are also similar, with the DCLSTMAE only surpassing the DCAE in channel Fp1. The Fz channel is once again the only channel where the original segment has a higher PCC

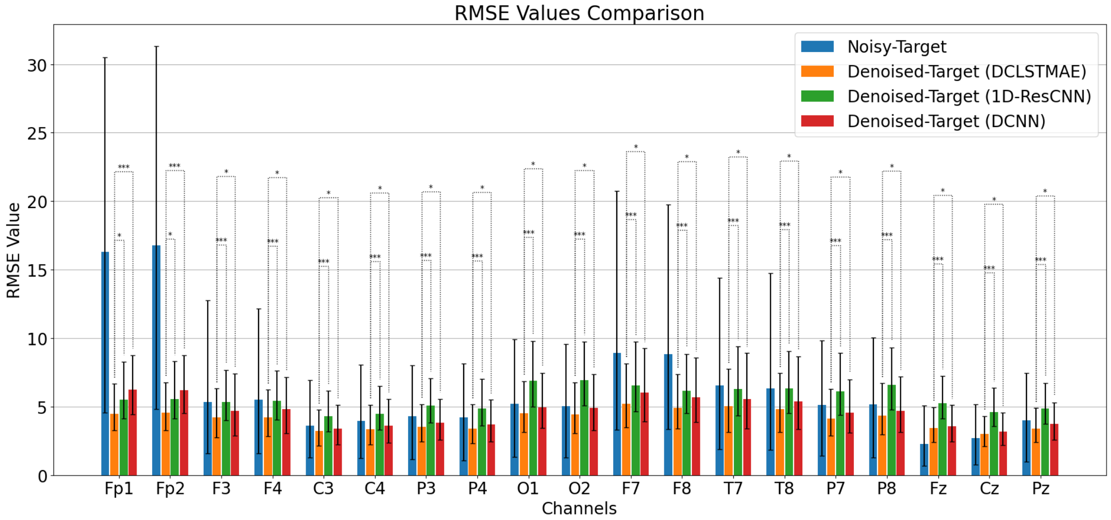


Figure 5.21: Second approach RMSE performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

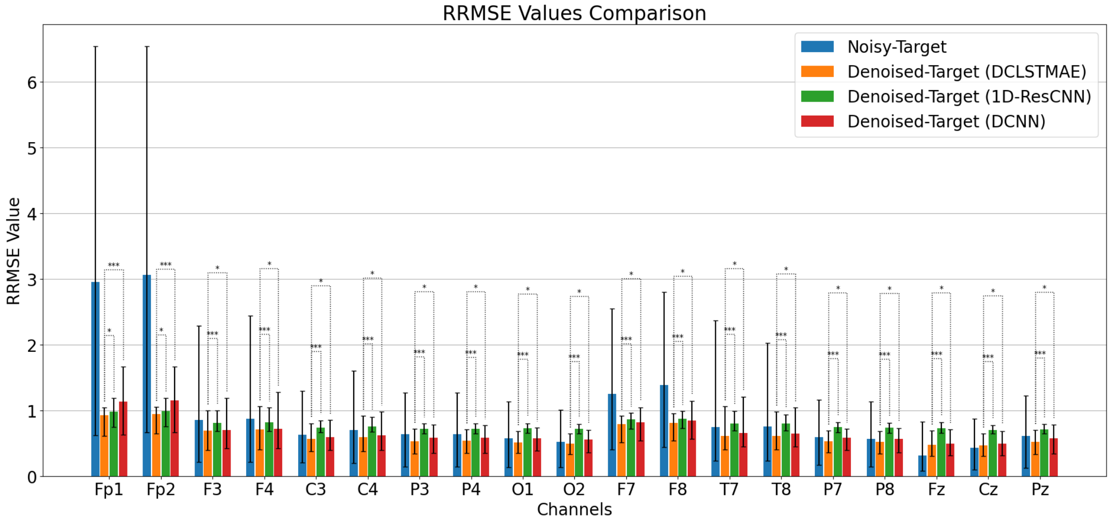


Figure 5.22: Second approach RRMSE performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

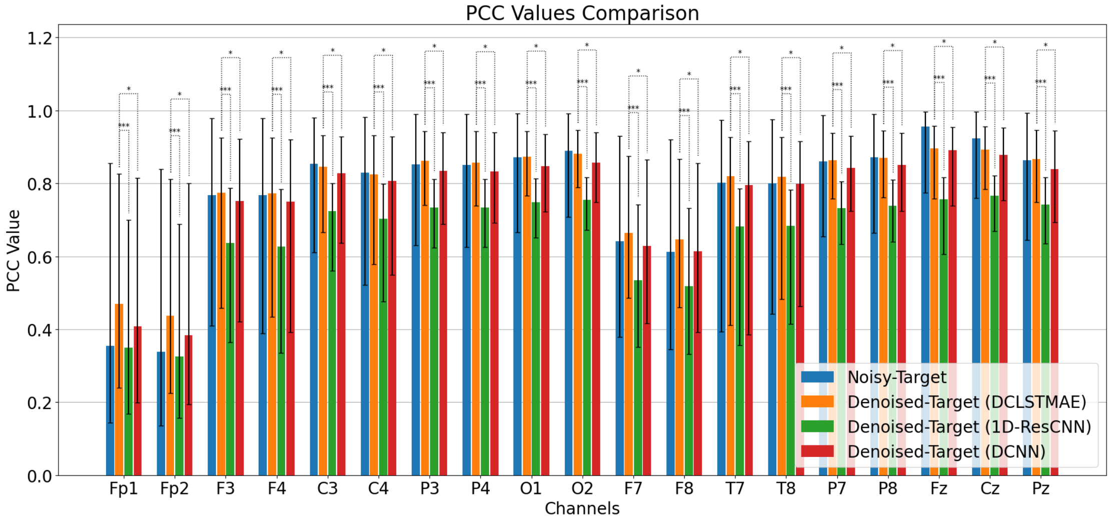


Figure 5.23: Second approach PCC performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

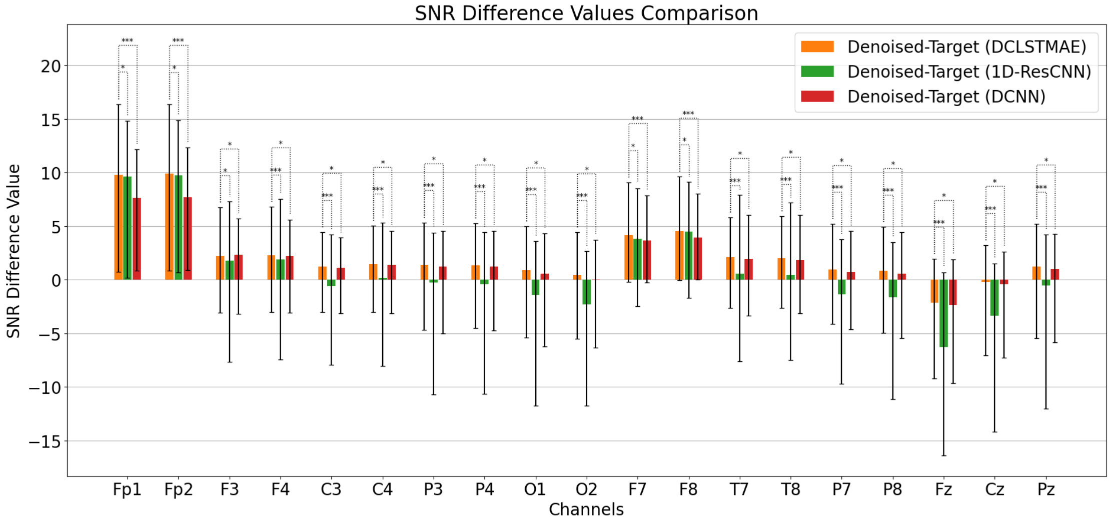


Figure 5.24: Second approach SNR$_{Diff}$ performance results. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha =$ 0.01, and three if they were found using $\alpha = 0.001$.

value than either reconstructed signal. Observing the SNR$_{Diff}$ values in Figure 5.28, the DCLSTMAE model shows slightly higher values in channels Fp1, Fp2, O2 and F8. However, the SNR$_{Diff}$ value for channel Fz is negative, which is not the case for the DCAE architecture.

Since we knew from previous tests that distributions of the DCLSTMAE model, as well as the ones from the DCAE model and the MC model from the first approach did not follow a normal distribution, the Kruskal-Wallis was once again used, followed by a pairwise comparison using the *multcompare* with the Dunn-Šidák test. These results are shown in Figures 5.25, 5.26, 5.27, and 5.28, with the number of asterisk symbol referring to the significance levels of 0.05, 0.01, and 0.001, respectively. Comparing our model to the MC architecture, statistical significant differences were found for every channel in every metric. However, several channels did not present significant differences between the SC models: channel F4 did not present differences in RMSE values; in RRMSE values, only channels Fp1 and Fp2 showed significant statistical differences; in PCC values, channels Fp1, Fp2 and F8 did not present statistical differences; and channels O1, O2, F7 and F8 did not show statistical differences in SNR$_{Diff}$ values.

Clear differences show up in the evaluation metrics results when they are determined for 5-second signals and for 10-minute segments, both for the original and for the reconstructed signal values.

When RMSE and RRMSE values are determined for 10-minute segments, they go up in every channel. However, this increase is much more sharp in the original segments, resulting in the model providing a bigger noise reduction. The reconstructed error in channel Cz becomes smaller than the error in the original signals (something that did not happen before).

The PCC results present the biggest differences when determined for longer signals (compared to the 5-second segments). Values decrease for both original and reconstructed segments in every channel, but the reduction is larger in original segments. In channels where, before, the original values surpassed the ones generated by the model, the reconstructed signals' PCC values become higher than the original segments' - the DCLSTMAE model achieves higher PCC values than the original segments in every channel apart from Fz.

Much like RMSE and RRMSE, SNR$_{Diff}$ values increase in every channel, with channel Cz going from a negative to a positive value.

To try to understand these differences, histograms were constructed represent-

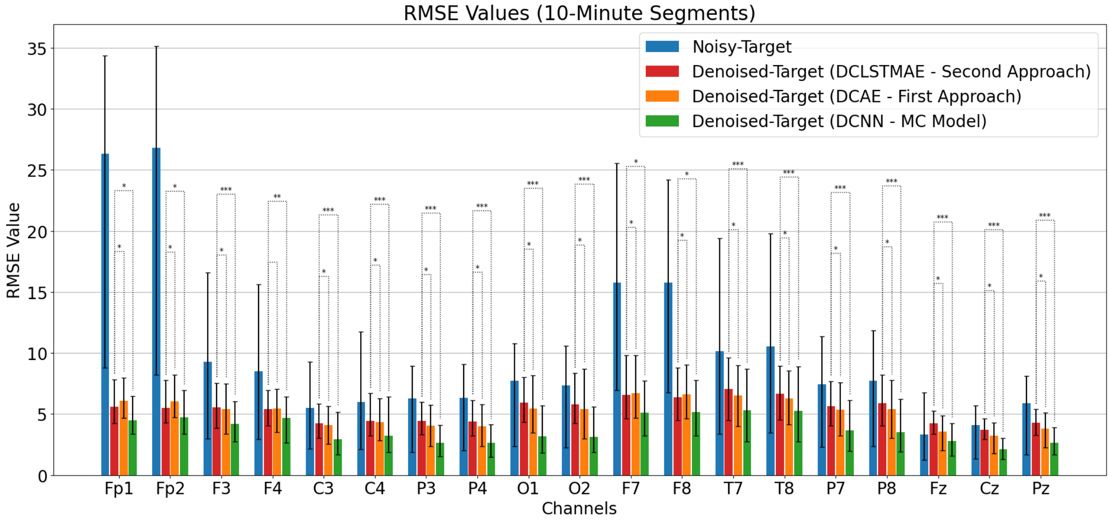


Figure 5.25: RMSE performance results comparing both approaches. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

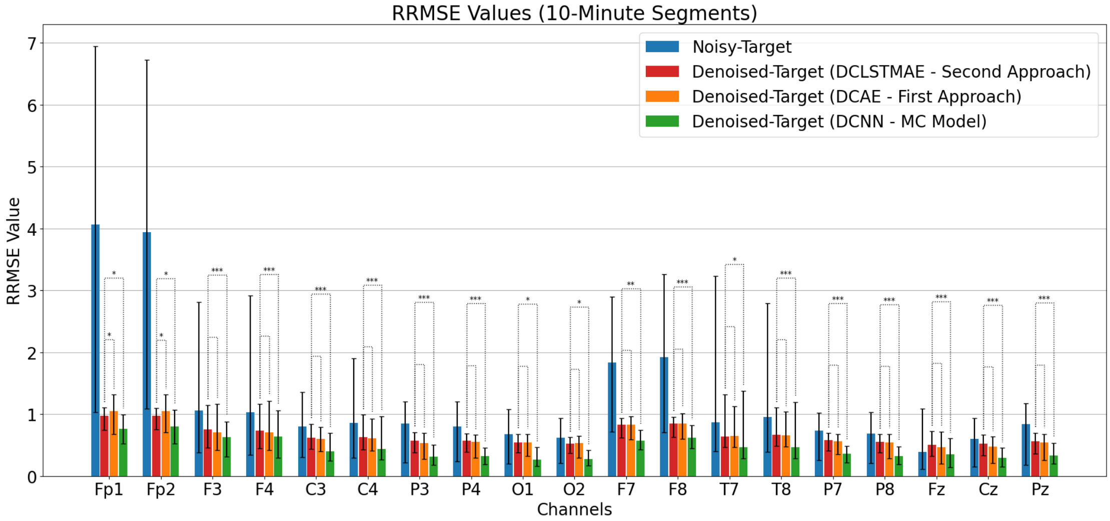


Figure 5.26: RRMSE performance results comparing both approaches. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

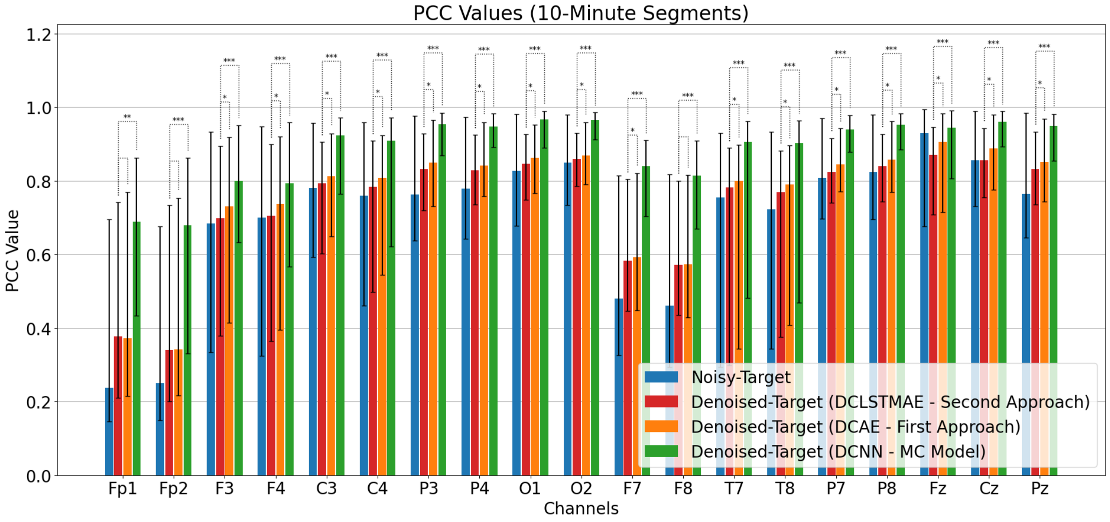


Figure 5.27: PCC performance results comparing both approaches. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

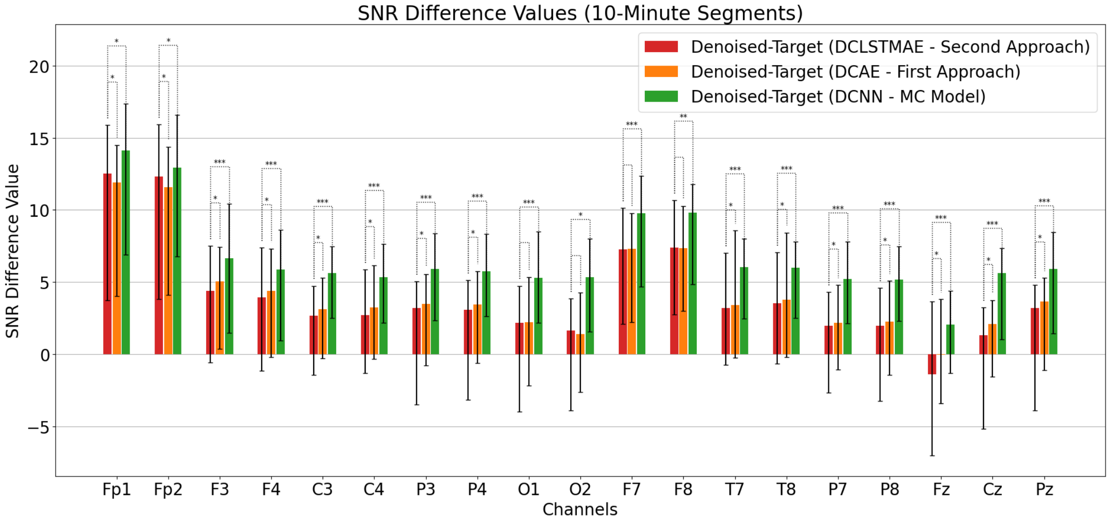


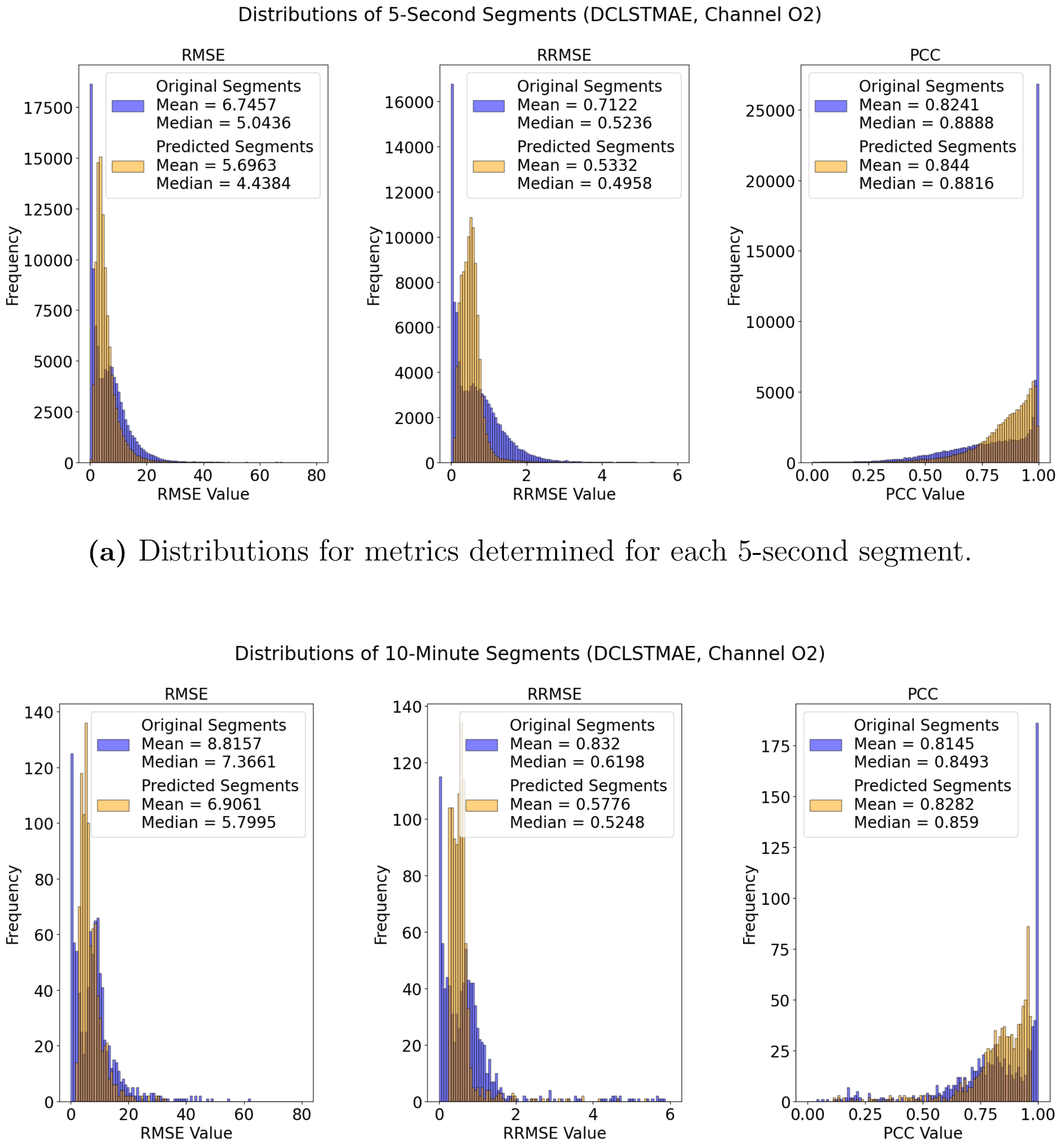
Figure 5.28: SNR$_{Diff}$ performance results comparing both approaches. Each bar corresponds to the median value and the interquartile range. Asterisk symbols refer to the significance levels in statistical tests results - one symbol if significant statistical differences were found using $\alpha = 0.05$, two if they were found using $\alpha = 0.01$, and three if they were found using $\alpha = 0.001$.

ing the distributions of each metric, both for the original segments and the reconstructed signals of each channel, firstly for values calculated from 5-second segments and then from 10-minute segments. In Figure 5.29, these distributions can be observed for channel O2, which is one of the channels where the PCC values for the reconstructed signals are lower than the originals when the metric is determined for 5-second segments, but higher when determined for 10-minute segments.

Ten-minute clean segments, to which no Independent Components (ICs) were removed during the visual analysis phase, when divided into 5-second windows, generate 120 segments all with a perfect PCC. Noisy segments however, once divided, can generate both windows with a very low PCC value and others with a very high PCC value, meaning that there will be a higher number of original segments with a high PCC value, affecting its average statistics. Although PCC is the metric with the highest differences, the same phenomenon (that can be observed in the aforementioned barplots) happens to the error values, which are higher when the metrics are determined for 10-minute segments rather than 5-second ones.

Exploring this matter even further, for each of the 10-minute segments in the test dataset, the evaluation metrics values for the entire segment were compared to the mean of the values of all 5-second signals that compose that segment. This comparison was performed using a line graph for each channel and each metric, where each point in the x-axis corresponds to one of the dataset segments, and the four values for that point represent the original 10-minute segment value, the predicted 10-minute segment value, the mean of all 5-second original segments values, and the mean of all 5-second predicted segments values. Figure 5.30 shows the PCC and RMSE line graphs for segments 800 to 850 in the dataset, since there are several segments of interest in this interval.

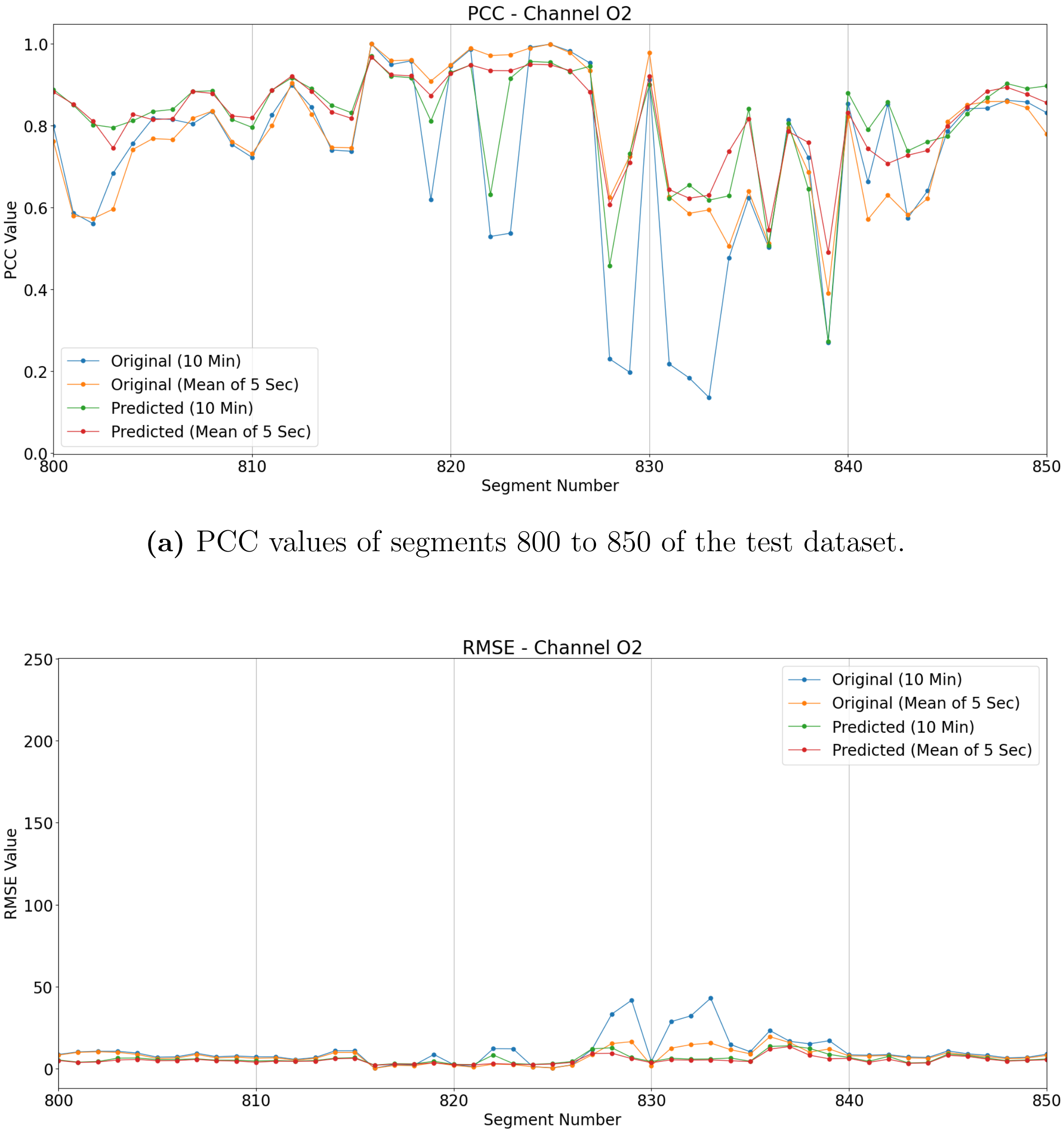
Analyzing the line graphs for the channels where the PCC differences are the highest, a selection of segments can be found where the relation of original-predicted segments values of the 10-minute segments is very different from that relation in the mean of all 5-second segments. Although these differences are also present in the error values, are much more sharp in the PCC metric. Plotting a few of these segments such as segment 822, which can be seen in Figure 5.31, there is a clear pattern: these segments show one or two short but very strong peaks of noise, in the middle of an otherwise almost noiseless signal. This leads to a lot of 5-second segments with very high PCC values and relatively low error values and a few windows with very low PCC and high errors, bringing the PCC especially down when this metric is determined for the entire 10-minute segment.



(a) Distributions for metrics determined for each 5-second segment.

(b) Distributions for metrics determined for each 10-minute segment.

Figure 5.29: Distributions for RMSE, RRMSE and PCC metrics for channel O2 segments, both the original signals and the signals reconstructed by the DCLSTMAE.



(a) PCC values of segments 800 to 850 of the test dataset.

(b) RMSE values of segments 800 to 850 of the test dataset.

Figure 5.30: Zoomed in line graphs representing the different PCC and RMSE values for each 10-minute segment in the test dataset.

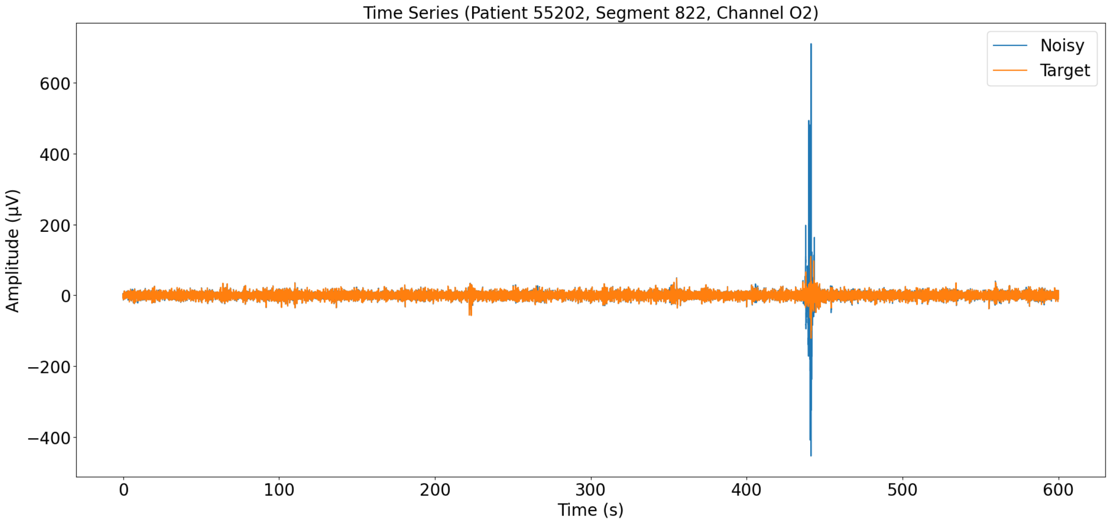


Figure 5.31: Example of segment with very different PCC values.

5.2.2.2 Artifact Removal and Signal Reconstruction

The same noisy examples of the previous approach were again used, firstly to compare the performances of the three architectures developed for 5-second segments, and then to compare the reconstruction abilities of the DCLSTMAE model to the DCAE model of the first approach.

Observing the segment with an eye blink artifact in Figure 5.32, the three architectures are able to reduce the noise, with the 1D-ResCNN model performing the worst and the DCLSTMAE model the best, even though this architecture still shows difficulties in reconstructing the details of the signal, with the PSD showing lower values than the target segment across all frequencies. This behavior is coherent with the first approach, since the model behaves very similarly to the first approach DCAE, which also had this limitation - see Figure 5.33.

In the segment showcasing eye movement, in Figure 5.34, as in the previous segment, although the DCLSTMAE can reduce the noise better than the 1D-ResCNN and DCNN architectures, it does it excessively, again not being able to reconstruct the finer details and presenting a lower PSD. Nevertheless, it can be seen in Figure 5.35 that it does perform better than the first approach architecture, which can only partially remove the noise from the artifact.

Figures 5.36 and 5.37 showcase a segment presenting muscle activity. The DCNN performs the worst, removing less noise than the other two architectures and with its PSD showing values closer to the original segment than to the target signal. The other two models present similar results. The DCLSTMAE model performs better than the first approach DCAE, which still maintains a fraction of the noise and has PSD levels closer to the original segment.

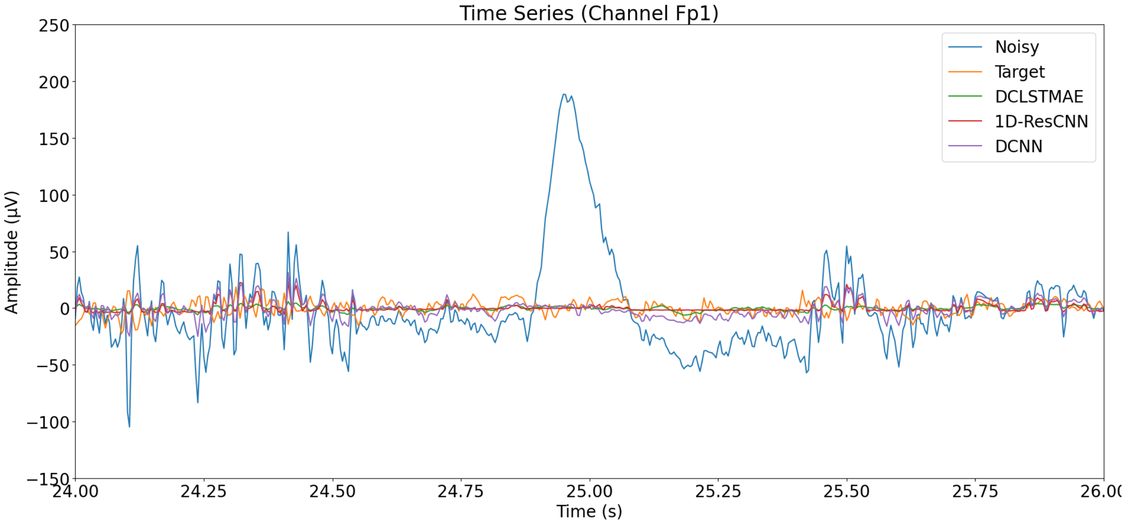
The segment with the pulse artifact in Figure 5.38 shows that the DCLSTMAE and the DCNN models present the same behaviour, with the former being able to reduce a larger fraction of the noise, while the 1D-ResCNN does not faithfully reconstruct the signal. The DCLSTMAE performs similarly to the DCAE model from the first approach, if not slightly better (Figure 5.39).

In Figures 5.40 and 5.41, representing a segment with cardiac artifact, all three new architectures, as well as the model from the first approach and the MC model, were unable to remove the noise (with the 1D-ResCNN model performing slightly better than the others), which once again can be explained by the lack of examples of this artifact in the used dataset.

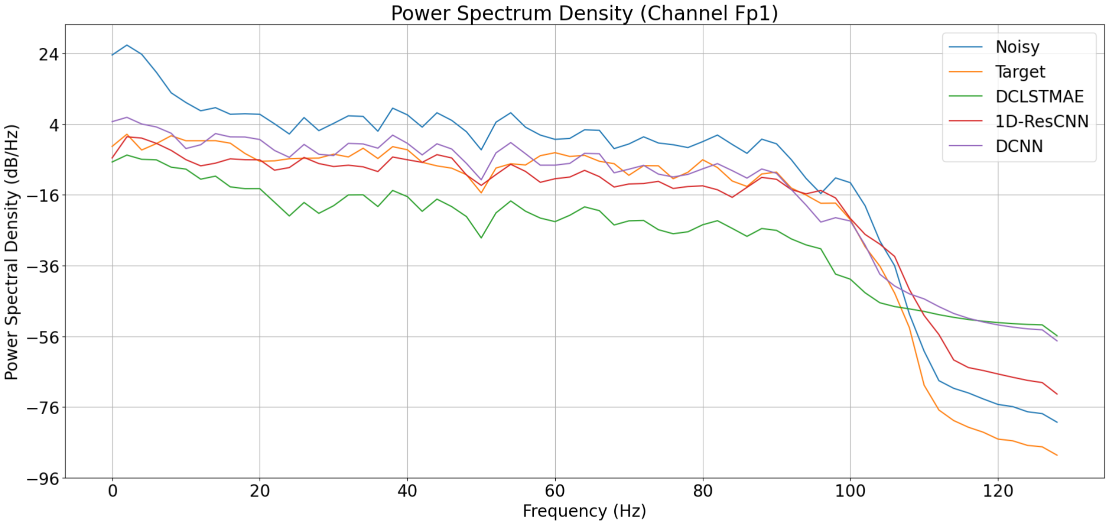
Comparing the DCAE SC model developed in the first approach to the previously trained DCNN MC model, in high amplitude short artifacts, such as eye blinks, even though the artifact is removed, the noise reduction may be excessive, there being an inability in reconstructing the smaller details of the signal. In longer types of high-amplitude artifacts, the MC architecture is able to remove a larger fraction of the noise than our SC model. Just as with the MC model, the DCAE expresses more difficulties in reconstructing the low frequency details of signals with high amplitude artifacts. Since amplitude and frequency are normally inversely proportional in EEG signals, this behaviour may be explained, because loss functions initially learn to remove the high amplitude artifacts, and only then to reconstruct the low amplitude characteristics of the target segment. However, in artifacts with lower amplitudes, such as pulse artifacts, the SC architecture performs better than the MC approach.

This difference in performances can be most likely due to the fact that EEG signals for each channel are always partially influenced by the signals of all other channels, a concept present in the very definition of Independent Component Analysis (ICA), which can only isolate as many ICs as the number of sources (in this case, channels). By receiving all channels as input instead of only one at a time, the DCNN MC model will be able of better explore this correlations and use them to enhance its signal reconstruction capabilities.

Regarding now the DCLSTMAE architecture developed in the second approach, this architecture outperformed the other SC models, which were adapted or reconstructed from literature and trained using the same 5-second signals. Putting it up

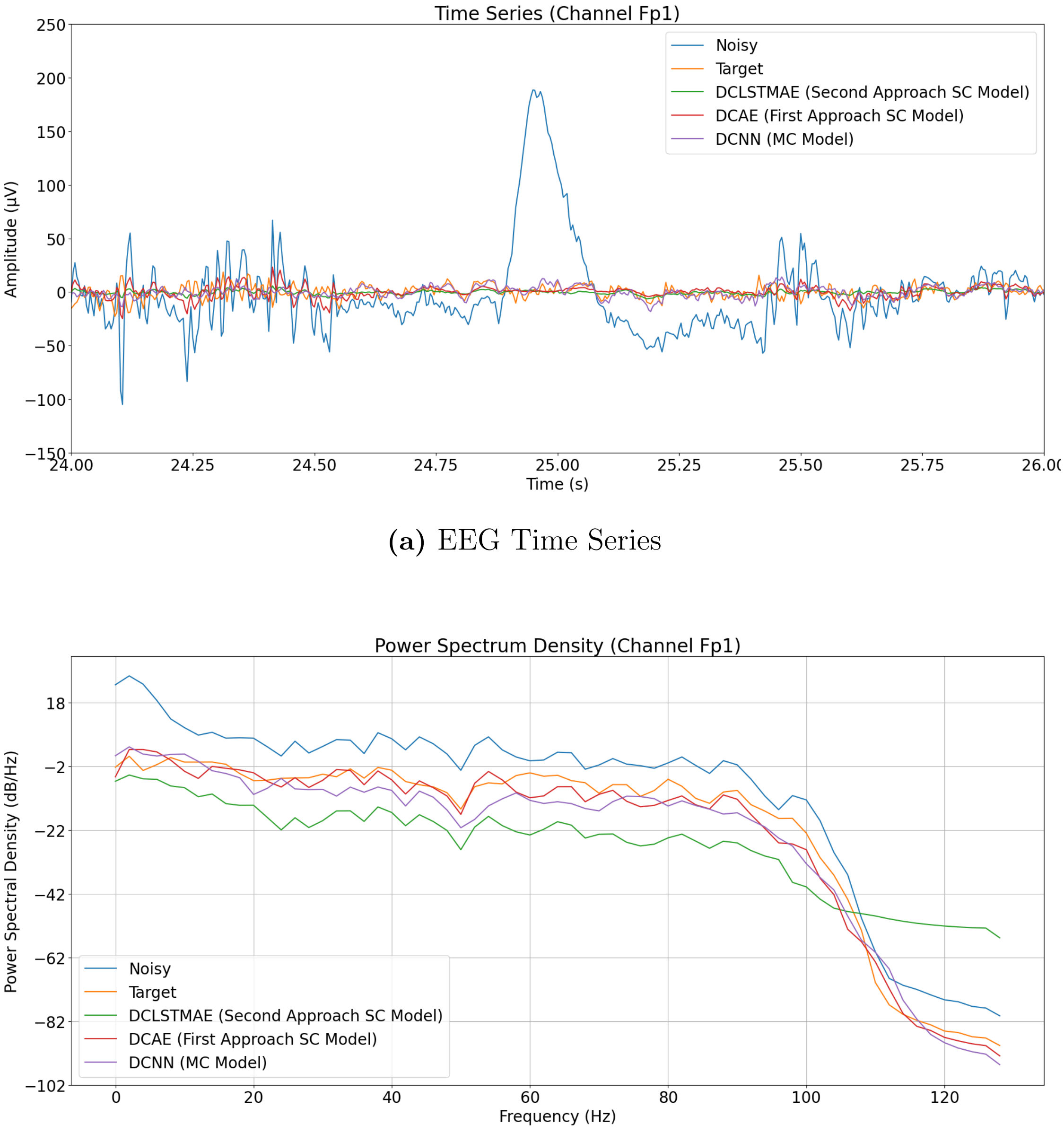


(a) EEG Time Series



(b) PSD

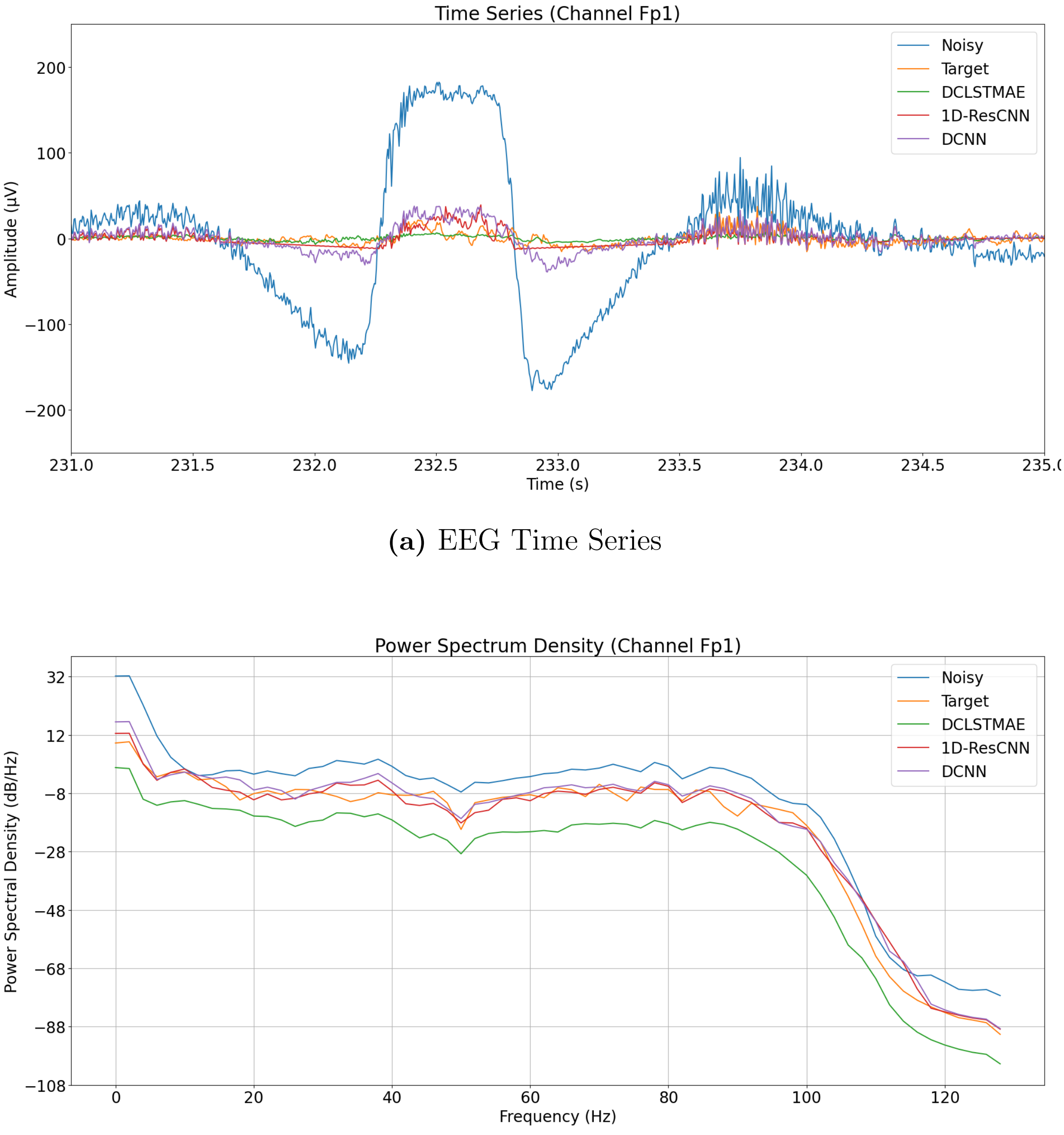
Figure 5.32: Example EEG segment from the test set containing eye blink artifact comparing the three architectures developed in the second approach.



(a) EEG Time Series

(b) PSD

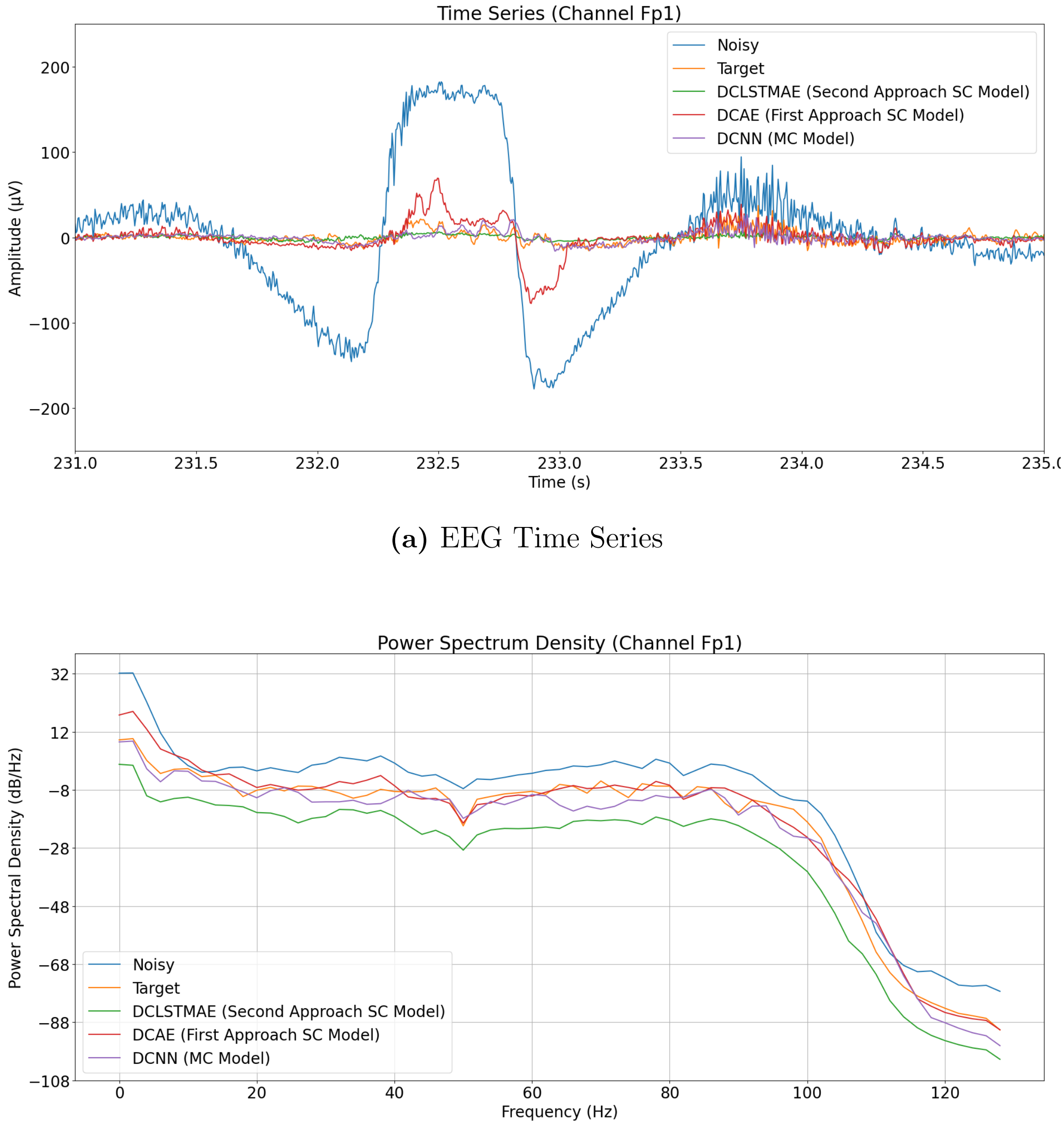
Figure 5.33: Example EEG segment from the test set containing eye blink artifact comparing the DCLSTMAE to the DCAE model from the first approach.



(a) EEG Time Series

(b) PSD

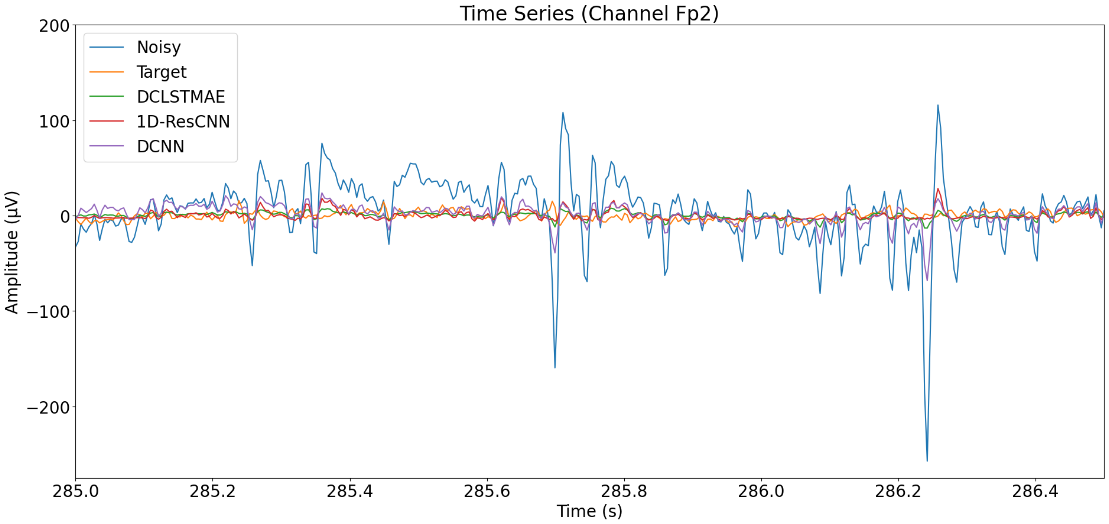
Figure 5.34: Example EEG segment from the test set containing eye movement artifact comparing the three architectures developed in the second approach.



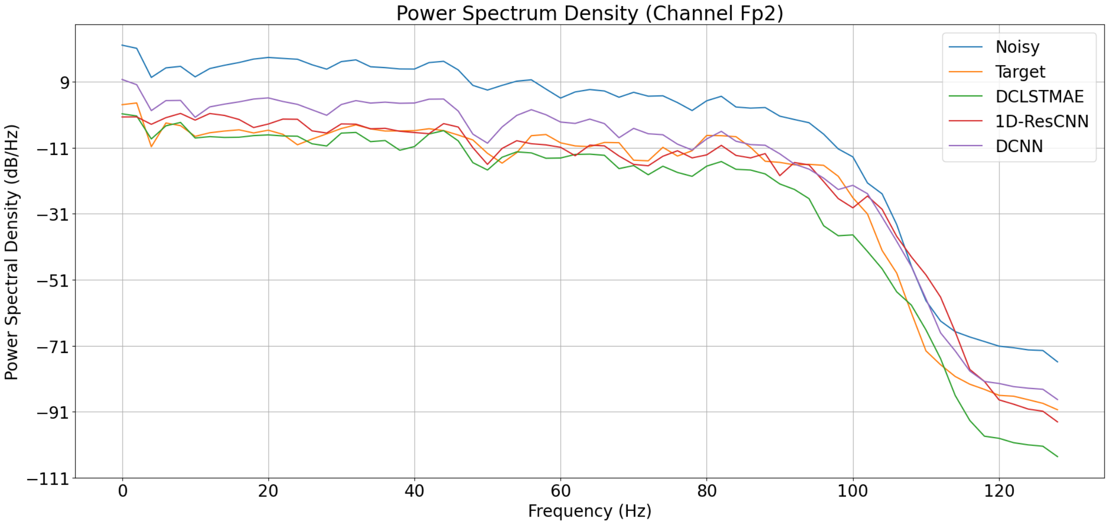
(a) EEG Time Series

(b) PSD

Figure 5.35: Example EEG segment from the test set containing eye movement artifact comparing the DCLSTMAE to the DCAE model from the first approach.

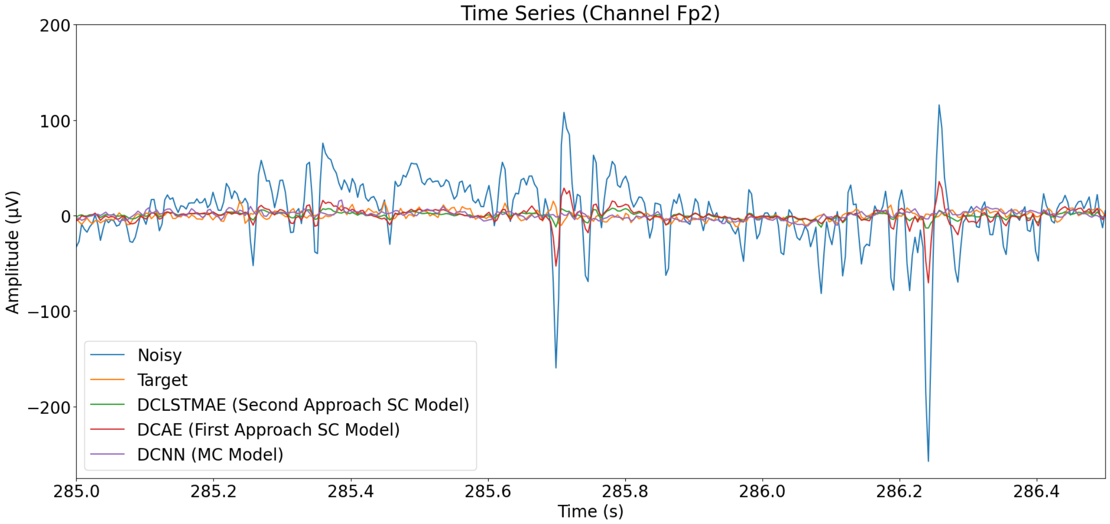


(a) EEG Time Series

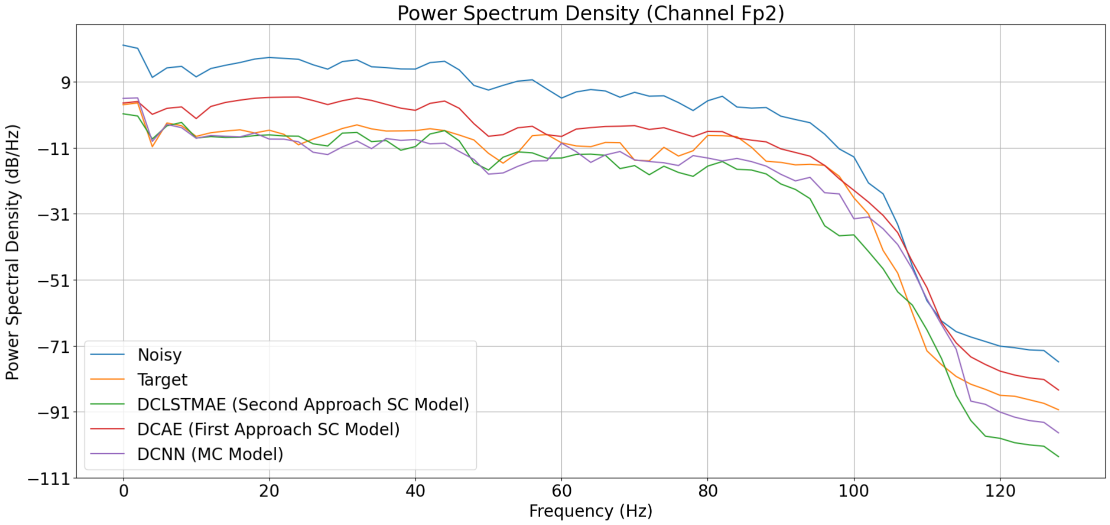


(b) PSD

Figure 5.36: Example EEG segment from the test set containing muscle activity comparing the three architectures developed in the second approach.

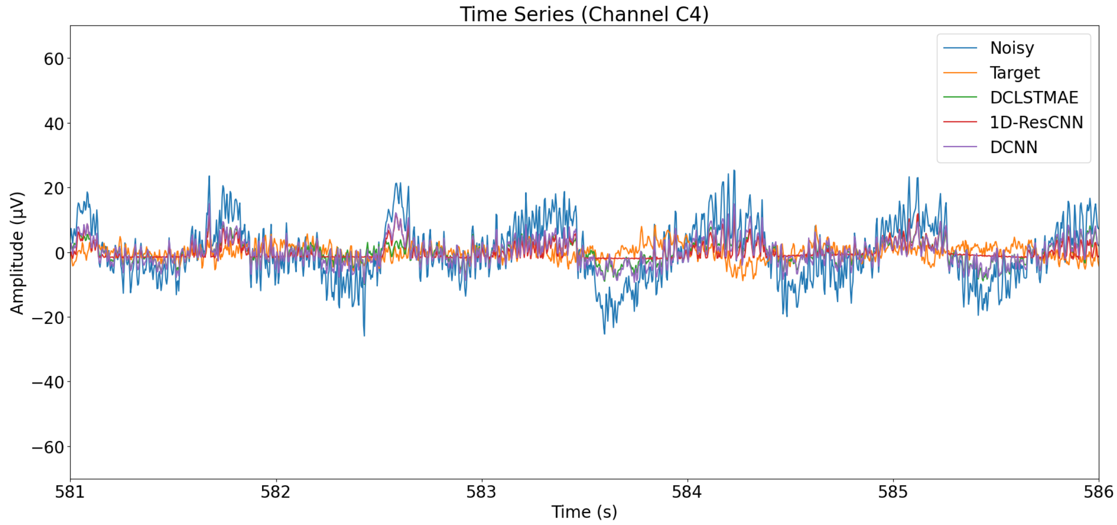


(a) EEG Time Series

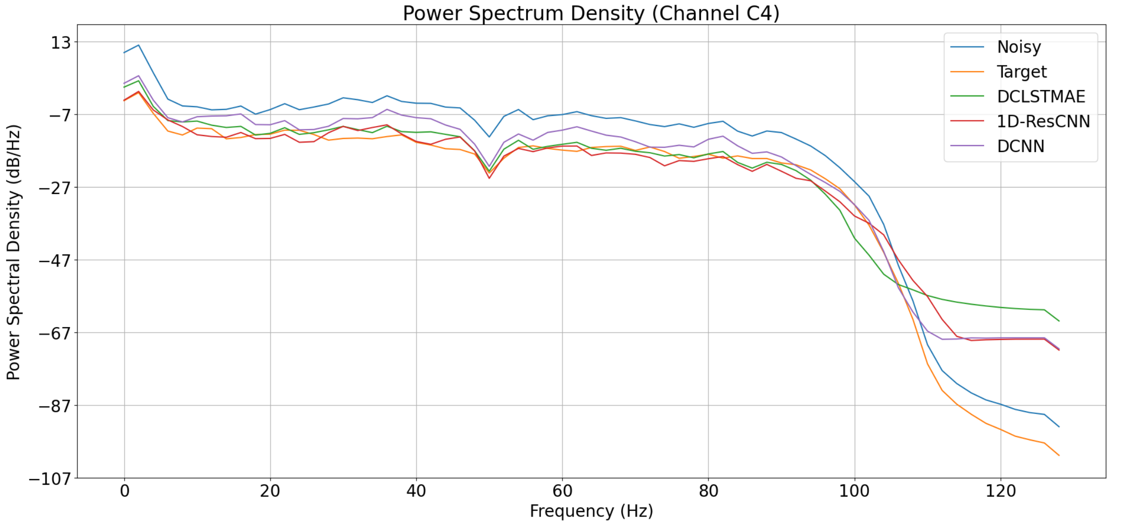


(b) PSD

Figure 5.37: Example EEG segment from the test set containing muscle activity comparing the DCLSTMAE to the DCAE model from the first approach.

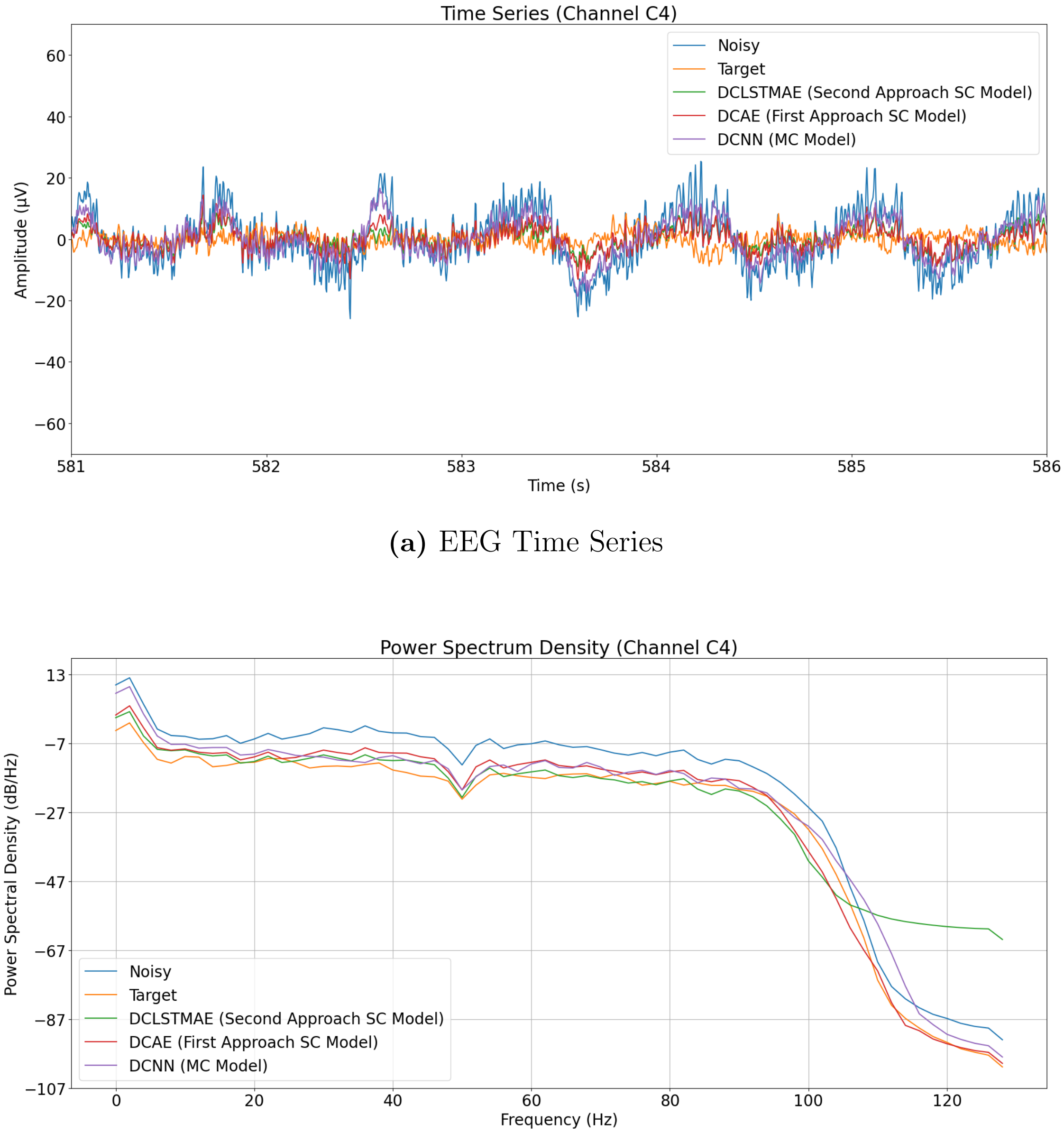


(a) EEG Time Series



(b) PSD

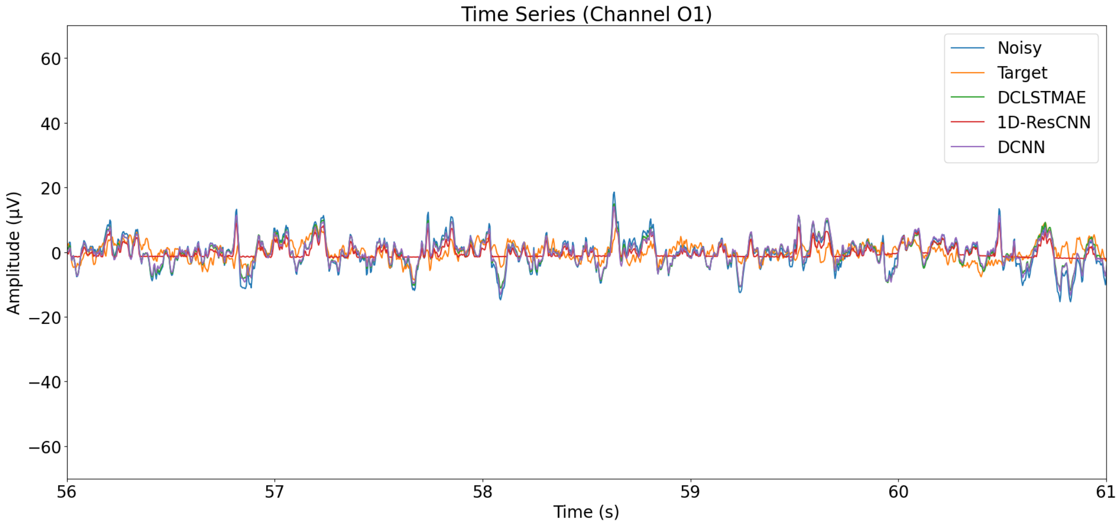
Figure 5.38: Example EEG segment from the test set containing pulse artifact comparing the three architectures developed in the second approach.



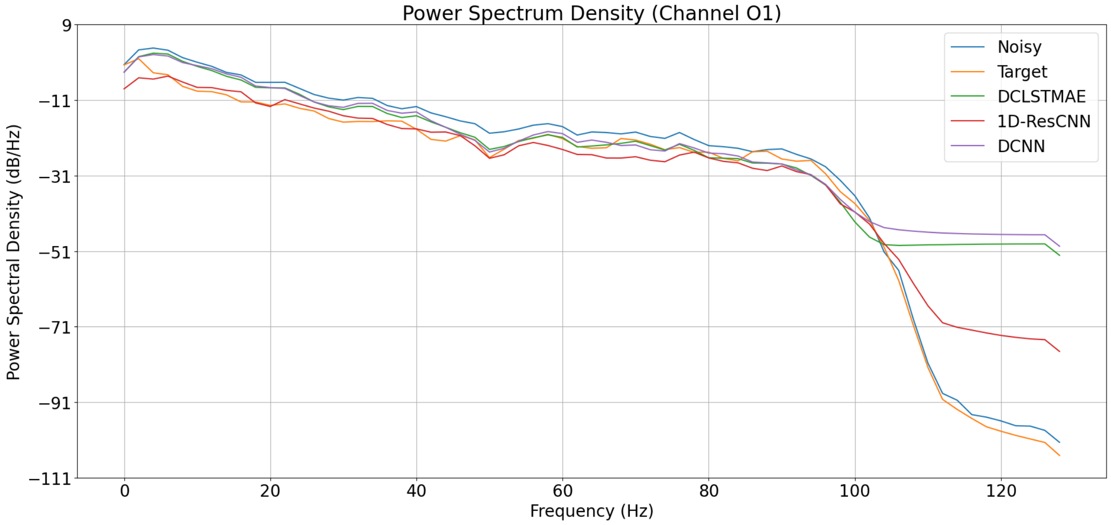
(a) EEG Time Series

(b) PSD

Figure 5.39: Example EEG segment from the test set containing pulse artifact comparing the DCLSTMAE to the DCAE model from the first approach.

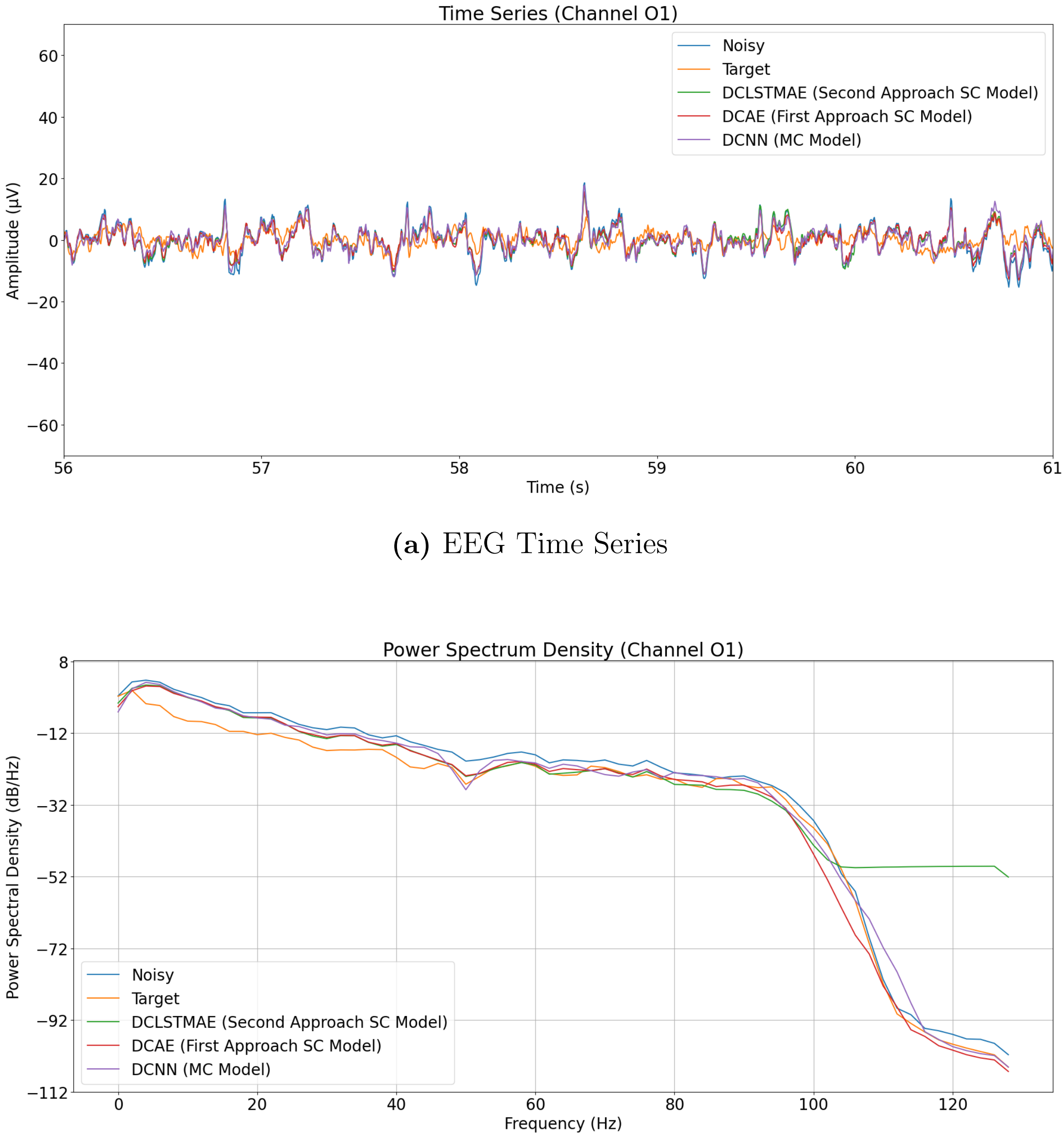


(a) EEG Time Series



(b) PSD

Figure 5.40: Example EEG segment from the test set containing cardiac artifact comparing the three architectures developed in the second approach.



(a) EEG Time Series

(b) PSD

Figure 5.41: Example EEG segment from the test set containing cardiac artifact comparing the DCLSTMAE to the DCAE model from the first approach.

against the DCAE architecture from the first approach, we can see from the example segments figures that, although both models behave similarly, the DCLSTMAE is able to reduce a larger fraction of the noise. However, difficulties in reconstructing signals' small details persist, and the MC is still significantly better.

Even though the majority of the channels report statistically significant differences between DCLSTMAE and the DCAE architectures, their behaviour is similar and differences in their performances are small. This might be explained by the counterbalance effect to the increase in the architecture's complexity caused by the segment sample size reduction from 10 minutes to 5 seconds - although the increase in number of filters and the introduction of a BiLSTM layer was proven to enhance the architecture's results, enabling it to obtain nearly identical results to the model using 10-minute segments, the lower sample size limits the information that the model receives in each input. This leads us to believe that without hardware limitations, if it was possible to apply the DCLSTMAE architecture to the 10-minute segments, performance could improve past the results of the DCAE model.

6

Conclusion

Electroencephalogram (EEG) recordings are affected by several different types of artifacts, which must be removed so that a rigorous analysis of the brain activity can be performed. However, EEG signal processing techniques are either only able to partially remove noisy artifacts or require human inspection, making it time-consuming and inadequate for real-time use. Furthermore, Deep Learning (DL) multi-channel (MC) models are restricted to recordings with the same number of channels with which they were trained. As such, the research at hand focused on the advancement of the state of the art of EEG signal pre-processing, through the development and training of automatic single-channel (SC) models based on Deep Learning Neural Networks (DLNNs) capable of removing artifacts and reconstructing EEG signals.

The models were trained and tested using real EEG data from patients with epilepsy in pre-surgical monitoring. Initially, we developed a Deep Convolutional Autoencoder (DCAE) using 10-minute segments and compared it to a previously trained MC Deep Convolutional Neural Network (DCNN). Results indicate that even though our model was able to largely reduce the noise and remove several artifacts, it was outperformed by the MC DCNN, presumably because of the influence that one channel has over the other channels, a connection that can be explored by a MC model but not by a SC model.

In the second approach, segment size was reduced from 10 minutes to 5 seconds, and the complexity of the DLNN was increased, with a higher number of filters and the incorporation of a Bidirectional LSTM (BiLSTM) layer. Along with this model, two other SC architectures were either retrieved or adapted from literature and trained with the same data, for performance comparison purposes. Our model obtained better results than the other SC models, and similar values in the evaluation metrics than the DCAE. Upon visual inspection, this new model could reduce larger portions of some of the artifacts compared to the architecture from the first approach, even though difficulties in detail reconstruction persisted in both models.

Although these results reinforce the notion that DLNNs are a powerful tool in artifact removal and EEG signal cleaning, several limitations, namely in the development of SC architectures, should be addressed in future works. To resolve problems in the reconstruction of the small details in the signals, there is a need to try different training setups, such as trying different loss functions to find one better suited for this purpose. In order for the SC models to achieve comparable performances to those of MC ones, complexity of the networks, training times and dataset size must be further increased so that it can compensate the input of a unique channel. This might prove challenging due to hardware and time constraints, since the increase in complexity leads to the training of the models being more time consuming and utilizing more of the device's memory.

Furthermore, the reduction in sample size also creates an imbalance, specifically by originating a lot of 5-second segments with practically no noise, which was demonstrated by the difference in metrics when they were calculated for 5-second and 10-minute segments, and might not prove to be the most useful for model training. In the future, this should be taken into account so that there are a comparable number of clean and noisy segments. The fact that dataset preparation was performed using Independent Component Analysis (ICA), which is also a MC pre-processing technique, can also indicate that the data was prepared for the development of MC and not SC models.

This work paves the way for future research, that can either improve upon the architectures developed here or test different kinds of DLNNs that can achieve an even better reconstruction of EEG signals, and that are not limited to any kind of acquisition system, unlike MC models.

Bibliography

[1] M. Teplan *et al.*, "Fundamentals of eeg measurement," *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.

[2] M. M. Siddiqui, S. Rahman, S. H. Saeed, and A. Banodia, "Eeg signals play major role to diagnose sleep disorder," *International Journal of Electronics and Computer Science Engineering (IJECSE)*, vol. 2, no. 2, pp. 503–505, 2013.

[3] S. Chokroverty *et al.*, "Overview of sleep & sleep disorders," *Indian J Med Res*, vol. 131, no. 2, pp. 126–140, 2010.

[4] S. J. M. Smith, "Eeg in the diagnosis, classification, and management of patients with epilepsy," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, no. suppl 2, pp. ii2–ii7, 2005.

[5] X. Jiang, G. Bian, and Z. Tian, "Removal of artifacts from eeg signals: A review," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[6] S. Sanei and J. Chambers, *Introduction to EEG*, ch. 1, pp. 1–34. John Wiley & Sons, Ltd, 2007.

[7] B. Yang, K. Duan, C. Fan, C. Hu, and J. Wang, "Automatic ocular artifacts removal in eeg using deep learning," *Biomedical Signal Processing and Control*, vol. 43, pp. 148–158, 2018.

[8] N. M. N. Leite, E. T. Pereira, E. C. Gurjao, and L. R. Veloso, "Deep convolutional autoencoder for EEG noise filtering," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, Dec. 2018.

[9] R. Ghosh, N. Sinha, and S. K. Biswas, "Automated eye blink artefact removal from EEG using support vector machine and autoencoder," *IET signal process.*, vol. 13, pp. 141–148, Apr. 2019.

[10] W. Sun, Y. Su, X. Wu, and X. Wu, "A novel end-to-end 1d-rescnn model to remove artifact from eeg signals," *Neurocomputing*, vol. 404, pp. 108–121, 2020.

[11] F. Lopes, A. Leal, J. Medeiros, M. F. Pinto, A. Dourado, M. Dümpelmann, and C. Teixeira, "Automatic electroencephalogram artifact removal using deep convolutional neural networks," *IEEE Access*, vol. 9, pp. 149955–149970, 2021.

[12] M. Jurczak, M. Kołodziej, and A. Majkowski, "Implementation of a convolutional neural network for eye blink artifacts removal from the electroencephalography signal," *Frontiers in Neuroscience*, vol. 16, 2022.

[13] M. Mathe, M. Padmaja, and B. Tirumala Krishna, "Intelligent approach for artifacts removal from eeg signal using heuristic-based convolutional neural network," *Biomedical Signal Processing and Control*, vol. 70, p. 102935, 2021.

[14] Z. Zhang, X. Yu, X. Rong, and M. Iwata, "A novel multimodule neural network for eeg denoising," *IEEE Access*, vol. 10, pp. 49528–49541, 2022.

[15] W. Zhang, W. Yang, X. Jiang, X. Qin, J. Yang, and J. Du, "Two-stage intelligent multi-type artifact removal for single-channel eeg settings: A gru autoencoder based approach," *IEEE Transactions on Biomedical Engineering*, vol. PP, pp. 1–1, 03 2022.

[16] S. Siuly, Y. Li, and Y. Zhang, *EEG Signal Analysis and Classification*, ch. 1, pp. 3–21. Springer, Cham, 2016.

[17] A. K. Shah and S. Mittal, "Invasive electroencephalography monitoring: Indications and presurgical planning," *Ann. Indian Acad. Neurol.*, vol. 17, pp. S89–94, Mar. 2014.

[18] V. Jurcak, D. Tsuzuki, and I. Dan, "10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems," *NeuroImage*, vol. 34, pp. 1600–1611, 2007.

[19] R. Oostenveld and P. Praamstra, "The five percent electrode system for high-resolution eeg and erp measurements," *Clinical Neurophysiology*, vol. 112, no. 4, pp. 713–719, 2001.

[20] Y. Wang, J. Yan, J. Wen, T. Yu, and X. Li, "An intracranial electroencephalography (ieeg) brain function mapping tool with an application to epilepsy surgery evaluation," *Frontiers in Neuroinformatics*, vol. 10, p. 15, 2016.

[21] "Invasive electroencephalography (eeg) monitoring before epilepsy surgery," 2017. Last accessed May 2021.

[22] J. Parvizi and S. Kastner, "Promises and limitations of human intracranial electroencephalography," *Nature Neuroscience*, vol. 21, pp. 474–483, 2018.

[23] J. N. Acharya and V. J. Acharya, "Overview of eeg montages and principles of localization," *Journal of Clinical Neurophysiology*, vol. 36, no. 5, 2019.

[24] J. N. Acharya, A. J. Hani, P. D. Thirumala, and T. N. Tsuchida, "American clinical neurophysiology society guideline 3: A proposal for standard montages to be used in clinical EEG," *J. Clin. Neurophysiol.*, vol. 33, pp. 312–316, Aug. 2016.

[25] L. Leuchs, "Choosing your reference – and why it matters," 2019. Last accessed December 2021.

[26] Y. Zhai and D. Yao, "A study on the reference electrode standardization technique for a realistic head model," *Computer Methods and Programs in Biomedicine*, vol. 76, no. 3, pp. 229–238, 2004.

[27] D. Yao, "A method to standardize a reference of scalp EEG recordings to a point at infinity," *Physiological Measurement*, vol. 22, pp. 693–711, oct 2001.

[28] H. J. Eun, "Basics of electroencephalography for neuropsychiatrist," *J. Korean Neuropsychiatr. Assoc.*, vol. 58, no. 2, p. 76, 2019.

[29] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, "Chapter 2 - technological basics of eeg recording and operation of apparatus," in *Introduction to EEG- and Speech-Based Emotion Recognition* (P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, eds.), pp. 19–50, Academic Press, 2016.

[30] M. K. Islam, A. Rastegarnia, and Z. Yang, "Methods for artifact detection and removal from scalp EEG: A review," *Neurophysiol. Clin.*, vol. 46, pp. 287–305, Nov. 2016.

[31] M. Hartmann, K. Schindler, T. Gebbink, G. Gritsch, and T. Kluge, "Pureeeg: Automatic eeg artifact removal for epilepsy monitoring," *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 44, no. 5, pp. 479–490, 2014.

[32] K. T. Sweeney, T. E. Ward, and S. F. McLoone, "Artifact removal in physiological signals—practices and possibilities," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, pp. 488–500, 2012.

[33] I. Kaya, *A Brief Summary of EEG Artifact Handling.* 07 2021.

[34] X. Chen, A. Liu, H. Peng, and R. K. Ward, "A preliminary study of muscular artifact cancellation in single-channel eeg," *Sensors*, vol. 14, no. 10, pp. 18370–18389, 2014.

[35] J. A. Urigüen and B. Garcia-Zapirain, “EEG artifact removal-state-of-the-art and guidelines,” *J. Neural Eng.*, vol. 12, p. 031001, June 2015.

[36] G. Garcia-Molina, “Direct brain-computer communication through scalp recorded eeg signals,” 01 2004.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ch. 1, pp. 1–26. MIT Press, 2016.

[38] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2014.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ch. 11, pp. 409–430. MIT Press, 2016.

[40] G. Roffo, “Ranking to learn and learning to rank: On the role of ranking in pattern recognition applications,” 6 2017.

[41] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[42] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014.

[43] M. Phi, “Illustrated guide to lstm’s and gru’s: A step by step explanation,” 2018. Last accessed November 2021.

[44] “Applied deep learning - part 3: Autoencoders,” 2020. Last accessed May 2021.

[45] J. Rocca, “Understanding variational autoencoders (vaes),” 2019. Last accessed June 2022.

[46] G. L. Wallstrom, R. E. Kass, A. Miller, J. F. Cohn, and N. A. Fox, “Automatic correction of ocular artifacts in the EEG: a comparison of regression-based and component-based methods,” *Int. J. Psychophysiol.*, vol. 53, pp. 105–119, July 2004.

[47] M. M. van den Berg-Lenssen, J. A. van Gisbergen, and B. W. Jervis, “Comparison of two methods for correcting ocular artefacts in EEGs,” *Med. Biol. Eng. Comput.*, vol. 32, pp. 501–511, Sept. 1994.

[48] M. M. N. Mannan, M. A. Kamran, and M. Y. Jeong, “Identification and removal of physiological artifacts from electroencephalogram signals: A review,” *IEEE Access*, vol. 6, pp. 30630–30652, 2018.

[49] M. Bandarabadi, C. A. Teixeira, J. Rasekhi, and A. Dourado, "Epileptic seizure prediction using relative spectral power features," *Clinical Neurophysiology*, vol. 126, no. 2, pp. 237–248, 2015.

[50] S. Romero, M. A. Mañanas, and M. J. Barbanoj, "Ocular reduction in EEG signals based on adaptive filtering, regression and blind source separation," *Ann. Biomed. Eng.*, vol. 37, pp. 176–191, Jan. 2009.

[51] M. Golabbakhsh, M. Masoumzadeh, and M. Sabahi, "Ecg and power line noise removal from respiratory emg signal using adaptive filters," *Majlesi Journal of Electrical Engineering*, vol. 5, 12 2011.

[52] J. DEMPSTER, "Chapter six - signal analysis and measurement," in *The Laboratory Computer* (J. DEMPSTER, ed.), Biological Techniques Series, pp. 136–171, London: Academic Press, 2001.

[53] M. Izzetoglu, A. Devaraj, S. Bunce, and B. Onaral, "Motion artifact cancellation in NIR spectroscopy using wiener filtering," *IEEE Trans. Biomed. Eng.*, vol. 52, pp. 934–938, May 2005.

[54] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *Pervasive Computing, IEEE*, vol. 2, pp. 24– 33, 08 2003.

[55] G. Welch and G. Bishop, "An introduction to the kalman filter," *Univ. North Carolina at Chapel Hill, Chapel Hill,*, 2006.

[56] D. Crisan, "Particle filters - a theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, pp. 17–41, New York, NY: Springer New York, 2001.

[57] B. Yang, T. Zhang, Y. Zhang, W. Liu, J. Wang, and K. Duan, "Removal of electrooculogram artifacts from electroencephalogram using canonical correlation analysis with ensemble empirical mode decomposition," *Cognit. Comput.*, vol. 9, pp. 626–633, Oct. 2017.

[58] M. K. I. Molla, M. Rabiul Islam, T. Tanaka, and T. M. Rutkowski, "Artifact suppression from eeg signals using data adaptive time domain filtering," *Neurocomput.*, vol. 97, p. 297–308, nov 2012.

[59] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 01, pp. 1–41, Jan. 2009.

[60] S. Gaci, "A new ensemble empirical mode decomposition (eemd) denoising method for seismic signals," *Energy Procedia*, vol. 97, pp. 84–91, 2016. European Geosciences Union General Assembly 2016, EGU Division Energy, Resources the Environment (ERE).

[61] S. Talebi, "The wavelet transform," 2020. Last accessed December 2021.

[62] D. Safieddine, A. Kachenoura, L. Albera, G. Birot, A. Karfoul, A. Pasnicu, A. Biraben, F. Wendling, L. Senhadji, and I. Merlet, "Removal of muscle artifact from EEG data: comparison between stochastic (ICA and CCA) and deterministic (EMD and wavelet-based) approaches," *EURASIP J. Adv. Signal Process.*, vol. 2012, Dec. 2012.

[63] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 84, pp. 626–638, Apr. 1996.

[64] L. Shoker, S. Sanei, and M. A. Latif, "Removal of eye blinking artifacts from EEG incorporating a new constrained BSS algorithm," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2004, pp. 909–912, 2004.

[65] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review," *Neural Inf. Process. Lett. Rev*, vol. 6, 11 2004.

[66] R. Vigario and E. Oja, "BSS and ICA in neuroinformatics: from current practices to open challenges," *IEEE Rev. Biomed. Eng.*, vol. 1, pp. 50–61, 2008.

[67] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Netw.*, vol. 13, pp. 411–430, May 2000.

[68] T. W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources," *Neural Comput.*, vol. 11, pp. 417–441, Feb. 1999.

[69] J. Palmer, K. Kreutz-Delgado, and S. Makeig, "Amica: An adaptive mixture of independent component analyzers with shared components," 01 2011.

[70] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, 1999.

[71] A. Belouchrani, K. abed meraim, J.-F. Cardoso, and E. Moulines, "Second order blind separation of temporally correlated sources.," *Proceeding of the International Conference on Digital Signal Processing*, 05 1993.

[72] W. De Clercq, A. Vergult, B. Vanrumste, W. Van Paesschen, and S. Van Huffel, “Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram,” *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 2583–2587, Dec. 2006.

[73] X. Yong, R. Ward, and G. Birch, “Artifact removal in eeg using morphological component analysis,” pp. 345–348, 04 2009.

[74] L. Shoker, S. Sanei, and J. Chambers, “Artifact removal from electroencephalograms using a hybrid bss-svm algorithm,” *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 721–724, 2005.

[75] A. Mognon, J. Jovicich, L. Bruzzone, and M. Buiatti, “ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features,” *Psychophysiology*, vol. 48, pp. 229–240, Feb. 2011.

[76] H. Nolan, R. Whelan, and R. Reilly, “Faster: Fully automated statistical thresholding for eeg artifact rejection,” *Journal of neuroscience methods*, vol. 192, pp. 152–62, 09 2010.

[77] M. Chaumon, D. Bishop, and N. Busch, “A practical guide to the selection of independent components of the electroencephalogram for artifact correction,” *Journal of neuroscience methods*, vol. 250, 03 2015.

[78] I. Winkler, S. Haufe, and M. Tangermann, “Automatic classification of artifactual ICA-components for artifact removal in EEG signals,” *Behav. Brain Funct.*, vol. 7, p. 30, Aug. 2011.

[79] T. Radüntz, J. Scouten, O. Hochmuth, and B. Meffert, “Automated eeg artifact elimination by applying machine learning algorithms to ica-based features,” *Journal of Neural Engineering*, vol. 14, p. 046004, 08 2017.

[80] G. Tamburro, P. Fiedler, D. Stone, J. Haueisen, and S. Comani, “A new ica-based fingerprint method for the automatic removal of physiological artifacts from eeg recordings,” *PeerJ*, vol. 6, 02 2018.

[81] L. Pion-Tonachini, K. Kreutz-Delgado, and S. Makeig, “Iclabel: An automated electroencephalographic independent component classifier, dataset, and website,” *NeuroImage*, vol. 198, pp. 181–197, 2019.

[82] M. A. Klados and P. D. Bamidis, “A semi-simulated EEG/EOG dataset for the comparison of EOG artifact rejection techniques,” *Data Brief*, vol. 8, pp. 1004–1006, Sept. 2016.

[83] M. Terzano, L. Parrino, and A. Smerieri, “Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (cap) in human sleep,” *Sleep Med*, vol. 3, pp. 187–199, 01 2001.

[84] A. Goldberger, L. Amaral, L. Glass, S. Havlin, J. Hausdorg, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, H. Stanley, and P. Physiobank, “Components of a new research resource for complex physiologic signals,” *PhysioNet*, vol. 101, 01 2000.

[85] G. Moody and R. Mark, “The impact of the mit-bih arrhythmia database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[86] H. Shahabi, S. Moghimi, and H. Zamiri-Jafarian, “Eeg eye blink artifact removal by eog modeling and kalman filter,” in *2012 5th International Conference on BioMedical Engineering and Informatics*, pp. 496–500, 2012.

[87] K. Sweeney, H. Ayaz, T. Ward, M. Izzetoglu, S. Mcloone, and B. Onaral, “A methodology for validating artifact removal techniques for physiological signals,” *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 16, pp. 918–26, 07 2012.

[88] M. Ihle, H. Feldwisch-Drentrup, C. A. Teixeira, A. Witon, B. Schelter, J. Timmer, and A. Schulze-Bonhage, “Epilepsiae – a european epilepsy database,” *Computer Methods and Programs in Biomedicine*, vol. 106, no. 3, pp. 127–138, 2012.

[89] F. Lopes, A. Leal, J. Medeiros, M. F. Pinto, A. Dourado, M. Dümpelmann, and C. Teixeira, “Ensemble deep neural network for automatic classification of eeg independent components,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 559–568, 2022.

[90] F. Perrin, J. Pernier, O. Bertrand, and J. Echallier, “Spherical splines for scalp potential and current density mapping,” *Electroencephalography and Clinical Neurophysiology*, vol. 72, no. 2, pp. 184–187, 1989.

[91] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[92] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” 2015.

[93] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[94] K. Kawaguchi, J. Huang, and L. P. Kaelbling, "Effect of Depth and Width on Local Minima in Deep Learning," *Neural Computation*, vol. 31, pp. 1462–1498, 07 2019.

[95] Z. Hanusz, J. Tarasinska, and W. Zieliński, "Shapiro–wilk test with known mean," vol. 14, pp. 89–100, 02 2016.

[96] R. H. C. Lopes, *Kolmogorov-Smirnov Test*, pp. 718–720. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[97] N. Nachar, "The mann-whitney u: A test for assessing whether two independent samples come from the same distribution," *Tutorials in Quantitative Methods for Psychology*, vol. 4, 03 2008.

[98] E. Ostertagova, O. Ostertag, and J. Kováč, "Methodology and application of the kruskal-wallis test," *Applied Mechanics and Materials*, vol. 611, pp. 115–120, 08 2014.

[99] A. Dinno, "Nonparametric pairwise multiple comparisons in independent groups using dunn's test," *The Stata Journal*, vol. 15, no. 1, pp. 292–300, 2015.