Development of Machine Learning-based Open-Source Tools for Processing and Analysis of Extracellular Neuronal Signals to Facilitate Disease Monitoring

Marcos Ignacio Fabietti

Department of Computer Science

A thesis submitted in partial fulfilment of the requirements of Nottingham Trent University for the degree of

Doctor of Philosophy

July 2022

“A journey of a thousand miles begins with a single step” - Lao Tzu. This thesis is dedicated to all that have supported me from close and afar, as the joy of this achievement would not exist without them to share it with.

Acknowledgements

The work described in this thesis was carried out at Nottingham Trent University between October 2019 and July 2022, while I was working as a full-time doctoral research student.

First of all I am thankful to my director of studies, Dr. Mahmud. He has been a pillar of support through out it, making sure I lived up to my potential and helping me sharpen my skills to the required level. His continuous motivation, work ethic, good will and sense of humour is something I will whole-heartedly cherish forever. Also, I would like to thank the co-supervisor Prof. Lotfi, who has been the compass to rely on when I was feeling amiss, always making sure I kept on track. His advice has always been constructive and has allowed me to improve as a researcher. I am utmost thankful to both my supervisors for their mentorship in the journey that has been this PhD.

To my lovely wife Elysia, for her unconditional love and motivation. You have been the best partner I could have wished for, and are the reason I was able to get through the difficult times. To my family, Silvia, Luis, Constanza and Plinio, for their continuous support. It is not easy to part ways for so long, but they never allowed me to feel their absence. Thank you for letting me go when it was my time to leave, and having your welcoming arms open at my return. To Lindsey and Joe, for the family I found in them. Your hospitable welcome into your lives has made my life enormously richer, and I am thankful to be able to be a part of it. To old and new friends, from close and afar, who have kept rooting for me all this time.

Finally, thank you to my colleagues at the Computational Intelligence and Applications research group for the time shared

during the past few years. To all the collaborators I've had the pleasure to work with: Shamim Kaiser, Michaela Chiappalone, Alberto Averna, Roberto Fontana, Alessandro Leparulo, Cristina Fassolato and everyone who in one way or the other contributed to the success of my research. This work was supported by Nottingham Trent University's PhD Scholarship, for the duration 2019 - 2022.

Marcos Ignacio Fabietti
July 2022

Abstract

Neuronal signals are recordings of the electrical activity of the brain, which allow gaining insight into a diverse range of information. Like other physiological signals, extensive processing and analysis must be carried out in order to extract useful information. In this context, the neuroscience community has developed different open-access tools and pipelines for the different steps involved to facilitate the studies and make more advancements in the field. The aim of the research reported in this thesis is the development of tools and pipelines to facilitate the use of machine learning techniques in chronically recorded invasive signals for early disease detection. This includes the selection of the state-of-the-art for artefact detection and removal, the processing of the signal to feed the models, and lastly a robust machine learning based classifier. The main contributions of this thesis to the application of machine learning in neuronal signal processing include an open-access tool for benchmarking the performance of artefact detection and removal with ML with over 120 articles, the creation of a toolbox with novel methods to detect and remove artefacts from extracellular neuronal signals recorded in the form of local field potentials, a novel channel-independent artefact removal method based on the forecasting of normal activity to replace affected segments, an innovative ML pipeline to detect and classify brain states from the processed local field potentials, and lastly finding novel biomarkers from these models and properly assess them against the existing literature.

Publications

As a result of the research presented in this thesis, the following publications have been published:

Refereed Journal Papers:

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alessandro Leparulo, Roberto Fontana, Stefano Vassanelli, and Cristina Fasolato, "Early detection of alzheimer's disease from cortical and hippocampal local field potentials using an ensembled machine learning model", [submitted to **IEEE Transactions on Neural Systems and Rehabilitation Engineering**], 2023.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, and M. Shamim Kaiser, "Abot: An open-source online benchmarking tool for machine learning-based artefact detection and removal methods from neuronal signals", [Accepted for publication by **Brain Informatics**, Springer], 2022.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Channel-independent recreation of artefactual signals in chronically recorded local field potentials using machine learning", **Brain Informatics**, vol. 9, no. 1, pp. 1–17, 2022.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, et al., "Santia: a matlab-based open-source toolbox for artifact detection and removal from extracellular neuronal signals", **Brain Informatics**, vol. 8, no. 1, pp. 1–19, 2021.

Refereed Conference Papers:

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alessandro Leparulo, Roberto Fontana, Stefano Vassanelli, and Cristina Fasolato, "Detection of Healthy and Unhealthy Brain States from LFP using ML", in *International Conference on Brain Informatics*. Springer, 2022.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "A matlab-based open-source toolbox for artefact removal from extracellular neuronal signals", in *International Conference on Brain Informatics*. Springer, 2021, pp. 351–365.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Anomaly detection in invasively recorded neuronal signals using deep neural network: Effect of sampling frequency", in *International Conference on Applied Intelligence and Informatics*. Springer, 2021, pp. 79–91.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "On-chip machine learning for portable systems: Application to electroencephalography-based brain-computer interfaces", in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alberto Averna, David Guggenmos, Randolph Nudo, and Michela Chiappalone, "Signal power affects artefact detection accuracy in chronically recorded local field potentials: preliminary results", in *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2021, pp. 166–169.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alberto Averna, David Guggenmos, Randolph Nudo, and Michela Chiappalone, "Adaptation of convolutional neural networks for multi-channel artifact detection in chronically recorded local field potentials", in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 1607–1613.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alberto Averna, David Guggenmos, Randolph Nudo, and Michela Chiappalone, "Artifact detection in chronically recorded local field potentials using long-short term memory neural network", in *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2020, pp. 1–6.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Machine learning in analysing invasively recorded neuronal signals: Available open access data sources", in *Brain Informatics*, Mufti Mahmud, Stefano Vassanelli, M. Shamim Kaiser, and Ning Zhong, Eds., Cham, 2020, pp. 151–162, Springer International Publishing.

Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Effectiveness of employing multimodal signals in removing artifacts from neuronal signals: An empirical analysis", in *Proc. Brain Informatics*, 2020, pp. 183–193.

Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alberto Averna, David Gugganmos, Randolph Nudo, and Michela Chiappalone, "Neural network-based artifact detection in local field potentials recorded from chronically implanted neural probes", in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

Contents

Nomenclature

Acronyms

AD	Alzheimer's Disease
AUROC	Area Under the Receiver Operating Characteristic
ABOT	Artefact removal Benchmarking Online Tool
BCI	Brain-Computer Interface
CDT	Classification Decision Trees
CNN	Convolutional Neural Network
L2/3	Cortex layers 2 to 3
L4/6	Cortex layers 4 to 6
DL	Deep Learning
DG	Dentate Gyrus
DOI	Digital Object Identifier
DT	Double Transgenic B6.152H
DTW	Dynamic Time Warping
ECG	Electrocardiogram
ECoG	Electrocorticography
EEG	Electroencephalography
EMG	Electromyogram
EOG	Electrooculogram
GUI	Graphical User Interface
WT	Healthy C57BL/6J
ICA	Independent Component Analysis
KNN	k-Nearest Neighbours

LDA	Linear Discriminant Analysis
LFP	Local Field Potential
LSTM	Long-Short Term Memory
ML	Machine Learning
MEG	Magnetoencephalography
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
MLP	Multilayer Perceptron
MI	Mutual Information
NN	Neural Network
1D-CNN	One Dimensional Convolutional Neural Network
PD	Parkinson's Disease
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
SANTIA	SigMate Advanced: a Novel Tool for Identification of artefacts in Neuronal Signals
ST	Single Transgenic PS2.30H
SVM	Support Vector Machines

Definitions

Afterpotentials are depolarising processes that happen in the membrane after an action potential, that returns the potential to the level before the action potential.

Synchronised synaptic potentials are the continuation of an action potential from generated from a pre-synaptic neuron transmitted to a post-synaptic neuron through neurotransmitters.

Voltage-gated membrane oscillations are variations of the polarity of the membrane of a cell due to the flow of sodium and potassium ions.

Hyper-parameters are used to control the learning process of a model's parameters during training. They are defined before the training, and as such cannot be estimated through the data.

Neuronal signals are defined as chronically recorded when they are captured

during a long period of time, as opposed to acute or short term.

List of Figures

List of Tables

Chapter 1

Introduction

In the UK, approximately 1 in 6 people suffer from a neurological disease, totalling 16.5 million cases [2]. There are a limited number of trained physicians who can distinguish the intricacies between them, for example, differentiating Alzheimer's Disease (AD) from vascular dementia. Early diagnosis is key to successful treatment and recovery where possible, and different technologies are employed to extract information about the brain's health. To discover new biomarkers, animal models are used to obtain recordings of the neuronal activity of the different structures over time, known as neuronal signals.

Neuronal signals are one of the cornerstones of neuroscience in the process of understanding brain activity. They are of crucial importance in the diagnosis and treatment of brain disorders, including neurodegenerative diseases and mental health problems. There are distinct types of neuronal signals, i.e. invasive or non-invasive, depending on the recording techniques. The non-invasive type includes those obtained by EEG as well as MEG, whereas the invasive type includes ECoG, local field potentials (LFPs) and neuronal spikes (which include Multi-unit Activities, Single-unit Activities and Patch-Clamp recordings). The spatiotemporal resolutions of these techniques are shown in Figure 1.1.

Among the different techniques to capture the neuronal signals, LFPs are recorded by micro-electrodes (metal, silicon, or glass micropipettes) in deeper layers of the brain by low pass filtering of the extracellular electrical potential to under 100-300 Hz. They are used to investigate the dynamics and the function

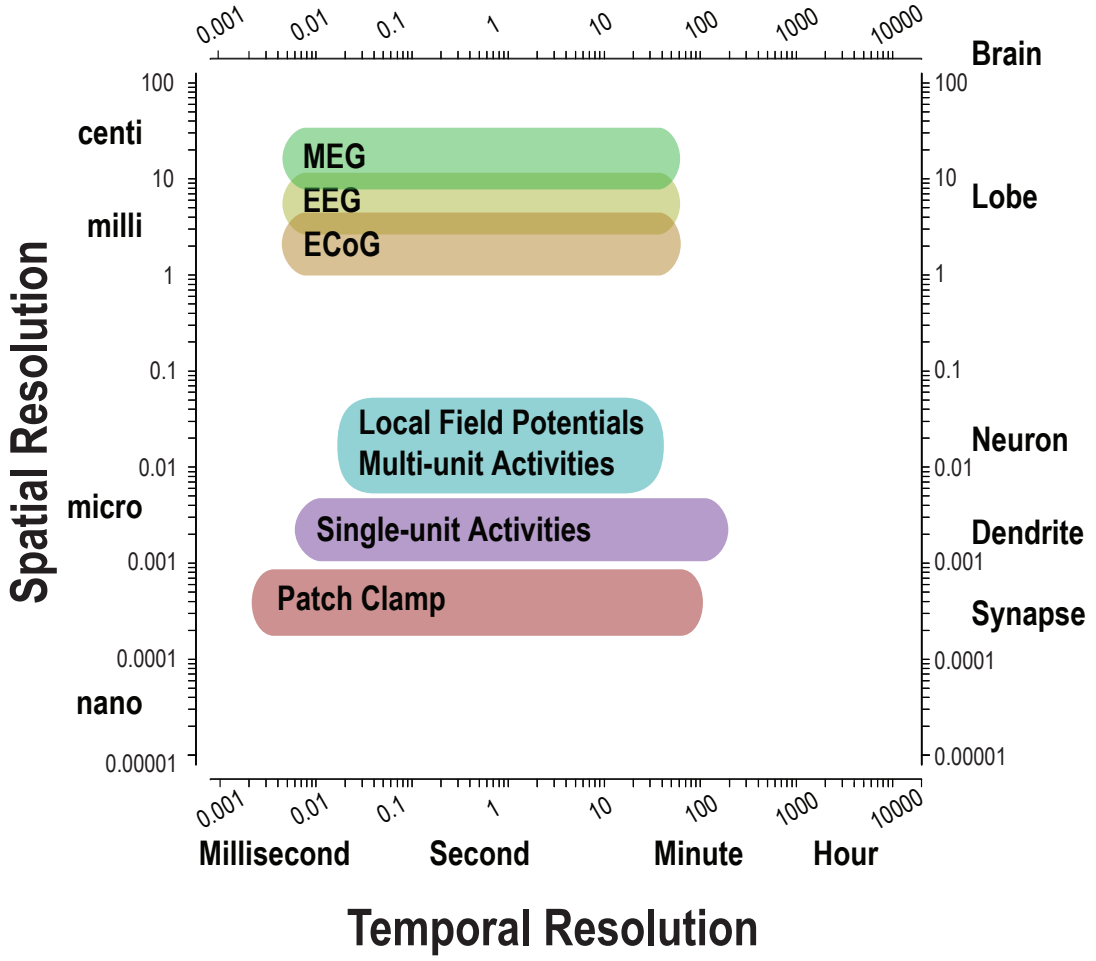

Figure 1.1: Spatio-temporal resolution of Electroencephalography (EEG), Magnetoencephalography (MEG), Electrocorticography (ECoG), Local Field Potentials, Multi-unit Activities, Single-unit Activities and Patch Clamp recordings.

of neural circuits under different conditions, given their ability to capture various activities within a wide scope of frequencies. Furthermore, they provide a stable signal for a longer period of time than multi-unit spiking activity and are therefore useful for long-term chronic experiments and clinical applications, such as brain-computer interfaces (BCI) [3]. Despite the large volumes of data the recordings generate, limited research has been conducted to exploit it via machine learning (ML), when this technology can aid in discovering new insights. However, to successfully employ ML algorithms, a computational background is required to

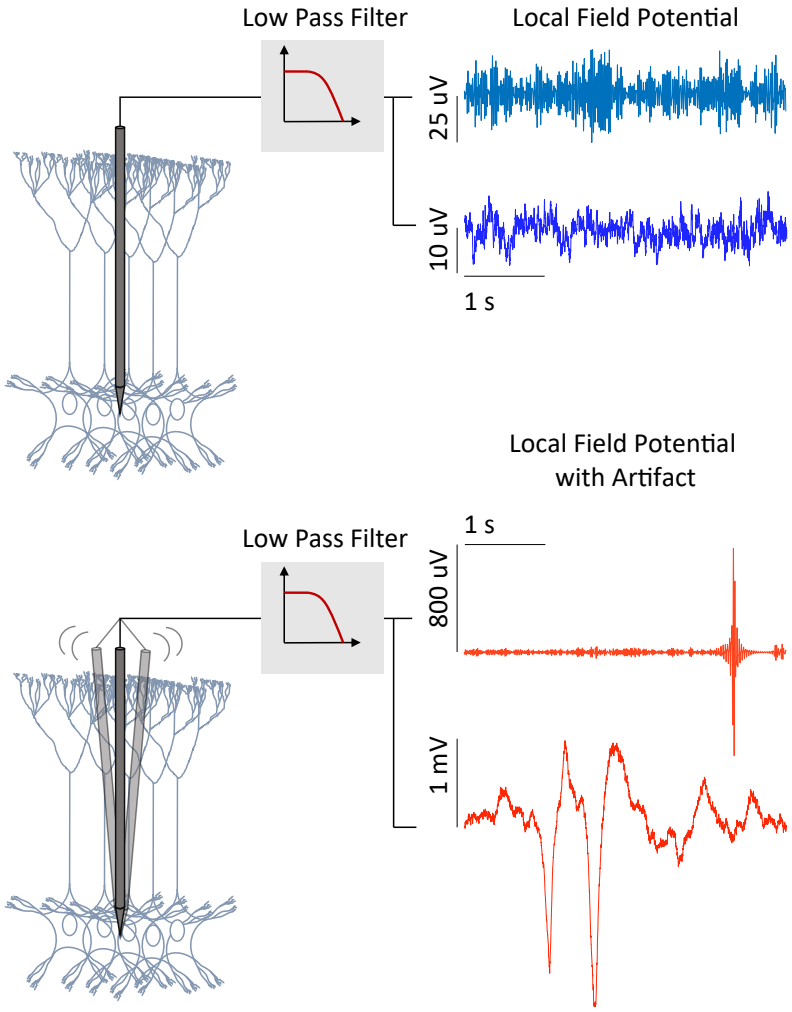

Figure 1.2: Recording of extracellular neuronal signals from behaving rodents using a linear implantable neural probe (shown in grey). Representative LFP signals with and without movement artefacts are shown from two datasets. The blue traces denote signals without artefacts, and the red traces show examples of movement artefacts present in the signals.

apply them, as there are intricacies such as defining hyper-parameters. Thus, toolboxes and pipelines that facilitate the usage of these powerful computational algorithms are beneficial to the neuroscience community.

A challenge neuronal recordings share is that they are vulnerable to electrical sources outside the structure of interest, such as the intrinsic electrical activity of the body or external one produced by the recording equipment and

other devices. Figure 1.2 shows examples of representative LFP signals with and without movement artefacts are from two datasets. Each type of artefact manifests in a specific frequency, and amplitude bands can be periodic or irregular and single or multi-channel. The main internal artefacts include Electrooculogram (EOG) generated by eye movements, Electroretinogram and blinking [4, 5], Electromyogram (EMG) produced by the contractions of muscles, Electrocardiogram (ECG) caused by the electric activity of the heart [6] or the spiking activity of local neurons in extracellular recordings [7], in other words, spike leakage. In addition, there are other artefacts such as skin potentials or respiration [8, 9, 10]. External artefacts are those generated by electronic devices or external electromagnetic waves, e.g., power lines [11], cellphone signals and light stimulation [12]. In studies where electrical stimulation is done to the brain, said signal may also appear in the recordings, which are known as stimulation artefacts [13]. Lastly, artefacts may be generated due to instrumental errors in the recording process, such as an electrode's poor contact, popping, lead movements and electrode drift, i.e., changes in the electrode's position in relation to the brain [14].

In regards to the artefact's properties, EOG changes the potentials of the electrodes in the frontal region, appearing as high amplitude 3-10 Hz signals, and its repetition produces slow waves similar to delta waves. EMG amplitudes and waveforms vary on the muscle and the degree of contraction, spanning a frequency range from 2 Hz up to 200 Hz. It can be harder to detect due to their fewer repetitions than the other artefacts. ECG has a regular pattern with frequencies near 1.2 Hz and an amplitude in the millivolt range. Respiration artefacts manifest in the 5 to 10 Hz range, overlapping with the theta band in rats. Interference artefacts such as the electricity power lines (that have a frequency of 50 or 60 Hz) can be easily removed using a notch filter, whereas cell phone signals are in the order of megahertz and can be avoided in the experimental set-up. Lastly, instrumental artefacts generated by poor electrode contact are of low frequency, whereas lead motions have a more irregular shape that bears little resemblance to neuronal activity.

Due to the spectral overlap of the artefacts with the signal, and even among themselves, filtering without producing information loss or a distortion of the

signal becomes difficult. Similarly, the detection of stress in a rodent is mostly done by a manual review of the data or supporting channels (other signals or video recordings). The detection and removal of stressed segments are crucial, as acute stress in rodents can cause the LFP to be tuned to the underlying theta oscillations, while exposure to chronic stress decreases the power of slow and fast gamma oscillations [15].

The most common practice is a manual review of the recording, where segments that have been affected by artefacts are discarded, as they have a negative impact on the posteriors analysis done to the recording. This causes a loss of information, which is costly and may distort the analysis by omitting relevant information. Because of this, computational techniques have been developed to detect and remove said artefacts, where machine learning solutions stand out due to their flexibility and computational speed. The vast range of these solutions offers a challenge to newcomers who wish to implement the best one suited for their data or those looking to compare the performance of new methods. This is worsened by the lack of consensus in the community regarding which performance metrics to use.

The rest of this chapter is divided into several sections: Section 1.1 presents the motivation behind this thesis. Section 1.2 provides an overview of the research undertaken in this thesis. Section 1.3 presents the project's aim and objectives. Section 1.4 discusses the major research challenges identified for the completion of this work. Section 1.5 presents the major contributions achieved throughout the undertaken work. Finally, Section 1.6 outlines the organisation of the remaining chapters of the thesis.

1.1 Motivation

The recording of neuronal activity with multi-channel probes generates significant volumes of data, however, little has been done to exploit them via ML that may lead to gaining new insights [16]. Research groups in the community have developed and shared toolboxes for analysing these neural recordings, however, few incorporate ML models into them (See Table 2.1 in Section 2.5). This calls for the development of tools and pipelines that aid researchers in the use of these

computational methods for processing and analysis of the signals, reducing the time spent on the artefact removal and classification steps, so that they may focus on the interpretation of results and diagnosis.

To train a ML classification model, data has to be processed for it to be standardised within each group. In neuronal signals, outliers and abnormalities are present in the form of artefacts, which require identification and removal [17]. Even though it is most often done manually, ML methods offer the computational speed and flexibility to automate this process. However, most methods have been applied to multi-channel non-invasive neuronal signals, while limited applications in their invasive counterparts are found in the literature (See Figure 4.6 in Section 4.3). Ergo, there is a need to research the development of ML methods to detect and remove artefacts in invasive neuronal recordings. As artefacts are detrimental to any posterior analysis of neuronal signals, a growing amount of research has been published. The lack of a standarised metric or benchmarking dataset is one of the known key challenges within the topic, so ways to solve this issue are sought after [18].

Improving the understanding of neuronal diseases is key for an early diagnosis and treatment, so the search for alternative and objective biomarkers is crucial. Data-driven solutions can recognise new markers and can aid the physicians' decision-making. The explainability of these ML models is key for user adoption and confidence in these systems, particularly in the healthcare field [19, 20]. While invasive recordings allow to study specific areas of the brain, they may cause harm to human subjects. Due to this, transgenic mouse lines are accepted as AD amyloidosis research models [21], and commonly employed to investigate medicines capable of altering illness progression and restoring memory functions [22]. This work focuses on developing an explainable ML pipeline that allows for the detection of neuronal disease in an animal model, with AD at an early stage as a case study.

1.2 Overview of the Research

This research aims to establish a computer science solution to aid disease detection in chronically recorded invasive signals. In order to achieve consistent

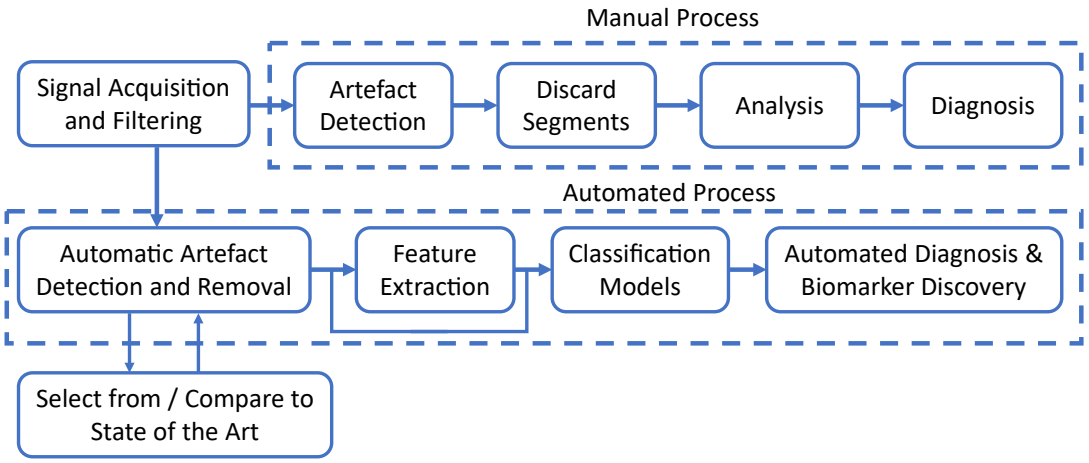

Figure 1.3: Overview of the research outcomes of the research.

results from the raw time series, there are several required steps. First, data has to be pre-processed regarding scaling, band-pass filtering, structuring, and labelling. Second, abnormalities in the recordings need to be addressed, which includes many methods for identifying the sections and removing them. An evaluation scale must be used to evaluate its performance against other methods in the literature. Following this, the processed signals can have features extracted or fed directly to a classifier. After training, validating and testing the model, it will return a label indicating the presence or absence of the disease for every example shown to it. Furthermore, insight into the model decision-making can indicate new or validate known biomarkers. The overview of the research outcomes of the research is summarised in Figure 1.3.

From the overview figure, sub-areas of investigation that were carried out can be defined:

- Comparison of artefact detection and removal methods: this includes research into the different performance metrics used by the authors, the wide range of methods and architectures, the diverse types of artefacts identified and the reported performance. The creation of a dataset with all the information and the creation of an open-access tool to make it available to the community.
- Artefact Detection and Removal: this involves the investigation of means of detecting abnormalities in a signal and its subsequent removal to have a

normalised input in the classification model. A wide range of supervised methods were explored for detection. In addition, the development of a channel-independent method based on recurrent neural networks is presented. The current state of open access toolboxes available for this task was reviewed.
- Disease Staging: this includes the development of a pipeline composed of various supervised learning techniques for detecting AD across multiple domains. A further investigation was undertaken to establish the performance of the classifiers in an ensemble manner. The robustness of the pipeline to an increasing number of missing channels and stress segments were explored as well.

1.3 Aim and Objectives

This research aims to develop tools and pipelines to facilitate the use of ML techniques in neuronal signals recorded using multi-site extracellular neuronal probes, for disease detection, with AD at an early stage as a case study. As the recorded signals are often contaminated by unwanted elements from external and internal sources, artefact-free signals are needed for accurate prediction and monitoring of these diseases.

To achieve the above research aim, the following objectives were identified:

- Develop an open-access tool for benchmarking the performance of artefact detection and removal with ML with over 120 articles.
- Creation of a toolbox with novel methods to detect and remove artefacts from extracellular neuronal signals recorded in the form of LFPs.
- Propose a novel channel-independent artefact removal method based on the forecasting of normal activity to replace affected segments.
- Propose an interpretable ML pipeline to detect and classify brain states (i.e., presence of AD) from the processed LFP signals.
- Find novel biomarkers from the interpretable models in the proposed pipeline, and properly assess them against the existing literature.

1.4 Research Challenges

In the process of completing the aforementioned research aim and objectives, four main research challenges have been identified:

- Availability of open-access data and their quality: the use of open-access datasets to validate and benchmark the developed methods thoroughly in the computer science domain. Not only are invasive neural recording datasets more scarce than their non-invasive counterparts, but data quality is also a limiting factor. The authors of the dataset could not fully reply to our questions about the data; some datasets contained very few examples or were incorrectly labelled, among other difficulties.
- Lack of standard metric for ML-based artefact detection and removal: while developing and implementing ML methods, the comparison against the state-of-the-art is widely used, however there is a lack of a standard metric to contrast results or a benchmarking dataset.
- Limited literature on ML for invasive neural recordings: due to the limited access to the data, little work has been carried out on this niche topic. While that allows exploring a wide range of unproposed solutions, there are limited publications to compare the progress.

As an additional challenge, the COVID-19 pandemic hindered the ability to travel to the laboratories where the recording of experiments took place. This posed a challenge to have dynamic exchanges, set back the thorough understanding of the experimental setups, and delay the transfer of data and reporting results. Due to this, alternatives had to be devised in regards to acquiring data.

1.5 Major Contributions

The major contributions of this thesis are summarised as follows:

- A standard metric to benchmark the performance of ML-based artefact detection and removal, integrated into an open-access tool. The

contribution is that the metric is the first attempt to address this major issue in the field, which is composed of a series of comparable characteristics of an ML-based approach.

- An ML-based artefact removal method that is channel-independent. Its impact is due to the capability to remove artefacts in situations where there are single channel recordings, or if in a multiple channel recording multiple channels have been affected, as most methods fail in these circumstances.

- A toolbox that facilitates the process of artefact detection and removal in LFPs via the use of neuronal networks, including the aforementioned method. The significance comes from the lack of ML-based artefact removal methods found in community-developed toolboxes for LFP analysis, and the facilitation of their use to researchers without a computational background.

- An explainable pipeline that allows the detection of brain AD in LFP signals. The impact is due to a higher performance than previously reported in the literature, and in addition the model shows robustness to absent data.

- Objective biomarkers of AD in LFPs. The contribution is the validation of the role of the hippocampus and the role of fast oscillations in AD, which can be further studied in humans via non-invasive methods such as computerised tomography and electroencephalography, respectively.

1.6 Thesis Outline

Given the research aim defined in Section 1.3 about the development of tools and pipelines to facilitate the use of machine learning techniques in chronically recorded invasive signals for disease detection, this thesis provides a series of novel computational solutions to the processing of neuronal signals for disease monitoring. Figure 1.4 visualises the structure of the thesis in the different chapters, where the remainder of this thesis is organised as follows.

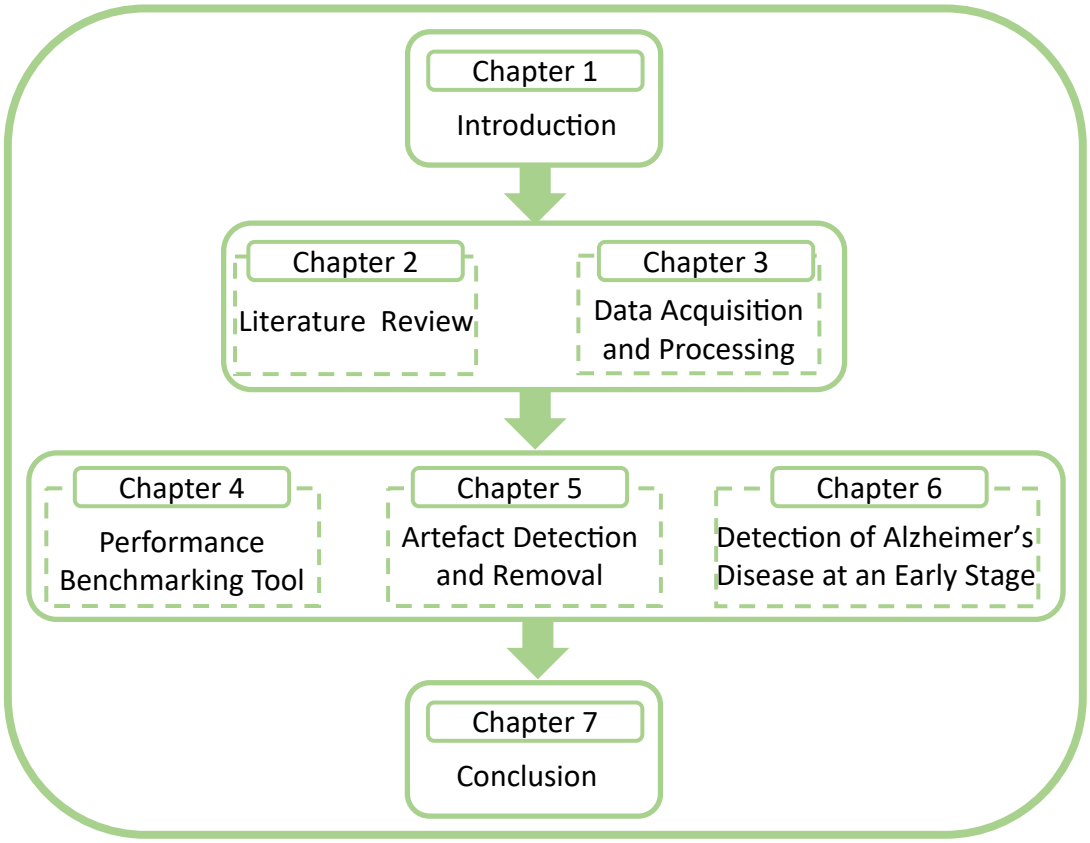

Figure 1.4: Thesis structure showing the organisation of the chapters.

Chapter 2 - Literature Review - This chapter gives an overview of recent work across the topic and thoroughly examines important literature in key areas.

Chapter 3 - Experimental Pipeline - This chapter discusses the methods often used in processing and classifying neural signals. This includes the description of how data was collected and the signal processing and feature extraction.

Chapter 4 - Performance Benchmarking Tool - This chapter describes in detail the developed comparison scale for artefact detection and removal in neuronal signals and the online tool created to make it accessible to the community.

Chapter 5 - Artefact Detection and Removal Toolbox - This chapter presents a solution to the presence of artefacts in neuronal recording via their detection

and subsequent removal. Furthermore, these techniques were incorporated into a toolbox that facilitates their use for those with little expertise in the area.

Chapter 6 - Detection of Alzheimer's Disease at an Early Stage- This chapter presents a novel ML pipeline for AD detection at an early stage on artefact-free signals.

Chapter 7 - Conclusion - This chapter discusses the thesis' conclusions and proposes future research directions on the development of ML-based tools for disease monitoring.

Chapter 2

Literature Review

2.1 Introduction

Early diagnosis of neuronal diseases can aid the patient's recovery or help dampen its progress. Computational techniques such as ML can help to identify singular patterns in data otherwise indistinguishable to clinicians and researchers, in other words, discover new biomarkers and classify new data based on them. With this in mind, neural signals need to be processed to be able to have unbiased results, including the removal of abnormalities. The detection and posterior removal of said abnormalities can also be addressed with ML. While there exist open-access toolboxes for neuronal signal processing, these do not incorporate intelligent methods. From this, it can be claimed that developments in applied ML have the potential to be critical in signal processing for disease staging [23, 24].

In line with this, this chapter provides an overview of ML and the diverse methods it includes. This is followed by a comprehensive review of previous work in signal processing, specifically artefact detection and removal methods, as well as its available toolboxes. In continuation, disease staging methods in invasive neural recordings are discussed. This chapter aims to justify the experimental and development work carried out throughout this thesis by means of identifying, comparing, evaluating and discussing the current state of ML-based processing pipeline methods for disease monitoring.

The remainder of this chapter is organised as follows: Section 2.2 reviews what ML is and the diverse range of techniques that are used. Section 2.3 discusses the methods to detect artefact in invasive neuronal signals. Section 2.4 discusses the methods to remove artefacts in neuronal recordings, emphasising methods used to restore missing values. Section 2.5 discusses the open-access toolboxes to deal with artefacts in neural recordings. Section 2.6 discusses the methods to classify diseases in LFPs. Section 2.7 presents the conclusions drawn from the literature and the research opportunities identified through the analysis of previous work.

2.2 Background

ML techniques are algorithms that are able to learn from data and are able to generalise from it. These algorithms have been applied in diverse fields. They have become popular in recent years due to an increase in computational power which enables the analysis of large volumes of data. There are many ways to group them, for example, by the task (classification, regression, clustering), by type of learning (supervised, unsupervised, reinforcement and other variations) or whether they employ deep neural networks (NNs), defined as non-deep learning and deep learning (DL). In the next sections, first we will overview the different NN-based methods, followed by those methods that do not employ them.

2.2.1 Neural Networks-based Methods

Neuronal networks are algorithms that are composed of "units" called neurons organised in multiple layers, can learn non-linear relationships in data between the input and output, and were inspired by their biological counterpart [25]. Each neuron calculates an inner product of its inputs (x_i) and their respective weights (w_i), then the bias, b, is added and finally the nonlinear activation function is applied, which in most cases is a sigmoid function, tan hyperbolic or rectified linear unit. Thus, the output of a neuron can be expressed as detailed in Equation 2.1.

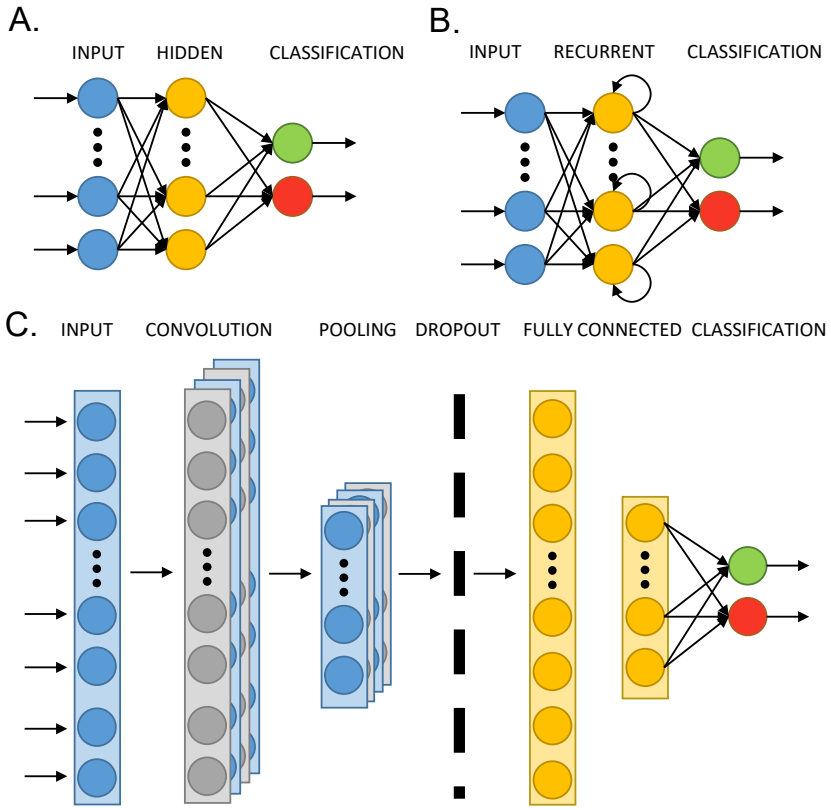

Figure 2.1: Architectures of different neural network models: multi-layer perceptron (A), long short-term memory (B) and one-dimension convolutional neural network (C). Each circle represents a neuron, multiple rectangles a layer's depth, and the arrows how the information is propagated throughout each network.

$$z_i = f(\sum_{i=1}^{n} x_i w_i + b) \tag{2.1}$$

To propagate the information and train the network, the output of a layer is fed as input to the subsequent unit in the next layer. The result of the final output layer is used as the solution for the problem.

NN architectures have many variations based on their principles in determining their rules [26]. The architectures that have been applied and discussed throughout the thesis are described in the subsequent sections, and diagrams of their main components are illustrated in Figure 2.1.

2.2.1.1 Multilayer Perceptron

A multilayer perceptron (MLP) consists of no less than 3 parts: an input layer, a hidden layer and an output layer, where the units of the latter two use the previously mentioned nonlinear activation functions (see Fig. 2.1.A). When it contains multiple numbers of hidden layers, it helps in modelling complex nonlinear relations better than the shallow architecture [27]. For both shallow and multilayered networks, the neuron applies the hyperbolic tangent (*tanh*) activation function as shown in Equation 2.2.

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.2}$$

Here $f(x)$ acts as a step function which squashes a real-valued number to the range [-1, 1] and mimics the behaviour of the neural activity. The *tanh* also has its derivative, which takes the following form:

$$f'(x) = 1 - f(x)^2 \tag{2.3}$$

Now, assuming a network with $l = 1, 2, 3, ..., L$ layers, with each layer containing n_l neurons. The network maps from $\mathbb{R}^{n_1}$ to $\mathbb{R}^{n_L}$. The matrix of weights at layer l is expressed as $\boldsymbol{W}^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, and the weight that a neuron j at a layer l applies to the output from a neuron k at a layer $l-1$ as $w_{jk}^{[l]}$. Equivalently, the vector of biases for layer l is expressed as $b^{[l]} \in \mathbb{R}^{n_l}$, and a neuron j at layer l uses the bias $b_j^{[l]}$. Thus, the output of a neuron j at layer l, given an input $x \in \mathbb{R}^{n_1}$ can be expressed as Equation 2.4.

$$a^{[l]} = f(W^{[l]} a^{[l-1]} + b_j^{[l]}) \in \mathbb{R}^{n_l}, \text{for } l = 1, 2, 3, ..., L. \tag{2.4}$$

To train the network and propagate the information, the output of a layer is fed as input to the subsequent unit in the next layer.

In the case of our dataset, with N training bins in $\mathbb{R}^{n_1}$, $\left\{x^{\{i\}}\right\}_{i=1}^{N}$, there are target outputs defined as $\left\{y\left(x^{\{i\}}\right)\right\}_{i=1}^{N}$ in $\mathbb{R}^{n_L}$ that correspond to the class label. Taking these definitions into account, the goal is to reduce the quadratic cost function shown in Equation 2.5, in other words, reduce the error of the model's output to generate accurate predictions.

$$Cost = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|y\left(x^{\{i\}}\right) - a^{[L]}\left(x^{\{i\}}\right)\right\|_2^2, \tag{2.5}$$

2.2.1.2 Convolutional Neural Network

The modules that characterise Convolutional Neural Networks (CNN) are the convolutional layers and the pooling layers. The former aims to extract high-level features, and the latter's to reduce the spatial size and summarise the information. Overall, the network learns filters that identify specific aspects of the input [28]. The layers of these networks are illustrated in Figure 2.1 (C), and the operations carried out by each one are as follows:

Fully connected The output of fully connected layer y_i given an input x_i and W the matrix of weights, can be expressed as:

$$y_i = f(Wx_i + b_i) \tag{2.6}$$

The fully connected layers applies a non-linear transformation and combination to the features extracted by the convolutional layers.

Dropout The dropout layer y_i turns off a random set of neurons, expressed as:

$$y_i = (1-p)f(Wx_i + b_i) \tag{2.7}$$

where p is the dropout distribution. The dropout layer prevents overfitting by avoiding neurons co-adapting, to correct the error of other ones, achieved by making their presence unreliable.

Convolution Layer Given a input vector $x_{i_{hw}}$ of m x n dimensions where $h = 1, ..., m$ and $w = 1, ..., n$, and filter H of size l x l, the resulting vector $y_{i_{hw}}$ of the convolution between the two is expressed as:

$$y_{i_{hw}} = \sum_{u=1}^{l}\sum_{s=1}^{l} H_{us}x_{i_{(h,u)(w,s)}} \tag{2.8}$$

The convolution layer extracts feature maps, via the convolution of the input with filters (or kernels), that highlight certain characteristics of the input.

Pooling For a feature map x_i having dimensions x_{i_h} x x_{i_w} x x_{i_c} the dimensions of output y_i obtained after a pooling layer are:

$$\frac{(x_{i_h} - f + 1)}{s} \text{x} \frac{(x_{i_w} - f + 1))}{s} \text{x} x_{i_c} \tag{2.9}$$

where s is the stride, and f is the filter size.

The pooling layer reduces the number of parameters of the feature maps generated in the convolution layers, and summarises them to the most important information, making the model more robust to variations of positions in the input.

Softmax The activation function softmax is used to calculate d class probabilities, expressed as:

$$y_i = \frac{exp(x_i)}{\sum_{j=1}^{d} exp(x_j)} \tag{2.10}$$

2.2.1.3 Recurrent Neural Network

A recurrent neural network (RNN) (see Figure 2.1 (B) is a feed-forward network spanning adjacent time steps in a manner that, at every point, the neurons take the current data input and the hidden neurons values that collect the information of the previous time steps [29]. Despite their popularity, these models are hard to train as they suffer the vanishing gradient problem caused by the re-application of the hidden layer's weight to themselves during back-propagation. Long-Short Term Memory (LSTM) is a sub-type of RNN that can overcome this problem by fixing the weight to value 1.

The long-short term memory cells include a forget gate which decides what information is kept and what information is discarded from the cell state. If the value of the forget gate f_t or $f(t)$ is 1, the relevant information is saved, but if the value of the forget gate is 0, it is forgotten. Equation 2.11 shows the mathematical expression of this specific long-short term memory cell.

$$
\begin{aligned}
f_t &= \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f), \\
i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \\
\tilde{c}_t &= tanh(W_{\tilde{c}h}h_{t-1} + W_{\tilde{c}x}x_t + b_{\tilde{c}}), \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \\
o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o), \\
h_t &= o_t \cdot tanh(c_t)
\end{aligned}
\tag{2.11}
$$

where the variable x_t is the input vector, W holds the weights, b is the bias, and σ is the sigmoid function. Additionally, f_t is the forget gate, i_t is the update gate, $\tilde{c}_t$ is the cell input, c_t is the cell state, o_t is the output gate and h_t the hidden state or output vector of the cell at time t.

2.2.2 Non-Neural Networks Based Methods

While there are a substantial amount of non-neural network-based methods, three main models were explored that can be used to deal with non-linear data such as neural signals. These techniques were chosen due to their popularity in supervised learning tasks and are the following:

2.2.2.1 Support Vector Machines

Support vector machines (SVM) are learning machines premised on the statistical learning theory developed by Vapnik et al. [30]. From the classification context, at a given configuration of SVM, it optimally separates the classes by determining the maximal margin hyperplane defined as:

$$
f(x) = x'\beta + b = 0 \tag{2.12}
$$

where x' is the transpose of input features, and $\beta$and b are parameters of the hyperplane. This optimal hyperplane should maximise the margin between the data from the positive class and the negative class. For some inseparable data, a penalty is imposed on the objective function for the sample located on the inverse

side of the class boundary as follows:

$$\begin{aligned} \beta, b = argmin_{\beta,b} \frac{1}{2} \|\beta\|^2 + \frac{1}{2} \|b\|^2 + g \sum \xi_j \\ s.t. \\ y_i f(x_j) \geq 1 - \xi_j \\ \xi_j \geq 0 \end{aligned} \tag{2.13}$$

where ξ_j represents the slack variables used to penalise the samples crossing the margin boundary and g is the penalty multiplier to control the maximum penalty imposed on margin-violating samples. The Lagrange multiplier method is applied to optimise the objective function and find the optimal hyperplane. After the optimal hyperplane is found based on the training set, the SVM is well trained.

2.2.2.2 k-Nearest Neighbours

The k-Nearest Neighbours (kNN) [31] classifier employs metrics of distance to labelled examples as to categorise new ones. For example, Euclidean distance can be calculated with Equation 2.14.

$$D(q, p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2} \tag{2.14}$$

where p and q are subjects to be compared with n characteristics. The parameter k indicates the number of neighbours that will be chosen to compare, which has a major influence on the accuracy of the kNN algorithm. By choosing a large k, one risks ignoring small but important patterns, reducing the impact of variance caused by random error. Thus, some authors suggest setting k equal to the square root of the number of observations in the training dataset [32].

2.2.2.3 Decision Trees

Classification and regression trees [33] can be used to build both classifications and regression trees, where each internal node has exactly two outgoing edges, namely binary trees. The splits are selected using Gini index as a splitting criterion

and the obtained tree is pruned by cost–complexity pruning. The Gini index is described in the following equation:

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{2.15}$$

where p_i is the probability of an object being classified to a particular class. Binary classification decision trees (CDT) are easier to interpret than other ML models, as the decision process in the tree is easily apparent and mimics the rule-based systems used for medical diagnosis. CDTs are ML methods in which the decision process is both local and global and integrated into the methods themselves.

The performance of CDTs can be further improved by aggregating models, namely bagging and boosting [34]. For bagging, given a dataset $D = (x_1, y_1), ...(x_n, y_n)$, sample T sets of n elements from D (with replacement), obtain $D_1, D_2, ..., D_T \rightarrow T$ quasi replica training sets, followed by train a machine on each $D_i, i = 1, ..., T$ and obtain a sequence of T outputs $f_1(x), ...f_T(x)$. The final classification score is expressed as:

$$\bar{f}(x) = sign(\sum_{i=1}^{T} f_i(x)) \tag{2.16}$$

2.2.2.4 Clustering

Clustering encompasses a group of non-supervised ML methods, whose goal is to group elements in a dataset so that they are those within each group are more similar to each other than to those in other ones. Examples of its use in artefact detection include K-means[35], divisive mode hierarchical clustering [36] or Gaussian mixture clustering [37]. They can be classified into partition based (e.g. K-means), distribution based (e.g. Gaussian mixture clustering), hierarchy based (e.g. divisive mode hierarchical), density based clustering, grid based and model based clustering [38].

2.2.3 Performance Metrics

It is very important to select an appropriate performance metric to evaluate a method's performance [39]. If a method aims to assign a label to an example, the task is a classification. On the other hand, if the aim is to predict a continuous number or series of numbers, the task is a regression. The most recurrent metrics found in the literature for each [40], are elaborated below .

2.2.3.1 Classification

When evaluating a model, four different outcomes can be obtained in a confusion matrix: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). From them, the following scores can be measured:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ Sensitivity &= \frac{TP}{TP+FN} \\ Specificity &= \frac{TN}{FP+TN} \\ F1score &= \frac{2\times TP}{2\times TP+FP+FN} \end{aligned}$$

Accuracy is the amount of correct predictions made over the total. While it can be useful when the classes are balanced, in cases where a class is present in a much lower frequency (e.g. of 100 patients 1% has AD), a model with 99% accuracy that does not detect any of them is misleading. Sensitivity is the proportion of known positives that are predicted correctly, and similarly, specificity the proportion of known negatives that are predicted correctly. The issue with these two metrics is that if we have models that assign all examples to positive or negative, then the sensitivity or specificity will be high without accounting for the FP or FN, respectively. Alternatively, the F1 score is a more robust metric that balances both sensitivity and specificity, however it does not factor in the TN.

Additionally, it is possible to calculate the Area Under the Receiver

Operating Characteristic (AUROC), the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example, expressed as:
$P(score(x^+) > score(x^-))$.
In a multi-class classification, there are two manners of calculating AUROC: a macro average (calculate it with all the correct classifications and incorrect classifications across all classes) or micro average (calculate it per pair of classes, creating N metrics equal to the number of classes).

2.2.3.2 Regression

For this task, the performance metrics indicate how efficiently the artefact has been removed, or the original artefact-free signal has been reconstructed. One of the most popular metrics is the root mean squared error (RMSE), which is defined as;

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

where $x(i)$ and $y(i)$ are the restored signal and the original artefact-free signals, respectively, and n is the number of samples. This was chosen over the mean absolute percentage error (MAPE) due to the fact that the signal has been zero-centred during the pre-processing, so the number of zero crossings a segment has is significant, which distorts the MAPE as it takes an undefined value in those points and they must be removed.

2.2.3.3 Clustering

As the examples used to train a clustering model are not labelled, evaluating the performance can be a complex task. The silhouette metric [41] is utilised to evaluate the clusters of a dataset, and is defined as:

$$Sil_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

where a_i is the average distance of point i from all other points in its cluster and a_i

is the smallest average distance of i to all points in any other cluster. The metric is bounded between -1 and 1, where negative values indicate that the point i is in the wrong cluster, and positive the opposite. In cases where the metric is 0, it is an indication of an overlap of the clusters. Other metrics include the Dunn index to evaluate the clustering [42] and the Elbow metric to find the optimal number of clusters in K-means [43].

2.2.4 Evaluation Strategy

Further to selecting appropriate metrics for the specific classification problems, adopting an adequate cross-validation strategy is crucial to guarantee the robustness of the results. Leave-Out cross-validation strategies are widely employed for evaluating DL models with large sample sizes [44]. By the employment of Leave Out, the dataset is shuffled and split into 3 partitions.

The training set is composed of the set of data that will be used to train the model. The validation set is the set of data used to have an unbiased evaluation of a model trained on the training data while tuning model hyperparameters. Lastly, a test set is a set of data used to evaluate the hyper-parameter tuned model unbiasedly. Common partitions include 70-15-15 %, or 80-10-10 % for training, validation, and testing, respectively. The strategy of 70-15-15 % is used throughout Chapters 5 and 6, as there is sufficient data to train the models.

2.2.5 Explainability

Machine Learning techniques can make errors in their predictions without providing information about what exactly causes them to arrive at their predictions. This limits the usefulness of these models, especially in areas such as healthcare [45]. In this context, interpretability is when an observer can understand how a model made a certain prediction, which differs from explainability, where an additional method is used to understand how the model works.

There are some models which are globally interpretable, i.e. they possess meaningful parameters and features from which useful information can be extracted in order to explain predictions, such as DT. We can obtain the feature

importance of the DT model via the calculation of the Gini Index (see Eq. 2.15). Despite the benefits of using an interpretable model, the performance of decision trees can be further improved by aggregating models. However, when there is an ensemble of a large number of trees, it is no longer possible to represent the resulting statistical learning procedure using a single tree, and it is no longer clear which variables are most important to the procedure. Thus, bagging and boosting improve prediction accuracy at the expense of interpretability.

Regarding NNs, they require an explainable method in the form of Grad-CAM heatmaps [46] to understand why the individual example was classified. The heatmaps are generated by visualising the gradients of the last convolution layer's feature maps for a given signal. The visualisations are class-discriminative, in other words, a feature area is associated with the class with strongest reliance on that area. Applying the approach to the neuronal signals reveal the intuitive class discriminative regions that our model have learned.

Another explainable method for ML methods is the Shapely values [47]. The Shapley values are calculated by observing how changes in the input features affect the model's output. In other words, Shapley values are defined as the average marginal contribution for each input feature to the overall model score.

To summarise, ML can be aided with explainable methods to gain insight into their decision making, aiding physicians to complement their own. Likewise, the models benefit from the feedback generated as a result of the researchers' feedback on erroneous diagnoses, improving their performance.

2.3 Artefact Detection in Invasive Signals

Non-invasively recorded neuronal signals are easier to acquire than their invasive counterparts regarding experimental setup and the required protocols and ethics committee approvals [48]. Furthermore, they are more susceptible to artefacts during the recording process [18]. These two factors of data availability and the need for solutions have tilted the development of artefact detection and removal techniques towards them (evidence is presented further down in the thesis in Figure 4.6 of Section 4.3). Only a few techniques have been developed for invasive

recordings and a limited number use ML for the task. The following methods are reviewed in order of spatial resolution of recording: electrocorticography (ECoG) comes first, LFP second and neuronal spikes last.

Alagapan et al. [49] developed an artefact removal algorithm for ECoG labelled shape adaptive non-local artefact removal (SANAR). This approach works by approximating the Euclidean median of kNN of each artefact in a non-local manner, acquiring a template of the artefact, which then is removed from the original signal. It was applied to data obtained from a single subject carrying out a working memory task while being simultaneously stimulated, as well as a simulated ECoG and direct cortical stimulation, where an antenna connected to a function generator acts as a virtual dipole, and a saline solution emulates the conductivity of the grey matter. Artefact residue index was used to measure performance, which should be close to 0. ICA achieved 0.430 ± 0.015, while SANAR 0.388 ± 0.011, reaching better performance. Nonetheless, one must consider the extended calculation time as one of the main limitations of the method.

From another perspective, Tuyisenge et al. [50] developed a model for detecting bad channels in ECoG recordings of seizure patients undergoing pre-surgical recordings and stimulation. They extracted the correlation, variance, deviation, amplitude, gradient, Hurst exponent and Kurtosis from each channel and fed it to a bagging tree model for classification. They explored the model's performance based on the number of subjects used to train it, which plateaued at 99.7% accuracy with 110 subjects. The wrong channels consisted of artefacts such as electrode pop, power line noise and intermittent electrical connection.

Nejedly et al. [51] proposed using CNN with five different frequency bands of the recordings as inputs to identify between physiological, pathological, noise and muscle activity and power line noise. Their analysis was made using two large datasets. They made a general model (trained with one dataset and validated with the other) and a specific model (retraining the available model with 8% of the second dataset and validating with the remaining data). The general model achieved an F1 score of 0.89 in the noise and muscle activity class, while the specific model achieved 0.98 and 0.97 in power line and noise and muscle activity

classes, respectively. The overall performance of the specific model was 0.96, including the physiological and pathological ECoG classes.

For LFPs, no article was found on ML applications on artefact detection within a signal. However, Li et al. [52] carried out a bad channel detection, where twelve hand-crafted features were fed to a wide range of ML classifiers, where the random forest achieved the best performance of 90.02% accuracy in LFPs recorded from pigeons. Having said this, the research done by Bukhtiyarova et al. [53] for identifying slow waves in LFP can be extrapolated to identify segments without artefacts. Slow waves were acquired from 5 young (2-3 months) male C57Bl/6 mice and manually tagged based on their characteristics, while everything else was tagged as "noise". Slow waves and "noise" exclusive segments were used to train the NN, which consisted of an input layer, a hidden layer of 20 neurons and two output neurons, one for each class. After 3 to 5 iterations of training, the best performance achieved was 96.1 $\pm$ 1.6% accuracy. As a disadvantage, the tagging process requires an expert to differentiate slow waves from "noise" to preselect corresponding templates for each recording electrode.

Lastly, regarding neuronal spikes, Klempivr et al. [54] approached artefact detection using transfer learning with a CNN based on AlexNet. The dataset was composed of thousands of ten-second extracellular microelectrode recordings of 58 patients with Parkinson's disease. Approximately 75% of the recordings did not contain any artefacts, and the preprocessed dataset consisted of nearly 100,000 one-second signal segments. Continuous wavelet transform was applied to generate a time-frequency image, which was the input to the network. This pipeline attained an accuracy of 88.1% for artefact identification and 75.3% accuracy for the individual classes of artefact identification.

From another angle, Hosny et al. [55] explored the use of machine learning to detect artefacts from multi-electrode recordings. Their data consisted of recordings from 17 Parkinson's disease who showcased artefacts such as mechanical motion, electromagnetic interference, baseline drift, irritated neuron and others. Power spectral density and wavelet packet decomposition was used to obtain 106 features, which were used to train classifiers such as Gaussian SVM, decision trees, AdaBoost, Bagging learners, LogitBoost and an LSTM network with 3785 examples. The best performing model was the LSTM

network, with an accuracy of 97.49% on the test set.

These articles reveal that obtaining good performance with ML classifiers for artefact detection is possible. This presents an opportunity for LFPs, where there little research has been carried out so far, as listed above. Furthermore, a diverse range of ML and DL models have been used, indicating that the use of the latter without depending on feature extraction is feasible.

2.4 Artefact Removal

A brief discussion, of the typical computational approaches used for artefact removal, is provided in the following paragraphs.

Regression: A regression method begins by defining the relationship in amplitude between a reference channel and a neural signal using transmission factors, then removing the estimated artefacts from the signal [56]. In a single-channel approach without a reference channel, this approach is not possible. Furthermore, a reference channel can be used for a sub-set of artefacts that are known to affect the recordings beforehand.

Adaptive Filtering: To apply adaptive filtering, a reference channel is given as one of the inputs to the filter, so the degree of artefactual contamination in the neural signal is measured by iteratively changing the weights according to the optimisation method and then removed [57]. As with regression, the lack of a reference channel invalidates applying this approach. To improve the decoding accuracy of a force signal, Khorasaniet al. [58] applied a weighted common average referencing (CAR) algorithm in combination with Kalman filtering in LFPs.

Template Subtraction: When artefacts have a unique shape, as they come from a specific source, they can be approximated and subtracted to restore the neural signal [59]. As a result of the variance of the shapes of the artefacts in the datasets, as they can be of different unidentified sources, make it impossible to accurately subtract them without introducing further error. Qian et al. [60]

proposed a technique for removing stimulation artefacts in LFPs during deep brain stimulation, based on stimulus artefact template reconstruction and subtraction.

Inter-Channel Interpolation: When a channel in an array is impacted locally by an artefact, that segment can be replaced using the average or other methods that consider the surrounding channels, which isn't possible in a single channel approach [61]. Mahmud et al.[62] approached the removal of microstimulation-induced stimulus artefacts in LFPs via identifying an artefact's length based on signal derivative and removing it from the recordings through interpolation.

Decomposition: These methods break down a signal into a series of low frequency and high frequency components, via filters (wavelet) or maxima and minima (empirical mode). One major drawback of decomposition methods is that they cannot remove artefacts completely if the measured signal's spectral properties overlap with the artefacts' spectral properties of the artefacts [63]. Most of the artefacts manifest in the same bands as the physiological signal, and if more than one is present, they may even overlap with each other.

Blind Source Separation: Blind source separation is a popular method for removing artefacts in neuronal signals and includes methods such as Independent Component Analysis (ICA), canonical correlation analysis, and principal component analysis [18]. It is defined as methods that aim to separate unknown and independent sources using observed signals, where these sources appear mixed, and is called blind due to no other supporting information being used besides the mixtures. However, these methods assume that the number of artefact sources should at least be equal to the number of channels, limiting the single channel applications.

From a perspective of restoration of missing values in neuronal signals, there have been approaches that are and are not based on ML applied to electroencephalography (EEG) signals. From the first group, Svantessona et al. [64] trained a CNN trained a CNN to upsample from four or fourteen channels, or to dynamically recover single missing channels in order to reconstruct a

21-channel recording. A visual evaluation by board-certified clinical neurophysiologists was conducted, and the generated data was not distinguishable from real data. In a similar approach, Saba-Sadiya et al. [65] employed a convolutional autoencoder, which takes as an input a padded EEG electrode map during 16ms (8x8x8 tensor) with 1 occluded channel, which is expected as the output. They compared it to spherical splines, Euclidean distance and geodesic length methods, outperforming them and showing the method can restore the missing channel with high fidelity to the original signal. Lastly, Thi et al. [66] utilised a linear dynamical system (Kalman Filter) to model multiple electroencephalogram signals to reconstruct the missing values. This method showed 49%, and 67% improvements over singular value decomposition and interpolation approaches, respectively.

In the second group, there are published papers such as de Cheveigne and Arzounian [67], and Chang et al. [68]. In [67] authors have detected EEG and magnetoencephalography (MEG) artefacts by their low correlation to other channels and replaced them with the weighted sum of normal channels, a method called 'Inpainting'. On the other hand, Chang et al. employed artefact subspace reconstruction on twenty EEG recordings taken during simulated driving experiments, in which large-variance components were rejected and channel data were reconstructed from remaining components improving the quality of a subsequent ICA decomposition. Sole-Casals et al. [69] evaluated the performance of four tensor completion algorithms and average interpolation across trials on missing BCI data (across 6 channels and segments) and evaluated the reconstruction by the performance of machine-learning-based motor imagery classifiers.

These approaches rely on the information from other channels of the arrays, thus they fail when a global artefact is present or if the number of affected channels is more than the ones not affected.

2.5 Toolboxes for Neuronal Signal Processing

Research groups in the neuroscience community have developed, and shared toolboxes for analysing neural recordings [70, 71, 72]. Given the wide range of

neuronal signals, data formats, analysis techniques and purposes, each one has advocated their efforts into specific elements. Table 2.1 lists the available open toolboxes and their functions regarding aiding noise detection and removal in LFP signals. An in-depth analysis of these toolboxes is reported in [73]. Hence, the description below will be dedicated to elaborate on the reported toolboxes.

Brainstorm [74] is an open-source application dedicated to neuronal data visualisation and processing, emphasising cortical source estimation techniques and their integration with anatomical magnetic resonance imaging data. It offers an intuitive interface and powerful visualisation tools, and its database structure allows the user to work at a higher level. BSMART [75] is a toolbox intended for spectral analysis of continuous neural time series data recorded simultaneously from multiple sensors. It is composed mainly of tools for auto-regressive model estimation, spectral quantity analysis and network analysis. All functionality has been integrated into a graphical user interface environment designed for easy accessibility.

Chronux [76] is an open-source Matlab software project for analysing neural

Table 2.1: Open source toolboxes and artefact detection and removal functionalities.

Toolbox	SADR	UNoC	SE	Up	AR	DF	DV	SA	SAR	FO	MF
Brainstorm [74]	X	✓	✓	✓	✓	✓	✓	✓	X	X	✓
BSMART [75]	X	✓	✓	X	X	X	✓	✓	X	X	✓
Chronux [76]	X	✓	✓	X	X	X	✓	✓	X	X	X
Elephant [77]	X	✓	✓	✓	X	X	✓	✓	X	X	✓
Fieldtrip [78]	X	✓	✓	X	X	✓	✓	✓	X	X	✓
Klusters, NeuroScope, NDManager [79]	X	✓	✓	X	X	✓	✓	✓	X	✓	✓
Neo [80]	X	✓	✓	✓	X	X	✓	X	X	✓	✓
NeuroChaT [81]	X	✓	✓	✓	X	X	✓	✓	X	X	✓
Spycode [82]	X	✓	✓	X	X	✓	✓	✓	X	X	✓
Sigmate [83]	X	X	X	X	✓	✓	✓	✓	✓	✓	✓

Legend: SADR - State-of-the-art artefact Detection and Removal, UNoC - Unlimited Number of Channels, SE - Supported Environment, Up - Updates, AR- Artefact Removal, DF - Digital Filtering, DV - Data Visualisation, SA - Spectral Analysis, SAR - Stimulation artefact Removal, FO - File Operations, MF - Multiple Formats

signals via signal specialised modules for spectral analysis, spike-sorting, local regression, audio segmentation, and other tasks. Similarly, Elephant [77] is a Python library for analysing electrophysiological data, such as LFPs or intracellular voltages. It offers a broad range of functions for analysing multi-scale data of brain dynamics from experiments and brain simulations, such as signal-based analysis, spike-based analysis and methods combining both signal types.

FieldTrip [78] is an open-source software package developed to analyse electrophysiological data. It supports reading data from many different file formats. It includes algorithms for data preprocessing, event-related field/response analysis, parametric and non-parametric spectral analysis, forward and inverse source modelling, connectivity analysis, classification, real-time data processing, and statistical inference. Klusters, NeuroScope, NDManager [79] are a free software suite for neurophysiological data processing and visualisation. NeuroScope is an advanced viewer for electrophysiological and behavioural data with limited editing capabilities; Klusters is a graphical cluster cutting application for manual and semi-automatic spike sorting, and NDManager is an experimental parameter and data processing manager.

Neo [80] is a tool whose purpose is to handle electrophysiological data in multiple formats. It was added due to its unique property of being able to read or write the data from or to various commonly used file formats. NeuroChaT [81] is a Python open-source toolbox created to standardise open source analysis tools available for the analysis of neuronal signals recorded *in vivo* in the freely-behaving animals.

Spycode [82] is a smart tool for multi-channel data processing which possesses a vast compendium of algorithms for extracting information both at a single channel addition to at the whole network level and the capability of autonomously repeating the same set of computational operations to multiple recording streams, all without manual intervention.

Outside of Sigmate, the only one that allows for artefact detection is Brainstorm. It allows for manual inspection and automatic detection of artefacts, mainly of muscular and movement origin, by filtering the signals in frequency bands (ocular $1.5 - 15$ Hz; for cardiac: $10 - 40$ Hz; for muscle noise

and some sensor artefacts: $40 - 240$ Hz and subject movement, eye movements and dental work $1 - 7$ Hz) and classifying the absolute value of signal with a standard deviation threshold. However, artefacts can span a large bandwidth, and studies show that they can overlap with those of the neural signals [84]. For example, the alpha band ($8 - 12$ Hz) can have high amplitude oscillations and be falsely detected as an artefact.

There other toolbox that deals with LFP artefact detection is SigMate [83, 85, 86], a Matlab-based tool that incorporates standard methods to analyse spikes and EEG signals and in-house solutions for LFP analysis. The functionality provided by SigMate include: artefact removal, both fast [87] and slow [84], angular tuning detection [88], noise characterisation [89], cortical layer activation order detection and network decoding [90, 91, 92, 93], sorting of single trial LFPs [94, 95, 96, 97], etc. It deals with slow stimulus artefact removal through an algorithm that subtracts an estimation of the signal by averaging the peaks and valleys detected in it, eliminating the offset. In addition, it allows for visualisation of the spectrogram using a short-time Fourier transform of the recording to allocate artifactual frequency bands and allow their filtering, among many other analysis functionalities. Despite its many options, this toolbox can be improved with new functions to make it more competitive as it has not been regularly updated, such as allowing an unlimited number of channels, a supported environment, and state-of-the-art artefact detection and removal.

2.6 Disease Monitoring in Local Field Potentials

The spatial resolution of the LFPs allows to have insights into specific brain structures and the network connectivity. This is crucial for the discovery of biomarkers, and improving the understanding of disease genesis and progression. Large amounts of statistical analysis is carried out by researchers, where the difference between conditions can be exploited by intelligent methods. Examples of the use of these signals in conjunction with ML to detect neuronal diseases

include: Parkinson, epilepsy and AD.

Regarding Parkinson's Disease (PD), Darbin et al. [98] applied ML classification models for comparisons between healthy parkinsonians with off-medication and for LFPs obtained from non-human primates rendered parkinsonian. A range of methods, including bagged CDTs, kNN, logistic regression, NN, SVM, CDT, and Naive Bayes, were trained with extracted features such as power of frequency bands of interest, entropy, Higuchi's fractal dimension, rescaled range analysis, and Poincaré where the bagged CDTs achieved the best accuracy of near 90%. Similarly, Bore et al. [99] utilised SVM and kNN trained with the relative power in each frequency band and the phase-amplitude coupling to classify between naive and mild PD states in a nonhuman primate model. Both classifiers obtained a performance nearing 98% in the primary motor cortex, where high-frequency oscillations were shown to be the primary marker.

From the perspective of detecting seizures, Toth et al. [100], Park et al. [101] and Aghagolzadeh et al. [102] applied ML to classification of LFPs. Toth et al. utilised random kitchen sink trained with the discrete wavelet transform relative wavelet energy and multiscale entropy as features to achieve 97.3% accuracy in classifying ictal (i.e. seizure) and interictal states. Park et al. achieved a performance of 93.8% sensitivity of ictal and 99.8% specificity of interictal by training a cost-sensitive SVM with features such as the mean, variance, and Fano factor of the power in each frequency band. Lastly, Aghagolzadeh et al. trained a CNN with a feature matrix of channels by the power spectrum of 2Hz bands spanning up to 100 Hz. The model was able to detect four out of five seizures.

Regarding the detection of AD in rodents, Beker et al.[103] applied an SVM model in classifying LFPs into wild-type or AD model and achieved an accuracy of 82.6%. The model was fit with different features such as: correlation coefficients (R^2), fit sum squared error, trough frequency, trough amplitude standard deviation, half-width amplitude, AC maximal power, Gaussian fit separation, coefficient of variation, inter spike interval, frequency maximum power, mean signal to noise ratio and amplitude kurtosis. The most discriminative feature for the LFPs was the R^2, as normal neuronal activity is bimodal, whereas the AD model has a weaker bimodality due to struggles

between regular state transitions.

These examples show that ML models can be used for classifying LFPs for disease monitoring. Authors explored a wide range of ML models, showing that there is no optimal one for this task as the no free lunch theorem suggests [104]. As most authors listed in this section have not exploited the capabilities of DL methods, as described in the reviews above, this indicates that there is an opportunity to contribute to the field.

2.7 Discussion and Research Opportunity

The literature analysis has revealed that ML can be used to process the signal to detect and remove artefacts. While other computational techniques have been developed for this task, such as the ones mentioned in Section 2.4 [56, 58, 60, 62, 63, 18], most of them can only be applied in multi-channel recordings or work under the assumption that there is no overlap between the neuronal signal and it in the spectral domain. From the perspective of restoring missing values, the techniques are also limited by the expected good quality of the surrounding channels. This suggests that there is a need for computationally effective channel-independent artefact detection and removal techniques.

The review of ML applications for the artefact in neuronal signals revealed that there is a large number of articles that deal with this issue. While the analysis has been carried out in invasive recordings, it is important to highlight that over 100 articles were found for non-invasive recordings. The large literature size can make it daunting to find, analyse, compare and then select either for application to new data or as a benchmark for a new method.

The community has made increasing efforts to generate open-access toolboxes, each developed for different purposes. However, there are still many open challenges in ones that regard the identification and removal of artefacts in chronically recorded neuronal signals. The ones who do address it mostly rely on filtering techniques for general purposes, while computational techniques have not been fully exploited.

On the topic of disease monitoring, the use of ML has proven to achieve successful results across different neuronal pathologies. However, as shown in

the literature, there is still room for improvement. Despite the advances in DL architectures, most of them have not been explored by authors. Furthermore, while most have manually extracted temporal or spectral domain features, little analysis has been done in the spatial domain. Further analysis of the biomarkers can be carried out, as mostly the training individual models for each feature have been used to understand their impact.

The conclusions drawn from the analysis of the literature on computational methods for signal processing for disease staging indicate that there are research opportunities to be explored via ML. With this in mind, the efforts of this thesis are focused on presenting new solutions and making them available to the community via open-access tools.

Chapter 3

Data Acquisition and Processing

3.1 Introduction

This Chapter presents the description of the collected data utilised during this thesis, including the subject's description, the area of the brain the LFPs were recorded from, the processing done to them and the context in which they were acquired. Subsequently, the processing done to them in order to extract the individual examples of the ML models are given, including how the signal is partitioned, and labelled and finally, the description of the extracted features from each example.

3.2 Datasets

Four different datasets are used throughout the thesis, three for artefact detection and removal, and one for AD detection at an early stage. A summary of the four datasets is shown in Table 3.1. For the first task, the recording devices, sampling frequency, area of acquisition, and the purpose of each dataset differ, but all of them were recorded in rodents under laboratory conditions.These differences help validate the method's performance achieved on the private dataset with the various conditions of the open-access ones.

Table 3.1: Characteristics of the datasets utilised in the thesis

Dataset	Type	Purpose	Subjects	Ch	Cl	SF	EpC
1	Private	Stimulation	1	16	2	1,017 kHz	147296
2	Open	Ketamine	4	4	2	2 KHz	137844
3	Open	Sleep	1	1	2	250 Hz	1548
4	Private	AD	30	24	3	500 Hz	6776

Legend:Ch- Channels, Cl- Classes, SF- Sampling Frequency, EpC - Examples per Class, AD- Alzheimer's Disease

3.2.1 Dataset #1 - Private Artefact Dataset

This dataset was obtained via the collaboration of Nottingham Trent University via Dr. Mahmud with Dr. Michela Chiappalone, researcher of the Instituto Italiano di Tecnologia located in Genova, Italy. The collaboration consisted of their part to provide data as well as support the project, and from our part the development of a decision support system for neural signals that would allow for a quicker analysis of the recorded neural signals which will be appropriate for a number of applications in research as well as in clinic.

Five adult, male Long Evans rats weighing 350–400 g at 4 months old were used for LFP signal acquisition. The experiments were performed at the University of Kansas Medical Center, and their Institutional Animal Care and Use Committee approved the protocols for animal use. This approval adheres to the Guide for the Care and Use of Laboratory Animals (National Research Council (US). Committee for the Update of the Guide for the Care and Use of Laboratory Animals., Institute for Laboratory Animal Research (US), and National Academies Press (US), 2011). The detailed procedures were described in [105].

The rats were induced with gaseous Isoflurane prior to surgery within a sealed vaporiser chamber. Anaesthetisation followed with injections of Ketamine (80-100 mg/kg IP) and Xylazine (5-10 mg/kg). Maintenance boluses of Ketamine (10-100 mg/kg/h ip or im) were repeatedly injected as needed throughout the procedure. Either Lidocaine/Prilocaine cream or Bupivacaine were applied to the scalp prior to making a skin incision spanning rostrocaudal between 6 mm rostral to bregma and 5 mm distal to the atlantooccipital junction. A craniectomy was performed to expose the primary motor (Caudal

Forelimb Area: CFA) and premotor (Rostral Forelimb Area: RFA) cortical areas. Electrophysiological procedures were facilitated by the removal of the dura and the application of sterile silicone oil to the cortex.

A four-shank, sixteen-contact site electrode (A4x4-3mm-100-125-177-CM16LP, NeuroNexus) was chronically implanted into the RFA (premotor cortex) at a maximum depth of 1600 μm. A second four-shank, sixteen-contact electrode (A4x4-3mm-100-125-177-CM16LP, NeuroNexus) was chronically implanted into S1 (forelimb area) and used for stimulation through a single contact. All groups were recorded daily for 21 days. Each animal could move freely inside a self-made plastic cage for the entire duration of each experimental session but was not required to perform any sensorimotor tasks. The daily recording consisted of 80-minute periods of stimulation flanked by a 30-minute period (before stimulation) and a 30-minute period (after stimulation) of no stimulation for a cumulative 2 hours and 20 minutes of recorded data per day. For this research, a subset of the data has been used, which comprises only the baseline period of recording before stimulation, from a single animal for 10 days. The neural recordings were downsampled to 1,017.3 Hz and low-pass filtered (with cutoff frequencies of 0-500 Hz and without removing the 60 Hz noise) in order to obtain LFPs from them.

3.2.2 Dataset #2 - Open access Artefact Dataset

A publicly available dataset [106] was selected to further test the methods in a reproducible manner, as researchers can access the data and duplicate the results. Thorough details of the recording and experiment are explained in the article linked to the dataset [107]. Male Long Evans rats (Charles River, Frederick, MD, USA) weighing from 280 to 300 g were trained to walk on a circular treadmill. The recorded LFP were sampled at 2 kHz, and after low-pass filtering, they were amplified times a thousand and band-pass filtered (0.7–150 Hz).

For the purpose of our analysis, only the baseline recordings (prior to Ketamine injection) were used. Baseline recordings were composed of at least two five-minute counter-clockwise walking cycles on a slow-moving treadmill

and two 40-second rest periods without artefacts. Visual evaluation and videotaped motor activity were used to classify artefact-free periods of 100 seconds in treadmill-on epochs and 40 to 100 second periods in treadmill-off epochs.

Only a subset of rodents with no visible discontinuities in the LFP signal, i.e., significant interruptions in the recording, were employed. They are detailed in Table 3.2. After contacting the researchers who acquired the data, they responded that the poor quality can be attributed to construction processes above the laboratory (such as turning on equipment, etc) and that the electrical noise was also worse on more humid days.

3.2.3 Dataset #3 - Open access Artefact Dataset

A second open-source dataset [108] was selected based on the amplitude of the artefacts, which were ranging between 0.15% and 13.48% of the recordings, as highlighted by the authors of related works. The open-access dataset is composed of uninterrupted baseline recording days for sleep research, where LFPs were recorded from 9 male Sprague-Dawley rats (3–4 months old). The dataset contains LFP that was acquired at the prefrontal and cortex parietal cortex, sampled at 250 Hz. Recordings were cut into 4-second long epochs and labelled depending on the state of the animal (awake, rapid eye movement, or

Table 3.2: Second dataset's guide to determine best channels and epochs to use of baseline walk and rest recordings in the medial prefrontal cortex (mPFC) and the mediodorsal (MD) thalamus, as mentioned in the file named "Coherence Phase Plot Guide". The first column is the rat identification, columns 2 and 3 are the selected two best channels of the mPFC recordings and 4 and 5 of the MD recordings. Finally, column 6 shows the range of artefact-free epochs during walking and column 7 during resting, respectively

Rat	mPFC chan1	mPFC chan2	MD chan1	MD chan2	walk epoch	rest epoch
KF9	5	6	3	7	960-1160	3780-3820
KF10	3	4	3	8	670-860	1260-1390
KF14	2	6	5	7	740-940	3350-3550
KF15	3	4	5	7	450-640	1600-1700

Table 3.3: Third dataset's total time of awake segment per rodent part 1.

rodent	recording ID	segment (s)	total (s)
AsiagoBleu	180626	3508	9500
	180627	1776	
	180628	1728	
	180629	2488	
bobmarley	102819	4236	18196
	102919	4776	
	103119	4228	
	110319	3012	
	110419	1944	
cheaptrick	110619	2480	17240
	110719	11608	
	111319	3152	
EZBrie	180707	4320	19064
	180708	4320	
	180709	3228	
	180710	7196	
FetaMozz	180628	7124	17340
	180629	6860	
	180630	3356	

non-rapid eye movement sleep).

It is worth noting that the dataset has intra-subject variability, as these recordings range from 3 to 8 consecutive days out of 40 that are not shared. Furthermore, there are differences between states such as high-frequency components which may distort the detection and removal of artefacts. Therefore, in order to reduce the variability, the longest awake period of each day was extracted (see Table 3.3 and Table 3.4) and chose the rodent with the longest consistent awake recordings (i.e., rodent 'MuensterMonty'). The final dataset is composed of the recordings of one rodent during the awake state across five recording sessions for a total of 26956 seconds.

Table 3.4: Third dataset's total time of awake segment per rodent part 2.

rodent	recording ID	segment (s)	total (s)
Manchego	180620	4456	16156
	180621	5396	
	180622	2188	
	180623	4116	
MuensterMonty	180720	3408	26956
	180721	7648	
	180723	5024	
	180726	3380	
	180727	7496	
NachoGouda	180705	2656	9140
	180706	2496	
	180707	2136	
	180708	1852	
neilyoung	111719	2840	26516
	111819	3320	
	111919	3272	
	112119	3284	
	112219	4148	
	112619	4040	
	112819	3200	
	112919	2412	

3.2.4 Dataset #4 - Private Alzheimer's Disease Dataset

This dataset was obtained via the collaboration of Nottingham Trent University via Dr. Mahmud with Dr. Cristina Fassolato, researcher of the University of Padova located in Padova, Italy. The partnership consisted of their side providing data and collaborating on the study, and our part exploring the differences between control and animal models using ML [10], in order to exploit further the data they've gathered.

Sets of ten three-month-old female healthy C57BL/6J (WT), single transgenic PS2.30H (ST) and double transgenic B6.152H (DT) mice were used. All of the animals were kept in a specific pathogen-free animal facility with a 12-hour duration of light and dark cycles, and unrestricted access to food and water. All experimental procedures were performed according to the European

Committee guidelines (decree 2010/63/CEE)and the Animal Welfare Act (7 USC 2131), in compliance with the ARRIVE guidelines, and were approved by the Animal Care Committee of the University of Padua and the Italian Ministry of Health (authorisation decree 522/2018-PR). For more details, the reader is referred to the published experimental analysis [10, 109].

The mice were sedated with an intraperitoneal injection of urethane dissolved in 0.9 percent NaCl and a combination of Xylazine/Tiletamine-Zolazepam mixed in phosphate buffer. The spontaneous LFP activity was recorded using a linear 32-electrode silicon probe coupled to the acquisition equipment through a 32-channel head stage and an SPI connection. In order to allow the first 24 channels of the probe to record from the deepest layer of the dentate gyrus up to the cortical layers, the probe was introduced into the posterior parietal cortex and lowered to 2.3 mm.

Through the open graphic user interface software included with the Open-Ephys acquisition equipment, the LFPs were visualised, captured, and digitalised at 10 kHz. Other physiological signals were captured at the same time as the LFPs to evaluate the animal's health status. ECG recordings were 10× amplified and filtered between 1 and 100 Hz using a DAM 50 Amplifier to monitor the heartbeat. Positive and negative electrocardiogram derivations were placed subcutaneously into the forelimbs. The piezoelectric properties of the temperature probe converted respiration-induced chest wall motions into voltage variations. A DP-301 amplifier was used to amplify the respiration signal 100X and band-pass it between 0.1 and 100 Hz. Physiological signals were digitalised at 10 kHz using a PCI-6071E I/O card in the differential mode in conjunction with a BNC-2090 terminal block and recorded using a custom-written LabView script. After obtaining a stable level of anaesthesia, spontaneous brain activity was recorded for 30–40 minutes.

The raw signal files are converted to Matlab files from Open-Ephys format and subsequently, the 50 Hz line noise is removed by applying a Gaussian filter. The first 24 channels' signals were filtered using the built-in non-causal zero-phase distortion filtering algorithm, which in order to avoid phase distortion, the data is processed in both forward and reverse directions using coefficients from the built-in Butterworth transfer function. Using a median estimation approach, baseline

drift was eliminated from all signals (LFP, ECG, and respiration). Afterwards, the recordings were low-pass filtered (filter order: 5; cut-off frequency: 190 Hz for LFP, 25 Hz for ECG, and 10 Hz for respiration) and down-sampled to 500 Hz, 50 Hz and 20 Hz, respectively.

3.3 Processing

In this Section, the different post-processing done to the signals are described, including the labelling of the signals, segmentation of the signal into smaller segments and the features extracted from them.

3.3.1 Labelling

For Dataset #1, the LFPs were manually annotated to obtain ground truth about the artefacts. The annotation process consisted of taking segments composed of 100 data points (referred to as bins) and labelling them as 0 (non-artefact) or 1 (artefact) if the total power of the bin exceeds a predefined power threshold which is defined as the mean power of the non-artefacts signal. The annotation was carried out by researchers who collected the data, and validated the labelling via visual inspection, making use of their expertise. The annotating process didn't aim to identify individual sources of the artefact which can be related to movement, chewing or other different sources.

This concept was utilised to label Dataset #2 & Dataset #3. For the former, the rest periods without artefacts were used to extract the power threshold for each channel, for each defined window size. For the latter, visual confirmation of the lack of artefact was used to define a portion of the signal in order to extract the threshold.

For Dataset #4, three steps were used to identify stable LFP windows using respiratory and ECG data. In the first step, by calculating the respiration and heartbeat rates and using specific upper and lower bounds, anomalous patterns were identified. By taking the median of the individual rates and adding/subtracting heuristically-selected rate constants, whose respective values were 0.5 and 2 for respiration and heartbeats, the boundary values were

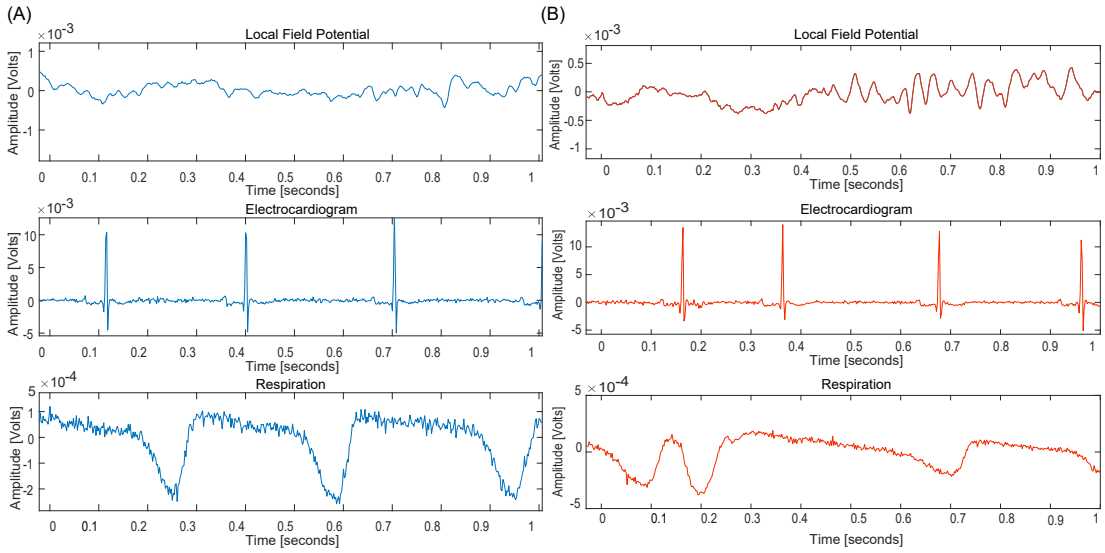

Figure 3.1: One-second segments of local field potential, electrocardiogram and respiration signals in a normal state (A) and stressed state (B) from Dataset #4.

obtained. As a result, a segment of LFP was labelled as normal where both signals were steady and stressed when either one was not stable. Figure 3.1 showcases the behaviour of the three signals under normal (A) and stress (B) cases, where in the latter, there are some fast oscillations in the LFP, irregular heartbeats in the electrocardiogram, and abnormal cycles in the respiration recording. From the longest segment without stress, non-overlapping one-second windows across all channels were extracted from each rodent. The same principle was applied to extract stressed portions of the LFPs. A secondary label indicating the rodent type (i.e., WT/ST/DT) was also assigned to each segment.

Subsequently, all data was normalised in the range of $[-1, 1]$ in order to have a standardised input to the neural networks.

3.3.2 Signal Segmentation

Once the signals are labelled, the following action is to divide the long recordings into shorter segments in order to facilitate the posterior feature extraction and training of the models. The sliding windows approach is used, a

popular method utilised for streams of data, where these are divided into sub-segments time windows of equal duration. The algorithm takes two inputs, the length of the window (N) and the overlap between them (O). As these parameters are often arbitrary, the role of N was explored in the detection and removal of artefacts, while O is set to 0 as there is sufficient data.

3.3.3 Feature Extraction

Deep learning models have the property of being able to learn non-linear patterns from data, and due to this, several classifiers were trained with the segments of the raw signal. This means that for the task of artefact detection and removal, the architecture takes the principal role in extracting relevant features from the data.

In regards to the disease staging, different features were extracted across the different domains. The temporal models used the raw signals as input. From the spatial domain, different metrics were used to evaluate the similarities between two given channels. First, a correlation matrix was built based on the Pearson correlation of a channel x and y, given the equation:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{3.1}$$

where $i = 1, ..., 500$. For comparison, a dynamic time warping (DTW) algorithm was used [110].

In addition, spatial maps based on the mutual information (MI) [111] of any two channels x, and y were extracted using the following equation:

$$MI(x;y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(x(i), y(j)) \cdot log\left(\frac{p(x(i), y(j))}{p(x(i)) \cdot p(y(j))}\right) \tag{3.2}$$

where $i, j = 1, ..., 500$.

Examples of the three extracted spatial features of a WT example are shown in Figure 3.2. In total, three spatial features were extracted from every example, each being a correlation map of the 24 channels based on Pearson correlation, DTW, and MI.

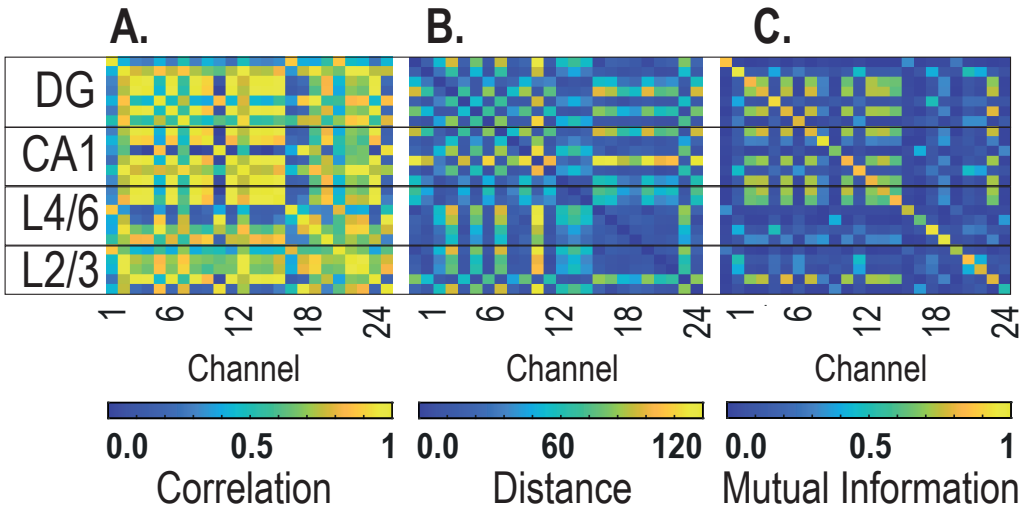

Figure 3.2: Spatial feature maps: correlation (A), dynamic time warping (B) and mutual information (C), across the brain areas: dentate gyrus (DG), first region in the hippocampal circuit (CA1), cortex layers 4 to 6 (L4/6), and cortex layers 2 to 3 (L2/3)).

From the spectral domain, the spectral bands were defined as slow oscillations (0.1-1.7 Hz), delta (1.7-4.7), theta (4.7-10), beta (10-25 Hz), low gamma (25-45 Hz), high gamma (45-90 Hz) and fast oscillations (90-125 Hz). First, the signals across the channels were averaged in the selected areas (1-24 all, 1-7 dentate gyrus (DG), 8-13 CA1, 14-19 cortex layers 4 to 6 (L4/6), 20-24 cortex layers 2 to 3 (L2/3)). Subsequently, the absolute and relative power of all the bands were extracted, and the ratio of the power of slow oscillation to delta bands and the ratio of the power of the bands from 0.1-4.7 Hz to 4.7-125 Hz were also used.

3.4 Conclusion

This section has presented the four datasets utilised in the research. They were described, including the number of subjects, their phenotype and age, the experiment setting, the recording devices, the duration of the recording, the sampling frequencies of the recorded signals and the filtering applied to them. Despite their different experimental settings and sources, they all present artefacts or stress segments in varying proportions, indicating the need for the

developed algorithms in this work. Subsequently, the signal segmentation techniques to break the LFPs into smaller segments were elaborated upon, and how each sub-segment was labelled to its corresponding class. Lastly, the diverse range of hand-crafted features extracted from the fourth dataset across temporal, spatial and spectral domains were described. The results gathered from these datasets with methods developed to contribute to the neuroscience domain are described in the following chapters.

Chapter 4

Performance Benchmarking Tool

4.1 Introduction

A common characteristic of neuronal signals is their vulnerability to artefacts, which can be of internal or external origin, as previously described in Chapter 1. The undesired effects of its presence range from causing a BCI device to operate erroneously, misdiagnosis of diseases or brain conditions (as in the diagnosis of schizophrenia, sleep disorders, and AD [112]) or producing false alarms (as in generating false alarms for brain seizures [113]). A collection of applications that benefit from artefact removal, such as the aforementioned ones, are presented in Figure 4.1.

Given the consequences of artefacts on acquired neuronal signals, many scientists have been interested in developing methods of detecting and removing them [17]. There has been a rise in recent years of new approaches based on ML techniques [114], as there are benefits of employing them over other computational methods.

Selecting the best method which suits the researcher's experiments, can be challenging for experimenters who don't need to know the technical details of the artefact detection and removal process to select an appropriate method for their acquired signals [115, 1]. In the literature, meta-studies of artefact detection and removal methods are compiled in Table 4.1. It is shown that not all reviews describe the various artefacts and that all of them focus exclusively on EEG and

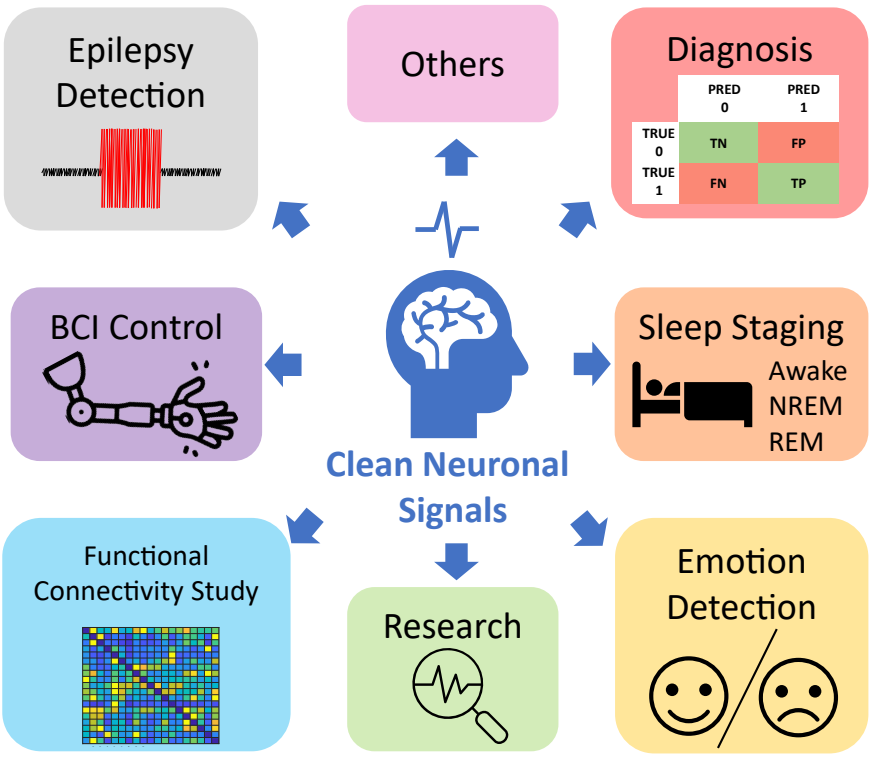

Figure 4.1: Overview of the application areas that benefit from the removal of artefacts.

not other neuronal signals. Furthermore, only four of them mention ML-based methods, with only a small subset of the available literature compared. Overall, there is a lack of a holistic overview of the artefact detection and removal across all neuronal signals from an ML perspective.

To address these challenges and facilitate access to appropriate ML and data-driven artefact detection and removal methods, ABOT (Artefact removal Benchmarking Online Tool) was developed. The literature was surveyed as resources for the tool to create an up-to-date dataset and define key features for users to compare.

The remainder of the Chapter is divided into five sections, where Section 4.2 describes the online benchmarking tool from the software development perspective. Subsequently, Section 4.3 covers the creation of the bibliographic dataset. Lastly, Section 4.4 discusses challenges and future perspectives within the field and Section 4.5 makes concluding remarks.

Table 4.1: Comparison of available reviews on methods applied to artefact removal, sorted by year of publication. The table is modified from [1]

Author	Year	AD	Neuronal Signals	ML	LC	#MLA
Sweeney et al. [116]	2012	✓	EEG	X	✓	0
Khatwani et al. [117]	2013	X	EEG	X	X	0
Barua et al. [118]	2014	✓	EEG	✓	✓	34
Rahman et al. [119]	2015	✓	EEG	✓	X	1
Urigüen et al. [120]	2015	✓	EEG	X	X	0
Tandle et al. [121]	2015	✓	EEG	X	X	0
Islam et al. [122]	2016	X	EEG	✓	✓	12
Jung et al. [123]	2016	✓	EEG	X	X	0
Lai et al. [124]	2018	X	EEG	X	X	0
Manaan et al. [125]	2018	✓	EEG	X	✓	0
Jiang et al. [17]	2019	✓	EEG	X	X	0
Sadiya et al. [126]	2021	X	EEG	✓	✓	13
Fabietti et al. [127]	**2022**	✓	**MEG, EEG, ECoG, LFP, Spikes**	✓	✓	**127**

Artefact Description (AD), Machine Learning methods (ML), Literature Comparison (LC), Number of Machine Learning Articles Compared (#MLA).

4.2 Modules and Description

The online tool available from https://nachodev.shinyapps.io/ABOT/ has been developed using the R language. It is dependent on the packages "Shiny" [128], which facilitates the construction of interactive web apps, "DT" which provides an interface to the JavaScript library DataTables, "shinyjs" for performing common useful JavaScript operations in Shiny, "shinyWidgets" to control the appearance,"shinyalert" for error messages, "readxl" to import the tables, "dplyr" to filter them, "ggplot" and "plotly" to generate the graphs. A user-friendly three-tab layout constitutes the app with a simplistic theme. A functional block diagram is displayed in Figure 4.2. The relationships between the displays and user inputs in the GUI, the functions in the business logic layer, and the files in the back end are related. Upon opening the tool, a pop-up message is displayed, introducing the app and how to operate it.

A screenshot of the main page of the online tool is shown in Figure 4.3. The

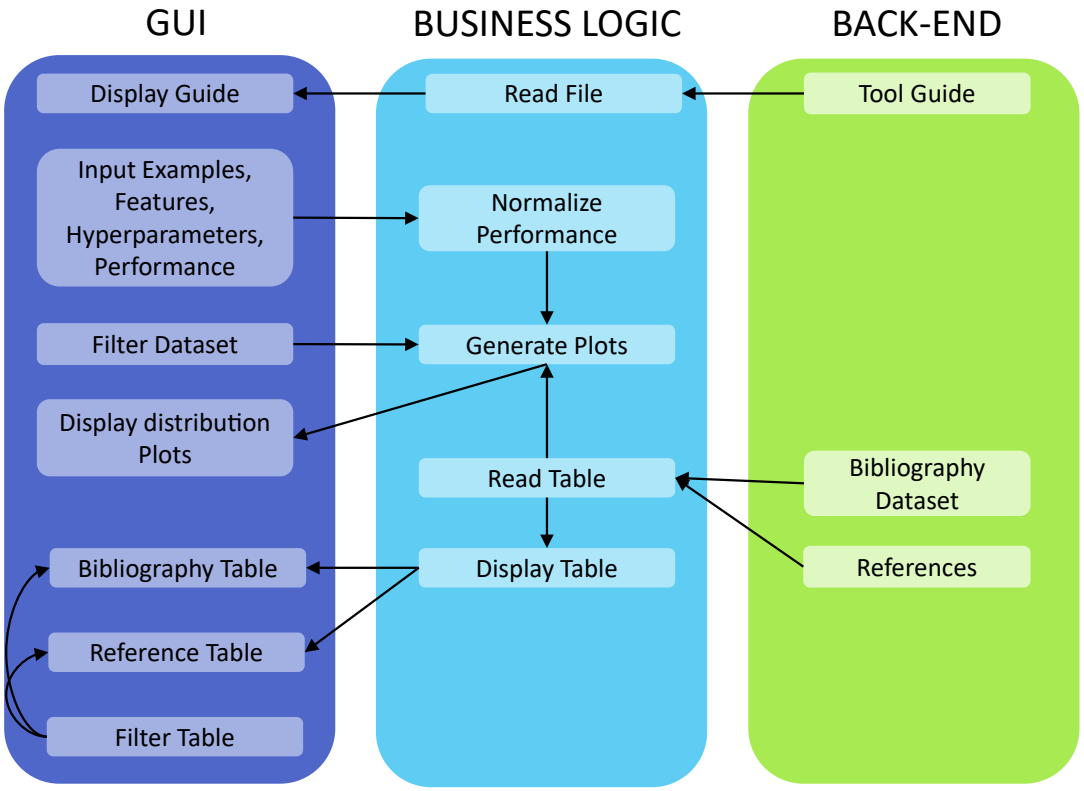

Figure 4.2: Functional block diagram of ABOT, where the relationships between elements of the graphical user interface, the business logic and the back-end are displayed.

user can input the metrics of their approach in the side panel (training examples, features extracted, hyperparameters, select which performance metric they have used and its value) to compare their results to the literature. In addition, they can send suggestions of literature to add or comments about the tool, such as features they wish to see implemented via an online form. The first tab, "Comparison Plots", has four plots, each displaying the violin plot and the scatter points of each metric from the collected dataset. Below them, options are available to filter the plots based on the signal, method or artefact type, year of publication, the number of examples, features extracted, hyperparameters and normalised performance. Multiple filters can be used simultaneously, and if the search yields no result, an alert appears to notify the user. In addition, the app takes the inputs of the side panel and displays the value as a white triangle to differentiate it clearly from

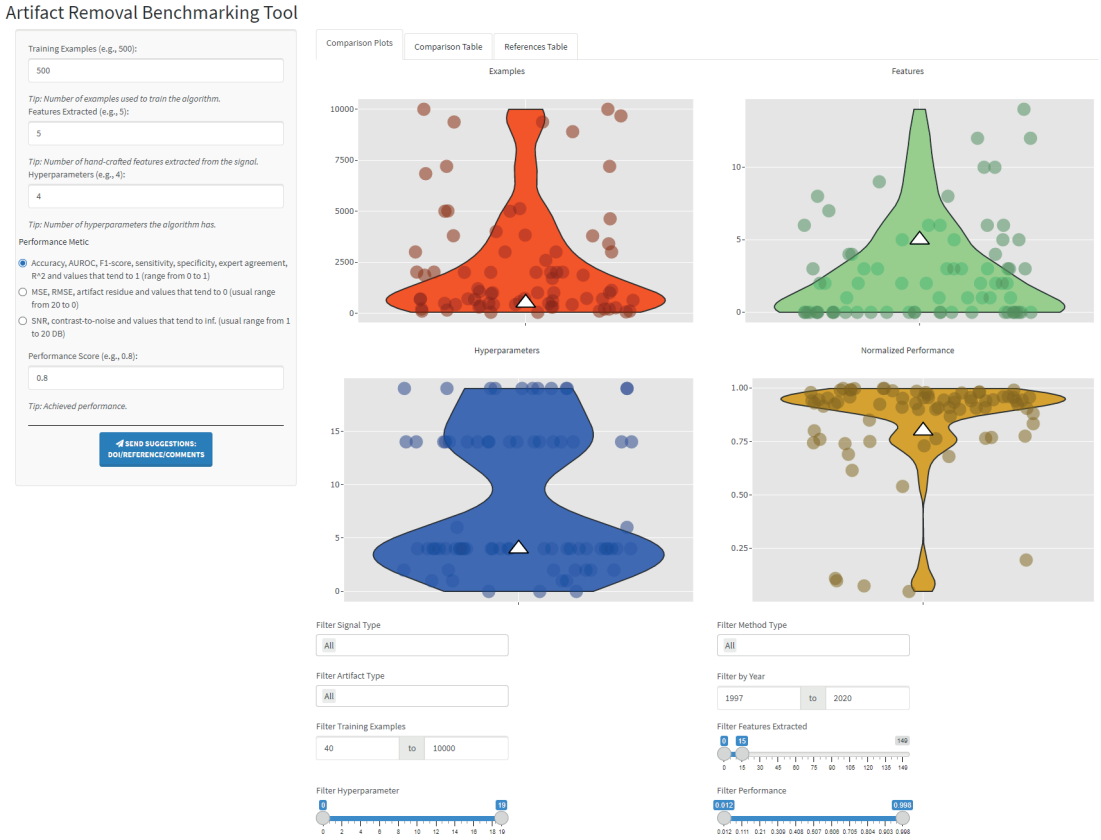

Figure 4.3: Screenshot of the main page of ABOT, where input values and display filters have been applied to showcase the functionality.

the other values. By hovering over a point, the collected information (reference, year, signal, etc.) specific to it is displayed, allowing easier identification for the posterior use of the other tabs.

The second tab, "Comparison Table", contains the table with the methodologies found across the literature and their previously reported details, the original performance metric reported, and the normalisation used. The user can select how many entries are shown, filter through each detail, define intervals in numeric details, or use keywords to search for specific ones. The last tab, "References Table", contains the list of Digital Object Identifiers (DOI) and the complete references, which can be sorted by author or keywords as well. The second and third tabs are depicted in Figure 4.4 A and B, respectively.

The collected data and the tool's code are made publicly available at

A.

Comparison Plots | Comparison Table | References Table

Show 25 entries — Search:

	Reference	Year	Signal	Method	Artifact	Metric	Result	Training Examples	Features	Hyperparameters	Normalized Performance
	All			All	All	All		All		All	All
1	Alagapan et al.	2018	ECoG	kNN	EMG and EOG	artifact residue	0.388	115500	0	1	0.72
2	Anastasiadou et al.	2015	EEG	random forest	EMG	MCC	0.68	360	2	6	0.68
3	Anastasiadou et al. (2)	2017	EEG	random forest	EMG	accuracy	0.8072	3600	17	6	0.807
4	Assi et al	2014	EEG	K means	EOG	classification improvement	0.121		3	2	0.12
5	Aydemir et al.	2012	EEG	kNN	EMG and EOG	accuracy	0.94	100	6	1	0.94
6	Balaiah et al.	2012	EEG	neuro fuzzy filter	ECG,EMG and EOG	SNR	5.0016		0	14	0.833
7	Bang etl al.	2013	EEG	SVM	head movement	error rate	0.0322	64000	16	4	0.968
8	Bartels et al.	2010	EEG	SVM	EMG and EOG	accuracy	0.9	2000	3	4	0.9
9	Burger et al.	2014	EEG	MLP	EOG	RMSE	9.0746	630	0	14	0.099
10	Chadwick et al.	2011	EEG	random forests	EMG and EOG	accuracy	0.54	5012	6	6	0.54
11	Chambayil et al.	2010	EEG	MLP	EOG	correlation	0.90856	303	3	14	0.909
12	Croce et al.	2018	EEG and MEG	CNN	EEG,EOG and sensor malfunction	accuracy	0.956	2012	0	19	0.956
13	de Munk et al.	2013	EEG	Hierarchical Clustering	gradient and pulse artifacts				0	2	
14	Dhindsa	2017	EEG	SVM	EMG and EOG	accuracy	0.933	1050	6	4	0.933
15	Duan F. et al.	2013	MEG	SVM	EMG,EOG and HAM	accuracy	0.979	956	5	4	0.979
16	Duan L. et al.	2017	EEG	SVM	EOG	accuracy	0.7595	7200	0	4	0.76
17	Fabietti et al. (1)	2020	LFP	MLP	noise	accuracy	0.932	1782346	0	14	0.932
18	Fabietti et al. (2)	2020	LFP	RNN	noise	accuracy	0.871	1782346	0	14	0.871
19	Fabietti et al. (3)	2020	LFP	CNN	noise	accuracy	0.951	1782346	0	19	0.951
20	Fairley et al.	2010	EEG	genetic algorithm	noise	sensitivity	0.9844	18268	20	4	0.984
21	Fairley et al.	2010	EEG	genetic algorithm	noise	specificity	0.8093	18268	20	4	0.809
22	Gao et al.	2010	EEG	SVM	EOG	sensitivity	0.987	480	2	4	0.987
23	Gao et al.	2010	EEG	SVM	EOG	specificity	0.979	480	2	4	0.979
24	Gao et al. (2)	2010	EEG	kNN	EOG	accuracy	0.9979	60	2	1	0.998
25	Gao et al. (2)	2010	EEG	MLP	EOG	accuracy	0.9238	2000	2	14	0.924

Showing 1 to 25 of 128 entries — Previous 1 2 3 4 5 6 Next

B.

Comparison Plots | Comparison Table | References Table

Show 25 entries — Search:

	Reference	DOI	Description
	All	All	All
1	Alagapan et al.	10.1088/1741-2552/aaf2ba	S. Alagapan, H. W. Shin, F. Fr ̈ohlich, and H.-t. Wu, "Diffusion geometryapproach to effici
2	Abbaspour et al.	10.4028/www.scientific.net/JBBBE.41.91	Abbaspour, H., Mehrshad, N., Razavi, S. M., & Mesin, L. (2019). Artefacts removal to detec
3	Anastasiadou et al.	10.1016/j.clinph.2017.06.247	M. N. Anastasiadou, M. Christodoulakis, E. S. Papathanasiou, S. S. Papa-costas, and G. D.
4	Anastasiadou et al. (2)	10.1109/EMBC.2015.7318765	M. Anastasiadou, M. Christodoulakis, E. S. Papathanasiou, S. S. Papa-costas, and G. D. M
5	Assi et al	10.1109/EMBC.2014.6945154	E. B. Assi, S. Rihana, and M. Sawan, "Kmeans-ica based automaticmethod for ocular arti
6	Aydemir et al.	https://www.researchgate.net/publication/257922102_Classifying_Various_EMG_and_EOG_Artifacts_in_EEG_Signals	O. Aydemir, S. Pourzare, and T. Kayikcioglu, "Classifying various emgand eog artifacts in
7	Bahador et al.	10.1088/1741-2552/abb5bd	Bahador, N., Erikson, K., Laurila, J., Koskenkari, J., Ala-Kokko, T., & Kortelainen, J. (2020)
8	Bahador et al. (2)	10.1109/EMBC44109.2020.9175711	Bahador, N., Erikson, K., Laurila, J., Koskenkari, J., Ala-Kokko, T., & Kortelainen, J. (2020,
9	Balaiah et al.	10.3844/ajassp.2012.1583.1593	P. Balaiahet al., "Comparative evaluation of adaptive filter and neuro-fuzzy filter in artif
10	Bang etl al.	10.3390/s130506272	J. W. Bang, J.-S. Choi, and K. R. Park, "Noise reduction in brainwavesby using both eeg si
11	Bartels et al.	10.1109/IEMBS.2010.5626481	G. Bartels, L.-C. Shi, and B.-L. Lu, "Automatic artifact removal from eeg-a mixed approac
12	Behera et al.	10.1109/ICAML48257.2019.00016	Behera, S., & Mohanty, M. N. (2019, May). Detection of Ocular Artifacts Using Bagged Tre
13	Burger et al.	10.1016/j.bspc.2014.09.009	C. Burger and D. J. van den Heever, "Removal of eog artefacts by com-bining wavelet ne
14	Cao et al.	10.1109/JBHI.2021.3057891	Cao, J., Chen, L., Hu, D., Dong, F., Jiang, T., Gao, W., & Gao, F. (2021). Unsupervised Eye B
15	Chadwick et al.	10.1109/DEST.2011.5936640	N. A. Chadwick, D. A. McMeekin, and T. Tan, "Classifying eye and headmovement artifact
16	Chambayil et al.	https://www.researchgate.net/publication/45534358_EEG_Eye_Blink_Classification_Using_Neural_Network	B. Chambayil, R. Singla, and R. Jha, "Eeg eye blink classification us-ing neural network,"
17	Croce et al.	10.1109/TBME.2018.2889512	P. Croce, F. Zappasodi, L. Marzetti, A. Merla, V. Pizzella, and A. M.Chiarelli, "Deep convolu
18	de Munk et al.	10.1016/j.neuroimage.2012.09.022	J. C. de Munck, P. J. van Houdt, S. I. Gon ̧calves, E. van Wegen, and P. P.Ossenblok, "Nove
19	Dhindsa	10.1016/j.bspc.2017.06.012	K. Dhindsa, "Filter-bank artifact rejection: High performance real-timesingle-channel ar
20	Dora et al.	10.1016/j.irbm.2020.08.002	Dora, C., Patro, R. N., Rout, S. K., Biswal, P. K., & Biswal, B. (2021). Adaptive SSA Based M
21	Duan F. et al.	10.1109/EMBC.2013.6610929	F. Duan, M. Phothisonothai, M. Kikuchi, Y. Yoshimura, Y. Minabe,K. Watanabe, and K. Aiha
22	Duan L. et al.	10.1016/S1005-8885(17)60215-2	L. Duan, Z. Hongxin, M. S. Khan, and M. Fang, "Recognition of motorimagery tasks for bc
23	Fabietti et al. (1)	10.1109/IJCNN48605.2020.9207320	Fabietti, M., Mahmud, M., Lotfi, A., Avarua, A., Gugganmos, D., Xudo, R., & Chiappalone, M
24	Fabietti et al. (2)	10.1109/AICT50176.2020.9368638	Fabietti, M., Mahmud, M., Lotfi, A., Averna, A., Guggenmos, D., Nudo, R., & Chiappalone, M
25	Fabietti et al. (3)	10.1109/SSCI47803.2020.9308165	Fabietti, M., Mahmud, M., Lotfi, A., Averna, A., Guggenmos, D., Nudo, R., & Chiappalone, M

Showing 1 to 25 of 131 entries — Previous 1 2 3 4 5 6 Next

Figure 4.4: Second and third tab of ABOT, showing (A) the comparison table and (B) references table.

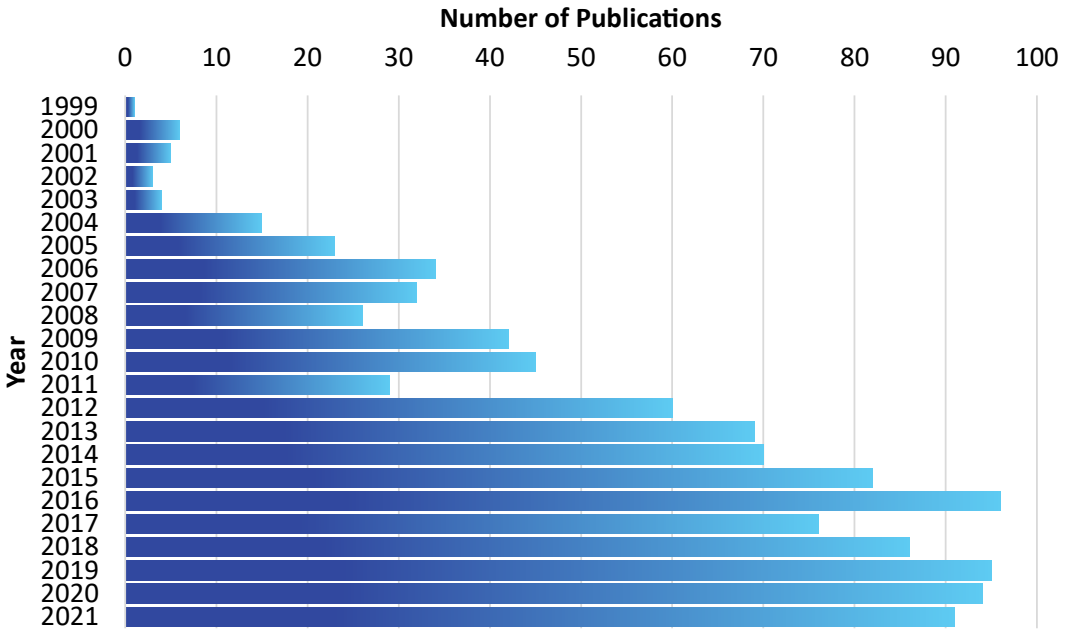

Figure 4.5: Publication trend in artefact removal from neuronal signals. The results were obtained by searching scientific databases with the term "artefact removal".

https://doi.org/10.5281/zenodo.5789913 to foster reproducibility by making it F.A.I.R. (Findable, Accessible, Interoperable and Reusable) [129]. This allows users to find relevant, multiple research models or customise one to their needs. In the next section, the creation of the tool's dataset from articles with ML methods for artefact removal and detection is presented.

4.3 Bibliographic Dataset Creation

To survey the literature, three databases were searched: the ISI Web of Knowledge database of the Clarivate Analytics[1], the IEEEXplore[2] and the Scopus database of the Elsevier[3]. The article title, abstract and author keywords fields of these three databases were searched using search phrases

[1]ISI Web of Knowledge: https://apps.webofknowledge.com/
[2]IEEEXplore: https://ieeexplore.ieee.org/
[3]Scopus: https://scopus.com/

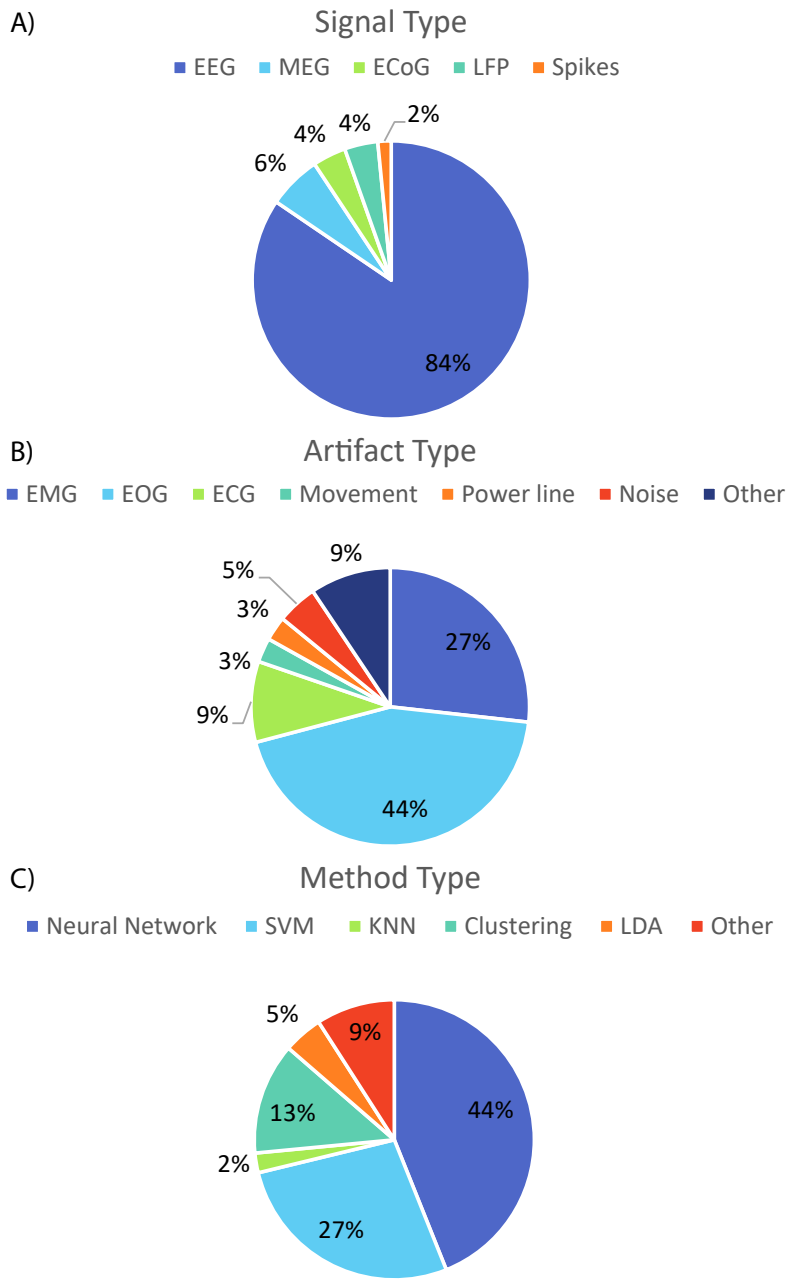

Figure 4.6: Distribution of EEG, MEG, ECoG, LFP, EMG, EOG, ECG, SVM, KNN, LDN and others (A) signal type and (B) artefact type and (C) method type in the extended dataset.

composed of "artefact removal" in conjunction with "EEG", "MEG", "ECoG", "LFP" and "spikes". The obtained results from three databases formed three datasets and were saved in separate comma-separated values (CSV) files. The datasets were compared and pruned by removing duplicate and irrelevant entries returned by the search results. The pruned and combined final dataset contained 1084 publications whose yearly publication frequency has been reported in Figure 4.5.

Out of the shortlisted 1084 papers, manual scrutiny revealed that only 95 of them applied ML-based techniques in artefact detection and removal. To have the most up-to-date review, the aforementioned literature search is complemented with Google Scholar, reaching a total of 127 articles that apply machine learning to artefact detection and removal from neuronal signals.

The distribution of the type of signal, artefact type and method type in the extended dataset are presented in Figure 4.6. For each distribution, articles that dealt with more than one type had each of them counted separately. For example, an approach for EEG that deals with EOG and EMG using a single support vector machine (SVM) model equals values of: 1 EEG, 1 EOG, 1 EMG and 1 SVM.

There is a significant difference in the number of approaches published for EEG (84%), which is followed by MEG at 6% and the invasive recordings between 4 to 2%. This difference can be attributed to the accessibility of non-invasive recording methods, and the number of open-access datasets [130]. Regarding the type of artefact, EOG and EMG represent 71% of all approaches. The former has high amplitude and disturbs mainly the recordings' anterior scalp regions. In contrast, the latter has a large frequency span and the activity of the head, face and neck muscles are conducted through the entire scalp, so detecting and removing them is vital. The remaining 29 % is divided between ECG, power line, noise and others. The "noise" category was assigned to those which didn't address the origin of the artefacts and referred to them as such, while "other" included ones such as electrode pop, ballistocardiogram or electromagnetic interference. The most popular method has been neural networks (NNs) (44%), which are composed of multiple layers of neurons for processing non-linear information and were inspired by how the human brain works. They are known to achieve good performances across domains but require large amounts of information. They are

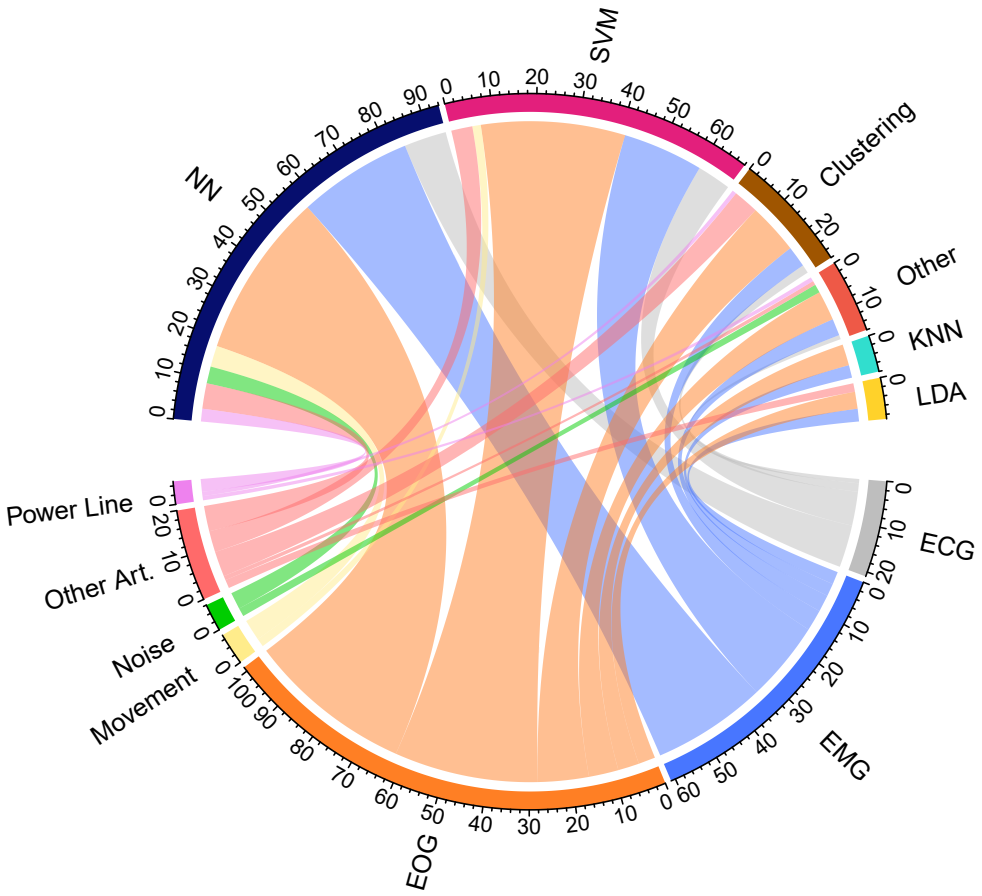

Figure 4.7: Frequency of artefacts in regards to the machine learning methods.

followed by support vector machines (27%) and any form of clustering (13%), while the least applied are linear discriminant analysis (LDA) (5%), k-nearest neighbours (kNN) (2%) and other techniques (9%), which includes approaches such as swarm algorithms or sparse representation.

Figure 4.7 shows the relationship between artefact type and ML method used to identify or remove it. Both EMG and EOG follow SVM, NNs, clustering, and kNN, while ECG, noise, movement, power line and other artefacts are addressed mostly by NNs instead of SVM.

Table 4.2: Examples of hyperparameters of machine learning methods.

Method	Hyperparameters	
Neural network	Learning rate	Activation function
	Momentum	Regularisation
	Weight decay	Dropout
	Epochs	Weight initiation
	Batch size	Loss function
	Number of layers	Output function
	Neurons per layer	Number of classes [131]
Convolutional neural network (Spatial Feature Learning)	Learning rate	Activation function
	Momentum	Regularisation
	Weight decay	Dropout
	Epochs	Weight initiation
	Batch size	Loss function
	Number of layers	Output function
	Neurons per layer	Number of classes
	Patch size	Number of filters
	Convolutional layers	Pooling layers
	Fully connected layers	Filter size [132]
Support vector machine	Kernel	Gamma
	Cost	Degree [133]
k-nearest neighbour	Distance function	k
k-means clustering	Distance function	N clusters
Linear discriminant analysis	Solver [134]	

Our survey results show that there is no standard performance metric used in research papers to consider the effectiveness of artefact detection removal algorithms, constituting a challenge to compare the different results obtained by authors. To address this issue, Valipour et al. [135] have compiled the different metrics frequently used in research papers to consider the effectiveness of EOG removal algorithms. In addition, Islam et al. [122] stated the necessity for using a unique standard evaluation method composed of quantitative and qualitative

metrics.

While developing a metric or scale suitable for all the different applications may not be feasible, to compare them, observing four key characteristics of ML models are proposed: the amount of training examples, the amount of extracted features, the model's hyperparameters and the model's performance. Machine learning models require information to make predictions accurately. However, the amount varies depending on each one [136], neural networks in particular are known as data-hungry algorithms due to the amount they need to be trained [137]. Out of two models of equal or similar performance, the one which needs fewer data should be favoured. The second characteristic is the number of handcrafted features extracted. Extracting handcrafted features require expertise and directly correlated to the amount of information fed to a model. Thus, between two models of comparable performance, the one that requires less information should be preferred. The third characteristic is the algorithm's number of hyperparameters, where examples of ML algorithms and their hyperparameters are listed in Table 4.2. This was chosen to reward algorithms that are less complex to train; for example, a feed-forward NN has fourteen hyperparameters, whereas the kNN has two. The last characteristic is a normalised performance score, as described in the following expression:

$$\text{normalised performance} = \begin{cases} \text{metric}, & \text{if best performance of metric} = 1 \\ \frac{1}{1+metric}, & \text{if best performance of metric} = 0 \\ \left(1 - \frac{1}{1+metric}\right), & \text{if best performance of metric} = \infty \end{cases} \tag{4.1}$$

Here, the first case applies to metrics such as accuracy, the area under the receiver operating characteristic (ROC), F1-score, sensitivity, specificity, expert agreement, R2; the second one to mean squared error (MSE), root mean squared error (RMSE), artefact residue; and the third one to signal-to-noise ratio, contrast-to-noise ratio and others. Due to the lack of a standard metric for evaluating artefact detection and removal, Equation 4.1 was devised to aid the visual selection of methods, as the normalised performance scales all metrics between 0 and 1. This information is meant to be utilised in conjunction with

the rest of the information provided in the tool via hoovering or via the description table, not as an absolute comparison criterion on the method's performance. This proposed scale is the first step to closing a major gap in the field, and its limitations are discussed in Section 4.4.

These four characteristics were chosen because they are the most consistently reported elements that can be used to compare the algorithms. In contrast, time complexity, memory complexity, parallelizability, portability and interpretability were inconsistently reported. The selected characteristics were extracted from all the collected articles, and their distributions are displayed in Figure 4.8. Overall, the majority have used less than 65,000 training examples, which is expected for machine learning models. Regarding the features extracted, most of them are under 20, with the presence of some outliers. The distribution is right-skewed as neural networks don't depend on them to achieve good performance and are the most utilised technique (see Figure 4.6). The hyperparameters take discrete values, generating a gap in the distribution where there are no methods with that particular amount. The upper half is the neural network approaches, which present 44% of the total approaches, and the lower half is the remaining methods. Lastly, the normalised performance concentrates above 0.7 since research is sought to be published after achieving a successful performance level. Values lower than that are due to the normalisation function, where removal techniques that haven't achieved low MSE are significantly penalised.

4.4 Discussion

The toolbox allows filtering the data to find approaches that match the application of interest and compare them. However, if the user does not filter the data, is it a valid comparison between different types of signals, types of subjects and types of artefacts? For example, a method for removing muscle artefacts from human EEG recordings may not be very useful when developing or searching for methods for removing stimulation artefacts in a rodent's patch-clamp data. The difference between the different neuronal signals across subjects is significant enough to expect deviations when adapting from one to another. In addition, preprocessing such as filtering and feature extraction may

also need to be adapted to obtain a working model. Regarding adopting a model of one type of artefact to another, some authors have applied the same model to different artefacts and achieved similar performances ([138, 139, 140]).

A limitation of the tool is that while classification and regression have defined metrics we can compare via the normalised performance, clustering approaches do not. The results of clustering artefacts have been presented without a metric [141], as signal-to-noise ratio [142] or contrast-to-noise ratio [36] of removed independent components, or as the improvement of the signal's classification [143]. This adds a layer of complexity in the comparison of unsupervised methods with supervised ones, adding a layer of complexity to the benchmarking. Furthermore, the approaches that address multiple types with a single model do so at a performance similar to those with the same model that only deals with one type. Overall, the comparison outside the application must be made diligently, knowing that it does not mean that the performances will be maintained. Still, it can help orient those looking for which method to apply by discarding those that don't perform well. The normalised scale may also miss-represent results; for example, a high classification accuracy will be mapped to a value near 1; however, one must achieve an extremely low RMSE to achieve the same results. Nonetheless, it provides an approximation of the performance of the approach, which can be evaluated further on the table with the same metrics if desired. To the best of our knowledge, no other attempts to solve this issue have been made before so that it can be used as a starting point.

Out of the many challenges the field currently poses, replicability is the main one. Most studies have used private datasets and outside the few hosted in Physionet [144] or BCI competitions [145], the data has been removed from their respective websites. A limitation of this work is that despite the key characteristics of the approaches have been listed, details such as preprocessing steps or the layers of the neural networks are not listed. That information can be behind paywalls, leading to the inability to reproduce and compare results among studies. A second shortcoming is that the listed characteristics may not be sufficient for some researchers to decide; for example, the processing time is crucial information for selecting algorithms when looking for online implementations. However, the selected characteristics were mainly present

throughout the surveyed literature, whereas others, such as the normalisation procedures, hyperparameter values, and hardware utilised for training and computational time, were very inconsistent. Lastly, the toolbox focuses on machine learning solutions, excluding the wide range of artefact removal methods listed in Section 2.4. While this limits the tool's utility for those looking for the "best approach across the board", it is hoped that it will be useful for those looking at data-driven solutions or those with academic purposes such as method development or comparisons of machine learning methods.

Moving forward, automatic removal has a significant role to play in neuroscience. Craik et al. [23] review of deep learning for electroencephalogram classification tasks indicates that 63% of the studies did not methodically remove the artefact. Moreover, 29% removed them manually, and only 8% of studies used automatic methods, which mainly relied on ICA-based algorithms. In addition, Roy et al. [24] deep learning-based electroencephalography analysis review showcases in their survey that 47% did not remove artefacts, 30% did not mention if it was applied at all and only 23% removed artefacts. Thus, there is an excellent opportunity to apply the methods listed in the tool in classification and other tasks. As previously indicated in Figure 4.5, there is a consistent growth in the number of articles published every year that mention artefact removal. Filtering them, extracting the key characteristics and incorporating them into the tool takes time, another limitation of the proposed work. However, it is hoped that users will help us with its improvement via the suggestion email option by drawing our attention to where updates are needed. Hopefully, this will mean that the tool will remain valuable and necessary.

As online processing is taking more relevance, computationally and energy-efficient methods are desired. The trend shows that machine learning will most likely be the future direction in the field, given that those approaches suit the requirements mentioned above. This means that the next step focuses on developing more interpretable models, especially those that include neural networks, providing insight into how variables interact and how the model operates. Parallel to them, explainable techniques such as Grad-CAM[46], Shapley values [47] and Local Interpretable Model-Agnostic Explanations

(LIME) [146] should also be developed for black-box models. By understanding the models and their outputs, the users' confidence in them will improve, and their feedback can help improve the models' performance. There is a big opportunity to explore this in the field, as across the literature we find few examples of explainability: for ECoG, we find a binarised artefacts probability matrix[147], for EEG, Grad-CAM was applied to the signal [148] and a topographic map of the scalp[149], and for MEG we find Grad-CAM applications in the signal[150] and scalp topography [151]. In addition, models should allow interaction, such as choosing which artefact to detect, turning it off when it is not required, or allowing modification of the classification probability threshold. There is no wide range of techniques that excel for all possible artefacts and conditions; therefore, approaches should improve the robustness across multiple subjects and different biological contexts [152]. The use of several processing stages in which each phase eliminates each artefact to increase the signal's quality by using a series of algorithms remains a possibility [120].

4.5 Conclusion

To analyse brain signals without interference from artefacts, researchers have proposed different means to detect and remove them. Because of the extensive literature on the topic and the wide range of signals, artefacts, and ML techniques, an online tool that facilitates browsing through the literature has been developed. The user can compare the performance of approaches for benchmarking or for implementation via the graphs and tables available in the tool. The literature has been successfully surveyed and extracted key characteristics of the different machine learning methods for the tool to showcase. Hopefully, the community will adopt the tool; for that purpose, it has been made open-access and made its code available and allowed users to send suggestions via the tool. As future work, a survey on user acceptance should be conducted to see how the consensus of the community has evolved (further detailed in Section 7.5). By facilitating the benchmarking of new methods, as the state of the art of artefact detection and removal techniques improve over time, so will the results of brain studies and

BCI applications.

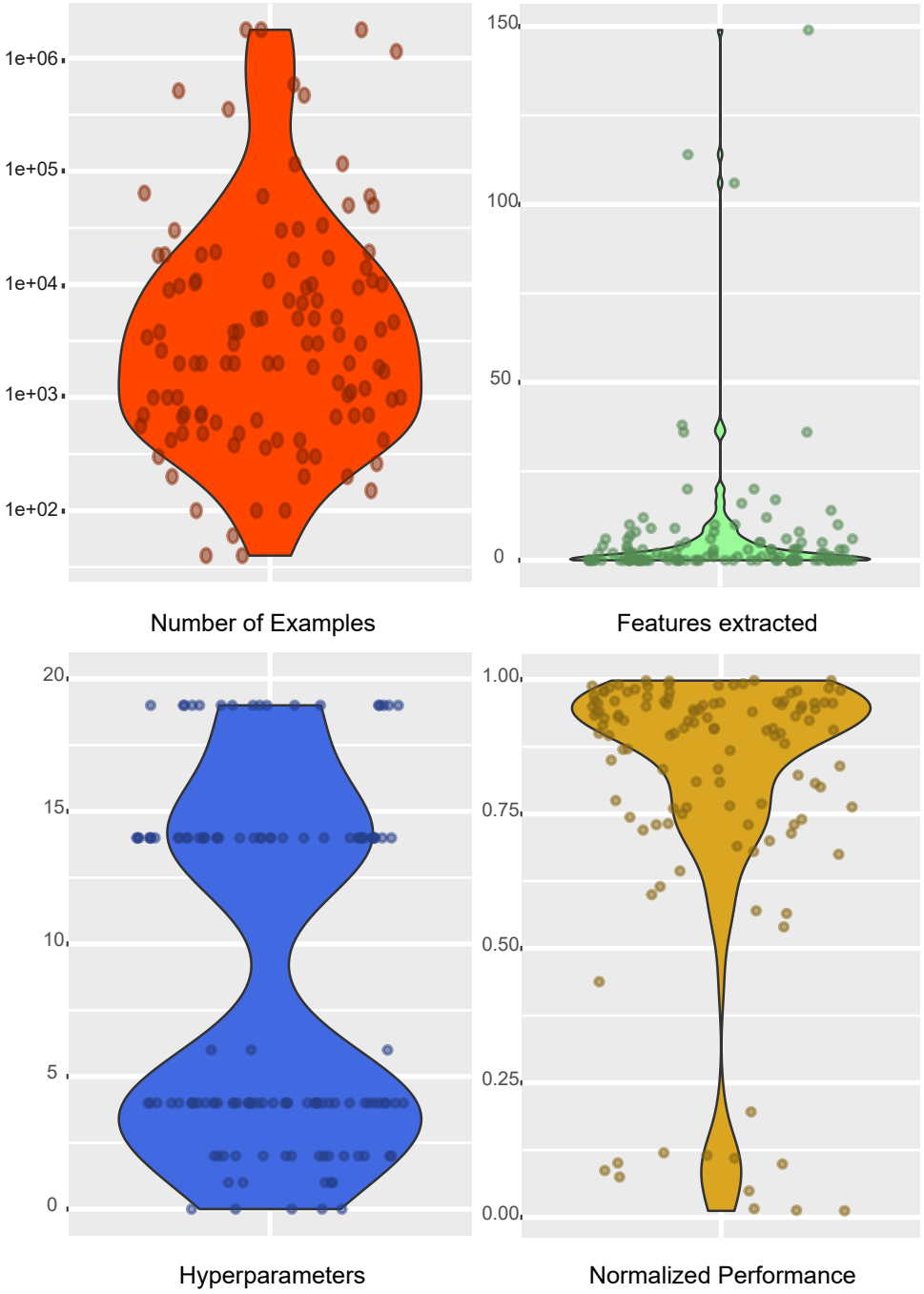

Figure 4.8: Distribution of the four characteristics from all the collected articles. Each dot represents a surveyed article, area of the graph represents the probability that a dot will take the given value, with wider sections along the x-axis representing a higher probability.

Chapter 5

Artefact Detection and Removal

5.1 Introduction

Neural recordings give insight into the brain's structures and functions. The recording systems aim to capture the electrical activity of the biological structures; however, these are not isolated systems and activities from other sources are also recorded. Besides, faulty equipment handling, electrical stimulation or electrode movements can distort the recordings. As part of the recording process, the recordings must be reviewed to identify corrupted segments and address them, as they are detrimental to any posterior analysis. This includes artefact removal (e.g. filtering, template subtraction or advanced computational techniques) or discarding the segment.

Each neural recording session produces a huge volume of data, especially if it is obtained over a long period of time and the experiment requires repetition. The amount of data gets multiplied by the number of recording sites. The post-experimental reviewing process consisting of annotating long recordings for evoked responses or unusual activities, which may happen in a much smaller time scale (e.g., 0.1 seconds in an hour), is a tedious task. By automating this task, the researcher can focus on the interpretation task for diagnosis or an application. Employing ML algorithms, which can learn from patterns to predict unseen data, have been successful in the literature. However, a computational background is required to apply them successfully as there are

intricacies, such as defining hyper-parameters.

While there is a wide range of toolboxes for neuronal signal analysis, SigMate [83], is one of the few that deals with LFP artefact detection and removal. It has been expanded with new functionalities to offer a more competitive toolbox, named SANTIA (SigMate Advanced: a Neuronal Tool for Identification of abstracts). These include state-of-the-art modules for artefact detection and removal, the analysis of any number of channels, regular updates and a supported environment. The SANTIA toolbox is a friendly user interface that aids the offline identification of artefacts process by simplifying the steps to train powerful computational algorithms with the minimum user input. The toolbox only includes NN-based approaches, not only as they are state-of-the-art in the topic (see Section 4.3), but due to their flexibility and computational speed.

The recording of neuronal data, especially when using multi-electrode arrays, can lead to electronic files of notable size. Figure 5.1 illustrates a conducted survey of the formats of invasive neural recordings in open datasets [153]. The data shows that '.mat' is the preferred extension for storage by a substantial margin. This emphasises the necessity to develop tools which address the datasets available in '.mat' format. Therefore, SANTIA was implemented in Matlab and worked with single files containing multi-channel data files in a variety of formats. The toolbox only depends on the Deep Learning Toolbox and the basic version of Matlab 2020a and above, therefore can function in any operating system. SANTIA has been developed with the latest app development environment of Matlab, which allows it to be supported for longer and be improved with new modules, such as GUI improvements which are planned for the next update.

The remainder of the paper is composed of 5 sections: Section 5.2 describes the toolbox, Section 5.3 describes the artefact detection method and results, followed by the removal method and results in Section 5.4. Lastly, in Sections 5.5 and 5.6 discussion and conclusion are presented, respectively.

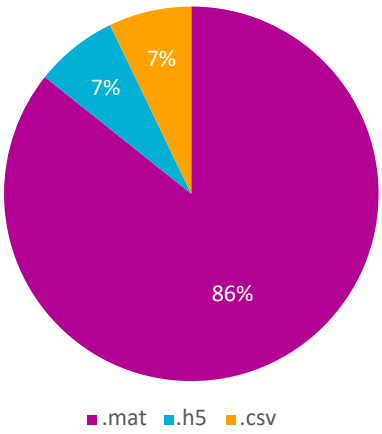

Figure 5.1: Distribution of formats of local field potential signals in open datasets.

5.2 SANTIA Toolbox

For wider adoption by the community, the toolbox is freely available online via the Github repository at:

https://github.com/IgnacioFabietti/SANTIAtoolbox

Having launched the toolbox, the GUI provides easy access to all modules. It's important to note that SANTIA is a generic environment built on a single interface with individual features implemented rather than a library of functions with a GUI added to make access easier.

The app is divided into four modules, each performing a different process and analysing neuronal signal files. The app consists of four modules which perform various processes and analyses on neuronal signal files. The first one's major functionalities are as follows: data loading, scaling, reshaping, channel selection, labelling, saving and 2D display. The second module is composed of: data loading and splitting, hyper-parameter setting, network load or design, network train, test set classification, threshold setting and saving. The third one comprehends data and network loading, classification, 2D data display and saving. Lastly, the fourth module is composed of: data loading, normal segments extraction, hyper-parameter setting, network train, test set visualisation, replace segments, plot replaced channels and saving.

The user communicates with the GUI by selecting functions, settings, and

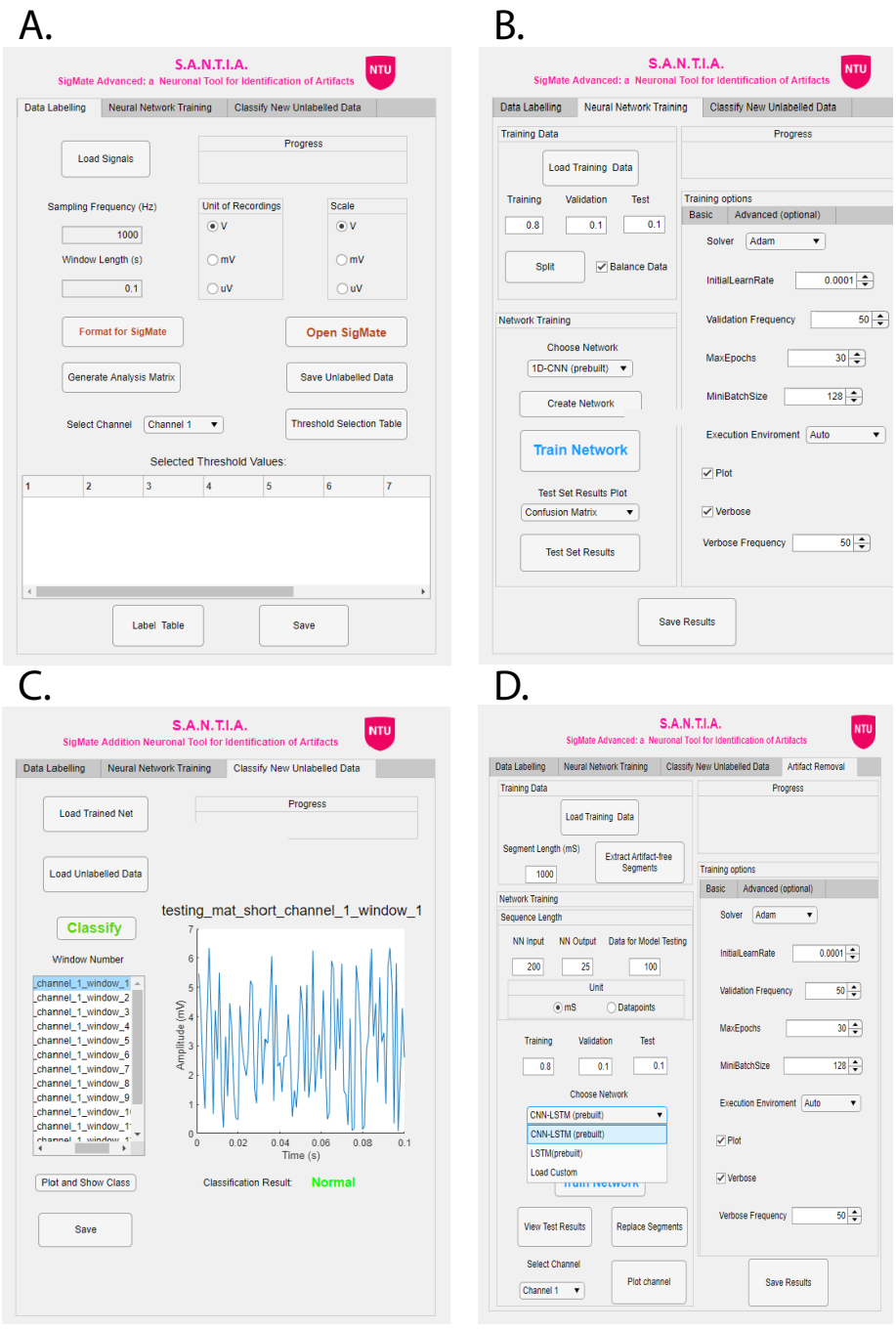

Figure 5.2: Screenshots of the SANTIA toolbox graphical user interface: Data Labelling (A), Neural Network Training (B), Classify New Unlabelled Data (C) and Artefact Removal (D).

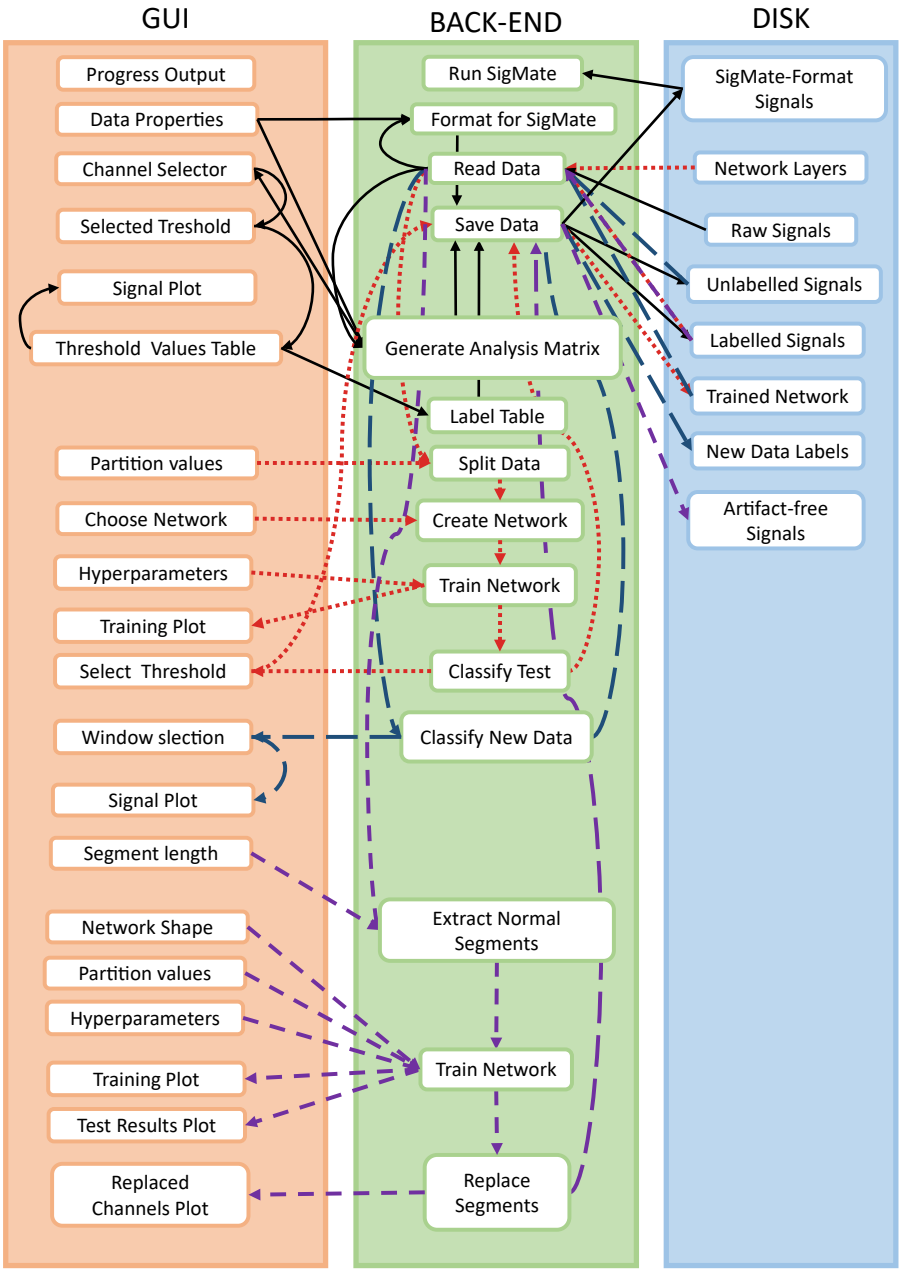

Figure 5.3: Functional block diagram of SANTIA. Arrows in black correspond to the "Data Labeling" module , in red to the "Neural Network Training" module, in dark blue to the "Classify New Unlabelled Data" module and in purple "Artefact Removal" module.

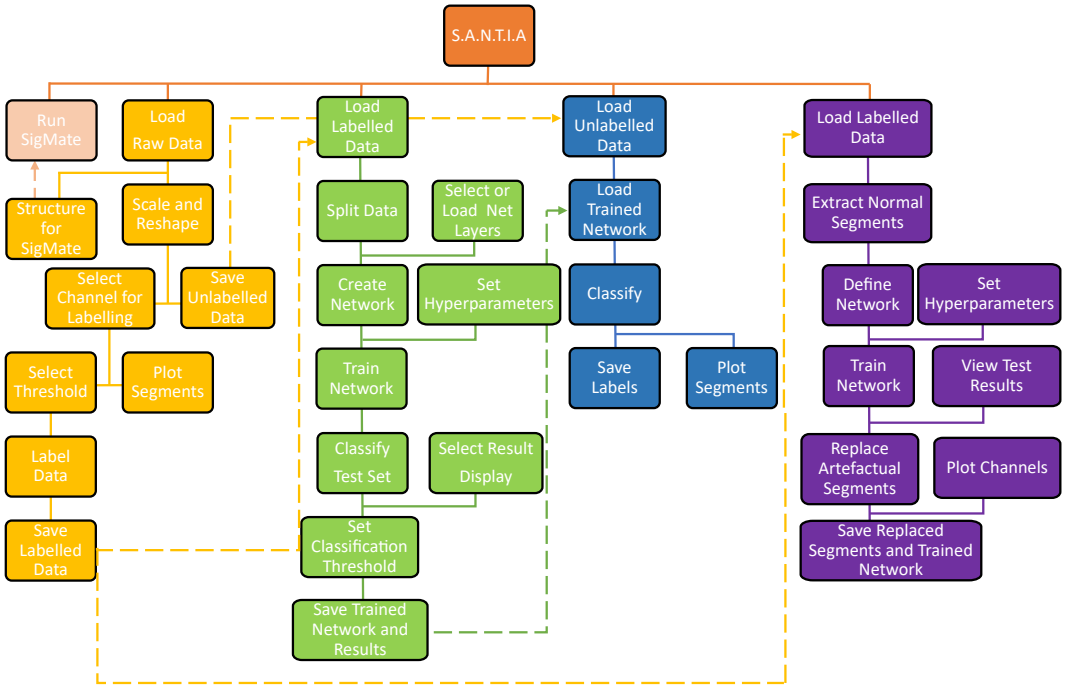

Figure 5.4: Workflow of SANTIA, where the "Data Labeling" modules are coloured yellow, the "Neural Network Training" modules in green, "Classify New Unlabelled Data" modules in blue and "Artefact Removal" modules in purple.

keyboard inputs handled in the back-end. Before running any function, a check routine runs to ensure that the user has not missed a process or failed to include all of the required inputs or parameter selections. This reduces the likelihood of human error and the amount of time spent. When the cursor has hovered over a component of the GUI, tooltips with a short clarification appear in case the user has doubts.

The back-end is where the functions for displaying alert signals, generating figures, and computing labelling, training, or classification are allocated. The respective outputs can be exported to a '.mat' file at the end of each module, which can be used in other applications thanks to the format's accessibility.

The following sections describe the individual modules in greater detail. As a visual aid, Figure 5.2 shows the screenshots of the software package, Figure 5.3 illustrates the function block diagram, and finally, Figure 5.4 shows the workflow

diagram.

5.2.1 Labelling Module

The first step is loading the neural recordings, which is done with the import wizard launched by the 'Load Signals' button of the first unit, as a matrix with m number of channels and n number of data points for each channel. ASCII-based text(e.g., .txt, .dat, .out, .csv), spreadsheet files (e.g., .xls, .xlsx, .xlsm) , and Matlab files (e.g., .set, .mat) are the formats that are compatible with the toolbox. In order to structure the data, the user must provide the sampling frequency in Hz and the window duration in seconds. The options for data scaling are available to avoid the common incorrect magnitude annotations.

Once all of these parameters have been filled, the 'Generate Analysis Matrix' will structure the data for posterior analysis. This means that given a window length w, and sampling frequency f, the $m \times n$ matrix becomes a new $p \times q$ one, where $p = \frac{m \times n}{w \times f}$ and $q = w \times f$. This is incorporated into a table with row names that follow the format 'file_id+_channel_+i+_window_j' where file_id is the name of the LFP data file, i the number of channels where $i = 1, ..., m$ and j the corresponding window. In addition, its columns are named: first "window_power" followed by the values of the signal t_k where $k = 1, ..., q$. As this process involves creating p amount of row names and window's power, a memory check is done to read available memory and alert if the usage of more than 80% of the available memory would be needed.

The option to save this data for posterior classification is presented as 'Save Unlabelled Data'. The following step consists of labelling the data, carried out by giving segments whose power exceeds a user-defined threshold a binary label (0 for normal and 1 for artefact). The power of each segment is estimated as the sum of the absolute squares of the values of the signal divided by the segment length. The toolbox allows for three options, either of which the user can use to their preference. The first is a table layout that holds the segment power in the first column and the values of the signal in the subsequent columns. The user may sort any column to define a value which optimally divides both classes and visualise any segment they select. The second option is the 'histogram threshold',

where a histogram of the segments' power shows the distribution, and the user can select with a slider the cut-off value or visualise a segment. Alternatively, the threshold values can be typed into the table displayed on the module. While all three options have their pros and cons, users are recommended to utilise the 'histogram threshold', where the outliers in the power distributions can be more easily identified.

Once all channels have been filled, the signals are labelled and saved as a standardised struct, which includes the original filename, the structured data with its labels, the sampling frequency, window length, the scale, and the threshold values. The purpose of the format is to allow users to select and contrast the various datasets they build due to different window lengths or threshold values they may have chosen. Users can see when each stage has been finished with the help of text in the 'Progress' banner, which is duplicated across each unit.

The user can also structure the data for SigMate analysis. The toolbox expects a $datapoints(n) \times channels(m)$ format, with the first column as timestamp and each of the channel's signals in the following columns. In addition, as it only handles 5 channels at a time, $m/5$ files have to be generated. Thus, SANTIA transposes the input matrix, generates the timestamp based on the declared sampling frequency and generates the files.

5.2.2 Neural Network Training

The second module starts with loading structured data from the previous module. The user is asked to set the training, validation and test splitting values. This is common practice to avoid over and under-fitting results. As artefacts are rare events, the datasets usually present a strong imbalance which can cause bias in training, a tick box for balancing the data via down-sampling to match the number of examples per class is present next to the 'Split' button. Clicking it generates three datasets with non-repetitive randomised elements from the original matrix.

This is followed by choosing the network, where the options are MLP, LSTM, one-dimensional CNN (1D-CNN), or for the user to load their custom set of layers. This is done by choosing a Matlab file which has a Layer-type

variable, i.e. layers that define the architecture of NNs for deep learning, without the pre-trained weights. These can be modified via console or the Deep Network Designer Toolbox, for more information, the reader is directed to the Mathworks website [1]. While employing different architectures might yield better results, it is also possible that they might not be structured properly and lead to underfitting, overfitting or failure to learn at all. Therefore a limitation of employing custom networks is the time consumption that takes getting the correct combination of layers, as well as setting parameters such as filter size or activation function. Optionally, the user can customise the training hyper-parameters such as the solver, initial learning rate, execution environment, among others. These intentionally mirror the ones included in the Deep Network Designer to facilitate its usage to those familiarised with it. These are removed for the MLP option, as it uses a different toolbox (i.e. PatternNet of Deep Learning Toolbox [154]), thus doesn't allow the same configurations. Clicking the 'Create Network" button loads the training options and sets the input layer to match the window size.

The 'Train Network' button runs the train network function, inheriting the previously defined training options and network. For the 1D-CNN, as the deep learning toolbox is intended for images, the 2D matrices are resized to a 4D vector: 1 × window length × 1 × number of windows, originally intended to be: width × height × channels × number of examples. A display of the training process automatically appears unless the user decides not to, which enables monitoring the process and early stopping.

Having completed the training, the user can select whether the 'Classify Test Set' displays the confusion matrix, the AUROC curve or opens up a new window where the accuracy, F1 score and confusion matrix appears along with the possibility to modify the classification threshold (set at 0.5 by default). Finally, 'Save Results' creates a struct with data's filename, the trained network, the training information, the test set's classification threshold, AUROC, accuracy, F1 score and confusion matrix.

[1] Mathworks: https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.layer.html

5.2.3 Classify New Unlabelled Data

This module begins with loading a trained net along with its classification threshold and unlabelled structured data. After its classification, the options to plot each of the windows with the corresponding colour-coded label appear. Finally, users can save the labels as a table with the corresponding window name. Having described the toolbox's methods, components and its functions, a test case with real recorded LFP is subsequently presented.

5.2.4 Artefact Removal

The initial step of this unit is to load the structured file generated by the first module. Once complete, the user must input the duration of artefact-free segments they wish to extract from the file to train the model. A progress bar indicates the extraction progress, followed by a notification of the number of segments extracted upon completion. The following step is the configuration of the input and output of the model, with the option of selecting either data points or milliseconds as units. They must also input how to split the data for training, validation, and test sets, as they are crucial to avoid over-fitting.

For the third step, users can make use of the CNN-LSTM architecture, the LSTM or for the user to load his/her custom set of layers. A side panel allows the customisation of training hyper-parameters such as the validation frequency, max epochs, verbose, mini-batch size, and others. The training process is run by clicking on the 'Train Network' button, which loads all the user-defined inputs so far and generates a training plot for the user to evaluate the process and do an early stopping if required.

A pop-up notification alerts the user of the root mean square error of the test set, and the user can visualise the examples of the test set in contrast to their forecast. The user can either adjust the network and training parameters to get a desirable result or proceed to the last step. This consists of swapping the windows labelled as artefacts for the network's forecast, where a progress bar is displayed to show the advancement. The newly obtained signals can be visualised by first selecting which channel to display and the 'Plot Channel' Button. The last step is to save all the obtained information in the form of a struct with the data's

filename, the trained network, the training information, the test set's RMSE, the test set original, and replaced segments and the data with the artefactual data removed, where the user sets the file name and directory to store it.

5.2.5 Outputs

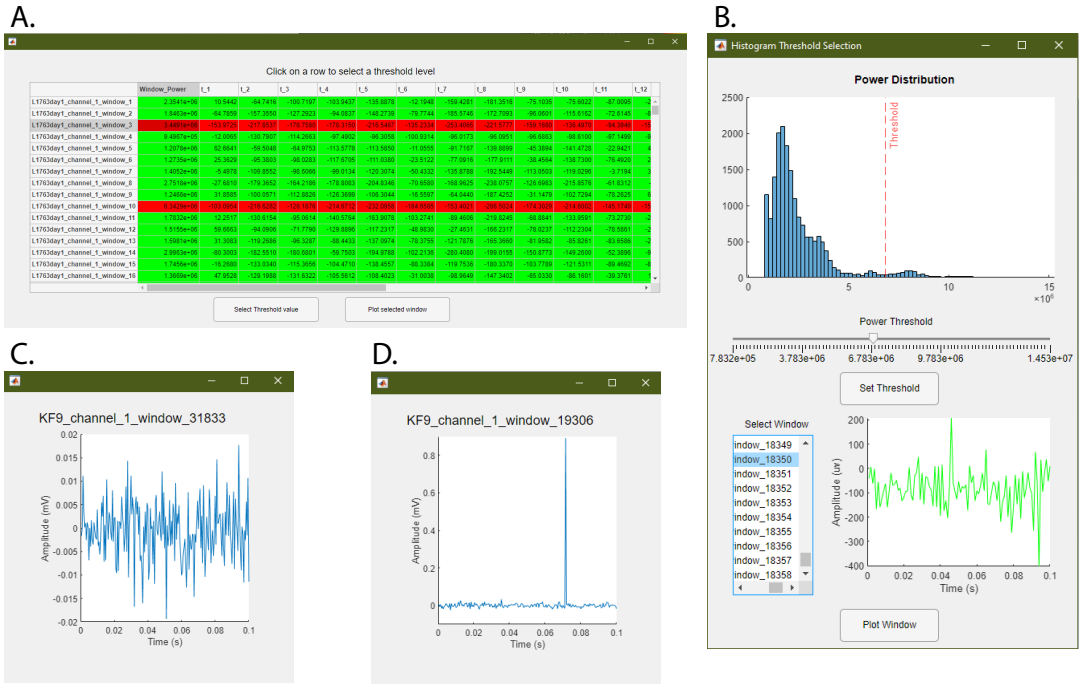

Figure 5.5: Screenshots of the SANTIA's threshold selection outputs: threshold selection table (A), threshold histogram selection slider (B), a window of a non-artifactual signal (C) and a window of an artifactual signal (D).

Figures 5.5 to 5.9 show output windows of the toolbox generated during its operation. Figure 5.5 displays output windows generated after the data file is loaded. They include the selection of threshold via the table as shown in Figure 5.5 (A), where green lines show windows representing data above the threshold and red lines show below it, the selection of the threshold via histogram in Figure 5.5 (B), and two representative figures of normal (in Figure 5.5 (C)) and artifactual windows (in Figure 5.5 (D)). Figure 5.6 shows the output windows for the NN training process, which currently support MLP, LSTM, LSTM-CNN

Figure 5.6: Screenshots of SANTIA's network training outputs: multi-layer perceptron training process (A) and one-dimensional convolutional neural network training process (B).

and 1D-CNN. As the networks come from different Matlab toolboxes, their individual configurations require separate processes, which are represented in Figures 5.6 (A) and 5.6 (B) for MLP and 1D-CNN/LSTM/LSTM-CNN, respectively.

After completing the training for artefact detection, the different plots of the test set classification results of the second dataset for the 50 ms window are shown

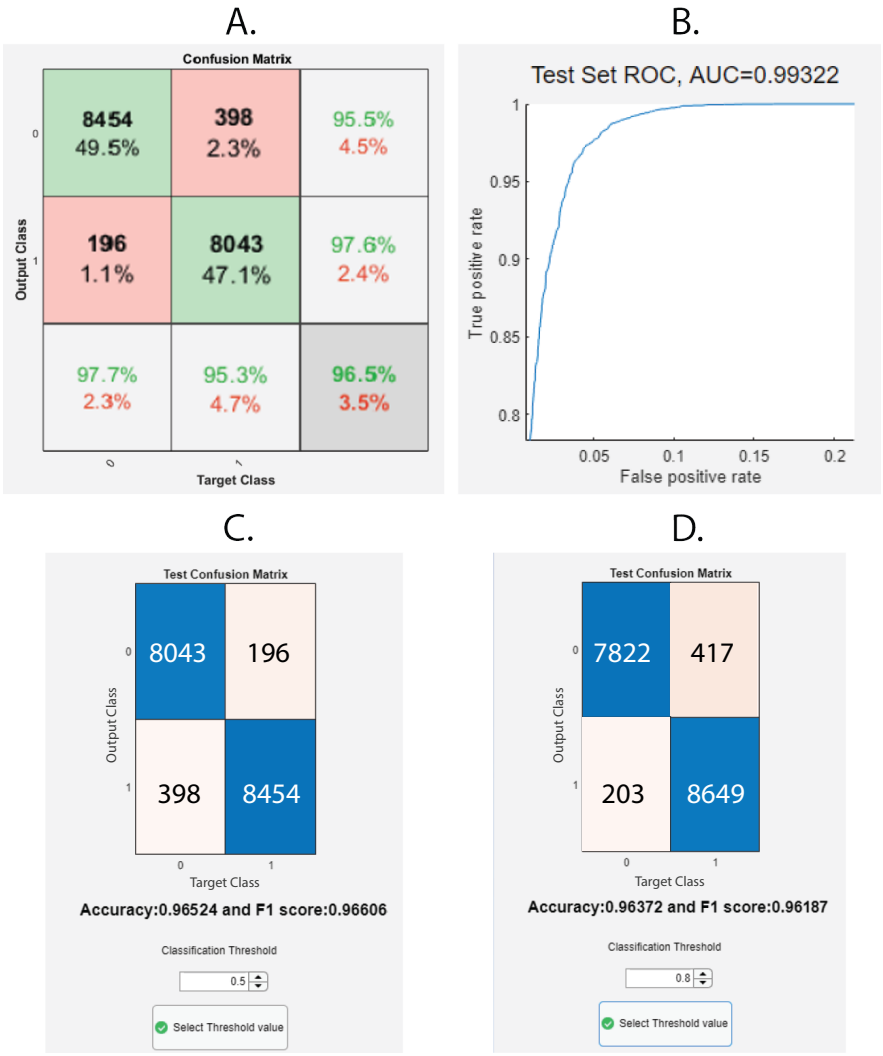

Figure 5.7: Screenshots of the SANTIA's network test set results outputs: confusion matrix (A), AUROC curve (B), threshold selection window with default (C) and custom values (D). The rows in (A) refer to the class predicted by the modcel (Output Class), and the columns refer to the true class (Target Class). Correctly classified samples are located in the diagonal cells (green colored), and incorrectly classified ones in the off-diagonal cells (red colored). In each cell it indicates both the amount of examples (in bold) and relative percentage to the overall number of samples. The precision and false discovery rate are displayed in the column on the far right, while the sensitivity and specificity are displayed at the bottom row. Lastly, the total accuracy is indicated at the bottom right cell.

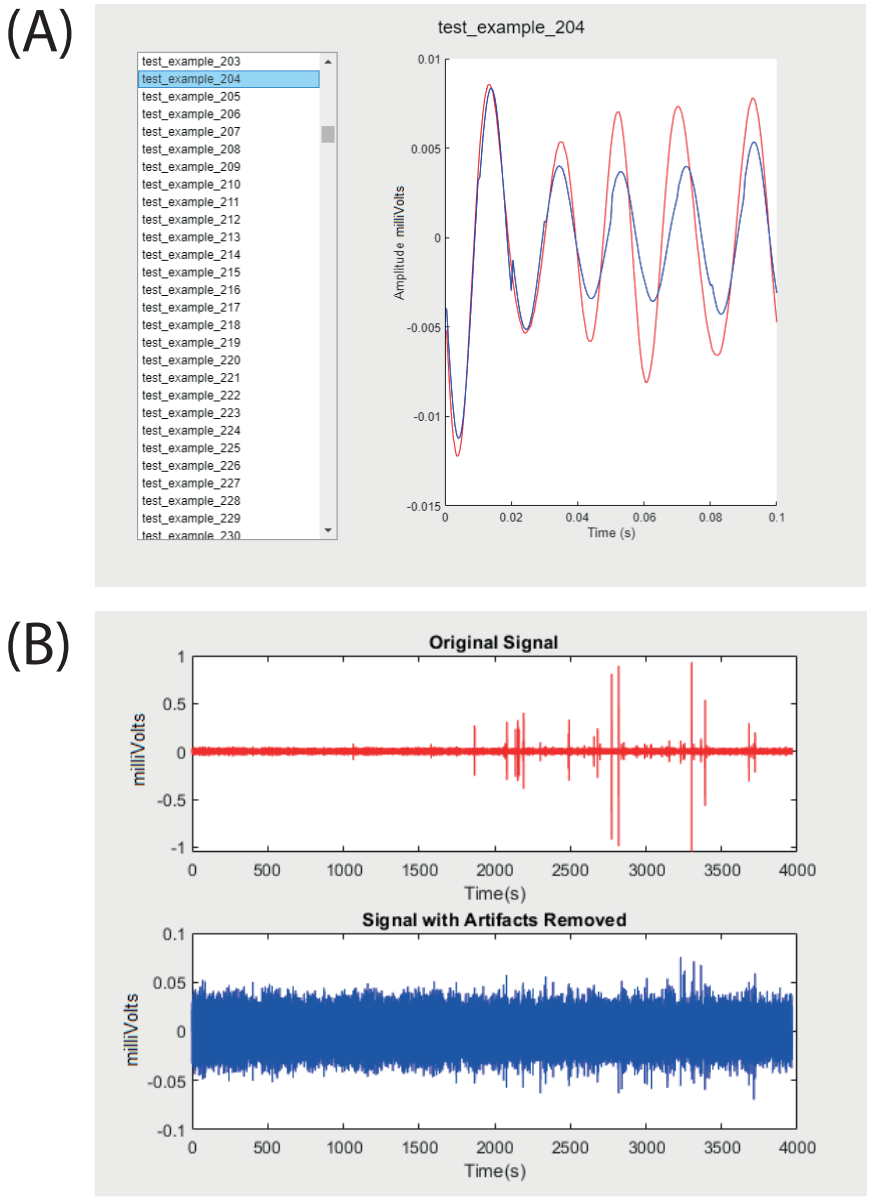

Figure 5.8: Visualisation of the test set (A) and comparison of the original signal with artefacts removed (B) are the outputs of the artefact removal module. The original signal appears in red in both outputs, while the predicted or artefact-free signal appears in blue.

in Figure 5.7. As a part of allowing the user to evaluate the performance of the models, these figures show the confusion matrix (see Figure 5.7 (A)), AUROC curve (see Figure 5.7 (B)), and accuracy and F1 score for given classification

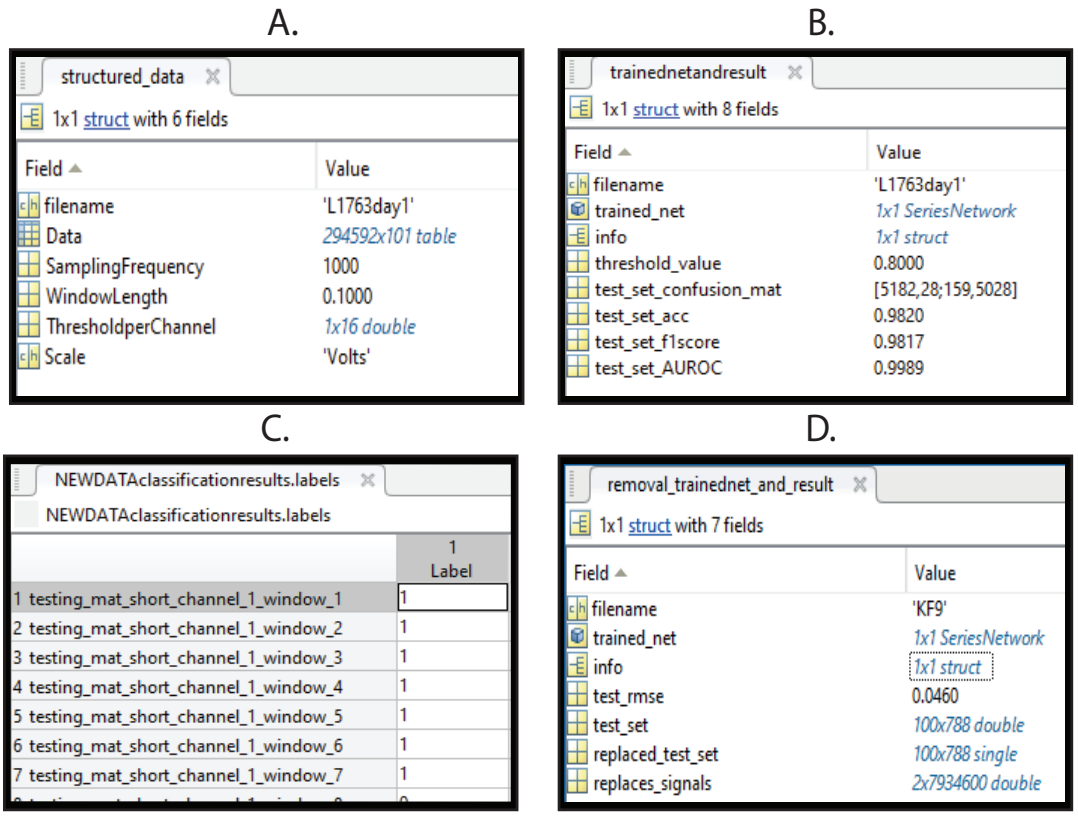

Figure 5.9: Screenshots of the SANTIA's saved files: labelled data (A), trained network and results (B), new data labels (C) and removed artefacts (D).

thresholds (see Figures 5.7 (C) and 5.7 (D)). The outputs generated by the toolbox for artefact removal are showcased in Figure 5.8, which displays examples of the outputs of 'View Test Results' (A) and 'Plot Channel' (B). In the upper sub-figure, an element of the test set is showcased in red in contrast to the forecast of the CNN-LSTM network in blue. In this particular example, while the forecast of the first peak is nearly identical to the signal, the following peaks have slightly less amplitude, which can be attributed to the fact that they are taking in the previous forecasts of the network. The sub-figure below showcases a channel before (red) and after removal (blue). The high amplitude artefacts which spanned 2 mV peak-to-peak have been removed and replaced by 50 milliseconds windows, and now the channel shows a uniform range of ± 0.05 mV, indicating the success of the methodology.

Finally, Figure 5.9 illustrates the contents of output files generated in each module. These files are saved in Matlab format (.mat) and contain key values for the user to quickly access them, as well as the processed variables needed for any posterior predictions.

5.3 Artefact Detection

A single channel classification model based on NNs, including MLP, LSTM networks and 1D-CNN, is proposed to detect artefacts. The basic diagrams of the main components of these architectures are depicted in Figure 2.1 and described in Section 2.2.1. For the MLP, a three-layer architecture was chosen, where the first layer equals the input size, the middle layer a tenth of the first layer, and the output layer 2 neurons, one for each class. Similarly, the LSTM network follows the same structure, but instead of having *tanh* activations for the output, it uses SoftMax to calculate the class probability. Lastly, the 1D-CNN network is an adaptation of AlexNet [155], which was done by flattening one dimension of the filters and pooling layers, and the components of the 12-layer architecture are listed in Table 5.1. The convolutional layers' filter sizes are expressed inside brackets, multiplied by the quantity and succeeded by the stride (s) and the same notation is used for the pooling window's sizes and stride. The input size, number of rectified linear units in the fully connected layers and SoftMax units in the classification layer are within brackets as well. Due to the lower dimensionality of the input, the number of filters was also reduced to multiples of 16.

Different window sizes were constructed and fed to the model to understand the effect of window size on the artefact detection process. Those longer than the labelling window were created by taking two consecutive bins with the same tag (or label) and expanding the original window (e.g., 100 ms) to avoid mixing of bins containing artefact and non-artefact labels. Due to the class imbalance between artefactual and non-artefactual windows, the latter were randomly downsampled to have an equal amount.

Table 5.1: Architecture of the 1D-CNN model used for artefact detection

Architecture	Component
Layer 1	Input [150]
Layer 2	Convolution 1 [1,11] x 32, s = 1
Layer 3	Max Pooling 1 [1,3], s = 2
Layer 4	Convolution 2 [1,5] x 64, s = 1
Layer 5	Max Pooling2 [1,3],s=2
Layer 6	Convolution 3 [1,3] x 128, s = 1
Layer 7	Convolution 4 [1,3] x 128, s=1
Layer 8	Convolution 5 [1,3] x 128, s = 1
Layer 9	Max Pooling 3 [1,3], s=2
Layer 10	Fully Connected [1024]
Layer 11	Fully Connected [512]
Layer 12	Classification [2]

5.3.1 Dataset #1

From the dataset, examples of different sizes were extracted, and from each size, a total of 294, 592 examples were used in this task, which was divided into training (80%), validation (10%) and testing(10%) sets. For the 1D-CNN and LSTM, the optimisation algorithm used was Adam, with an initial learning rate of 0.001, the momentum of 0.9 and a batch size of 1, 280. On the other hand, the MLP was optimised via a scaled conjugate gradient function. Figure 5.10 displays the performance of the training and validation set of the different sequence lengths for the three architectures, while the results have been compiled in Table 5.2. Models with an input size of 150 milliseconds have the lowest losses and highest accuracies, meaning it is the best trade-off between information fed to the model and its performance among the chosen window sizes for this dataset. While all three models achieved an AUROC of 0.99, the 1D-CNN has the highest test accuracy and has been able to properly classify the 150 ms samples shown in Figure 5.11, while the MLP and LSTM models have not been able to.

5.3.2 Dataset #2

For this dataset, different windows of 0.05, 0.1, 0.15 and 0.2 seconds were taken and fed to the model. The number of examples obtained after downsampling to

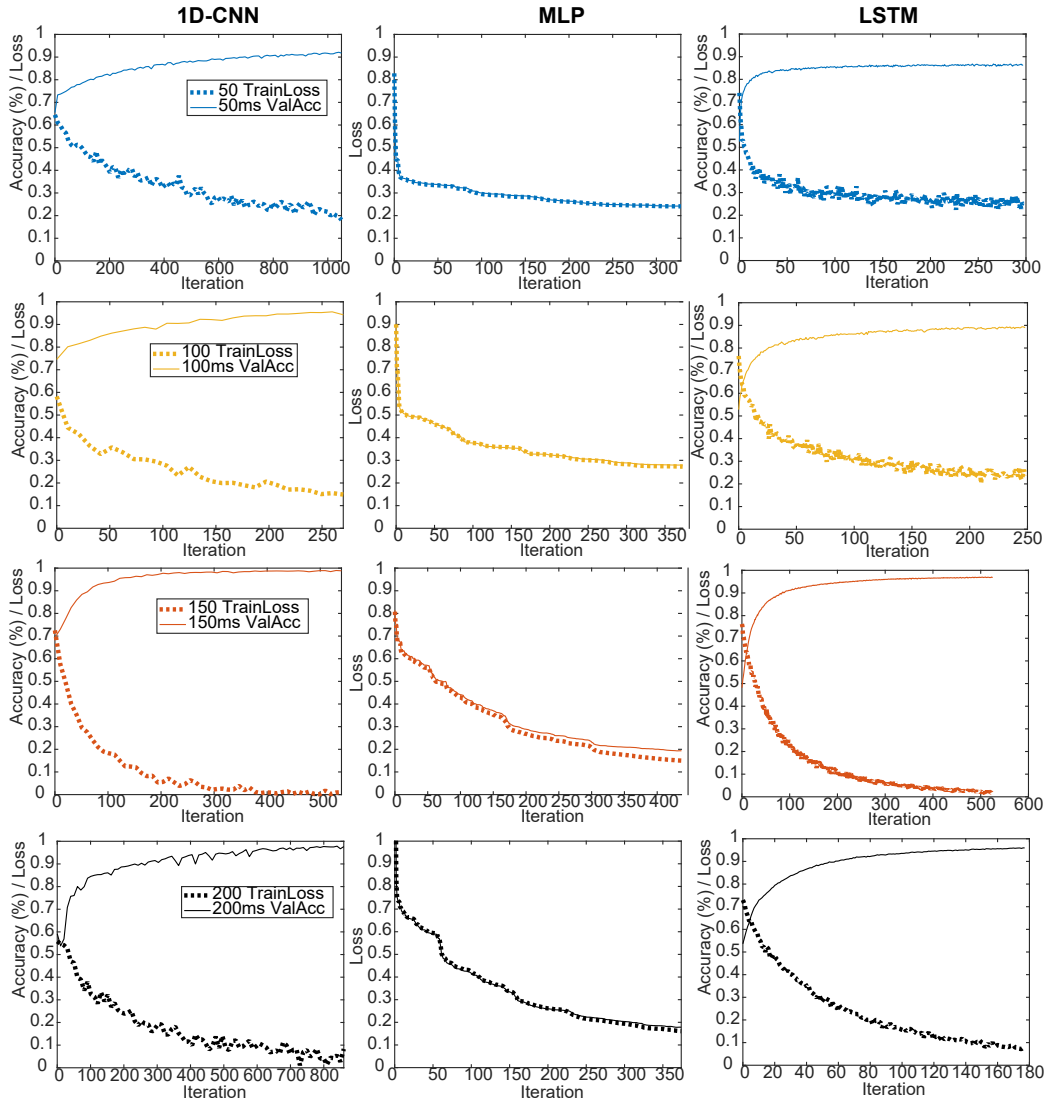

Figure 5.10: Training plots for artefact detection models trained with the first dataset.

balance the classes was, on average 275,687 per window size. The models were trained using the same hyper-parameter configuration used in the previous dataset. The performance of the models during training is shown in Figure 5.12. As they originate from different toolboxes, the MLP does not generate the accuracy throughout the training, thus it is not shown.

These results are consistent with previously obtained ones. They indicate that

Table 5.2: First dataset's results for different architectures and sequence length: Training Loss, Validation Accuracy, Testing Accuracy and Testing AUROC.

Parameters		Metrics			
Network	Input (ms)	Training Loss	Val. Acc.	Test Acc.	Test AUROC
MLP	50	0.24	0.78	0.78	0.86
	100	0.27	0.89	0.86	0.94
	150	0.15	0.94	0.95	0.99
	200	0.16	0.94	0.96	0.98
1D-CNN	50	0.18	0.92	0.91	0.97
	100	0.15	0.94	0.96	0.97
	150	**0.01**	**0.99**	**0.99**	**0.99**
	200	0.08	0.98	0.97	0.99
LSTM	50	0.25	0.86	0.86	0.94
	100	0.26	0.89	0.89	0.96
	150	0.02	0.97	0.97	0.99
	200	0.07	0.96	0.96	0.99

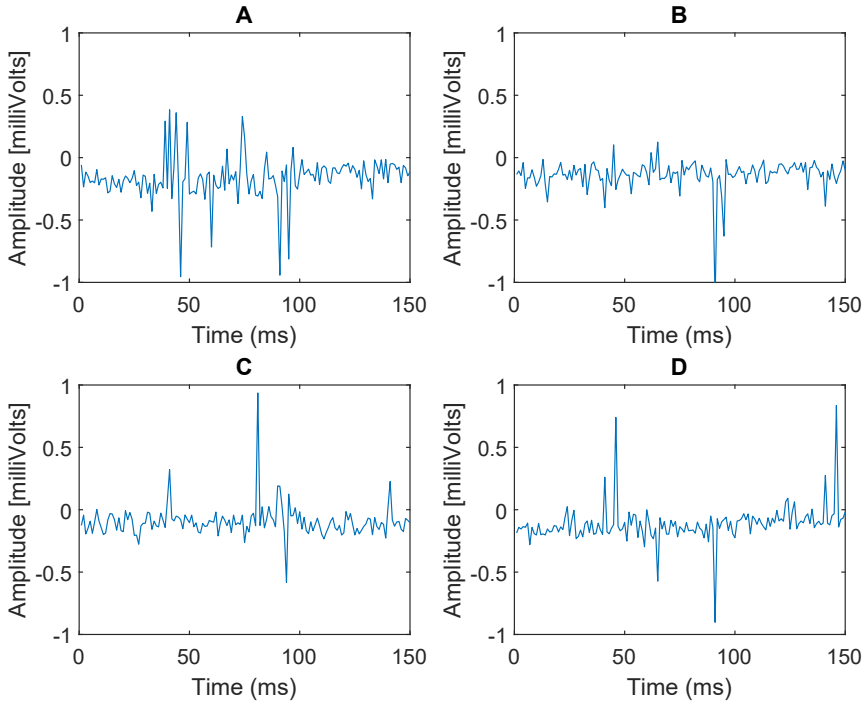

Figure 5.11: Examples of 150 ms bins of the first dataset correctly identified by the 1DCNN model, but not by the MLP or LSTM

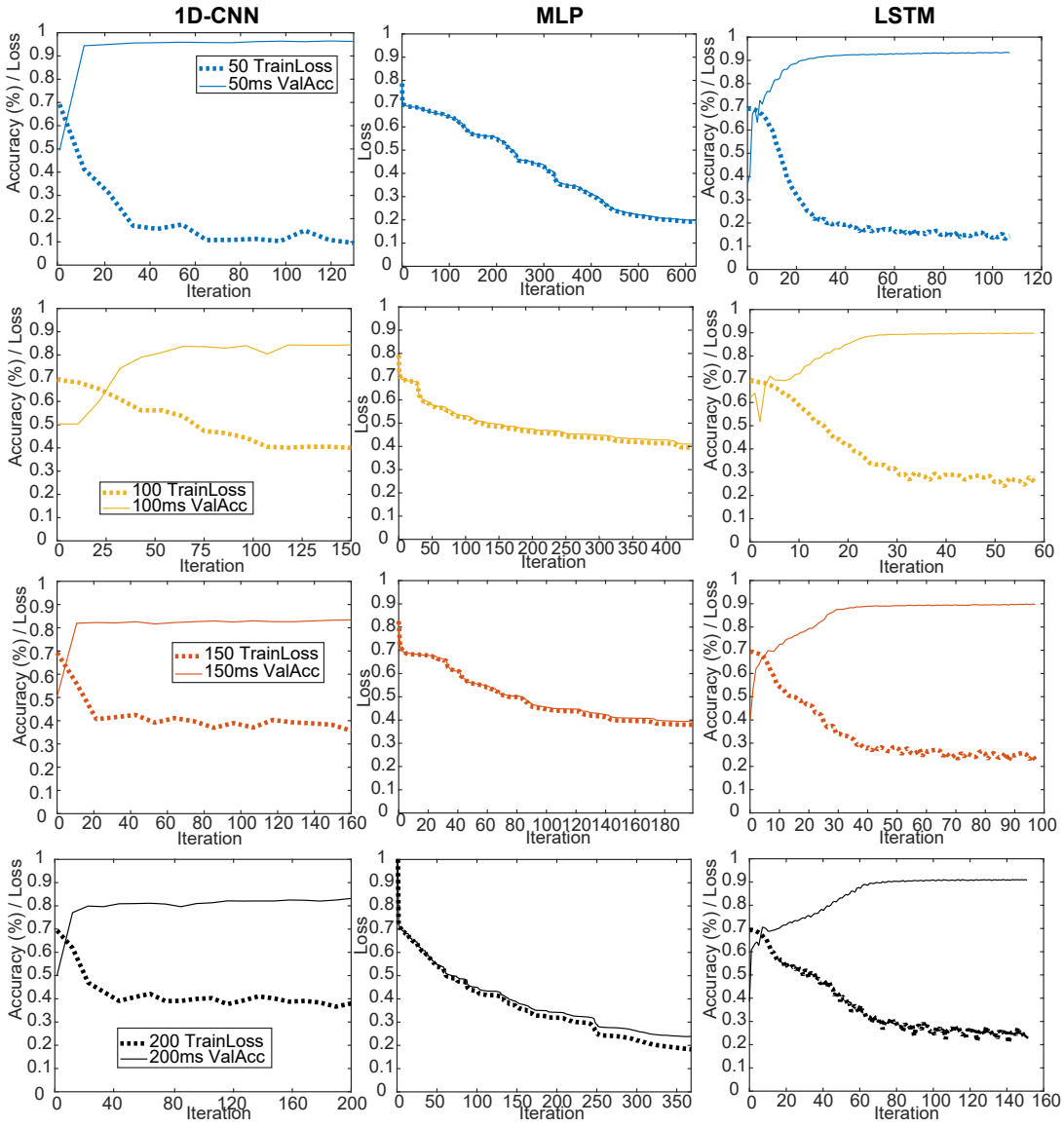

Figure 5.12: Training plots for artefact detection models trained with the second dataset.

since the filters from the 1D-CNN learn from regions of the signal instead of the individual values, they can learn more robust features of the signals and lead to better classification. Performance on the test sets is similar to that obtained in the validation set, as shown in Table 5.3. The best test set classification results were achieved by the 50 ms 1D-CNN, an accuracy of 96.5% and an AUROC of

Table 5.3: Second dataset's results for different architectures and sequence length: Training Loss, Validation Accuracy, Testing Accuracy and Testing AUROC.

Parameters		Performance Metrics			
Network	Input (ms)	Training Loss	Val. Acc.	Test Acc.	Test AUROC
MLP	50	0.20	0.92	0.92	0.98
	100	0.41	0.82	0.81	0.90
	150	0.39	0.83	0.83	0.90
	200	0.24	0.91	0.91	0.97
1D-CNN	**50**	**0.10**	**0.96**	**0.97**	**0.99**
	100	0.39	0.84	0.84	0.89
	150	0.37	0.83	0.83	0.91
	200	0.36	0.83	0.83	0.91
LSTM	50	0.16	0.93	0.94	0.99
	100	0.26	0.9	0.91	0.97
	150	0.25	0.89	0.90	0.97
	200	0.25	0.91	0.90	0.97

0.993, indicating the network has been able to learn successfully. Despite that the LSTM and MLP models of 50ms achieve a test AUROC of near 0.99, they were not able to correctly identify the samples shown in Figure 5.13, while the 1D-CNN could.

5.3.3 Dataset #3

Due to the low sampling rate of this dataset (250 Hz compared to 1 and 2 kHz), the window sizes were taken across two seconds. For this same reason, the number of artefacts is lower, with the datasets containing slightly over 3000 balanced examples, so the models were trained at a batch size of 64. The performance of the models during training is shown in Figure 5.14 while the results have been compiled in Table 5.4. Across all window sizes, the 1D-CNN outperforms the MLP and LSTM, with the best performance achieved with the 1500 ms input size. The LSTM models performs significantly poor than the other two models due to an underfitting caused by having a smaller training set than Datasets #1 and #2, when as this architecture has the higher number of parameters out of the three. As shown, different datasets are probable to have different optimal

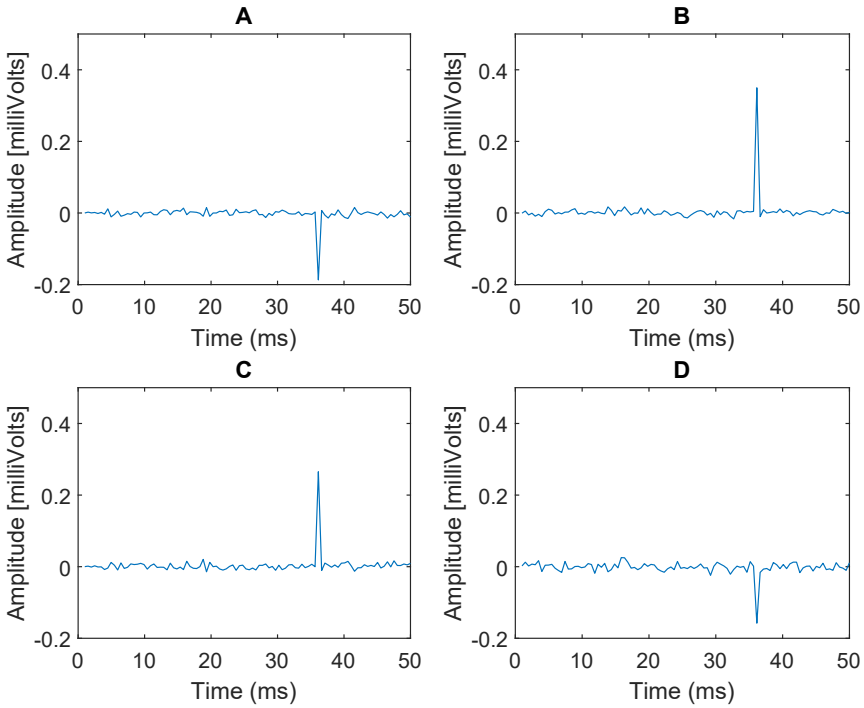

Figure 5.13: Examples of 50 ms bins of the second dataset correctly identified by the 1DCNN model, but not by the MLP or LSTM

trade-offs between window size and accuracy due to factors such as sampling rate and artefact frequency.

The results are on par with the previous datasets, indicating that the method is robust and generalisable. The 1D-CNN model has been shown to obtain the best scores in all cases, establishing it as the better architecture for this data type.

5.4 Artefact Removal

It is expected that training an LSTM network to forecast artefact-free data reliably can be successfully utilised to substitute artefactual sections of signals when information from other channels has been corrupted and cannot be used to approximate its real behaviour. The NN architecture was chosen due to the known capabilities of RNN, specifically LSTM, in recognising patterns from

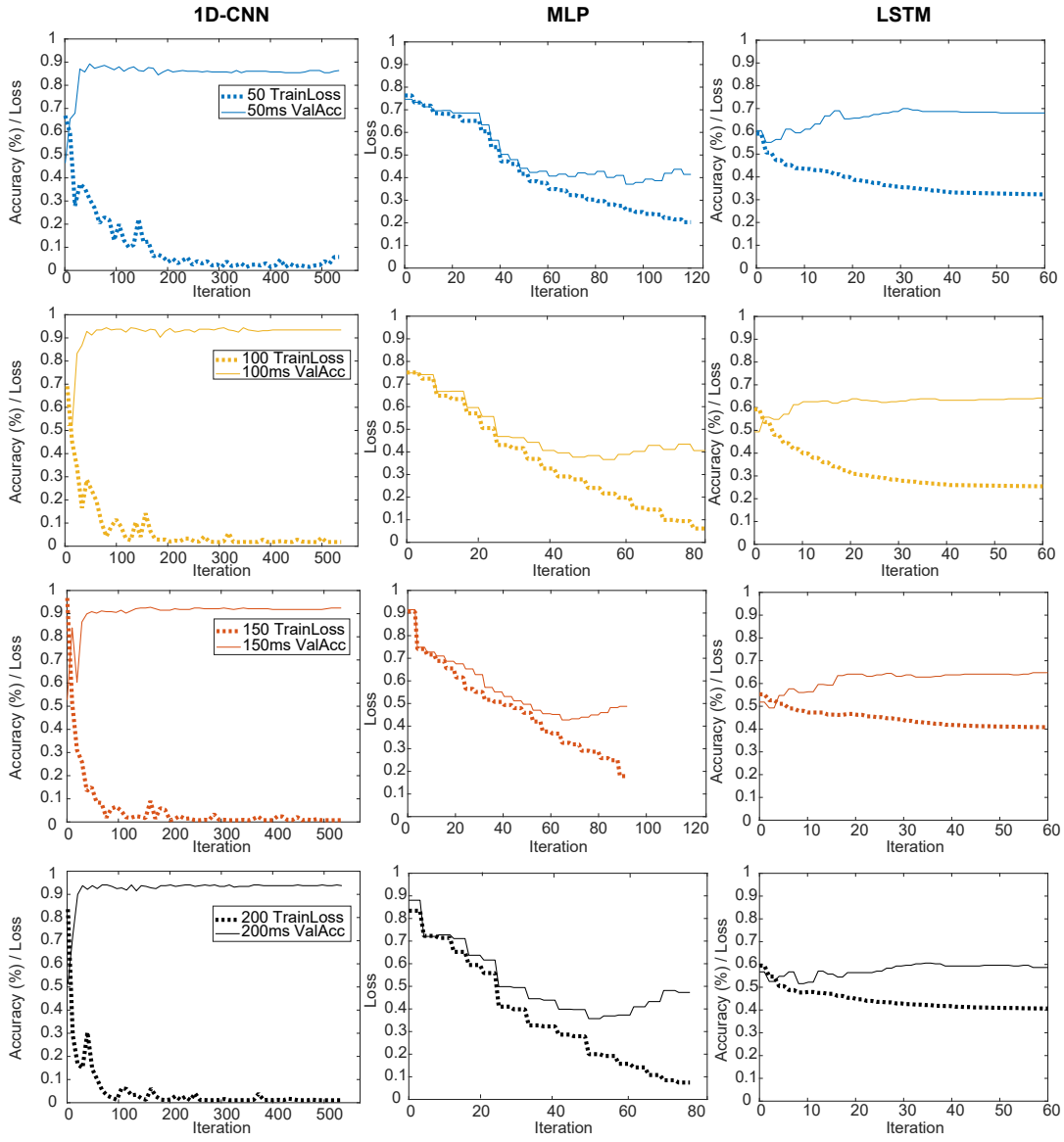

Figure 5.14: Training plots for artefact detection models trained with the third dataset.

sequential data. Kim et al. [156] have proven that it is possible to predict the behaviour of LFP from 10 to 100 milliseconds forward in time via the use of a regressive LSTM network. A similar approach was established by Paul [157], who used a stacked LSTM to forecast a single point of an EEG signal by feeding the previous 70 milliseconds. Their test data was composed of 9 subjects, in

Table 5.4: Third dataset's results for different architectures and sequence length: Training Loss, Validation Accuracy, Testing Accuracy and Testing AUROC.

Parameters		Performance Metrics			
Network	Input (ms)	Training Loss	Val. Acc.	Test Acc.	Test AUROC
MLP	500	0.217	0.85	0.85	0.88
	1000	0.216	0.87	0.85	0.90
	1500	0.326	0.83	0.83	0.88
	2000	0.200	0.85	0.85	0.86
1D-CNN	500	0.052	0.87	0.92	0.94
	1000	0.001	0.93	0.96	0.98
	1500	**0.001**	**0.93**	**0.97**	**0.98**
	2000	0.001	0.94	0.95	0.99
LSTM	500	0.523	0.68	0.61	0.68
	1000	0.523	0.64	0.65	0.71
	1500	0.600	0.64	0.64	0.70
	2000	0.570	0.60	0.50	0.59

which they achieved correlation coefficients of over 0.8 across all of them. In addition, there have been recently reported applications of LSTM in artefact detection [158, 159, 160, 161] as well as RNN in artefact removal [162, 163, 164].

Figure 5.15 shows how an LSTM network was trained to predict typical behaviour using a sliding window method. The sliding window approach employs data at a time t to predict the value at $t+1$, then uses the new predicted value when forecasting the value at $t+2$. Given a channel of LFP (X_t) with datapoints from $t=1$ to N, where each segment segment of duration l has a label $y_k \epsilon \{-1, 1\}$, and a trained forecasting network TFN of input size $In = 1, .., I$ and output size $Out = 1, ..., O$. The replacement algorithm is expressed as described in Algorithm 1, where the resulting channel with removed artefacts in stored in RX_t.

Matlab's Deep Learning Toolbox [165] was used to build and train the network of LSTM cells. The LSTM models were made up of the following layers: an input layer, a hidden layer equal to one-tenth of the input, and an output layer equal to the number of predicted points. For comparison, a more complex architecture was trained composed of convolutional and recurrent layers CNN-LSTM described in Table 5.5. The optimisation algorithm used was Adam, a method that computes

Algorithm 1 Artefact Removal Algorithm

Input: X_t, y_k, TFN
Output: RX_t

Initialisation :

X_t = sample channel where N = total data points
y_k = label where l = label window data point size
TFN = trained forecasting network where In = network input size and Out = network output size
$W = N/l$
$Inwin = round(In/l)$
$RX_t = X_t$
for $k := Inwin$ to W **do**
  if $(y_k == 1)$
  $p = 1$
  while $p \leq l$ **do**
    $position = k * l + p$
    $Ninput = X_t((position - In) : position)$
    $Pred = predict(\text{TFN}, Ninput)$
    $RX_t(position : (position + Out)) = Pred$
    $p = p + Out$
  end while
end for
return RX_t

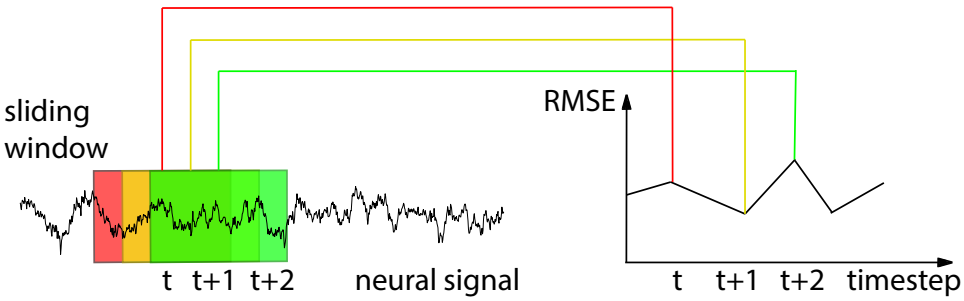

Figure 5.15: An illustration of sliding window approach diagram.

individual adaptive learning rates for each of the parameters via the combination of two algorithms, momentum and root mean square propagation [166]. The optimiser was set to an initial learning rate of 0.0001, the momentum of 0.9 and a batch size of 516 for the first dataset, 128 for the second dataset and 64 for

Table 5.5: Architecture of the CNN-LSTM model used for artefact removal.

Layer	Type	Description
1	sequenceInput	-
2	sequenceFolding	
3	convolution2d	size=5, filters=32, dilation=1
4	batchNormalization+elu	
5	convolution2d + elu	size=5, filters=32,dilation=2
6	convolution2d + elu	size=5, filters=32,dilation=4
7	convolution2d + elu	size=5, filters=32,dilation=8
8	convolution2d + elu	size=5, filters=32,dilation=16
9	averagePooling2d	size=1,stride=5
10	sequenceUnfolding	with flattening
11	gru	128
12	lstm	64
13	dropout	0.25
14	lstm	32
15	dropout	0.25
16	regression	-

the third dataset, due to the different sample sizes. The loss function of the regression layer was the half-mean-squared-error of the predicted responses for each time step, not normalised by N:

$$loss = \frac{1}{2S}\sum_{i=1}^{S}\sum_{j=1}^{N}(x_{ij} - \hat{x}_{ij})^2 \tag{5.1}$$

where x_i is a forecasted data point, $\hat{x}_i$ the real value of the LFP at that data point, S is the output sequence length, and N the number of examples in the training or validation set. The testing set was used to calculate the RMSE of the output over an unseen segment.

The linear approximator autoregressive moving average with extra input (ARMAX) was applied to the same testing and model evaluation data for a performance reference. It was chosen as auto-regressive models are very popular methods in time series forecasting, due to their robustness and efficiency [167], and have been used to benchmark NN-based approaches [168, 169]. Following the description by Yan et al. [170], given a LFP time series (X_t, y_t) for $t = 1$ to

N, where $X_t = t(x_{t1}, x_{t2}, ..., x_{tk})$ is the input vector at time t with k elements and y_t is the corresponding neuronal activity voltage at time t, this model approximates a polynomial equation, written as:

$$A(q)y_t = \sum_{i=1}^{k} B_i(q)x_{ti} + C(q)e(t) \tag{5.2}$$

where $A(q), B(q)$ and $C(q)$ are the polynomials expressed with a time shift term q^{-1} shown in Equation 5.3 and $e(t)$ is the white-noise disturbance value.

$$\begin{cases} A(q) = 1 + a_1q^1 + ... + a_{n_a}q^{n_a} \\ B_i(q) = 1 + b_{1i}q^1 + ... + b_{n_{bi}}q^{n_{bi}+1} \\ C(q) = 1 + c_1q^1 + ... + c_{n_c}q^{n_c} \end{cases} \tag{5.3}$$

Here, the hyper-parameters n_a, n_b, n_c denote the orders of the ARMAX model's auto-regressive part, external input vector with k elements and moving average, respectively. Finally, a_i, b_{ik} and c_i are the polynomial coefficients determined by using polynomial curve fitting. Having described the methodology, the datasets used to evaluate it are described.

5.4.1 Dataset #1

The results of the 108 models are captured in Figure 5.16, where the test set RMSE of 100 ms is shown. Regarding the LSTM network, the test performance improves from single value predictions to fifty points one and declines with the hundred point one. Regarding the time input, larger sequences above 0.5 s are noticeably under-performing, indicating that the network struggles to correlate long-passed events with current ones. The best performing LSTM model is the $400ms$ input and 50 points prediction model with an RMSE of 0.773. On the other hand, the CNN-LSTM models perform better with longer output and input sequences. Overall, the performance of the CNN-LSTM is better than the LSTM models, with the best score being 0.699 of the $600ms$ input and 100 points prediction model.

To confidently prove the effectiveness of this method, it has been compared to ARMAX. The ARMAX model was given the same $600ms$ examples for

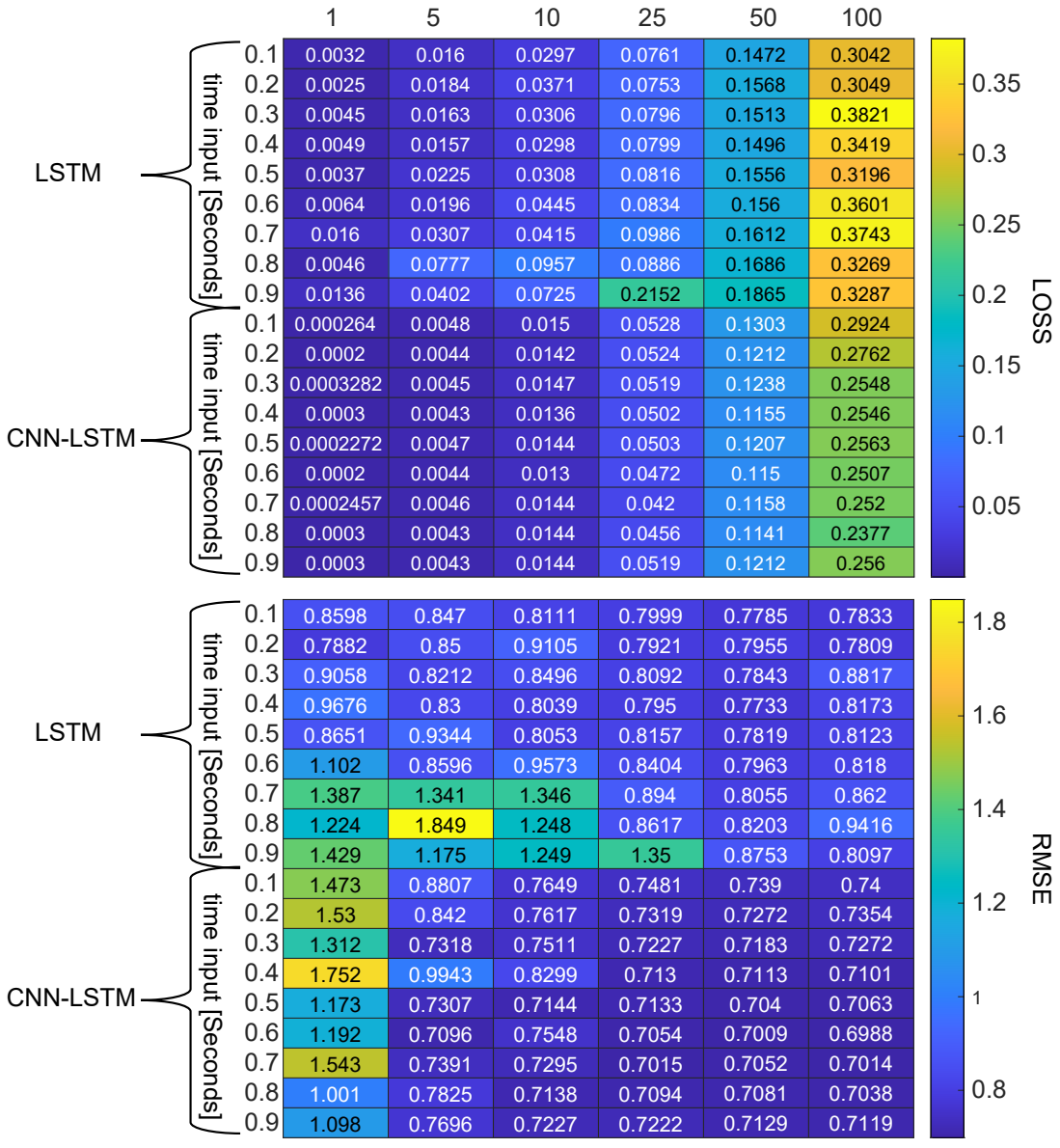

Figure 5.16: Validation loss (top) and test set RMSE (bottom) of each predicted points (column) vs time input (row) of the LSTM and CNN-LSTM models trained with dataset 1.

defining the model and the 100 points to calculate the RMSE to contrast their performance. To confidently prove the effectiveness of this method, it has been compared to ARMAX, which was given the same $600ms$ examples for defining the model and the 100 points to calculate the RMSE. The computational time

was evaluated by forecasting 0.1 s of recording and averaged over 100 iterations. Results are depicted in Table 5.6, where our method outperforms it in RMSE as in computational time. The time difference is mainly because the ARMAX needs to estimate the grades of the polynomials for every new sequence for accuracy, unlike the CNN-LSTM, which can forecast very rapidly once it has been trained. All models were tested on a general-purpose Alienware M17 R4 laptop consisting of 32 gigabytes of RAM and Intel® Core™ i9-10980HK CPU @ 2.40 GHz processor.

Having defined the best model, a total of $19,100$ one-second artefactual segments were extracted from the data of the last two recording days, with the condition that the first $600ms$ had to be artefact-free. Each $0.1s$ sub-portion of these segments, which had the 'artefact' label, was replaced by the forecast of the network, thus, it was used as part of the input if the following sub-portion also shared the same label. An exemplary case is shown in Figure 5.17, where the network has been able to mimic normal behaviour more than the ARMAX model.

In regards to segment's power, Figure 5.18 shows the violin plot [1] distribution of the three groups: the normal segments, artefactual segments and after replacing them. The method has been successful in replacing the high power artefactual segments with ones which resemble normal activity. While the range is still larger than the artefact free, the distribution has shifted considerably to lower power levels. The presence of high-power segments indicates a shortcoming of the method, where surrounding information has high power, but only one or two sub-portions do exceed the defined threshold, so the total sum of the processed segment still has a high value.

[1] Function extracted from https://github.com/bastibe/Violinplot-Matlab

Table 5.6: Performance Comparison for Forecasting Methods in the first dataset

Method	RMSE	Time (seconds)
LSTM	0.773	0.0046
CNN-LSTM	0.699	0.0059
ARMAX	1.871	1.4841

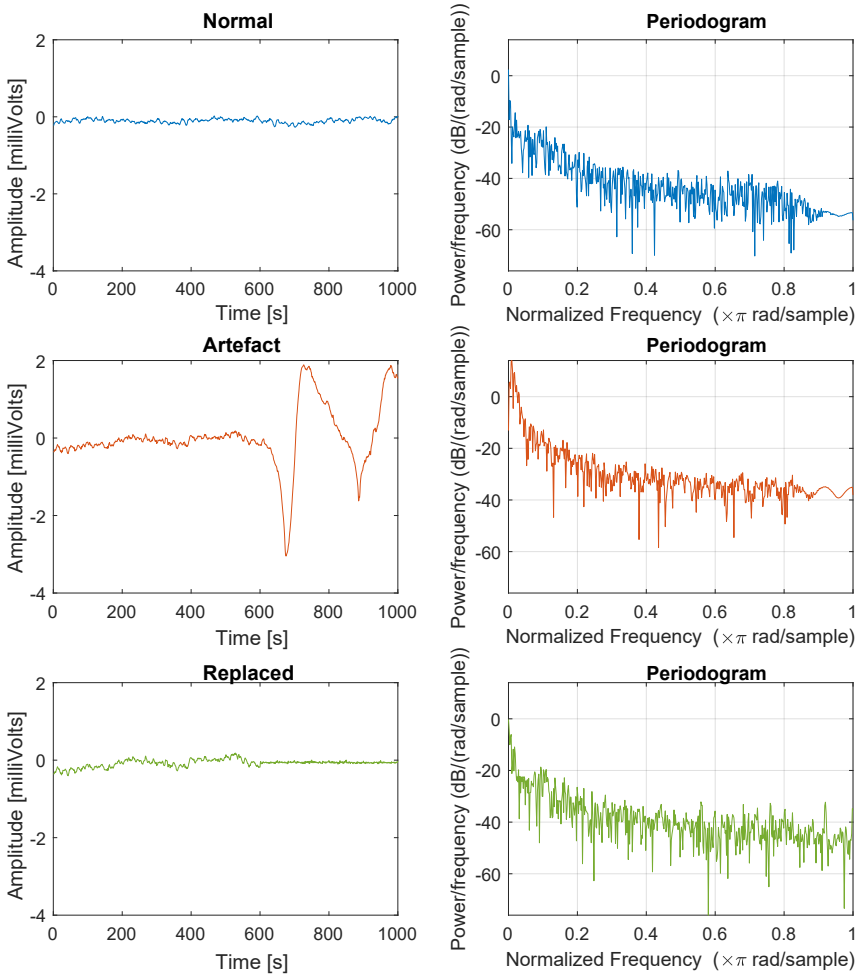

Figure 5.17: Examples of normal (blue), artefactual (red) and replaced-segments signals(green) alongside their periodograms for dataset 1. The method has recreated the normal signal in amplitude and spectral properties.

5.4.2 Dataset #2

Figure 5.19 shows the performance of the 54 LSTM models in the form of validation loss and test set RMSE over 100 milliseconds. In regards to the output of the network, the test performance improves from single value predictions to fifty points one and then remains constant. Regarding the time input, larger sequences above 0.6 seconds don't present any major performance improvements. The best performing LSTM model is the 600 milliseconds input and 10 points prediction model with an RMSE of 0.1538.

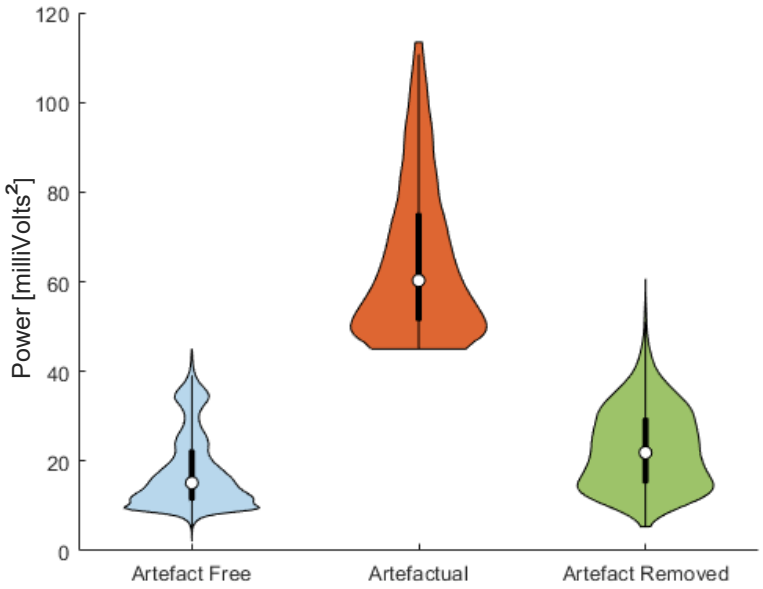

Figure 5.18: Violin Plot of power in the normal (blue) 1-second segments, artefactual segments before (red) and after (green) processing from dataset 1. The method has reduced the power of the artefactual segments to similar values to the artefact-free segments.

On the other hand, out of the 54 CNN-LSTM models, the best performance is achieved with an output of 20 data points across all inputs, while the worst performances are achieved with 50 or 100 output points. Overall, the performance of the CNN-LSTM is better than the LSTM models, with the best score being 0.1463 of the 200 milliseconds input and 20 points prediction model. The differences in performance of the networks between Datasets #1 and #2 for the same length sequences can be attributed to the variations between them such as sampling frequency (1kHz vs 2kHz), filtering (0-500 Hz vs 0.7–150 Hz) and the difference in the type of the artefacts (movement vs instrumental).

The ARMAX was given the same 200 milliseconds examples for defining the model and the 100 milliseconds to calculate the RMSE, which achieves a performance of 0.1449. This indicates a slightly better performance than the NNs. However, it must be factored in that the signals have been significantly low-passed filtered, and the signals have a near-sinusoidal shape. If used on a

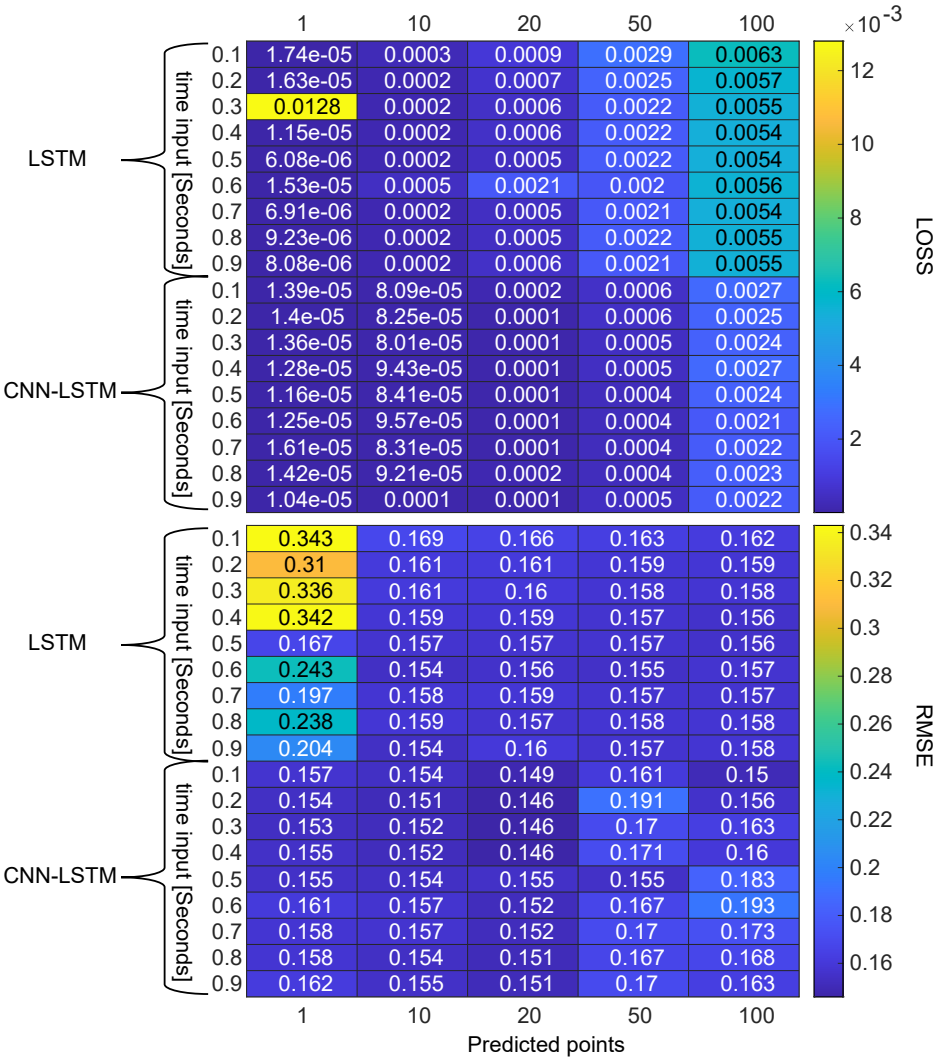

Figure 5.19: Validation loss (top) and test set RMSE (bottom) of each predicted points (column) vs time input (row) of the LSTM and CNN-LSTM models trained with dataset 2.

different set that retains higher frequency components, the performance of the ARMAX model would be challenged, as shown in the previous dataset.

Results are depicted in Table 5.7, where despite the similar RMSE, the deep learning method is more computationally efficient; thus, CNN-LSTM is chosen as the best compromise between the two. Having defined the best model, a total of 7275 one-second artefactual segments were extracted from the data of the rodents,

Table 5.7: Performance Comparison for Forecasting Methods in the second dataset.

Method	RMSE	Time (seconds)
LSTM	0.1538	0.0433
CNN-LSTM	0.1456	0.0547
ARMAX	0.1449	1.5425

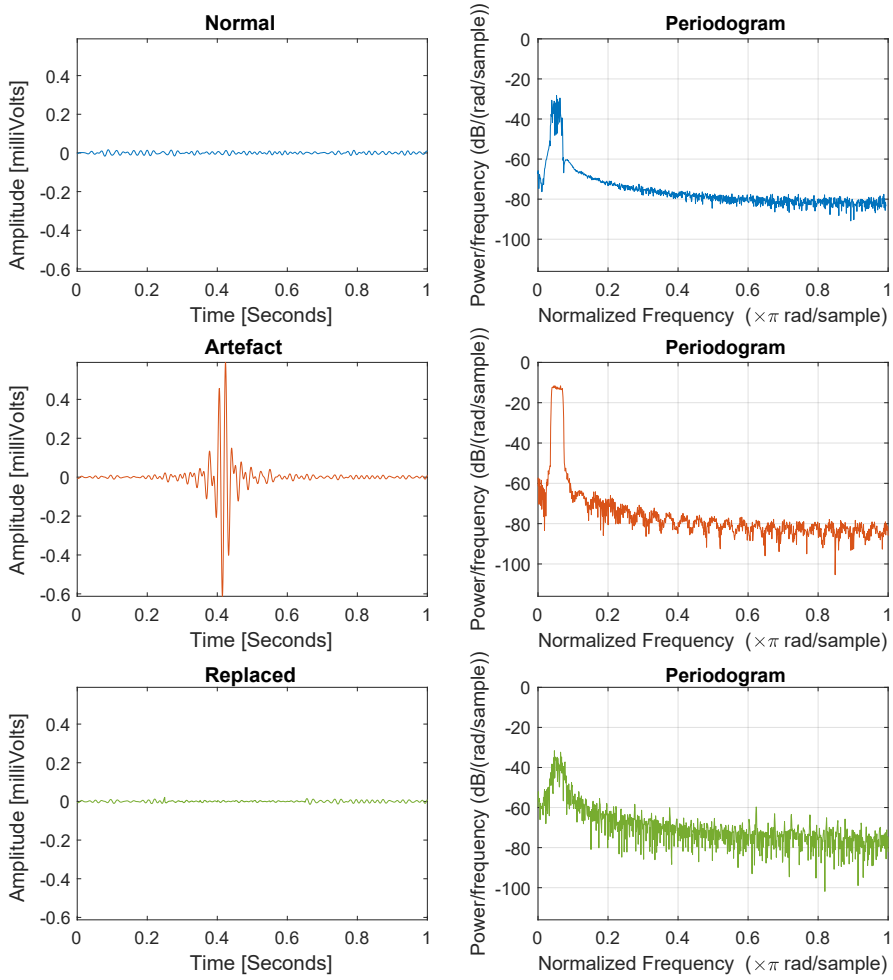

Figure 5.20: Examples of normal (blue), artefactual (red) and replaced-segments signals(green) alongside their periodograms for dataset 2. The method has recreated the normal signal in amplitude and spectral properties.

with the condition that the first 200 milliseconds had to be artefact-free. The forecast produced by the network replaced every 50 milliseconds window labelled

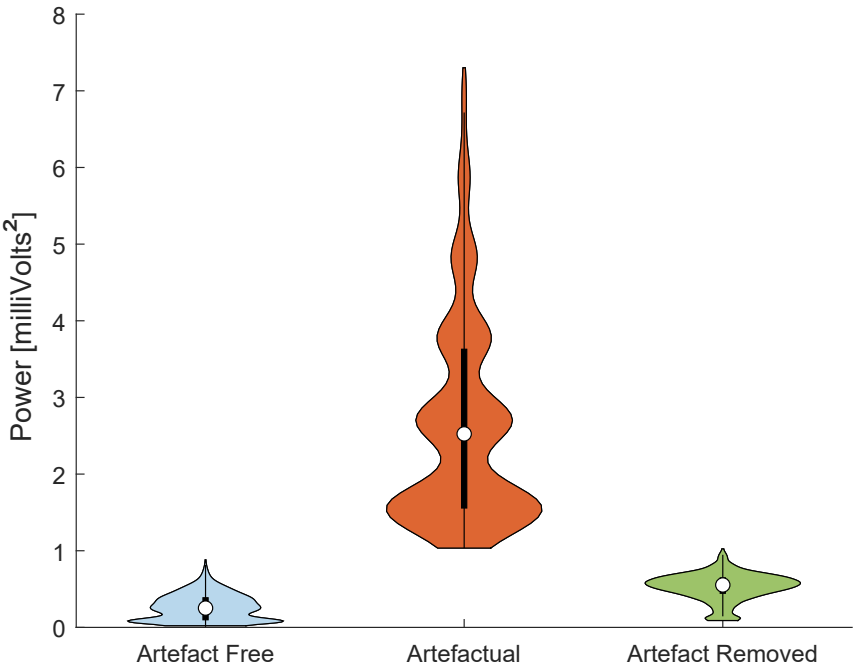

Figure 5.21: Violin Plot of power in the normal (blue) 1-second segments, artefactual segments before (red) and after (green) processing from dataset 2. The method has reduced the power of the artefactual segments to similar values to the artefact-free segments.

'artefact' in each segment, which in turn was used as part of the input if the following window also shared the same label.

The first comparison of the results is made through visual inspection. Examples of normal, artefactual, and replaced-segments signals alongside their periodogram are illustrated in Figure 5.20. The new signal after the processing had had its high amplitude artefact removed, demonstrating the method's success. This can also be observed in the periodogram, where the artefactual example possesses a low-frequency component that exceeds the -20 dB, but the physiological and the processed signal have a power of approximately -40 dB.

Regarding segment's power, Figure 5.21 shows the violin plot distribution of the three groups: the normal segments, artefactual segments and after replacing them. The method has been successful in replacing the high-power artefactual segments with ones that resemble normal activity. While the median is higher than the artefact-free, the distribution has shifted considerably to lower power

Table 5.8: Performance Comparison for Forecasting Methods in the third dataset.

Method	RMSE	Time (seconds)
LSTM	0.7418	0.0035
CNN-LSTM	0.7341	0.0087
ARMAX	3.1813	0.3645

levels.

5.4.3 Dataset #3

The results of the different models are compiled in Figure 5.22, where the validation loss and the RMSE over 1 second of the test set are shown. For the 15 LSTM models, the performance improves with longer output sequences but is best with 2 seconds of input. Thus, the best performing LSTM model is the 2-second input - 1-second output, with an RMSE of 0.7418. Regarding the CNN-LSTM models, performance does not vary significantly across input or output length; however, the best model is obtained with 1-second input - 1-second output with RMSE of 0.7341. Across all combinations, the CNN-LSTM outperforms the LSTM, as it can extract richer features.

Subsequently, the comparison to the ARMAX model was carried out. The ARMAX was given 1 second of recording to define the model and asked to forecast the subsequent second to calculate the RMSE, achieving a score of 3.1813. The difference in the performance of the ARMAX between the two datasets can be attributed to the fact that the one being evaluated has not been heavily filtered, and retains high-frequency components, making it more difficult to adjust a model. When looking at the overall performance of RMSE and computational time in Table 5.8, the CNN-LSTM stands out as the best-performing method.

With these results, four-second (i.e., 1000 data points at 250 Hz) artefactual segments are extracted with the condition that the first second had to be artefact-free, for a total of 3826 examples. The forecast produced by the network replaced every 1-second window labelled 'artefact' in each segment, which in turn was used as part of the input if the following window also shared the same label. To

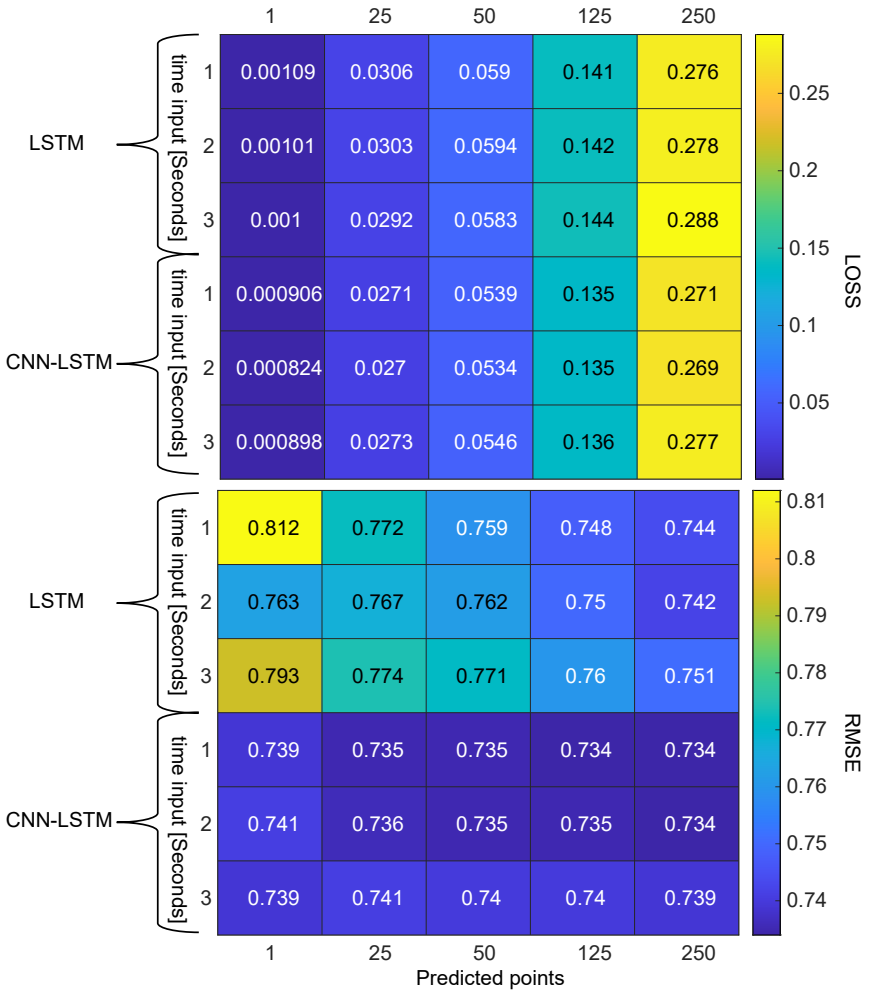

Figure 5.22: Validation loss (top) and test set RMSE (bottom) of each predicted points (column) vs time input (row) of the LSTM and CNN-LSTM models trained with dataset 3.

evaluate the results, examples of the three signals (i.e., normal, artefactual, and replaced-segments signals) with their corresponding periodogram are shown in Figure 5.23. Compared to normal segments, artefacts have higher amplitude and frequency, in other words, a non-physiological waveform. This can be observed in the periodogram in the repeated round peaks, and the higher frequencies don't decay as much power-wise. By replacing the segment, the smoothness of the spectrum power decay is returned.

Lastly, the violin plot of the power of the four-second segments of the three

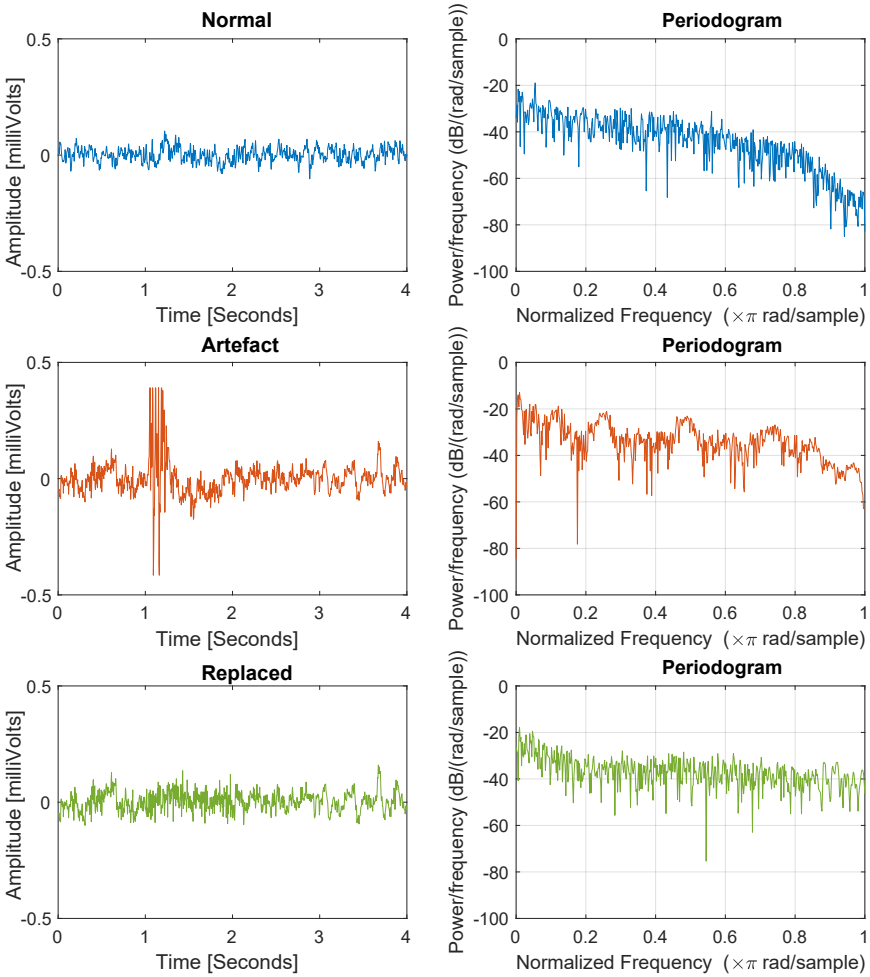

Figure 5.23: Examples of normal (blue), artefactual (red) and replaced-segments signals (green) alongside their periodogram from dataset 3. The method has recreated the normal signal in amplitude and spectral properties.

signals is displayed in Figure 5.24. Although the distribution has lowered significantly to values resembling normal activity, the previously mentioned shortcoming is still present, as cases with surrounding high power are not replaced as they have not exceeded the threshold.

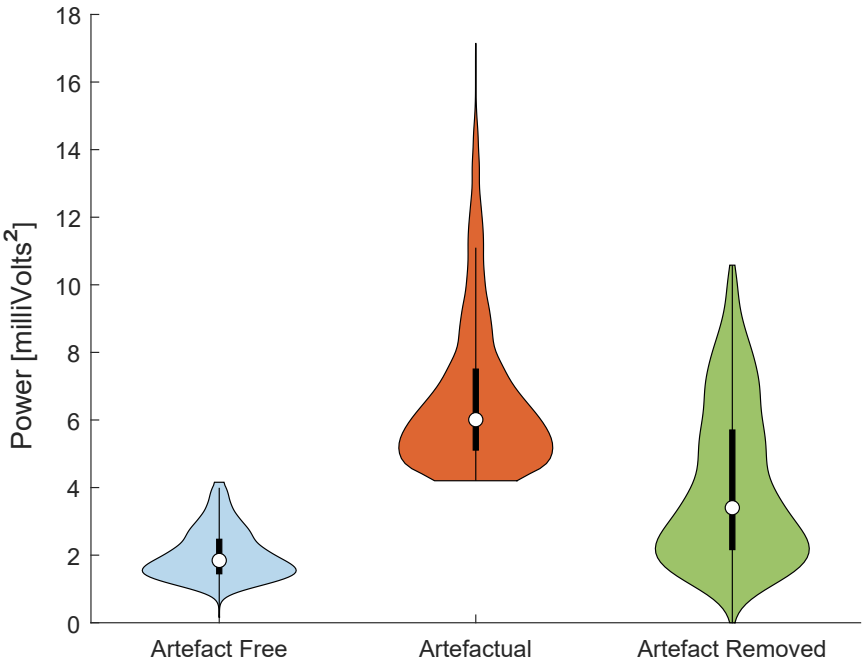

Figure 5.24: Violin Plot of power in the normal (blue) 1-second segments, artefactual segments before (red) and after (green) processing from Dataset #3. The method has reduced the power of the artefactual segments to similar values to the artefact-free segments.

5.5 Discussion

The SANTIA toolbox was developed to facilitate and aid in the reproducibility of detecting and removing artefacts in recorded LFP. The simple four-module GUI is designed for researchers without a programming background. The built-in methodology will allow them to quickly label, detect and remove the artefacts in their data. It is a project under constant development, and the current version provides an environment where new features can quickly be implemented and adapted to the toolbox. Examples of future developments include:

Explainability With the help of explainability, artefact detection may be improved. Explainable models can assist new researchers in determining which aspects of the signal indicating the existence of artefacts and the origin of the artefacts. For example, Grad-CAM can be used to generate heatmaps of the

signal that show the areas affected by the artefact. Alternatively, the decision-making of a NN-based model may be understood using a surrogate, interpretable model [171]. Consequently, researchers' confidence in these systems is improved thanks to the model's feedback.

Multi-State Classification Currently, the toolbox allows for binary classification, which is an output that assigns the label of 'normal' or 'artefact' only. In the future, the user will be allowed to customise the output of the classification model for multiple outputs. This is useful when there is more than a single normal state, for example, the sleep stages present in Dataset #4, or when there are multiple artefacts of known origin they wish to identify.

Online Processing The tool currently allows for offline labelling, but it can be expanded to allow the analysis of signals as they are being recorded to optimise the process.

Expand Format Compatibility There are different libraries for deep learning, such as the TensorFlow-Keras, Caffe, and the ONNX (Open Neural Network Exchange) model formats for NN layers [172]. The possibility to read those formats and the options to import from and save to HDF5 files for the neuronal data under the epHDF standard [173].

User Experience As this app is adopted by the community, it will improve its shortcomings with feedback. The inclusion of testing data and a video tutorial are also planned. Optimising some routines via parallelism is also a feature desired to be included due to the possible large sizes of data files.

Multi-Modality Incorporating another source of information (e.g. sensor signal or video) can facilitate and improve the detection of artefacts [174]. A new module would allow the incorporation of such data to facilitate the labelling process or as part of a classification model's input.

Portability As a long-term goal, the implementation in a portable device, e.g. FPGA or Arduino board is considered to expand the practicality of its usage.

From the methods perspective, the artefact detection showed consistent results across all three datasets, where the 1D-CNN stood out. The novel artefact removal method based on forecasting achieved good performance as well. However, it requires significant tuning of the network's input-to-output ratio, which is time-consuming. In addition, if the surrounding information of an artefact has high power, but only one or two sub-portions do exceed the defined threshold, these will not be removed.

5.6 Conclusion

This chapter showcases the development of artefact detection and removal algorithms for in-vivo neural recordings, which were ported to a toolbox that allows their use without needing a programming background. The performance of the toolbox is promising, as it was tested with a private dataset and validated with two open-access datasets. Three deep learning models were compared for artefact detection, out of which the 1D-CNN proved to achieve the best performance. A prediction method using a sliding window technique was presented for artefact removal. Two NNs architectures with recurrent and convolutional layers and ARMAX were compared. The best performance was achieved by the CNN-LSTM model, revealing that the forecasted data may be used to replace artefact parts successfully in LFP recordings. It is under continuous improvement, and plans include features such as the expansion of the format compatibility, improving the user experience and allowing the use of multi-modality to improve the models by complementing the information of the neuronal recordings, among others. To conclude, SANTIA has been further developed as a useful tool for those looking to either detect or remove artefacts automatically. It is hoped that the neuroscience community adopts this tool, and with their feedback and our future plans, an improved toolbox will be achieved.

Chapter 6

Detection of Alzheimer's Disease at an Early Stage

6.1 Introduction

Improving the understanding of Alzheimer's Disease (AD) is key for an early diagnosis and treatment, so the search for alternative and objective biomarkers is crucial. To this end, in recent years, data-driven solutions have been developed to tackle this disease [175]. These techniques, whose benefits have been described in Chapter 1.1, gather the information to construct a model that can be used to make inferences about unseen data and have been widely used in diverse fields.

When using an ML-based approach in neuronal signal classification, features from the different domains, such as temporal, spatial and spectral, can be extracted. However, little work has been done to see the impact of the cross-domain features on the model's decision-making process. Whilst the results remain accurate, most ML models show no explanation of the decision-making process. This makes them appear as black boxes and limits their adoption by scientists, physicians and other experts. Explainability is required to develop models grounded in reality, which follow our understanding of the disease's effects. In this Chapter, three animal models (control (WT), single transgenic AD model (ST) and double transgenic AD model (DT)) were used to record brain activities as LFPs (see Section 3.2). They applied

appropriate ML models to detect the disease at its early stage accurately. The decision-making process of the models was also explored to map them to known AD markers using activation maps. Late fusion of features was employed for the best-performing models from each domain to improve the overall detection performance. Furthermore, the robustness of the models to artefacts was evaluated by randomly masking channels to simulate the removal of affected segments and assess the classification of stress-period segments.

The remainder of the Chapter is partitioned into the methodology in Section 6.2. Section 6.3 shows and discusses the results obtained by the different models for each feature set. Lastly, Section 6.4 concludes the Chapter.

6.2 Methods

A block diagram of the proposed ML pipeline is shown in Figure 6.1. Subsequent sections describe its different models in detail.

6.2.1 Machine Learning Models

The following sections present the different ML models developed for the temporal, spatial and spectral domains.

6.2.1.1 Models for Temporal Domain

For temporal models, the raw signal is used as an input. That means that the method must be able to extract relevant features by itself, which is one of the properties of NNs. Since their development, there have been many improvements in architecture and their complexity. For the temporal models, the traditional MLP is first explored, followed by a state-of-the-art CNN for EEG recordings EEGNET [176] (imported from Keras-python). Subsequently, models known to perform well for time series are employed, that is, a recurrent (i.e. LSTM) network that averages the inputs across channels to feed it to said layer. Lastly, a combination of LSTM with CNN as a feature extractor, the LSTM-CNN network. The layers of these networks are illustrated in Figure 6.2 (A).

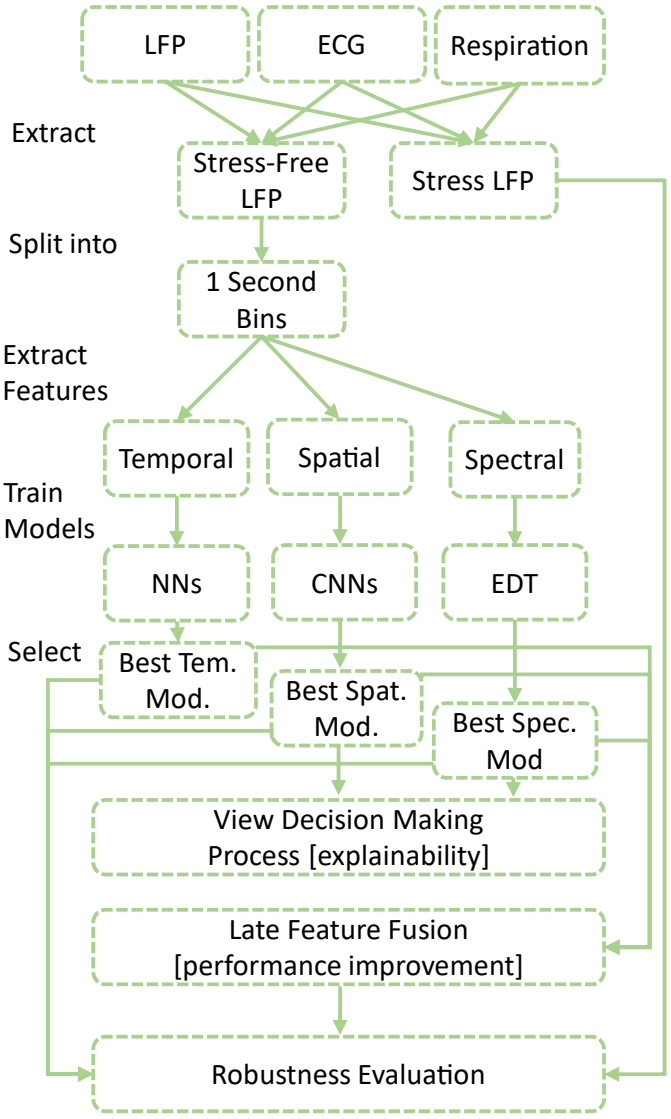

Figure 6.1: The proposed ML pipeline. Individual models considered are: neural networks (NNs) for the temporal domain, convolutional neural networks (CNNs) for the spatial domain and ensemble decision trees (EDT) for the spectral domain.

The MLP was chosen as it is the simplest architecture, however, it does not have any intrinsic properties to deal with time series. The EEGNET was chosen as it is the current benchmarking standard for EEG, due to its channel-wise convolutions, nonetheless, there is the possibility that the model's performance won't translate to different data. LSTM networks were chosen as recurrent

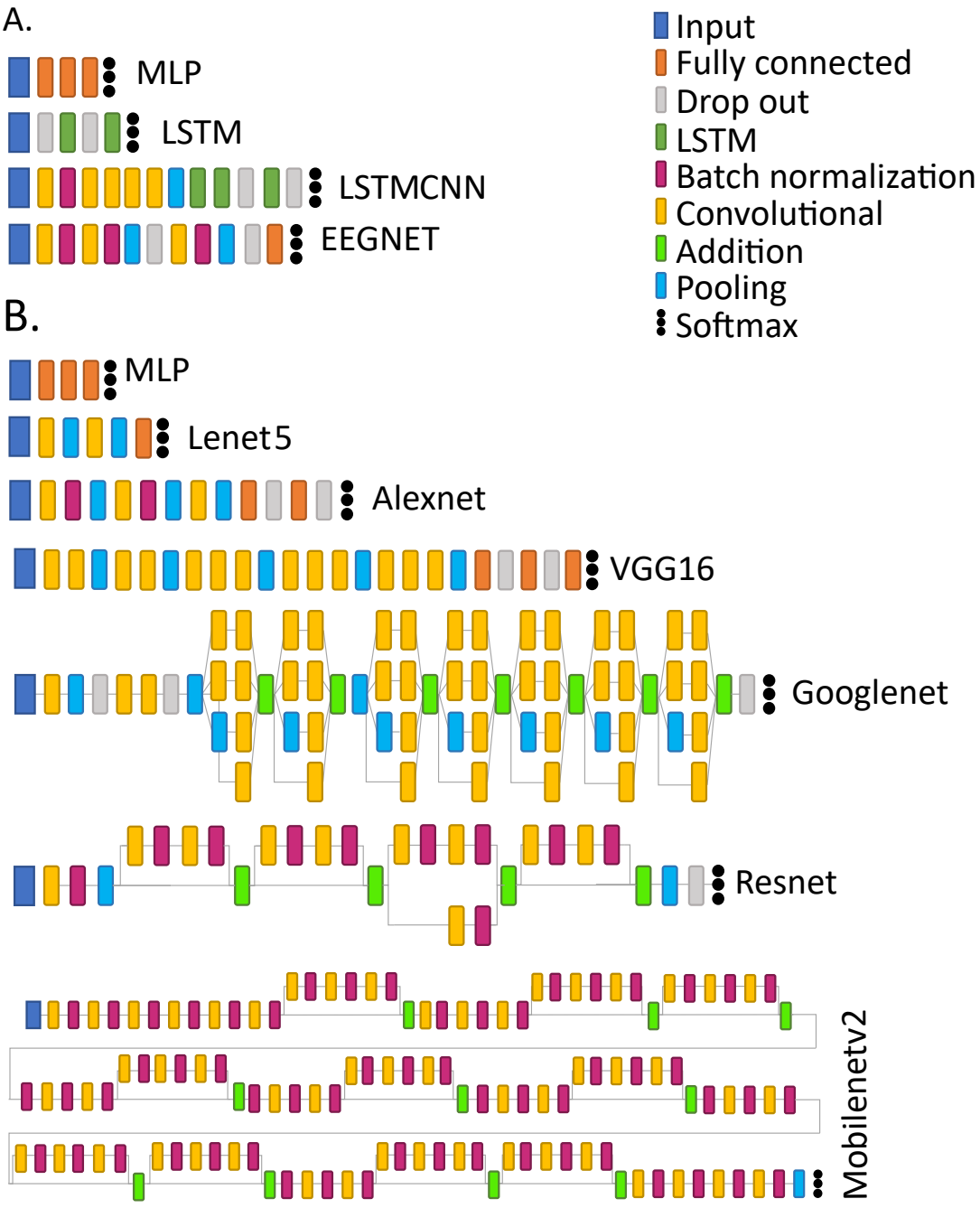

Figure 6.2: Architectures of the temporal models (A): MLP, LSTM, LSTM CNN and EEGNET; the spatial models (B): MLP, Lenet-5, Alexnet, VGG16, Googlenet, and Resnet.

networks are designed to handle sequence dependence (see Section 2.2.1, therefore are widely used in time series problems, despite that, they can struggle with long-time dependencies which can be an issue as our input is of 500 steps. Lastly, LSTM-CNN was chosen as it is an improvement over the LSTM network, as higher-level features are fed to the LSTM layers, but the higher data requirements may lead to underfitting due to our training data size.

6.2.1.2 Models for Spatial Domain

Brain connectivity is affected by AD [177] thus, it is important to look at the relationship between different structures through spatial maps. The different extracted spatial maps can be considered images of 24 by 24 pixels. For this task, CNNs are the prime candidate. The CNNs are a specific form of NN that is well suited to computer vision applications due to their capacity to hierarchically abstract representations of spatial operations.

Adaptations of different popular architectures such as Lenet-5, Alexnet, VGG16, Googlenet, Mobilenetv2 and Resnet were explored [178]. Lenet-5 was selected as it is a small-parameter CNN, while Alexnet and VGG16 are deeper CNNs, Googlenet introduces the inception module with multi-level convolutions, Resnet due to its skipped connections, and lastly Mobilenetv2 becasue it incorporates skipped connections together with depthwise convolution. Overall, these networks have architectures that differ as to how they extract richer features for accurate classification. Networks were adjusted to work with an input map of 24 x 24, including reducing the number of filters and filter size and reducing the fully connected layer's neurons. In those where the repeated number of pooling layers limits the input (i.e. Googlenet and Resnet), iterative modules were removed. To understand the impact of network connectivity, Grad-CAM heatmaps (see Section 2.2.5) were extracted and averaged across all examples for each class of the test set. The layers of these networks are described in Figure 6.2 (B). The optimisation algorithm used was Adam, with an initial learning rate of 0.001, the momentum of 0.9 and a batch size of 256 for all temporal and spatial models.

6.2.1.3 Models for Spectral Domain

Unlike the temporal model, which took time series as inputs, and the spatial model, which took images as inputs, for the spectral models, the input is individual features. While a large range of methods is available for this task, explainable models were selected. Classification and regression trees [33] can be used to build both classifications and regression trees, where each internal node has exactly two outgoing edges, namely binary trees. Binary decision trees are easier to interpret than other ML models, as the decision process in the tree is easily apparent and mimics the rule-based systems medical diagnosis. Decision trees are ML methods in which the decision process is both local and global and integrated into the methods.

The performance of decision trees can be further improved by aggregating models, namely bagging and boosting [34]. For boosting the multi-class version of Adaboost, AdaBoost.M2 was utilised [179]. Ensemble classifiers are used for the spectral models, and based on the best-performing one across the different areas, the performance of the best classifier for different permutations of the classifications was explored. Furthermore, the predictor importance of features can be calculated by summing changes in the node risk due to splits on every predictor and then dividing the sum by the total number of branch nodes. This allows gaining insight into the selection of markers.

6.2.2 Ensemble Model and Late Feature Fusion

In areas such as neuroimaging, the fusion of different modalities to improve the understanding of pathologies and their diagnosis is widely used. The use of feature fusion across the best-performing models in order to improve the classification results was proposed, as it has been successfully applied to non-invasive neuronal recordings [180]. As it is important to maintain insight into the different modalities provided by the individual models, late fusion models are proposed.

Denoting P_m the confidence (or probability) score yielded by classification models $\mathrm{ML}_m, (m = 1, ..., n)$, where $n = 3$, ML_1 is the temporal model, ML_2 comes from the spatial model, and ML_3 refers to the spectral model. Four fusion rules are

considered: average fusion, maximum fusion, minimum fusion, and Naive-product fusion. The average rule simply calculate the simple mean of the ML-classifiers outputs.

$$fmean = \frac{1}{n}\sum_{m=1}^{n} P_m \tag{6.1}$$

The maximum rule outputs the maximum value over the classifier responses,

$$fmax = max(P_m) \tag{6.2}$$

while the minimum rule is

$$fmin = min(P_m) \tag{6.3}$$

Assuming classifiers' independence given the different domains, the Naive-product rule is expressed.

$$fNprod = \frac{\prod_{m=1}^{n} P_m}{\prod_{m=1}^{n} P_m + \prod_{m=1}^{n}(1 - P_m)} \tag{6.4}$$

6.2.3 Robustness to Artefacted Channels

Lastly, the performance of each domain's model and the ensemble model is evaluated by increasingly removing random channels to find the optimal model. The process is described in Algorithm 2, where given a matrix $M[C, T, N]$ of the neuronal recording, $C = 1, ..., 24$ are the total number of channels, $T = 1, ..., 500$ are the data points, $N = 1, ..., 20328$ are the number of examples, $L = 1, ..., 12$ are the channels are selected for occlusion in each iteration, *randi(a,b,c)* is the built-in Matlab random integer generation of a matrix of b by c numbers ranging from 1 to a, and the final result OcM is a 4D matrix. The selected channels are replaced by the average of the remaining ones to maintain spatial cohesion.

Algorithm 2 Random Channel Occlusion Algorithm

Input: M, L
Output: OcM

Initialisation :

$M[C, T, N]$ = sample matrix
C = total number of channels
T = data points
N = number of samples
L = maximum replaced channels, $C/2$ in our case
for $l := 1$ to L **do**
  $OcM[:, :, :, l] = M$
  for $n := 1$ to N **do**
    Oc =randi$[C, l, 1]$
    remaining_ch= $OcM[:, :, n, l]$
    remaining_ch$[Oc, :, n, l] = [\ ]$;
    mean_remaining_ch=mean[remaining_ch,1]
    for $o := 1$ to l **do**
      $OcM[OC[o], T, n, l]$ =mean_remaining_ch
    end for
  end for
end for
return OcM

6.3 Results and Discussion

This Section presents the performances achieved by the different ML models, the performances of the various ensemble models, and lastly, how the best-performing individual and ensemble models perform in the presence of stress segments and artefacted channels.

6.3.1 Performance of Machine Learning Models

The models' performances for each domain are shown in the subsequent sections.

6.3.1.1 Models for Temporal Domain

Results of the temporal models are shown in Table 6.1, and the confucion matrices in Figure 6.3. The MLP has been able to identify WT at over 99%, but its

Table 6.1: Performance of temporal models on the test set.

Model	acc (test)	sen (wt)	sen (2t)	sen (1t)	F1 score	AUROC
MLP	82.4	99.1	70.0	86.1	82.35	98.94
LSTM	58.2	69.2	50.4	66.4	59.18	98.08
LSTMCNN	78.1	96.7	62.1	97.3	78.09	98.08
EEGNET	**86.3**	**97.6**	**79.2**	**84.4**	**86.3**	**98.95**

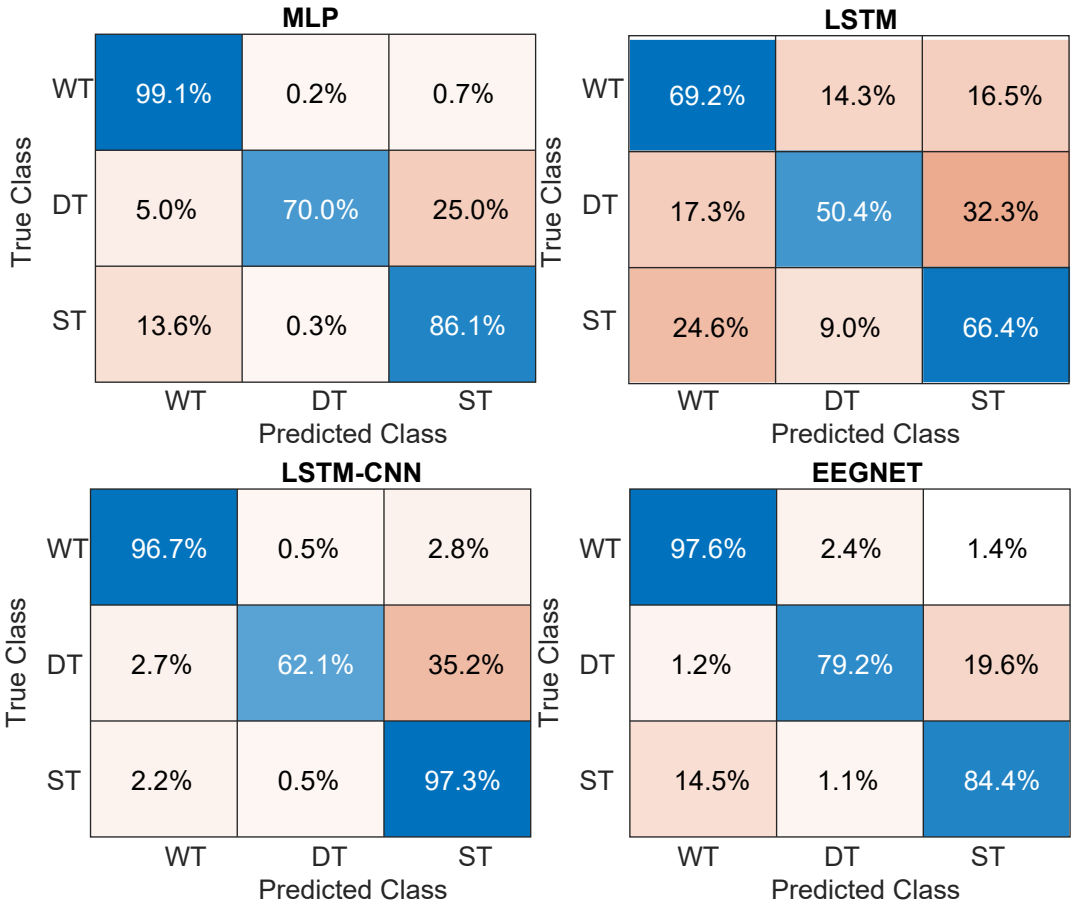

Figure 6.3: Confusion matrices of the temporal models on the test set.

sensitivity for DT and ST is much lower. The LSTM model classifies all three classes worse than the MLP, with the best sensitivity being the WT at 69.2 %. The LSTM-CNN does achieve a high sensitivity of the WT and ST, but at the cost of a lower sentivity of DT. Lastly, EEGNET achieves the best DT sensitivity while maintaining a high WT and ST sensitivity. Across all models, the most

Table 6.2: Performance of spatial models on the test set.

Function	Model	acc.	sen. (WT)	sen. (DT)	sen. (ST)
Corr	MLP	84.0	99.7	100.0	51.5
	LeNet	86.1	92.0	94.0	71.9
	AlexNet	97.3	99.8	100.0	91.9
	VGG16	98.3	98.8	98.1	98.8
	ResNet	87.5	99.7	99.9	62.4
	MobileNetv2	98.1	98.8	99.8	95.7
	GoogLeNet	**99.1**	**99.0**	**98.9**	**99.4**
MI	MLP	90.5	99.8	99.8	71.5
	LeNet	77.5	90	93.5	48.3
	AlexNet	85.2	98.4	99.5	57.2
	VGG16	70.8	79.9	96.3	35.7
	ResNet	84	99.4	99.3	52.5
	MobileNetv2	85	99.1	99.8	55.3
	GoogLeNet	98.2	97.7	97.9	99.1
DTW	MLP	85.6	99	99.9	57.1
	LeNet	88.8	98.3	99.2	68.4
	AlexNet	90.3	99.7	99.6	71.1
	VGG16	99.4	99.1	99.8	99.4
	ResNet	83.8	99.6	99.9	51.1
	MobileNetv2	88.3	99	99.6	65.8
	GoogLeNet	98.9	98.5	99.9	98.3

common miss-classification is of DT into ST, followed by identifying ST as WT. Looking at the F1 score and the AUROC, the EEGNET achieves the highest scores, with the MLP also achieving good performance, however the former miss-classifies ST from DT at a much lower rate. With these results, the EEGNET is chosen as the best temporal model.

6.3.1.2 Models for Spatial Domain

The compilation of performance of the NNs in the test set is compiled in Table 6.2. Out of the correlation connectivity maps models, the MLP, LeNet, and ResNet achieved the worst performances due to the poor sensitivity of the ST. Conversely, Alexnet, Mobilenetv2, VGG16, and GoogLenet achieved the best scores with over 97%. For MI connectivity maps, most models could not accurately identify ST,

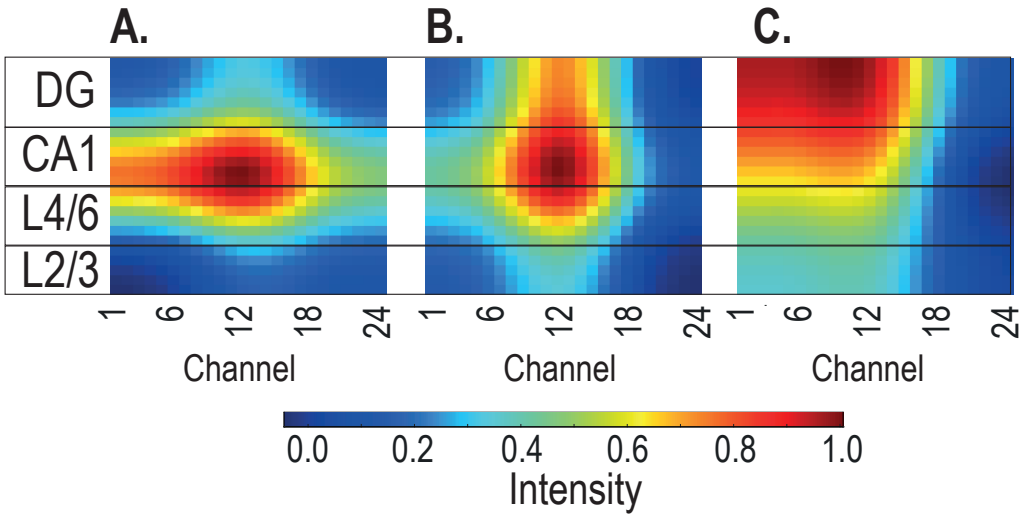

Figure 6.4: Averaged Grad-CAM map for WT (A), DT (B) and ST (C), across the brain areas: dentate gyrus (DG), first region in the hippocampal circuit (CA1), cortex layers 4 to 6 (L4/6), and cortex layers 2 to 3 (L2/3)).

averaging less than 50% sensitivity for Lenet, AlexNet, VGG16, ResNet, and MobileNetv2. On the other hand, MLP and Googlenet achieved scores of 71.5% and 99.6%, respectively. Lastly, for DTW connectivity maps models, the best scores are achieved by GoogLenet, followed closely by VGG16. The remainder of the models have a lower accuracy due to a lower ST sensitivity, ranging from 51.1% of the ResNet upto 71.1% of the AlexNet.

Overall, spatial correlation maps perform better than the MI and the DTW spatial maps. While all networks achieved good sensitivity for the WT and DT classes, most struggled to distinguish ST, which was miss-classified as DT. Nonetheless, the GoogLenet model was the only model to achieve good performance across all spatial maps consistently. Subsequently, the latter is utilised to extract the average grad cam for each class in the test set, which is shown in Figure 6.4. The Grad-CAMs (A) and (B) reveal that the model focuses on the local connectivity of the CA1 area for both WT and DT. Similarly, Grad-CAM (C) reveals that for the ST, the CNN looks at the connectivity of DG with CA1. This correlates with where the AD manifests the most in early stages, i.e. the hippocampus [181], whereas, at this stage, the connectivity between the cortex and it has not been significantly affected, so it

Table 6.3: Performance of ensemble tree per area on the test set.

Ensemble	Area	accuracy	sen. (WT)	sen. (DT)	sen. (ST)
Bagged	DG	84.9	83.9	87.1	83.9
	CA1	81.4	80.4	82.8	81.1
	L4/6	85.4	83.8	86.1	86.4
	L2/3	81.1	81.2	82.1	79.9
	ALL	**84.7**	**82.6**	**88.8**	**82.9**
Adaboost	DG	75.7	69.0	77.0	81.0
	CA1	74.4	66.5	79.0	77.6
	L4/6	76.4	68.6	78.5	81.7
	L2/3	71.9	67.7	75.1	73.0
	ALL	75.7	69.9	79.4	77.7

is not a marker for the network.

6.3.1.3 Models for Spectral Domain

The results achieved on the test set by the ensemble tree models per area are compiled in Table 6.3. Overall, the bagged models outperform the Adaboost models. This indicates that a single tree possesses high variance and that by bagging, the stability of the model is improved, which suits the data better instead of improving the complexity of the model, i.e. boosting. There isn't a major drop in the classifier's performance for the bagged trees across the different areas. The best sensitivity for WT is obtained in the DG, and the same with DT, while for the ST is at L4/6. In the case of Adaboost, the best sensitivity for WT and DT is across all, while for the ST is at L4/6 as well.

With these results in mind, the different classification combinations for the bagged tree model are explored further, as shown in Table 6.4. The model works best in binary classifications, especially WT vs DT, with accuracy over 94%; however, the performance drops 5-10% when doing a three-way classification. This means that the distribution of the biomarkers for differentiating the ST AD model may overlap more with the DT or WT distributions.

The feature importance of the different three-class bagged models are listed in Table 6.5. The models emphasise higher frequency features, such as absolute and relative fast oscillations, followed by absolute high and low gamma. On the other

Table 6.4: Performance of bagged trees for different classification sets on the test set.

Classification	Area	acc.	sen. (WT)	sen. (DT)	sen. (ST)
WT VS ST VS DT	DG	84.9	83.9	87.1	83.9
	CA1	81.4	80.4	82.8	81.1
	L4/6	85.4	83.8	86.1	86.4
	L2/3	81.1	81.2	82.1	79.9
	ALL	84.7	82.6	88.8	82.9
WT VS DT	DG	94.3	92.8	95.8	
	CA1	91.3	90.5	92.0	
	L4/6	93.5	92.8	94.3	
	L2/3	92.2	89.9	94.5	
	ALL	94.2	93.3	95.1	
WT VS ST	DG	90.6	87.8		93.5
	CA1	89.3	88.0		90.7
	L4/6	90.8	88.0		93.6
	L2/3	88.9	85.1		92.7
	ALL	89.3	87.8		90.9
ST VS DT	DG	89.4		89.4	89.4
	CA1	88.8		88.1	89.5
	L4/6	88.6		88.9	88.3
	L2/3	86.1		85.9	86.3
	ALL	89.6		90.8	88.4

hand, low frequencies such as absolute and relative slow oscillations are ranked last. Both features, which are ratios, are also ranked low, indicating they are not markers for the ML model.

Neural network activity is aberrantly increased in AD patients, and animal models due to functional deficits in and decreased activity of GABA inhibitory interneurons [182]. In particular, Kalemaki et al. [183] found that loss of GABAergic inhibition produces significant reduction and disorganisation of the gamma frequency range (30–80 Hz) with an aberrant peak at a high gamma frequency range (80–150 Hz), which translate to our high gamma (45-90 Hz) and fast oscillations (90-125 Hz) ranges. This correlates with the model finding these as discriminative features. The role of these bands is that the gamma synchrony across the hippocampus plays a central role in the coordinated

Table 6.5: Feature ranking per area of the spectral model.

Feature	Ranking per Area				
	DG	CA1	L4/6	L2/3	ALL
fast oscillations	1	1	1	1	1
relative fast oscillations	2	2	2	2	3
high gamma	3	3	3	3	2
relative high gamma	7	4	7	8	8
low gamma	4	5	8	7	4
relative low gamma	8	6	5	5	7
beta	6	7	6	4	6
relative beta	5	8	4	6	5
theta	9	9	11	9	9
relative theta	13	13	13	12	13
delta	10	10	9	11	10
relative delta	11	11	10	10	11
slow oscillations	15	14	15	15	14
relative slow oscillations	16	16	16	16	16
slow oscillations delta ratio	14	15	14	14	15
low high ratio	12	12	12	13	12

reactivation of stored memories [184].

6.3.2 Performance of Ensemble Model and Late Feature Fusion

Table 6.6: Performance of the feature fusion classifiers for the temporal and spectral models.

Model	accuracy	sen. (WT)	sen. (DT)	sen. (ST)
temporal	86.3	97.6	79.2	84.4
spectral	84.7	82.6	88.8	82.9
mean fusion	95.6	99.7	99.5	88.7
max fusion	**95.6**	**99.7**	**99.4**	**88.9**
min fusion	95	99.7	99.6	87.4
nav. prod. fusion.	95.5	99.7	99.6	88.5

As the spatial models have achieved very accurate classification results, first we look at the combinations of the temporal and spectral models in Table 6.6.

Regarding the individual models, the best-performing one is the spatial model, meaning it has the most discriminative features. This correlates with the fact that the best temporal model is EEGNET because it actually exploits spatial properties due to depth-wise convolutions embedded in the architecture. Lastly, the spectral model has a higher sensitivity to the ST than the temporal model but a lesser overall performance. Out of the four fusions, the max fusion achieves the best accuracy of 95.6% with the highest ST sensitivity of 88.9%. Overall, there is an improvement across the sensitivities of all classes, with the four fusion models achieving an accuracy over 95%.

While a two model solution has achieved an acceptable performance, we are also interested in the spatial biomarkers provided via the third model, thus the feature fusion models for the three domains are listed in Table 6.7. All but the mean fusion improve the performance of the individual classifiers, specifically the sensitivities to the DT and the ST classes, which are improved to above 99%. Out of all of them, the best-performing model is the min fusion. This means that if any of the three models is certain that an example does not belong to a said class, that is the most accurate decision. It is important to highlight that none of these ensemble methods require training, unlike other approaches such as regressors, and still achieve good results. Having obtained the best models for each modality and ensemble, their robustness when the rodent is under stress and when channels are missing due to artefacts is evaluated next.

Table 6.7: Performance of the feature fusion classifiers for temporal, spectral and spatial models.

Model	accuracy	sen. (WT)	sen. (DT)	sen. (ST)
temporal	86.3	97.6	79.2	84.4
spectral	84.7	82.6	88.8	82.9
spatial	99.1	99	98.9	99.4
mean fusion	98.8	97.4	98.9	100
max fusion	99.3	98.8	99.2	100
min fusion	**99.4**	**98.7**	**99.7**	**100**
nav. prod. fusion	99.2	98.4	99.4	100

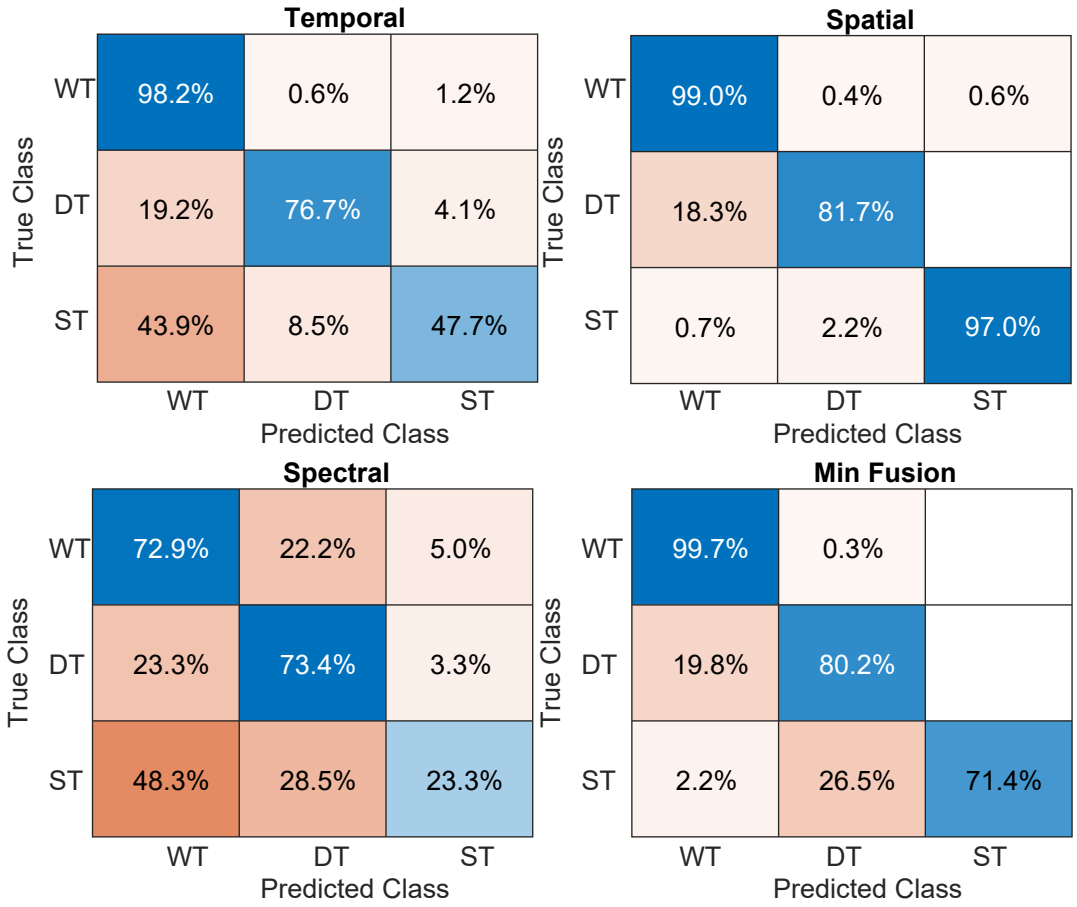

Figure 6.5: Confusion matrices of the stress segments classified by the best models for each domain and the proposed ensemble.

6.3.3 Robustness to Stress Segments

The extracted stress segments for each rodent type were classified by the best-performing model of each domain and the proposed ensemble model. The results are illustrated in Figure 6.5 in the form of confusion matrices to highlight the type of incorrect classifications made by the models.

In the temporal model, the detection of WT maintains a high performance. On the other hand, the performance of the DT drops due to a miss-classification into WT and ST. Similarly, the ST performance achieves a near-guess level due to the model identifying it as WT. This was expected due to the abnormal waveforms of the signals in the stressed state. The spatial model correctly classifies most

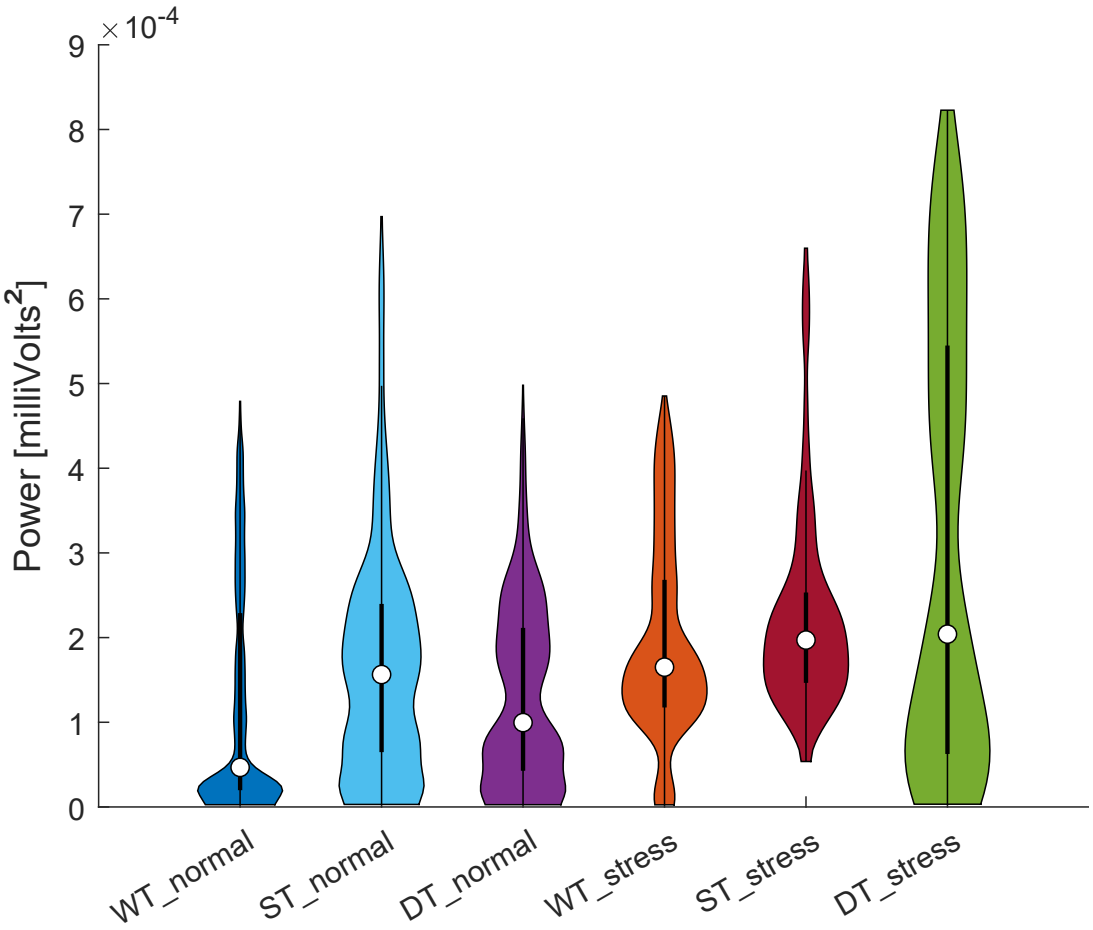

Figure 6.6: Distribution of the power of the fast oscillations band for normal and stressed segments of each class

examples, as the stress does not significantly disturb the network's connectivity. However, there is a decline in the detection of DT.

The spectral model shows the lowest performance out of the three. The sensitivity of the WT is 72.9%, of the DT 73.4% and the ST 23.3%, indicating that the distribution of the markers overlaps more in this state. The WT is mostly miss-classified as DT and vice versa, whereas ST is identified more as WT and DT than correctly as ST. To understand this further, the distributions for each class of the main spectral marker (i.e. fast oscillations) are shown in Figure 6.6. For WT, the distribution has shifted to higher values while maintaining the same range. In the case of ST, the distribution has shifted to

higher values, with the minimum also being higher. Lastly, the DT shows a higher power and a larger range. Overall, there is a higher overlap of the distributions, leading to incorrect classifications.

Finally, the ensemble model achieves a better performance than the temporal and spectral models but poorer performance than the spatial model due to the miss-classification of the ST.

6.3.4 Robustness to Artefacted Channels

The progress of the different performance metrics such as accuracy and class sensitivities as an increasing number of channels is randomly replaced are shown in Figure 6.7. Regarding the accuracy, it can be seen that while the spectral model starts with the lowest performance, it is the least affected one, as all the signals are averaged before feature extraction. On the other hand, the temporal model, which performed better, loses a significant amount of performance, especially in the first 3 removals. The spatial model also suffers from reduced performance but is lesser than the temporal model. Lastly, the min fusion model maintains a high accuracy throughout, with an accuracy over 90% even with 5 channels replaced.

Looking at the class-specific sensitivities, DT stands out as being the least affected of the three, meaning that it has the most discriminative features out of them. On the other hand, ST sensitivity decreases the most in the temporal model and the WT sensitivity in the spatial model. Overall, the spectral model maintains the performance due to what was previously mentioned. The min fusion model maintains high performance for the WT but is affected more than the spectral model for more than 6 replaced channels.

6.4 Conclusion

The work presented in this chapter aimed to develop a methodology that allows understanding AD via the discovery of biomarkers, for which a robust model was needed. A multimodal approach was used to label and structure the neuronal recordings, which subsequently had different features extracted from each domain as input for the classification models. Different combinations of ML models,

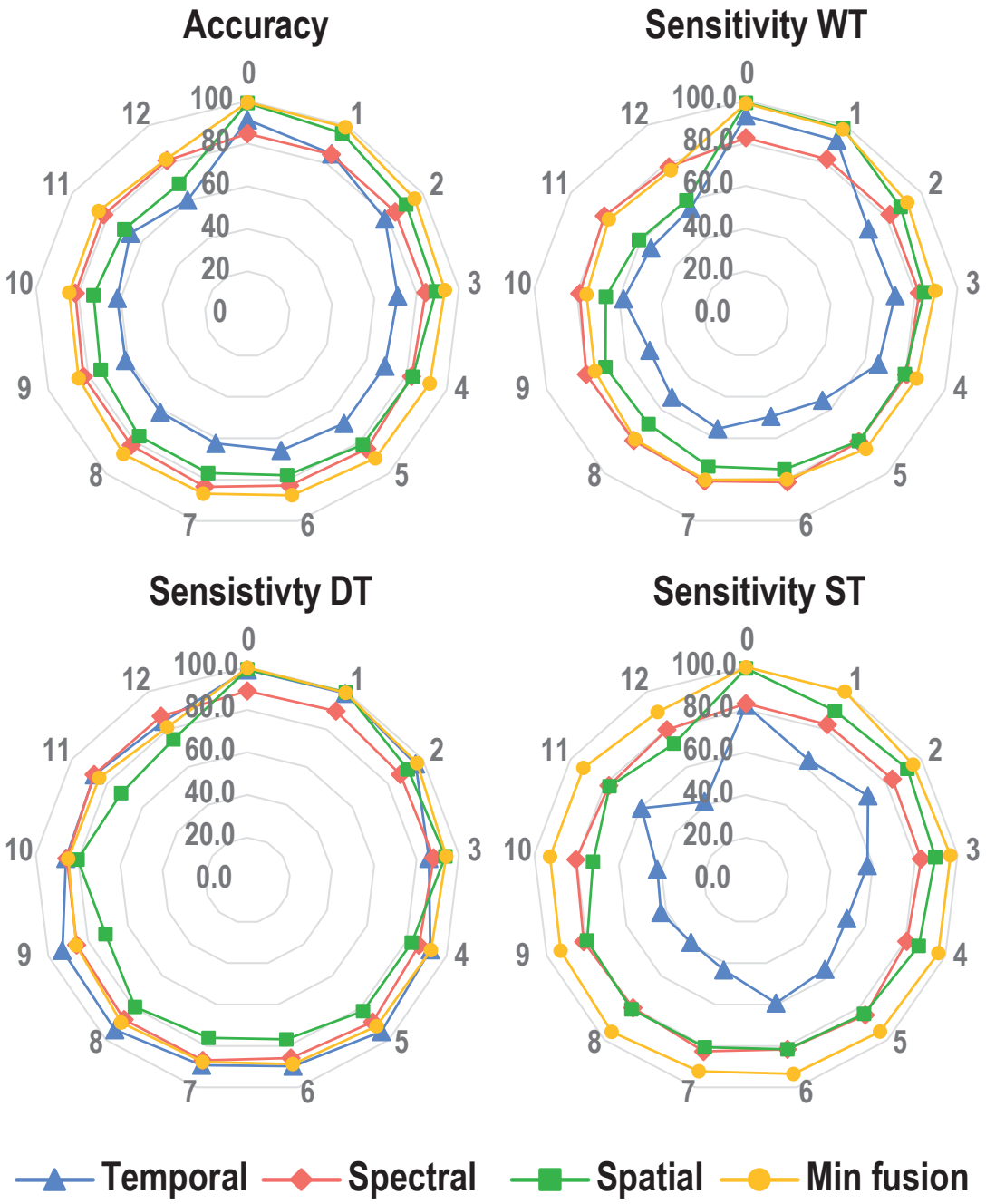

Figure 6.7: Performance of the individual and ensemble models based on the number of replaced channels.

diverse NN architectures, and boosting methods were evaluated for each domain to find the best model for each one. Subsequently, various fusion techniques were

explored to boost the overall performance. The final model achieved an accuracy of 99.4 % over the test set.

After analysing the insights provided by the individual models, it was discovered that the higher frequency bands, such as high gamma and fast oscillations, play a key role in the spectral domain. From the spectral models, it was validated that the hippocampus has an important role in AD detection. To evaluate the robustness of the models to segments where the rodents felt stressed, examples of these segments were shown to the models, where the classification performance dropped significantly. Results showed that the changes in the spectral properties of the signal due to stress were detrimental to the correct identification of AD, highlighting the importance of removing them during pre-processing. Subsequently, their performance was tested by randomly replacing channels in each example with up to half of the available ones. The ensemble's performance remained consistently satisfactory even with five channels replaced.

In human subjects, Van Deursen et al.[185] found that gamma band power recorded via EEG was elevated in 15 AD patients compared to both mild cognitive impairment patients (20 subjects. and the control group (20 subjects), proposing it could be used as a reliable biomarker. The role of the hippocampus as a spatial biomarker can be translated to human patients via medical imaging techniques such as functional MRI [186], positron emission tomography (PET) scans [187] and computed tomography [188], that would allow comparing the mass, volume, or blood oxygenation level of the hippocampus to detect AD.

Overall, the proposed pipeline has proven to be both accurate and robust, as well as maintaining the ability to provide the feature insights needed for a deeper analysis.

Chapter 7

Conclusion

7.1 Thesis Summary

The work undertaken in this thesis has presented a group of open-source tools for processing and analysis of extracellular neuronal signals for disease monitoring. The motivation behind this work is the need of tools that facilitate the use of ML in chronically recorded invasive signals for diagnosis at an early stage. The efforts in the thesis have been directed to the processing of the signal needed for the classification model to have a stable input, in order to obtain unbiased and accurate models that lead to biomarker discovery and validation.

Throughout this work, the research has focused on developing ML-based tools to aid the analysis of neuronal signals, and how these can be made accessible to the research community. In Chapter 4, a performance benchmarking tool of artefact detection and removal methods was presented. Chapter 5 explored ML methods to identify and remove artefacts. Lastly, Chapter 6 studied the classification of animal models of AD via ML. The contributions presented throughout the thesis are showcased in Figure 7.1, which showcases how the processing pipeline presented in Figure 1.3 incorporates them. The presented tools aid researchers in each of the different steps, allowing them to speed up the process and focus instead on interpreting results.

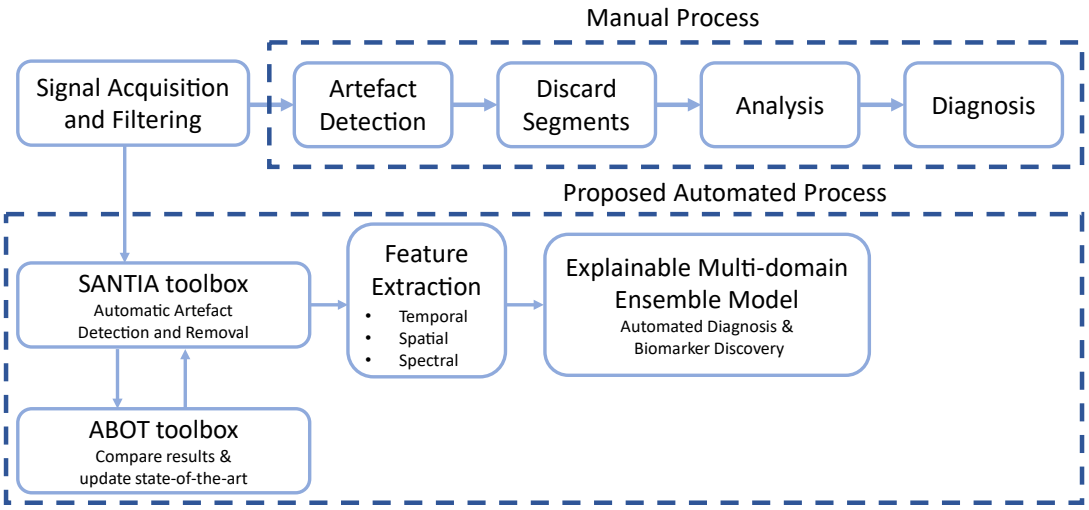

Figure 7.1: Processing pipeline with the thesis contributions.

7.2 Concluding Remarks

The output generated in the thesis shows the utility of developing tools that facilitate disease monitoring in extracellular neuronal signals. The results of the research carried out have led to the creation of several tools that can benefit the neuroscience community. The conclusions for the different contributions made across the thesis are presented in continuation.

7.2.1 Performance Benchmarking Tool

The overview of the literature in Chapter 4 revealed that there was a need for a method to compare the vast literature. Our proposed scale, with four metrics, is a step in that direction. While the selection of the metrics is limited, these were the few consistently reported in the articles. Efforts have been made to make it F.A.I.R., including the collection of the DOIs, making an open access repository, creating an online domain, and the ability to easily contact with suggestions and comments via an online form. In the future, annual or bi-annual updates can be expected, where more advanced architectures may appear, such as transformer networks. While most of the entries are of non-invasive techniques and will continue to increase due to the increasing

accessibility of portable devices, it is hoped that their invasive counterparts will grow too. As shown across this thesis, there are many opportunities to research invasive signals, for which clean signals are needed.

7.2.2 Artefact Detection and Removal

The goal of this project was to find ML solutions for the detection and removal of artefacts in neuronal recordings. In particular, an alternative to discarding a segment when there are artefacts in single-channel recordings or global artefacts present in multi-channel recordings. Results show that neural networks are reliable for both tasks, as they were validated across three datasets. These were incorporated into a toolbox to make them accessible to those without a programming background. Regarding the methods, the detection of artefacts can be improved with the aid of interpretability and explainability. The use of explainable models can help new researchers understand what properties of the signal indicate their presence and even help to identify the origin of the artefacts. By using interpretability, the decision-making of more complex models (usually more accurate) can also be understood. The benefits include feedback on the model by the researchers' review, and the researchers' trust in these systems is improved.

7.2.3 Detection of Alzheimer's Disease at an Early Stage

The correct AD diagnosis can be hard to achieve in the early stages, as it can be confused with other neuropathologies. For this reason, the discovery of biomarkers is crucial to achieving a successful diagnosis and, subsequently, treating the patient to dampen the deteriorating condition. Using invasive recordings in animal models, information about the early stages in the specific brain areas can be studied.

The results showcased in Chapter 6 outline the plausibility of recognising AD in neuronal recordings via ML with the proposed pipeline. In addition to high accuracy of 99.4%, better than previously reported in the literature of 82.6% by Beker et al. [103], it was possible to extract meaningful biomarkers from the

information provided by the models. While the models were sensitive to periods of stress during recording, they showed robustness to absent data.

Three main conclusions can be drawn:

1. ML models can accurately detect AD at an early stage in invasive neuronal recordings.

2. The spatial and spectral domains present rich information which leads to biomarker discovery and validation.

3. The aggregation of the models for each domain is more robust than the individual ones.

Overall, it has been proved that detecting AD at an early stage in invasive neuronal recordings via ML is feasible.

7.3 Contributions

The major contributions of this thesis are summarised as follows:

1. Proposed a benchmarking standard for the performance of artefact detection and removal with ML, which was embedded in an open-access tool comparing over 120 articles.

2. Introduced an innovative artefact removal approach based on normal activity forecasts to replace affected segments.

3. Developed of a toolbox encompassing novel methods for detecting and removing artefacts from extracellular neural signals recorded as LFPs.

4. Proposed a new ML pipeline for detection of AD at an early stage from processed LFP data that is robust to absent data.

5. Extracted new biomarkers from the proposed pipeline and validate them with the known literature.

7.4 Limitations

From the contributions presented in the thesis, the following limitations were found:

- The ABOT toolbox allows users to filter data to find approaches suitable for their application, but it may not be valid for comparing different types of signals, subjects, and artefacts. There are other limitation in the comparison metric, such as not providing metrics for clustering artefacts, which adds complexity to benchmarking. In addition,the normalised scale may miss-represent results, but it provides an approximation of the approach's performance, which can be evaluated further. No other attempts have been made to solve this issue, making it a starting point for future research.
- The SANTIA toolbox does not currently support explainability methods. This may hinder the adoption of the toobox by those looking to identify the nature of the artefacts. However, the toolbox does allow the visualisation of the individual classified segments of the signal with the categorisation of normal or artefact, as an initial step in this direction.
- The explainable pipeline for the detection of detection of AD at an early stage is robust at the expense of computational complexity, as it requires training three distinct ML models. On the aspect of explainability, the temporal model does not provide any insight into the decision making, as a temporal signal with areas highlighted via a Grad-CAM does not provide the same degree of information as a spectral feature or the anatomical area affected of the spatial model.

7.5 Future Work

Following the work undertaken in this thesis, this section outlines the main directions for future work:

- The field of artefact detection and removal continues to grow due to the popularity of portable systems. However, most efforts are directed to non-invasive signals. Further work focused on invasive signals is still needed. Utilising supporting modalities [174] can help in this process, for which pipelines that incorporate them need to be developed. Implementing multiple processing algorithms, each removing a specific artefact to improve the signal's quality through a succession of algorithms, remains an option.

- In the future, the community may achieve a consensus on evaluating the performance of artefact detection and removal methods. This may be in the form of a benchmarking dataset or a standard metric. The work presented in Chapter 4 is the first step in that direction. As a growing field, it is expected that new techniques incorporating state-of-the-art ML, such as transformers or graph neural networks, will appear.

- The entry barrier for ML lowers every year due to its popularity, with tools such as Auto-ML [189]. Nonetheless, some intricacies do require expertise. The SANTIA toolbox presented in Chapter 5 aims to help that while flexible enough to allow a more experienced user to customise it to their needs. Having already had several updates, feedback from the community and planned updates will help improve it.

- A novel pipeline for AD detection at an early stage is proposed in Chapter 6, achieving promising results. The incorporation of new classes into the model, which includes the animal at different stages or ages, should translate into further advances in understanding AD. To achieve further advances in understanding AD, the study of subjects at 6 and 12 months of age should be explored, to observe the progression of the disease and how it affects the subjects and the biomarkers. In other words, the development and evaluation of a multi-class ML model that includes the ages and phenotypes of the subjects. This includes classification tasks of the age of the mice as well as the progress of the disease.

- A significant contribution to the field of disease monitoring would be to test

models across several pathologies to achieve a multi-purpose model. While creating such a dataset is unlikely by a single team, the efforts of multiple sources could make it happen via open-access data repositories [130].

- With the growth of the use of ML models, the need for interpretable and explainable models does as well. In the future, we can expect further development of explainable methods that complement black-box models, especially in health care, where physicians rely on these systems to complement their diagnoses [190]. Through explainability and interpretability, the accuracy of the diagnosis can improve and the models' performance as well [191].

- Lastly, to bridge the gap between animal models and humans, the discovered biomarkers need to be mapped to the cortex. In this manner, diagnosis via non-invasive and portable systems can be used, facilitating the diagnosis for physicians and thus improving the patients' prognosis.

References

[1] M. Mahmud and S. Vassanelli, "Processing and analysis of multichannel extracellular neuronal signals: State-of-the-art and challenges", *Frontiers in Neuroscience*, vol. 10, no. JUN, pp. 248, 2016. xxi, 49, 51

[2] Neurological Alliance, "Neuro numbers 2019: a report by neurological alliance", 2019. 1

[3] Tomislav Milekovic, Anish A Sarma, Daniel Bacher, John D Simeral, Jad Saab, Chethan Pandarinath, Brittany L Sorice, Christine Blabe, Erin M Oakley, Kathryn R Tringale, et al., "Stable long-term bci-enabled communication in als and locked-in syndrome using lfp signals", *Journal of neurophysiology*, vol. 120, no. 7, pp. 343–360, 2018. 2

[4] Banghua Yang, Tao Zhang, Yunyuan Zhang, Wanquan Liu, Jianguo Wang, and Kaiwen Duan, "Removal of electrooculogram artifacts from electroencephalogram using canonical correlation analysis with ensemble empirical mode decomposition", *Cognitive Computation*, vol. 9, no. 5, pp. 626–633, 2017. 4

[5] Mehrdad Fatourechi, Ali Bashashati, Rabab K Ward, and Gary E Birch, "Emg and eog artifacts in brain computer interface systems: A survey", *Clinical neurophysiology*, vol. 118, no. 3, pp. 480–494, 2007. 4

[6] Weidong Zhou and Jean Gotman, "Removal of emg and ecg artifacts from eeg based on wavelet transform and ica", in *Proc. EMBC*, 2004, vol. 1, pp. 392–395. 4

[7] Kianoush Banaie Boroujeni, Paul Tiesinga, and Thilo Womelsdorf, "Adaptive spike-artifact removal from local field potentials uncovers

prominent beta and gamma band neuronal synchronization", *Journal of Neuroscience Methods*, vol. 330, pp. 108485, 2020. 4

[8] Roberto Fontana, Mario Agostini, Emanuele Murana, Mufti Mahmud, Elena Scremin, Maria Rubega, Giovanni Sparacino, Stefano Vassanelli, and Cristina Fasolato, "Early hippocampal hyperexcitability in ps2app mice: role of mutant ps2 and app", *Neurobiology of aging*, vol. 50, pp. 64–76, 2017. 4

[9] Adriano BL Tort, Simon Ponsel, Jakob Jessberger, Yevgenij Yanovsky, Jurij Brankačk, and Andreas Draguhn, "Parallel detection of theta and respiration-coupled oscillations throughout the mouse brain", *Scientific reports*, vol. 8, no. 1, pp. 6432, 2018. 4

[10] Alessandro Leparulo, Mufti Mahmud, Elena Scremin, Tullio Pozzan, Stefano Vassanelli, and Cristina Fasolato, "Dampened slow oscillation connectivity anticipates amyloid deposition in the ps2app mouse model of alzheimer's disease", *Cells*, vol. 9, no. 1, pp. 54, 2020. 4, 42, 43

[11] GAA Hudhud and Martin J Turner, "Digital removal of power frequency artifacts using a fourier space median filter", *IEEE Signal Processing Letters*, vol. 12, no. 8, pp. 573–576, 2005. 4

[12] Sanja Mikulovic, Stefano Pupe, Helton Maia Peixoto, George C Do Nascimento, Klas Kullander, Adriano BL Tort, and Richardson N Leão, "On the photovoltaic effect in local field potential recordings", *Neurophotonics*, vol. 3, no. 1, pp. 015002, 2016. 4

[13] M. Mahmud, S. Girardi, M. Maschietto, and S. Vassanelli, "An automated method to remove artifacts induced by microstimulation in local field potentials recorded from rat somatosensory cortex", in *Proc. BRC*, 2012, pp. 1–4. 4

[14] Kenneth D Harris, Rodrigo Quian Quiroga, Jeremy Freeman, and Spencer L Smith, "Improving data quality in neuronal population recordings", *Nature neuroscience*, vol. 19, no. 9, pp. 1165–1174, 2016. 4

[15] Anupratap Tomar, Denis Polygalov, and Thomas J McHugh, "Differential impact of acute and chronic stress on ca1 spatial coding and gamma oscillations", *bioRxiv*, 2021. 5

[16] Ewan Nurse, Benjamin S Mashford, Antonio Jimeno Yepes, Isabell Kiral-Kornek, Stefan Harrer, and Dean R Freestone, "Decoding eeg and lfp signals using deep learning: heading truenorth", in *Proceedings of the ACM international conference on computing frontiers*, 2016, pp. 259–266. 5

[17] Xiao Jiang, Gui-Bin Bian, and Zean Tian, "Removal of artifacts from eeg signals: A review", *Sensors*, vol. 19, pp. 987, 02 2019. 6, 49, 51

[18] Md Kafiul Islam, Amir Rastegarnia, and Zhi Yang, "Methods for artifact detection and removal from scalp eeg: A review", *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 46, no. 4-5, pp. 287–305, 2016. 6, 25, 29, 35

[19] Khansa Rasheed, Adnan Qayyum, Mohammed Ghaly, Ala Al-Fuqaha, Adeel Razi, and Junaid Qadir, "Explainable, trustworthy, and ethical machine learning for healthcare: A survey", *Computers in Biology and Medicine*, p. 106043, 2022. 6

[20] Johannes Allgaier, Lena Mulansky, Rachel Lea Draelos, and Rüdiger Pryss, "How does the model make predictions? a systematic literature review on the explainability power of machine learning in healthcare", *Artificial Intelligence in Medicine*, vol. 143, pp. 102616, 2023. 6

[21] Scott J Webster, Adam D Bachstetter, Peter T Nelson, Frederick A Schmitt, and Linda J Van Eldik, "Using mice to model alzheimer's dementia: an overview of the clinical disease and the preclinical behavioral changes in 10 mouse models", *Frontiers in genetics*, vol. 5, pp. 88, 2014. 6

[22] Bart Laurijssens, Fabienne Aujard, and Anisur Rahman, "Animal models of alzheimer's disease and drug development", *Drug Discovery Today: Technologies*, vol. 10, no. 3, pp. e319–e327, 2013. 6

[23] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal, "Deep learning for electroencephalogram (eeg) classification tasks: a review", *Journal of neural engineering*, vol. 16, no. 3, pp. 031001, 2019. 13, 63

[24] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert, "Deep learning-based electroencephalography analysis: a systematic review", *Journal of neural engineering*, 2019. 13, 63

[25] AD Dongare, RR Kharde, Amit D Kachare, et al., "Introduction to artificial neural network", *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012. 14

[26] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadlamani Ravi, and Alan Peters, "A review of deep learning with special emphasis on architectures, applications and recent trends", *Knowledge-Based Systems*, vol. 194, pp. 105596, 2020. 15

[27] L. N. Kanal, "Perceptrons", in *International Encyclopedia of the Social & Behavioral Sciences*, Neil J. Smelser and Paul B. Baltes, Eds., pp. 11218–11221. Pergamon, Oxford, Jan. 2001. 16

[28] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al., "Recent advances in convolutional neural networks", *Pattern Recognition*, vol. 77, pp. 354–377, 2018. 17

[29] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Atiah, Vadlamani Ravi, and Alan Peters, "A review of deep learning with special emphasis on architectures, applications and recent trends", *Knowledge-Based Systems*, p. 105596, 2020. 18

[30] Vladimir Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013. 19

[31] Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets", *SpringerPlus*, vol. 5, no. 1, pp. 1–9, 2016. 20

[32] Zhongheng Zhang, "Introduction to machine learning: k-nearest neighbors", *Annals of translational medicine*, vol. 4, no. 11, 2016. 20

[33] Richard A Berk, "Classification and regression trees (cart)", in *Statistical learning from a regression perspective*, pp. 1–65. Springer, 2008. 20, 112

[34] Thomas G Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000. 21, 112

[35] Ajay Kumar Maddirala and Kalyana C Veluvolu, "Eye-blink artifact removal from single channel eeg with k-means and ssa", *Scientific Reports*, vol. 11, no. 1, pp. 11043, 2021. 21

[36] Nicoletta Nicolaou and Slawomir J Nasuto, "Automatic artefact removal from event-related potentials via clustering", *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 48, pp. 173–183, 2007. 21, 62

[37] Chin-Teng Lin, Chih-Sheng Huang, Wen-Yu Yang, Avinash Kumar Singh, Chun-Hsiang Chuang, Yu-Kai Wang, et al., "Real-time eeg signal enhancement using canonical correlation analysis and gaussian mixture clustering", *Journal of healthcare engineering*, vol. 2018, 2018. 21

[38] Arpita Nagpal, Arnan Jatain, and Deepti Gaur, "Review based on data clustering algorithms", in *2013 IEEE conference on information & communication technologies*. IEEE, 2013, pp. 298–303. 21

[39] M Gopal, *Applied Machine Learning*, McGraw-Hill Education, Columbus, OH, May 2019. 22

[40] Guy S Handelman, Hong Kuan Kok, Ronil V Chandra, Amir H Razavi, Shiwei Huang, Mark Brooks, Michael J Lee, and Hamed Asadi, "Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods", *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, 2019. 22

[41] Robert Layton, Paul Watters, and Richard Dazeley, "Evaluating authorship distance methods using the positive silhouette coefficient", *Natural Language Engineering*, vol. 19, no. 4, pp. 517–535, 2013. 23

[42] Tanvi Gupta and Supriya P Panda, "Clustering validation of clara and k-means using silhouette & dunn measures on iris dataset", in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 10–13. 24

[43] Mengyao Cui et al., "Introduction to the k-means clustering algorithm based on the elbow method", *Accounting, Auditing and Finance*, vol. 1, no. 1, pp. 5–8, 2020. 24

[44] Sebastian Raschka, "Model evaluation, model selection, and algorithm selection in machine learning", *arXiv preprint arXiv:1811.12808*, 2018. 24

[45] Sherin Mary Mathews, "Explainable artificial intelligence applications in nlp, biomedical, and malware classification: A literature review", in *Intelligent Computing-Proceedings of the Computing Conference*. Springer, 2019, pp. 1269–1292. 24

[46] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626. 25, 63

[47] Luke Merrick and Ankur Taly, "The explanation game: Explaining machine learning models using shapley values", in *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*. Springer, 2020, pp. 17–38. 25, 63

[48] Stephan Waldert, "Invasive vs. non-invasive neuronal signals for brain-machine interfaces: will one prevail?", *Frontiers in neuroscience*, vol. 10, pp. 295, 2016. 25

[49] Sankaraleengam Alagapan, Hae Won Shin, Flavio Fröhlich, and Hautieng Wu, "Diffusion geometry approach to efficiently remove electrical stimulation artifacts in intracranial electroencephalography", *Journal of neural engineering*, vol. 16, no. 3, pp. 036010, 2019. 26

[50] Viateur Tuyisenge, Lena Trebaul, Manik Bhattacharjee, Blandine Chanteloup-Forêt, Carole Saubat-Guigui, Ioana Mîndruţă, Sylvain Rheims, Louis Maillard, Philippe Kahane, Delphine Taussig, et al., "Automatic bad channel detection in intracranial electroencephalographic recordings using ensemble machine learning", *Clinical Neurophysiology*, vol. 129, no. 3, pp. 548–554, 2018. 26

[51] Petr Nejedly, Jan Cimbalnik, Petr Klimes, Filip Plesinger, Josef Halamek, Vaclav Kremen, Ivo Viscor, Benjamin H Brinkmann, Martin Pail, Milan Brazdil, et al., "Intracerebral eeg artifact identification using convolutional neural networks", *Neuroinformatics*, vol. 17, no. 2, pp. 225–234, 2019. 26

[52] Mengmeng Li, You Liang, Lifang Yang, Haofeng Wang, Zhongliang Yang, Kun Zhao, Zhigang Shang, and Hong Wan, "Automatic bad channel detection in implantable brain-computer interfaces using multimodal features based on local field potentials and spike signals", *Computers in biology and medicine*, vol. 116, pp. 103572, 2020. 27

[53] Olga Bukhtiyarova, Sara Soltani, Sylvain Chauvette, and Igor Timofeev, "Supervised semi-automatic detection of slow waves in non-anaesthetized mice with the use of neural network approach", *Transl. Brain Rhythmicity*, vol. 1, no. 1, pp. 14–18, 2016. 27

[54] Ondřej Klempíř, Radim Krupička, Eduard Bakštein, and Robert Jech, "Identification of microrecording artifacts with wavelet analysis and convolutional neural network: an image recognition approach", *Measurement Science Review*, vol. 19, no. 5, pp. 222–231, 2019. 27

[55] Mohamed Hosny, Minwei Zhu, Wenpeng Gao, and Yili Fu, "A novel deep lstm network for artifacts detection in microelectrode recordings", *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 1052–1063, 2020. 27

[56] Garrick L Wallstrom, Robert E Kass, Anita Miller, Jeffrey F Cohn, and Nathan A Fox, "Automatic correction of ocular artifacts in the eeg: a comparison of regression-based and component-based methods", *International journal of psychophysiology*, vol. 53, no. 2, pp. 105–119, 2004. 28, 35

[57] Rahul Kher and Riddhish Gandhi, "Adaptive filtering based artifact removal from electroencephalogram (eeg) signals", in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2016, pp. 0561–0564. 28

[58] Abed Khorasani, Vahid Shalchyan, and Mohammad Reza Daliri, "Adaptive artifact removal from intracortical channels for accurate decoding of a force signal in freely moving rats", *Frontiers in neuroscience*, vol. 13, pp. 350, 2019. 28, 35

[59] Jan C de Munck, Petra J van Houdt, Sónia I Gonçalves, Erwin van Wegen, and Pauly PW Ossenblok, "Novel artefact removal algorithms for co-registered eeg/fmri based on selective averaging and subtraction", *Neuroimage*, vol. 64, pp. 407–415, 2013. 28

[60] Xing Qian, Yue Chen, Yuan Feng, Bozhi Ma, Hongwei Hao, and Luming Li, "A method for removal of deep brain stimulation artifact from local field potentials", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 12, pp. 2217–2226, 2016. 28, 35

[61] Nooshin Bahador, Jarno Jokelainen, Seppo Mustola, and Jukka Kortelainen, "Reconstruction of missing channel in electroencephalogram using spatiotemporal correlation-based averaging", *Journal of Neural Engineering*, vol. 18, no. 5, pp. 056045, 2021. 29

[62] Mufti Mahmud, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "An automated method to remove artifacts induced by microstimulation in local field potentials recorded from rat somatosensory cortex", in *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*. IEEE, 2012, pp. 1–4. 29, 35

[63] Giuseppina Inuso, Fabio La Foresta, Nadia Mammone, and Francesco Carlo Morabito, "Wavelet-ica methodology for efficient artifact removal from electroencephalographic recordings", in *2007 international joint conference on neural networks*. IEEE, 2007, pp. 1524–1529. 29, 35

[64] Mats Svantesson, Hakan Olausson, Anders Eklund, and Magnus Thordstein, "Virtual eeg-electrodes: Convolutional neural networks as a method for upsampling or restoring channels", *bioRxiv*, 2020, doi: 10.1101/2020.04.20.049916. 29

[65] Sari Saba-Sadiya, Tuka Alhanai, Taosheng Liu, and Mohammad M Ghassemi, "Eeg channel interpolation using deep encoder-decoder netwoks", *arXiv preprint arXiv:2009.12244*, 2020. 30

[66] Ngoc Anh Nguyen Thi, Hyung-Jeong Yang, and Sun-Hee Kim, "Exploiting patterns for handling incomplete coevolving eeg time series.", *Int. J. Contents*, vol. 9, no. 4, 2013. 30

[67] Alain de Cheveigné and Dorothée Arzounian, "Robust detrending, rereferencing, outlier detection, and inpainting for multichannel data", *Neuroimage*, vol. 172, pp. 903–912, 2018. 30

[68] Chi-Yuan Chang, Sheng-Hsiou Hsu, Luca Pion-Tonachini, and Tzyy-Ping Jung, "Evaluation of artifact subspace reconstruction for automatic artifact components removal in multi-channel eeg recordings", *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 4, pp. 1114–1121, 2019. 30

[69] Jordi Sole-Casals, Cesar Federico Caiafa, Qibin Zhao, and Adrzej Cichocki, "Brain-computer interface with corrupted eeg data: a tensor completion approach", *Cognitive Computation*, vol. 10, no. 6, pp. 1062–1074, 2018. 30

[70] Arnaud Delorme and Scott Makeig, "Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis", *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004. 30

[71] U Egert, Th Knott, C Schwarz, M Nawrot, A Brandt, S Rotter, and M Diesmann, "Mea-tools: an open source toolbox for the analysis of multi-

electrode data with matlab", *Journal of neuroscience methods*, vol. 117, no. 1, pp. 33–42, 2002. 30

[72] Pierre Yger, Giulia LB Spampinato, Elric Esposito, Baptiste Lefebvre, Stéphane Deny, Christophe Gardella, Marcel Stimberg, Florian Jetter, Guenther Zeck, Serge Picaud, et al., "A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo", *Elife*, vol. 7, pp. e34518, 2018. 30

[73] Valentina Andreevna Unakafova and Alexander Gail, "Comparing open-source toolboxes for processing and analysis of spike and local field potentials data", *Frontiers in neuroinformatics*, vol. 13, pp. 57, 2019. 31

[74] François Tadel, Sylvain Baillet, John C Mosher, Dimitrios Pantazis, and Richard M Leahy, "Brainstorm: a user-friendly application for meg/eeg analysis", *Computational intelligence and neuroscience*, vol. 2011, 2011. 31

[75] Jie Cui, Lei Xu, Steven L Bressler, Mingzhou Ding, and Hualou Liang, "Bsmart: a matlab/c toolbox for analysis of multichannel neural time series", *Neural Networks*, vol. 21, no. 8, pp. 1094–1104, 2008. 31

[76] Hemant Bokil, Peter Andrews, Jayant E Kulkarni, Samar Mehta, and Partha P Mitra, "Chronux: a platform for analyzing neural signals", *Journal of neuroscience methods*, vol. 192, no. 1, pp. 146–151, 2010. 31

[77] Alper Yegenoglu et al., "Elephant–open-source tool for the analysis of electrophysiological data sets", in *Proc. Bernstein Conference*, 2015, pp. 134–135. 31, 32

[78] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen, "Fieldtrip: open source software for advanced analysis of meg, eeg, and invasive electrophysiological data", *Computational intelligence and neuroscience*, vol. 2011, 2011. 31, 32

[79] Lynn Hazan, Michaël Zugaro, and György Buzsáki, "Klusters, neuroscope, ndmanager: a free software suite for neurophysiological data processing and visualization", *Journal of neuroscience methods*, vol. 155, no. 2, pp. 207–216, 2006. 31, 32

[80] Samuel Garcia, Domenico Guarino, Florent Jaillet, Todd R Jennings, Robert Pröpper, Philipp L Rautenberg, Chris Rodgers, Andrey Sobolev, Thomas Wachtler, Pierre Yger, et al., "Neo: an object model for handling electrophysiology data in multiple formats", *Frontiers in neuroinformatics*, vol. 8, pp. 10, 2014. 31, 32

[81] Md Nurul Islam, Seán K Martin, John P Aggleton, and Shane M O'Mara, "Neurochat: a toolbox to analyse the dynamics of neuronal encoding in freely-behaving rodents in vivo", *Wellcome Open Research*, vol. 4, 2019. 31, 32

[82] Luca Leonardo Bologna et al., "Investigating neuronal activity by spycode multi-channel data analyzer", *Neural Networks*, vol. 23, no. 6, pp. 685–697, 2010. 31, 32

[83] Mufti Mahmud, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "Sigmate: A matlab-based neuronal signal processing tool", in *Proc. IEEE EMBC*, 2010, pp. 1352–1355. 31, 33, 68

[84] Mufti Mahmud, Stefano Girardi, Marta Maschietto, M Mostafizur Rahman, Alessandra Bertoldo, and Stefano Vassanelli, "Slow stimulus artifact removal through peak-valley detection of neuronal signals recorded from somatosensory cortex by high resolution brain-chip interface", in *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer, 2009, pp. 2062–2065. 33

[85] Mufti Mahmud, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, Elisabetta Pasqualotto, and Stefano Vassanelli, "SigMate: A comprehensive software package for extracellular neuronal signal processing and analysis", in *Proc. NER*, 2011, pp. 88–91, doi: 10.1109/NER.2011.5910495. 33

[86] Mufti Mahmud, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "SigMate: A Matlab-based automated tool for extracellular neuronal signal processing and analysis", *J.*

Neurosci. Methods, vol. 207, no. 1, pp. 97–112, 2012, doi: 10.1016/j.jneumeth.2012.03.009. 33

[87] Mufti Mahmud, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "An automated method to remove artifacts induced by microstimulation in local field potentials recorded from rat somatosensory cortex", in *Proc. BRC*, 2012, pp. 1–4, doi: 10.1109/BRC.2012.6222169. 33

[88] Mufti Mahmud, Stefano Girardi, Marta Maschietto, Elisabetta Pasqualotto, and Stefano Vassanelli, "An automated method to determine angular preferentiality using LFPs recorded from rat barrel cortex by brain-chip interface under mechanical whisker stimulation", in *Proc. EMBC*, 2011, pp. 2307–2310, doi: 10.1109/IEMBS.2011.6090580. 33

[89] Mufti Mahmud, Stefano Girardi, Marta Maschietto, Mohammad Mostafizur Rahman, and Stefano Vassanelli, "Noise characterization of electrophysiological signals recorded from high resolution brain-chip interface", in *Proc. ISBB*, 2009, pp. 84–87. 33

[90] Mufti Mahmud, Alessandra Bertoldo, Marta Maschietto, Stefano Girardi, and Stefano Vassanelli, "Automatic detection of layer activation order in information processing pathways of rat barrel cortex under mechanical whisker Stimulation", in *Proc. EMBC*, 2010, pp. 6095–6098, doi: 10.1109/IEMBS.2010.5627639. 33

[91] Mufti Mahmud, Marta Maschietto, Stefano Girardi, and Stefano Vassanelli, "A Matlab based tool for cortical layer activation order detection through latency calculation in local field potentials recorded from rat barrel cortex by brain-chip interface", in *Proc. BRC*, 2012, pp. 1–4, doi: 10.1109/BRC.2012.6222170. 33

[92] Mufti Mahmud, Elisabetta Pasqualotto, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "An automated method for detection of layer activation order in information processing pathway of rat barrel cortex under mechanical whisker stimulation", *J. Neurosci. Methods*, vol. 196, no. 1, pp. 141–150, 2011, doi: 10.1016/j.jneumeth.2010.11.024. 33

[93] M. Mahmud and et al., "An Automated Method for Characterization of Evoked Single-Trial Local Field Potentials Recorded from Rat Barrel Cortex Under Mechanical Whisker Stimulation", *Cogn. Comput.*, vol. 8, no. 5, pp. 935—-945, 2016, doi: 10.1007/s12559-016-9399-3. 33

[94] Mufti Mahmud, Davide Travalin, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "A contour based automatic method to classify local field potentials recorded from rat barrel cortex", in *Proc. CIBEC*, 2010, pp. 163–166, doi: 10.1109/CIBEC.2010.5716087. 33

[95] Mufti Mahmud, Davide Travalin, Amir Hussain, Stefano Girardi, Marta Maschietto, Florian Felderer, and Stefano Vassanelli, "Single LFP sorting for high-resolution brain-chip interfacing", *Proc. BICS*, vol. 7366 LNAI, pp. 329–337, 2012, doi: 10.1007/978-3-642-31561-9_37. 33

[96] Mufti Mahmud, Davide Travalin, and Amir Hussain, "Decoding network activity from LFPS: A computational approach", *Proc. ICONIP*, vol. 7663 LNCS, pp. 584–591, 2012, doi: 10.1007/978-3-642-34475-6_70. 33

[97] Mufti Mahmud, Davide Travalin, Alessandra Bertoldo, Stefano Girardi, Marta Maschietto, and Stefano Vassanelli, "An automated classification method for single sweep local field potentials recorded from rat barrel cortex under mechanical whisker stimulation", *J. Med. Biol. Eng.*, vol. 32, no. 6, 2012, doi: 10.5405/jmbe.923. 33

[98] Olivier Darbin, Nobuhiko Hatanaka, Sayuki Takara, Masaya Kaneko, Satomi Chiken, Dean Naritoku, Anthony Martino, and Atsushi Nambu, "Local field potential dynamics in the primate cortex in relation to parkinsonism reveled by machine learning: A comparison between the primary motor cortex and the supplementary area", *Neuroscience research*, vol. 156, pp. 66–79, 2020. 34

[99] Joyce Chelangat Bore, Brett A Campbell, Hanbin Cho, Raghavan Gopalakrishnan, Andre G Machado, and Kenneth B Baker, "Prediction of mild parkinsonism revealed by neural oscillatory changes and machine learning", *Journal of Neurophysiology*, vol. 124, no. 6, pp. 1698–1705, 2020. 34

[100] Emilia Toth, Sachin S Kumar, Ganne Chaitanya, Kristen Riley, Karthi Balasubramanian, and Sandipan Pati, "Reliable detection of focal-onset seizures in the human anterior nucleus of the thalamus using non-linear machine learning", *medRxiv*, 2020. 34

[101] Yun S Park, Leigh R Hochberg, Emad N Eskandar, Sydney S Cash, and Wilson Truccolo, "Early detection of human epileptic seizures based on intracortical local field potentials", in *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2013, pp. 323–326. 34

[102] Mehdi Aghagolzadeh, Leigh R Hochberg, Sydney S Cash, and Wilson Truccolo, "Predicting seizures from local field potentials recorded via intracortical microelectrode arrays", in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 6353–6356. 34

[103] Shlomit Beker, Vered Kellner, Gal Chechik, and Edward A Stern, "Learning to classify neural activity from a mouse model of alzheimer's disease amyloidosis versus controls", *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, vol. 2, pp. 39–48, 2016. 34, 129

[104] David Gómez and Alfonso Rojas, "An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification", *Neural computation*, vol. 28, no. 1, pp. 216–228, 2016. 35

[105] Alberto Averna, Valentina Pasquale, Maxwell D. Murphy, Maria Piera Rogantin, Gustaf M. Van Acker, Randolph J. Nudo, Michela Chiappalone, and David J. Guggenmos, "Differential Effects of Open- and Closed-Loop Intracortical Microstimulation on Firing Patterns of Neurons in Distant Cortical Areas", *Cereb Cortex*, vol. 30, no. 5, pp. 2879–2896, May 2020. 38

[106] Katrina Furth, "Replication Data for: Neuronal correlates of ketamine and walking induced gamma oscillations in the medial prefrontal cortex and mediodorsal thalamus", 2017. 39

[107] Katrina E Furth, Alex J McCoy, Caroline Dodge, Judith R Walters, Andres Buonanno, and Claire Delaville, "Neuronal correlates of ketamine and

walking induced gamma oscillations in the medial prefrontal cortex and mediodorsal thalamus", *PLoS One*, vol. 12, no. 11, pp. e0186732, 2017. 39

[108] Jacob G Ellen and Michael B Dash, "An artificial neural network for automated behavioral state classification in rats", *PeerJ*, vol. 9, pp. e12127, 2021. 40

[109] Alessandro Leparulo, Marta Bisio, Nelly Redolfi, Tullio Pozzan, Stefano Vassanelli, and Cristina Fasolato, "Accelerated aging characterizes the early stage of alzheimer's disease", *Cells*, vol. 11, no. 2, pp. 238, 2022. 43

[110] Meinard Müller, "Dynamic time warping", *Information retrieval for music and motion*, pp. 69–84, 2007. 46

[111] Jorge R Vergara and Pablo A Estévez, "A review of feature selection methods based on mutual information", *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, 2014. 46

[112] Malik M Naeem Mannan, M Ahmad Kamran, Shinil Kang, and Myung Yung Jeong, "Effect of eog signal filtering on the removal of ocular artifacts and eeg-based brain-computer interface: A comprehensive study", *Complexity*, vol. 2018, 2018. 49

[113] Udaya Seneviratne, Armin Mohamed, Mark Cook, and Wendyl D'Souza, "The utility of ambulatory electroencephalography in routine clinical practice: a critical review", *Epilepsy research*, vol. 105, no. 1-2, pp. 1–12, 2013. 49

[114] Elena Nedelcu, Raluca Portase, Ramona Tolas, Raul Muresan, Mihaela Dinsoreanu, and Rodica Potolea, "Artifact detection in EEG using machine learning", in *Proc. ICCP*, Sept. 2017, pp. 77–83. 49

[115] Kevin T. Sweeney, Hasan Ayaz, Tomás E. Ward, Meltem Izzetoglu, Seán F. McLoone, and Banu Onaral, "A Methodology for Validating Artifact Removal Techniques for Physiological Signals", *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 5, pp. 918–926, Sept. 2012. 49

[116] Kevin T. Sweeney, Tomás E. Ward, and Seán F. McLoone, "Artifact Removal in Physiological Signals-Practices and Possibilities", *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 3, pp. 488–500, May 2012. 51

[117] Priyanka Khatwani and Archana Tiwari, "A survey on different noise removal techniques of eeg signals", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 1091–1095, 2013. 51

[118] Shaibal Barua and Shahina Begum, "A review on machine learning algorithms in handling eeg artifacts", in *Proc. Workshop SAIS*, 05 2014, vol. 14, pp. 1–10. 51

[119] Faridah Abd Rahman, Mohd Fauzi Othman, and Nurul Aimi Shaharuddin, "A review on the current state of artifact removal methods for electroencephalogram signals", in *2015 10th Asian Control Conference (ASCC)*. IEEE, 2015, pp. 1–6. 51

[120] Jose Antonio Urigüen and Begoña Garcia-Zapirain, "Eeg artifact removal—state-of-the-art and guidelines", *Journal of neural engineering*, vol. 12, no. 3, pp. 031001, 2015. 51, 64

[121] Avinash Tandle and Nandini Jog, "Classification of artefacts in eeg signal recordings and overview of removing techniques", *International Journal of Computer Applications*, vol. 975, pp. 8887, 2015. 51

[122] Md Kafiul Islam, Amir Rastegarnia, and Zhi Yang, "Methods for artifact detection and removal from scalp eeg: A review", *Clinical Neurophysiology*, vol. 46, no. 4-5, pp. 287–305, 2016. 51, 59

[123] Chang Jung, "A review on eeg artifacts and its different removal technique", *Asia-pacific Journal of Convergent Research Interchange*, vol. 2, pp. 45–62, 12 2016. 51

[124] Chi Qin Lai, Haidi Ibrahim, Mohd Zaid Abdullah, Jafri Malin Abdullah, Shahrel Azmin Suandi, and Azlinda Azman, "Artifacts and noise removal

for electroencephalogram (eeg): A literature review", in *Proc. ISCAIE*, 2018, pp. 326–332. 51

[125] 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. 51

[126] Sari Sadiya, Tuka Alhanai, and Mohammad M Ghassemi, "Artifact detection and correction in eeg data: A review", in *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2021, pp. 495–498. 51

[127] Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, and M Shamim Kaiser, "Abot: an open-source online benchmarking tool for machine learning-based artefact detection and removal methods from neuronal signals", *Brain Informatics*, vol. 9, no. 1, pp. 19, 2022. 51

[128] RStudio, Inc, *shiny: Easy web applications in R*, 2014, URL: `http://shiny.rstudio.com`. 51

[129] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al., "The fair guiding principles for scientific data management and stewardship", *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016. 55

[130] Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Machine learning in analysing invasively recorded neuronal signals: Available open access data sources", in *Brain Informatics*, Mufti Mahmud, Stefano Vassanelli, M. Shamim Kaiser, and Ning Zhong, Eds., Cham, 2020, pp. 151–162, Springer International Publishing. 57, 133

[131] Alexios Koutsoukas, Keith J Monaghan, Xiaoli Li, and Jun Huan, "Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data", *Journal of cheminformatics*, vol. 9, no. 1, pp. 42, 2017. 59

[132] Nicholus Mboga, Claudio Persello, John Ray Bergado, and Alfred Stein, "Detection of informal settlements from vhr images using convolutional neural networks", *Remote sensing*, vol. 9, no. 11, pp. 1106, 2017. 59

[133] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms.", *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019. 59

[134] R Christina Marion P Gilberet, Ria Susan Roy, NJ Sairamya, D Narain Ponraj, and S Thomas George, "Automated artifact rejection using ica and image processing algorithms", in *2017 International Conference on Signal Processing and Communication (ICSPC)*. IEEE, 2017, pp. 354–358. 59

[135] Samaneh Valipour, GR Kulkarni, and AD Shaligram, "Study on performance metrics for consideration of efficiency of the ocular artifact removal algorithms for eeg signals", *Indian Journal of Science and Technology*, vol. 8, no. 30, pp. 1–6, 2015. 59

[136] Xiangxin Zhu, Carl Vondrick, Charless C Fowlkes, and Deva Ramanan, "Do we need more training data?", *International Journal of Computer Vision*, vol. 119, no. 1, pp. 76–92, 2016. 60

[137] Junghwan Cho, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do, "How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?", 2016. 60

[138] Prabhat Garg, Elizabeth Davenport, Gowtham Murugesan, Ben Wagner, Christopher Whitlow, Joseph Maldjian, and Albert Montillo, "Automatic 1d convolutional neural network-based detection of artifacts in meg acquired without electrooculography or electrocardiography", in *2017 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, 2017, pp. 1–4. 62

[139] Montri Phothisonothai, Fang Duan, Hiroyuki Tsubomi, Aki Kondo, Kazuyuki Aihara, Yuko Yoshimura, Mitsuru Kikuchi, Yoshio Minabe, and Katsumi Watanabe, "Artifactual component classification from meg data

using support vector machine", in *The 5th 2012 Biomedical Engineering International Conference*. IEEE, 2012, pp. 1–5. 62

[140] David B Stone, Gabriella Tamburro, Patrique Fiedler, Jens Haueisen, and Silvia Comani, "Automatic removal of physiological artifacts in eeg: the optimized fingerprint method for sports science applications", *Frontiers in human neuroscience*, vol. 12, pp. 96, 2018. 62

[141] SI Gonçalves, PJW Pouwels, JPA Kuijer, RM Heethaar, and JC De Munck, "Artifact removal in co-registered eeg/fmri by selective average subtraction", *Clinical Neurophysiology*, vol. 118, no. 11, pp. 2437–2450, 2007. 62

[142] Udit Patidar and George Zouridakis, "A hybrid algorithm for artifact rejection in eeg recordings based on iterative ica and fuzzy clustering", in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2008, pp. 50–53. 62

[143] Elie Bou Assi, Sandy Rihana, and Mohamad Sawan, "Kmeans-ica based automatic method for ocular artifacts removal in a motorimagery classification", in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 6655–6658. 62

[144] "Physionet", Accessed on: Feb. 12, 2020. 62

[145] "Berlin brain-computer interface", Accessed on: Feb. 12, 2020. 62

[146] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "" why should i trust you?" explaining the predictions of any classifier", in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144. 64

[147] Petr Nejedly, Jan Cimbalnik, Petr Klimes, Filip Plesinger, Josef Halamek, Vaclav Kremen, Ivo Viscor, Benjamin H Brinkmann, Martin Pail, Milan Brazdil, et al., "Intracerebral eeg artifact identification using convolutional neural networks", *Neuroinformatics*, pp. 1–10, 2018. 64

[148] Michele Lo Giudice, Nadia Mammone, Cosimo Ieracitano, Maurizio Campolo, Arcangelo Ranieri Bruna, Valeria Tomaselli, and Francesco Carlo Morabito, "Visual explanations of deep convolutional neural network for eye blinks detection in eeg-based bci applications", in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 01–08. 64

[149] Giuseppe Placidi, Luigi Cinque, and Matteo Polsinelli, "Convolutional neural networks for automatic detection of artifacts from independent components represented in scalp topographies of eeg signals", *arXiv preprint arXiv:2009.03696*, 2020. 64

[150] Prabhat Garg, Elizabeth Davenport, Gowtham Murugesan, Ben Wagner, Christopher Whitlow, Joseph Maldjian, and Albert Montillo, "Automatic 1d convolutional neural network-based detection of artifacts in meg acquired without electrooculography or electrocardiography", in *2017 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, 2017, pp. 1–4. 64

[151] Prabhat Garg, Elizabeth Davenport, Gowtham Murugesan, Ben Wagner, Christopher Whitlow, Joseph Maldjian, and Albert Montillo, "Using convolutional neural networks to automatically detect eye-blink artifacts in magnetoencephalography without resorting to electrooculography", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 374–381. 64

[152] Mai-Anh T Vu, Tülay Adalı, Demba Ba, György Buzsáki, David Carlson, Katherine Heller, Conor Liston, Cynthia Rudin, Vikaas S Sohal, Alik S Widge, et al., "A shared vision for machine learning in neuroscience", *Journal of Neuroscience*, vol. 38, no. 7, pp. 1601–1607, 2018. 64

[153] Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Machine learning in analysing invasively recorded neuronal signals: Available open access data sources", in *Proc. Brain Informatics*, 2020, pp. 151–162. 68

[154] V Arulmozhi, "Classification task by using matlab neural network tool box–a beginner's view", *International Journal of Wisdom Based Computing*, vol. 1, no. 2, pp. 59–60, 2011. 75

[155] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 82

[156] Louis Kim, Jacob Harer, Akshay Rangamani, James Moran, Philip D Parks, Alik Widge, Emad Eskandar, Darin Dougherty, and Sang Peter Chin, "Predicting local field potentials with recurrent neural networks", in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 808–811. 89

[157] Avijit Paul, "Prediction of missing eeg channel waveform using lstm", in *2020 4th International Conference on Computational Intelligence and Networks (CINE)*. IEEE, 2020, pp. 1–6. 89

[158] Marcos Fabietti, Mufti Mahmud, Ahmad Lotfi, Alberto Averna, David Guggenmo, Randolph Nudo, and Michela Chiappalone, "Artifact detection in chronically recorded local field potentials using long-short term memory neural network", in *Proc. AICT*, 2020, pp. 1–6. 90

[159] Mohamed Hosny, Minwei Zhu, Wenpeng Gao, and Yili Fu, "A novel deep lstm network for artifacts detection in microelectrode recordings", *Biocybern. Biomed. Eng.*, vol. 40, no. 3, pp. 1052–1063, 2020. 90

[160] Nitheesh Kumar Manjunath et al., "A low-power lstm processor for multi-channel brain eeg artifact detection", in *Proc. ISQED*, 2020, pp. 105–110. 90

[161] D Kim and S Keene, "Fast automatic artifact annotator for eeg signals using deep learning", in *Proc. SPMB*, 2019, pp. 1–5. 90

[162] S Selvan and R Srinivasan, "Recurrent neural network based efficient adaptive filtering technique for the removal of ocular artefacts from eeg", *IETE Tech. Rev.*, vol. 17, no. 1-2, pp. 73–78, 2000. 90

[163] Abbas Erfanian and Babak Mahmoudi, "Real-time ocular artifact suppression using recurrent neural network for electro-encephalogram based brain-computer interface", *Med. Biol. Eng. Comput.*, vol. 43, no. 2, pp. 296–305, 2005. 90

[164] Jasman Pardede, Mardi Turnip, Darwis Robinson Manalu, and Arjon Turnip, "Adaptive recurrent neural network for reduction of noise and estimation of source from recorded eeg signals", *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, 2015. 90

[165] Matlab, "MATLAB. Deep Learning Toolbox R2020a", 2017. 90

[166] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014. 91

[167] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions", *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021. 92

[168] Iliana Paliari, Aikaterini Karanikola, and Sotiris Kotsiantis, "A comparison of the optimized lstm, xgboost and arima in time series forecasting", in *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2021, pp. 1–7. 92

[169] Peter T Yamak, Li Yujian, and Pius K Gadosey, "A comparison between arima, lstm, and gru for time series forecasting", in *Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence*, 2019, pp. 49–55. 92

[170] Xing Yan and Nurul A Chowdhury, "Mid-term electricity market clearing price forecasting: A hybrid lssvm and armax approach", *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 20–26, 2013. 92

[171] Kacper Sokol and Peter Flach, "Limetree: Interactively customisable explanations based on local surrogate multi-output regression trees", *arXiv preprint arXiv:2005.01427*, 2020. 105

[172] Mufti Mahmud, M Shamim Kaiser, Martin T McGinnity, and Amir Hussain, "Deep learning in mining biological data", *Cognitive Computation*, vol. 13, no. 1, pp. 1–33, 2021. 105

[173] JeffL Teeters and Friedrich T Sommer, "ephdf–a standard for storing electrophysiology data in hdf5", *F1000Research*, vol. 4, 2013. 105

[174] Marcos Fabietti, Mufti Mahmud, and Ahmad Lotfi, "Effectiveness of employing multimodal signals in removing artifacts from neuronal signals: An empirical analysis", in *Proc. Brain Informatics*, 2020, pp. 183–193. 105, 132

[175] Charlotte Cecere, Christen Corrado, and Robi Polikar, "Diagnostic utility of eeg based biomarkers for alzheimer's disease", in *Proc. NEBEC*, 2014, pp. 1–2. 107

[176] Vernon J Lawhern et al., "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces", *J. Neural Eng.*, vol. 15, no. 5, pp. 056013, 2018. 108

[177] Madelaine Daianu et al., "Breakdown of brain connectivity between normal aging and alzheimer's disease: a structural k-core network analysis", *Brain Connect.*, vol. 3, no. 4, pp. 407–422, 2013. 111

[178] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi, "A survey of the recent architectures of deep convolutional neural networks", *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020. 111

[179] Yoav Freund and Robert E Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997. 112

[180] Noor Kamal Al-Qazzaz et al., "Eeg feature fusion for motor imagery: A new robust framework towards stroke patients rehabilitation", *Comput. Biol. Med.*, vol. 137, pp. 104799, 2021. 112

[181] Aleksandra Maruszak and Sandrine Thuret, "Why looking at the whole hippocampus is not enough-a critical role for anteroposterior axis, subfield and activation analyses to enhance predictive value of hippocampal changes for alzheimer's diagnosis", *Front. Cell. Neurosci.*, vol. 8, pp. 95, 2014. 117

[182] Yilan Xu, Manna Zhao, Yuying Han, and Heng Zhang, "Gabaergic inhibitory interneuron deficits in alzheimer's disease: Implications for treatment", *Front. Neurosci.*, vol. 14, pp. 660, 2020. 119

[183] Katerina Kalemaki et al., "Mice with decreased number of interneurons exhibit aberrant spontaneous and oscillatory activity in the cortex", *Front. Neural Circuits*, vol. 12, pp. 96, 2018. 119

[184] Margaret F Carr, Mattias P Karlsson, and Loren M Frank, "Transient slow gamma synchrony underlies hippocampal memory replay", *Neuron*, vol. 75, no. 4, pp. 700–713, 2012. 120

[185] JA Van Deursen, EFPM Vuurman, FRJ Verhey, VHJM van Kranen-Mastenbroek, and WJ Riedel, "Increased eeg gamma band activity in alzheimer's disease and mild cognitive impairment", *Journal of neural transmission*, vol. 115, pp. 1301–1311, 2008. 126

[186] Liang Wang, Yufeng Zang, Yong He, Meng Liang, Xinqing Zhang, Lixia Tian, Tao Wu, Tianzi Jiang, and Kuncheng Li, "Changes in hippocampal connectivity in the early stages of alzheimer's disease: evidence from resting state fmri", *Neuroimage*, vol. 31, no. 2, pp. 496–504, 2006. 126

[187] Satoshi Yamaguchi, Kenichi Meguro, Masatoshi Itoh, Chika Hayasaka, Masumi Shimada, Hideki Yamazaki, and Atsushi Yamadori, "Decreased cortical glucose metabolism correlates with hippocampal atrophy in alzheimer's disease as shown by mri and pet.", *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 62, no. 6, pp. 596–600, 1997. 126

[188] Evan Porter, Patricia Fuentes, Zaid Siddiqui, Andrew Thompson, Ronald Levitin, David Solis, Nick Myziuk, and Thomas Guerrero, "Hippocampus segmentation on noncontrast ct using deep learning", *Medical physics*, vol. 47, no. 7, pp. 2950–2961, 2020. 126

[189] Xin He, Kaiyong Zhao, and Xiaowen Chu, "Automl: A survey of the state-of-the-art", *Knowledge-Based Systems*, vol. 212, pp. 106622, 2021. 132

[190] A Saranya and R Subhashini, "A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends", *Decision analytics journal*, p. 100230, 2023. 133

[191] Vitor Bento, Manoela Kohler, Pedro Diaz, Leonardo Mendoza, and Marco Aurelio Pacheco, "Improving deep learning performance by using

explainable artificial intelligence (xai) approaches", *Discover Artificial Intelligence*, vol. 1, pp. 1–11, 2021. 133