POLITECNICO DI TORINO

Master degree course in Data Science and Engineering

Master Degree Thesis

Unsupervised and Self-Supervised Machine-Learning for Epilepsy Detection on EEG Data

Supervisors

Prof. Daniele Jahier Pagliari[1]

Prof. Luca Benini[2,3]

Dr. Andrea Cossettini[2]

Dr. Alessio Burello[1,3]

PhD. student Thorir Mar Ingolfsson[2]

Dr. Simone Benatti[3]

Candidate

Luca Benfenati

1. Politecnico di Torino, 2. ETH Zurich, 3. Università di Bologna

Academic year 2022-2023

Abstract

Epilepsy is a neurological disorder characterized by abnormal electrical activity of the brain that causes recurrent seizures. Electroencephalography (EEG) data can help in the detection of such seizures. However, labelled EEG datasets are scarce because the labelling process of this type of data is a time-consuming and expertise-requiring activity. On the other hand, vast amounts of unlabelled data are available. The objective of this work is to understand if and how it is possible to exploit unannotated datasets for seizure detection on EEG data. Since supervised methods are limited by the amount of labelled data available, the thesis focuses on unsupervised and self-supervised methods.

Firstly, two different fully-unsupervised methods proposed by the literature are considered. These methods exploit non-seizure data to learn their distribution and then recognize seizures based on how much they differ from the training distribution. However, since the results obtained with these two methods were not promising, the focus shifted to self-supervised methods. In this context, BENDR, inspired by Large Language Model BERT and self-supervised speech recognition approach wav2vec 2.0, was proposed. BENDR is pre-trained on a huge unlabelled EEG dataset (TUEG) and fine-tuned for different Brain-Computer Interface (BCI) tasks and datasets. Starting from the pre-trained weights made available by the authors of BENDR, the thesis adapts this self-supervised approach to a different downstream task (seizure detection) and a different dataset (CHB-MIT). The idea is to exploit the knowledge learned on a huge amount of unlabelled data to understand and capture the underlying structure of data. Once this first unsupervised task has been carried out, the model is then fine-tuned on a more specific task and a smaller labelled dataset. An extensive search of the optimal fine-tuning strategy is carried out, considering various aspects. The impact of model size when fine-tuning is evaluated, as well as the impact of pre-processing and post-processing techniques, the impact of further pre-training on the downstream dataset itself, and the impact of reducing the available training

data.

The key takeaways that have been found and validated during the development of the thesis are: (i) a model of smaller size may prevent overfitting on a smaller dataset than the one on which it was pre-trained; (ii) regularization techniques (especially heavy dropout, early stopping mechanism, learning rate scheduler, and new losses) reduce overfitting and improve the generalization on different patients; (iii) finally, pre-processing and post-processing techniques have the biggest impact on performance improvement.

At the end of this extensive search, performance comparable with the current supervised state of the art was obtained (even slightly better under certain conditions): specifically, 99.9% specificity, 66.6% sensitivity, and 0.698 FP/h. This work further validated the effectiveness of a huge large language-inspired model as BENDR and of the self-supervision approach in EEG-based tasks. The thesis successfully showed the potential of the transfer learning scheme applied to EEG in the seizure detection task, leveraging the huge amounts of unlabelled EEG data available.

Acknowledgments

First of all, I would like to thank Alessio Burello and Thorir Mar Ingolfsson for the complete support and the useful suggestions during the development of my Master's Thesis. I would like to thank Doctor Andrea Cossettini and Professor Luca Benini for the hospitality and the opportunity they have given me at ETH of Zurich. I would like to thank also Professor Daniele Jahier Pagliari from Politecnico di Torino, without him, it would not have been possible for me to have this great and challening research experience at ETHZ.

A final general thank to all the people at the Integrated Systems Laboratory for the warm welcome and, again, the hospitality.

Contents

List of Figures 9

List of Tables 13

1 Introduction 15

2 Background 19
2.1 Epilepsy . . . 19
2.1.1 Epileptic Seizure . . . 19
2.1.2 Scalp Electroencephalogram . . . 20
2.1.3 Seizure detection . . . 22
2.2 Machine Learning and Deep Learning . . . 24
2.3 Self-supervised learning . . . 24
2.3.1 The Transformer . . . 25
2.3.2 BERT . . . 29
2.3.3 The wav2vec 2.0 framework . . . 30

3 Related Work 35
3.1 Supervised learning . . . 36
3.2 Unsupervised learning . . . 38
3.3 Self-supervised learning . . . 40

4 Matherial and Methods 43
4.1 Unsupervised approaches for seizure detection . . . 43
4.1.1 Variational Autoencoder . . . 44
4.1.2 Combining CNN with Anomaly Detection Methods . . . 47
4.2 Self-supervised approaches for seizure detection . . . 49
4.2.1 BENDR . . . 49
4.2.2 MAEEG . . . 53
4.3 An extensive research for fine-tuning . . . 55

4.3.1 Scalability . . . 57
4.3.2 Pre-training architecture . . . 57
4.3.3 Processing techniques . . . 59
4.3.4 Reducing available training data . . . 64
4.3.5 Overall training procedure . . . 65

5 Results 67
5.1 Datasets . . . 67
5.2 Metrics . . . 69
5.3 Training details . . . 72
5.4 Results . . . 73
5.4.1 Scalability results . . . 74
5.4.2 Pre-training architecture results . . . 77
5.4.3 Processing techniques results . . . 78
5.4.4 Reducing training data available . . . 81
5.5 Discussion . . . 82

6 Conclusion and Future Works 89

A Appendix 91
A.1 Project description . . . 91
A.2 Unsupervised replicated results . . . 91
A.3 A different architecture for the fine-tuning process . . . 92
A.4 False alarms distribution . . . 93

List of Figures

2.1 Electrode arrangement according to the international 10/20 system [13] . . . 22
2.2 Different recognition tasks for diagnosis of epilepsy: **(a)** seizure detection, **(b)** seizure prediction, **(c)** seizure type classification [4] . . . 23
2.3 The transformer architecture [5] . . . 26
2.4 Scaled Dot-Product Attention (left). Multi-Head Attention consists of several attention layers running in parallel (right). [5] . . . 29
2.5 BERT architecture for the pre-training process (on the left) and for the fine-tuning process (on the right) [15]. . . . 31
2.6 The wav2vec 2.0 architecture [16] . . . 33

3.1 Number of articles on seizure detection, prediction and classification published since the 1970s till 2021 as reported by Web of Science [4] . . . 36

4.1 The general approach of unsupervised learning for the seizure detection task on EEG data . . . 44
4.2 The encoder-decoder scheme of a Variational Autoencoder. x is the input data, μ_x and σ_x are respectively the mean and the standard deviation of the Gaussian distribution $N(\mu_x, \sigma_x)$ to which z, the latent representation, belongs. $\hat{x}$ is the reconstructed data, which is a function d of the decoder [48] . . . 46
4.3 Variational Autoencoder architecture for non-seizure EEG data reconstruction [35] . . . 47
4.4 Two-branch architecture of the feature extractor. The blue dotted line highlights the part of the network that is used to extract representations of data on which the anomaly detection method is then trained . . . 48

4.5 Pre-training architecture of BENDR . . . 52
4.6 Fine-tuning architecture of BENDR . . . 53
4.7 Fine-tuning architecture of BENDR Linear . . . 54
4.8 The pre-training architecture of BENDR (A) and MAEEG (B) [51] . . . 55
4.9 Training strategy with Leave-One-Out Cross-Validation. In the figure, it is possible to see an example for patient chb01, with file6 test set. This is repeated for all the files of the patient and aggregated results are computed by averaging the metrics on all the different test sets. . . . 56
4.10 Scalability: the different architecture sizes tested . . . 58
4.11 Alternative training strategy implemented to test BENDR vs MAEEG . . . 59
4.12 SMOTE [52] . . . 61
4.13 WRS [53] . . . 61
4.14 Smoothing criteria applied to the output of the model. Notice how since only windows of odd width are considered, the majority voting is implemented by computing the average of the values in the window and then approximating to the nearest integer. . . . 62
4.15 Post-processing technique that group together consecutive false positives. The first array shows the target or the ground truth of the EEG segments considered. The second array shows the output of the model and how the FP are counted before the post-processing is applied. The third array is how the FP are counted after the post-processing is applied. On the right it is possible to see how the specificity is computed . . . 63
4.16 Segment-based approach vs Event-based approach to computing sensitivity. . . . 64

5.1 Metrics describing the TUH-EEG corpus. [Top left] histogram showing number of recording sessions (each patient can have from 1 - most common - up to 6 recording sessions); [top right] histogram showing number of sessions recorded per calendar year; [bottom left] histogram of patient ages; [bottom right] histogram showing number of EEG-only channels (purple); and total channels (green) [54]. . . . 68

5.2 4 different model sizes compared with each other. If looking at the FP/h metric it is possible to spot a sort of optimal point with the 55% architecture. If the model size is either increased or decreased, the FP/h worsens. 76
5.3 Comparison between smoothing techniques: majority voting vs minimum value 79

A.1 False positive distribution with respect to elapsed time between false positive and actual seizure onset, for all patients and all test files. 94

List of Tables

3.1 State-of-the-art performances of the supervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column. 38

3.2 State-of-the-art performances of the unsupervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column. 40

3.3 State-of-the-art of the self-supervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column. Notice also that only results that regard the seizure detection task are shown in the *Performance* column (or anomaly detection, since a seizure can be considered an anomaly). The other works for which it is not shown any results are not related to seizure detection and thus, in order to avoid confusion, nothing is shown in the *Performance* column. 42

4.1 Dimensions explored during the fine-tuning optimization process. 65

5.1 CHB-MIT Scalp EEG Database details per patients 69

5.2 Confusion Matrix 70

5.3 Hyperparameters search space. Optimal values are in **bold**. . 74

5.4 First results obtained fine-tuning BENDR. On the left side, there are results obtained exploiting the available pre-trained weights, on the right side results without pre-trained weights. . 75
5.5 Results based on different model configurations (different number of convolutional blocks used in the first stage, different number of heads and layers used in the transformer 76
5.6 Comparison of BENDR and MAEEG when pre-training on all the patients of CHB-MIT except the one on which the fine-tuning is run. The first row shows the performance of BENDR without pre-training on other CHB-MIT patients. 77
5.7 Comparison between different pre-processing and post-processing techniques. The configurations differ from each other because of the pre-processing techniques applied. In the table, it is possible to see how the post-processing techniques change the performances. In this case, "Smoothing" refers to the Majority voting strategy based on sliding windows on the outputs with length 3, while "Event+Grouped" refers to the event-based sensitivity combined with the grouping consecutive false positive techniques, discussed in 4.3.3. 79
5.8 Result of configuration 1 and 2 patient by patient 80
5.9 Best results without considering critical patients 81
5.10 Results comparison with the state of the art 81
5.11 Results of BENDR when fine-tuned on less data. 82
5.12 Number of seizure events per hour for each CHB-MIT patient 86

A.1 Comparison between original and replicated results of the unsupervised approaches. VAE is the variational autoencoder method, while CNN+AD is the one that combines CNN with the anomaly detection method. 92
A.2 Fine-tuning results obtained with the two different architectures: BENDR and Linear BENDR 93

Chapter 1

Introduction

Epilepsy is one of the most common chronic diseases of the nervous system that affects almost 1% of the worldwide population, i.e., around 70 million people [1]. It is a brain disease defined primarily by frequent and unpredictable disruptions in normal brain activity, causing what is known as epileptic seizures. These recurrent seizures are brief episodes of involuntary movement that may involve a part of the body (partial) or the entire body (generalized). Seizure episodes are a result of excessive electrical discharges in a group of brain cells. Different parts of the brain can be the site of such discharges.

There are several treatment approaches that can be used depending on the individual and the type of epilepsy. However, about 30–40% of epilepsy patients exhibit treatment resistance to drug therapy and suffer from complex epilepsy symptoms, called drug-refractory epilepsy (DRE [2]). In this case, electroencephalogram (EEG) becomes an important tool for the diagnosis of epilepsy. EEG is a recording of brain activity: the brain's neurons contain ionic current, which creates voltage fluctuations that EEG can measure. This electrical activity is spontaneous and recorded over a period of time from many electrodes to form an EEG signal. Electroencephalography was first introduced by Hans Berger [3] to measure the electrical activity of different regions in the human brain, which can be particularly useful in the diagnosis of different types of brain disorders. Such a tool helps neurologists study the fluctuations in the brain that occur during epileptic seizures. The analysis of these fluctuations can aid in accurately distinguishing between healthy and unhealthy functionalities of the brain.

Over the last three decades [4], the use of EEG recordings in the study of

epileptic seizures has risen sensibly in tasks such as seizure detection, prediction, and classification using EEG signals. The use of machine learning and deep learning architectures for analyzing these EEG recordings has recently shown several advantages and improvements, starting from diagnosis support, and minimizing the need for trained specialists. In this context, automatic epilepsy detection on EEG with machine learning (ML) and deep learning (DL) models is a challenging task, both due to the limited amount of labelled data often available for the training of a classifier, and for the high accuracy standards required for a monitoring device to detect all seizure events without raising highly stressful false alarms. However, the recent availability of large EEG datasets has paved the way for the use of deep learning models for EEG decoding and classification. The main problem with these datasets is the lack of labelling, which is a time and effort-consuming process that requires the supervision of an experienced clinician. To address this problem, unsupervised and self-supervised learning techniques leverage unlabelled datasets to learn complex structures in the data via new supervised-learning tasks: for example, by occluding portions of the data and expecting the model to predict what has been hidden, or by providing two samples from the same patient and training the model to associate them strongly. In addition, in recent years, most of the researchers that worked on the epilepsy detection task have focused on developing their own model, with specific and well-suited characteristics based on the type of data and task that they wanted to solve. A small number of them tried to exploit the benefits of a "general" model that can be adapted for different types of tasks. Once again, in this context, self-supervised learning techniques propose an alternative to classical shallower machine learning models, which are built and thought, through a time and effort-consuming process, for a singular specific task.

Self-supervised learning has seen some recent success in natural language processing (NLP) with Language Models (LMs): one of the key aspects of this success is the Transformer architecture, introduced in the ground-breaking paper "Attention is All You Need" [5] by Google in 2017. This new architecture allows the model to handle large amounts of data more efficiently and capture complex patterns in the data thanks to its self-attention mechanism, which is able to attend to different parts of the input sequence while computing the output. With this new approach, the model can capture long-range dependencies between words and handle long sequences of text more efficiently than previous RNN-based or CNN-based NLP models. The recent pace of progress has increased dramatically and led to self-supervised deep representations that appear to approach and possibly even surpass that of

fully-supervised representations. This has raised hopes that self-supervised methods could indeed replace the widespread annotation-intensive paradigm of supervised deep learning going forward. These models are understood to work by making a very general model of language and appear to be even immediately capable of performing tasks they were not explicitly trained to accomplish. The main idea is to first define an unsupervised task on a huge unlabelled dataset in order to learn the representations of data as broadly as possible, which captures the underlying structure and pattern of that specific type of data. Once the model has completed this "pre-learning" phase, the general representations can be used to fine-tune on downstream tasks, which are usually related tasks for which a relatively small amount of labelled data is available. In comparison with shallower neural networks, which have proven to be more effective classifiers than their deeper counterparts in several EEG applications (as in [6], [7], [8]), the range of learnable features is relatively limited. The performance of shallower models more quickly saturates to lower performance levels as compared to a deeper network alternative ([9]), suggesting that more complex features could be developed using deeper neural networks when using training data that was more consistent. Overcoming the limitations of shallower networks in favour of deeper DNNs that could surpass feature engineering approaches likely requires addressing the large variability between different contexts.

The objective of the following work is to test unsupervised and self-supervised approaches for seizure detection with EEG data, exploring the techniques and the advantages of the use of huge unlabelled EEG datasets, which, as already discussed, have become recently more and more available. In this context, for unsupervised approaches, the thesis will focus mostly on Anomaly detection methods, while for the self-supervised ones, the focus will be on architectures inspired by the Large Language Model seen in the NLP field. A more thorough approach is considered for the self-supervised part: specifically, a very promising model inspired by the recent NLP developments is studied in detail, analyzing the benefits of exploiting both Convolutional Neural Network (CNN) and Transformers architecture. In this context, several tests are carried out on a famously renowned EEG dataset collected at the Children's Hospital Boston, known as CHB-MIT, for the seizure detection task, considering a multitude of combinations of architectures, pre-training alternatives, pre-processing and post-processing techniques, amount of data available for the fine-tuning task, and others. The end goal is to discover the potential of self-supervised learning in the EEG field and see if there are the conditions for success as there have been with Transfer Learning (TL) in Computer

Vision. The main focus is to understand if it is possible to develop and use a huge *general model* that is able to leverage knowledge learned from one context (in an unsupervised way) such that it may be useful in a different one (in a supervised way).

The contributions of this thesis are:

- an extensive overview of several state-of-the-art approaches for the seizure detection task on EEG data that either do not employ labelled data (unsupervised) or exploit unlabelled data (self-supervised). In this first part, the focus is on the replication and validation of the most promising and interesting approaches to the CHB-MIT dataset;
- the adaptation of a large language-inspired model which has been proposed by the literature to work with EEG data, to the seizure detection task on the CHB-MIT dataset. In this part, once the data from CHB-MIT has been properly prepared and processed, the model is modified to cope with a different task and a different dataset with respect to the ones on which it was originally developed.
- the optimization of the fine-tuning step of the model. In this context, several fine-tuning dimensions are considered: the thesis searches for the optimal model size, the optimal pre-processing and post-processing techniques, the optimal training strategy and the data used during training.

The thesis is organized as follows: Chapter 2 explains the background and necessary theory underlying to understand the project fully. What is an epileptic seizure, how is EEG data gathered, the problem being solved, and why is it challenging. Chapter 3 explores state-of-the-art methods, considering supervised, unsupervised, and self-supervised methods. It also gives a brief overview of the Large Languages models and Self-supervised approaches in the NLP field that inspired the main contribution of this work. Chapter 4 discusses the different architectures and methods tested, dividing it into two sections: one for the unsupervised approach and one for the self-supervised one. Chapter 5 presents all the experiments and the results obtained during the work: once again, the chapter is divided into two sections. Finally, chapter 6 addresses some final thoughts and considerations on the results and, most importantly, on the overall project, as well as some comments on possible future works.

Chapter 2

Background

This chapter addresses all the theoretical aspects and background that are necessary to fully understand the scenario in which the project has been developed. The chapter is divided into two different sections. The first section (2.1) gives the details for defining an epileptic seizure, how data is gathered in this field, and what types of different tasks are usually carried out with EEG. The second section (2.3) gives an overview of the background that is required to know to fully understand the model that is going to be used in the Material and Methods chapter (4). After a brief explanation of the self-supervised learning paradigm, the Transformer architecture is introduced, which is the building block of all the architectures considered in the following work; then, based on this, a famously renowned Masked Large Language Model is presented; and finally, an adaptation of this Language model to the speech recognition scenario.

2.1 Epilepsy

2.1.1 Epileptic Seizure

Neurons are cells within the brain that are capable of generating, propagating, and processing electric signals. They connect to other neurons to form functional networks: thus, the brain can be viewed as a collection of interacting neural networks. The inputs to a neural network can be excitatory if they promote activity among neurons or inhibitory if they suppress it [10].

Epileptic seizures are transient periods involving hyperactivity and hyper synchronization of a large number of neurons within one or more neural networks. These transient states may arise because of a perturbation that

creates an imbalance favouring the excitation of a neural network over its inhibition. The imbalance may arise because of the following:

- defects within a neuron (e.g., ion channel dysfunction);
- defects in connections between neurons (e.g., deficient inhibitory neurotransmitter synthesis);
- defects in neural network organization (e.g., the formation of aberrant excitatory connections between neurons).

Defects within neurons, neuronal connections, or neural network organization may result from a genetic disorder or from trauma to the central nervous system during life. Epileptic seizures are broadly classified according to their cerebral site of origin and spread. *Focal* seizures arise from a localized region of the brain's cortex and have clinical manifestations that reflect that region of the brain. As an example, a focal seizure originating in the temporal lobe, the part of the brain that processes emotions and short-term memory, may result in feelings such as euphoria, fear, and deja vu or hallucinations of taste or smell. Focal seizures may spread to involve other regions of the brain or the entire brain. *Generalized* seizures begin with abnormal electrical activity that appears to encompass the entire cerebral cortex. The manifestations of such widespread abnormal electrical activity often include the loss of consciousness. Motor manifestations of these seizures may include whole-body rigidity and jerking (tonic-clonic seizure) or whole-body loss of muscle tone (atonic seizure). A seizure that begins focally and then generalizes is referred to as a secondarily generalized seizure.

2.1.2 Scalp Electroencephalogram

Electrical activity generated by collections of neurons in the brain can be monitored using electroencephalogram (EEG) signals, a multi-channel recording of this activity: different channels measures the activity within different brain regions. The recording can be done using non-invasive electrodes arrayed on the scalp of patients, referred to as scalp EEG, or by implanting electrodes inside brain tissues during surgery, referred to as intracranial EEG signals (iEEG) [11]. The main difference between these two types of data is that the scalp EEG offers poor spatial resolution but high spatial coverage, while the intracranial EEG offers high spatial resolution but less spatial coverage.

The property of scalp and intracranial EEG that most complicates the seizure detection task is its *variability* across individuals with epilepsy, both in the seizure and non-seizure states. Typically, following the onset of a seizure, a set of EEG channels develops rhythmic activity that reflects underlying neuronal hypersynchrony. Both the location of the involved EEG channels as well as the spectral content of the rhythmic activity varies across individuals. Furthermore, the EEG signature of one patient's seizure may closely resemble the signature of abnormal, non-seizure EEG gathered from the same patient or different patient [12]. Within the scalp, EEG the seizure detection task is further complicated by the physical properties of the signal. The scalp EEG is most sensitive to the activity of neurons on the brain surface; consequently, the activity of neurons within deep brain structures has almost no influence on the scalp EEG.

The International 10–20 system is the standard to describe and apply the location of scalp electrodes in the context of recording sEEG signals. The numbers "10" and "20" refer to the distances between adjacent electrodes, which are either 10% or 20% of the total distance (front-back or right-left) of the skull. The total distance is based on the anatomical locations on the scalp: nasion and inion (front-back direction) and the two preauricular points (right-left direction) as seen in Figure 2.1. Using these anatomical landmarks, the placement of the electrodes can be determined along with these directions with the pre-specified proportions: 10% is used from the anatomical landmarks and the first electrode in that direction, and 20% is used between the other electrodes. Each electrode placement site has a letter to identify the lobe or area of the brain it is reading from pre-frontal (Fp), frontal (F), temporal (T), parietal (P), occipital (O), and central (C). Note that there is no 'central lobe'; due to their placement and depending on the individual, the 'C' electrodes can exhibit/represent EEG activity more typical of frontal, temporal, and some parietal-occipital activity. In addition, a 'z' (zero) denotes electrodes placed on the centre line, on the midline sagittal plane of the skull (FpZ, Fz, Cz, Oz). The 'z' electrodes are often utilized as 'grounds' or 'references'. Finally, even-numbered electrodes (2, 4, 6, 8) refer to electrode placement on the right side of the head, whereas odd numbers (1, 3, 5, 7) refer to those on the left.

An EEG signal or channel is formed by considering the difference between potentials measured at two electrodes. Channel FP1 - F7, for example, considers the difference between the potentials measured at the electrode FP1 and at the electrode F7. Each EEG channel summarizes activity localized within a region of the brain: channel FP1 - F7, for example, shows the

behaviour of neural activity originating within the frontal lobe of the left hemisphere. The onset of a focal seizure involves a change in activity on the few scalp EEG channels that lie above or near the site of the brain, giving rise to a seizure; on the other hand, the onset of a generalized seizure involves activity on all scalp EEG channels.

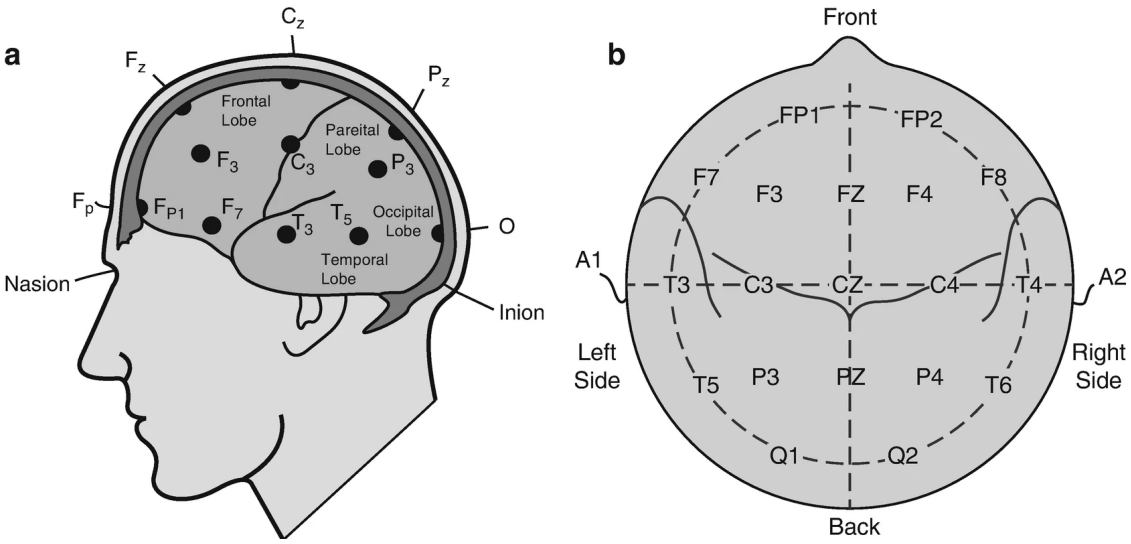

Figure 2.1. Electrode arrangement according to the international 10/20 system [13]

2.1.3 Seizure detection

As already discussed in the Introduction, it is possible to distinguish between three different tasks in the study of epileptic seizures. These three tasks can be distinguished and categorized depending on their final objective, specifically:

- in Seizure detection, a model identifies the presence or lack of seizures or abnormal activities after analyzing EEG signals;
- in Seizure prediction, a model can predict the likelihood of the occurrences of imminent epileptic seizures early on, by identifying the patient's pre-ictal state;
- in Seizure classification, a model can categorize different types of seizures or seizure phases. In other scenarios, the classification term is used for

classifying different seizure phases, known in the literature as EEG/phase classification.

Figure 2.2 shows an overview of the detection, prediction, and classification tasks. Seizure detection can be performed when it is required to review EEG recordings and evaluate seizure occurrences as a posthoc analysis: in Figure 2.2**a**, it is possible to see when the ideal case of seizure detected verifies, as soon as the seizure onset happens. On the other hand, seizure prediction is usually deployed to take into account safety precautions: the objective of this task is, indeed, to spot upcoming seizures with some forewarning and alert the patient or the health care professionals: in Figure 2.2**b**, it is possible to see a seizure being identified before the seizure onset. Finally, seizure classification wants to identify and classify the type of seizure (Figure 2.2**c**): a seizure can be tonic (i.e. stiffness of the muscles), atonic (i.e. relaxing of muscles), myoclonic (i.e. short jerking) and clonic (i.e. periods of shaking). This may be a crucial tool for neurologists to take the appropriate medical decision.

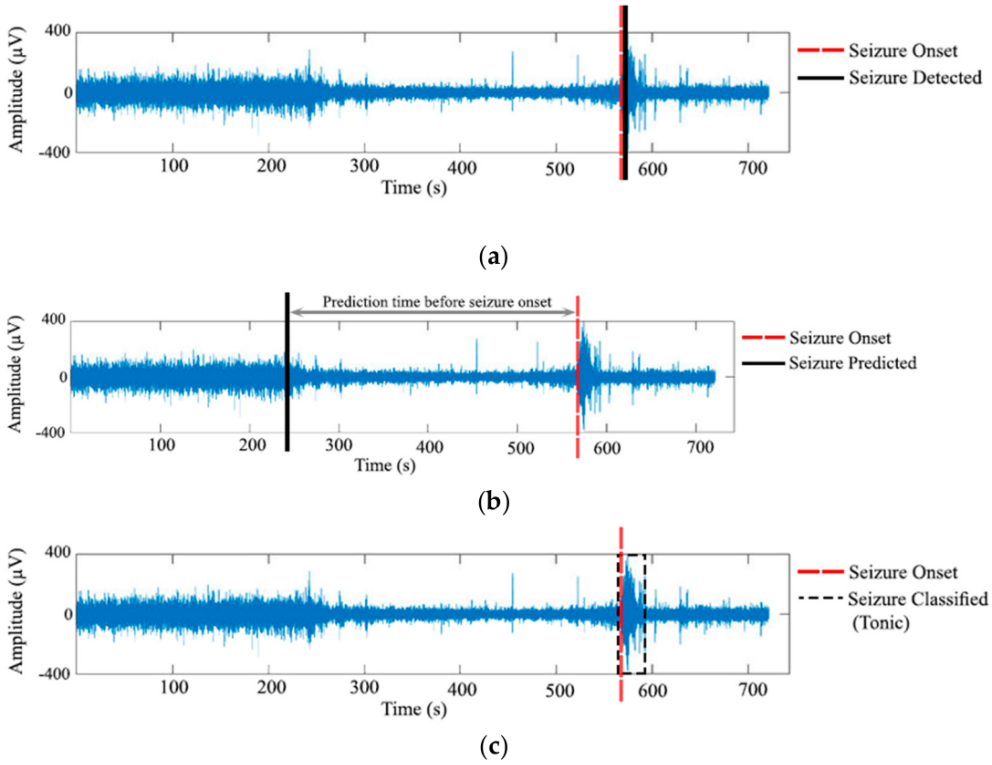

Figure 2.2. Different recognition tasks for diagnosis of epilepsy: **(a)** seizure detection, **(b)** seizure prediction, **(c)** seizure type classification [4]

2.2 Machine Learning and Deep Learning

Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. More precisely, machine learning describes the capacity of systems to learn from problem-specific training data to automate the process of analytical model building and solve associated tasks [14]. It is common to talk about Deep Learning, rather than simple or shallow machine learning, when there is a layered structure of algorithms that forms an artificial neural network, composed of multiple layers of processing used to extract progressively higher-level features from data. For many applications, deep learning models outperform shallow machine learning models and traditional data analysis approaches.

There is an important distinction between learning paradigms according to the amount of supervision they require. Machine and Deep Learning methods can be divided into three different categories:

- Supervised learning algorithms require a training set of example inputs and their corresponding desired outputs, which the algorithm uses to learn a model of the mapping from inputs to outputs. Once the model is learned, the algorithm can generate outputs for new inputs.
- Unsupervised learning algorithms do not require a training set, but instead, learn a model of the input data by detecting patterns in it. Unsupervised learning algorithms can be used to discover structure in data or to cluster data into groups.
- Self-supervised learning algorithms require only a training set of input data; the desired outputs are not provided. Instead, the algorithm learns a model of the input data and the desired outputs by detecting patterns in the data.

The focus of the thesis will be only on Unsupervised and Self-Supervised learning methods. Because of its complexity, the self-supervised learning paradigm and the common architectures used need to be further addressed. For this reason, a more specific and detailed Section (2.3) is presented.

2.3 Self-supervised learning

Self-supervised learning has emerged as a dominant paradigm in the field of Natural Language Processing (NLP) because of its ability to learn from

huge amounts of unlabelled data. In traditional supervised learning, human-labelled data is required to train a machine learning model, and labelling large amounts of data is expensive and time-consuming. The main advantage of self-supervised learning is that this type of paradigm does not require labelled data, or better can exploit the huge amount of unlabelled data available. Self-supervised learning consists of two distinct tasks:

- a *pre-text task* (or pre-training): it is usually a task without a specific objective if not the one of learning the underlying structure of the data available. This is typically carried out in an unsupervised way, to exploit the massive amount of unlabelled data. The idea is to transform the unsupervised problem into a supervised problem by auto-generating the labels from originally unlabelled data, to make the model learn intermediate representations of data.

- a *downstream task* (or fine-tuning): after the pre-training, the model is then fine-tuned exploiting the intermediate representations learned during the first task (more general) in a more specific field. Usually, the dataset on which the model is fine-tuned is smaller than the one on which it has been trained, and the task is more specific. The downstream task can be either carried out in an unsupervised way or in a supervised way

Self-supervised learning has seen huge success in recent years because it is the paradigm used to train Large Language models (LLM) in NLP. A large language model is a type of machine learning model that can perform a variety of natural language processing tasks with great performance. The success of these models trained with self-supervision is mainly due to the use of the Transformer, which can handle large quantities of data.

2.3.1 The Transformer

The transformer (Figure 2.3), introduced in 2017 by [5], is a neural network that learns context and thus meaning by tracking relationships in sequential data, thanks to a new mechanism called self-attention that allows the understanding of the overall context of the input data. The main advantage is that data can be fed all at once, differently from what happens for Recurrent Neural Networks (RNNs), that at the time were the state of the art of NLP applications. It is important to understand how each component of its architecture works.

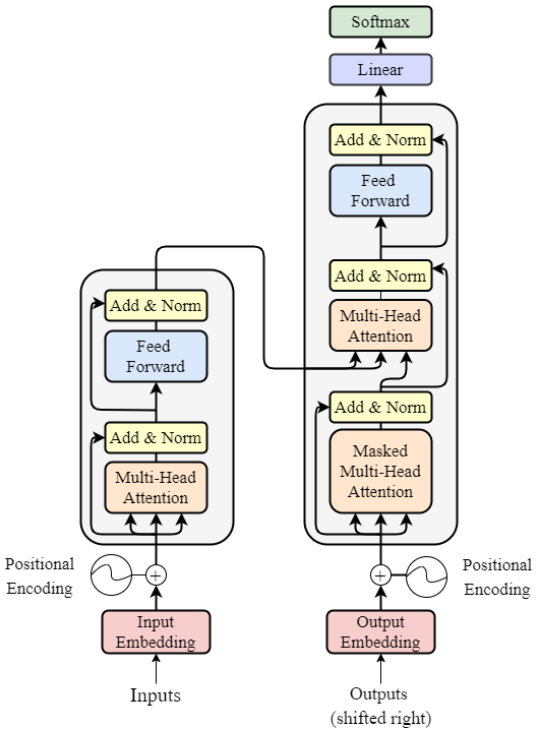

Figure 2.3. The transformer architecture [5]

The data needs to be encoded to identify the positions of the inputs: for this reason, the data is fed to the **input embedding** combined with **positional encoding** before the Encoder. There are many types of positional encoding, the most used use is a sine and cosine method which is a linear operation given by:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{n^{2i/d_{model}}}\right); \quad \mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{n^{2i/d_{model}}}\right) \tag{2.1}$$

where pos is the position in the input sequence, d_{model} is the output dimension of the embedding, n is a scalar and i is the dimension.

The **Encoder** (left part of Figure 2.3) has two main components, a Multi-head attention and a feed-forward layer. The encoder has N stacks of identical layers where:

- Multi-Head Attention: the positional encoded data goes through a multi-head Attention mechanism. This involves representing the data in three different vectors (value, key and query) which will go through attention scoring and will output one final vector;

- Add & Normalize: output data from self-attention is added to the input followed by a layer normalization;
- Feed-forward: normalized output goes through a simple feed-forward layer.

The output data from the encoder is once again fed through an output embedding combined with the positional encoding block. The **Decoder** (right part of Figure 2.3) has an additional step with respect to the encoder, the masked multi-head attention, which is used to train the model to predict the hidden parts of the data and estimate what is missing. The query and key vector are taken from the output of the Encoder but the value vector comes from the masked multi-head attention.

The **linear** layer takes the output from the decode and passes it to the final classification layer that uses the softmax activation function. The softmax outputs the predicted next-token probabilities as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad i = 1, ..., K, \quad z = [z_1, ..., z_K] \in \mathbb{R} \tag{2.2}$$

where K is the total number of tokens and z is the output vector from the linear layer. As already said, the self-attention mechanism is the key to the success of this architecture. This mechanism is carried out in multiple so-called heads of both the encoders and the decoders: within these heads, multiple operations and steps are carried out. A graphical summary of these operations is shown in Figure 2.4. Practically, an input matrix X is created from the input embeddings and translated into three different matrices: a **Query** matrix Q, a **Key** matrix K and a **Value** matrix V, with corresponding weight matrix $W^Q \in \mathbb{R}^{d_{model} \times d_v}$, $W^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W^V \in \mathbb{R}^{d_{model} \times d_v}$, where $d_v = d_k = d_{model}/h$. These matrices are created by a matrix multiplication operation:

$$Q = XW^Q \qquad K = XW^K \qquad V = XW^V \tag{2.3}$$

where $X \in \mathbb{R}^{N \times d_{model}}$ and the vectors x_i, q_i, k_i and v_i composed the matrices $Q, K, V \in \mathbb{R}^{N \times d_{model}/h}$.

$$A = \begin{bmatrix} - & x_1 & - \\ - & x_2 & - \\ & \vdots & \\ - & x_N & - \end{bmatrix}, Q = \begin{bmatrix} - & q_1 & - \\ - & q_2 & - \\ & \vdots & \\ - & q_N & - \end{bmatrix},$$

$$K = \begin{bmatrix} - & k_1 & - \\ - & k_2 & - \\ & \vdots & \\ - & k_N & - \end{bmatrix}, V = \begin{bmatrix} - & v_1 & - \\ - & v_2 & - \\ & \vdots & \\ - & v_N & - \end{bmatrix}$$

Then, considering a vector x_i that is being transformed to vectors q_i, k_i and v_i, a score s_j is computed for all the elements in vector k_i. The score is computed by matrix multiplication, i.e. the dot product of the query Q and the key K:

$$q_i \dot{k}_j = s_j, \qquad j \in J = N \tag{2.4}$$

In order to have stabler gradients, the score s_j is scaled with respect to the dimension of the key vector d_k, as in:

$$s_{j,\text{scaled}} = \frac{s_j}{\sqrt{d_k}} \tag{2.5}$$

Then, the score is normalized using soft-max:

$$\text{score}_i = \text{softmax}(s_{1,\text{scaled}}, s_{2,\text{scaled}}, ..., s_{j,\text{scaled}}) \tag{2.6}$$

Once this has been computed for all inputs, the value vectors are multiplied with the respective soft-max score, thus:

$$z_i = \text{score}_i \times v_i \tag{2.7}$$

The previous computations can also be simply represented with the matrix formulation:

$$Z = \text{Attention}(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.8}$$

Each one of the head h of the encoder or decoder produces one of these attention matrices Z. These matrices are then concatenated and multiplied by a weight matrix $W^O \in \mathbb{R}^{h\dot{d}_\text{v} \times d_\text{model}}$:

$$\begin{aligned} &\text{MultiHead}(Q, K, V) = Concat(Z_1, ..., Z_h)W^O \\ &\text{where} \quad Z_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \tag{2.9}$$

Since the transformer uses a MultiHead self-attention mechanism, there are h weight matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ where $i \in 1, ..., h$, which result in h Z_i matrices that will be concated at the end. Notice that

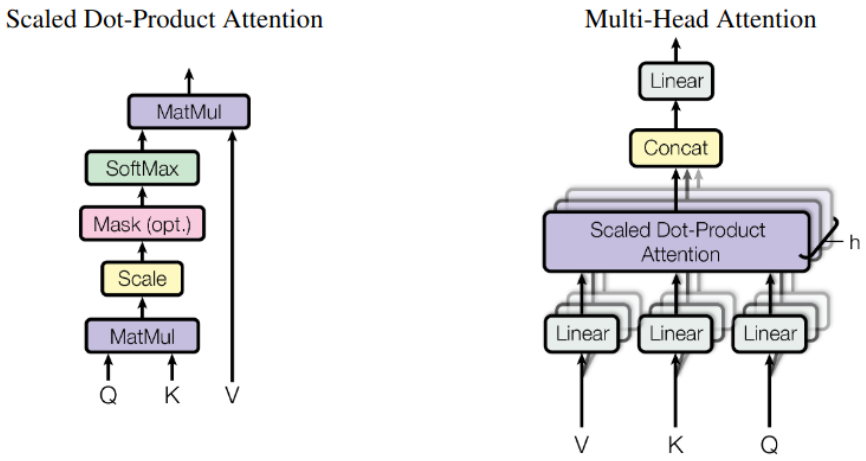

Figure 2.4. Scaled Dot-Product Attention (left). Multi-Head Attention consists of several attention layers running in parallel (right). [5]

there could also be some additional steps if there is, for example masking: in this case, it would be done between the scaling and the soft-max operations.

To summarize, in the **self-attention mechanism**, the input sequence is transformed into a sequence of *key, query*, and *value* vectors (K, Q, V), which are used to compute a weighted sum of the input sequence. The weights of the sum are determined by a softmax function applied to the dot product of the query vector and the key vector (attention score). The attention score is computed for each element in the input sequence, which is used to compute a weighted sum of the value vectors. The resulting output vector represents the attended representation of the input sequence, where each element of the output vector is a weighted sum of the input elements. One advantage of the self-attention mechanism is that it allows the model to selectively attend to different parts of the input sequence based on their relevance to the current processing step, rather than relying on fixed-length window-based approaches

2.3.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) was introduced in [15] by Google to address the limitations of the NLP context. The key innovation of BERT lies in its bidirectional nature, due to its Transformer-based architecture. Unlike previous models that processed text in a strictly left-to-right or right-to-left manner, BERT takes into account

both the left and right context of each word. BERT was designed to fine-tune downstream tasks on pre-trained deep bidirectional representations from unlabelled text. Utilizing massive datasets, the model learns the deep bidirectional representations to learn attributes of the data, which allows for less training when it comes to a supervised downstream task, hence smaller sets of labelled data are required.

The pre-training process of BERT considers as input a large amount of unlabelled data that are tokenized into smaller units. During pre-training, two unsupervised tasks are performed simultaneously:

1. Masked Language Modeling (MLM): part of the input token is randomly selected and masked. These masked tokens are then predicted based on the context provided by the surrounding tokens. The objective is for BERT to learn to understand and reconstruct the masked words.

2. Next Sentence Prediction (NSP): predicting whether two sentences appear consecutively in the original text or are randomly paired. This helps BERT learn the relationships between sentences and capture the context beyond individual sentences.

After the model is pre-trained, various supervised tasks can be fine-tuned. The pre-training and fine-tuning processes are summarized in Figure 2.5: it is immediate to notice how the objective change during these two phases. While during pre-training the model learns what is missing from the masked input based on the context of the input itself, during fine-tuning the objective can differ a lot based on the specific downstream task the model is trying to solve: from answering a question to translating in another language, and so on.

2.3.3 The wav2vec 2.0 framework

Wav2Vec 2.0 is a deep learning model for self-supervised speech representation learning developed by Facebook AI Research (FAIR) [16]. This model and its approach are of crucial importance for the thesis. In Section 4, the model that will be used as starting point of the self-supervised part of the thesis is, citing the authors, *an adaptation of wav2vec 2.0 to EEG*, in terms of both the architecture and the training approach. Wav2vec 2.0 has been developed to leverage large amounts of unlabelled speech data to learn meaningful representations that capture the underlying structure of spoken language. By pre-training on unlabelled data, the model can learn to understand the

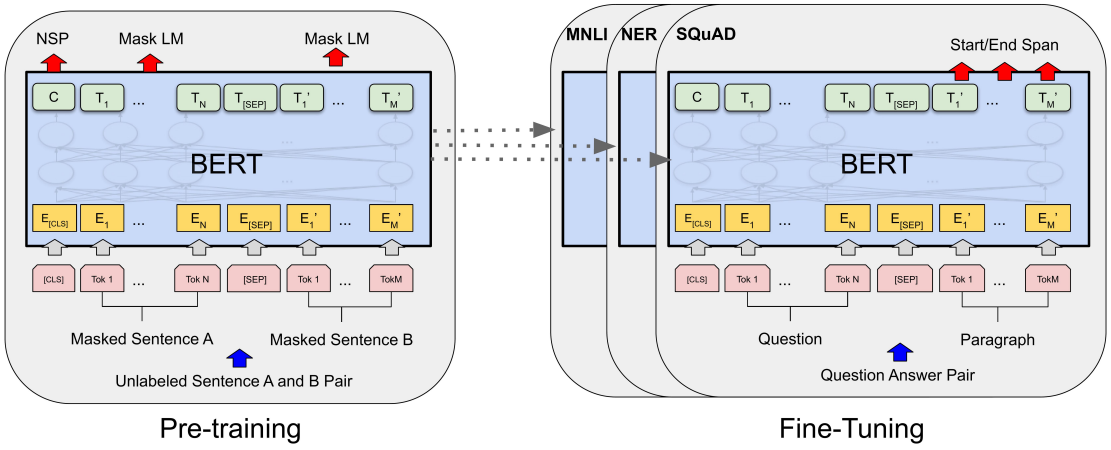

Figure 2.5. BERT architecture for the pre-training process (on the left) and for the fine-tuning process (on the right) [15].

characteristics of speech signals and encode them into rich representations. Wav2vec 2.0 uses a similar framework as BERT, but with an additional step at the bottom of the architecture: a CNN-based feature extractor is added before the Transfomer-based module.

The architecture of wav2vec 2.0 consists of three different modules:

1. a multi-layer convolutional neural network (**feature encoder**), that takes as input raw audio X, and converts it into a sequence of latent representations that capture the underlying structure of the. These representations can be seen as the new features $z_1, ..., z_T$ for T time steps.

$$\text{Encoder:} \quad f : X \mapsto Z \tag{2.10}$$

   This first network is made of seven sequential blocks with 1D Temporal Convolution, Layer normalization and GELU activation function. The CNN processes the input data with a sliding window that moves across the input data. This allows the CNN to capture local patterns in the data and learn relevant features.

2. A **quantization module** that converts continuous representations of the encoder Z into finite quantized representations Q.

$$\text{Quantization module:} \quad f : Z \mapsto Q \tag{2.11}$$

3. A **transformer encoder** similar to the one already used in BERT, with the difference that it employs a relative positional embedding instead of

the fixed positional embedding seen in Figure 2.3. The relative positional embeddings are learned during training using convolutional layers followed by a GELU activation function. They are then added to the input features Q and a layer normalization is applied before they are processed by the transformer layers. The output of the transformer is the context representations C.

$$\text{Transformer:} \quad f : Q \mapsto C \tag{2.12}$$

The architecture described is shown in Figure 2.6. The model is trained using contrastive learning where the loss of the task is defined by the quantized representation and the final output of the transformer. Specifically, a certain proportion of time steps in the latent feature encoder space Z are masked, and the model learns to identify the correct quantized latent representations in a set of so-called 'distractors' at each masked time step. The contrastive loss is defined as:

$$L_m = \text{-log}\frac{\exp(cossim(c_t, q_t)/k)}{\sum_{\tilde{q}\in Q_t}\exp(cossim(c_t, \tilde{q})/k)} \tag{2.13}$$

where $cossim$ is the cosine similarity, q_t are the quantized latent speech representations, $\tilde{q} \in Q_t$ are the candidate representations of dimension $K+1$, and Q_t is the set that contains the correct representations q_t and K distractors.

Wav2vec 2.0 uses self-supervision to learn from unlabelled training data and to enable speech recognition systems for many more languages, dialects, and domains. With one hour of labelled training data, wav2vec 2.0 outperforms the previous state of the art on the 100-hour subset of the LibriSpeech benchmark, using almost 100 times less labelled data. Both the architecture and the training paradigm have seen great success in the Natural Language Processing field, dominating the state of the art and defining the starting point for every new study.

In conclusion, in order to understand what is the relevance of this Background chapter and the contact with the thesis, it is possible to state that: the transformer is the key to the success of architectures such as BERT, that inspired self-supervised speech recognition approach wav2vec 2.0, which have been adapted to the EEG world by BENDR ([17]), which is the starting point of the self-supervised part of the thesis.

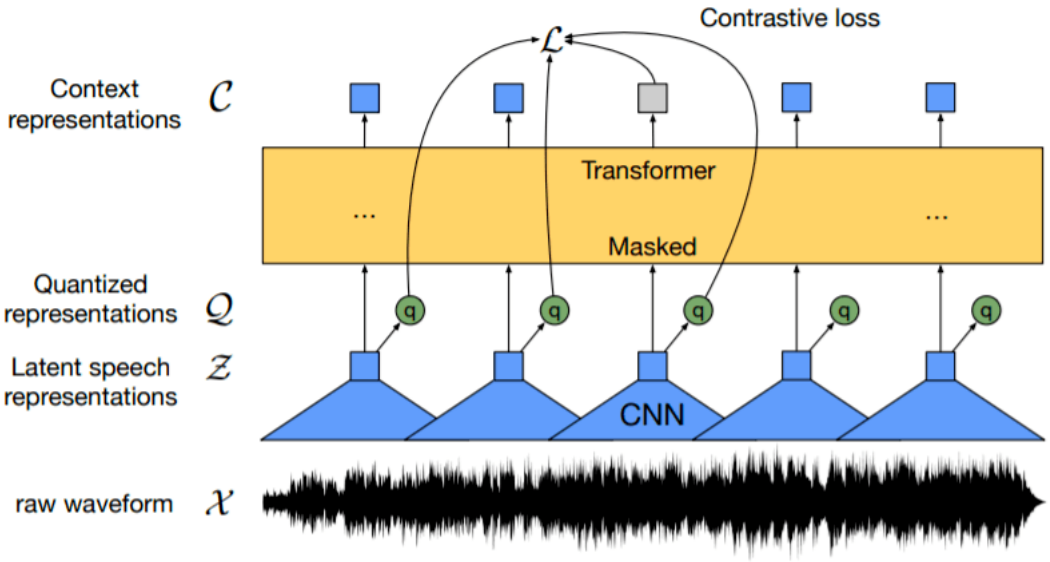

Figure 2.6. The wav2vec 2.0 architecture [16]

Chapter 3

Related Work

In this chapter, the current state-of-the-art of the seizure prediction task with Machine Learning and Deep Learning models is explored. The chapter is divided into sections based on the type of learning and the use of labels in the training phase: supervised, unsupervised and self-supervised learning. Briefly, supervised learning algorithms require a training set of example inputs and their corresponding desired outputs, which the algorithm uses to learn a model of the mapping from inputs to outputs. After the model is learned, the algorithm can generate outputs for new inputs. Unsupervised learning algorithms do not require a training set, but instead, learn a model of the input data by detecting patterns in it. Unsupervised learning algorithms can be used to discover structure in data or to cluster data into groups. Finally, self-supervised learning algorithms require only a training set of input data and the desired outputs are not provided. Instead, the algorithm learns a model of the input data and the desired outputs by detecting patterns in the data. More specifically, in the "pretext task" the algorithm generates "pseudo-labels" itself and then supervised training is carried out on these newly generate labels. Then, the model is fine-tuned with the specific original task ("downstream task").

It has already been discussed how the use of EEG for epilepsy-related tasks has seen a huge rise in the last three decades (Figure 3.1). Machine Learning and Deep Learning algorithms and their capability to process large amounts of data have been key players in this process, enabling the discovery and extraction of usable knowledge in a field that has always required experienced professionals.

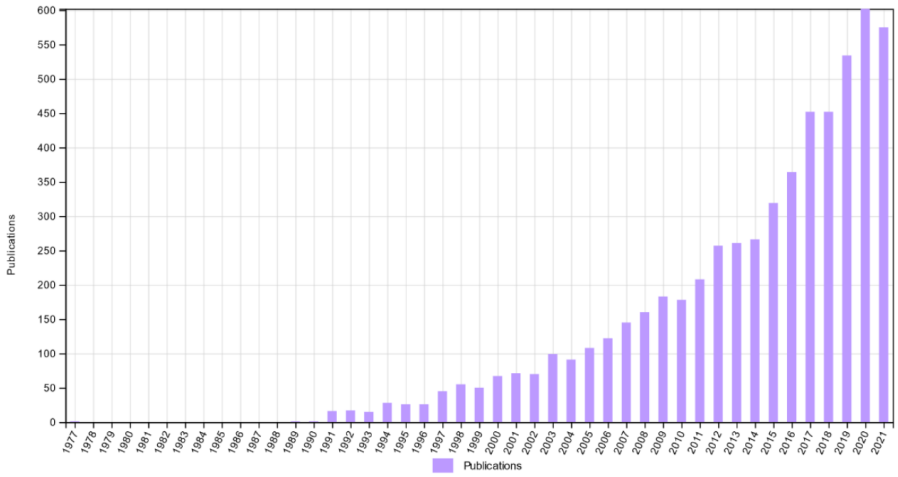

Figure 3.1. Number of articles on seizure detection, prediction and classification published since the 1970s till 2021 as reported by Web of Science [4]

3.1 Supervised learning

Several supervised learning methods of the state-of-the-art exploit the so-called "shallow" learning. Shallow Learning represents all the machine learning algorithms the techniques that are not "deep learning". These methods usually belong either to the broad class of traditional machine learning models (i.e. Decision Trees, Random Forests, Support Vector Machine, Logistic Regression and so on) or to the class of neural networks with a small number (0-2) of hidden layers. At the moment, these are the approaches that can reach the highest performances in the seizure detection task with EEG, even if, just recently, some unsupervised and self-supervised approaches challenged them with the same and, in some specific cases, even better performances across several datasets. It must be said that there are also some exceptions in the supervised learning scenario, with also some non-shallow architectures that have shown great performance recently.

Most of the effort with these supervised learning models has been laid down on the research of the most appropriate features to use and the most adequate pre-processing techniques: these two steps can give a huge advantage when considering the same model. Extracted features are used to train supervised machine learning algorithms to identify whether a given EEG segment contains a seizure or not. These algorithms employ both shallow models, including support vector machines, decision trees, and nearest neighbour

methods, as well as deep learning models, including convolutional neural networks (CNN). Another important aspect that has been taken into consideration is the deployment of models on ultra-low-power wearable devices. In this context, of course, inference execution time and power consumptions are two other crucial aspects to keep in consideration when assessing the feasibility of real-time continuous monitoring. An example of methods where both of these constraints are addressed can be found in [18], where the authors explore different classification approaches (Support Vector Machines, Random Forest, Extra Trees, AdaBoost) and different pre/post-processing techniques to maximize sensitivity while guaranteeing no false alarms. Other works focused on defining a new interpretable and highly discriminative feature for EEG: specifically, [19] proposed approximate zero-crossing (AZC), a feature obtained by applying a polygonal approximation to mimic how the brain selects prominent patterns among noisy data and then using a zero-crossing count as a measure of the dominating frequency. There a several works that focused on different pre-processing techniques that modified the type of data in the input of the model. In [20], the authors proposed the use of spectral graphs to extract spatial-temporal patterns for seizure detection. In [21], they used the wavelet transform, which has been applied to the time-frequency domain for the detection of epileptic activity. In [22], the authors proposed an effective feature extraction algorithm named discrete short-time Fourier transform (DSTFT), which is an adaptive generalization of the classical short-time Fourier transform (STFT). In [23], the authors apply the Kriging methods on EEG signals in a wearable system configuration to reduce the latency in real-time epileptic seizure detection. In [24], the same Kriging methods have been used for the application of early seizure detection. The authors in [25] use discrete wavelet transform and statistical features to apply preprocessed EEG signals to a neural network classifier to detect epileptic seizures. A very interesting work [26] explores knowledge distillation for IoT wearable devices. Knowledge distillation is a procedure for model compression, in which a small (student) model is trained to match a large pre-trained (teacher) model. The student model requires only the ECG signal and can be applied on low-power wearable IoT platforms to perform epileptic seizure detection, while the huge teacher model has been trained offline on EEG signals. Another interesting work that differs a lot from the previous ones is [27], where authors, leveraging the promising ability of transformers in capturing long-term raw data dependencies in time series, present a compact transformer model for more adaptable seizure detection obtaining great performances on a smaller dataset with respect to the original one.

For what concerns deep learning-based supervised seizure identification methods, they have lately shown state-of-the-art performances, reducing the need for manual feature extraction, as in [28], [29]. Deep models improved even more in combination with long-short term memory (LSTM) networks to aid time-series modelling [30], adversarial training to generalize identification across patients [31], autoencoder-based feature extraction [32] and attention mechanisms [33]. Table 3.1 summarizes all the information in the literature on supervised methods for seizure detection or prediction that has been cited in this section. Specifically, the paper, the year, the dataset, some metrics performances and a short comment on the method used are presented.

	Year	Dataset	Performance				Method
			Spec	Sens	FP/h	Acc	
[18]	2021	CHB-MIT	99.1	85.3	3.6		SVM, RF, Extra Trees, AdaBoost
[19]	2022	CHB-MIT		82.3			Approximate Zero Crossing Features
[20]	2018	CHB-MIT (only 18 patients)		98			Temporal Synchronization of EEG Signals
[21]	2018	private	99.5	92.1			Wavelet-based directed transfer function
[22]	2014	Bonn	93.8	99.2			Rational Discrete Short-Time Fourier
[23]	2018	open-source dataset	100	94.74			Simple, Ordinary, Universal Kriging
[24]	2020	CHB-MIT (5 patients)		87.6		93.47	Ordinary Kriging Method in IoT
[25]	2019	open-source dataset				98.5	DWT and DNN classifier
[26]	2022	EPILEPSIAE	94.8	85.7			Knowledge Distillation
[27]	2022	CHB-MIT (7 patients)	100	86.6			Transformer in real-time on MCUs
[28]	2018	Bonn	90	88.7			Deep CNN
[29]	2020	Bonn				98.5	One-dimensional DNN
[30]	2021	CHB-MIT	99.9	99.9	0.03		Long short-term memory network
[31]	2020	TUH EEG				0.805	DNN with attention mechanism
[32]	2018	Bonn, Freiburg, Kaggle				98.85	Recurrent autoencoder
[33]	2020	CHB-MIT, Bonn, TUSZ	96.05	92.41			Channel-embedding squeeze-and-excitation

Table 3.1. State-of-the-art performances of the supervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column.

3.2 Unsupervised learning

Despite the success of supervised learning methods, the problem is that they require expert labels indicating EEG segments that contain seizures, which are difficult to obtain due to the stochastic nature of EEG. For this reason, in recent years, some unsupervised seizure detection methods have been explored and proposed in the literature. Most of the success of these methods is due to the application of the anomaly detection approach. Anomaly detection is the task of identifying test data that do not fit the normal data distribution learned during training. The concept of anomaly detection is

applied in various domains such as video analysis and remote sensing: the strength of this approach is that can overcome the imbalance problem. Thus, each one of these methods does not use any data containing seizures during training: in this way, the model can deeply understand the nature of "normal" data, where normal is the most occurring scenario in which there are no seizure events. Recently, [34] implemented an unsupervised deep learning method for seizure identification on EEG, however with some manual feature extraction required before training. Specifically, they preprocess EEG to extract spectrogram images and train a GAN on the spectrograms that do not contain seizures. For each spectrogram at testing time, they have to search for the latent GAN input that leads to the smallest loss value and use the corresponding generated spectrogram for seizure identification. As the GAN is trained with non-seizure activity, test spectrograms that significantly differ from the spectrograms generated by GAN are successfully identified to contain seizures. Based on this approach, [35] applied a fully-unsupervised VAE on raw EEG, without the need for any pre-processing and feature extraction. The seizure identification metric is based on reconstruction errors made by the VAE, which is trained on non-seizure activity and does not require sophisticated min-max optimization such as GAN training. Another anomaly detection approach is proposed in [36], where an autoencoder involving a transformer encoder is trained via an unsupervised loss function, incorporating a novel geometric masking strategy uniquely designed for multivariate time-series data, such as EEG. Seizures are then identified by reconstruction errors at inference time. Inspired by [37], [38] proposed a two stages framework, where the first stage aims to train a CNN-classifier as a feature extractor and the second stage can adopt any existing generative or discriminative method to establish an anomaly detector based on the feature representations from the well-trained feature extractor. [39] proposes a hybrid system integrating an unsupervised module that serves to quickly locate the determinate subjects (or the "easy" one) and the indeterminate subjects, with a more robust seizure detection module for the indeterminate subjects using an EasyEnsemble algorithm, a class-imbalance learning method, that can potentially decrease the generalization error of the seizure-free segments. Table 3.2 summarizes all the information in the literature about unsupervised methods for seizure detection or prediction that has been mentioned in this section. Once again, as for the supervised case, the paper, the year, the dataset, some metrics performances and a short comment on the method used are presented.

	Year	Dataset	Performance				Method
			Spec	Sens	FP/h	Acc	
[34]	2020	private		96.3	0.14		Generative adversarial network
[35]	2022	CHB-MIT, UPenn, TUH				68.8	Variational autoencoder
[36]	2023	CHB-MIT, UPenn, TUH				87.1	Multi-variate time-series transformers
[38]	2022	CHB-MIT, UPenn, private			0.16	90.1	CNN + anomaly detection
[39]	2022	CHB-MIT	92.57	95.55		92.62	Isolation Forest and EasyEnsemble

Table 3.2. State-of-the-art performances of the unsupervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column.

3.3 Self-supervised learning

Even if deep learning is the dominant approach for developing seizure prediction methods, the success of its applications relies heavily on the availability of annotated datasets. Medical datasets carefully annotated by experts are hard to create at scale. Unlike labelled datasets, unlabelled datasets can also be leveraged to build self-supervised models that learn complex structures in the data via new supervised-learning tasks. It has already been said as in the first phase of training, the models are trained on a preparation task, referred to as a 'pretext task', but because the data used for such pretext tasks are unlabelled, the trained model (often referred as the "featurizer") cannot yet solve the task which is supposed to address. In the second phase of training, the featurizer is trained on a dataset of explicit labels. This enables the model to incorporate its knowledge of the data to perform the relevant medical task.

The recent success of this type of learning in Computer Vision and, mainly, in Natural Language Processing has drawn a lot of attention also in the Medical AI field. Even if the use of self-supervised learning in seizure prediction methods is a relatively new research area, different approaches have been tested. [40] investigated self-supervised learning to learn representations of EEG signals., exploring two tasks based on temporal context prediction as well as contrastive predictive coding. Linear classifiers trained on SSL-learned features (or "embeddings") consistently outperformed purely supervised deep neural networks in low-labelled data regimes while reaching competitive performance when all labels were available. Similarly, [41] approach concerns extracting features from a single channel at a time as opposed to considering all channels simultaneously. In [42] a series of scaling transformations are performed on the original EEG data to generate self-labelled

scaled EEG data, where different labels correspond to different scaling transformations. Then using the self-labelled normal EEG dataset, a multi-class classifier can be trained to accurately predict the scaling transformations on new normal EEG data, but not accurately on abnormal (epileptic) EEGs.

A complete and recently published overview of the applications of self-supervised learning to Biomedical signals is carried out in [43]. Some of those are specifically related to the EEG field, but not strictly to seizure detection, as in [44], [45] and [46] for example. One of the most interesting studies in the self-supervising approach for EEG data is [17]: the authors studied how to adapt techniques and architectures used for language modelling that appear capable of ingesting awesome amounts of data toward the development of encephalography modelling with DNNs in the same vein. Specifically, they adapted an approach effectively used for automatic speech recognition, which similarly to language models uses a self-supervised training objective to learn compressed representations of raw data signals, combining encoders and transformers [5]. This study is the starting point of the following thesis and is deeply addressed and explained in 4.2. Generally, although recent research has shown promising results using self-supervised learning techniques on EEG data, it is still not clear which is the most promising direction (in terms of architecture, pretext tasks and adaptation on downstream tasks) to follow to fully exploit its potential. Table 3.3 summarizes the aforementioned methods. Notice that, differently from supervised and unsupervised approach, there is no *Methods* column, but rather the *Downstream tasks* one, where the task addressed are presented (more information about Self-supervision in Section 2.3). For this reason, the *Performance* column, in this case, does not refer to the seizure detection task, but only to downstream tasks (more specifically, the first mentioned).

Authors	**Year**	**Dataset**	**Performance (AUC)**	**Downstream task**
[40]	2021	Physionet, TUH		Sleep monitoring and pathology screening
[41]	2020	SEED, TUH, SleepEDF		Emotion recognition, Normal/Abnormal EEG, Sleep monitoring
[42]	2020	UPenn	0.941	Anomaly detection
[44]	2021	SleepEDF		Sleep monitoring
[45]	2022	TUSZ	0.875	Seizure detection and classification
[17]	2021	MMI, BCIC, ERN, P300, SSC		Brain Computer Interface

Table 3.3. State-of-the-art of the self-supervised methods taken into consideration in the preliminary study of this thesis. Notice that when more datasets are present in the *Dataset* column, the performance of the first dataset in order of appearance is shown in the *Performance* column. Notice also that only results that regard the seizure detection task are shown in the *Performance* column (or anomaly detection, since a seizure can be considered an anomaly). The other works for which it is not shown any results are not related to seizure detection and thus, in order to avoid confusion, nothing is shown in the *Performance* column.

Chapter 4

Matherial and Methods

The following chapter presents all the contributions of the thesis. It is organized into two different sections: one for the unsupervised learning methods and one for the self-supervised. The first part (4.1) focuses on testing and validating two unsupervised methods proposed by the literature. No further contribution is made other than the validation of these two methods on the dataset chosen for the thesis. The results of these two first unsupervised methods are shown in Appendix A. On the other hand, the second part (4.2), starting from a model proposed by the literature, explores self-supervised learning and how the model can be adapted for the seizure detection task. In this section, the problem is tackled by considering different points of view. First of all, it is important to understand if the pretext task produces some valuable knowledge. Once that question is answered, the objective is then to find the best-suited model for the task considering 4 different dimensions: size, architecture, data and post-processing.

4.1 Unsupervised approaches for seizure detection

The focus of this first section is to explore and validate different unsupervised approaches proposed in the literature that address the problem of seizure detection as an anomaly detection problem. First, an approach based on Variational Autoencoders (VAEs) is proposed, then one based on Transformers that learn to reconstruct downsampled masked signals, and lastly one that combines a Large Convolutional Neural Network with a general anomaly detection method. The idea is to start with fully-unsupervised methods and

then shift towards self-supervision which is thoroughly addressed in the next section 4.2. As already discussed in 2, this type of learning addresses the problem of seizure detection as an Anomaly Detection problem. A general scheme of this approach is drawn out in Figure 4.1.

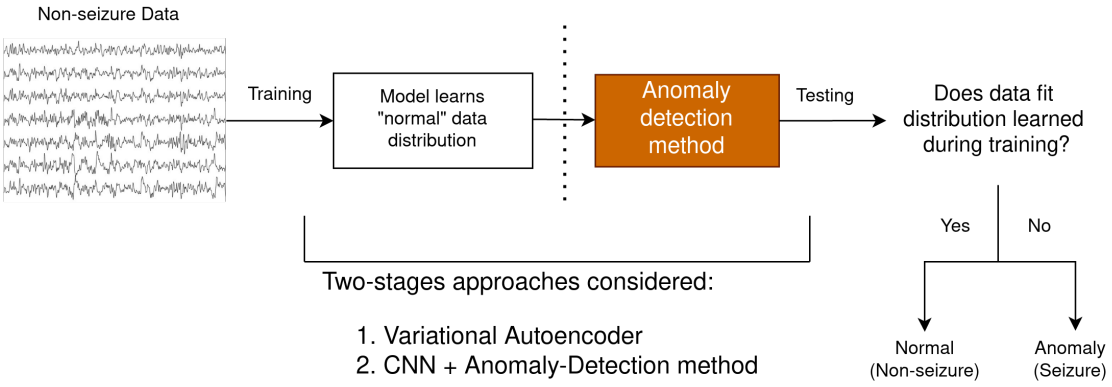

Figure 4.1. The general approach of unsupervised learning for the seizure detection task on EEG data

4.1.1 Variational Autoencoder

In this section, the model and approach proposed in [35] are explored and tested. The aim is to design an unsupervised method that does not rely on ground-truth expert labels during learning and can identify the existence of seizures in a given EEG recording. To this end, a variational autoencoder (VAE) [47] neural network architecture is employed. In order to fully understand how it works, it is crucial to introduce the concept of Autoencoders. An Autoencoder is a type of neural network that can learn to reconstruct images, text, and other data from compressed versions of themselves. An autoencoder aims to learn a lower-dimensional representation (encoding) for higher-dimensional data, typically for dimensionality reduction, by training the network to capture the most important parts of the input data. The autoencoder architecture consists of 3 parts:

1. an Encoder: a module that compresses the input data into an encoded representation that is typically several orders of magnitude smaller than the input data;

2. a Bottleneck (latent space): a module that contains the compressed

knowledge representations, which ensures that only the main structured part of the information can go through and be reconstructed;

3. a Decoder: a module that helps the network decompress the knowledge representations and reconstructs data back from its encoded form.

The training of such type of architecture has usually the goal of finding the encoder-decoder couple that minimises the reconstruction error. The limitation of this type of architecture is that Autoencoder is solely trained to encode and decode with a loss as low as possible, no matter how the latent space is organised. They learn to generate compact representations and reconstruct their inputs well, but the latent space they convert their inputs to and where their encoded vectors lie, may not be continuous, or allow easy interpolation. The consequence is that a trained Autoencoder with very low reconstruction loss can lack the ability to generalize well and It is straightforward to understand that during training, the network may take advantage of any overfitting possibilities to achieve the reconstruction task as faster as possible.

A Variational Autoencoder tries to address the overfitting risk of regular Autoencoders: indeed, it can be defined as an Autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable the generative process. Just like a standard Autoencoder, a Variational Autoencoder is an architecture composed of both an encoder and a decoder that is trained to minimise the reconstruction error between the encoded-decoded data and the initial data. To introduce some regularisation of the latent space, instead of encoding an input as a single point, the input is encoded as a distribution over the latent space. The training process is defined as follows:

- an input is encoded as a distribution over the latent space;
- a point from the latent space is sampled from the distribution that characterizes the latent space;
- the sampled point is then decoded and the reconstruction error is computed.

As for the Autoencoder, the reconstruction error is backpropagated through the network for the gradient descent-based training. Generally, the loss function is then composed of a reconstruction term, to make the encoding-decoding scheme efficient, and a regularisation term, to make the latent space

regular and avoid overfitting. A simple architecture of a Variational Autoencoder is shown in Figure 4.2. Differently from the traditional loss function

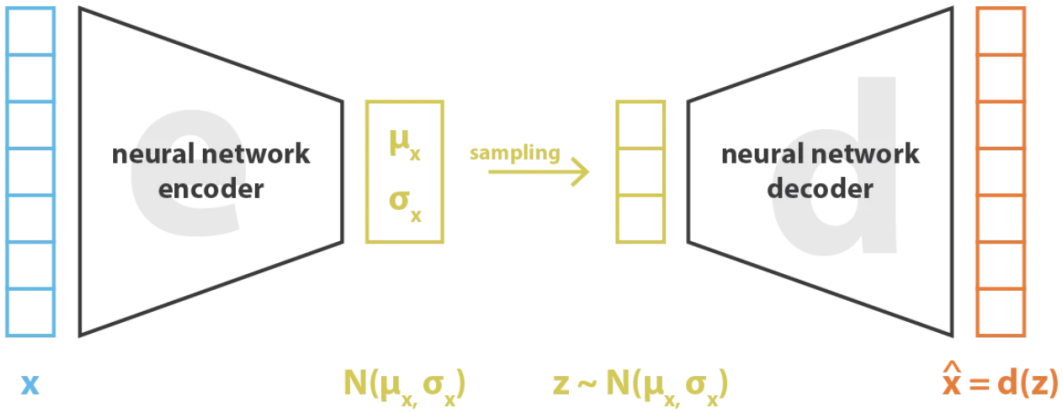

Figure 4.2. The encoder-decoder scheme of a Variational Autoencoder. x is the input data, μ_x and σ_x are respectively the mean and the standard deviation of the Gaussian distribution $N(\mu_x, \sigma_x)$ to which z, the latent representation, belongs. $\hat{x}$ is the reconstructed data, which is a function d of the decoder [48]

.

of Variational Autoencoders, [35] defines a Sparcitiy-enforcing loss function, that replaces the second term that usually performs maximum-likelihood estimation of the generative model parameters (decoder), with the l_1-norm of the reconstruction error. The architecture is then trained on EEG recordings that do not contain seizures, using the sparsity-enforcing loss function to suppress EEG artefacts. In this way, the learned latent features can capture the non-seizure activity rather than a seizure. At inference time, each reconstruction from the trained VAE is compared with the corresponding input recording. As training captures non-seizure activity, recordings with no seizures are expected to be reconstructed with low error. Meanwhile, a larger reconstruction error with respect to the input recording indicates evidence of a seizure.

The encoder architecture is built with Convolutional layers (with 4x4 filters), followed by batch-normalization and fully-connected (FC) layers to extract latent features. The decoder contains Convolutional transpose and FC layers in order to upsample the latent representations and reconstruct the original signal. The architecture is shown in Figure 4.3.

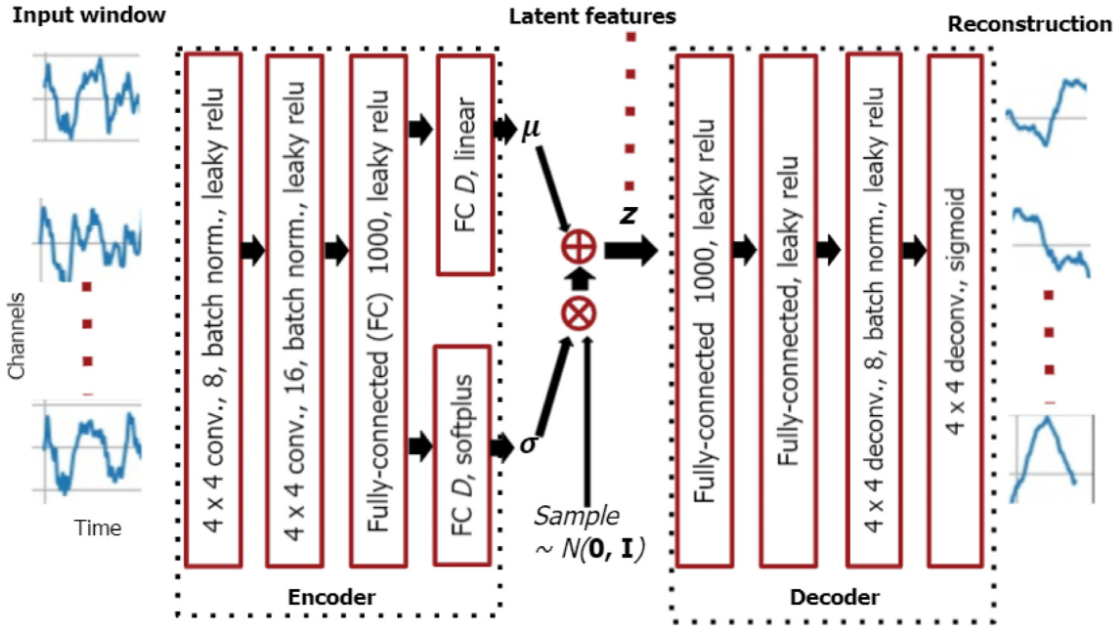

Figure 4.3. Variational Autoencoder architecture for non-seizure EEG data reconstruction [35]

4.1.2 Combining CNN with Anomaly Detection Methods

The objective of this section is to validate the approach proposed in [38], which makes use of available normal EEGs and expert knowledge about abnormal EEGs to train a more effective feature extractor (first stage) for the subsequent development of anomaly detector (second stage). As already discussed in the introduction, the problem of seizure detection is addressed by considering only normal (or without seizures) EEG data during training. However, the interesting contribution of this work is that it proposes a new strategy to train a feature extractor that can extract features of both normal and abnormal EEG data. Specifically, considering that abnormal EEGs are characterized by increased wave amplitude or temporally slowed or abrupt wave signals, two special transformations are designed to generate simulated abnormal EEG data. These two transformations work on both the amplitude and the frequency of the EEG signal. Specifically:

- one temporally increases or decreases the amplitude of normal EEG;
- one temporally increases or decreases the frequency of normal EEG.

Using these simple transformations and based on multiple normal EEG data, two classes of self-labelled abnormal data will be generated, with one class representing anomaly in amplitude, and the other representing anomaly in frequency. These two newly generated types of signals are then added to the original normal EEG class to form a 3-class dataset for the training.

For what concerns the feature extractor, a specific CNN classifier based on ResNet34 backbone is designed. Since adjacent EEG channel data do not indicate spatial proximity between two brain regions, one-dimensional (1D) convolutional kernels to learn to extract features from each channel. Differently from previous studies [49], kernels of longer size (1x7) are considered to take into account that lower-frequency features may last for a a longer period and therefore would not be captured by shorted kernels over multiple convolutional layers. Additionally, a shortcut branch of one convolutional layer is added from the output of the first convolutional layer to the second last layer: in this way, a short anomaly is not completely omitted after the down-sampling effect of several layers, and some potential correlation across all the channels may be captured. The architecture just described of the CNN feature extractor is shown in Figure 4.4

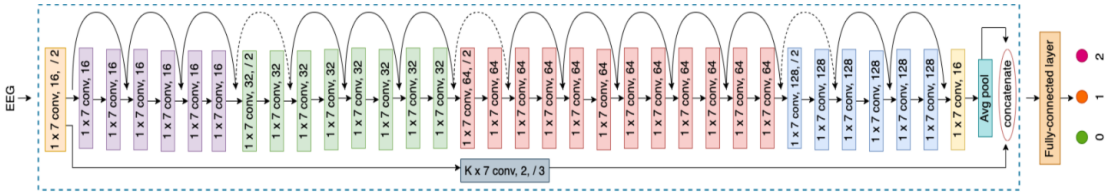

Figure 4.4. Two-branch architecture of the feature extractor. The blue dotted line highlights the part of the network that is used to extract representations of data on which the anomaly detection method is then trained

For what concerns the second stage of the architecture, a generative approach is applied based on feature representations of normal EEG segments produced by the CNN feature extractor discussed in the previous paragraph. As a generative approach, the multivariate Gaussian distribution $G(\mu, \Sigma)$ is used to describe the distribution of normal EEG available during training. The mean μ and the covariance matrix Σ are directly estimated from the feature vectors of all normal EEGs in the training set, with each vector being the output of the feature extractor given a normal EEG input. With the Gaussian model $G(\mu, \Sigma)$, the degree of abnormality for any new EEG data z can be estimated based on the Mahalanobis distance between the mean μ

and the feature representation $f(z)$ of the new data z, as in:

$$A(z) = \sqrt{(f(z) - \mu)^\top \Sigma^{-1} (f(z) - \mu)} \tag{4.1}$$

where $A(z)$ is the abnormality score: a large value of $A(z)$ means that z is more likely to be an abnormal segment, and vice versa.

4.2 Self-supervised approaches for seizure detection

The focus of this second section presents all the tested approaches for self-supervision learning for seizure detection on EEG data. The idea is to exploit knowledge learned from a huge amount of unlabelled data to improve the performance on a smaller and labelled dataset. To fully understand what has been implemented, 2.3 presents all the notions to understand before reading the following Section. 4.2.1 presents the architecture exploited for the self-supervision task carried out in the thesis. And finally 4.3 shows the extensive research on the implementation of the model presented in the previous sections.

4.2.1 BENDR

Following the success of the architectures and Self-supervised approaches presented in 2.3 in NLP, the objective of this work is to explore the potential of these methods in a different data domain. The idea of this work is to exploit the huge availability of unlabelled EEG data to pre-trained large language-inspired models in an unsupervised way and fine-tune it on the seizure detection task on a smaller and labelled dataset. BENDR [17] has been proposed with the intent of effectively using those techniques on EEG data to learn compressed representations of the raw data signals. The goal is to address and overcome the challenges of DL on raw EEG data, focusing on the lack of generality and the struggle to learn lower-level features that are transferable to unseen and different tasks. To this end, BENDR employs the same approach seen with masked language models (as BERT), but uses individual samples of raw EEG data rather than text tokens. The same unsupervised learning approach to extract useful representations from high-dimensional data (Contrastive Predictive Coding), seen in wav2vec 2.0, is used to encode raw EEG segments as a sequence of learned vectors called

BErt-inspired Neural Data Representations. After the unsupervised pre-training on a huge amount of EEG data, the model can then be fine-tuned on a downstream (related) task.

The BENDR architecture for the **pre-training task** is similar to the wav2vec 2.0 one (Figure 2.6), :

1. A **feature encoder** that processes 20 channels of raw EEG recording X and converts them into a sequence of latent representations to capture the underlying properties of data $b_1, ..., b_T$ for T timesteps. For the initial configuration of BENDR, six convolutional blocks are considered, each containing:

   - a 1D temporal convolution with 512 filters, each block with receptive fields of 2, except for the first one, which has 3, and strides match the length of the receptive field for each block;
   - a Dropout layer that randomly zeros out entire channels. Each channel will be zeroed out independently on every forward call with probability p using samples from a Bernoulli distribution, this will reveal to be a crucial contribution to reducing overfitting;
   - a GroupNorm that divides the channels into groups and computes the mean and the variance for normalization within each group;
   - a GELU activation function

   The result of the downsampling of these six blocks is the latent space representations, called BENDR, B. Notice that the effective sampling frequency of BENDR is 96 times smaller ($\approx 2.67Hz$) than the original sampling frequency ($256Hz$).

2. following the approach of MLM, for each sample 10 contiguous sequences are **masked** with probability $p_{mask} = 0.065$, such that, for each sample, the likelihood of being the beginning of a contiguous section is p_{mask}, and overlap is allowed over.

3. A **Transfomer encoder** with 8 layers and 8 heads, model dimension of 1'536 and internal feed-forward dimension of 3'076 takes as input the masked output B. Some modifications are made to the wav2vec 2.0 transformer encoder:

   - relative positional embedding is learned during training deploying an additive grouped convolutional layer with the receptive field of

25 and 16 groups followed by a GELU activation. In this way, the model should be sequence-length independent, so the downstream task does not have to be the same length as in the pre-training phase. The learned embeddings are then added to the input features B and a layer normalization is applied

- internal batch normalization layers are removed
- T-fixup [50] is used to initialize the internal layers to avoid the vanishing gradient effect and the exploding gradients effect.
- LayerDrop and Dropout layers are added during pre-training to reduce overfitting.

To use the transformer for classification a fixed token is added to the beginning of B before passing it to the transformer, thus being able to recognize the start of the sequence. The transformer outputs the final vectors C.

As for wav2vec 2.0, the training objective is to produce outputs C that are as similar as possible to the unmasked input B at position t. For this reason, the same contrastive loss function is shown in Equation 2.13:

$$L_t = \text{-log}\frac{\exp(cossim(c_t, b_t))/k}{\sum_{b_i \in B_D} \exp(cossim(c_t, b_i))/k} \tag{4.2}$$

where c_t is the output of the transformer and b_t is the un-masked BENDR at position t. B_D is the set of 20 uniformly selected distractors. The sensitivity of the cosine similarity function is set to $k = 0.1$. The BENDR architecture just described is shown in Figure 4.5.

When **fine-tuning** the model for the seizure detection downstream task the architecture of BENDR is modified. Specifically:

- the masking step is removed: thus, the output of the convolutional feature encoder B (i.e. BENDR) is passed directly to the Transformer;
- a Classification layer that takes as input the first output token of the transformer is added. The classification layer is composed of a Linear layer followed by a Softmax activation to find the class with the highest probability of occurring.
- the weights of the network that are updated during this fine. Then fine-tune the entire model, both the pre-trained and the new classification layer, to classify the downstream targets.

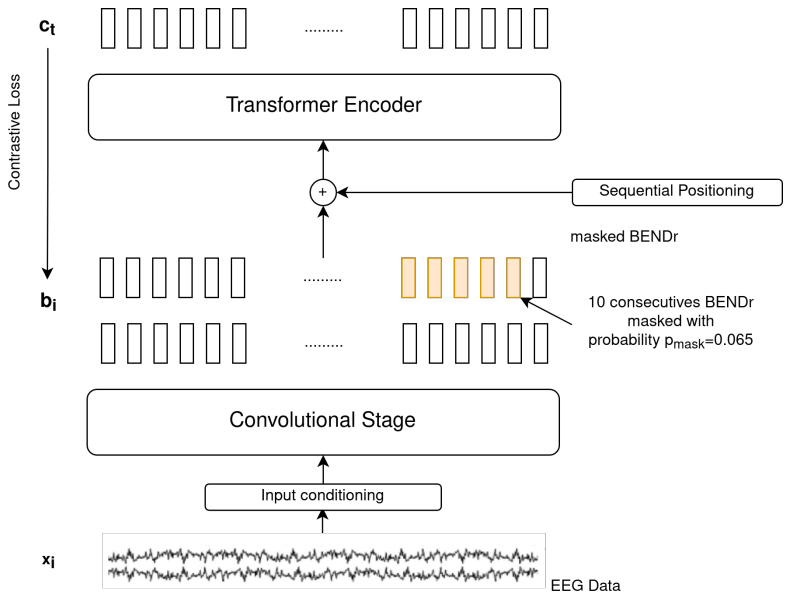

Figure 4.5. Pre-training architecture of BENDR

During fine-tuning, the model objective does not depend anymore on the production of output that is similar to the unmasked input. The problem is now, given an input EEG segment at time t, to recognize if the segment is a normal one (non-seizure) or an abnormal one (seizure). For this reason, different loss functions will be considered to achieve this goal. **Cross entropy loss** is a very common metric used to measure how well a classification model performs. The objective of training is to estimate the parameters of the Maximum Likelihood Estimation paradigm, to learn the underlying data distribution. Thus, the loss function is used to evaluate how well the model fits the data distribution. In this context, cross-entropy can measure the error (or difference) between two probability distributions. The cross-entropy loss can be defined as:

$$l = -\sum_{c=1}^{N} y_c \log(p_c) \tag{4.3}$$

where $c \in N$ is the class being predicted, p_c is the predicted probability of class c and y_c is the ground truth of class c. In the case of seizure detection that counts two classes (seizure vs non-seizure), the Cross-Entropy Loss becomes:

$$l = -(y\log(p) + (1-y)\log(1-p)) \tag{4.4}$$

where $y \in [0,1]$, meaning non-seizure or seizure. The used architecture of

BENDR for the fine-tuning task is shown as well in Figure 4.6.

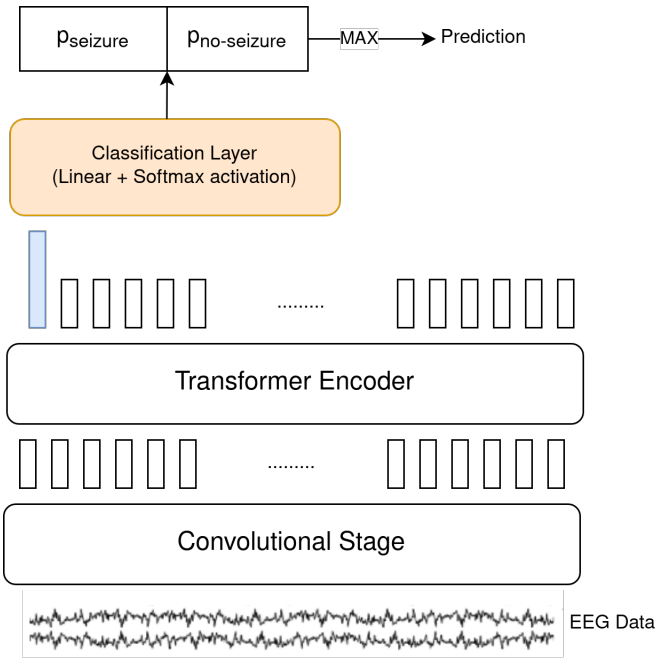

Figure 4.6. Fine-tuning architecture of BENDR

When fine-tuning, a different and smaller BENDR architecture has been tested as well. In particular, this architecture, referred to as Linear, ignores the pre-trained transformer entirely and uses only the pre-trained convolutional stage (i.e., only use the so-called BENDR, thus the embeddings produced by the Convolutional stage). It basically creates a consistent-length representation by dividing the BENDR into four contiguous sub-sequences, averaging each sub-sequence and concatenating them. In this phase, the choice of the four sub-sequences is arbitrary. Again, as for the previous architecture, a new linear layer with softmax activation is added to classify the downstream targets. The scheme of this Linear architecture for fine-tuning is shown in Figure 4.7.

4.2.2 MAEEG

In MAEEG (Masked-Autoencoder for EEG Representation Learning) [51] authors further explore representation learning using reconstruction-based SSL on EEG data. Starting from BENDR, they propose a different pre-training technique to learn meaningful EEG representations, obtaining better

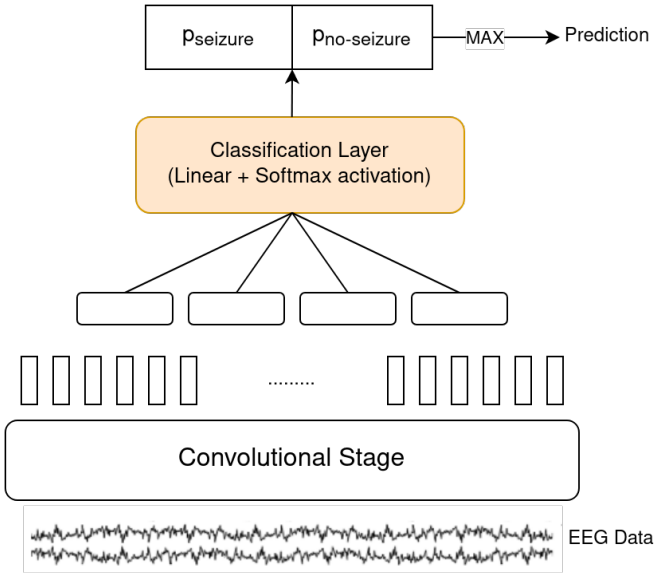

Figure 4.7. Fine-tuning architecture of BENDR Linear

performance on sleep stage classification. Another focus of this work is to explore how this different architecture behaves in the seizure detection task.

The MAEEG model has a similar architecture to BENDR 4.5, but has two additional layers to map the transformer output back to the raw EEG dimensions. Specifically, considering the output of the transformer:

- a Linear layer is added to reconstruct the signal starting from the output back to its original size in the temporal dimension (i.e. back to the original number of samples)

- a Convolutional Layer to reconstruct the signal on the spatial dimension (i.e. back to the original number of channels). This is specifically done with a 1D convolution with kernel size 1x3 and padding 1.

During the pre-training phase, the loss is changed as well. The objective is now to compare the reconstructed signal with the original signal in the output. To achieve that the 1-cosine similarity is deployed. The reconstruction loss is computed by comparing the reconstructed EEG ($\hat{x}$) and input EEG (x) signals, or:

$$l = 1 - \frac{\hat{x} \cdot x}{||\hat{x}||||x||} \tag{4.5}$$

The key difference between BENDR and MAEEG is that instead of using contrastive learning, MAEEG learns representations by minimizing the reconstruction loss. A scheme of the comparison between BENDR and MAEEG pre-training architecture is shown in Figure 4.8.

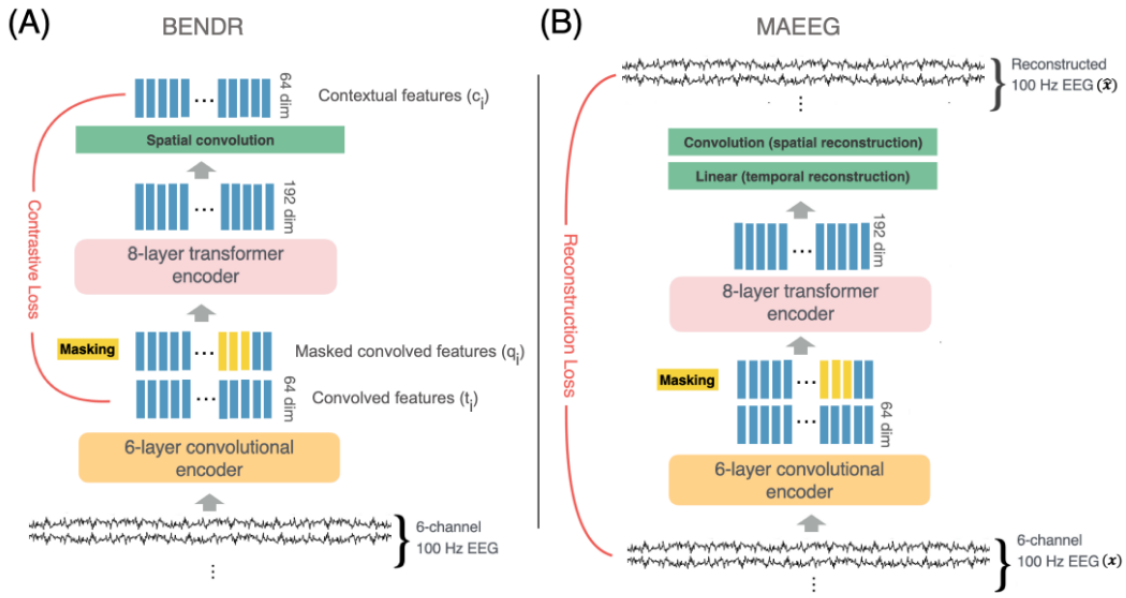

Figure 4.8. The pre-training architecture of BENDR (A) and MAEEG (B) [51]

During the fine-tuning phase, the architecture used is the same as BENDR (Figure 4.6). The two additional layers are now removed and a new Linear layer followed by a softmax activation is added to take as input the last token of the transformers. Once again, the loss used in this second task is the Cross-Entropy for Binary Classification shown in Equation 4.4. In summary, there is absolutely no difference in the fine-tuning architecture between BENDR and MAEEG. These two methods are only compared in terms of the quality of the learned features during the pre-training task.

4.3 An extensive research for fine-tuning

The objective of the thesis is to explore how self-supervised learning and large language model-inspired methods work in a different world from the NLP, on which they were initially developed. When applied to the EEG world, these models can leverage the huge amount of unlabelled data available during the pre-training task and then be fine-tuned on a more specific downstream task.

To validate the use of self-supervised learning in EEG, the BENDR model, described in 4.2.1, is implemented in a different dataset and a different task with respect to the ones on which it was developed. Once this method has been implemented for the seizure detection task, extensive research on the fine-tuning process is carried out. The model has been fine-tuned considering patient-specific training with **Leave-One-Out Cross-Validation** strategy to reduce the variability of the results. A simplified scheme for patient chb01 is shown in Figure 4.9. For each patient, one file at a time is considered a test. The remaining files are merged and split between train and validation sets with the stratified option. This means that the distribution between seizure and non-seizure in the train set and the validation remains the same.

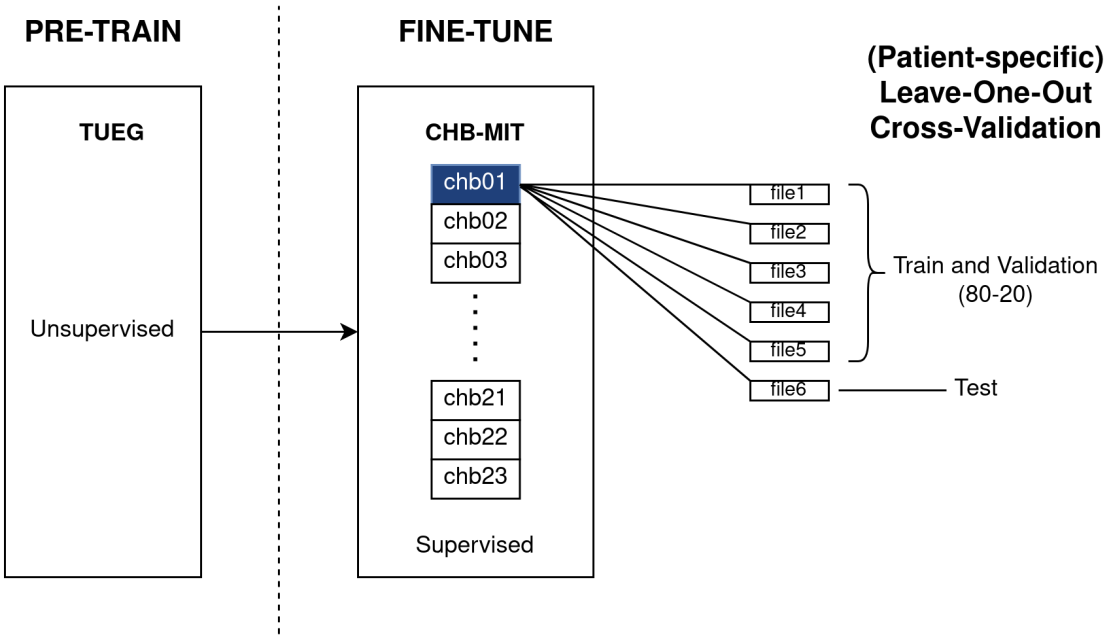

Figure 4.9. Training strategy with Leave-One-Out Cross-Validation. In the figure, it is possible to see an example for patient chb01, with file6 test set. This is repeated for all the files of the patient and aggregated results are computed by averaging the metrics on all the different test sets.

The objective of this section is to highlight the different dimensions explored during the fine-tuning of BENDR for seizure prediction. The following section introduces all the different spaces that have been searched to adapt BENDR.

4.3.1 Scalability

First of all, the scalability aspect of the model is taken into account. Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands. Specifically, in order to test this measure, the idea is to reduce and increase the size of the model and study the correlation between model size and performance. The BENDR model can count on two different modules, as shown in 4.2.1:

- a Feature encoder (Convolutional stage), composed by 6 blocks of 1D convolution followed by a Group Norm layer and a GELU activation function;
- a Transformer encoder with 8 layers and 8 heads.

The objective of this first dimension study is to understand how the behaviour of the model changes when reducing the size of the feature encoder, which means reducing the number of convolutional blocks in the model, thus reducing the ability of the model to downsample the original signal. It is important to notice that the feature encoder is part of the model responsible for the creation of the latent representations that are used during the pre-training task to learn the underlying structure of EEG data. Additionally, also the number of heads and layers of the transformer encoder is modified. The idea is to modify the number of parameters, either decreasing or increasing it, to find the best-suited model that avoids overfitting. In Figure 4.10, it is possible to see the different tested configurations (i.e. the combination of the number of convolution blocks and heads/layers of the transformer).

4.3.2 Pre-training architecture

The second dimension explored to optimize the fine-tuning of the model on a different dataset and a different task is the pre-training architecture. Specifically, a comparison between BENDR and MAEEG, the model seen in 4.2.2, is carried out to understand which is the model that is able to learn the most appropriate representations of the data. In this context, the two different pre-training architectures presented, respectively, in 4.2.1 and 4.2.2, BENDR and MAEEG are compared. However, since the pre-trained weights learned during the unsupervised pre-training task were already made available by [17] and thus it was not the focus of the thesis, it was not possible to test MAEEG as an alternative pre-training architecture, even because of

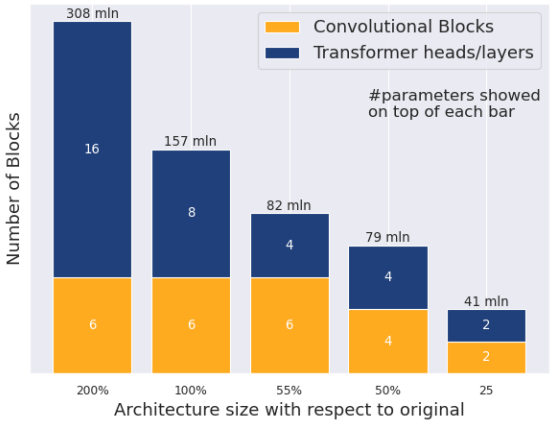

Figure 4.10. Scalability: the different architecture sizes tested

the limited resources available. Therefore, the idea is to understand if there is a benefit in exploiting not only the pre-trained weights on TUEG but also the CHB-MIT patients themselves. More specifically, the aim is, once the model has been initialized with the TUEG pre-trained weights, to run a *second pre-training* on all the patients of CHB-MIT except the one on which the model is fine-tuning. The question asked in this section is the following: is the knowledge embedded in other CHB-MIT patients' data transferable for fine-tuning patients? Is there some useful information that can be transferred between patients? To answer that question, a slightly different training strategy is carried out with respect to the one already seen in Figure 4.9. The current training strategy is shown in Figure 4.11. Using this strategy, it is possible to compare the two architectures in the so-called second pre-training task, and which of the two can extract the most useful information for the fine-tuning phase. It is important to remember that these two architectures differ mainly in the training objective, briefly:

- BENDR during pre-training compares the output of the transformers with the representation learned by the first convolutional stage;

- MAEEG, on the other hand, compares the reconstructed signal, which is the results of the Linear and the convolutional layer on top of the transformer, with the original input itself.

Of course, both of these two architectures are initialized with the pre-trained weights on TUEG. Results are shown in Table 5.6.

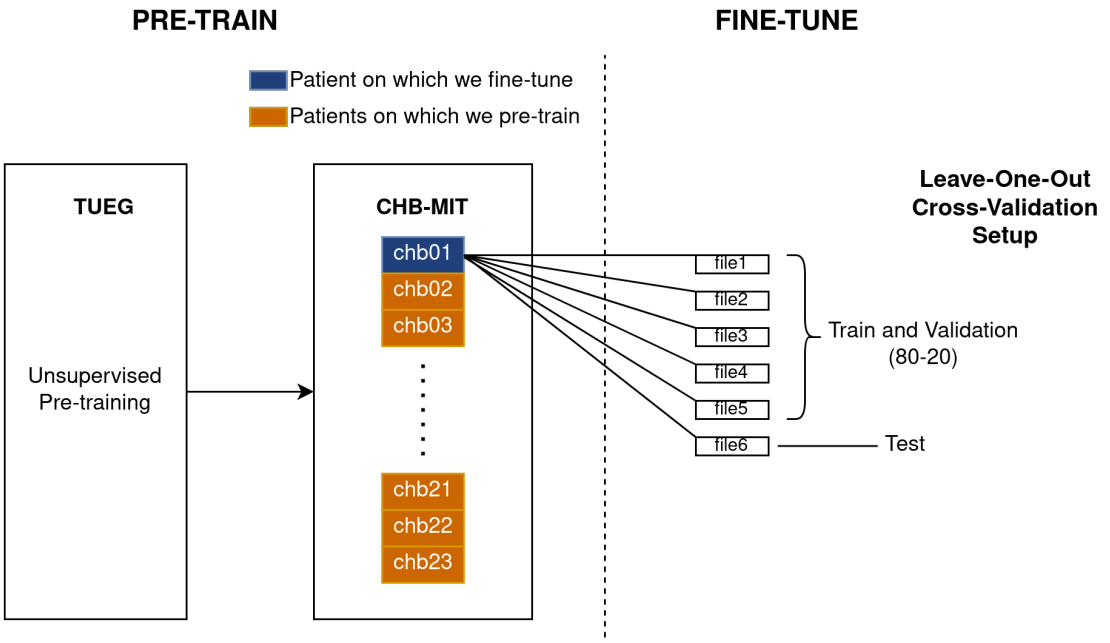

Figure 4.11. Alternative training strategy implemented to test BENDR vs MAEEG

4.3.3 Processing techniques

Pre-Processing techniques

Several **pre-processing** techniques are explored to further optimize the fine-tuning process. First of all, the effect of a **Butterworth Bandpass filter** is studied. The Butterworth filter is a type of signal processing filter designed to have a frequency response that is as flat as possible in the pass band (i.e. no ripple).

More specifically, when working on the CHB-MIT dataset, EEG recordings are pre-processed before being passed to the model. First of all, a mapping of the CHB-MIT channels is carried out in order to match the number of channels used during the pre-training phase on TUEG. Indeed, even if data from the CHB-MIT dataset has 23 channels, only 20 were used during the pre-training. Thus, only 20 EEG channels of CHB-MIT are considered and renamed accordingly to TUEG naming. Then, the recordings are filtered via a 5^{th} order Butterworth bandpass filter with a frequency range of 0.5–50 Hz. Additionally, some EEG segments after the end of a seizure are disregarded. This is a common practice suggested by experts, because it may happen to have some spikes in data after a seizure that should not be accounted as such.

In detail, the 15 minutes that follow the seizure end are disregarded.

Then, to consider samples with the same length, sliding windows are extracted over each recording, where each window contains T time points with no overlapping between consecutive windows. Following approaches in the literature ([18] and other state-of-the-art approaches), 4s EEG windows (or segments) are considered: considering a sampling rate of 256 Hz, then T=1024 samples are considered. This means that the input of the model is a single EEG segment of size $[C, T]$, or when considering batches $[B, C, T]$, where B is the batch size considered, $C = 20$ is the number of EEG channels and $T = 1024$ are the number of samples of the window.

Different **oversampling** techniques are tested to compensate unbalancing nature of EEG data. The most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling TEchnique (SMOTE), which works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. Standard SMOTE works by utilizing a k-nearest neighbour algorithm to create synthetic data. Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbours for that example is found. A randomly selected neighbour is chosen and a synthetic example is created at a randomly selected point between the two examples in the feature space. In addition to SMOTE, Weighted random sampling (WRS) is considered as well. WRS asks for sampling items (elements) from a set such that the probability of sampling item i is proportional to a given weight w_i, which is proportional to the number of items i in the considered dataset with respect to the number of items j, with $j \neq i$. Considering the seizure detection with EEG scenario, with two different classes, seizure and non-seizure, S, NS, with $i \in S$ and $j \in NS$, with WRS it is guaranteed that the lowest represented class, the seizure one, will be sampled enough time to compensate the unbalancing problem, thus resulting in a balanced data. The main difference between these two oversampling methods is that SMOTE generates new data based on the actual real data, while WRS just uses the existing data.

Finally, different **normalization** techniques on all windows are tried to aid the convergence of gradient-based training. Specifically, the min-max normalization technique is tested, where given an input EEG segment x, the input is scaled as:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.6}$$

where x_{max} and x_{min} are respectively the maximum value and the minimum

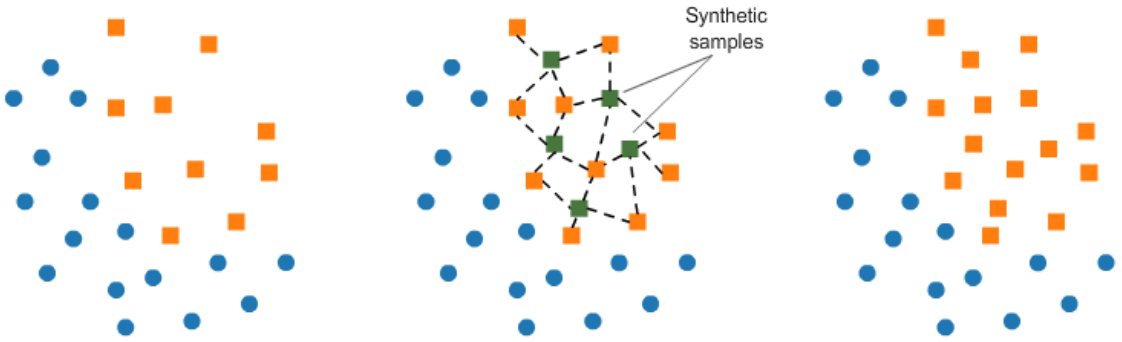

Figure 4.12. SMOTE [52]

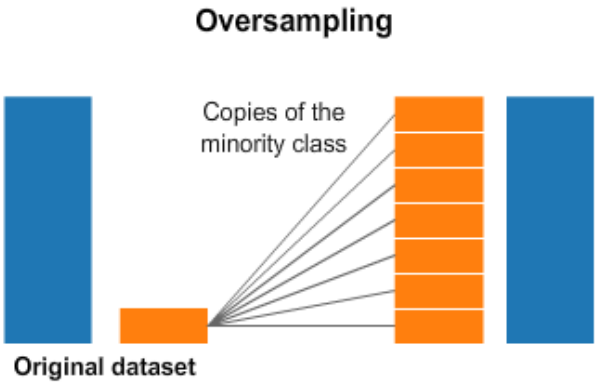

Figure 4.13. WRS [53]

value computed on the training set. Mean-standard deviation normalization is tested as well. Specifically, given an input EEG segment x, the input is scaled as:

$$x_{scaled} = \frac{x - \mu_x}{\sigma_x} \tag{4.7}$$

where μ_x and σ_x are, respectively, the mean and the standard deviation computed on the training set.

Post-processing techniques

Several **post-processing** techniques have been tested as well to improve results. Once the model has produced the output for the specific batch of EEG segments, the output is post-processed to refine the prediction. Specifically, a so-called smoothing technique is applied to reduce the number of false

alarms. It may happen to have sudden spikes in isolated EEG segments, that may be not-related to seizure events but unjustified brain activity. To avoid a positive prediction of those isolated segments, smoothing applies a sliding window over the output that computes the prediction based on the value of nearby predictions. Practically, the output of a specific EEG segment is compared with the outputs that belong to the "smoothing" window and a criterion is applied to define the final prediction. Two different criteria were tried:

- majority voting criteria: the most occurring predictions in the window are considered instead of the original output;
- minimum criteria: the minimum value appearing in the window is considered, thus being more conservative.

These two criteria are both tested considering different widths of the windows, from 3 to 7, windows centred in the considered EEG segments whose output is being smoothed. To better understand, an overview of these two methods is shown in Figure 4.14.

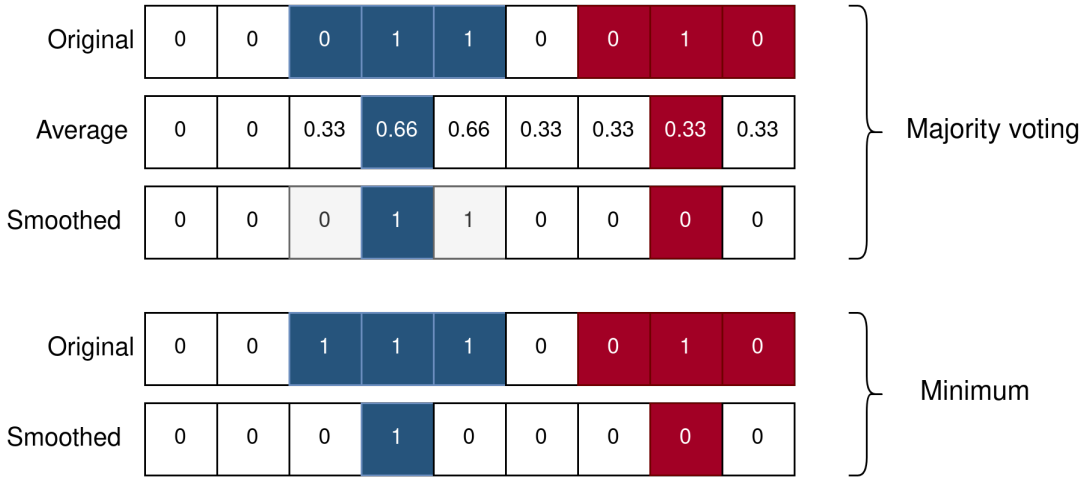

Figure 4.14. Smoothing criteria applied to the output of the model. Notice how since only windows of odd width are considered, the majority voting is implemented by computing the average of the values in the window and then approximating to the nearest integer.

To further reduce the number of false positives, together with the smoothing technique, grouping together false positives is considered as well. Specifically, the idea is that consecutive positively predicted EEG segments, when

there should be no seizure event, should not be considered distinct false positives. The rationale is that there is probably something in those consecutive EEG segments that is triggering the positive prediction by the model. The idea is then to consider consecutive false positives as one false positive event. Once again, to better understand, an overview of this approach is shown in Figure 4.15. These first two post-processing techniques, smoothing and grouping consecutive false positives, have the objective of increasing the specificity of the model, or the ability of the model to detect non-seizure events and avoid false positives, compromising a bit the sensitivity of the model.

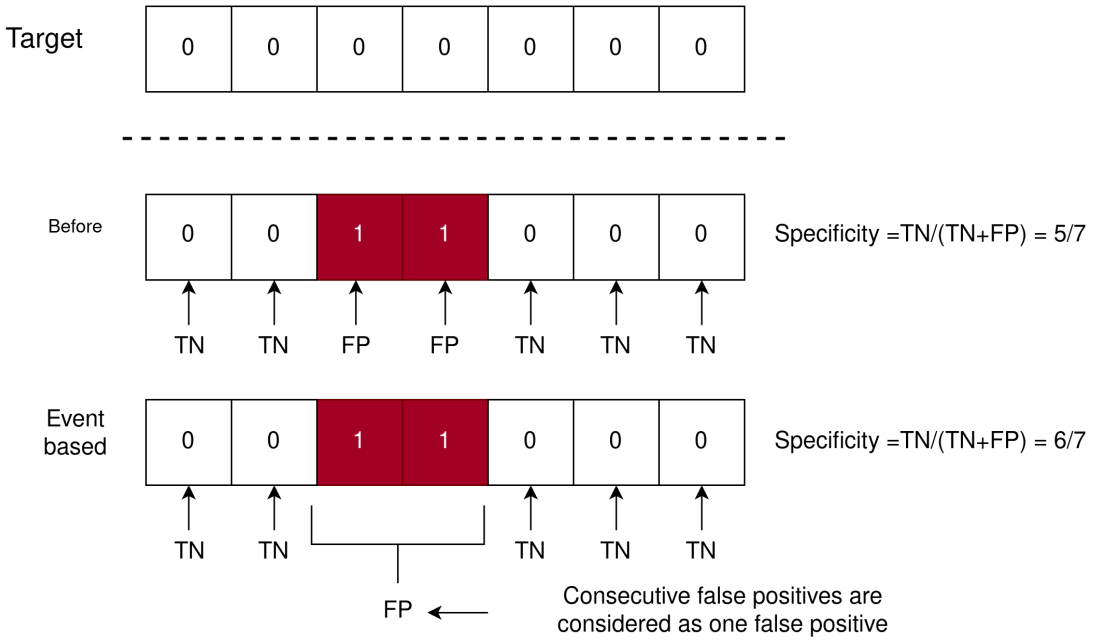

Figure 4.15. Post-processing technique that group together consecutive false positives. The first array shows the target or the ground truth of the EEG segments considered. The second array shows the output of the model and how the FP are counted before the post-processing is applied. The third array is how the FP are counted after the post-processing is applied. On the right it is possible to see how the specificity is computed

Additionally, a specific post-processing technique is applied to improve sensitivity and compensate for the effect of the previous two techniques. The idea is to evaluate the model based on how many seizure events are detected, rather than single EEG segments. A seizure event usually consists of more EEG segments: with this approach, a seizure event is correctly classified if

at least one of its EEG segments is correctly classified as a seizure segment. Using this technique, the sensitivity is then computed on the number of seizures detected over all the seizures of the specific segments. Later on, the sensitivity derived by this approach is called "event-based" sensitivity, while the "regular" sensitivity, computed on single segments, is called "segment-based" sensitivity. A simple scheme of this approach is shown in Figure 4.16.

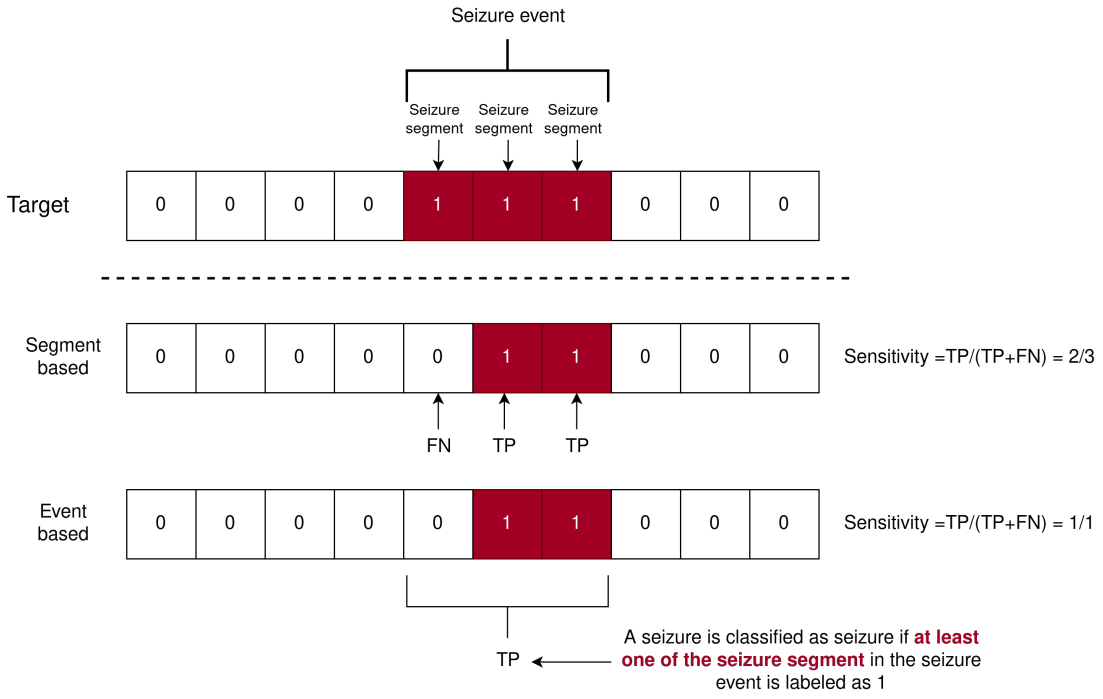

Figure 4.16. Segment-based approach vs Event-based approach to computing sensitivity.

4.3.4 Reducing available training data

Finally, the behaviour of model performances is evaluated when the data available for fine-tuning is evaluated. The idea is to understand how much more data the model needs to adapt the pre-trained weights learned on the unlabelled data to the seizure detection task. This means that data in the pre-training dataset remains the same, but data in the fine-tuning dataset is gradually reduced. The objective is to understand if the model is able to detect seizure when half of the labelled data of the fine-tuning dataset

initially considered is used.

4.3.5 Overall training procedure

In this Section, all the variables and all the alternatives tested are shown. This small section can be very useful to have a panoramic view of all the experiments that have been carried out during the fine-tuning optimization proposed by the thesis. Table 4.1 shows all the different dimensions and variables that have been tested during the optimization of the fine-tuning process.

Dimension	**Variable**	**Value**
Scalability	Convolutional Blocks	[6, 4, 2]
	Transformers Heads/Layers	[16, 8, 4, 2]
Pre-training	Pre-train only on TUEG	[BENDR]
	Second pre-training on CHB-MIT	[BENDR, MAEEG]
Pre-processing	Filtering	[None, Butterworth]
	Oversampling	[SMOTE, WRS]
	Normalization	[MinMax, MeanStd]
Post-processing	Smoothing	Majority voting criteria: [3, 5, 7] Minimum criteria: [3, 5, 7]
	Grouping False Positive	None
	Seizure detection	[segment-based, event-based]
Data used	Available training data	[100%, 50%]

Table 4.1. Dimensions explored during the fine-tuning optimization process.

Chapter 5

Results

The following chapter presents the results obtained and it is organized in sections. Since for the unsupervised approaches, the only results obtained are the ones that replicate the results obtained by the authors in the original works, respectively [35] and [38], these are not displayed in this chapter. The replicated results are shown for completeness in the Appendix, in A. For what concern this chapter, 5.1 describes the datasets that have been used for both pre-training and fine-tuning. 5.2 presents the metrics considered for the evaluation of the results. Once the metrics have been described, **??** describes how pre-processing and post-processing techniques are applied.

5.1 Datasets

Two different datasets are taken into consideration. For the **pre-training** task, Temple University Hospital EEG Corpus (TUEG) [54] is considered. Among the different versions of this dataset, versions 1.1 and 1.2 are taken into account, for a total of 1.5 TB of European data format (EDF). It consists of clinical recordings using a mostly conventional recording configuration (monopolar electrodes in a 10–20 configuration) of over 10,000 people, some with recording sessions separated by as much as 8 months apart. The subjects were 51% female, and ages range from under 1 year old to over 90. Some more interesting metrics of TUEG are reported in Figure 5.1. The main features of this dataset are its size and its unannotated nature: these two traits make it perfect to be exploited to learn the underlying structure of data. It is also important to understand that the TUEG dataset had little if any inspection for data quality. However, pre-training on this dataset is not carried out during the scope of the thesis. Authors that proposed BENDR

[17] have made the pre-trained weights learned on TUEG in an unsupervised fashion available to the public. Thus, these weights are the starting point of the adaptation of BENDR on the seizure detection task and a different dataset with respect to the ones on which it was tested in [17]. However, the pre-training task is extensively described in 4.2.1 and shown in Figure 4.5.

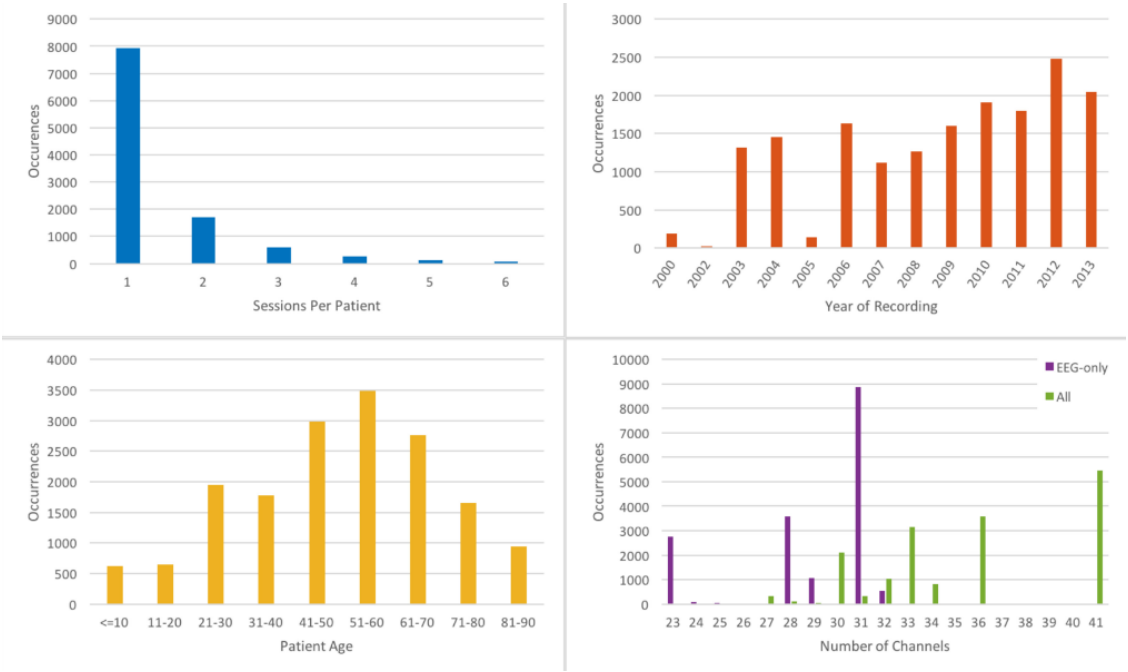

Figure 5.1. Metrics describing the TUH-EEG corpus. [Top left] histogram showing number of recording sessions (each patient can have from 1 - most common - up to 6 recording sessions); [top right] histogram showing number of sessions recorded per calendar year; [bottom left] histogram of patient ages; [bottom right] histogram showing number of EEG-only channels (purple); and total channels (green) [54].

For the **fine-tuning** task, CHB-MIT Scalp EEG Database [55] is considered. CHB-MIT sEEG database, collected at the Children's Hospital Boston, consists of EEG recordings of 24 pediatric subjects with intractable seizures. Subjects were monitored for up to several days following withdrawal of anti-seizure medication in order to characterize their seizures and assess their candidacy for surgical intervention. The dataset consists of 664 EDF files, and 185 of those contain one or more seizure events. During the experiments carried out for the thesis, only EDF files that contain one or more seizures are considered. All signals were sampled at 256 samples per second with 16-bit resolution. Every multi-channel EEG signal has been collected using

the International 10-20 system (described in Figure 2.1, monitoring 23 EEG channels over time. Some more information about the CHB-MIT is reported in Table 5.1. Patient chb24 was not considered because it was added to the collection later, in December 2010, and it is not included in the files with information on each patient (indeed, no gender nor age is specified).

Subject	**Gender**	**Age**	**seizure**	**(hh: mm: ss)**
chb01	F	11	7	40:33:08
chb02	M	11	3	35:15:59
chb03	F	14	7	38:00:06
chb04	M	22	4	156:03:54
chb05	F	7	5	39:00:10
chb06	F	1.5	10	66:44:06
chb07	F	14.5	3	67:03:08
chb08	M	3.5	5	20:00:23
chb09	F	10	4	67:52:18
chb10	M	3	7	50:01:24
chb11	F	12	3	34:47:37
chb12	F	2	27	20:41:40
chb13	F	3	12	33:00:00
chb14	F	9	8	26:00:00
chb15	M	16	20	40:00:36
chb 16	F	7	10	19:00:00
chb17	F	12	3	21:00:24
chb18	F	18	6	35:38:05
chb19	F	19	3	29:55:46
chb20	F	6	8	27:36:06
chb21	F	13	4	32:49:49
chb22	F	9	3	31:00:11
chb23	F	6	7	26:33:30
chb24	-	-	16	21:17:47
Total			185	979:56:07

Table 5.1. CHB-MIT Scalp EEG Database details per patients

5.2 Metrics

To evaluate and compare the results of different methods, several metrics are taken into account. Specifically, the following section presents the Confusion matrix, F1 score, Receiver Operator Characteristic (ROC), specificity, sensitivity and False Positives per hour (FP/h). For the unsupervised approaches, F1 score and Area Under (ROC) Curve (AUC) are considered, while for self-supervised approaches specificity, sensitivity and FP/h are considered.

Confusion Matrix

The Confusion Matrix is a technique for summarizing the performance of a classification algorithm. Classification accuracy alone can be misleading if there is an unequal number of observations in each class of the dataset. Calculating a confusion matrix can give a better idea of what the classification model is getting right and what types of errors it is making.

The number of correct and incorrect predictions is summarized with count values and broken down by each class. In a two-class problem such as seizure detection, it is important to discriminate between observations with a specific outcome, from normal observations, such as non-seizure and seizure. In this way, it is possible to assign the event "seizure" as “positive” and the non-event "non-seizure" as “negative“. With this coding, the event (seizure) column of predictions is categorized as “true” and the nom-event (non-seizure) as “false“. More specifically:

- True Positive (TP), for correctly predicted event or seizure;
- False Positive (FP), for incorrectly predicted event or seizure;
- True Negative (TN), for correctly predicted non-event or non-seizure;
- False Negative (FN), for incorrectly predicted non-event or non-seizure.

Table 5.2 shows a simple Confusion Matrix scheme.

		Predicted	
		P	N
Actual	P	True Positive (TP)	False Negative (FN)
	N	False Positive (FP)	True Negative (TN)

Table 5.2. Confusion Matrix

F1 Score

F1 Score is an appropriate metric when dealing with an unbalanced dataset, such as CHB-MIT. F1 Score is the Harmonic Mean between precision and recall. To define this metric, two additional metrics are needed: Precision

and Recall. Precision is a measure of the correctly identified as positive in all positive predicted:

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

Recall is a measure of the correctly identified as positive in all the really positive:

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells how precise the classifier is (i.e. how many instances it classifies correctly), as well as how robust it is (i.e. it does not miss a significant number of instances). High precision but lower recall is extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better the performance of our model. Mathematically, it can be expressed as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{5.3}$$

F1 Score tries to find the balance between precision and recall.

Receiver Operator Characteristic

A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds and it is one of the most widely used metrics for the evaluation of binary classification problems. The ROC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. True Positive and False Positive Rates are defined as:

- **True Positive Rate**, that corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points:

$$TPR = \frac{TP}{FN + TP} \tag{5.4}$$

- **False Positive Rate**, that corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points:

$$FPR = \frac{FP}{TN + FP} \tag{5.5}$$

These terms both have values in the range $[0, 1]$. FPR and TPR both are computed at varying threshold values such as $(0.00, 0.02, 0.04, \ldots., 1.00)$ and a graph is drawn. AUC is the area under the curve of plot FPR vs TPR at different points in $[0, 1]$. The ROC curve is a good way to compare classifiers when the accuracy parameters are not enough. Generally, the bigger the AUC, the better the performance.

Specificity, Sensitivity, FP/h

The most important metrics, which are the ones considered for the self-supervised task, are specificity, sensitivity and false positives per hour. Specificity can be defined as:

$$Specificity = \frac{TN}{TN + FP} \tag{5.6}$$

Specificity highlights the ability of the model to recognize the non-seizure event and it is influenced by the number of false positives. The lower the false positives, the higher the specificity, and vice versa. Sensitivity can be defined as:

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{5.7}$$

Sensitivity highlights the ability of the model to recognize seizure events. The definition of sensitivity is the same as the TPR seen in Equation 5.4. Then, false positives per hour can be defined as:

$$FP/h = \frac{3600}{length} \times (1 - Specificity) \tag{5.8}$$

and maps the number of false alarms in an hour (i.e. 3600 s). This is one of the most important metrics in a seizure detection problem. It is crucial to have a consistent model that does not raise too many alarms in such a sensible scenario.

5.3 Training details

The original **loss function** proposed by [17] and shown in Equation 4.4 is modified to prioritize specificity and sensitivity. Specifically, two terms are added to take into account their contribution to the loss and to force the model to minimize them. The updated loss is defined as:

$$l = l_{ce} + \alpha(1 - specificity) + \beta(1 - sensitivity) \tag{5.9}$$

where l_{ce} is the cross-entropy loss described in Equation 4.4, $\alpha \in [0,1]$ is the weight of the specificity contribution and $\beta \in [0,1]$ is the weight of the sensitivity contribution. In this way, the backward propagation step learns to update the weights to minimize the contribution of these two metrics. It is immediate to see that the higher the specificity and the sensitivity, the lower the loss. During training, after some hyperparameter search, $\alpha = 0.3$ and $\beta = 0.5$. The minority class (seizure) in the training set is **oversampled** considering the two different techniques discussed in Section 4.3.3. Then all, the minimum, the maximum, the mean and the standard deviation of the training set are computed and used to normalize the three sets, training, validation and test with the chosen normalization step. Adam optimizer is considered with learning rate $lr = 1e-4$ and L2-regularization weight decay parameter set to 0.01. A learning rate scheduler is used, which reduces the learning rate on the plateau, with parameters $mode = min$, $factor = 0.1$ and $patience = 5$. Then, a custom early stopping mechanism is applied as well that monitors the validation loss with $patience = 15$ and $\epsilon = 0$. Generally, with few exceptions, a batch size of 256 is considered, and the upper bound of the number of epochs to train the model is set to 120. However, with Early stop, for most of the configurations, the model is fine-tuned for no more than 35 epochs. Finally, and most importantly, during training, every Dropout added between the 1d-convolution and the group norm layer (as explained in 4.2.1) drops entire channels of the network with probability $p = 0.5$. This will reveal to be a crucial contribution to the reduction of overfitting.

Table 5.3 shows the hyperparameters' search space for all the parameters discussed above.

5.4 Results

The first results obtained and, thus, the first results shown are the ones obtained in the first fine-tuning test carried out. These results represent the starting benchmark, from which the extensive research for the optimal fine-tuning strategy started. In this case, the model is the original BENDR described in 4.2.1, with the training strategy explained in Figure 4.9 and details explained in 5.3. These first results are shown also by patients just to have an idea of the variability of the performance of the model within patients. Generally, from now on, only averaged results will be shown. Table 5.4 shows the first results obtained with BENDR, exploiting the pre-trained weights made available by [17], and fine-tuning for each patient. As expected,

Parameter	Values
segment_length	[1, **4**, 8, 16]
batch_size	[128, **256**]
learning_rate	[1e-3, **1e-4**, 1e-5, 5e-5]
second pre-training epochs	[**10**, 20, 50]
fine-tuning epochs	[50, 100, **120**]
α	[**0.3**, 0.5, 0.7]
β	[0.3, **0.5**, 0.7]
patient EarlyStopper	[10, **15**]
delta EarlyStopper	[**0**, 1e-4]
optimizer scheduler	[**None**, OneCycleLR]

Table 5.3. Hyperparameters search space. Optimal values are in **bold**.

using such a huge model as BENDR without initializing the weights of the model with the pre-trained weights available leads to huge overfitting. Indeed, it is immediate to notice that, training only on CHB-MIT dataset, without using the weights learned on TUEG, leads to a model that is unresponsive to seizures, predicting only non-seizure events. The model that does not exploit pre-trained weights (the right part of Table 5.4) tends to have a very high specificity, and thus very low false positive per hour, but a very low sensitivity: it never predicts a seizure, and since the test data is highly unbalanced, specificity is very high.

5.4.1 Scalability results

Starting from these initial results, the different dimensions discussed in 4.3 are explored. First of all, it is studied how the model size influences the performance of the model. As shown in Figure 4.10, an extensive search for the optimal size of the model is carried out. Table 5.5 shows the results obtained for different configurations. In addition to the original architecture, 4 different architectures have been tested, modifying the number of convolutional blocks and the number of heads/layers of the transformer. Ultimately, they differ in the number of trainable parameters. It is also important to underline that this architecture change has been made only when fine-tuning, while the pre-training architecture remains the same (i.e. the original one). This means that, when fine-tuning, only the layers already present during

	Exploting pre-trained weights			Without pre-trained weights		
patient	**specificity**	**sensitivity**	**fp/h**	**specificity**	**sensitivity**	**fp/h**
chb01	99.904	90.539	0.867	99.984	0.000	0.144
chb02	86.142	8.333	124.724	100.000	0.000	0.000
chb03	96.935	84.389	27.584	100.000	0.000	0.000
chb04	89.921	54.484	90.715	100.000	0.000	0.000
chb05	99.908	86.427	0.827	100.000	0.000	0.000
chb06	98.759	2.597	11.173	100.000	0.000	0.000
chb07	99.945	36.508	0.498	100.000	0.000	0.000
chb08	98.569	86.145	12.880	100.000	0.000	0.000
chb09	97.505	68.566	22.459	100.000	0.000	0.000
chb10	98.728	95.698	11.446	100.000	0.000	0.000
chb11	100.000	44.078	0.000	99.888	0.000	1.008
chb12	51.462	55.558	436.844	100.000	0.000	0.000
chb13	98.248	77.601	15.771	100.000	0.000	0.000
chb14	99.713	44.675	2.580	100.000	0.000	0.000
chb15	7.143	92.857	835.714	100.000	0.000	0.000
chb16	98.904	8.333	9.861	100.000	0.000	0.000
chb17	99.432	72.989	5.111	99.848	1.515	1.367
chb18	95.337	75.902	41.963	100.000	0.000	0.000
chb19	96.265	45.614	33.611	98.303	22.719	15.271
chb20	98.537	41.868	13.164	100.000	0.000	0.000
chb21	98.091	42.321	17.179	100.000	0.000	0.000
chb22	99.925	53.439	0.679	100.000	0.000	0.000
chb23	99.189	84.250	7.299	100.000	0.000	0.000
Average	91.677	58.834	74.911	99.914	1.054	0.773

Table 5.4. First results obtained fine-tuning BENDR. On the left side, there are results obtained exploiting the available pre-trained weights, on the right side results without pre-trained weights.

pre-training, or their equivalent, are initialized with the pre-trained weights, whilst the others are randomly initialized. This is one of the reasons why the number of convolutional blocks was not increased: during the thesis, it was found that adding blocks in the first part of the network (i.e. the one that extracts the representations of raw EEG data) was counterproductive to the generalization of the pre-trained weights learned on TUEG.

Interestingly, a smaller model behaves better than the original one, while a bigger one tends to overfit. This is immediate to notice when looking at the relationship between specificity and sensitivity: similarly to what is shown

in Table 5.4, the model, when its size is increased, tends to predict only non-seizure events, thus lowering its sensitivity. This could also be a consequence of the fact that more than half of the model's weights are randomly initialized (as the one in the right part of Table 5.4), thus compromising its sensitivity once again. Another interesting thing to notice is that a model that is a quarter of the size of the original one (the 25% model) behaves similarly: this could mean that the most important information needed to adapt TUEG to the CHB-MIT dataset is mainly found in the first layers of the transformer. This will be further discussed in the Section (5.5). A representation of some of the results obtained is shown also in Figure 5.2.

	Model configuration			**Results**		
	Parameters (M)	**Conv. blocks**	**Transformer h-l**	**Specificity**	**Sensitivity**	**FP/h**
Original	157	6	8	91.677	58.834	74.911
200%	308	6	16	93.119	21.324	61.933
55%	82	6	4	**95.740**	**60.763**	**38.338**
50%	79	4	4	87.853	68.516	109.322
25%	41	2	2	91.954	59.581	72.417

Table 5.5. Results based on different model configurations (different number of convolutional blocks used in the first stage, different number of heads and layers used in the transformer

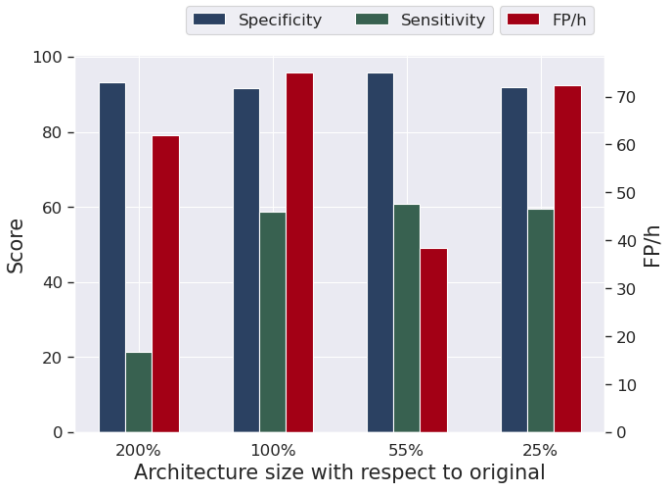

Figure 5.2. 4 different model sizes compared with each other. If looking at the FP/h metric it is possible to spot a sort of optimal point with the 55% architecture. If the model size is either increased or decreased, the FP/h worsens.

5.4.2 Pre-training architecture results

The second dimension that has been explored is the further pre-training of the model also on the other patients of the CHB-MIT dataset, except on the one on which fine-tuning is been carried out. This further pre-training is referred to as *second pre-training*. To do that, we use the two different architectures that have been presented in Chapter 4: BENDR, seen in 4.2.1, and MAEEG 4.2.2. The training protocol to obtain these results is the one already explained in Figure 4.11. The idea is to use the pre-trained weights learned on TUEG to initialize both models, BENDR and MAEEG, and then further pre-train them on the other CHB-MIT patients before fine-tuning them on the specific patients with Leave-One-Out Cross-Validation strategy. Results are shown in Table 5.6. With respect to the original, it is immediate to notice that with the BENDR architecture, a second pre-training slightly improves both the specificity and the sensitivity, thus lowering the FP/h. The same cannot be said for the MAEEG architecture, for which all three metrics worsen with a second pre-training. This could be in part justified by the fact that both models are initialized with pre-trained weights that have been obtained with the BENDR architecture, and for this reason, MAEEG benefits less from this different initialization. As also stated by the authors ([51]), MAEEG has been proposed as an alternative architecture for the pre-training on the huge unlabelled dataset TUEG, and not for this second pre-training task. For this reason, it would be more interesting to compare these two methods on the pre-training task. However, this test has not been carried out during the thesis because of the huge computing resource and the time required to complete the pre-training on TUEG.

	Specificity	**Sensitivity**	**FP/h**
Original	91.677	58.834	74.911
Pre-trained BENDR	92.711	60.326	65.599
Pre-trained MAEEG	86.860	61.754	118.254

Table 5.6. Comparison of BENDR and MAEEG when pre-training on all the patients of CHB-MIT except the one on which the fine-tuning is run. The first row shows the performance of BENDR without pre-training on other CHB-MIT patients.

5.4.3 Processing techniques results

In this section, the results of the pre-processing and post-processing techniques are reported. Since showing the results of the comparison one by one of all the different techniques considered would be confusing, some of the pre-processing and post-processing strategies are combined and two different combinations are shown in this section. With these two combinations, the idea is to show how some of the techniques influence the results. First of all, it is necessary to define the two combinations. Combination 1, also called "higher-sensitivity" combinations, as the name suggests, is the combination that prioritizes sensitivity over specificity, thus having a higher FP/h. Combination 1 is characterized by the following pre-processing steps:

- no filter during data loading;
- min-max normalization;

On the other hand, Combination 2 is characterized by the following pre-processing steps:

- Butterworth bandpass filter during data loading;
- mean-std standardization.

In these two configurations, the effect of the three post-processing techniques is studied. Results are shown in Table 5.7. More extensive research on the appropriate smoothing technique is also considered. Given the same model and the same output, the two methods, Majority voting and Minimum value, with different window lengths, are compared in Figure 5.3. As expected, a wider window hugely lowers the sensitivity of the model, improving the specificity. This is easy to understand: for a wider window, it is required that several EEG segments of the same window are detected as seizure in order to identify the considered segment as a seizure one. This behaviour is even more evident when the minimum criterion is chosen over the majority. The minimum criterion is probably too conservative: indeed, sensitivity is already compromised with a window length equal to 3. On the other hand, the majority voting criterion is less conservative and, in this case, sensitivity decreases slowly with the increase of the window width, compared to the minimum case. In general, as results show, smoothing the output avoids random peaks of EEG signal that could trigger a positive prediction by the model, resulting in a false positive. And by consequence, specificity benefits from the reduction of false positives, but sensitivity does not. The optimal

	Post-processing		Results		
	Smoothing	**Event+Grouped**	**Specificity**	**Sensitivity**	**FP/h**
Configuration 1	N	N	95.208	59.477	43.110
	Y	N	96.465	88.889	31.815
	Y	Y	98.428	81.159	14.149
Configuration 2	N	N	97.366	41.815	23.710
	Y	N	98.315	40.498	15.169
	Y	Y	99.765	65.217	2.111

Table 5.7. Comparison between different pre-processing and post-processing techniques. The configurations differ from each other because of the pre-processing techniques applied. In the table, it is possible to see how the post-processing techniques change the performances. In this case, "Smoothing" refers to the Majority voting strategy based on sliding windows on the outputs with length 3, while "Event+Grouped" refers to the event-based sensitivity combined with the grouping consecutive false positive techniques, discussed in 4.3.3.

balance between specificity and sensitivity seems to be found with majority voting with a window of length 3.

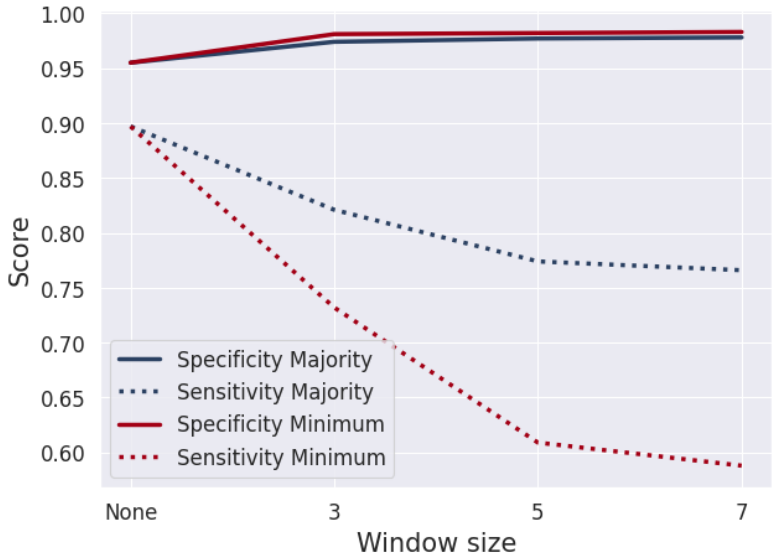

Figure 5.3. Comparison between smoothing techniques: majority voting vs minimum value

To give the complete overview of the performance achieved, the best results of the two different configurations shown in Table 5.7 are shown also patient

by patient. In this way, it is possible to compare it with the initial results shown in Table 5.4. Patient-by-patient results of configurations 1 and 2 are shown in Table 5.8.

	Configuration 1			Configuration 2		
patient	**specificity**	**sensitivity**	**fp/h**	**specificity**	**sensitivity**	**fp/h**
chb01	99.950	100.000	0.451	100.000	100.000	0.000
chb02	99.079	0.000	8.285	99.898	0.000	0.917
chb03	99.848	100.000	1.364	100.000	100.000	0.000
chb04	96.688	100.000	29.805	99.901	50.000	0.893
chb05	100.000	100.000	0.000	100.000	100.000	0.000
chb06	99.885	0.000	1.034	99.988	0.000	0.105
chb07	99.880	100.000	1.082	99.989	100.000	0.100
chb08	99.207	100.000	7.140	99.789	100.000	1.901
chb09	99.754	100.000	2.215	99.756	100.000	2.196
chb10	99.773	100.000	2.046	99.979	100.000	0.192
chb11	100.000	100.000	0.000	99.950	100.000	0.448
chb12	96.197	33.333	34.231	96.312	33.333	33.188
chb13	99.285	100.000	6.433	99.791	100.000	1.878
chb14	100.000	66.667	0.000	100.000	16.667	0.000
chb15	78.898	100.000	189.919	99.782	100.000	1.965
chb16	99.089	0.000	8.199	99.950	0.000	0.447
chb17	99.518	100.000	4.336	99.898	100.000	0.919
chb18	99.936	100.000	0.574	100.000	100.000	0.000
chb19	100.000	66.667	0.000	100.000	66.667	0.000
chb20	99.130	100.000	7.827	100.000	0.000	0.000
chb21	97.983	100.000	18.157	100.000	0.000	0.000
chb22	99.910	100.000	0.814	99.898	33.333	0.915
chb23	99.832	100.000	1.514	99.724	100.000	2.487
Average	98.428	81.159	14.149	99.765	65.217	2.111

Table 5.8. Result of configuration 1 and 2 patient by patient

For both configurations, it is possible to notice there are some "critical" patients, i.e. patients with an oddly high fp/h ratio (or low specificity) with respect to others. In configuration 1, for example, patients chb04, chb12, and chb15 are considered critical patients, while for configuration 2 only chb12 can be considered critical. The removal of these patients is discussed in detail in section 5.5. These patients compromise the average results since FP/h is hugely out of scale with respect to the others. Table 5.9 shows the best results of the two configurations removing the critical patients discussed

above.

	critical	specificity	sensitivity	fp/h
Config 1	4, 12, 15	99.603	81.667	3.574
Config 2	12	99.922	66.667	0.698

Table 5.9. Best results without considering critical patients

Lastly, a comparison with the state-of-the-art performance is shown. Specifically, an XGBoost-based model trained in a supervised way is considered to be the state of the art (Ingolfsson et al., Unpublished, 2023). The best performances of the fine-tuned BENDR model are taken into account, both with and without critical patients. Results are shown in Table 5.10. It is immediate to notice how, with the appropriate precautions, the best performances of BENDR overcome the state-of-the-art ones. Of course, this is achieved considering several post-processing techniques that were not deployed in the state-of-the-art, but, still, the results are promising considering the starting point (Table 5.4).

	critical	specificity	sensitivity	fp/h
Config 2	None	99.765	65.217	2.111
Config 2	12	**99.922**	**66.667**	**0.698**
State of the Art	None	99.910	64.090	0.850

Table 5.10. Results comparison with the state of the art

5.4.4 Reducing training data available

The last test tried to study the fine-tuning process considering reducing the amount of training data available. The idea is to see how the model behaves when it is trained on only half the data originally available for fine-tuning for each patient. Practically, looking at Figure 4.9, suppose that, for example, for patient chb01, not 6 edf files are available, but only 3, how would the model behave with respect to the "all data" case? To carry out a more robust test, for each patient, the files for each patient that should not be considered have been chosen randomly and the test is repeated two different times. A more robust approach would maybe run the test more times. For patients with an odd number of edf files, an approximation by excess has been considered: for example for patient chb02, which originally has 3 edf files that contain a

seizure, only 2 were considered. In the end, on average, slightly more than 50% of the original data are considered, but the study remains meaningful. The best architecture found in Table 5.5 is considered as a comparison, thus BENDR with the same number of convolutional blocks in the first stage, but half of the heads/layers in the transformer encoder. This has been done, both to speed up training and to combine the contribution of a reduction in the size of the model and the reduction in data. Results are shown in Table 5.11. It can be noticed that, when reducing the data available during fine-tuning, specificity is slightly increased (and thus FP/h decreases with it), but sensitivity decreases. This could be similar to what happens when the model is randomly initialized rather than with the pre-trained weights (Table 5.4). Reducing the amount of fine-tuning data prevents the model to adapt from the pre-trained weights learned on TUEG to a different downstream task and a different dataset. Even if the distribution between seizure and non-seizure is the same as the 'all data' case, it may be that a further hyperparameters search is needed (Table 5.3). In conclusion, it can be said that the increase in specificity, and the consequent decrease in FP/h do not justify the decrease in sensitivity. The model, since it has less data to be trained on, does not learn to properly recognize the seizures, but predicts more easily a non-seizure. This is also very similar to what happened with the 200% model (Table 5.5): in that case, the amount of fine-tuning data was the same, what changed was the model size, where there was a model twice the size of the original that did not have enough time to learn to predict seizures.

	Specificity	**Sensitivity**	**FP/h**
All data	95.740	60.763	38.338
Half data	96.651	43.319	30.144

Table 5.11. Results of BENDR when fine-tuned on less data.

5.5 Discussion

The results obtained were in part in line with what was expected. The first results (Table 5.4) obtained were not promising for most patients and needed a lot of work to become competitive and comparable with the state of the art. However, from the beginning, it is clear how a huge model such as BENDR benefits from the pre-trained weights on a large unlabeled dataset ([17]). Indeed, it is immediate to notice that, when the model's weights are

randomly initialized the model heavily overfits on CHB-MIT: this is evident because the model learns how to predict only non-seizure segments, and thus from here the higher specificity (and very low FP/h), without even trying to predict seizure. However, on the training set, which is balanced between seizure and non-seizure, validation loss continues to decrease. The ineffectiveness observed with the randomly initialized full architecture could be the first evidence of the validity of the use of this powerful emerging architecture, which hugely benefits from pre-training, especially when deployed on a limited-data scenario. Of course, this may be due to the large number of parameters that these types of models require, making training difficult without sufficient data and hardware resources. Starting from these first results, the objective is to explore the search space of possible alternatives, considering different architectures, pre-training schedules, processing techniques and less data.

The impact of model size on the performance is evident in both Table 5.5 and Figure 5.2. The approach benefits from a smaller model size, reducing overfitting and increasing all the considered metrics. This may be justified by the fact that the complex architecture used during the unsupervised pre-training does not allow the model to properly adapt to the seizure detection task, and thus, a small number of parameters to be trained can speed up convergence without compromising the performances. An interesting thing to notice is how the 50% BENDR architecture behaves: reducing the number of convolutional blocks used in the first phase proves to be less effective than just reducing the number of heads and layers that the model has. This can be explained considering the complexity of EEG data and considering fewer blocks would mean reducing the space on which the input EEG segments are mapped. However, when reducing both the number of convolutional blocks and the number of heads/layers hugely, as in the 25% BENDR, the model benefits from this simplification. This could be justified by the overall balance between the first and the second stage: indeed, for the 50% architecture, the first stage was less refined with respect to the transformer. Lastly, considering Figure 5.2, it is immediate to notice a sort of convex trend considering the 100%, 55% and 25% BENDR, with what may seem to be a global minimum in terms of FP/h in the 55% one.

Then, the potential of training on other CHB-MIT patients except for the one on which the model is being fine-tuned was explored. Here, the pre-trained BENDR architecture outperforms MAEEG. This was not an expected result. MAEEG was indeed proposed by [51] as an alternative pre-training

technique to BENDR, that improve its ability to extract meaningful information from unlabeled data. However, the bad behaviour of MAEEG may be justified by the fact that the pre-training of BENDR and MAEEG carried out in this thesis is not considered on the huge unlabelled dataset TUEG, but rather on the relatively smaller CHB-MIT. It is plausible to think that MAEEG outperformed BENDR when both of these architectures were pre-trained on TUEG, and then fine-tuned on the same tasks, using the same classification architecture, since it has been proven in [51] that MAEEG has better understanding capabilities on huge unlabeled data. It would be interesting to see, in future works, how MAEEG behaves when is pre-trained on TUEG and then adapted for the seizure detection task.

Results of the different processing techniques shown the greatest improvement in performance. What has been found is that (Table 5.7) using appropriate pre-processing techniques is possible to obtain a model that is more sensitivity sensible (as configuration 1), and one that is more specificity sensible (as configuration 2). Once these two models are trained, they can be hugely improved with post-processing techniques. In particular, it was found that:

- Smoothing the output (especially with majority voting on a window of length 3) avoids random and sudden peaks of EEG signal that could trigger a positive prediction by the model, resulting in a false positive. So, smoothing helps reduce false positives and thus increases specificity.
- Grouping together false positives again slightly improved specificity. This strategy is justified by the fact that if there are consecutive false positives it is likely due to some consecutive abnormal EEG segments that trigger the model's positive prediction. However, it is fair to consider these as a single false positive event, rather than, for example, 3 different false positives, since the model will alert the patient or the clinician just once when considering the seizure event.
- Considering seizure events rather than seizure segments improved the sensitivity of the model and slightly decreased its specificity. Future works may also consider a threshold to recognize a seizure: for example, instead of one, a seizure is recognized if the majority of its EEG segments are classified as a seizure. This post-processing technique is probably the most powerful tool implemented to improve the performance of the model.

For what concerns the so-called "critical" patients, it was not immediately

noticed if there was a common trait that characterizes all of them. The only noticeable aspect that these three patients (chb04, chb12, chb15) share is either the amount of data available or the number of seizure events they experienced. Indeed, from Table 5.1:

- patient chb04 has the longest amount of cumulative recordings (with more than 156 hours), and also one of the lowest numbers of seizures;
- Patients chb12 and chb15 are the top-2 patients in terms of seizure events within their recordings.

Generally speaking, it can be stated that all of these 3 patients share a non-regular distribution of seizure events in their recordings: either having a lot in a relatively short time or very few in a long time. An interesting metric to look at to grasp how data from these 3 patients, in particular chb04 and chb12, differs a lot from the others is the number of seizure events per hour. Table 5.12 shows this metric. It is immediate to notice how chb04 is the one with the lowest number of seizures per hour and, on the other hand, chb12 is the patient with the highest number of seizures per hour. Patient chb15 has a higher than-average number of seizures per hour, but it is less evident with respect to the other two patients. This could justify some of the bad performances, especially for chb12, which is the only critical patient in configuration 2. Indeed, chb12 has experienced 27 different seizures in just 20 hours of recording, thus it may explain the difficulties of the model in recognizing the seizure (with 33% sensitivity) and the high value of false positives (more than 33 false positives per hour). This also means that the model is trying somehow to respond and predict seizures but with scarce success. It should be also mentioned, however, that several works in the state of the art disregarded some of the patients of CHB-MIT, as in [27], [56], [57]. Configuration 2, without patient chb12, can overcome the state-of-the-art performances as seen in Table 5.10. However, it must be said that the state-of-the-art method applied only smoothing as a post-processing technique and did not either group together consecutive false positives or counted seizures detected instead of segments. It is likely that, if the same techniques were applied in the SoA, the performances of the SoA approach may be higher than the ones obtained with BENDR.

The last results show how the model behaves, given the same architecture and training setup, with almost half the data. It is interesting to notice how specificity improves when fewer data are available: this is explained by the fact that in an unbalanced dataset as CHB-MIT, reducing data has much

Paitient	**Seizure**	**Hours**	**Ratio**
chb01	7	40	0.18
chb02	3	35	0.09
chb03	7	38	0.18
chb04	4	156	**0.03**
chb05	5	39	0.13
chb06	10	66	0.15
chb07	3	67	0.04
chb08	5	20	0.25
chb09	4	67	0.06
chb10	7	50	0.14
chb11	3	34	0.09
chb12	27	20	**1.35**
chb13	12	33	0.36
chb14	8	26	0.31
chb15	20	40	**0.50**
chb 16	10	19	0.53
chb17	3	21	0.14
chb18	6	35	0.17
chb19	3	29	0.10
chb20	8	27	0.30
chb21	4	32	0.13
chb22	3	31	0.10
chb23	7	26	0.27
chb24	16	21	0.76

Ratio metrics	
Mean	0.26
Median	0.16
Min	0.03
Max	1.35

Table 5.12. Number of seizure events per hour for each CHB-MIT patient

more effect on the seizure data because, even if the distribution between seizure and non-seizure is observed, having fewer seizure data available prevent the model from learning the seizure characteristics of EEG signals. And thus, sensitivity decreases with respect to the original one.

Overall, it can be said that, with the appropriate measures, when all dimensions are optimized, the model is able to obtain satisfactory results. Consider for example results for each patient of Configuration 2 of Table 5.8, it can be seen that for 13 patients out of 23 100% sensitivity is reached and with an acceptable FP/h for all patients except one. Generally speaking,

average results are more than satisfying and comparable with the state-of-the-art, while even better when the critical patient is disregarded (as seen in Table 5.10).

Chapter 6

Conclusion and Future Works

Labelled EEG datasets are scarce because of the cost in terms of time and expertise required for the labelling process. On the other hand, a huge amount of unlabelled EEG data is currently available and it is not, largely, exploited. The objective of the thesis was to explore approaches to exploit this huge amount of unlabelled EEG data available for seizure detection. The first tested fully unsupervised approaches trained on non-seizure data did not provide promising results. The focus then shifted to self-supervised approaches. Inspired by masked language model-like training, BENDR was proposed and pre-trained on a huge unlabelled dataset to understand the underlying structure and characteristics of EEG. Starting from this pre-trained model, the thesis focused on its fine-tuning with a modified architecture, for a different task and on a different, and smaller, dataset. Several aspects of the fine-tuning process were taken into consideration: scalability, pre-processing and post-processing techniques, further pre-training architectures and reducing the data available. An extensive search was carried out on these different dimensions, to find the best configuration for the seizure detection task. The best model was able, with some precautions, to obtain results competitive with the state of the art, demonstrating and validating the potential of these types of learning paradigms. The key takeaways for the optimization of the fine-tuning process are:

- a model of smaller size may prevent overfitting on a smaller dataset;

- regularization techniques (especially heavy dropout, early stopping mechanism, learning rate scheduler and the modification of the loss to prioritize the metrics that are being considered) reduced again overfitting and improved the generalization on different patients;
- pre-processing and post-processing techniques had the biggest impact on performance improvement.

Future works may consider improving the pre-training approach by adjusting the neural network architecture and pre-training configuration such that it becomes more data-domain EEG appropriate. Indeed, BENDR applied a paradigm developed specifically for text and speech recognition, without any major update on the architecture, even though the nature of this data is very different from EEG. Additionally, future works may also consider a different pre-training dataset, which may be more related to the seizure detection task. BENDR was originally proposed to solve downstream tasks related to Brain-Computer Interfaces (BCI), while during the thesis it was tested on a seizure detection task. Furthermore, the ability of this model as a seizure predictor rather than a detector may be explored (as seen in the A).

This work showed the potential of transfer learning scheme applied to EEG in the seizure detection task, leveraging the immense amounts of unlabelled data available in this field. Additionally, the work validated, even more, the effectiveness of a huge large language-inspired model as BENDR. The use of a general model that can be pre-trained on EEG data and then fine-tuned and adapted to the specific EEG-related tasks may overcome the difficulties encountered when developing a custom model for a specific task which relies on the amount of available labelled data.

Appendix A

Appendix

A.1 Project description

Epilepsy detection on EEG data is a challenging task, both due to the limited amount of labelled data often available for the training of a classifier, and for the high accuracy standards required for a monitoring device to detect all seizure events without raising highly stressful false alarms. On top of that, in practical usage conditions with wearable devices, EEG is often affected by artefacts, which in turn are often mistaken for seizures due to their morphological similarity in both amplitude and frequency, making seizure detection systems susceptible to higher false alarm rates. Consequently, artefact detection and removal appear paramount for the successful deployment of epilepsy detectors on wearable devices. However, labelling artefacts together with seizures imposes an even higher burden on clinicians. Within this context, this project aims to explore unsupervised machine learning approaches to perform seizure detection with minimal or no labelling required.

A.2 Unsupervised replicated results

In this section, the replicated results of the unsupervised methods seen in 4.1 are shown. These results are not shown in the results section to avoid confusion between actual contributions and replicated results. If for the self-supervised approach, several contributions were made and presented in the results section, for the unsupervised approaches results were only replicated, since they were not that promising, these approaches were not further explored. Table A.1 shows the comparison between the performances obtained

by the authors that proposed the approaches and the results that were obtained during the thesis. Notice how, especially for the second method, the one that combined CNN with an anomaly detection method, it was not possible to replicate results, especially in terms of F1 score. Even if both the model and the training were set with the same configuration as the authors suggested, replicated results significantly differ from the original ones. For these reasons, since these results were obtained in the first part of the development of the thesis, and since it was not possible to replicate the results of promising methods, such as the one that combines CNN with anomaly detection, it was decided to shift the attention on self-supervised methods.

	Original		**Replicated**	
	F1 score	**AUC**	**F1 score**	**AUC**
VAE	0.589	0.681	0.561	0.689
CNN+AD	0.841	0.924	0.451	0.910

Table A.1. Comparison between original and replicated results of the unsupervised approaches. VAE is the variational autoencoder method, while CNN+AD is the one that combines CNN with the anomaly detection method.

A.3 A different architecture for the fine-tuning process

In this section, the results obtained with the Linear BENDR architecture (described in Figure 4.7) are reported. In Table A.2, a comparison with the standard fine-tuning architecture (described in Figure 4.6) is presented. This test has been carried out at the beginning of the second part of the thesis when BENDR was being adapted to the EEG scenario. For this reason, no considerations on model size, processing techniques and so on are being made. On the left, there are the results obtained with the BENDR fine-tuning architecture, on the right there are the results of the Linear fine-tuning architecture. Notice that the results of BENDR are the same presented on the left part of Table 5.4. Indeed, for both of these architectures, pre-trained weights on TUEG are being used.

	BENDR fine-tuning			Linear fine-tuning		
Patient	**Specificity**	**Sensitivity**	**FP/h**	**Specificity**	**Sensitivity**	**FP/h**
chb01	99.90	90.54	0.87	70.35	95.71	266.89
chb02	86.14	8.33	124.72	83.96	10.00	144.38
chb03	96.94	84.39	27.58	66.22	90.28	303.98
chb04	89.92	54.48	90.71	74.87	87.55	226.13
chb05	99.91	86.43	0.83	60.88	92.70	352.06
chb06	98.76	2.60	11.17	96.96	99.86	27.36
chb07	99.94	36.51	0.50	69.39	97.76	275.48
chb08	98.57	86.15	12.88	78.38	87.36	194.58
chb09	97.50	68.57	22.46	60.24	70.34	357.83
chb10	98.73	95.70	11.45	94.70	94.43	47.67
chb11	100.00	44.08	0.00	96.38	67.29	32.62
chb12	51.46	55.56	436.84	89.84	64.69	91.47
chb13	98.25	77.60	15.77	93.00	71.45	62.96
chb14	99.71	44.68	2.58	62.93	81.40	333.62
chb15	7.14	92.86	835.71	62.93	97.76	333.65
chb16	98.90	8.33	9.86	68.16	64.88	286.54
chb17	99.43	72.99	5.11	71.90	75.03	252.94
chb18	95.34	75.90	41.96	83.39	99.15	149.47
chb19	96.27	45.61	33.61	68.70	90.08	281.69
chb20	98.54	41.87	13.16	73.89	63.69	235.02
chb21	98.09	42.32	17.18	77.65	70.09	201.13
chb22	99.92	53.44	0.68	93.26	81.29	60.68
chb23	99.19	84.25	7.30	71.28	62.65	258.46
Average	91.68	58.83	74.91	76.92	78.93	207.68

Table A.2. Fine-tuning results obtained with the two different architectures: BENDR and Linear BENDR

A.4 False alarms distribution

This section studies the distribution of false alarms. More specifically, the objective is to see how much time before the seizure onset a false alarm is raised, in order to understand if there is the potential for this method to solve also a seizure prediction task (difference between detection and prediction in 2.1.3). The output of the best model is considered, then all the false positives are considered and all the actual seizure events are taken. Then for every false positive, the distance from the nearest seizure onset is computed. The distance in EEG segments is then translated into seconds and minutes. The

results of this analysis are shown in Figure A.1. It is very interesting to notice

Figure A.1. False positive distribution with respect to elapsed time between false positive and actual seizure onset, for all patients and all test files.

how much time before a seizure onset the model is raising a false alarm. The frequency is much higher in the first 25 to 30 minutes before the seizure and becomes even higher in the seconds just before the seizure onset. It may be very interesting to explore the use of this model as a seizure predictor since the graph shows promising initial results. The seizure predictor's objective is to alert the patient, as well as the clinicians, of an imminent seizure event with some advance notice, to allow the subject to take some measures. Even if the objective of this work was to explore seizure detection, from this short analysis, it is possible to catch a glimpse of the potential of this model as a seizure predictor.

Acronyms

AUC Area Under (ROC) Curve

AZC Approximate Zero-Crossing

BCI Brain-Computer Interfaces

BENDR BErt-inspired Neural Data Representations

CNN Convolutional Neural Network

DL Deep Learning

EDF European Data Format

EEG Electroencephalography

FC Fully Connected

FP/H False Positives per Hour

FPR False Positive Rate

iEEG intracranial EEG

k-NN k-Nearest Neighbors

LLM Large Language Model

LM Language Models

MAEEG Masked Autoencoder for EEG

ML Machine Learning

MLE Maximum Likelihood Estimation

MLM Masked Language Modeling

NLP Natural Language Processing

NSP Next Sentence Prediction

RNN Recurrent Neural Network

ROC Receiver Operator Characteristic

sEEG Scalp EEG

SMOTE Synthetic Minority Oversampling TEchnique

SVM Support Vector Machines

TPR True Positive Rate

VAE Variational Autoencoder

WRS Weighted Random Sampling

Bibliography

[1] Roland D Thijs et al. "Epilepsy in adults". In: *The Lancet* 393.10172 (2019), pp. 689–701. ISSN: 0140-6736. DOI: https://doi.org/10.1016/S0140-6736(18)32596-0. URL: https://www.sciencedirect.com/science/article/pii/S0140673618325960.

[2] Antonella Fattorusso et al. "The Pharmacoresistant Epilepsy: An Overview on Existant and New Emerging Therapies". In: *Frontiers in Neurology* 12 (2021). ISSN: 1664-2295. DOI: 10.3389/fneur.2021.674483. URL: https://www.frontiersin.org/articles/10.3389/fneur.2021.674483.

[3] P.V. Motika and D.C. Bergen. "Electroencephalography (EEG)". In: *Encyclopedia of Movement Disorders*. Ed. by Katie Kompoliti and Leo Verhagen Metman. Oxford: Academic Press, 2010, pp. 441–444. ISBN: 978-0-12-374105-9. DOI: https://doi.org/10.1016/B978-0-12-374105-9.00026-5. URL: https://www.sciencedirect.com/science/article/pii/B9780123741059000265.

[4] Mohamed Sami Nafea and Zool Hilmi Ismail. "Supervised Machine Learning and Deep Learning Techniques for Epileptic Seizure Recognition Using EEG Signals—A Systematic Literature Review". In: *Bioengineering* 9.12 (2022). ISSN: 2306-5354. DOI: 10.3390/bioengineering9120781. URL: https://www.mdpi.com/2306-5354/9/12/781.

[5] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[6] Robin Tibor Schirrmeister et al. "Deep learning with convolutional neural networks for EEG decoding and visualization". In: *Human Brain Mapping* 38.11 (2017), pp. 5391–5420. DOI: 10.1002/hbm.23730. URL: https://doi.org/10.1002\%2Fhbm.23730.

[7] Vernon J Lawhern et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces". In: *Journal of Neural Engineering* 15.5 (2018), p. 056013. DOI: 10.1088/1741-2552/aace8c. URL: https://doi.org/10.1088\%2F1741-2552\%2Faace8c.

[8] Fabien Lotte et al. "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update". In: *Journal of Neural Engineering* 15 (Feb. 2018). DOI: 10.1088/1741-2552/aab2f2.

[9] Demetres Kostas and Frank Rudzicz. "Thinker invariance: Enabling deep neural networks for BCI across more people". In: *Journal of neural engineering* 17 (Sept. 2020). DOI: 10.1088/1741-2552/abb7a7.

[10] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, eds. *Principles of Neural Science*. Third. New York: Elsevier, 1991.

[11] Syed Muhammad Usman et al. "Using scalp EEG and intracranial EEG signals for predicting epileptic seizures: Review of available methodologies". In: *Seizure* 71 (2019), pp. 258–269. ISSN: 1059-1311. DOI: https://doi.org/10.1016/j.seizure.2019.08.006. URL: https://www.sciencedirect.com/science/article/pii/S1059131119302213.

[12] Ali H. Shoeb and John V. Guttag. "Application of Machine Learning To Epileptic Seizure Detection". In: *International Conference on Machine Learning*. 2010.

[13] Alan Weintraub and John Whyte. "Electroencephalography". In: *Encyclopedia of Clinical Neuropsychology*. Ed. by Jeffrey S. Kreutzer, John DeLuca, and Bruce Caplan. Cham: Springer International Publishing, 2018, pp. 1282–1284. ISBN: 978-3-319-57111-9. DOI: 10.1007/978-3-319-57111-9_24. URL: https://doi.org/10.1007/978-3-319-57111-9_24.

[14] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning". In: *Electronic Markets* 31.3 (2021), pp. 685–695. DOI: 10.1007/s12525-021-00475-2. URL: https://doi.org/10.1007%2Fs12525-021-00475-2.

[15] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[16] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. arXiv: 2006.11477 [cs.CL].

[17] Demetres Kostas, Stéphane Aroca-Ouellette, and Frank Rudzicz. "BENDR: Using Transformers and a Contrastive Self-Supervised Learning Task to Learn From Massive Amounts of EEG Data". In: *Frontiers in Human Neuroscience* 15 (2021). ISSN: 1662-5161. DOI: 10.3389/fnhum.2021.653659. URL: https://www.frontiersin.org/articles/10.3389/fnhum.2021.653659.

[18] Thorir Mar Ingolfsson et al. *Towards Long-term Non-invasive Monitoring for Epilepsy via Wearable EEG Devices*. 2021. arXiv: 2106.08008 [eess.SP].

[19] R Zanetti et al. "Approximate zero-crossing: a new interpretable, highly discriminative and low-complexity feature for EEG and iEEG seizure detection". In: *Journal of Neural Engineering* 19.6 (2022), p. 066018. DOI: 10.1088/1741-2552/aca1e4. URL: https://dx.doi.org/10.1088/1741-2552/aca1e4.

[20] Miaolin Fan and Chun-An Chou. "Detecting Abnormal Pattern of Epileptic Seizures via Temporal Synchronization of EEG Signals". In: *IEEE Transactions on Biomedical Engineering* 66 (2019), pp. 601–608.

[21] Dong Wang et al. "Epileptic Seizure Detection in Long-Term EEG Recordings by Using Wavelet-Based Directed Transfer Function". In: *IEEE Transactions on Biomedical Engineering* 65.11 (2018), pp. 2591–2599. DOI: 10.1109/TBME.2018.2809798.

[22] Kaveh Samiee, Péter Kovács, and Moncef Gabbouj. "Epileptic Seizure Classification of EEG Time-Series Using Rational Discrete Short-Time Fourier Transform". In: *IEEE Transactions on Biomedical Engineering* 62.2 (2015), pp. 541–552. DOI: 10.1109/TBME.2014.2360101.

[23] Ibrahim L. Olokodana et al. "EZcap: A Novel Wearable for Real-Time Automated Seizure Detection From EEG Signals". In: *IEEE Transactions on Consumer Electronics* 67.2 (2021), pp. 166–175. DOI: 10.1109/TCE.2021.3079399.

[24] Ibrahim Olokodana et al. "Real-Time Automatic Seizure Detection Using Ordinary Kriging Method in an Edge-IoMT Computing Paradigm". In: *SN Computer Science* 1 (Aug. 2020). DOI: 10.1007/s42979-020-00272-2.

[25] Md Sayeed et al. "Neuro-Detect: A Machine Learning Based Fast and Accurate Seizure Detection System in the IoMT". In: *IEEE Transactions on Consumer Electronics* PP (May 2019), pp. 1–1. DOI: 10.1109/TCE.2019.2917895.

[26] Saleh Baghersalimi et al. *Many-to-One Knowledge Distillation of Real-Time Epileptic Seizure Detection for Low-Power Wearable Internet of Things Systems*. 2022. arXiv: 2208.00885 [eess.SP].

[27] Paola Busia et al. "EEGformer: Transformer-Based Epilepsy Detection on Raw EEG Traces for Low-Channel-Count Wearable Continuous Monitoring Devices". In: *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. 2022, pp. 640–644. DOI: 10.1109/BioCAS54905.2022.9948637.

[28] U. Rajendra Acharya et al. "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals". In: *Computers in Biology and Medicine* 100 (2018), pp. 270–278. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2017.09.017. URL: https://www.sciencedirect.com/science/article/pii/S0010482517303153.

[29] Wei Zhao et al. "A Novel Deep Neural Network for Robust Detection of Seizures Using EEG Signals". In: *Computational and Mathematical Methods in Medicine* 2020 (Apr. 2020), pp. 1–9. DOI: 10.1155/2020/9689821.

[30] Satarupa Chakrabarti, Aleena Swetapadma, and Prasant Kumar Pattnaik. "A channel independent generalized seizure detection method for pediatric epileptic seizures". In: *Computer Methods and Programs in Biomedicine* 209 (2021), p. 106335. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2021.106335. URL: https://www.sciencedirect.com/science/article/pii/S0169260721004090.

[31] Xiang Zhang et al. "Adversarial Representation Learning for Robust Patient-Independent Epileptic Seizure Detection". In: *IEEE Journal of Biomedical and Health Informatics* 24.10 (2020), pp. 2852–2859. DOI: 10.1109/JBHI.2020.2971610.

[32] Leilei Sun et al. "Unsupervised EEG feature extraction based on echo state network". In: *Information Sciences* 475 (2019), pp. 1–17. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.09.057. URL: https://www.sciencedirect.com/science/article/pii/S0020025518307692.

[33] Yang Li et al. "Epileptic Seizure Detection in EEG Signals Using a Unified Temporal-Spectral Squeezeand-Excitation Network". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* PP (Feb. 2020), pp. 1–1. DOI: 10.1109/TNSRE.2020.2973434.

[34] Sungmin You et al. "Unsupervised automatic seizure detection for focal-onset seizures recorded with behind-the-ear EEG using an anomaly-detecting generative adversarial network". In: *Computer Methods and Programs in Biomedicine* 193 (2020), p. 105472. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2020.105472. URL: https://www.sciencedirect.com/science/article/pii/S0169260719320000.

[35] İlkay Yıldız et al. "Unsupervised seizure identification on EEG". In: *Computer Methods and Programs in Biomedicine* 215 (2022), p. 106604. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2021.106604. URL: https://www.sciencedirect.com/science/article/pii/S0169260721006787.

[36] İlkay Yıldız Potter et al. "Unsupervised Multivariate Time-Series Transformers for Seizure Identification on EEG". In: *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2022. DOI: 10.1109/icmla55696.2022.00208. URL: https://doi.org/10.1109\%2Ficmla55696.2022.00208.

[37] Chun-Liang Li et al. *CutPaste: Self-Supervised Learning for Anomaly Detection and Localization*. 2021. arXiv: 2104.04015 [cs.CV].

[38] Yaojia Zheng et al. *Task-oriented Self-supervised Learning for Anomaly Detection in Electroencephalography*. 2022. arXiv: 2207.01391 [cs.LG].

[39] Yao Guo et al. "Epileptic Seizure Detection by Cascading Isolation Forest-Based Anomaly Screening and EasyEnsemble". In: *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society* 30 (Mar. 2022), pp. 915–924. DOI: 10.1109/TNSRE.2022.3163503.

[40] Hubert Banville et al. *Uncovering the structure of clinical EEG signals with self-supervised learning*. 2020. arXiv: 2007.16104 [stat.ML].

[41] Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. "Contrastive Representation Learning for Electroencephalogram Classification". In: *Proceedings of the Machine Learning for Health NeurIPS Workshop*. Ed. by Emily Alsentzer et al. Vol. 136. Proceedings of Machine Learning Research. PMLR, 2020, pp. 238–253. URL: https://proceedings.mlr.press/v136/mohsenvand20a.html.

[42] Junjie Xu et al. "Anomaly Detection on Electroencephalography with Self-supervised Learning". In: *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2020, pp. 363–368. DOI: 10.1109/BIBM49941.2020.9313163.

[43] Federico Pup and Manfredo Atzori. *Applications of Self-Supervised Learning to Biomedical Signals: where are we now*. Apr. 2023. DOI: 10.36227/techrxiv.22567021.

[44] Xue Jiang et al. *Self-supervised Contrastive Learning for EEG-based Sleep Staging*. 2021. arXiv: 2109.07839 [cs.LG].

[45] Siyi Tang et al. *Self-Supervised Graph Neural Networks for Improved Electroencephalographic Seizure Analysis*. 2022. arXiv: 2104.08336 [eess.SP].

[46] Thi Kieu Khanh Ho and Narges Armanfard. *Self-Supervised Learning for Anomalous Channel Detection in EEG Graphs: Application to Seizure Analysis*. 2023. arXiv: 2208.07448 [cs.LG].

[47] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML].

[48] Joseph Rocca. *Understanding Variational Autoencoders (VAE)*. 2019. URL: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[49] Guanghai Dai et al. "HS-CNN: A CNN with hybrid convolution scale for EEG motor imagery classification". In: *Journal of Neural Engineering* 17 (Sept. 2019). DOI: 10.1088/1741-2552/ab405f.

[50] Xiao Shi Huang et al. "Improving Transformer Optimization Through Better Initialization". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4475–4483. URL: https://proceedings.mlr.press/v119/huang20f.html.

[51] Hsiang-Yun Sherry Chien et al. *MAEEG: Masked Auto-encoder for EEG Representation Learning*. 2022. arXiv: 2211.02625 [eess.SP].

[52] Cory Maklin. *Synthetic Minority Over-sampling TEchnique (SMOTE)*. 2022. URL: https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c.

[53] Nour Al-Rahman Al-Serw. *Undersampling and oversampling: An old and a new approach*. 2021. URL: https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392.

[54] Iyad Obeid and Joseph Picone. "The Temple University Hospital EEG Data Corpus". In: *Frontiers in Neuroscience* 10 (2016). ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00196. URL: https://www.frontiersin.org/articles/10.3389/fnins.2016.00196.

[55] Ali Shoeb and John Guttag. "Application of Machine Learning To Epileptic Seizure Detection". In: Aug. 2010, pp. 975–982.

[56] Yikai Gao et al. "A general sample-weighted framework for epileptic seizure prediction". In: *Computers in Biology and Medicine* 150 (2022), p. 106169. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.106169. URL: https://www.sciencedirect.com/science/article/pii/S0010482522008770.

[57] Lisha Zhong et al. "Epileptic prediction using spatiotemporal information combined with optimal features strategy on EEG". In: *Frontiers in Neuroscience* 17 (2023). ISSN: 1662-453X. DOI: 10.3389/fnins.2023.1174005. URL: https://www.frontiersin.org/articles/10.3389/fnins.2023.1174005.