Università degli Studi di Napoli Federico II
Ph.D. Program in
Information Technology and Electrical Engineering
XXXV Cycle

Thesis for the Degree of Doctor of Philosophy

Machine Learning and XAI methods for improving EEG-based BCI classification systems

by
Salvatore Giugliano

Advisor: Prof. Roberto Prevete

Co-advisor: Prof. Francesco Isgrò

Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

Machine Learning and XAI methods for improving EEG-based BCI classification systems

Ph.D. Thesis presented

for the fulfillment of the Degree of Doctor of Philosophy

in Information Technology and Electrical Engineering

by

Salvatore Giugliano

October 2022

Approved as to style and content by

Roberto Prevete

Prof. Roberto Prevete, Advisor

Francesco Isgrò

Prof. Francesco Isgrò, Co-advisor

Università degli Studi di Napoli Federico II

Ph.D. Program in Information Technology and Electrical Engineering

XXXV cycle - Chairman: Prof. Stefano Russo

http://itee.dieti.unina.it

Candidate's declaration

I hereby declare that this thesis submitted to obtain the academic degree of Philosophiæ Doctor (Ph.D.) in Information Technology and Electrical Engineering is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.
Parts of this dissertation have been published in international journals and/or conference articles (see list of the author's publications at the end of the thesis).

Napoli, January 3, 2023

Salvatore Giugliano

Abstract

The use of Machine Learning (ML) techniques for EEG signal classification is gaining increasing attention in Brain-Computer interfaces (BCI) applications thanks to promising performances reported by many ML systems, from one side, and the non-invasiveness and high time resolution of the EEG acquisitions from the other one. However, several EEG-based BCI applications suffer the main drawbacks of the EEG signals, such as their non-stationarity, which makes the employing systems particularly sensitive to changes in users or time acquisitions. Performance with different acquisition times or subjects remains low in several applications. Therefore, such systems can be unreliable, particularly when used in safety-critical domains. From the ML point of view, the non-stationarity of EEG signals can be viewed as an instance of the well-known Dataset Shift problem, where, training and test data can belong to different probability distributions, leading ML systems toward poor generalisation performances. The research work of this PhD thesis was conducted with the long-term goal of exploiting the knowledge from eXplainable Artificial Intelligence (XAI) domain to develop EEG-based classification systems which overcome the performance returned by the current ones. XAI methods try to explain the behaviour of AI systems, such as ML ones, by providing explanations about the response of an AI system, given a specific input, in terms of relevant input features. More specifically, the contribution of this PhD thesis is threefold: firstly, a study on BCI systems that relied on EEG signals is made, leading to two different proposals for two different tasks: EEG-based emotion recognition and SSVEP classification. These proposals explore advanced ML techniques such as convolutional neural networks and domain adaptation methods on well-known EEG datasets. Secondly, a study on modern XAI methods is made, converging toward a new method to build explanations in an image classification task. Finally, on the basis of the results obtained in the previous investigations, an experimental analysis of explanations produced by several XAI methods on an ML system trained on EEG data for emotion recognition is made. Preliminary results suggest the plausibility to develop ML methods for BCI systems able to leverage on XAI methods to generalise across different subjects and different times without further efforts.

Keywords: Machine Learning, EEG, BCI, XAI

Sintesi in lingua italiana

L'uso di tecniche di ML per la classificazione dei segnali EEG sta guadagnando sempre più attenzione nelle applicazioni BCI grazie alle promettenti prestazioni riportate da molti sistemi ML e alla non invasività e all'alta risoluzione temporale dei segnali EEG . Tuttavia, molte applicazioni BCI basate su EEG soffrono dei principali inconvenienti dei segnali EEG, come la loro non stazionarietà, che rende i sistemi sensibili al variare degli utenti o del tempo durante le acquisizioni. Le prestazioni con tempi di acquisizione o con soggetti diversi rimangono basse in molte applicazioni. Pertanto, tali sistemi possono risultare inaffidabili, soprattutto se utilizzati in ambiti critici per la sicurezza. Dal punto di vista del ML, la non stazionarietà dei segnali EEG può essere vista come un'istanza del noto problema del Dataset Shift, in cui, i dati di addestramento e quelli di test possono appartenere a distribuzioni di probabilità diverse, portando i sistemi ML a scarse performance di generalizzazione. Il lavoro di ricerca di questa tesi di dottorato è stato condotto con l'obiettivo a lungo termine di sfruttare le conoscenze del dominio XAI per sviluppare sistemi di classificazione basati su segnali EEG che superino le prestazioni restituite da quelli attuali. I metodi XAI cercano di spiegare il comportamento dei sistemi di ML fornendo spiegazioni sulla risposta di un sistema di ML, dato un input specifico, in termini di caratteristiche rilevanti dell'input. Più specificamente, il contributo di questa tesi di dottorato è triplice: in primo luogo, viene effettuato uno studio sui sistemi BCI che si basano sui segnali EEG, portando due proposte diverse su due compiti differenti: riconoscimento dell'emozioni basato su EEG e classificazione di segnali SSVEP. Queste proposte esplorano tecniche avanzate di ML, come le reti neurali convoluzionali e i metodi di domain adaptation, su noti set di dati EEG. In secondo luogo, viene effettuato uno studio sui moderni metodi XAI, che converge verso un nuovo metodo per costruire spiegazioni in un task di classificazione di immagini . Infine, sulla base delle indagini precedenti, viene effettuata un'analisi sperimentale delle spiegazioni prodotte da diversi metodi XAI su un sistema ML addestrato su dati EEG per il riconoscimento delle emozioni. I risultati preliminari suggeriscono la plausibilità di sviluppare architetture di ML in grado di sfruttare i metodi XAI per generalizzarsi tra soggetti diversi e tempi diversi senza ulteriori sforzi.

Parole chiave: Machine Learning, EEG, BCI, XAI

Contents

List of Acronyms

The following acronyms are used throughout the thesis.

BCI	Brain-Computer Interface
SNR	Signal-to-Noise Ratio
EEG	Electroencephalography
SSVEP	Steady State Visually Evoked Potential
ML	Machine Learning
k-NN	k-Nearest Neighbor
SVM	Support Vector Machine
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
SCU	SSVEP Convolutional Unit
DS	Domain Shift
DA	Domain Adaptation

t-SNE	t-distributed Stochastic Neighbor Embedding
LOSO	eave One Subject Out
XAI	eXplainable Artificial Intelligence
MLF	Middle-Level Feature
LRP	Layer-wise Relevance Propagation
LIME	Local Interpretable Model-Agnostic Explanations

Chapter 1

Introduction

Brain-Computer Interfaces (BCIs) are an innovative technology able to create a direct communication link between the human brain and external devices, without the use of peripheral nerves and muscles, enhancing the user's ability to interact with the environment [155, 217].

Most common BCI systems use Electroencephalographic (EEG) signals to record brain activity due to their non-invasive nature and high temporal resolution. Various BCI method solutions based on EEG signals are gaining increasing appreciation from the scientific community due to their implication in the medical environment [32], as well as in other fields such as entertainment [165] and education [22].

In particular, BCI systems can be divided into two main categories: passive BCI and active BCI [13, 247].
In the passive type, the measurement and monitoring of electrical brain activity can be exploited to reveal precious information on the physiological, functional and pathological state of the brain, as well as to quantify a subject's attention or emotion levels. An example of passive BCI being used for the improvement of neuro-motor rehabilitation practices can be found in [23] where an EEG-based BCI method for the classification of engagement levels during therapy has been developed.
However, in the active BCI, EEG signals are detected to impose commands on external devices, such as a robot or mechanical limbs. Considering the need for high-performance BCI systems for online operation, special EEG signals are used such as Steady-State Visually Evoked Potential (SSVEP)

in which a specific physiological response occurs in the brain to the frequency of the visual stimulus. An example of an active BCI can be found in [21] where a single-channel SSVEP-based classification system was proposed.

There are several BCI solutions in the literature that adopt Machine Learning (ML) methods for the development of classifiers [151, 138, 137]. Typically, EEG data acquired from people subjected to known stimuli are used in the training phase. These data are labelled following an established protocol, which is task-dependent. For example, in an Emotion Recognition (ER) task, the stimuli may be images or videos which should induce particular emotions, or, in a command classification task, the stimuli may be visual stimuli that the person simply observes. Thus, labels can be deduced from the stimuli or stated by the subject, who will say whether or not he or she felt a particular emotion during the administration of the stimulus.

However, EEG-based BCI systems suffer from the main inconveniences that these signals have, such as their non-stationarity [196], in which the statistical characteristics of EEG signals change continuously over the time. In fact, also on the same subject as time changes, the EEG signal is subject to high variations.

The problem of non-stationarity can be seen as an instance of a well-known problem in the ML literature: dataset or Domain Shift (DS) problem [172], when there is a difference in probability distribution between the dataset used for training step and that used outside the training phase (evaluation or running phase). The standard ML assumption [194] of having the same data distribution for both training and test set in this scenario is not applicable. Consequently, standard ML approaches can produce ML systems with poor generalisation performance. This difference in the EEG data is even more pronounced because it is present also between examples of the same set, as EEG signals are non-stationary even when time changes on data from the same subject or in the same session.

A ML model should be able to generalise to new data taking into account these high changes in the statistical characteristics of EEG signals. Usually, many BCI systems have low performance, especially if they use models with inter-subjective (or model-independent) approaches in which an effort is made to generalise on EEG signals from new subjects.

Therefore, advanced techniques that take into consideration the problem of non-stationarity of EEG signals are needed to improve the classification performance of ML models in BCI systems.

In this thesis, both consolidated and emerging techniques such as Deep Neural Networks (DNNs) and Domain Adaptation (DA) methods will be explored to mitigate the dataset shift problem. DNNs make it possible to absorb the feature extraction phase during signal processing. Thanks to their different layers, these networks can receive as input the EEG signal in raw format and recognise specific patterns useful for correct classification. Instead, DA techniques are able to improve the performance in an ML model, especially the ability to generalise to new data. This is possible because DA methods reduce the effects of the dataset shift problem, and in this specific context, the variability of data on different subjects due to the non-stationarity of EEG signals.

An additional contribution is to use the explanations on the output of the ML model, using new techniques of eXplainable Artificial Intelligence (XAI). XAI is a sub-field of artificial intelligence (AI) that aims to explain the behaviour of AI systems, such as ML systems. There are several XAI methods in the literature that take into consideration creating good explanations that can be comprehensible to humans. For example, here [27] proposes an XAI framework for producing multiple explanations in terms of middle-level input features.

As an instance, XAI methods are applied in the application context of image recognition where the domain shift problem is slight or not present. One of the objectives of this thesis is to analyse the behaviour of several well-known XAI methods in the literature in explaining the decisions made by an ML system based on EEG input. The long-term goal is to exploit the explanations made by XAI methods to identify the main characteristics of the input for any given output, with the aim of building ML systems capable of generalising to different data from different probability distributions, in this context, sessions and subjects.

This thesis is organised as follows: in the chapter 2, there is an introduction to Machine Learning, including a description of the problems that can be solved, some of the most important algorithms and techniques present in the literature, and an analysis of particular problems; this is followed by the chapter 3, in which the EEG signal, its main characteristics

and the main signal processing phases are analysed.
After that, the thesis is divided into two parts:

- The first part is dedicated to Brain-Computer Interface in which an overview of BCI systems is given and an in-depth study of two specific problems solved by Machine Learning methods. This part is divided as follows:
  - in chapter 4, a general description of the main BCI systems is made, describing the main application scenarios and some open problems.
  - in chapter 5, an EEG-based BCI method for classifying levels of cognitive and emotional engagement is described. Different oversampling methods were used on the training data to overcome the data imbalance problem.
  - in chapter 6, several SSVEP-based systems were developed using standard and advanced ML models such as Deep Neural Networks and exploiting Domain Adaptation methods.
- The second part is dedicated to eXplainable Artificial Intelligence. This part is divided as follows:
  - in chapter 7, a brief overview is given describing the main XAI methods.
  - in chapter 8, a framework for producing multiple explanations in terms of middle-level input features is analysed.
  - in chapter 9 analyses the behaviour of different XAI methods in explaining the outputs of an EEG-based ML system.

Finally, chapter 10 concludes the thesis work by summarising the work carried out and the results obtained.

Chapter 2

Machine Learning

Our thinking machine possesses the capacity to be convinced of anything you like, provided it is repeatedly and persistently influenced in the required direction.

Georges Ivanovič Gurdjieff

Introduction

In this chapter, an introduction to machine learning will be made. Solvable problems, algorithms, techniques, evaluation methods and some special problems will be analysed.

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that exploits the use of data and algorithms to mimic human learning. The literature presents various definitions, for example, A. Samuel defined ML as *"a field of study that gives computers the ability to learn without being explicitly programmed"* [208]. E. Alpaydin defined ML as the field of *"Programming computers to optimise a performance criterion using example data or past experience"* [11]. These definitions share the basic concept of performing a task in a smart way by learning from repeated examples.

The great success of ML algorithms is largely due to their adaptation in different application contexts. Indeed, there are many problems that an ML system can solve, such as image classification [29], speech recog-

nition [75], recommendation systems [178], financial market and weather forecasting [108, 112], customer segmentation [84] and many more.

2.1 Machine learning paradigms

Machine learning is usually distinguished into four main learning paradigms: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. These paradigms differ in terms of the problems they can solve and the way the data is involved. Usually, the problem to be solved and the data directly determine the paradigm to be used.
This section provides an overview of what these learning paradigms are and what they can be used for.

2.1.1 Supervised learning

Supervised learning is the most common learning paradigm. In supervised learning, the ML model learns from a set of input-output pairs, called examples or labelled points. In particular, the tasks that can be solved with ML in this learning paradigm can be divided in classification problems and regression problems.

In classification, the objective is to predict a discrete label (or class) from a predefined list of possibilities. The class is usually represented as an integer. If there are only two possibilities, a binary classification problem occurs, otherwise, a multi-class classification problem when there are more than two classes. For example, classifying e-mails as spam or not can be considered a binary classification problem.

In regression tasks, on the other hand, the objective is to predict a real number such as the financial market performance of a particular brand. Both problems, classification and regression, fall under the supervised ML approach, where a training stage involving labelled data is performed. Therefore, this approach requires the presence of a supervisor who is able to label the data properly. Models with supervised methods learn from labelled points to infer a function associating the training data to the respective output in order to generalise to new points.

Formally, there is a set of pairs:

$$D = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \ldots, (\vec{x}^{(n)}, y^{(n)})\} \subseteq X \times Y$$

where X is the input space (usually $X \subseteq \mathbb{R}^d$ with d given) and Y (usually $Y \subseteq \mathbb{Z}$ for classification problem and $Y \subseteq \mathbb{R}$ for regression problems) is the desired output space, a supervised learning problem (see [233]) consists to infer a mapping function $f : X \to Y$ (which is called a *classifier*, if the output is discrete as in classification problems, or *regression function* if the output is continuous).

2.1.2 Unsupervised learning

Unsupervised learning is the second most widely used learning paradigm. It is not used as much as supervised learning, but it opens up different types of applications. In unsupervised learning the data is just a set of examples or points without labels. The objective is to determine patterns or hidden structural information such as groups of elements that share common properties (clustering) or representations of data that are projected from a high-dimensional space to a lower one (dimensionality reduction) [49].

One application of clustering could be to automatically separate a company's customers to create better marketing campaigns. Clustering can also be used as an exploration tool to gain insights into the available data and make informed decisions.

While, the goal of dimensionality reduction is to reduce the number of variables in a dataset while trying to preserve some properties of the data, such as distances between examples. Dimensionality reduction can be used for a variety of tasks, such as compressing the data, learning with missing labels, creating search engines, or even creating recommendation systems. Dimensionality reduction can also be used as an exploration tool to present a dataset in a reduced space in order to facilitate visualisation (see section 2.2.3).

Anomaly detection is another task that can be tackled in an unsupervised way. Anomaly detection concerns the identification of unexpected examples or events in dataset, which differ from the norm, considering them anomalous, and are also known as outliers.

2.1.3 Semi-supervised learning

In semi-supervised learning, one part of the data is labelled, as in supervised learning. Another part of the data, on the other hand, contains

only points, as in unsupervised learning.

The goal is to learn a predictive model by exploiting both data sets. Semi-supervised learning is thus a supervised learning problem in which some training labels are missing.

Typically, the unlabelled dataset is much bigger than the labelled one. One way to exploit this type of data is to use a different method between unsupervised and supervised. Another way is to use a self-training procedure in which a model is trained on the labelled data, missing labels are predicted, then trained on the entire dataset, missing labels are predicted again and so on [267].

2.1.4 Reinforcement learning

The third classical learning paradigm is called reinforcement learning, especially used with autonomous agents. Reinforcement learning is different from supervised and unsupervised learning since, the data from which to learn are obtained during the interaction with an external system called the environment. Reinforcement learning is used "to teach" agents, such as robots, to learn a task. The agent learns by performing actions in the environment and receiving feedback from this environment [119].

Typically, the agent begins the learning process by moving randomly through the environment and then it gradually learns from its experience to better perform the task, using a trial-and-error strategy. Learning is usually driven by a reward that is given to the agent based on its performance. More precisely, the agent learns a policy that maximises this reward. A policy is a model that predicts what action to take based on previous actions and observations. Reinforcement learning can be used, for example, by a robot to learn to walk in an environment.

2.2 Machine Learning algorithms

In this section, some algorithms for ML will be introduced in particular for supervised and unsupervised learning, such as k-nearest neighbour, support vector machine, neural networks, k-means and various specific architectures of neural networks.

2.2.1 Supervised methods

In this section various supervised (see 2.1.1) methods will be introduced.

k-Nearest Neighbor

K-Nearest Neighbor (k-NN) is a non-parametric ML method. It can be described as follows: given a set of already labeled points, a positive integer k, and a distance measure d (e.g., Euclidean), for a new input point p, k-NN labels p as the most present class among its k neighbors (through the measure d) that are in the labelled set.

Support Vector Machine

Support Vector Machine (SVM) is a binary classifier which separates data through a decision hyperplane. SVM considers the inputs as points in a vector space, finding an optimal hyperplane in order to maximize the distance from the class boundaries. Given a set of examples for training, each labelled with the class to which it belongs between the two possible classes, SVM builds a model that assigns the new examples to one of the two classes. An SVM model creates a representation of the examples as points in space, mapped in such a way that examples belonging to the two different categories are clearly separated by as large a space as possible. New examples are then mapped in the same space and the prediction of the category to which they belong is made on the basis of the side in which they fall.

Artificial Neural Network

Artificial Neural Network (ANN) aims to simulate the brain's activity in solving problems by mimicking the low-level functions of biological neurons. ANNs are widely used in pattern recognition and have the ability to learn from training data. An ANN consists of a number of elements, called neurons, arranged together into a structure. There are different architectures of neural networks, the ones that will be used in the following chapters are feed-forward networks. Networks in which there are the following properties:

- there are no cycles, as it is always possible to associate integers (indices) with inputs and nodes in such a way that each node only receives connections from inputs or nodes that have a lower index;
- the activation sequence is asynchronous and follows the topological order defined by the connections;

A sub-type of feed-forward networks are networks organised in layers. The network can be layered to have nodes on different layers: l disjoint subsets of layers $l_1, .., l_L$ (figure 2.1). The nodes of the last layer form the output layer while the other nodes make up the inner or hidden layers.

A particular architecture of layered feed-forward neural networks are fully-connected networks where each neuron is connected to all the nodes in the previous layer.

Usually, when than three layers (one for the input layer, one for the output layer and more than one inner layer) a Deep artificial Neural Network (DNN) model is defined. Each layer has weighted connections (W) entering from the previous layer and outgoing in the next one, so the propagation of the signal occurs forward without loops and without cross connections. The goal of the learning phase is to find the weights W that minimise the error function $E(W)$.

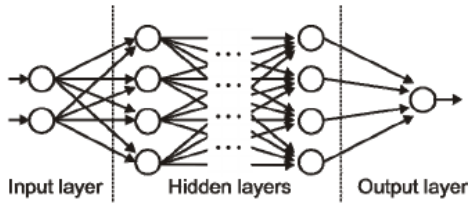


Figure 2.1. Example of an artificial neural network with several layers.

Convolutional Neural Network

Convolutional Neural Networks (CNNs), introduced by LeCun [142], are composed of several structured layers. CNNs are characterised by convolutional layers that have these features:

- sparsity of connections: each neuron in the layer is connected to a small number of neurons in the previous layer, so it performs local processing and there is a reduction in connections.
- sharing of weights: the weights are shared, in this way neurons of the same layer work on different points of the input and the number of weights is reduced.

In a convolution layer, the input is divided into overlapping regions of fixed size and each neuron is only connected to a single region of the input. The overlap and distance between consecutive windows is determined by a parameter called *stride*. The weight-sharing mechanism is obtained through the convolution operation [81]: the weights are stored in a matrix called *filter* that acts locally on each individual input window through the convolution operator; the output of the convolution operation is then used as the argument of an activation function; the final output is called *feature map*. Each convolutional layer can have many filters, and thus can produce many feature maps; thus, each filter can be seen as a group of neurons that share the same weights and that, through a convolution operation, act on different regions of the input. Usually, a *pooling* operation follows for each convolutional layer, whose task is to extract useful statistics from the local areas of the input.

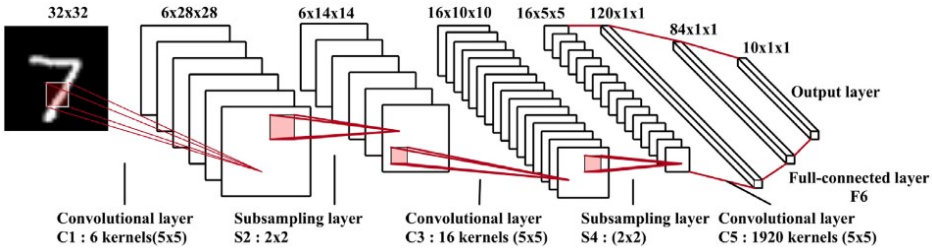


Figure 2.2. The architecture of the LeNet-5 network [97].

Specific neural network architectures for EEG data

This section describes some ML architectures, available in the literature, used to classify Electroencephalographic (EEG) signals (refer to chapter 3 for an analysis of EEG signal).

Deep SSVEP Convolutional Unit. In Aznan et Al. [39], the Deep SSVEP Convolutional Unit (SCU) neural network architecture was proposed, showing promising results in classification tasks using Steady-State Visually Evoked Potentials (SSVEP) signals (see section 3.1.2) as input. This processing strategy adopts all the EEG samples acquired in the time window. It consists of one or more neural network layers blocks (defined *SCU blocks*). Each SCU block is composed of the following layers:

- *1D Convolutional layer*:a 1D convolution is performed on the EEG samples. The time window (kernel) scrolls along one dimension, returning a feature maps on the basis of the number of filters chosen.

- *Batch Normalization layer*: a transformation is applied in order to keep the average and the standard deviation of the output close to 0 and 1, respectively.

- *Max Pooling layer*: it down-samples the input representation of the previous layer by taking the maximum value on a spatial window of size equal to 2.

- Rectifier Linear Unit (*ReLU*) activation function: it is applied at the end of each SCU block. It is a function that returns 0 if it receives negative input, otherwise it returns the received value, thus increasing the sparsity in the output.

Finally, fully-connected (Dense) layers equipped with *ReLU* activation functions are used as final layers of the network (see fig. 2.3).

PodNet. *PodNet* is a CNN developed by Podmore et Al. [192]. It is constituted by a number of blocks (called *Pods*), each one made up of a *Convolutional* layer, a *Drop-out* layer, a *Batch Normalization* layer, a *Rectifier Linear Unit (ReLU)* layer, and a *Max Pooling* layer. The final *Pod* contains a dense layer which outputs to a *Softmax* operation to classify the EEG. In fig. 2.4 a sketch of the architecture is shown. All network weights are initialized using the Xavier method [94] and updated following the Adam optimization algorithm [129].

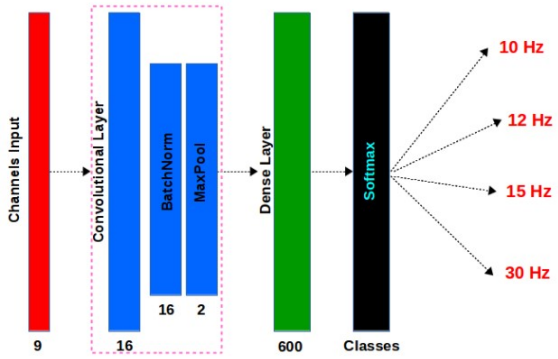


Figure 2.3. SSVEP Convolutional Unit (SCU), highlighted in pink [39].

EEGNet. *EEGNet* is a convolutional neural network originally designed to be applied to a wide variety of brain-computer interface paradigms (for details, please refer to [140]). It is composed of three main blocks.

- In the first block, two convolutional steps are performed in sequence. First, a number F_1 of *2D convolutional filters* are fitted. These convolutional filters output F_1 feature maps containing the EEG signal at different sub-bands. By properly setting the size of the temporal kernel, it is possible to capture frequency information at the desired resolution. The second step is a *depthwise convolution* of size C, where C is the number of EEG channels. This helps (i) to reduce the number of trainable parameters to fit, and (ii) to learn spatial filters for each temporal filters, enabling the efficient extraction of frequency-specific spatial filters. A depth parameter D controls the number of spatial filters to learn for each feature map. Both these convolutions are kept linear as no gains in performance when using nonlinear activation functions have been observed. Also, each spatial filter is regularized by using a maximum norm constraint of 1 on its weights. Then, Batch Normalization is applied along the feature map dimension, before applying the Exponential Linear Unit (ELU) as activation function. An *average pooling* layer is used to reduce the sampling rate of the signal. Finally, a *dropout* technique is adopted to regularize the model.

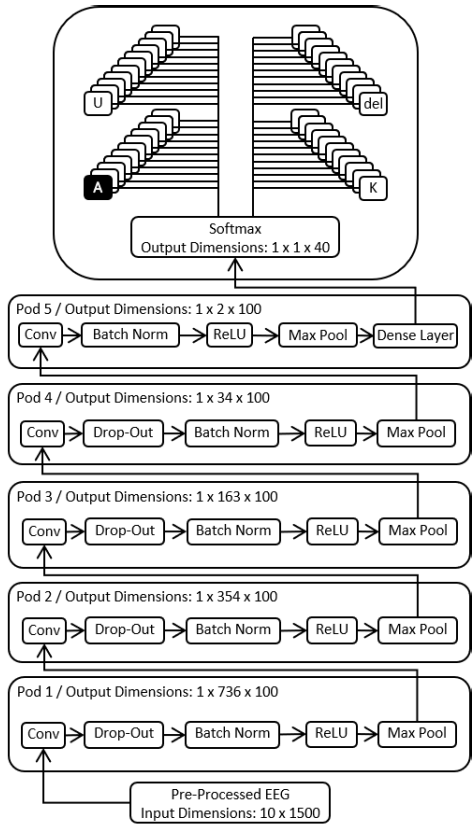


Figure 2.4. Diagram of PodNet configuration [192].

- In the second block, a *separable convolution* is used. It is a *depthwise convolution*, followed by F_2 *pointwise convolutions*. This helps to (i) reduce the number of parameters to fit, and (ii) explicitly decouple the relationship within and across feature maps. This operation is particularly useful for EEG signals, since different feature maps could represent data at different time-scales of information. After *batch normalization* and ELU application, an *average pooling* layer is used for dimension reduction. Finally, a *dropout* is applied.

- In the *classification block*, the features are passed directly to a *softmax* classification with N units, where N is the number of classes in

the data. The use of a dense layer for feature aggregation prior to the softmax classification layer is omitted to reduce the number of free parameters in the model.

2.2.2 Unsupervised methods

In this section, a particular unsupervised (see sec. 2.1.2) method algorithm will be introduced.

K-means

K-means [162] is one of the best known unsupervised learning algorithms. The aim of the algorithm is to minimise the total intra-group variance. Each group is identified by a centroid or midpoint. The algorithm follows an iterative procedure: initially it creates k partitions and assigns the entry points to each partition either randomly or using some heuristic information. It then calculates the centroid of each group. The algorithm then constructs a new partition associating each entry point with the group whose centroid is closest to it. Finally, the centroids for the new groups are recalculated, and so on, until the algorithm converges.

2.2.3 High-dimensional data visualisation techniques

The datasets used in machine learning are usually characterised by high dimensionality. For example, the dimensionality of an image is the number of pixels, or in an electroencephalographic trace are the number of electrodes (channels) and the signal update time (sample rate) for each channel. Since high-dimensional data, specifically data that require more than two or three dimensions to represent, can be difficult to interpret, the aim is to look for ways to effectively visualise such high-dimensional data before moving on to a processing step or to consolidate certain properties of the data.

To facilitate the visualisation of the structure of a dataset, it is necessary to reduce the dimensions in some way. Several linear dimensionality reduction algorithms have been designed such as Principal Component Analysis (PCA) [3] and Independent Component Analysis (ICA) [143]. These methods can be powerful, but often do not take into account important non-linear structures in the data.

Manifold learning can be seen as an attempt to generalise linear frameworks to be sensitive to non-linear data structure. Although supervised variants exist, the typical manifold learning problem is unsupervised: it learns the high-dimensionality structure of the data from the data itself, without the use of labels.

Two methods will be reported below: t-distributed stochastic neighbor embedding and isometric mapping.

T-distributed stochastic neighbor embedding

The t-distributed Stochastic Neighbor Embedding (t-SNE) [231] algorithm consists mainly of two steps. First, t-SNE creates a probability distribution over pairs of high-dimensional objects, associating similar objects with a higher probability and dissimilar points with a lower probability. Secondly, t-SNE defines a similar probability distribution on low-dimensional map points and minimises the Kullback-Leibler divergence (KL divergence) between the two distributions with respect to the positions of the points in the map. Although t-SNE plots often appear to show clusters, visual clusters can be strongly influenced by the parameters chosen and therefore a good understanding of the parameters of t-SNE is necessary.

Isometric Mapping

Isometric Mapping (Isomap) [229] is a non-linear dimensionality reduction method that seeks to preserve geodetic distances in the lower dimension. Isomap starts by creating a proximity network. Then, it uses the distance of the graph to approximate the geodesic distance between all pairs of points. Through the decomposition of the eigenvalues of the geodesic distance matrix, it finds the low-dimensional embedding of the dataset. In non-linear manifolds, the Euclidean distance metric is valid if and only if the neighbourhood structure can be approximated as linear. If the neighbourhood contains holes, Euclidean distances can be highly misleading. Conversely, if distance between two points is measured following the structure of the manifold, a better approximation of how far or close two points are will be obtained.

2.3 Selection and evaluation of Machine Learning models

ML models have to deal with two main issues: the search of optimal values for hyper-parameters and the over-fit problem. These issues are closely related to the validation phase.

2.3.1 Finding optimum values for hyperparameters

In addition to the weights or internal parameters that update during the learning phase, ML models need multiple hyperparameters (or free parameters), such as the number of neurons in a specific layer in an artificial neural network model or the regularization parameter in a SVM model. The values of these parameters can heavily change the performance of the model in a given task. It is good practice to find a good set of hyperparameters for a specific problem and model, but also good associated values. The values of the hyperparameters are difficult to establish a priori, which is why a technique known as hyperparameters tuning is used: a search mode for establishing the optimal values of hyperparameters. Different values can be arbitrarily chosen for each of these hyperparameters. The two main techniques for finding optimal values for hyper-parameters are as follows.

Grid search

For each hyper-parameter, a set of values in a given range is defined. For example, if two hyper-parameters are chosen and for each there is a set of 7 values, this will result in a 7x7 grid of possible values for a total combination of 49 models. From the best model in the validation phase, the values per hyper-parameter that will be the optimal ones will be selected. Thus, a grid of values is used for each hyper-parameter to be optimised and all models are tested with all possible combinations of the chosen parameters. This approach is recommended if the number of combinations is relatively low in relation to the time of the learning phase for each model.

Random Search

As reported in [46], a random approach in the selection of values per hyper-parameter is more efficient than grid mode. This is because by fixing values in a grid, the coverage of possible patterns is lower and less significant than by randomly selecting values. Figure 2.5 compares the two modes, where it can be seen that by having two hyper-parameters and 3 values per hyper-parameter, in the grid search these values generate only 3 significant models (it is also considered that on the x-axis there is a more significant hyper-parameter than on the y-axis); whereas in the random search, having the values randomly selected results in a greater number of 9 significant models. In other words, the point projections in the random case provide better coverage.

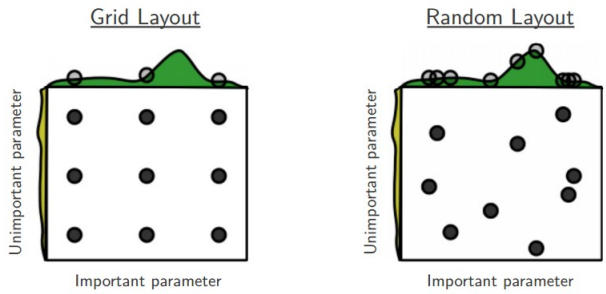


Figure 2.5. Grid search and random search compared [46].

2.3.2 Over-fit

Usually, to evaluate the performance of an ML model, the dataset is divided primarily into two parts, a training set Tr and a test set Te. As will be seen 2.3.3, in the validation modes, there are several strategies to partition the dataset according to the application domain as well.
A model is over-fit on the training set when it returns high performance on training data and low performance on new or test data. As the authors assert in [191] "when a model is chosen because of qualities exhibited by a particular set of data, predictions of future observations that arise in a similar fashion will almost certainly not be as good as might naively be expected". Therefore, it is important to use methods to increase the

generalisation of ML models to new data, and also to have strategies for the accurate evaluation of this characteristic.

A classic example of how the over-fit problem is handled on neural network-based ML models is to use a stop criterion during the learning phase. In practice, an additional set, the validation set, subset of the dataset and disjoint from the training and the test sets, is used only to tune the hyperparameters of a classifier. It is useful to understand when the network no longer generalise and thus stop the learning process. Figure 2.6 illustrates an example of the over-fit process and the point at which training should be stopped so that the performance on the validation set does not decrease.

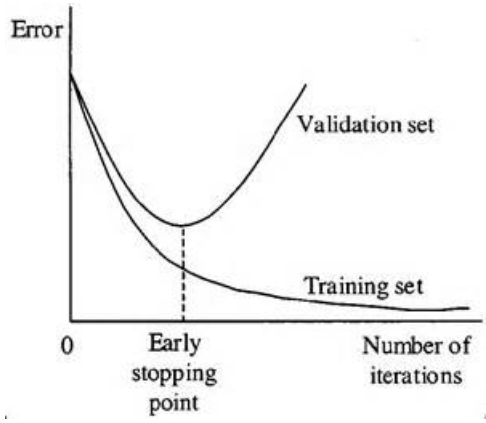


Figure 2.6. Example of overfitting with the stop criterion on machine learning models based on neural networks.

2.3.3 Model selection in cross-validation

Considering the problems analysed above, methods for correctly selecting an ML model are necessary. A simple form of validation is called hold-out validation [186], when the dataset is divided in two parts. It has a major disadvantage in that the evaluation depends heavily on how the dataset is split. One solution is to average the performance over several splits, resulting in a cross-validation estimate.

Cross-validation (CV) techniques are indispensable to understand when

a model is good in terms of generalisation and whether or not it should be chosen [173]. There are several methods to perform a CV, such as:

- Leave-P-Out cross-validation: p example as the Te and the remaining observations as the Tr. This is repeated on all ways to cut the original sample on a Te of p examples and a Tr.
- Leave-One-Out cross-validation: particular case of Leave-P-Out CV with p = 1.
- Leave-One-Group-out cross-validation: another particular of leave-p-out cross-validation that provides to split data according to a group information used to encode arbitrary domain specific stratification of the samples. A simple variant of this is the Leave-One-Subject-Out (LOSO) CV technique. It is widely used in the literature, for example, to test the generalisation of classifiers in brain-computer interface systems. It uses the subject information associated with the sample as a group.
- K-Fold cross-validation: a k-partition P of the data is made; a set from P is taken as Te, and the union of remaining set in P is taken as Tr; the process is then repeated for every possible pair (Tr, Te) in the partition.

CV is usually associated with the search of optimal values for hyperparameters (sec. 2.3.1) to determine the best model.

2.3.4 Metrics

In the ML, there are several metrics that help measure the effectiveness of model training. For classification problems, the terms True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are used to define metrics. The terms positive and negative refer to the classifier's prediction, while the terms true and false refer to the correspondence of that prediction with the trusted values. The most commonly used metrics in machine learning are:

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

Accuracy generally describes how well the model performs, calculating the ratio of the number of correct predictions to the total number of predictions.

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

Precision is calculated as the ratio of the number of correctly classified positive samples to the total number of samples classified as positive (both correctly and incorrectly).

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

Recall is calculated as the ratio of correct predictions for a class to the total number of cases in which it actually occurs.

2.4 Special problems

This section will discuss some issues that can be encountered in machine learning, such as unbalanced data and domain shift.

2.4.1 Unbalanced data

In an ideal training set for a classification problem, there should be the same number of instances for each class. It may occur, however, that there is an imbalance per class. Creating a good model becomes a real challenge due to the under-representation of minority classes instances. Consequently, even if the overall classification model achieves high accuracy, minority class results can be poor [212].

An important cause of unbalanced class representations in data sets is that some classes are difficult to collect or are rarely observed. This commonly occurs in data acquired from healthcare fields. For example, compared to healthy people, sick patients represent only a small part of the total population. More serious diseases, such as cancer and AIDS, have fewer cases compared to other less critical conditions. For example, if we want to discriminate between cancer-affected and healthy patients on a given dataset, the amount of healthy samples can be dominant, leading the model toward to have a poor discriminating ability on healthy patient sam-

ples. It is it is straightforward to say that identifying a cancer patients as a healthy patient is a more serious mistakes than vice versa [147]. Another example may be the number of subjects involved in the data acquisition campaign, if the number of involved subjects is too small, it is hard to lead the model toward a good generalization ability among different subjects.

Most solutions to this problem are data-level, in particular with under-sampling and over-sampling methods. In the preprocessing phase, the data are sampled to balance the distribution of samples per class. One of the most frequently used technique in the literature is Synthetic Minority Over-Sampling Technique (SMOTE) [59] and its variants: BorderlineSMOTE [106], Adaptive Synthetic sampling (ADASYN) [103], SVMSMOTE [181] and KMeansSMOTE [80].

SMOTE creates new synthetic data interpolating the data of the minority class. In particular, given a random sample point t belonging to the minority class, a randomly selected neighbor t' is chosen between the h nearest neighbors of t. Thus, a synthetic data point between t and t' is created. The number h of neighbors was considered as an hyper-parameter to be tuned during the validation procedure.

ADASYN expands the basic idea of SMOTE by adding a criterion to automatically decide the number of data to generate in the neighborhood of each minority sample. Relying on the assumption that data near the class decision boundary could be misclassified, BorderlineSMOTE generates instances in the border area between the classes. Similarly, SVMSMOTE creates boundary data exploiting an SVM to approximate a good boundary between the classes. Instead, with the help of the k-means [162] clustering algorithm, KMeansSMOTE selects the best domain areas to over-sample.

Usually, only the training set is balanced and then different metrics to validate the model on the test set are used. Indeed, in case of imbalanced test data, established performance measures, as the standard accuracy, are unreliable since they can be biased toward the dominant class [87]. Some metrics used as performance scores are the Matthew Correlation coefficient (MCC [42]) and the Balanced Accuracy (BA [54]).

Specifically, MCC and BA are defined as:

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

and

$$BA = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

MCC showed to be particularly reliable in machine learning problems with unbalanced data [53]. MCC is the correlation coefficient between the observed data and the predicted classifications. It is defined in the range $[-1, 1]$, where a MCC value of 1 means a perfect prediction, 0 a random prediction and -1 means a total misclassification. Balanced accuracy, similarly to the standard accuracy , is defined in the range $[0, 1]$, giving a more intuitive performance measure of the proposed method in imbalanced data condition.

2.4.2 Domain Shift problem

The Domain or Dataset Shift (DS) problem refers to the probability distribution shift that is present between the training set and the test set [172] in a given dataset. Most ML algorithms are based on the assumption that the two sets, training and test, are independent and identically distributed, ignoring the out-of-distribution scenarios that commonly occur in practice. This means that these models are not designed with the problem of domain switching in mind, and as a result, an ML model trained only with training data will suffer a significant drop in performance on the test set.

There can be several causes of dataset shift, the two most important are: sample selection bias and non-stationary environments. In the first case, the difference in the distribution is due to the fact that the data of the training set was obtained by biased methods, and thus does not truthfully represent the real operating environment. For example, due to cost, examples of a class are sampled at a lower rate than what the reality may be. Second, it appears when the training environment is different from the test environment, due to a spatial or temporal change. For example, in some scenarios, such as the medical one, where there are different patient data, the probability distribution is also variable between the individual subject data. On data of Electroencephalographic (EEG) signals due to their properties, as will be seen in detail in section 3.1.3, the distribution is even variable over time on the single subject. This property, called *non-stationarity*, of the EEG signal can be seen as an instance of the dataset

shift problem.

Domain adaptation

Possible solutions in the literature for this problem are based on Domain Adaptation (DA) [134] approaches. These solutions assume that unlabelled test set data are also available during the training phase.

Simple methods, in which normalisation functions are used, have been proposed in the literature in various works [62, 61]. In the preprocessing phase, the data are usually transformed using normalisation functions, such as re-scaling the data with respect to its mean and variance. The choice and mode of normalisation can impact the performance of classification in machine learning systems.

Considering a dataset, such as EEG data, consisting of data from different subjects and sessions for different subjects, different schemes to normalise the data are possible to use, such as:

1. All subjects; the entire dataset is normalised. This is the most widely used method where, however, if no test set data is available, it is not considered a DA technique.

2. Per subject; each subject is normalised individually. Here, it is assumed that each subject has a different distribution and therefore exploiting this type of transformation leads to an improvement in classification performance.

3. Per session; each session for each subject has its own normalisation. A special case of the previous one, in which, even sessions per subject can have different domains.

2.5 Conclusion

In some specific contexts the well-known DS problem leads ML systems to poor generalization performances. Therefore, advanced techniques that consider this problem are needed to improve the classification performance of ML models. DA methods, which can also make use of DNN, can be used to mitigate the dataset shift problem. DA techniques are able to improve the performance of an ML model in case of different data distribution

probabilities, leading the learned model toward the ability to generalise on new data. This is possible because DA methods reduce the effects of the dataset shift problem. DNNs, on the other hand, with their different internal layers, recognise specific patterns on the inputs useful for correct classification.

In several medical scenarios in which there are data from different patients, such as in an EEG dataset, data probability distributions change drastically among different subjects. Therefore, the probability distribution is time-varying among different subjects. The non-stationarity of the EEG signal, that can be seen as an instance of the dataset shift problem, will be discussed in the following chapters.

Chapter 3

Electroencephalographic Signals

Introduction

This chapter will introduce electroencephalographic signals, analysing the main types, characteristic properties and the main steps after signal acquisition.

3.1 Description

Electroencephalographic (EEG) signals measure the neurons' electrical activity in the brain. The collection of EEG signals requires electrodes to be placed on the scalp of the human head. Since the current is measured on the scalp, obstructions (e.g. the skull) greatly reduce the quality of the signal: the fidelity of the collected EEG signals, measured as Signal-to-Noise Ratio (SNR), is about 5% of that of the original brain signals [43]. EEG signals have a low spatial resolution due to the limited number of electrodes that can be placed. Whereas, the temporal resolution is above 1000 Hz as the electrical activity changes rapidly. Electrodes are usually installed in a headset, which makes the instrumentation portable and accessible for most uses.

EEGs are of many different types, in particular: spontaneous EEG and evoked potentials.

3.1.1 Spontaneous EEGs

In spontaneous EEGs the brain signals are measured under a specific state without external stimulation. For example, while the user is sleeping or performing a mental exercise. In this specific class, the aim is to monitor the user's attentional or emotional level and engagement. Machine learning models based on spontaneous EEG are a real challenge to train due to low SNR and higher variability across subjects [189].

3.1.2 Evoked potentials

Another class of EEG signals are Evoked Potentials (EP). An EP usually has a higher amplitude and lower frequency than a spontaneous EEG signal. This characteristic allows EP signals to be more robust across subjects. These signals can be divided into Event-Related Potentials (ERP) and Steady State Evoked Potentials (SSEP) based on the frequency of the external stimuli. The stimuli in an ERP scenario are separated from each other by long intervals, whereas in SSEP they are generated by repetitive periodic stimuli at a constant frequency. Based on the type of stimulus, SSEP can be divided into three categories: Steady-State Visually Evoked Potentials (SSVEP), Steady-State Auditory Evoked Potentials (SSAEP), and Steady-State Somatosensory Evoked Potentials (SSSEP).

Steady-State Visually Evoked Potentials

In particular, SSVEPs are a specific physiological brain response to continuously flickering visual stimuli, typically inducted after a latency varying from 80 ms to 160 ms [118]. Stimulation frequency bands usually range from 6 Hz to 30 Hz, although the best Signal to Noise Ratio (SNR) is achieved in the range 8-15 Hz [240]. Generally, the SSVEP shows a sinusoidal-like waveform, with a fundamental frequency equal to that of the gazed stimulus, and often higher harmonics [175], as shown in fig. 3.1. In practical applications, different visual stimuli (at different frequencies) are associated to specific commands: thus, such systems allow the user to perform a selection by simply looking at the related flickering stimulus.

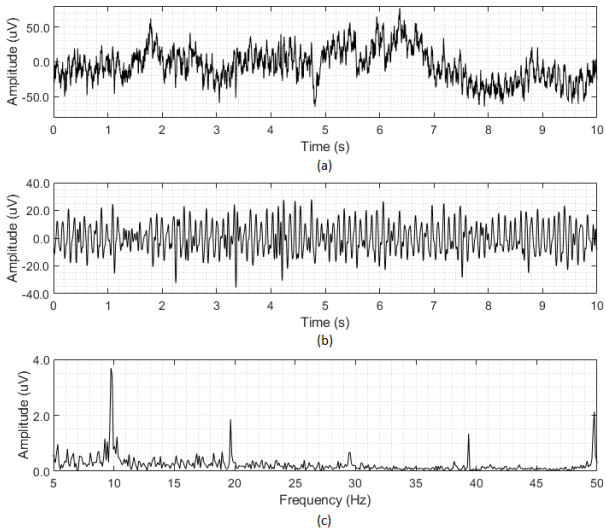


Figure 3.1. Example of a SSVEP of a user staring at a 10 Hz-flickering stimulus: EEG in the time domain (a); Filtered EEG in the time domain (b); EEG in the frequency domain (c) [21].

3.1.3 EEG signal properties

There are problems related to the electrophysiological properties of recorded EEG signals that include non-linearity, non-stationarity and noise.

The brain is a highly complex non-linear system in which chaotic behaviour of neurons can be detected. Therefore, brain signals can be characterised better with non-linear dynamic methods than with linear methods.

The non-stationarity attribute of brain signals represents a major problem in the development of a machine learning model [196]. The signals acquired change continuously over time, both between and within recording sessions. Non-stationarity can be seen as an instance of a familiar problem in the ML field known as 'domain shift' (see section 2.4.2). The underlying mental and emotional state in different sessions can contribute to the variability of signals. Fatigue and concentration levels are also considered factors of internal non-stationarity. Although, there are some types of signals, such as SSVEP signals, in which a stationary component of the signal

is prevalent in addition to the non-stationary one [116].

Noise is also significantly another problem. It includes unwanted signals caused by alterations in electrode positioning and environmental noise. Frequency band filtering helps remove noise and artefacts (see section 3.2). It can also provide significant help in managing non-stationarity factors. The advantage of using filtering is its simplicity, however, the effect of this method is degraded if the uncorrelated signal overlaps or is in the same frequency band as the signal of interest [268].

3.1.4 10/20 system

The 10/20 system is a universally recognised method that indicates the position of the electrodes on the scalp [163]. The numbers 10 and 20 indicate that the distances between adjacent electrodes are 10 or 20 per cent of the total front-to-back or right-to-left distance of the skull. At each site, a letter is used to indicate the lobe, while the position of the hemisphere is represented by a number. In the 10/20 system, the frontal, parietal, temporal and occipital lobes can be indicated by the letters F, P, T and O, respectively, as illustrated in figure 3.2. The letter Z (zero) indicates that the electrode is positioned on the midline. Even numbers (2, 4, 6, 8) are used to indicate right-hemisphere electrode positions, while left-hemisphere electrode positions are indicated by odd numbers (1, 3, 5, 7).

3.1.5 EEG devices

This section describes the devices for acquiring EEG signals: Emotiv Epoc+, Single-channel wearable and Neuroscan Synamps2 used in the tasks described into chapters 5, 6 and 9.

Emotiv Epoc+. The Emotiv Epoc+ is provided with a rechargeable battery and is able to transmit the data via Bluetooth. The electrodes, arranged on the scalp according to the International Positioning System 10/20, are placed on: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. P3/P4 and CMS/DRL (reference electrodes). A felt pad is placed on the electrodes coated with Ag/AgCl. The felts must be soaked

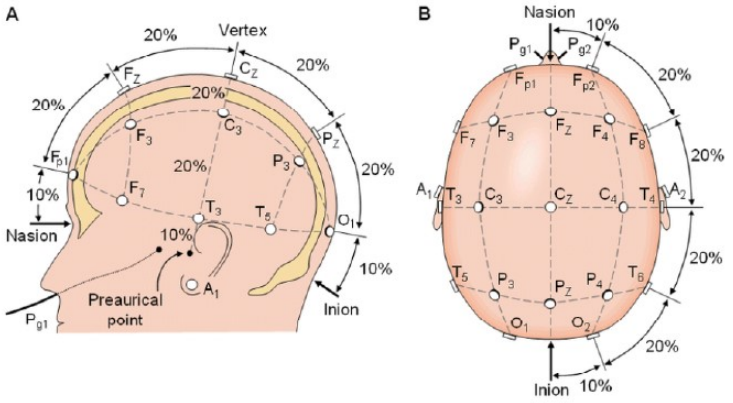


Figure 3.2. (A) and (B) are the left and above view of the international 10-20 system [114].

with an hydrator fluid like saline solution. The technical specifications of the *Emotiv Epoc+* are reported in [1].

Single-channel wearable. In this device, only three *Electrodes* are used: a pair of active and dry [68] electrodes are placed in *Oz*, *Fz* positions according to the 10-20 International System to capture the user EEG signal; while a passive electrode (*Driven Right Leg*, *DRL*) is placed on the earlobe and acts as a reference. In this way, a single-channel, differential configuration is implemented, reducing the common mode interference.

Neuroscan Synamps2. A *Synamps2* EEG acquisition unit (Neuroscan, Inc.) was used to record EEG data at a sampling rate of 1 kSa/s. 64 electrodes, according to international 10–20 system, were used to record whole-head EEG. The reference electrode was placed at the vertex of the user scalp (Cz). A notch filter at 50 Hz was applied to remove the power-line noise.

3.2 EEG signal preprocessing

Once the EEG signal data have been acquired, they are processed through preprocessing, feature extraction and classification steps. In the

first step, the signals are preprocessed in order to increase the SNR. The preprocessing phase can contain several sub-steps.

3.2.1 EEG filtering

In the acquisition process, the signal is usually contaminated with noise from various sources. The objective of this step is to eliminate or attenuate the noise in order to have a cleaner signal free of artefacts. Filters are applied so as not to introduce any change or distortion in the signals. High-pass filters with a cutoff frequency of usually less than 0.5 Hz are used to remove very low-frequency noise components, such as breath noise. Instead, high-frequency noise is attenuated using low-pass filters with a cutoff frequency of around 50-70 Hz. Notch filters with a null frequency of 50 or 60 Hz are often necessary to ensure the exclusion of the power line that creates noise at that frequency [209].

Below is a description of two particular filters: Butterworth filter, used for the task described in chapter 5 and the FIR filter exploited in the other tasks on SSVEP signals.

Butterworth filter. The Butterworth filter is a standard first-order low-pass filter, which can be modified to obtain a high-pass filter, and combined in series with others to obtain band-pass filters, band-elimination filters, and higher-order versions of these. The frequency response of these filters in the pass band is as flat as possible, while out-of-band has a monotonic transfer function, tending to zero. The Butterworth filter is the only filter that maintains the same response even for higher orders, with the steepest side slopes as the order increases.

Finite impulse response filter. Finite Impulse Response (FIR) filter is a filter whose impulse response has a finite duration, because it stabilises at zero in a finite time. This is in contrast to Infinite Impulse Response (IIR) filters, which can have internal feedback and can continue to respond indefinitely. An FIR filter has useful properties. FIR filters do not require feedback. This means that any rounding errors are not aggravated by the summed iterations. The same relative error occurs in each calculation. This also simplifies implementation. They are inherently stable, as the output is a sum of a finite number of finite multiples of the input values.

The main disadvantage of FIR filters is that they require considerably more computing power than an IIR filter.

3.2.2 Segmentation

It is common practice to divide the signal into smaller epochs in time. This segmentation makes it possible to give the signal discrete stationary properties and exploit them to produce online and real-time commands. Typical signal segmentations, but which depend on the type of problem, are $1, 2, 3, ..., 10$ seconds.

3.2.3 Normalization

Another preprocessing step is that of normalization in which the signal is normalised for each signal channel along the time-axis. There are several techniques to perform this normalisation such as: considering a $[minimum, maximum]$ interval usually equal to $[0, 1]$ and re-scaling the signals in this interval; or, computing the mean (u) and the standard deviation (s) of the acquired signals along each features and applying a $z-score$ transformation to the signals x: $z = (x - u)/s$. Then, in this normalisation phase, the signal data are finalised to give them as input to a feature extractor or to a machine learning model (deep learning).

Considering the non-stationarity of EEG signals, special normalisation schemes as described in section 2.4.2 should be taken into account.

3.3 EEG feature extraction

In this second phase of signal processing, after preprocessing has been performed, features are extracted from the data.

This part of data processing is done with specific algorithms that allow the extraction of significant features or patterns from the signal. Features are extracted according to two specific domains: time domain where there are features such as mean, standard deviation, variance and others; frequency domain where there are features such as, bands power, power spectral density values and others.

In some machine learning models such as deep models, this phase could be absorbed into the model itself. Indeed, within the different layers of the

deep model, feature extraction is done and finally other layers or a final layer will take care of the classification of the input signal.

Next, some techniques to extract features such as the common spatial pattern, used in Chapter 5, and other methods exploited on SSVEP signals in Chapter 6.

3.3.1 Common spatial pattern

The Common Spatial Pattern (CSP) is a procedure used in signal processing to separate a multivariate signal into sub-components with the highest variance differences between two windows [133]. The CSP method can be applied to multivariate signals and, in general, is commonly applied to EEG signals. In particular, the method is often used in brain-computer interfaces to retrieve component signals that best transduce brain activity for a specific task. It can also be used to separate artefacts from EEG signals [188].

3.3.2 Power spectral density analysis

The Power Spectral Density Analysis (PSDA) [240] uses a Fast Fourier Transform (FFT) on the EEG signal. Then, a PSD is performed in the neighborhood of each frequency and eventually its multiple m harmonics. This method requires a minimum time duration of the acquired EEG in order to correctly discriminate the harmonics, since an appropriate frequency resolution is required [66].

3.3.3 Canonical correlation analysis

Canonical Correlation Analysis (CCA) in time domain, is a multivariate statistical method of correlating linear relationships between two sets of data [156]. CCA is performed between the EEG data and a set of sine waves having the same frequencies of the stimuli, and eventually its multiple harmonics, and variable phase. A correlation coefficient ρ_{mn} is extracted for each stimulus frequency f_n and each harmonic m considered. Therefore, these coefficients are used for SSVEP classification. For the sake of example, in [156] the output of the classification was associated to

the frequency with the highest correlation coefficient extracted. The classification performance achieved with the use of CCA are typically better than PSDA [104].

3.3.4 Filter bank canonical correlation analysis

The Filter Bank Canonical Correlation Analysis (FBCCA) method is an enhancement of CCA [65] and consists of three major procedures: (i) filter bank analysis; (ii) CCA between SSVEP sub-band components and sinusoidal reference signals; and (iii) signal classification. First, sub-band decompositions are performed by the filter bank analysis by means of multiple filters with different pass-bands. In this way, the sub-band components from the original EEG are obtained. After the filter bank analysis, the standard CCA is applied to each of the sub-band components separately. This results in correlation values between the sub-band components and the sinusoidal reference signals corresponding to the stimulation frequencies. A correlation value is obtained for each frequency and each sub-band according to. Finally, the signal classification is performed on the basis that the observed frequency is that corresponding to the feature with the maximum value.

3.4 EEG and Machine Learning

In the classification step, a given input signal has to be assigned to a specific class. This step corresponds to determining target class using feature vectors. These vectors may be low-dimensional, in which case feature extraction is done on the data, or raw, in which the data has usually only been preprocessed. The classification can be made simply by setting thresholds per feature, or using more complex machine learning algorithms. Many classification techniques have been introduced in recent decades, as: SVM, ANN, CNN and many others (see section 2.2).

Unlike traditional machine learning models or shallow ANN models, deep models are often used to run directly on raw EEG signal data without the feature extraction phase, avoiding the time consumption for this processing and the possible loss of information [254].

A robust technique to validate a classification model using a dataset

composed of EEG signals is the Leave-One-Subject-Out Cross-Validation (LOSO CV). This is a variant of the k-fold cross-validation approach but with folds consisting of subjects. In LOSO CV, one subject is reserved for the evaluation and the model is trained on remaining subjects. The process is repeated each time with a different subject reserved for the evaluation and results are averaged over all folds (subjects). The experiments demonstrated the importance of using LOSO CV for estimating the performance of an ML model for new users and the risks of accuracy overestimates with traditional k-fold cross-validation [91]. This validation is applied in an inter-subjective context where an attempt is made to generalise to new subjects. In contrast, in an intra-subjective context, an attempt is made to create a specific model for each subject. In this case, a validation technique is the Leave-One-Session-Out, in which an attempt is made to generalise on the subject's sessions.

Part I

Brain-Computer Interface

Chapter 4

BCI: an overview

Introduction

In this chapter an overview of the brain-computer interfaces will be made. After a general description, some application contexts and open problems will be analysed.

4.1 General description

The use of brain signals to control prosthetic limbs was first developed in the early 1970s [184]. Since then, a new area of research has emerged that has been named Brain-Computer Interface (BCI). This research endeavours to improve the interpretation of brain waveforms in order to establish increasingly accurate control towards external devices. Advances made in recent years in both computer science, particularly in machine learning, and biological brain science have made BCI a very influential area of research in the applied sciences.

In general, a BCI system records brain activity and converts it into commands for external devices. A BCI system usually consists of the following key elements: signal acquisition, signal preprocessing, feature extraction, classification and application interface [190] (fig. 4.1). The acquired signals are sent to the preprocessing component for signal enhancement. Next, features are extracted from the preprocessed signals and sent to the classifier, which recognises the signals and converts them

into commands for external devices.

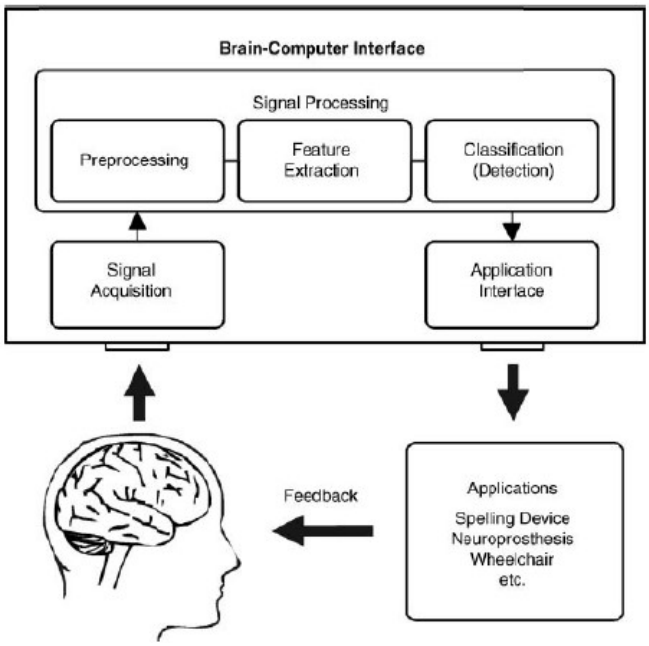


Figure 4.1. Components of a BCI system: signals from the user's brain are acquired and processed to extract specific features used for classification. The classifier output is transformed into a device command, which, at the same time, provides feedback to the user [190].

4.1.1 Signal acquisition

Brain waves are recorded with several methods that can be grouped mainly into: invasive and non-invasive techniques. For instance, Electro-CorticoGraphy (ECoG) and Electroencephalography (EEG) are the two most widely used invasive and non-invasive technologies [183]. In invasive methods, signals are collected via electrodes placed under the scalp (fig. 4.2). Invasive methods allow a high quality of brain signals as the electrodes are placed close to the neurons. In addition, they have a high spatial and temporal resolution and a high Signal-to-Noise Ratio (SNR), so they diminish artefacts such as eye blinking or movements. On the other

hand, in non-invasive techniques, signals are recorded by external sensors using electrical, metabolic or magnetic methods. Some non-invasive methods are: Electroencephalography (EEG), functional Near-InfraRed Spectroscopy (fNIRS), functional Magnetic Resonance Imaging (fMRI), ElectroOculoGraphy (EOG), and MagnetoEncephaloGraphy (MEG). EEG signals are the most widely used in BCI systems. For more information on the EEG signal, refer to chapter 3.

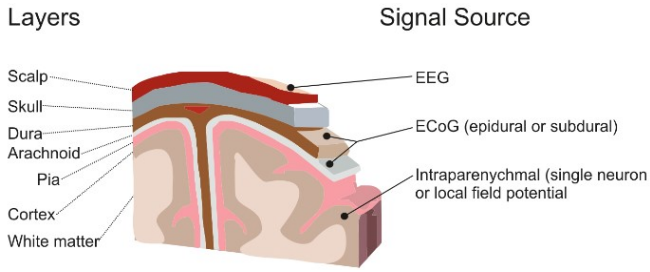


Figure 4.2. Drawing depicting the signals for BCI and their locations relative to the brain. Three general categories of signals are used for BCI applications [145].

4.2 Application contexts

BCI systems are applied in many scenarios including health care, rehabilitation, security, smart environment, but also games & entertainment.

4.2.1 Health care

In health care, BCI systems mainly work on the detection and diagnosis of mental diseases such as sleep disorders, Alzheimer's disease, epileptic seizures and other disorders. With regard to the detection of sleep disorders, most studies focus on the detection of the different sleep phases on the basis of spontaneous EEG. BCI technology can also be used for the early detection of abnormal brain structures and functions, examples of space-occupying lesions (e.g. brain cancer, encephalitis) and abnormal neuronal discharges (seizures). One study proposed a BCI system that identified abnormalities detected in the EEG due to tumours and seizures with an

accuracy of 98%, 93% and 87% for normality, epilepsy and brain tumour, respectively [151]. Early diagnosis of seizure disorders and their control using ANN have been proposed in [138]. Furthermore, BCI technology is useful for diagnosing dyslexia [85], attention deficit hyperactivity disorder (ADHD) [154] and a training program using BCI has been implemented to improve ADHD symptoms [153]. kumar2014epileptic, 9224368

4.2.2 Rehabilitation

One important aspect in the medical field is rehabilitation, such as the recovery of damaged motor functions or the ability to communicate and more generally the improvement of quality of life. Neurorehabilitation could be improved by using BCI systems for people suffering from motor, communication and control problems due to neurological damage. Numerous studies have suggested that neuroprosthetic devices, which utilise motor imagery-based BCI technology, could be useful in restoring normal levels of function to the hand of stroke patients who have not been able to recover previous levels of movement [56].

Furthermore, in therapeutic rehabilitation sessions, it may be very useful to know the engagement states of patients.

Engagement assessment

In rehabilitation context biosignals-based measurement methods are emerging. They allow an automated and real-time engagement assessment. In particular, eye-blinking [195], heart rate variability [90], and brain activity [107, 83] were used to detect changes in patient's engagement. Among these, the EEG signal [47] offers good temporal resolution and improves real-time performances.

Studies on EEG-based engagement detection were mainly conducted on adults and focused only on cognitive engagement. In [137], a computational framework was proposed for real-time cognitive engagement recognition using EEG. A deep Convolutional Neural Network was used to extract task discriminative spatio-temporal features and predict the CE level for two classes: engaged vs. disengaged. Experiments were conducted on 8 subjects performing the Go/No-Go paradigm to induce cognitive fatigue. An average inter-subjective accuracy of 88.13% was reached. In [187], the

EEG signals were acquired for monitoring cognitive engagement in stroke patients while they executed active and passive motor tasks. Event-related desynchronization differences between tasks were observed during both initial and post-movement periods. EEG data were used to classify each epoch as involving the active or passive motor task. Average classification accuracy was 80.7 ± 0.1% for grasping movement and 82.8 ± 0.1% for supination movement.

Recently, a first study on engagement in pediatric rehabilitation was proposed [72]. Positive/negative engagement of autistic patients was classified starting from EEG signals and gesture recognition. The EEG signals were acquired through the single-channel MindWave; Kinect was instead employed for gesture recognition. Five children (two with autism) undertook the experiment. An inter-subjective accuracy of 95.8% was achieved in classifying positive or negative engagement. However, the study does not specify the explored engagement dimensions (i.e. emotional, cognitive, or behavioral). To date, to the best of my knowledge, only one study is present in the literature on this topic. The reasons could be: (i) the engagement measure in the rehabilitation field has only recently become an object of interest [187], and (ii) EEG-based engagement assessment in pediatric rehabilitation requires the adoption of a respectful clinical protocol to protect the child and his psycho-physical integrity (i. e. a non-interventional observational approach). Although such a protocol is more comfortable for the children, it entails a general lack of control over the engagement levels resulting in imbalanced data collections.

BCI in virtual reality

Several virtual and augmented reality-based BCI approaches have been proposed for rehabilitation programs. A pilot study project suggests that virtual reality-based BCI is effective in stroke rehabilitation [160]. Preliminary results of a clinical study showed that the use of virtual reality and BCI with the functional electrical stimulator have a good degree of satisfaction, rapid adaptation to therapy and fast progress in user rehabilitation [200]. Augmented reality, on the other hand, gives rise to a new rehabilitation approach. The results of a review, which evaluated the effectiveness of augmented reality in shoulder rehabilitation, showed that augmented reality has more advantages than the traditional program [234].

4.2.3 Security

Security is a common area of interest for BCI researchers. The security problem can be divided in two ways: identification (or recognition) and authentication (or verification). The first is a multi-class classification problem to recognise the identity of the person [253]. The second is a binary classification problem to determine whether the person under consideration is authorised or not [252].

Existing biometric identification/authentication systems are mainly based on the intrinsic physiological characteristics of the individual (e.g. face, iris, retina, voice and fingerprint). However, such person identification systems are vulnerable: e.g. prosthetic masks can interfere with face recognition, contact lenses can fool iris recognition, vocoders can compromise voice identification and fingerprint films can trick fingerprint sensors. Considering this, EEG-based biometric identification systems are emerging as promising alternatives due to their high resistance to attack. An individual's EEG signals are virtually impossible for an impostor to imitate, thus making this approach less vulnerable. Koike et al. [132] adopted deep neural networks to identify the user based on VEP signals, while Mao et al. [164] applied CNNs for person identification based on RSVP signals. Instead, the authors in [252] combined EEG signals with gait information to introduce a dual authentication system with a hybrid deep learning model.

4.2.4 Smart Environment

The intelligent environment is a promising application scenario for BCI systems in the future. With the development of the Internet of Things (IoT), an increasing number of intelligent environments can be connected to BCI. For example, an assistance robot can be used in smart homes [255], where the robot can be controlled by brain signals.

The intelligent transport sector has also benefited from the BCI function of cognitive state monitoring. Driver behaviour has been studied in numerous studies. The use of EEG signals for fatigue detection was analysed in [236], while [51] discussed the use of the workload index to assess the driver's mental state. In [241], different models for distinguishing distracted drivers were examined.

4.2.5 Games & entertainment

Entertainment and gaming applications have opened up the market for non-medical BCIs. Various games have been presented in the literature, such as in [205], in which helicopters are flown in a virtual world. In [50], players can participate in a football match by means of two BCI systems. They can score goals by imagining left or right hand movements. On the other hand, some serious games via BCI have been used for emotional control and/or neuroprosthetic rehabilitation. In [227], Tan and Nijholt described the game Brainball, a game that aims to reduce stress levels. Users can only move the ball by relaxing, thus, the calmer player is more likely to win and thus learn to control stress while having fun.

4.3 Open Problems

The main open problems in BCI systems are due to electrophysiological characteristics of brain signals, the data collection and/or calibration phase, the classifier training step and evaluation methods.

4.3.1 Intrinsic signal properties

BCI systems that acquire signals by invasive methods have certain problems. Firstly, the implantation of electrodes requires a surgical procedure, which is expensive and risky due to potential medical complications such as transplant rejection. Secondly, the implanted electrodes are fixed and thus can only measure brain signals from the same locations. For these reasons, invasive BCI systems are mainly used in animals and people with severe disabilities (e.g. ALS patients) [4], and therefore non-invasive techniques are preferred.

Beyond the modalities of signal acquisition (invasive or non-invasive), there are problems related to the electrophysiological properties of recorded brain signals, analysed mainly on EEG signals (as seen in section 3.1.3).

4.3.2 Training

One of the initial processes in the implementation of a BCI system is data collection. This phase for the user undergoing acquisition is a time-

consuming activity considering the number of sessions required. Some BCI paradigms can also be fatiguing for the user as the usage time increases as seen in [213]. Furthermore, in subject-dependent BCI systems, a calibration is required before each session, increasing the user's time of use even more. For this reason, attempts have been made in recent years to develop BCI models that are able to generalise to new subjects while avoiding the calibration phase [235].

The training sets for learning classifiers in BCI systems are usually small, also considering the usability problem in signal acquisition as discussed above. On the other hand, large training sets require higher learning times for classifiers. It would be necessary to balance the amount of training data required with the technological complexity in interpreting the user's brain signals [10].

4.3.3 Evaluation methods

Classifiers of BCI systems should be evaluated online, as each BCI implementation takes place in an online situation. Furthermore, they should be validated to ensure that they have low complexity and can be calibrated quickly in real time. Domain adaptation and transfer learning could be an acceptable solution for the development of BCIs without calibration.
Various performance evaluation measures are used to evaluate BCI systems. However, when different evaluation metrics are used to evaluate BCI systems, it is almost impossible to compare them. Therefore, the BCI research community should establish a uniform and systematic approach to quantify a particular BCI application or metric [197].

Chapter 5

Problem 1: Engagement Detection

Introduction

Engagement assessment is fundamental in clinical practice to personalize treatments and improve their effectiveness. Indeed, patients involved in healthcare decision-making tend to perform better and to be healthier.

The standard tools used in clinical practice for engagement assessment are questionnaires or rating scales. Both take into account the patients' awareness of their health and their therapeutic process. In adult rehabilitation, the most used are: Patient Activation Measure (PAM-13) [109] and Patient Health Engagement (PHE) scale [96]. Recently, also in pediatric rehabilitation, engagement assessment scales have been developed and validated. The Pediatric Rehabilitation Intervention Measure of Engagement-Observation (PRIME-O) version [127] and the Pediatric Assessment of Rehabilitation Engagement (PARE) scale [74] were designed to capture signs of emotional, cognitive, and behavioral engagement for clients and service providers and in the client-provider interaction.

In this chapter, a module for cognitive and emotional engagement assessment, designed to be integrated into an automated [12] or semi-automated [95] rehabilitation system, is presented.

The combined assessment of cognitive and emotional engagement could lead a better adaptability of the therapy and an improvement in its effec-

tiveness [127].

In section 5.1, the multidimensional nature of engagement is presented. In section 5.2, the basic ideas, the architecture, and the data analysis of the proposed method are highlighted. Then, in section 5.3, the experimental validation is reported, by detailing the experimental setup, and the results are discussed.

5.1 Background

The term engagement, derived from the verb "engager", is often used as a synonym for commitment and/or involvement.

Several definitions have been provided over the years because of its multi-dimensional and heterogeneous nature. In 1990, Kahn based the definition of engagement on three broad dimensions: behavioural, cognitive, and emotional [120]. Behavioral engagement is the set of observable indicators (postures, gestures, actions, etc.) of persistence and participation. Cognitive engagement is the effort to extend one's intellectual commitment beyond the minimum required to complete the task. Finally, emotional engagement is the positive emotional reactions of individuals to a task.

In rehabilitation, Barello et al. [44], defined patient engagement as a "multi-dimensional psycho-social process, resulting from the conjoint cognitive, emotional, and behavioral enactment of individuals toward their health condition and management" . The cognitive dimension refers to the meaning given by the patient to the disease, its treatments, its possible developments, and its monitoring. The emotional dimension consists of the emotive reactions of patients in adapting to the onset of the disease and the new living conditions connected to it. The behavioral dimension is connected to all the activities the patient acts out to face the disease and the treatments.

Lequerica et al. defined engagement in rehabilitation as "a deliberate effort and commitment to working toward the goals of rehabilitation interventions, typically demonstrated through active, effortful participation in therapies and cooperation with treatment providers" [144]. Moreover, the authors highlighted the role of motivation in triggering and feeding engagement. Motivation can be intrinsic or extrinsic. Deci and Ryan [73] defined

intrinsically motivated behaviours as those "for which the rewards are internal to the person". Conversely, extrinsically motivated behaviours are performed to obtain external reward such as money or praise. According to the authors, intrinsic goals are more powerful motivators than extrinsic or externally imposed goals. Intrinsic motivational factors influencing therapeutic engagement are: (i) perception of the need for treatment; (ii) perception of the probability of a positive outcome; (iii) perception of self-efficacy in completing tasks, and (iv) re-evaluation of beliefs, attitudes and expectations [144].

In pediatric rehabilitation, it is difficult to achieve engagement by relying only on intrinsic motivation. Therefore, the extrinsic motivation is required. Children only react to what is real, concrete, present and immediately satisfying.

5.2 Methods

5.2.1 Basic Ideas

The aim of this study is to propose an EEG-based engagement detection system in the field of pediatric rehabilitation. The basic ideas of the proposed method are:

- *The use of both the emotional and the cognitive engagement*: an overcoming of the reductionist approach based only on the cognitive dimension, which is particularly unsuitable for children [127], is proposed.
- *Adoption of a subject-dependent approach*: the low inter-individual EEG reproducibility significantly influences the pattern classification in the engagement detection systems [161].
- *Support procedure for user calibration*: the system needs a calibration. To this aim, the user executes a set of rehabilitation sessions on different days. An observational non-interventional protocol is the best choice for maximizing children's comfort. However, this can lead to unbalanced data and a more challenging classifier training phase is required. The recent KMeansSMOTE method [80] is proposed to manage the imbalance of data.

5.2.2 Architecture

The proposed method is sketched in Fig. 5.1. The *semi-wet 14 channel EEG device* allows the EEG signals to be sensed directly from the scalp of the child. Channels are referred to CMS/DRL. Analog signals are conditioned by stages of amplification and filtering (*Analog Filter and Amplifier*). Then, they are digitized by the Analog Digital Converter *ADC* and sent by the *Wireless Transmission Unit* to the *Data Processing* block. The *Classifiers* receive the feature arrays from two trained *Common Spatial Pattern* procedures for detecting the cognitive and emotional engagement.

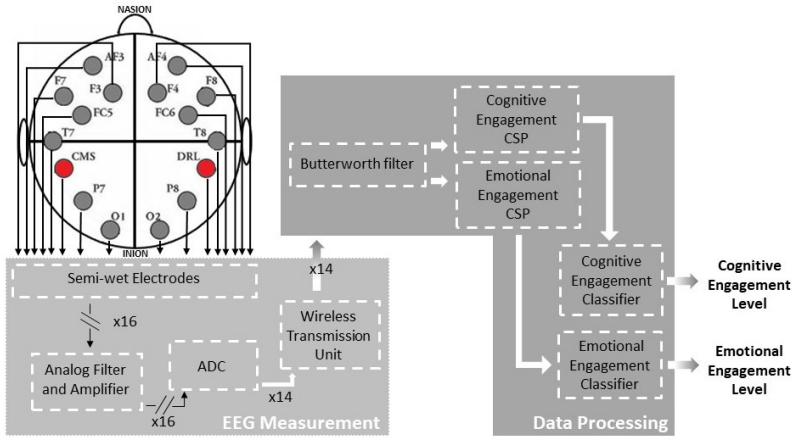


Figure 5.1. The proposed cognitive and emotional engagement detection method.

5.2.3 Data processing

In this section, *data preparation, training* and *classification* are presented.

1. *Data preparation and training*: the EEG tracks are acquired at a sample rate of 128 Sa/s into time windows of 9 s without overlap. EEG signals are filtered through a 4th order Butterworth band-pass filter, between 0.5 Hz and 45 Hz. During the calibration, data are collected and properly labeled by the therapist. Both cognitive and

emotional engagements are distinguished in two classes, high and low. Two Common Spatial Pattern procedures (CSP [133]) and two fully-connected feed-forward artificial neural network (ANN) classifiers, are separately trained on cognitive and emotional engagement data.

2. *Classification*: the trained CSPs project multi-channel EEG data belonging to different classes into a new space, where the differences between the variances along the dimensions are maximized. The two trained ANNs for emotional and cognitive engagement classification are fed with the outputs of the previous stage (Fig. 5.1).

5.3 Experimental Setup

5.3.1 Sample

Four children, three males and one female aged between 5 and 7 years, suffering from disturbances in motor-visual coordination, were selected for the experiment. Each subject was affected at least by one among the following diseases: double hemiplegia, motor skills deficit with dyspraxia, neuropsychomotricity delay, and severe neuropsychomotricity delay in spastic expression from perinatal suffering. Their main symptoms were: lack of strength, motor awkwardness, difficulty in maintaining balance, inadequate postures, spatial disorientation, problems with laterality (right, left confusion), difficulty in managing time, and learning difficulties.

5.3.2 Experimental setup

The experimental protocol was approved by the ethical committee of the University Federico II. Families agreed to the experimental activities by releasing a written informed consent before the experiment. Procedures were carried out according to relevant guidelines and regulations [2]. An observational non-interventional protocol maximized the children's comfort. Therefore, part of the ordinary rehabilitation sessions was monitored by EEG for a total of about thirty minutes per week for each subject. The data acquisition took place in a room illuminated by natural light and provided with air exchange.

The adopted therapeutic approach was the Perfetti-Puccini method, also known as Cognitive Therapeutic Exercise [57]. The method aims to recover the injury and activate the brain circuits that govern movement. The child was asked to perform a visual attention exercise while keeping the correct posture of the trunk, neck, and head. An interactive environment [52] was depicted on a screen placed at the eye level of the subject (Fig. 5.2). One of four characters (a bee, a ladybug, a girl, or a little fish) could be chosen to make the game more interesting. The child had to stare at the character on the screen to make it move while maintaining eye contact. Dynamic tracking techniques were employed. The game allowed to set (i) the direction of the character's movement (from right to left and vice versa, or from top to bottom and vice versa), and (ii) the background landscape, to adapt the difficulty level to the patient's needs. A background music was inserted into the game to improve the child engagement. The game provided some features to adapt the therapy to the state of the subject: (i) a simplification of the exercise, (ii) the introduction of elements of novelty, and (iii) a content change.

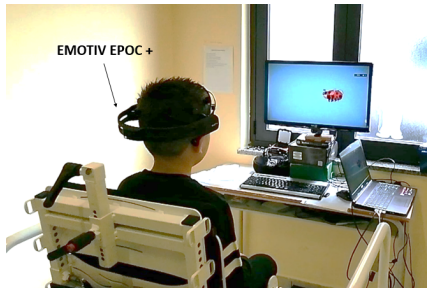


Figure 5.2. Neuromotor rehabilitation session.

The experimentation was conducted with the help of professional figures who contributed to the trial activity. Physiotherapists explained the exercise to the child (before the first session only), supervised rehabilitation, and helped the child maintaining eye contact and correct posture. A software engineer was responsible for starting the system and saving the data. A biomedical engineer was responsible for the EEG signal acquisition system and, therefore, for the correct setting-up, placement of the

device, and electrode-skin quality contact.

5.3.3 Experimental reference

Each session was video-recorded by two cameras (front and side framing).

The Pediatric Assessment of Rehabilitation Engagement (PARE) scale was employed for labeling the EEG signals. The emotional, cognitive, and behavioral components of engagement were expressed in terms of: participation, attention, activation, understanding, positive reactions, interest and enthusiasm, posture and movements of the child during the exercise, on a scale from 0 to 4. The PARE scale allowed to assess the rehabilitation session as a whole. The items of the scale were rearranged to be employed in shorter time intervals with the aim of improving the temporal resolution of observations. The behavioral component of engagement cannot be assessed starting from the EEG signal. Therefore, only the cognitive and emotional components of engagement are considered for research purposes. The items referring to the emotional and cognitive spheres were separately grouped. The evaluations were made by a multidisciplinary team while viewing the videos. The evaluators were asked to rate both the components of the engagement on two levels: high/low emotional engagement and high/low cognitive engagement. They also noted the status changes of the emotional and cognitive engagement and the correspondent time instants of occurrence. The consensus among the evaluators was statistically analyzed. The results revealed a total consensus of 95.2 % [135]. Evaluations were used as ground-truth to label the EEG dataset.

5.3.4 Hardware

EEG data were acquired through the portable, high resolution, 14-channel *Emotiv Epoc+*. See section 3.1.5 for more details.

5.3.5 Experimental validation

Each subject underwent five EEG recording sessions of 15 min. At the end of each session, the assessment described in Section 5.3.3 was carried out by the therapists. Only on the first day, an initial training phase was

implemented. In Fig. 5.3, the experimental paradigm as whole performed by the subjects is shown. A mean of 280 ± 46 epochs for subject was acquired for a total of 1121 epochs. Each epoch lasted 9 s. The different number of epochs was due to a less constrained experimental protocol, adopted to ensure a greater comfort for the patients.

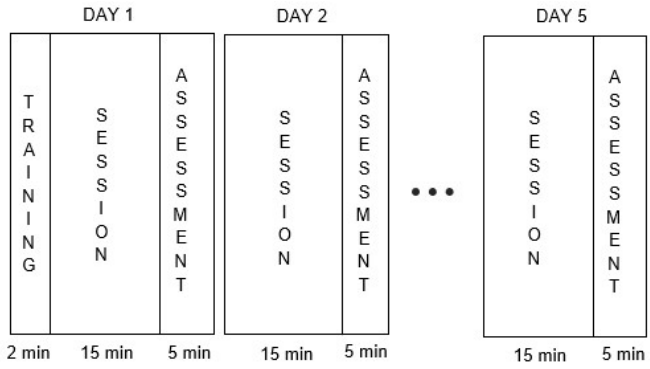


Figure 5.3. The experimental paradigm: only on the first day, a training phase is implemented.

Preprocessing

The EEG signals were filtered through a 4th order Butterworth band-pass filter between 0.5 Hz and 45 Hz. The full frequency spectrum was already investigated in the EEG data analysis literature, as for example in [9]. Next, the CSP (see section 3.3) was adopted. The optimal number of CSP components was found through a grid search Cross-Validation (CV) procedure, varying the number of components in the range $[4, 14]$. Due to the different number of data for each class, data were oversampled by five oversampling methods: SMOTE [59], BorderlineSMOTE [106], ADASYN [103], SVMSMOTE [181] and KMeansSMOTE [80]. See section 2.4.1 for more information on these techniques.

The impact of synthetic data was assessed by comparing the resulting performances with and without oversampling. In Table 5.1 the hyperparameters values for each oversampling method are reported.

Table 5.1. Oversampling methods, optimized Hyperparameters, and variation ranges.

Oversampling Method	Optimized Hyperparameter	Variation Range
SMOTE	K Nearest Neighbours	{5, 7, 11}
ADASYN	K Nearest Neighbours	{5, 7, 11}
SVMSMOTE	K Nearest Neighbours	{5, 7, 11}
BorderlineSMOTE	K Nearest Neighbours	{5, 7, 11}
KMeansSMOTE	k-Means Estimator	{1, 2, 4}
	Cluster Balance Threshold	{0.1, 0.9}
	K Nearest Neighbours	{2, 4, 8}

Table 5.2. Classifiers, optimized Hyperparameters, and variation ranges.

Classifier	Optimized Hyperparameter	Variation Range
k-Nearest Neighbour (k-NN)	Distance	{minkowski, chebychev, manhattan, cosine, euclidean}
	Distance Weight	{equal, inverse, squaredinverse}
	Num Neighbors	[1, 7] step: 1
Support Vector Machine (SVM)	C Regularization	{0.1, 1, 5}
	Kernel Function	{radial basis, polynomial}
	Polynomial Order	{1, 2, 3}
Artificial Neural Network (ANN)	Activation Function	{relu, tanh}
	Hidden Layer nr. of Neurons	[5, 505] step: 20
	Learning Rate	{0.0005, 0.0001, 0.001, 0.005, 0.01}

Classification

Three machine learning classifiers were used: k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Artificial Neural Network (ANN). For each classifier, the best model was found through a grid search CV procedure. Specifically, the regularization term C, the kernel type (polynomial or radial basis function) and the degree of the polynomial kernel function (in case of polynomial kernel) were found for the SVM. ANN classifiers were trained with the Adam learning algorithm [128], while the number of neurons, the number of hidden layers, and the learning rate were found using the CV procedure. For k-NN, the best number of neighbors and the distance function were the hyperparameters considered during the CV procedure. In Table 6.1, for each classifier model, hyperparameters values are reported.

For each subject, the model was trained using the first 70 % of the data of each session as a training set and the remaining 30 % as a test set, for the evaluation phase. The data were processed in the same temporal order of acquisition, so that the test set contains only data temporally

subsequent to the training data. Such a subdivision into training and test sets for intra-individual classification is widely used in literature for EEG data [148, 224, 264].

Evaluation metrics

Data were classified according to cognitive and emotional engagement. The oversampling procedure helped in the management of the imbalanced training data. However, the problem still remained in the test data. To have a fair evaluation of the proposed system, the Matthew Correlation coefficient (MCC [42]) and the Balanced Accuracy (BA [54]) were used as performance scores (see section 2.4.1).

5.4 Experimental Results

In Tables 5.3 and 5.4, the overall averages of the intra-individual balanced accuracies and MCC scores, given by the adopted classifiers, are reported for the cognitive engagement and the emotional engagement, respectively. To better understand to what extend the oversampling strategy can affect the results, the experiments were repeated with or without the application of the oversampling method.

As regards cognitive engagement, the oversampling method gave a slight improvement; as regards emotional engagement, the oversampling method gave a significant improvement to the performances, especially when the KMeansSMOTE method was employed.

The KmeansSMOTE is less likely to generate minority class data in domain areas predominantly dominated by majority class data. Thus, generated data are closer to the data of the minority class, as showed in Fig. 5.4 where the training data of a highly-unbalanced subject are shown using the t-SNE projection [231]. The data is oversampled with two methods: SMOTE (Fig. 5.4 A) and KMeansSMOTE (Fig. 5.4 B). The latter attenuates the noise thanks to clustering before data interpolation.

Figures 5.5 and 5.6 show the intra-subjective balanced accuracies obtained both on cognitive and emotional engagement, respectively, using the KMeansSMOTE oversampling method. The ANN classifiers returned the better scores in most subjects, both in the emotional and cognitive

Table 5.3. Overall mean of the intra-individual performances on cognitive engagement using three different classifiers: the balanced accuracy (BA) and the Matthews correlation coefficient (MCC) at varying the oversampling methods.

Oversampling	Metric	k-NN	SVM	ANN	Mean
none	BA	67.1	67.4	73.7	69.4 ± 3.0
	MCC	0.31	0.34	0.45	0.36 ± 0.06
SMOTE	BA	68.6	69.8	72.0	70.1 ± 1.4
	MCC	0.33	0.36	0.40	0.36 ± 0.03
BorderlineSMOTE	BA	70.3	70.9	73.6	71.6 ± 1.4
	MCC	0.36	0.38	0.43	0.39 ± 0.03
ADASYN	BA	68.1	68.3	72.5	69.6 ± 2.0
	MCC	0.33	0.33	0.42	0.36 ± 0.04
SVMSMOTE	BA	69.0	69.4	72.9	70.4 ± 1.7
	MCC	0.34	0.36	0.42	0.37 ± 0.03
KMeansSMOTE	BA	69.8	71.1	74.5	$\mathbf{71.8 \pm 1.98}$
	MCC	0.35	0.39	0.46	$\mathbf{0.39 \pm 0.04}$

engagement.

Furthermore, the MCC and the BA values ensure that the results are not affected by unbalancing bias in the test phase.

5.5 Discussion

Due to the covid-19 pandemic, only four children were selected for the experiment. In this situation, an evaluation of inter-subjective models is not statistically significant, so it was decided to develop and evaluate intra-subjective models with the aim that the results obtained in this experimental phase may also be useful in the development of inter-subjective models when more data become available.

The results reported in Tabs. 5.3 and 5.4 showed the improvement given by the oversampling methods in the proposed setup. More in detail, in the emotional engagement classification task, the improvements are more significant (e.g., an increase in accuracy of about 10 %) with respect to the cognitive engagement classification performance. This can be due to the different unbalancing ratios between the classes in the two tasks (i.e., a greater unbalanced data condition in the emotional engage-

Table 5.4. Overall mean of the intra-individual performances on emotional engagement using three different classifiers: the balanced accuracy (BA) and the Matthews correlation coefficient (MCC) at varying the oversampling methods.

Oversampling	Metric	k-NN	SVM	ANN	Mean
none	BA	56.3	57.0	61.4	58.2 ± 2.2
	MCC	0.16	0.20	0.26	0.21 ± 0.04
SMOTE	BA	57.6	61.2	67.1	62 ± 3.9
	MCC	0.16	0.24	0.35	0.25 ± 0.08
BorderlineSMOTE	BA	57.3	60.2	66.5	61.3 ± 3.8
	MCC	0.15	0.22	0.34	0.24 ± 0.08
ADASYN	BA	57.0	60.0	67.4	61.5 ± 4.4
	MCC	0.15	0.21	0.36	0.24 ± 0.09
SVMSMOTE	BA	57.3	61.0	64.4	60.9 ± 2.9
	MCC	0.15	0.25	0.31	0.24 ± 0.06
KMeansSMOTE	BA	57.9	63.6	71.2	$\mathbf{64.23 \pm 5.4}$
	MCC	0.18	0.30	0.43	$\mathbf{0.30 \pm 0.10}$

ment dataset with respect to the cognitive one). Indeed, in the proposed setup, the SMOTE algorithms generated greater amounts of data in case of strong unbalanced data condition having a greater impact on the classification performances. Therefore, also the cognitive dataset was artificially unbalanced to validate this hypothesis. To this aim, the number of samples was chosen so that the classes distribution was the same as the emotional engagement data. Next, an ANN classification step with and without KMeansSmote was carried out. The resulting performances without any oversampling strategy were 58.73 % and 0.25 for BA and MCC, respectively. Instead, BA and MCC increased to 65.14 % and 0.28, respectively, with KMeansSmote oversampling. The improvement given by KMeansSmote showed that the used oversampling strategy is particularly suitable for this type of data in case of imbalanced condition.

As concerns the data acquisition stage, Emotiv Epoch+ is only partially adaptable to different head sizes. Nevertheless, among the children involved in the experimental activity, the child with the smallest head exhibited an inion-naison distance of 31.0 cm that is within the range of variation in adults of [31,0 - 38,0] cm, well established in literature [177]. By assuming that the manufacturer optimized the product for an average

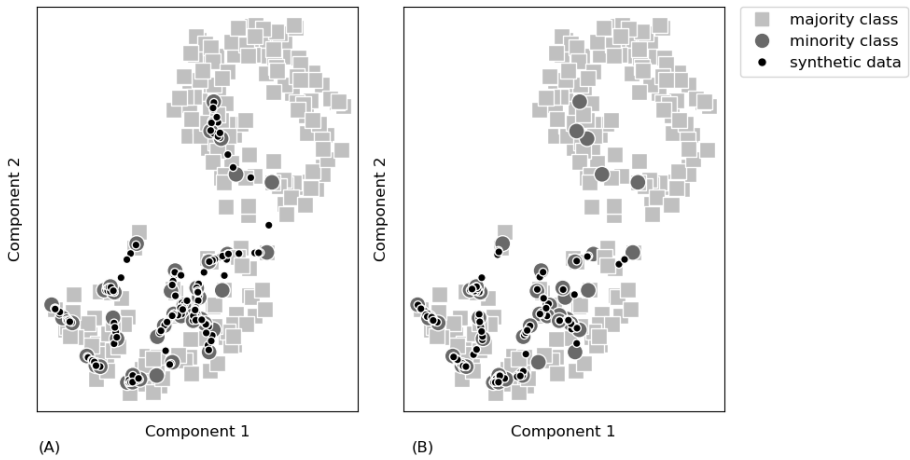


Figure 5.4. t-SNE projection of unbalanced EEG data (subject 4) oversampled with two different methods. The SMOTE method (A) randomically interpolates the data of the minority class. The KMeansSMOTE method (B) realizes a clustering before interpolation, attenuating the noise.

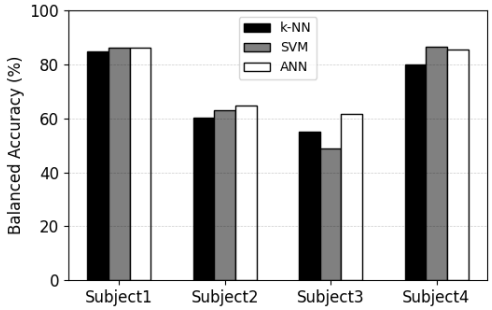


Figure 5.5. Cognitive engagement balanced accuracies for each subject based on KMeansSMOTE oversampling technique. Classifier performances are reported.

value of the inion-nasion distance of 34.5 cm in adults, in the case of a lower inion-nasion distance of 31.0 cm, the maximum electrode dislocation is about 1.4 cm with respect to the 10-20 International Positioning System. The maximum electrode position shift is appreciated in the frontal area

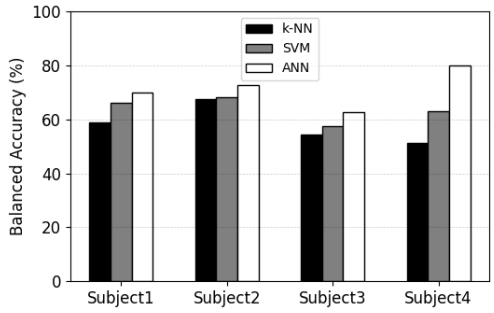


Figure 5.6. Emotional engagement balanced accuracies for each subject based on KMeansSMOTE oversampling technique. Classifier performances are reported.

and it gradually decreases until its disappearance, moving from the frontal area to the occipital area of the scalp. Therefore, the distance of each electrode from the reference of the 10-20 International Positioning System is to be considered in order to make reproducible the measurement. Despite the Emotiv Epoc+ device has the largest number of electrodes among the low-cost EEG devices available on the market, it does not guarantee a dense coverage of the parietal area of the scalp. The signal acquired in this area is particularly relevant for the assessment of the spatial attention [115, 158]. However, the device is equipped with 2 electrodes in the parietal areas (i.e., P7 and P8) and the spatial attention is only one component of engagement.

Finally, the proposed approach is data driven. Thus, it can be applied flexibly to different targets by identifying *ad-hoc* models suitable for different abled groups.

5.6 Conclusion

In this chapter, a EEG-based engagement (cognitive and emotional) detection system is proposed for pediatric rehabilitation. A subject-dependent approach is adopted and a specific easy calibration is provided for personalized medicine. The proposed method, based on KMeansSMOTE and ANN, showed experimentally a mean balanced accuracy of 71.2 % and 74.5 %

for the emotional and cognitive engagement, respectively. Furthermore, a comparison between several oversampling strategies was made, showing that the KMeansSMOTE can be a promising oversampling method for unbalanced EEG engagement datasets. The KMeansSMOTE method is the core of the proposed calibration procedure, but also a promising technique for researchers focused on the observation of the spontaneous children behavior. The distance of each electrode from the reference of the 10-20 International Positioning System was noted to make the measurement reproducible, being reproducibility a quality parameter of the measurement itself. In future works, new measurement solutions will be tested to guarantee more adaptivity to children's head size and more dense coverage of the parietal area.

Chapter 6

Problem 2: Enhancement of SSVEPs classification

Introduction

Among the major BCI paradigms, Steady-State Visually Evoked Potential (SSVEP) has rapidly gained interest for developing applications in several fields, such as rehabilitation [259, 36], gaming [166], entertainment [244], industrial inspection [14, 150], and health monitoring [35], since it is characterized by easier detection and higher Information Transfer Rates (ITRs) with respect to other available BCI paradigms [5, 257].
In particular, SSVEPs are a specific physiological brain response to continuously flickering visual stimuli, typically inducted after a latency varying from 80 ms to 160 ms [118]. Stimulation frequency bands usually range from 6 Hz to 30 Hz, although the best Signal to Noise Ratio (SNR) is achieved in the range 8-15 Hz [240]. Generally, the SSVEP shows a sinusoidal-like waveform, with a fundamental frequency equal to that of the gazed stimulus, and often higher harmonics [175], as shown in Fig. 3.1.
In practical applications, different visual stimuli (at different frequencies) are associated to specific commands: thus, such systems allow the user to perform a selection by simply looking at the related flickering stimulus.
In traditional SSVEP-based experimental setups, the SSVEPs are acquired through a multi-channel EEG data acquisition device [245], while the flickering stimuli are often visualized on a LCD monitor. However, this bench-

top instrumentation limits the portability of the system. Recently, wearable solutions, based on single-channel acquisitions, have been proposed in the literature [131, 30]. Additionally, the use of Augmented Reality (AR) Head-Mounted Displays (HMDs), which are emerging devices of the 4.0 scenario [34], is establishing itself as a promising strategy to render the flickering stimuli and guaranteeing, at the same time, more immersivity and engagement in the fruition of BCI applications [123, 228, 33].
Nevertheless, the overall performance of combined AR-BCI instruments strongly depends on the specifications of the HMD adopted; in particular, on two characteristics that are. The former, the field of view (FOV) of HMDs is generally limited to some tens of degrees: this limits the maximum number of flickering stimuli that can be rendered simultaneously on the HMD. At the state of the art, good performance has been achieved when, at most, two visual stimuli are simultaneously displayed [36]. The latter, AR HMDs exhibit a significant non-predictability of the frame rate. This uncertainty leads to a shift in the frequency values of the rendered stimuli, reducing the classification accuracy of the SSVEP elicited on the user's EEG [35].
To preserve wearability of SSVEP-based AR-BCI instrumentation, and at the same time ensuring acceptable performance, the challenge is to keep the results obtained using HMDs close to performance achieved through traditional setups [238].

Based on these considerations, in this chapter, an experimental characterization of a highly-wearable, AR-based SSVEP BCI is performed. The aim is twofold: firstly, evaluating the classification performance by comparing the adoption of the aforementioned classifiers (SVM, k-NN, ANN, CNN) with the state-of-the-art Canonical Correlation Analysis (CCA) (see section 3.3.3); to this aim, two framework were designed, implemented and comparatively tested after four different experiments. Each experiment was characterized by the use of a different AR HMD to generate the flickering stimuli. Allowing to compare the impact of different AR technologies in the elicitation of SSVEPs.
Secondly, two particular frameworks: i) custom ANN with a Variable Activation Function (VAF) [28], ii) EEGNet (2.2.1), a well-know model for EEG dataset, were tested on Benchmark data, a public dataset [238], composed by 35 patients subjected to 40 simultaneous flickering stimuli, using

different normalization techniques and exploit Domain Adaptation (see section 2.4.2) methods.
The chapter is organized as follows. Section 6.2 describes the proposal in detail. The experimental characterization is reported and discussed in Section 6.3, while the obtained results are shown in Section 6.4. Finally, in Section 6.5, conclusions are drawn.

6.1 Related works

In my knowledge, algorithms based on the CCA provide the best performance in terms both of classification accuracy and time response [238, 67]. Another promising strategy is the adoption of Machine Learning (ML) techniques [174], in particular: (i) classical ML classifiers, such as Support Vector Machine (SVM), k-Nearest Neighbors (k-NN) [221, 86], and (ii) Artificial and Convolutional Neural Networks (ANN, CNN) [15, 182]. Indeed, recent works [182, 192, 199] showed that, for low-channel EEG setups, these strategies outperform the results obtained through CCA. For example, in [182] a one-dimensional CNN was realized for a single-channel BCI instrument, tackling a five-class classification problem with an accuracy of 99 % at 4-s time response (whereas CCA reached 91 % in the same conditions). A multi-layer CNN, called *PodNet* (see section 2.2.1), was proposed in [192] for a three-channel setup: this CNN exceeded the results obtained by CCA by about 5 % at 2-s time response. Therefore, in a single-channel AR-based instrumentation, the adoption of traditional Machine Learning classifiers and Neural Networks can represent an effective alternative to CCA.

6.2 Proposal

In this chapter, an enhancement of the SSVEP classification performance for highly wearable BCI instrumentation is proposed. To this aim, the architecture of the single-channel BCI developed in [14, 36, 35] was considered. This measurement system is based on the real-time classification of users SSVEPs elicited by AR HMDs. Such instrumentation is particularly challenging for wearable applications as the number of elec-

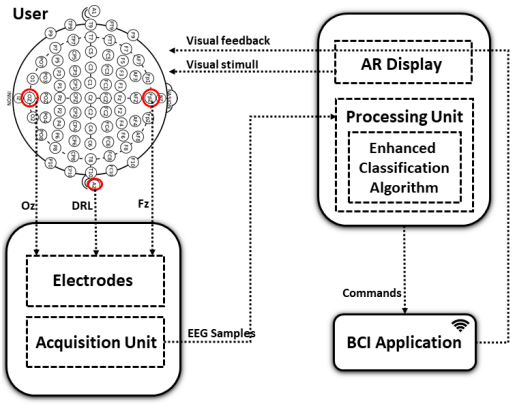


Figure 6.1. Architecture of the wearable BCI-SSVEP system used for testing the proposed alogirithm.

trodes is very limited.
Fig. 6.1 summarizes the major blocks of the system architecture. In particular, an *AR Display* renders the flickering stimuli in the range 8-15 Hz for the SSVEPs elicitation. The brain signal is digitized by a portable *Acquisition Unit*, which sends the EEG Samples to a portable *Processing Unit*. The signal is processed by adopting an *Enhanced Classification Algorithm*, and the detected command is sent in real time to the *BCI Application*, which actuates the received command and also provides a visual feedback to the User to show the output of the selection made.

In this chapter, four strategies are used. The first two tested on four different datasets obtained by using four AR devices:

1. Feature Reduction (FR);
2. Deep SSVEP Convolutional Unit (SCU).

The other two strategies tested on the Benchmark dataset:

1. Artificial Neural Network with Variable Activation Function (ANN VAF);
2. EEGNet.

In the following sections, all the strategies are presented and discussed in detail.

Table 6.1. Classifiers, optimized Hyperparameters, and Variation Ranges

Classifier	Optimized Hyperparameter	Variation Range
k-Nearest Neighbour (k-NN)[1]	Distance	{Minkowski, Chebychev, Manhattan, Cosine, Euclidean}
	Distance Weight	{equal, inverse, squaredinverse}
	Num Neighbors	{3, 5, 6, 7}
Support Vector Machine (SVM)[1]	C Regularization	{0.01, 0.10, 1.00, 1.77, 5.00, 10.00, 15.00}
	Kernel Function	{linear, radial basis, polynomial}
	Polynomial Order	{2, 3, 4}
Artificial Neural Network (ANN)[1]	Activation Function	{relu, tanh}
	Hidden Layer nr. of Neurons	[5, 505] step: 50
	Learning Rate	{0.0005, 0.0001, 0.0010, 0.0050, 0.0100}
	Validation Fraction	{0.2, 0.3}
Deep SSVEP Convolutional Unit[2]	Convolutional Layer nr. of Filters	[16, 1024] step: x2
	SCU Blocks	[1, 7] step: 1
	Kernel Size	{10, 20, 30}
	Dense Layer nr. of Neurons	[60, 1260] step: 200
	Dense Blocks	{1, 2}
	Learning Rate	{0.0001, 0.0010}
	Validation Fraction	{0.2, 0.3}

[1]FR Algorithm
[2]Deep SCU Algorithm

6.2.1 Features Reduction (FR)

The main blocks of the proposed FR scheme, are shown in Fig. 6.2(a). The *EEG Samples* are processed both in frequency and time domains, in order to obtain a reduced number of significant features.

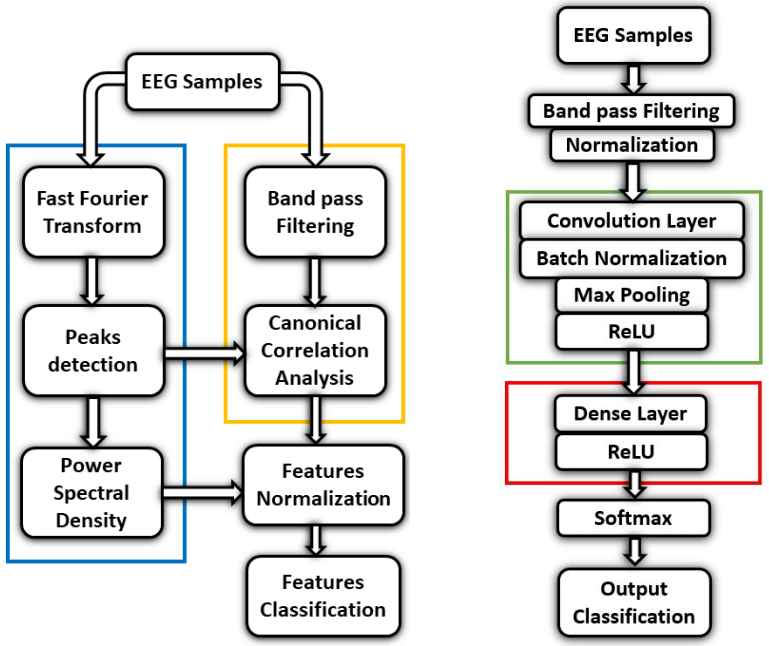
- In the frequency domain, first, a single-sided amplitude spectrum is obtained by means of a Fast Fourier Transform (FFT). No windowing is applied to the original samples. Then, the actual SSVEPs *Peaks* are detected around the n rendered stimulus frequencies: given a generic nominal frequency value f_n, the interval $[f_n \cdot 0.9, f_n \cdot 1.1]$ was used to find the actual peak frequency f_a. This interval was considered suitable in order to properly mitigate the uncertainty introduced by the Frame Per Second (FPS) variations of the AR HMDs in the rendering of the flickering stimuli. Consequently, the resulting Power Spectral Density (PSDs) coefficients [14] are more accurate.

- In the time domain, first, a *Band pass Filtering* between 5 and 25 Hz is applied by means of a Finite Impulsive Response (FIR) filter with linear phase response. Then, the *Canonical Correlation Analysis* between the filtered signal and a set of sinewaves, having the frequencies of the n detected peaks and variable phase [36], is performed. In this way, also the n canonical correlation coefficients obtained for each frequency are more accurate.

Ultimately, for a given brain signal composed of a number $f_s \cdot N$ of EEG samples and n classes (where f_s is the sampling frequency, N is the number of seconds, and n is the number of stimulus frequencies), only $2n$ features are extracted and normalized. Finally, the classification is compared on three ML classifiers: in particular, Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), and Artificial Neural Network (ANN) are employed since they guarantee the best results with acceptable computational effort [221, 86, 15].

6.2.2 Deep SSVEP Convolutional Unit (SCU)

The optimal number of SCU blocks (defined in 2.2.1), dense blocks, and the optimal values of the hyper parameters are found through a grid-search approach (see Table 6.1 for the parameters ranges used in the experiments). In the grid search, the number of SCU blocks varies from a minimum of 1 to a maximum of 7. In each sequential SCU block, convolutional layers with a variable number of filters are considered. For the sake of the example, with 7 SCU blocks the number of filters for each convolutional layer is the following: [16, 32, 64, 128, 256, 512, 1024]. Similarly, for Dense blocks, in which 1 or maximum 2 blocks were different combinations in the number of neurons between the different dense (fully-connected) layers are considered. In Fig. 6.2(b) an example of SCU architecture with one SCU block and one full-connected layer is shown. With respect to the approach proposed in [39], the Deep SCU architecture is now applied to a single-channel setup. Furthermore, the *EEG Samples* are pre-processed by a FIR *Band pass filter* between 5 and 25 Hz with linear phase response, and then normalized.



(a) (b)

Figure 6.2. Block diagram of the Features Reduction (a) and DeepSCU (b) classification algorithms. For the Feature Reduction architecture, the two boxes represent a processing conducted in frequency (blue box) and time (yellow box) domain. For the DeepSCU architecture, the SCU and Dense blocks are highlighted in green and red, respectively.

6.2.3 Artificial Neural Network with Variable Activation Function (ANN VAF)

Taking advantage of the FR processing, instead of a standard ANN model, a variant of it with learnable activation functions, is used as a classifier. Table 6.2 shows the adopted grid search for the tuning of the ANN VAF hyperparameters.

Table 6.2. Optimized hyperparameters and variation ranges for ANN VAF classifier.

Hyperparameter	Range
Fixed Activation Function	{ReLU, Tanh}
Hidden Layer Neurons	[5, 505] step: 50
VAF Layer Neurons	{3, 7, 11}
Learning Rate	{0.0005, 0.0001, 0.0010, 0.0050, 0.0100}
Validation Fraction	{0.2, 0.3}

6.2.4 EEGNet

The EEGNet model, as defined in section 2.2.1, exploits a random search with a maximum of 20 models for the hyperparameters as specified in table 6.3. For the length of the temporal kernel, the value as a function of the segmentation seconds is also taken into account. The 2D convolutional filters F_1 and F_2 utilise the same number of filters F. The other hyperparameters are fixed, such as: learning rate at 0.01 and using Adam [128] as optimizer, batch size at 32, as activation function the Exponential Linear Unit (ELU) [69] function and dropout rate at 0.5. Furthermore, during the learning phase, in order to avoid the problem of over-fit on the training set (refer to section 2.3.2), a stop criterion was used through a patience of 10 epochs.

Table 6.3. Optimized hyperparameters and variation ranges for EEGNet classifier.

Hyperparameter	Range
Temporal Kernel Length	$\{250, 512, \lfloor 250 * seconds \rfloor\}$
F 2D Convolutional Filters	{96, 125, 150}
D Spatial Filters	{1, 2}
Dropout Type	{Dropout, SpatialDropout2D}

Domain Adaptation

For this ML model, some Domain Adaptation techniques were applied on the Benchmark dataset:

1. **Standardisation Schemes.** With the test data available without

labels, it is possible to normalise the data (i) per subject or (ii) per block/session per subject. The normalisation is done by taking into account the EEG channels and standardising time features by removing the mean and scaling to unit variance. Thus, in (i) there will be one mean and one variance for each subject, in (ii) there will be n means and n variances for each subject (where n is the number of blocks). In this way, the non-stationarity of the EEG signal is reduced. For comparison, a canonical z-score normalisation will also be made without the use of unlabelled test set.

2. **Similarity between Subjects.** However, it is important to highlight that the SSVEP signal is not necessarily always clearly detectable in the subjects. In fact, considering a sample of persons subjected to these flashing light stimuli for the first time, what happens is that 'observable' SSVEP occurs in about 90% of them [99]. Thus, it is reasonable to assume that 3 out of the 27 people who were subjected to the experiment for the first time can be discarded in the validation and test set.
To select these subjects, the similarity between the subjects was calculated through Kullback-Leibler (KL) divergence [136]. KL divergence was calculated by taking the EEG channels for subjects and then averaging them:

$$\forall S_a, S_b, chan_i, a, b \in \{1{-}35\}, i \in \{1{-}10\} : mean(KL({S_a}^{chan_i}, {S_b}^{chan_i}))$$

In this way, the distances between the subjects in the dataset was computed. It results that some subjects can be considered as outliers. Three of these, who are also naive on the use of SSVEPs, were chosen to be not present in the validation and test set, so they will be part of the training set in the 10-run of 25-5-5 strategy.

6.3 Experimental Characterization

An experimental characterization of the proposed algorithm was performed by conducting four experiments involving healthy adult volunteers. For each campaign, a different AR HDM was adopted. These devices were used to elicit the users' SSVEPs in the range 8-15 Hz. Four distinct data

sets for the testing of the SSVEPs classification algorithms were provided (one for each HMD).

Successively, the ANN VAF and EEGNet algorithms were tested on the Benchmark dataset.

6.3.1 Hardware and software

The AR devices used in this chapter are listed below:

- *Epson Moverio BT-200*: Moverio BT-200 are AR Smart Glasses with a 60 Hz Refresh Rate and a 23° diagonal FOV. They are equipped with Android 4.0.
- *Epson Moverio BT-350*: Like BT-200 version, Moverio BT-350 have a 23° diagonal FOV; however, the refresh rate is limited to 30 Hz and the operative system on board is Android 5.1.
- *Microsoft Hololens 1*: Microsoft Hololens 1 is an Optical-See-Through (OST) AR HMD with a 60 Hz Refresh Rate and a diagonal FOV of 34°.
- *Oculus Rift S*: Oculus Rift S is a HMD with 80 Hz Refresh Rate. It is originally designed for Virtual Reality. Thus, the integration of a HD Stereoscopic Camera (*Zed Mini*) allows to use the device as a Video-See-Through (VST) AR HMD.

The software employed to realize the AR environment for the selected HMDs are described as follows.

- *Epson Moverio BT-200/350*: the AR applications running on the Moverio glasses was developed in Android Studio. In particular, the flickering squares were generated by means the Android library OpenGL.
- *Microsoft Hololens and Oculus Rift S*: the AR environment for Hololens and Oculus Rift was developed in Unity 3D.

In all these cases, the flickering frequencies were realized with a suitable white/black pixels alternation. For instance, given a refresh rate of 60 Hz, a 10-Hz frequency is generated with a white/black alternation each three

Table 6.4. Details of the datasets.

Data Set Index	#1	#2	#3	#4
AR Device	BT-200	BT-350	Hololens	Rift S
Volunteers	20	9	9	9
Classes	2	4	4	4
Signals/subject	24	20	20	20
Signal length (s)	10	10	10	10

frame [239], while not sub-multiple frequency values are obtained as a rounded average of a variable frequency stimulus [237].
The wearable Acquisition Unit chosen to acquire the users' brain signals is the Olimex EEG-SMT, a 10-bit, 256 S/s, open-source Analog-to-Digital converter. It was preferred to other consumer-grade EEG equipment such as *Emotiv Epoch* or *Neurosky Mindwave* [210, 167, 232] since: (i) a recent experimental characterization confirmed its suitability for BCI applications [31], as it showed strong linearity and no long-term drift; (ii) it has a very low cost (approximately 100 $). Finally, the digitized signal is processed by a Raspberry Pi 4, a portable single-board PC.

6.3.2 Data sets descriptions and validation strategy

AR Devices

Four different data sets were obtained by using each of the considered four AR devices. The algorithms were validated on each data set by means of Leave One Subject Out Cross Validation (LOSO CV). This represents a promising inter-individual validation approach aimed at increasing reproducibility [91]. A grid search for the tuning of the models hyperparameters was adopted. In Table 6.1, the hyperparameters values are reported for each classifier model (for FR and Deep SCU algorithm).
Furthermore, in Table 6.4 the experimental details, regarding the four AR devices, and the number of volunteers, classes, and signals acquired for each subject, are provided. The number of classes indicate the number of simultaneous flickering stimuli rendered by the AR Device. As visible, the processing of the data set #1 is a binary classification problem, since only two frequencies are used. Instead, data sets #2, #3, and #4 are characterized by the adoption of four frequencies. In particular, the frequencies chosen for each data set are listed below:

- Data set #1 (BT-200): [10.00, 12.00] Hz
- Data set #2 (BT-350): [8.00, 10.00, 12.00, 15.00] Hz
- Data set #3 (Hololens): [8.57, 10.00, 12.00, 15.00] Hz
- Data set #4 (Rift S): [8.00, 10.00, 11.43, 13.33] Hz

The rendered stimuli are placed at the edges of the display to avoid interferences. For each trial, each volunteer was asked to focus at the selected stimulus for 10 s.
The performance of the proposed method was assessed both on the accuracy and the related time response: the time response is the signal duration T (also called epoch) extracted for each trial and then classified; on the other hand, the classification accuracy is the percentage of data set correctly classified.

Benchmark

The Benchmark dataset has the following features:

- *Subjects:* 35 healthy subjects (17 females and 18 males, aged 17–34 years, mean age: 22 years), having normal or corrected-to-normal vision, participated in this study. 8 subjects had previous experience in SSVEP-based BCI. Each participant was asked to read and sign an informed consent form before the experiment. This study was approved by the Research Ethics Committee of Tsinghua University.
- *Stimulus Presentation:* An offline BCI experiment using a 40-target BCI speller was designed. The 5×8 stimulus matrix was presented on a 23.6-in LCD monitor (Acer GD245 HQ, response time: 2 ms) with a resolution of 1920×1080 pixels, and a refresh rate of 60 Hz. The viewing distance to the screen was 70 cm. The sizes of stimulus and character were 140×140 and 32×32 pixels square, respectively. The size of the whole matrix area was 1510×1037 pixels. Both the vertical and horizontal distances between two neighboring stimuli were 50 pixels. The stimulus program was developed under MATLAB using the Psychophysics Toolbox Ver. 3. The 40 characters were coded using a joint frequency and phase modulation

(JFPM) approach. In particular, the chosen frequencies were in the range [8.0-15.8] Hz with a 0.2 Hz step, while the phase values had a 0.5 π step. A sampled sinusoidal stimulation method was applied to present visual flickers on the LCD monitor.

- *Data acquisition:* A *Synamps2* EEG acquisition unit (Neuroscan, Inc.) was used to record EEG data (see section 3.1.5). For each subject, the experiment included six blocks. Each block was composed of 40 trials, corresponding to all 40 squares. Each trial started with a 0.5-s target cue. Subjects were asked to shift their gaze to the indicated target as soon as possible. After the cue, all stimuli started to flicker on the screen concurrently for 5 s. Then, the screen became blank for 0.5 s, before the start of the next trial. Subjects were asked to avoid eye blinks during the 5-s stimulation duration. A rest for several minutes between two consecutive blocks was foreseen. The selected channels were PO8, PO7, PO6, PO5, PO4, PO3, POz, O2, O1, and Oz.

The two algorithms were validated on the Benchmark dataset in two ways: i) by means of Leave-One-Subject-Out Cross-Validation (LOSO CV); ii) by means of 10 random run of $25-5-5$ strategy. As reported on [192] this strategy is characterised by: 25 subjects in the training set, 5 in the validation set and 5 in the test set. The difference is that in [192] only one run is performed and it is not indicated which subjects data compose the training, validation and test sets. This is a very important information, since the subjects have very different probability distributions. With a single run of the $25-5-5$ strategy, particular subjects in the validation and test set may result in high accuracy and other subjects in low accuracy on the test set. For this reason, it was decided to use 10 runs of this strategy in order to have at each run randomly the subjects in the three sets, also maintaining the constraint, as reported in [192]: *"The network test and validation datasets this study contain only, novel BCI-naïve subjects to simulate real-world usage."*

In addition, the ANN VAF algorithm will use FR processed data, while EEGNet will use Raw data from the Benchmark dataset, with segmentations of 1.5 and 5 seconds.

6.4 Results

6.4.1 AR Devices

Table 6.5 summarizes the results obtained through the proposed algorithms, compared with those achieved through the CCA used in [36].

Table 6.5. Classification accuracy and corresponding 1-σ reproducibility on the four datasets.

Data set #1 (Moverio BT-200)

T (s)	CCA [36] (%)	Deep SCU (%)	FR* (%)
0.5	70.8 ± 10.0	74.4 ± 9.5	**75.0 ± 9.5**
1.0	74.8 ± 18.1	81.6 ± 9.6	**82.1 ± 9.8**
2.0	84.9 ± 12.1	87.5 ± 8.0	**89.2 ± 7.8**
3.0	91.0 ± 9.4	91.9 ± 7.3	**93.7 ± 5.6**
5.0	95.4 ± 5.6	95.7 ± 4.9	**96.7 ± 3.9**
10.0	-	97.7 ± 4.5	**99.4 ± 2.7**

Data set #2 (Moverio BT-350)

T (s)	CCA [36] (%)	Deep SCU (%)	FR* (%)
0.5	-	30.9 ± 7.1	**39.2 ± 13.5**
1.0	-	35.8 ± 10.4	**46.3 ± 19.2**
2.0	51.9 ± 27.0	42.8 ± 13.2	**53.9 ± 23.5**
3.0	53.3 ± 25.6	43.5 ± 21.1	**56.7 ± 24.9**
5.0	56.7 ± 23.9	41.4 ± 17.9	**57.5 ± 23.7**
10.0	-	47.2 ± 23.0	**62.2 ± 24.5**

Data set #3 (Hololens)

T (s)	CCA [36] (%)	Deep SCU (%)	FR* (%)
0.5	-	**48.4 ± 11.3**	44.9 ± 10.0
1.0	-	56.9 ± 13.9	**66.8 ± 16.7**
2.0	58.9 ± 20.6	72.3 ± 14.4	**76.4 ± 16.9**
3.0	70.5 ± 18.5	77.0 ± 15.8	**82.6 ± 13.1**
5.0	72.9 ± 28.3	80.0 ± 13.8	**88.9 ± 8.6**
10.0	-	75.0 ± 19.3	**94.4 ± 8.3**

Data set #4 (Oculus Rift S)

T (s)	CCA [36] (%)	Deep SCU (%)	FR* (%)
0.5	-	36.7 ± 10.5	**42.7 ± 16.8**
1.0	-	40.6 ± 16.2	**54.0 ± 21.5**
2.0	56.1 ± 24.2	46.4 ± 18.6	**62.3 ± 23.5**
3.0	64.8 ± 20.9	56.3 ± 20.6	**65.7 ± 25.3**
5.0	68.5 ± 23.2	55.3 ± 18.6	**70.6 ± 23.8**
10.0	-	48.9 ± 21.2	**72.2 ± 23.3**

*Only the best result is reported for brevity.

It can be seen that the enhancement reached by the FR algorithm is

significant on each dataset used. The main contribution to this improvement is given by the peak detection block, which allows to obtain more accurate features both in time and frequency domains, thus mitigating the uncertainty caused by unpredictable FPS variation of AR devices.

On the other hand, Deep SCU algorithm outperforms CCA only on data sets #1 and #3. However, in all the data sets, the CCA strategy is characterized by a worse inter-individual 1-σ reproducibility. Thus, the model built by CCA offers lower possibility to be generalized.

With regards to the comparison between the performance of each AR HMD, it is visible that Epson Moverio BT-200 (data set #1) provides the best classification accuracy (almost 90% at 2 s). The main reason is that only two flickering stimuli were rendered simultaneously on the display. When considering the four-stimuli data sets (i.e., data set #2, #3, and #4) the performance are significantly worse. In fact, Microsoft Hololens 1 (data set #3) reaches a classification accuracy of about 76% at 2 s, while Epson Moverio BT-350 (data set #2) and Oculus Rift S (data set #4) achieve about 54% and 62%, respectively. Clearly, the larger field of view of Microsoft Hololens 1 (with respect to Epson Moverio BT-350), and its Optical See-Through technology (with respect to Oculus Rift) contribute to this difference in the outcomes. Overall, it is evident the need of an adequate field of view when the number of concurrent flickering stimuli increases, in order to avoid interferences when users stare at the desired icon.

Table 6.6. FR algorithm results obtained for dataset #1 (BT-200) for each considered model.

T (s)	k-NN (%)	SVM (%)	ANN (%)
0.5	72.8 ± 9.3	74.8 ± 9.6	**75.0 ± 9.5**
1.0	80.7 ± 9.8	82.0 ± 9.8	**82.1 ± 9.8**
2.0	88.3 ± 8.8	**89.2 ± 7.8**	**89.2 ± 7.8**
3.0	93.3 ± 5.9	93.6 ± 5.2	**93.7 ± 5.6**
5.0	96.4 ± 4.8	96.4 ± 4.7	**96.7 ± 3.9**
10.0	99.0 ± 2.9	99.2 ± 2.8	**99.4 ± 2.7**

An overview of the results obtained through the FR algorithm on data set #1 is provided in Fig. 6.4 and Table 6.6. In particular, Fig. 6.4 shows the scatter plots of the features extracted by the FR algorithm. As visible,

even with 1-s epochs, it is possible to discriminate the two classes. Clearly, increasing the duration of the epochs leads to an easier patterns separation and, thus, to an increase of the classification accuracy. Finally, Table 6.6 provides a focus on the obtained results for each model used. The best performance are obtained by ANN classifier; however, even a more simple classifier like k-NN reaches comparable accuracy levels.

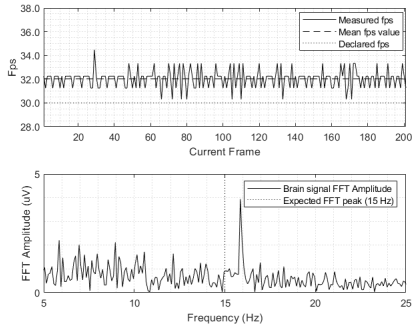


Figure 6.3. Epson Moverio BT-350: measured and expected FPS (top); measured and expected FFT peak of the relative user brain signal (bottom).

6.4.2 Benchmark

The obtained experimental results, considering a 5-s and 1.5-s acquisition times, are shown in bold respectively in Table 6.7 and 6.8 along with a comparison with other state-of-the-art techniques [192]. The classification accuracy is reported.

Since FR+ANN VAF processing on 1.5-s segmentation performed poorly, it was decided to use Domain Adaptation strategies only on the EEGNet model. The DA strategy is indicated with the δ symbol, in which, considering the similarity between the subjects, the 3 subjects considered as outliers will be part of the training set as discussed in section 6.2.4. With δ DA strategy, it can be seen how important the choice of subjects is in the test set, if the SSVEP signal is not observable or to a lesser degree in any of them, performance drops. The difference with the solutions found in the literature, which only perform one run, is even more evident. Therefore, using the two DA strategies of normalisation (i) and (ii) on sets created

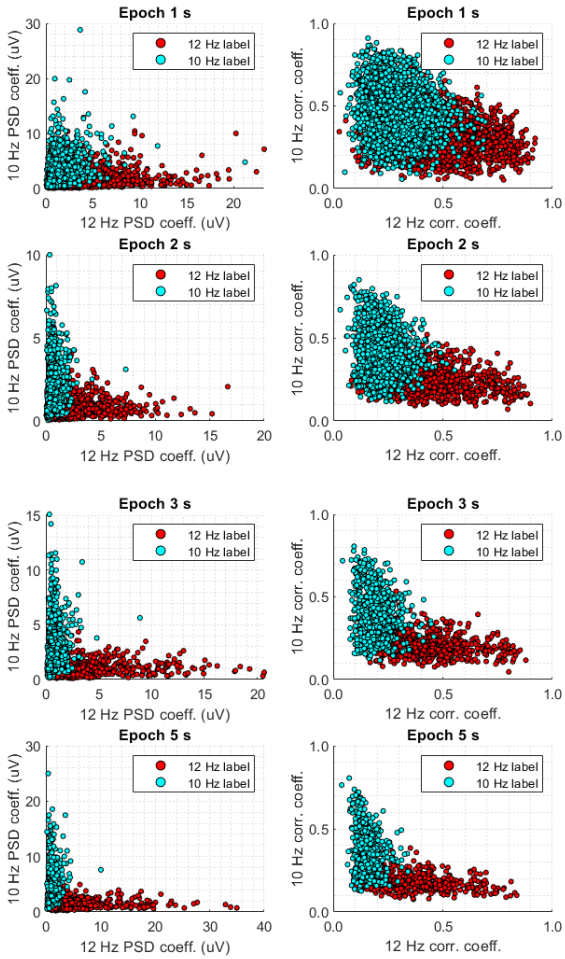


Figure 6.4. Scatter plots of the extracted features for data set #1 (BT-200) with different time responses (epochs).

using the δ approach, gives a greater contribution in terms of accuracy, whereas on sets created without considering the 3 outliers, the improvement is negligible.

Table 6.7. Classification accuracy on the Benchmark dataset with a 5-s acquisition time.

Method	Accuracy (%)
EEGNet δ norm(ii)	**96.54 ± 2.15**
EEGNet δ norm(i)	**96.43 ± 2.07**
EEGNet δ	**95.98 ± 1.91**
EEGNet norm(ii)	**94.45 ± 2.97**
EEGNet norm(i)	**94.46 ± 2.77**
EEGNet	**93.77 ± 3.79**
FR + ANN VAF	**95.51 ± 7.65**
1D SSVEP Convolutional Unit [192]	68.63
PodNet [192]	86.19
Filter Bank CCA [192]	97.92

Table 6.8. Classification accuracy on the Benchmark dataset with a 1.5-s acquisition time.

Method	Accuracy (%)
EEGNet δ norm(ii)	**76.46 ± 6.28**
EEGNet δ norm(i)	**76.41 ± 6.69**
EEGNet δ	**75.40 ± 6.83**
EEGNet norm(ii)	**70.58 ± 4.55**
EEGNet norm(i)	**70.49 ± 5.28**
EEGNet	**70.03 ± 5.16**
FR + ANN VAF	**39.86 ± 17.53**
1D SSVEP Convolutional Unit [192]	32.67
PodNet [192]	75.64
Filter Bank CCA [192]	84.00

6.5 Conclusion

This chapter proposed the adoption of ML techniques to enhance the classification performance of a highly wearable, single-channel instrumentation for BCI, based on the detection and classification of SSVEPs. In this measurement system, AR HMDs are used to generate the flickering stimuli necessary to SSVEPs elicitation; it guarantees greater immersivity and engagement with respect to traditional LCDs.
Then, a custom ANN, which relies on the adoption of Variable Activation Functions, and a Deep NN, EEGNet, with Domain Adaptation methods, were proposed to improve the SSVEPs classification, in terms of classification accuracy and time response. The proposed frameworks were tested

on the Benchmark dataset based on a traditional SSVEP-BCI setup. The obtained experimental results showed a significant enhancement of the performance with respect to the traditional state-of-the-art methods.

Part II

eXplainable Artificial Intelligence

Chapter 7

XAI: a background

Introduction

In this chapter, an overview of eXplainable artificial intelligence will be made. After a general description, the main methods in the literature will be analysed.

7.1 Definitions

A large part of Machine Learning (ML) techniques – including Support Vector Machines (SVM) and Deep Neural Networks (DNN) – give rise to systems having behaviours often complex to interpret [6]. More precisely, although ML techniques with reasonably well interpretable mechanisms and outputs exist, as, for example, decision trees, the most significant part of ML techniques give responses whose relationships with the input are often difficult to understand. In this sense, they are commonly considered as black-box systems. In particular, as ML systems are being used in more and more domains and, so, by a more varied audience, there is the need for making them understandable and trusting to general users [204, 37]. Hence, generating explanations for ML system behaviours that are understandable to human beings is a central scientific and technological issue addressed by the rapidly growing research area of eXplainable Artificial Intelligence (XAI).

Several definitions of interpretability/explainability for ML systems

have been discussed in the XAI literature [78, 37], and many approaches to the problem of overcoming their opaqueness are now pursued [180, 40, 18]. For example, in [170] a series of techniques for the interpretation of DNNs is discussed, and in [157] the authors examine and discuss the motivations underlying the interest in ML systems' interpretability, discussing and refining this notion.

In the literature, particular attention is given to *post-hoc* explainability [37], i.e., the methods to provide explanations for the behaviours of non-interpretable models after the training. In the context of this multifaceted interpretability problem, it is noted that in the literature, one of the most successful strategies is to provide explanations in terms of "visualisations" [204, 251].

More specifically, explanations for image classification systems are given in terms of low-level input features, such as relevance or heat maps of the input built by model-agnostic (without disclosing the model internal mechanisms) or model-specific (accessing to the model internal mechanisms) methods, like sensitivity analysis [219] or Layer-wise Relevance Propagation (LRP) [40], see figure 7.1. The main problem with such methods is that human users are left with a significant interpretive burden. Starting from each low-level feature's relevance, the human user needs to identify the overall input properties perceptually and cognitively salient to him [18].

7.2 XAI Methods

To the aim of illustrating the heterogeneity of XAI approaches proposed in the literature, the main XAI methods will be introduced and discussed in the following subsections.

7.2.1 Saliency

Saliency method is one the of the simplest and more intuitive method to build an explanation of a ML system. Proposed in [218], Saliency method is based on the gradient of the output function of the ML system respect to its input. In a nutshell, an explanation of the output $C(\mathbf{x})$ of a ML system fed with an input $\mathbf{x} \in \mathbb{R}^d$ is built generating a saliency map lever-

aging on the gradient $\frac{\partial C}{\partial \mathbf{x}}$ of C with respect to its input computed through backpropagation. The magnitude of the gradient indicates how much the features need to be changed to affect the class score.

7.2.2 Guided BackPropagation

Guided BackPropagation (Guided BP) [225] can be viewed as a slighlty variation of Saliency method proposed in [218]. The main difference is in the value used as gradient in case of rectified activation functions (ReLU): in Saliency method, the real gradient is used in computing the features relevance. Instead, Guided BP starts from the hypothesis that the user is not interested if a feature "decreases" (i.e., negative value) a neuron activation, but only in the most relevant ones. Therefore, instead of the true gradient, in guided BP a gradient transformation is used to prevent backward flow of negative values, avoiding to decrease the neuron activations and highlighting the most relevant features. Obviously, Guided BP can fail to highlight inputs that contribute negatively to the output due to "zero-ing" the negative values.

7.2.3 Layer-wise Relevance Propagation

Layer-wise Relevance Propagation (LRP) associates a relevance value to each input element (pixels in case of images) to build explanations for the ML model answer. In a nutshell, the output $C(\mathbf{x})$ of a ML system on an input $\mathbf{x} \in \mathbb{R}^d$ is decomposed as a sum of relevances on the single features composing $\mathbf{x}$, i.e. $C(\mathbf{x}) \simeq \sum_{i=1}^{d} R_i$ where R_i is a score of the local contribution of the i-th feature on the produced output. In particular, positive values denote positive contributions, while negative values negative contributions. Applied to ANN, this principle can be generalised across each pair of consecutive layers l and $l+1$ of a network composed of L layers such that $\sum_{i=1}^{q} R_i^{(l+1)} = \sum_{i=1}^{q'} R_i^{(l)}$ where q and q' are the features of the layers $l+1$ and l respectively. Since the final network output $C(\mathbf{x})$ of an ANN is the output of the L-th layer, it results that $C(\mathbf{x}) = \cdots = \sum_{i=1}^{q} R_i^{(l+1)} = \sum_{i=1}^{q'} R_i^{(l)} = \cdots = \sum_{i=1}^{d} R_i$. This rule can be in-

terpreted as a conservation rule, and leveraging on that different methods to compute the relevance have been proposed, depending on the type of features involved. In case of densely connected layers, the most known rule is the $z-rule$ [41], which takes care of the neuron activations of each layer to compute the final relevance of each layer.

Figure 7.1. Examples of predictions using LRP for the class "persons" [40].

7.2.4 Integrated Gradients

One of the main drawbacks of simple gradient-based method is that the gradient respect to the input should be small in the neighbourhood of the input features also for relevant ones. Instead of using only the gradient respect to the original input, [226] proposed to average all the gradients between the original input $\mathbf{x}$ and a baseline input $\mathbf{x}^{ref}$ (that is, an input s.t. $C(\mathbf{x}^{ref})$ results in a neutral prediction). In this way, if features of inputs closer to the baseline have higher gradient magnitudes, they are taken into account thanks to the average operator. More formally, the importance of each feature x_i computed by Integrated Gradient (IG) is defined as:

$$IG(x_i) = (x_i - x_i^{ref}) \int_{\alpha=0}^{1} \frac{\partial C\big(x_i^{ref} + \alpha(x_i - x_i^{ref})\big)}{\partial x_i} \, d\alpha$$

In other words, IG aggregates the gradients along the intermediate inputs on the straight-line between the baseline and the input, selected as $\alpha \in [0, 1]$ changes.

7.2.5 DeepLIFT

In [216] a method consisting in assigning feature relevance scores according to the difference between the neurons activation and a reference activation (such as the baseline for Integrated Gradient method) is proposed. The authors proposed to compute for each feature a multiplier entity similar to a partial derivative, but leveraging over finite differences instead of infinitesimal ones. Each multiplier can be defined as $m_{\Delta x \Delta t} = \frac{R_{\Delta x \Delta t}}{\Delta x}$ and represents the ratio between i) the contribution $R_{\Delta x \Delta t}$ of the difference $\Delta x = x - x^{ref}$ from the reference x^{ref} of each feature x to the difference $\Delta t = t - t^{ref}$ between the output t and the reference output t^{ref}, and ii) the difference Δx. Therefore, the authors proposed a set of rules to compute the features relevance based on the proposed multipliers exploiting a Back Propagation-based approach.

7.2.6 Local Interpretable Model-Agnostic Explanations

A popular method based on middle-level properties of the input is Local Interpretable Model-Agnostic Explanations (LIME) [203], which returns a set of image parts (superpixels), that could have driven the ML model to the given answer (see figure 7.2). This set of superpixels can be then considered as an explanation to the ML model response. This approach can be classified as model-agnostic.

Model-agnostic approaches correspond to XAI methods which are independent of the ML model to be explained [6], i.e., model-agnostic solutions are built relying only the relation between ML model inputs and outputs, without any consideration about the ML model internal state. Although this property ensures the applicability of these approaches to any ML model, on the other hand, the explanations of the model-agnostic methods could not be fully related to the actual causal relationships between model's inputs and outputs which have contributed to the given model response. For instance, LIME returns an explanation inspecting the behaviour of the model in the neighbourhood of the input, but nothing ensures that, for that particular input instance, the answer of the classifier has a totally different explanation (for example, a particular on the background of the specific input image which the model has already seen during the training stage, making the model biased).

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 7.2. Examples of predictions using LIME for the classes "Electric Guitar", "Acoustic guitar" and "Labrador" [203].

7.2.7 Prototypical Part Network

In contrast to the post-hoc methods seen so far, there are other methods in the literature that are part of ad-hoc approaches, where the network architecture or the training process is changed for better interpretability. In [60], the authors define a particular DNN: Prototypical Part Network (ProtoPNet). They defined a form of of interpretability in image processing ("this looks like that"). In fact, the model is able to identify several parts of the image where it thinks that this part of the image looks like that prototypical part of some class, and makes its prediction based on a weighted combination of the similarity scores between parts of the image and the learned prototypes. So, the model has a transparent reasoning process when making predictions.
Examples of post-hoc visualization techniques, such as saliency or LRP, do not explain the reasoning process of how a network actually makes its decisions. In contrast, ProtoPNet has a built-in case-based reasoning process, and the explanations generated by the network are used during classification and are not created post-hoc. ProtoPNet relates closely to works that build attention-based interpretability into CNNs [265, 261, 266]. Attention-based models can only tell us which parts of the input they are looking at, they do not point us to prototypical cases to which the parts they focus on are similar. On the other hand, ProtoPNet is not only able to expose the parts of the input it is looking at, but also point to prototypical cases similar to those parts.

7.2.8 Proxies Methods

To obtain the model-agnostic interpretability a surrogate or a simple proxy model can be used to learn a faithful approximation of a more complex model and black-box exploiting the outputs returned by the black-box model. One of the first methods in which an attempt is made to approximate the black-box model using a simple interpretable model can be found in [71] where the authors extracted comprehensible and symbolic representations from trained neural networks. The proposed method builds a decision tree that approximates the concepts represented by a given network. It is able to produce decision trees that maintain a high level of fidelity to the respective networks while being understandable and accurate.
In [55] a method for compressing large, complex ensembles into smaller, faster models, without significant loss in performance. This approach was a first example of knowledge distillation works [111].
The LIME method, described above 7.2.6, also falls into this type of approach. In fact, LIME builds explanation relying on a proxy model different from the model to explain.

7.2.9 Twin systems using examples

The paper [124] exploits twin-systems models for providing good explanations-by-example. In XAI, a twin system joins a complex black-box model with more transparent white-box model, e.g., a k-NN or Case-Based Reasoning (CBR) model, using the latter to explain the former by finding a mapping between them. While twin-systems are usually considered as a class of hybrid systems, in XAI they are be identified as a special case of a proxy system.
The authors proposed a hybrid system where an ANN or a DNN model and a CBR technique are combined together to meet the system requirements of accuracy and interpretability. Feature-weights, feature-importance, or predictive outcomes, learned from the ANN model, are mapped to the CBR system. The ANN model provides predictions, and the CBR module produces interpretability by explaining ANN outputs (in classification or regression), using factual, counterfactual or semi-factual cases.

7.3 Interesting for BCI

The interpretation of brainwaves is a fascinating challenge that many scientists have undertaken to study [185]. An explainable approach is very important in critical environments, such as hospitals, to understand which part or feature of the input caused the system to classify it in a certain way and which features on the other hand may cause misclassification [171]. Furthermore, explainability is strongly demanded by governmental institutions, such as the European Union's new General Data Protection Regulation (GDPR), which explicitly requires a 'right to explanation' for Artificial Intelligence (AI) algorithms [105].

According to researcher Kundu, AI in medicine must be explainable [139]. AI algorithms used for diagnosis and prognosis must be explainable and must not rely on a black box. Furthermore, interpretability should always be present in AI models in medicine and developed at an early stage.
Also in the opinion of the authors of [113], using XAI in the medical field can contribute to relevant results by enabling medical professionals to understand how and why an ML system made a certain decision.

7.3.1 XAI methods for BCI

In recent years, there has been an increasing amount of work in the literature based on XAI techniques for the explainability of classifier outputs in BCI systems, particularly with EEG signals.
Y. Al Hammadi et al. [8] used an XAI approach by analysing a wide range of physiological signals, including EEG, to score the importance of the features used in the model that led to the classification of different emotional states in 17 individuals using the AI algorithm.
Morabito et al. [171] proposed an XAI method to monitor individual changes in EEG related to degeneration from Mild Cognitive Impairment (MCI) to dementia due to Alzheimer's Disease (AD), using high-density electroencephalogram (HD-EEG). The study revealed which EEG channels (i.e. head region) and frequency ranges (i.e. sub-bands) are most active in the progression from MCI to AD using Grad-CAM methods.
Li et al. [148] used a recurrent neural network model to process a public EEG dataset of motor imagery in order to select the channels for motion

intention recognition using Grad-CAM visualisation technology.
In an interesting and recent work, Giudice et al. [93] exploit the XAI Grad-CAM and LIME techniques to visually explain with EEG traces and regions of relevance, the most significant temporal parts during voluntary or involuntary eye blink.

Chapter 8

XAI: Middle-Level input Features

Introduction

An XAI approach should alleviate the weakness of low-level approaches and overcome their limitations, allowing the possibility to construct explanations in terms of input features that represent more salient and understandable input properties for a user, which will be called here Middle-Level input Features (MLFs) (see Figure 8.1). Although there is a recent research line which attempts to give explanations in terms of visual human-friendly concepts [125, 92, 7] (Section 8.1), however it is noticeable that the goal to learn data representations that are easily factorised in terms of meaningful features is, in general, pursued in the *representation learning* framework [45], and more recently in the *feature disentanglement learning* context [159]. These meaningful features may represent parts of the input such as nose, ears and paw in case of, for example, face recognition tasks (similarly to the outcome of a clustering algorithm) or more abstract input properties such as shape, viewpoint, thickness, and so on, leading to data representations perceptually and cognitively salient to the human being. Based on these considerations, in this chapter, the aim is to develop an XAI approach able to give explanations for an image classification system in terms of features which are obtained by standard representation learning methods such as variational auto-encoder [64] and hierarchical image seg-

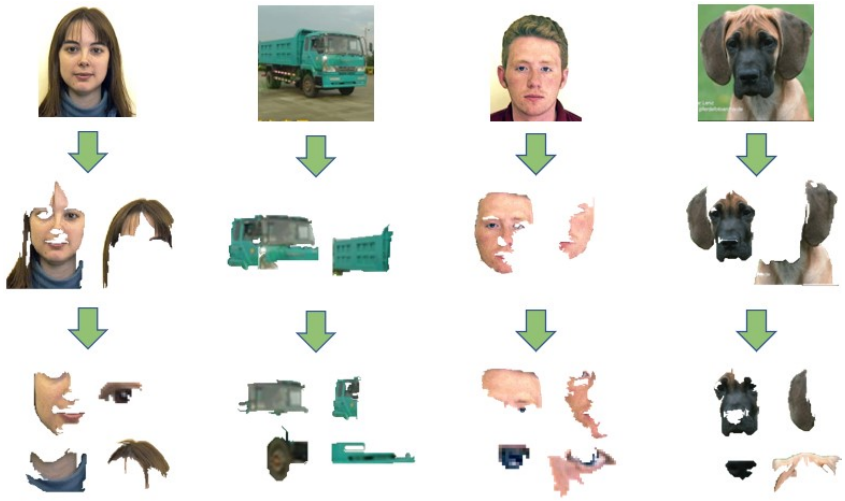


Figure 8.1. Examples of Middle Level input Features (MLFs). Each MLF represents a part of the input which is perceptually and cognitively salient to a human being, as for example the ears of a cat or the wings of an airplane. These features are intuitively more humanly interpretable respect to low-level features (as for example raw unrelated image pixels), so a decision explanation expressed in terms of MLF relevance can be easier to understand for a human being respect to explanations expressed in terms of low level features.

mentation [88]. In particular, middle-level data representations obtained by auto-encoder methods [58] is exploited to provide explanations of image classification systems. In this context, in an earlier work [24] an initial experimental investigation on this type of explanations exploiting the *hierarchical* organisation of the data in terms of more elementary factors is proposed. For example, natural images can be described in terms of the objects they show at various levels of granularity [230, 223, 258]. Or in [98] a hierarchical prototype-based approach for classification is proposed. This method has a certain degree of intrinsic transparency, but it does not fall into post-hoc explainability category.

To the best of my knowledge, in the XAI literature, however, there are relatively few approaches that pursue this line of research. In [203], the authors proposed LIME, a successful XAI method which is based, in case of image classification problems, on explanations expressed as sets of

regions, clusters of the image, said superpixels which are obtained by a clustering algorithm. These superpixels can be interpreted as MLFs. In [18, 17] the explanations are formed of elements selected from a dictionary of MLFs, obtained by sparse dictionary learning methods [77]. In [100] authors propose to exploit the latent representations learned through an adversarial auto-encoder for generating a synthetic neighbourhood of the image for which an explanation is required. However, these approaches propose specific solutions which cannot be generalised to different types of input properties. By contrast, in this chapter, the goal is to investigate the possibility of obtaining explanations using an approach that can be applied to different types of MLFs, which will be referred *General MLF Explanations* (GMLF). More precisely, the aim is to develop an XAI framework that can be applied whenever a) the input of an ML system can be encoded and decoded based on MLFs, and b) any Explanation method producing a Relevance Map (ERM method) can be applied on both the ML model and the decoder. In this sense, a general framework insofar is proposed as it can be applied to several different computational definitions of MLFs and a large class of ML models. Consequently, multiple and different explanations can be given based on different MLFs. In particular, the aim is to test a novel approach in the context of image classification using MLFs extracted by three different methods: 1) image segmentation by auto-encoders, 2) hierarchical image segmentation by auto-encoders, and 3) Variational auto-encoders. About the points 1) and 2), a simple method to represent the output of a segmentation algorithm in terms of encoder-decoder is reported. However, this approach can be used on a wide range of different data types to the extent that encoder-decoder methods can be applied.

Thus, the medium or long-term objective is to develop a XAI general approach producing explanations for an ML System behaviour in terms of potentially different and user-selected input features, composed of input properties which the human user can select according to his background knowledge and goals. This aspect can play a key role in developing *user-centred* explanations. It is essential to note that, in making an explanation understandable for a user, it should be taken into account what information the user desires to receive [125, 204, 16, 19]. Recently, it is becoming more and more evident that new directions to create better explanations

should take into account what a good explanation is for a human user, and consequently to develop XAI solutions able to provide user-centred explanations [125, 204, 152, 7, 126]. By contrast, much of the current XAI methods provide specific ways to build explanations that are based on the researchers' intuition of what constitutes a "good" explanation [152, 168].

To summarise, the following novelties are presented:

1. a XAI framework where middle-level or high-level input properties can be built exploiting standard methods of data representation learning is proposed;

2. proposed framework can be applied to several different computational definitions of middle-level or high-level input properties and a large class of ML models. Consequently, multiple and different explanations based on different middle-level input properties can be possibly provided given an input-ML system response;

3. The middle-level or high-level input proprieties are computed independently from the ML classifier to be explained.

The chapter is organised as follows: Section 8.2 describes in detail the proposed approach; in Section 8.1 differences and advantages of GMLF with respect similar approaches presented in the literature are discussed; experiments and results are discussed in Section 8.4. In particular, proposed approach with LIME method and performed both qualitative and quantitative evaluations of the results is compared; the concluding Section summarises the main high-level features of the proposed explanation framework and outlines some future developments.

8.1 Related Works

The importance of eXplainable Artificial Intelligence (XAI) is discussed in several papers [242, 168, 207, 38]. Different strategies have been proposed to face the explainability problem, depending both on the AI system to explain and the type of explanation proposed. Among all the XAI works proposed over the last years, an important distinction is between model-based and post-hoc explanaibility [176], the former consisting

in AI systems explainable by design (e.g., decision trees), since their inner mechanisms are easily interpreted, the latter proposing explanation built for system that are not easy to understand. In particular, several methods to explain Deep Neural Networks (DNNs) are proposed in the literature due to the high complexity of their inner structures. A very common approach consists in returning visual-based explanations in terms of input feature importance scores, as for example Activation Maximization (AM) [82], Layer-Wise Relevance propagation (LRP) [40], Deep Taylor Decomposition[48, 169], Class Activation Mapping (CAM) methods [265, 215], Deconvolutional Network [249] and *Up-convolutional network* [248, 79]. Although heatmaps seem to be a type of explanation that is easy to understand for the user, these methods build relevances on the low-level input features (the single pixel), while input middle-level properties which determined the answer of the classifier have to be located and interpreted by the user, leaving much of the interpretive work to the human beings. On the other side, methods as Local Interpretable Model-agnostic Explanations (LIME) [203] relies on feature partitions, as super-pixel in the image case. However, the explanations given by LIME (or its variants) are built through a new model that approximates the original one, thus risking to loose the real reasons behind the behaviour of the original model [204].

Recently, a growing number of studies [266, 125, 92, 7] have focused on providing explanations in the form of middle-level or high-level human "concepts" as intended in this chapter. In particular, in [125] the authors introduce the Concept Activation Vectors (CAV) as a way of visually representing the neural network' inner states associated with a given class. CAVs should represent human-friendly concepts. The basic ideas can be described as follow: firstly, the authors suppose the availability of an external labelled dataset XC where each label corresponds to a human-friendly concept. Then, given a pre-trained neural network classifier to be explained, say NC, they consider the functional mapping f_l from the input to the l-layer of NC. Based on f_l, for each class c of the dataset XC, they build a linear classifier composed of f_l followed by a linear classifier to distinguish the element of XC belonging to the class c from randomly chosen images. The normal to the learned hyperplane is considered the CAV for the user-defined concept corresponding to the class c. Finally,

given all the input belonging to a class K of the pre-trained classifier NC, the authors define a way to quantify how much a concept c, expressed by a CAV, influences the behaviour of the classifier, using directional derivatives to computes NC 's conceptual sensitivity across entire class K of inputs.

Building upon the paper discussed above, in [7] the authors provide explanations in terms of *fault-lines*[121]. Fault-lines should represent "high-level semantic aspects of reality on which humans zoom in when imagining an alternative to it". Each fault-line is represented by a minimal set of semantic *xconcepts* that need to be added to or deleted from the classifier's input to alter the class that the classifier outputs. Xconcepts are built following the method proposed in [125]. In a nutshell, given a pre-trained convolutional neural network CN whose behaviour is to be explained, xconcepts are defined in terms of super-pixels (images or parts of images) related to the feature maps of the l-th CN's convolutional layer, usually the last convolutional layer before the full-connected layer. In particular, these super-pixels are collected when the input representations at the convolution layer l are used to discriminate between a target class c and an alternate class c_{alt}, and they are computed based on the Grad-CAM algorithm [215]. In this way, one obtains xconcepts in terms of images related to the class c and able to distinguish it from the class c_{alt}. Thus, when the classifier CN responds that an input x belongs to a class c, the authors provide an explanation in terms of xconcepts which should represent semantic aspects of why x belongs c instead of an alternate class c_{alt}.

In [92] the authors propose a method to provide explanations related to an entire class of a trained neural classifier. The method is based on the CAVs introduced in [125] and sketched above. However, in this case, the CAVs are automatically extracted without the need an external labelled dataset expressing human-friendly concepts.

In addition, several works in literature build attention-based interpretability in DNN, which aims to highlight parts of an input that the network focuses on when making decisions [265, 261, 266]. By extending these ideas, into [60] the input parts considered relevant by the method are linked to similar prototype parts learned from the training set. Compared to our approach where MLFs are part of the input images, in [60] prototype parts are associated with other training images.

Many of the approaches discussed so far focus on *global* explanations, i.e., explanations related to en entire class of the trained neural network classifier (see [92, 125]). Instead, in proposed approach, *local* explanations are desired, i.e., explanations for the response of the ML model to each single input. Some authors, see for example [125], provide methods to obtain *local* explanations, but in this case, the explanations are expressed in terms of high-level visual concepts which do not necessarily belong to the input. Thus, again human users are left with a significant interpretive load: starting from external high-level visual concepts, the human user needs to identify the input properties perceptually and cognitively related to these concepts. On the contrary, the input (MLFs) high-level properties are expressed, in this approach, in terms of elements of the input itself.

Another critical point is that high-level or middle-level user-friendly concepts are computed on the basis of the neural network classifier to be explained. In this way, a short-circuit can be created in which the visual concepts used to explain the classifier are closely related to the classifier itself. By contrast, in proposed approach, MLFs are extracted independently from the classifier.

A crucial aspect that distinguishes this proposal from the above-discussed research line is grounded on the fact that an XAI framework is proposed able to provide multiple explanations, each one composed of a specific type of middle-level input features (MLFs). Proposed methodology only needs that MLFs can be obtained using methods framed into data representation research, and, in particular, any auto-encoder architecture for which an explanation method producing a relevance map can be applied on the decoder (see Section 8.2.1).

To summarise, the GMLF approach, although shares with the above describe research works the idea to obtain explanations based on middle-level or high-level human-friendly concepts, presents the following elements of novelty:

1. It is a XAI framework where middle-level or high-level input properties can be built on the basis of standard methods of data representation learning.

2. It outputs local explanations.

3. The middle-level or high-level input proprieties are computed inde-

pendently from the ML classifier to be explained.

Regarding points 2) and 3) note that a XAI method that has significant similarity with proposed approach is LIME [203] or its variants (see, for example, [260]). LIME, especially in the context of images, is one of the predominant XAI methods discussed in the literature [76, 260]. It can provide *local* explanations in terms of superpixels which are regions or parts of the input that the classifier receives, as already discussed in Section 8. These superpixels can be interpreted as middle-level input properties, which can be more understandable for a human user than low-level features such as pixels. In this sense, there is a similarity in the output between the approach GMLF and LIME. The explanations built by LIME can be considered comparable with the proposed approach but different in the construction process. While LIME (and other proxies methods, see 7.2.8) builds explanation relying on a proxy model different from the model to explain, the proposed approach relies only on the model to explain, without needing any other model that approximates the original one. To highlight the difference between the produced explanations, in section 8.3 a comparison between LIME and GMLF outputs is made.

8.2 Approach

This approach stems from the following observations.

The development of data representations from raw low-level data usually aims to obtain distinctive explanatory features of the data, which are more conducive to subsequent data analysis and interpretation. This critical step has been tackled for a long time using specific methods developed exploiting expert domain knowledge. However, this type of approach can lead to unsuccessful results and requires a lot of heuristic experience and complex manual design [146]. This aspect is similar to what commonly occurs in many XAI approaches, where the explanatory methods are based on the researchers' intuition of what constitutes a "good" explanation.

By contrast, representation learning successfully investigates ways to obtain middle/high-level abstract feature representations by automatic machine learning approaches. In particular, a large part of these approaches is based on Auto-Encoder (AE) architectures [58, 146]. AEs correspond to neural networks composed of at least one hidden layer and

logically divided into two components, an *encoder* and a *decoder*. From a functional point of view, an AE can be seen as the composition of two functions E and D: E is an encoding function (the encoder) which maps the input space onto a feature space (or latent encoding space), D is a decoding function (the decoder) which inversely maps the feature space on the input space. A meaningful aspect is that by AEs, one can obtain data representations in terms of latent encodings $\vec{h}$, where each h_i may represent a MLF ξ_i of the input , such as parts of the input (for example, nose, ears and paw) or more abstract features which can be more salient and understandable input properties for a user. See for example variational AE [130, 201, 149] or image segmentation [89, 63, 246, 256] (see Figure 8.1). Furthermore, different AEs can extract different data representations which are not mutually exclusive.

Based on the previous considerations, the goal is to build upon the idea that the elements composing an explanation can be determined by an AE which extracts relevant input features for a human being, i.e., MLFs, and that one might change the type of MLFs changing the type of auto-encoder or obtain multiple and different explanations based on different MLFs.

8.2.1 General description

Given an ML classification model M which receives an input $\vec{x} \in R^d$ and outputs $\vec{y} \in R^c$, this approach can be divided into two consecutive steps.

In the first step, an auto-encoder $AE \equiv (E, D)$ is built such that each input $\vec{x}$ can be encoded by E in a latent encoding $\vec{h} \in R^m$ and decoded by D. As discussed above, to each value h_i is associated a MLF ξ_j , thus each input x is decomposed in a set of m MLFs $\vec{\xi} = \{\xi_i\}_{i=1}^m$, where to each ξ_i is associated the value h_i. Different choices of the auto-encoder can lead to MLFs $\vec{\xi_i}$ of different nature, so to highlight this dependence this first step is re-formalised as follows: an encoder $E_{\vec{\xi}} : \vec{x} \in R^d \rightarrow \vec{h} \in R^m$ and a decoder $D_{\vec{\xi}} : \vec{h} \in R^m \rightarrow \vec{x} \in R^d$ are built, where $\vec{h}$ encodes $\vec{x}$ in terms of the MLFs $\vec{\xi}$.

In the second step of this approach, an ERM method (an explanation method producing a relevance map of the input) is used on both M and $D_{\vec{\xi}}$, i.e., by applying it to the model M and then use the obtained relevance

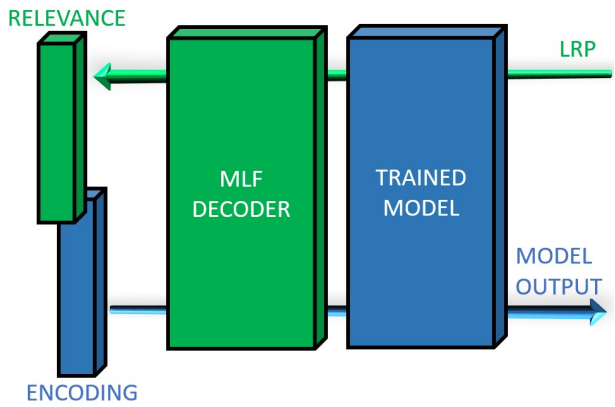


Figure 8.2. A general scheme of the proposed explanation framework. Given a middle-level feature encoder and the respective decoder, this last one is stacked on the top of the model to inspect. Next, the encoding of the input is fed to the decoder-model system. A backward relevance propagation algorithm is then applied.

values to apply the ERM method on $D_{\vec{\xi}}$ getting a relevance value for each middle-level feature. In other words, $D_{\vec{\xi}}$ is stacked on the top of M thus obtaining a new model $DM_{\vec{\xi}}$ which receives as input an encoding $\vec{u}$ and outputs $\vec{y}$, and uses an ERM method on $DM_{\vec{\xi}}$ from $\vec{y}$ to $\vec{u}$. In Figure 8.2 a graphic description of the approach GMLF is given, and in algorithm 1) it is described in more details considering a generic auto-encoder, while in algorithms 3 and 4 the approach (GMLF) is described in case of specific auto-encoders (see Section 8.2.2 and 8.2.3).

Thus, the aim is to search for a relevance vector $\vec{u} \in R^m$ which informs the user how much each MLF of $\vec{\xi}$ has contributed to the ML model answer $\vec{y}$. Note that, GMLF can be generalised to any decoder $D_{\vec{\xi}}$ to which a ERM method applies on. In this way, one can build different explanations for a M's response in terms of different MLFs $\vec{\xi}$.

In the remainder of this section, three alternative ways (segmentation, hierarchical segmentation and VAE) to obtain a decoder will be described such that a ERM method can be applied to, and so three ways of applying the approach GMLF. This framework is experimentally tested using all the methods.

Algorithm 1: Proposed method GMLF

Input: data point $\vec{x}$, trained model M, an ERP method RP
Output: Feature Relevances U

$\vec{y} \leftarrow M(\vec{x})$;
build an auto-encoder $AE \equiv (E_\xi, D_\xi)$;
$\vec{h} \leftarrow E_\xi(\vec{x})$;
define $R : \vec{h} \mapsto \vec{x} - D_\xi(\vec{h})$;
define $DM_\xi : \vec{h} \mapsto M(D_\xi(\vec{h}) + R(\vec{h}))$;
$U \leftarrow RP(DM_\xi, \vec{h}, \vec{y})$;
return U;

8.2.2 MLFs from image segmentation

Here the implementation of the GMLF approach is described to the case of an auto-encoder built of the basis of hierarchical segmentation. The approach is depicted in Figure 8.3, while an algorithmic formalisation is given in algorithms 2 and 3.

Given an image $\vec{x} \in R^d$, a segmentation algorithm returns a partition of $\vec{x}$ composed of m regions $\{q_i\}_{i=1}^m$. Some of the existing segmentation algorithms can be considered *hierarchical segmentation algorithms*, since they return partitions hierarchically organised with increasingly finer levels of details.

More precisely, following [102], a segmentation algorithm is considered hierarchical if it ensures both the causality principle of multi-scale analysis [101] (that is, if a contour is present at a given scale, this contour has to be present at any finer scale) and the location principle (that is, even when the number of regions decreases, contours are stable). These two principles ensure that the segmentation obtained at a coarser detail level can be obtained by merging regions obtained at finer segmentation levels.

In general, given an image, a possible set of MLFs can be the result of a segmentation algorithm. Given an image $\vec{x} \in R^d$, and a partition of $\vec{x}$ consisting of m regions $\{q_i\}_{i=1}^m$, each image's region q_i can be represented by a vector $\vec{v_i} \in R^d$ defined as follows: $v_{ij} = 0$ if $x_j \notin q_i$, otherwise $v_{ij} = x_j$, and $\sum_{i=1}^m \vec{v_i} = \vec{x}$. Henceforth, for simplicity and without loss of generality, $\vec{v_i}$ will be used instead of q_i since they represent the same

entities. Consequently, $\vec{x}$ can be expressed as linear combination of the $\vec{v}_i$ with all the coefficients equal to 1, which represent the encoding of the image $\vec{x}$ on the basis of the m regions. More in general, given a set of K different segmentations $\{S_1, S_2, \ldots, S_K\}$ of the same image sorted from the coarser to the finer detail level, it follows that, if the segmentations have a hierarchical relation, each coarser segmentation can be expressed in terms of the finer ones. More in detail, each region $\vec{v}_i^k$ of S_k can be expressed as a linear combination $\sum_j \alpha_j \vec{v}_j^{k+1}$ where α_j is 1 if all the pixels in $\vec{v}_j^{k+1}$ belong to $\vec{v}_i^k$, 0 otherwise. The same reasoning can be applied going from S_K to the image $\vec{x}$ considering it as a trivial partition S_{K+1} where each region represents a single image pixel, i.e., $S_{K+1} = \{\vec{v}_1^{K+1}, \vec{v}_2^{K+1}, \ldots, \vec{v}_d^{K+1}\}$, with $v_{ij}^{K+1} = x_j$ if $i = j$, otherwise $v_{ij}^{K+1} = 0$.

It is straightforward to construct a feed-forward full connected neural network of $K+1$ layers representing an image $\vec{x}$ in terms of a set of K hierarchically organised segmentations $\{S_k\}_{k=1}^K$ as follows (see Figure 8.3): the k-th network layer has $|S_k|$ inputs and $|S_{k+1}|$ outputs, the identity as activation functions, biases equal to 0 and each weights w_{ij}^k equal to 1 if the $\vec{v}_j^{k+1}$ region belongs to the $\vec{v}_i^k$ region, 0 otherwise. The last layer $K+1$ has d outputs and weights equal to $(\vec{v}_p^{K+1})_{p=1}^d$. The resulting network can be viewed as a decoder that, fed with the $\vec{1}$ vector, outputs the image $\vec{x}$.

Note that if one considers $K = 1$, it is possible to use the same approach in order to obtain an $\vec{x}$'s segmentation without a hierarchical organisation. In this case the corresponding decoder is a network composed of just one layer. It is intended to clarify that the segmentation module described in this section represents a way to build an auto-encoder which encodes latent variables that are associate to image segments. These image segments are candidate MLFs. Explanations are built by a selection of these candidate segments in the second computational step of this approach. It is emphasised that the first step of the framework is to build an auto-encoder so that each input can be decomposed in a set of MLFs where each latent variable is associated to a specific MLF. These MLFs represent candidate input properties to be included into the final explanation which is computed by the second computational step of this approach. In this second part, a number of candidate middle-level input features are selected by an explanation method producing a relevance map of the input such as Layer-wise Relavance Propagation method (LRP). However, different choices of

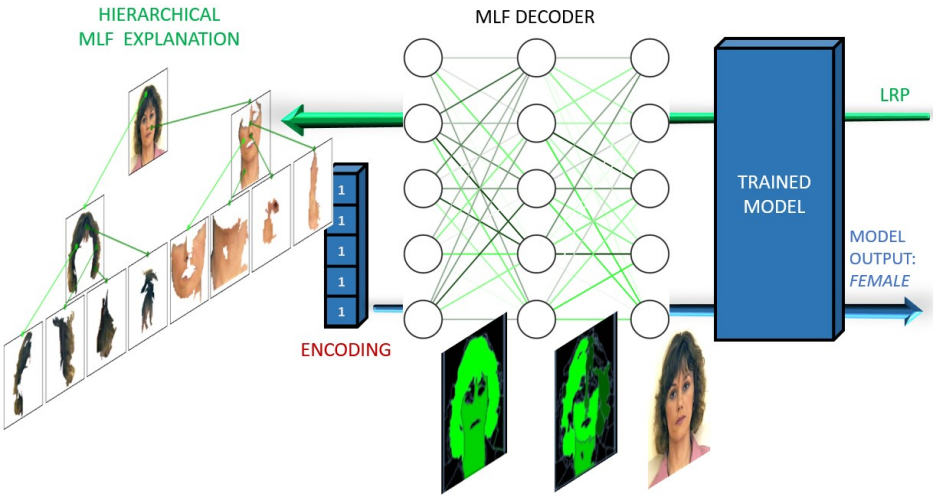


Figure 8.3. A segmentation-based MLF framework. MLF decoder is built as a neural network having as weights the segments returned by a hierarchical segmentation algorithm (see text for further details). The initial encoding is the "1" vector since all the segments are used to compose di input image. The relevance backward algorithm returns the most relevant segments.

the auto-encoder can lead to different MLFs of different nature.

Algorithm 2: Hierarchical segmantion-based Enconder-Decoder Generator

Input: data point $\vec{x}$, hierarchical segmentation procedure seg, hierarchical segmentation parameters $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_K)$

Output: A Decoder D_ξ, an Encoder E_ξ

$\{S_1, S_2, \ldots, S_K\} \leftarrow seg(\vec{x}, \vec{\lambda})$;
$S_{K+1} \leftarrow \emptyset$;
for $x_j \in \vec{x}$ **do**
  let $\vec{v}^{K+1} \in \{0\}^d$;
  $v_{jj}^{K+1} \leftarrow x_j$;
  $S_{K+1} \leftarrow S_{K+1} \cup \{\vec{v}^{K+1}\}$;
end
for $1 \leq k \leq K$ **do**
  let $W^k \in \{0\}^{|S_k| \times |S_{k+1}|}$;
  let $\vec{b}^k \in \{0\}^{|S_{k+1}|}$;
  for $1 \leq i \leq |S_k|$ **do**
    for $1 \leq j \leq |S_{k+1}|$ **do**
      if $\vec{v}_j^{k+1}$ ***belongs to*** $\vec{v}_i^k$ **then**
        $W_{ij} \leftarrow 1$;
      end
    end
  end
  define $identity : \vec{a} \mapsto \vec{a}$;
  $D_\xi \leftarrow generateNeuralNetwork($weights $= \{W^k\}_{k=1}^{K+1}$, biases $= \{\vec{b}^k\}_{k=1}^{K+1}$, activation function $= identity)$;
  define $E_\xi : \vec{x} \mapsto e \in \{1\}^{|S_1|}$;
  return D_ξ, E_ξ;
end

Algorithm 3: GMLF approach in case of Hierarchical segmentation-based auto-encoder

Input: a data point $\vec{x} \in R^d$, a $trainedNeuralNet$ returning the class scores given a data point, a hierarchical segmentation procedure seg, hierarchical segmentation parameters $\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_K)$, a relevance propagation algorithm RP returning a relevance vector for each network layer given: i) a neural network, ii) an input and iii) its class probabilities, a $generateNeuralNetwork$ function that returns a neural networks with weights, biases and activation function given as parameters

Output: relevances for the first K layers $\{\vec{u}_1, \ldots, \vec{u}_K\}$

1 $\vec{y} \leftarrow M(\vec{x})$;
   a) $\vec{y} \leftarrow TrainedNeuralNet(\vec{x})$;
2 build an auto-encoder $AE \equiv (E_\xi, D_\xi)$;
   a) $(E_\xi, D_\xi) \leftarrow buildAE(\vec{x}, seg, \vec{\lambda})$ ▷ see algorithm 2;
3 $\vec{h} \leftarrow E_\xi(\vec{x})$;
4 define $R : \vec{h} \mapsto \vec{x} - D_\xi(\vec{h})$:
   a) let $W_{res} \in \{0\}^{d \times d}$;
   b) $\vec{r} = \vec{x} - D_\xi(\vec{x})$;
   c) $\vec{b}_{res} \leftarrow \vec{r}$;
   d) define $identity : \vec{a} \mapsto \vec{a}$;
   e) $R \leftarrow generateNeuralNetwork$(weights $= \{W_{res}\}$, biases $= \{\vec{b}_{res}\}$, activation function $=$ $identity$);
5 define $DM_\xi : \vec{h} \mapsto M(D_\xi(\vec{h}) + R(\vec{h}))$:
   a) $DM_\xi \leftarrow$ stackTogether(D,R,M);
6 $U \leftarrow RP(DM_\xi, \vec{h}, \vec{y})$;
7 return $\{\vec{u}_1, \ldots \vec{u}_K\}$;

8.2.3 MLF from Variational auto-encoders

The concept of "entangled features" is strictly related to the concept of "interpretability". As stated in [110], a disentangled data representation is most likely more interpretable than a classical entangled data representation. This fact is due to the generative factors representation into separate latent variables representing single features of the data (for example, the size or the colour of the represented object in an image).

Using Variational Auto Encoders (VAE) is one of the most affirmed neural network-based methods to generate disentangled encodings. In general, a VAE is composed of two parts. First, an encoder generates an entangled encoding of a given data point (in this case, an image). Then a decoder generates an image from an encoding. Once trained with a set of data, the VAE output $\vec{\tilde{x}}$ on a given input $\vec{x}$ can be obtained as the composition of two functions, an encoding function $E(\cdot)$and a decoding function $D(\cdot)$, implemented as two stacked feed-forward neural networks.

The encoding function generates a data representation $E(\vec{x}) = \vec{h}$ of an image $\vec{x}$, the decoding function generates an approximate version $D(\vec{h}) = \vec{\tilde{x}}$ of $\vec{x}$ given the encoding $\vec{h}$, with a residual $\vec{r} = \vec{x} - \vec{\tilde{x}}$. So, it is possible to restore the original image data simply adding the residual to $\vec{\tilde{x}}$, that is $\vec{x} = \vec{\tilde{x}} + \vec{r}$. Consequently, the decoder neural networks are stacked with a further dense layer $R(\cdot)$ having d neurons with weights set to 0 and biases set to $\vec{r}$. The resulting network $R(E(\vec{h}))$ generates $\vec{x}$ as output, given its latent encoding $\vec{h}$.

In Figure 8.4 it is shown a pictorial description of GMLF approach when the auto-encoder is built based on VAE, the algorithmic description is reported in algorithm 4.

8.3 Experimental assessment

In this section, the aim is to describe the chosen experimental setup. The goal is to examine the applicability of this approach for different types of MLFs obtained by different encoders. As stated in Section 8.2.1, three different types of MLFs are evaluated: flat (non hierarchical) segmentation, hierarchical segmentation and VAE latent coding. For non-hierarchical/hierarchical MLF approaches, the segmentation algorithm proposed in [102] was used to make MLFs, since its segmentation constraints

Algorithm 4: GMLF approach in case of VAE auto-encoder

Input: a data point $\vec{x} \in R^d$, a $trainedNeuralNet$ returning the class scores given a data point, a $getTrainedVAE$ procedure returning a trained VAE, a relevance propagation algorithm RP returning a relevance vector given: i) a neural network, ii) an input and iii) its class probabilities, a $generateNeuralNetwork$ function returning a neural networks with weights, biases and activation function given as parameters

Output: relevances $\vec{u}$ of each latent variable

1 $\vec{y} \leftarrow M(\vec{x})$;
   a) $\vec{y} \leftarrow TrainedNeuralNet(\vec{x})$;
2 build an auto-encoder $AE \equiv (E_\xi, D_\xi)$;
   a) $(E_\xi, D_\xi) \leftarrow getTrainedVAE()$;
3 $\vec{h} \leftarrow E_\xi(\vec{x})$;
4 define $R : \vec{h} \mapsto \vec{x} - D_\xi(\vec{h})$:
   a) let $W_{res} \in \{0\}^{d \times d}$;
   b) $\vec{r} = \vec{x} - D_\xi(\vec{x})$;
   c) $\vec{b}_{res} \leftarrow \vec{r}$;
   d) define $identity : \vec{a} \mapsto \vec{a}$;
   e) $R \leftarrow generateNeuralNetwork$(weights $= \{W_{res}\}$, biases $= \{\vec{b}_{res}\}$, activation function $= identity$);
5 define $DM_\xi : \vec{h} \mapsto M(D_\xi(\vec{h}) + R(\vec{h}))$:
   a) $DM_\xi \leftarrow$ stackTogether(D_ξ, R, M);
6 $\vec{u} \leftarrow RP(DM_\xi, \vec{h}, \vec{y})$;
7 return $\vec{u}$;

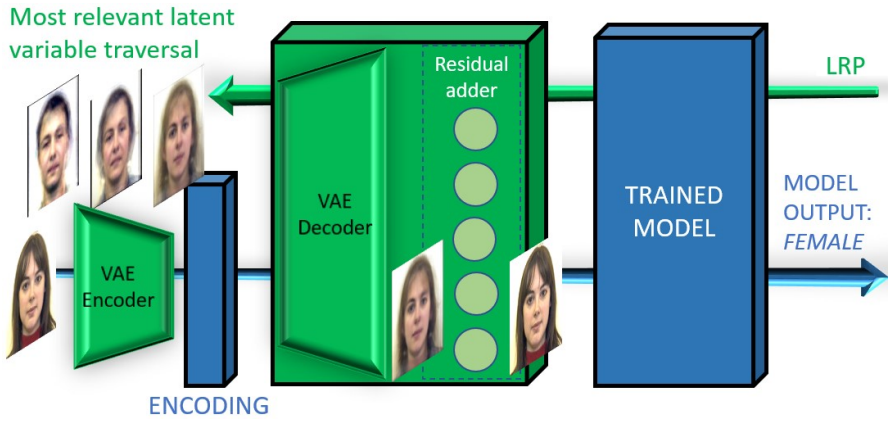


Figure 8.4. A VAE-based MLF framework. The MLF decoder is built as a neural network composed of the VAE decoder module followed by a full-connected layer containing the residual of the input (see text for further details). The initial input encoding is given by the VAE encoder module. The relevance backward algorithm returns the most relevant latent variables.

respect the causality and the location principles reported in Section 8.2.2. However, for the non-hierarchical method, any segmentation algorithm can be used (see for example [25]).

For the Variational Auto-Encoder (VAE) based GMLF approach, a β-VAE [110] is used as MLFs builder, since it results particularly suitable for generating interpretable representations. In all the cases, as image classifier a VGG16 [220] network pre-trained on ImageNet is used. MLF relevances are computed with the LRP algorithm using the $\alpha - \beta$ rule[40].

In Section 8.4 a set of possible explanations of the classifier outputs on image sampled from STL-10 dataset [70] and the Aberdeen data set from University of Stirling (http://pics.psych.stir.ac.uk) is shown. The STL10 data-set is composed of images belonging to 10 different classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck), and the Aberdeen database is composed of images belonging to 2 different classes (Male, Female). Only for the Aberdeen data-set the classifier was fine-tuned using an subset of the whole data-set as training set.

8.3.1 Flat Segmentation approach

For the flat (non-hierarchical) segmentation approach, images from the STL-10 and the Aberdeen data sets are used to generate the classifier outputs and corresponding explanations. For each test image, a set of segments (or superpixels) S are generated using the image segmentation algorithm proposed [102] considering just one level. Therefore, a one-layer neural network decoder as described in Section 8.2.2 was constructed using the segmentation S. The resulting decoder is stacked on the top of the VGG16 model and fed with the "1" vector (see figure 8.3). The relevance of each superpixel/segment was then computed using the LRP algorithm.

8.3.2 Hierarchical Image Segmentation Approach

As for the non-hierarchical segmentation approach, the segmentation algorithm proposed in [102] was used, but in this case, three hierarchically organised levels were considered. Thus, for each test image, 3 different sets of segments (or superpixels) $\{S_i\}_{i=1}^{3}$ related between them in a hierarchical fashion are generated, going from the coarsest ($i = 1$) to the finest ($i = 3$) segmentation level. Next, a hierarchical decoder is made as described in section 8.2.2 and stacked on the classifier (see Figure 8.3). As for the non-hierarchical case, the decoder is then fed with the "1"s vector. Finally, LRP is used to obtain hierarchical explanations as follows: 1) first, at the coarsest level $i = 1$, the most relevant segment $\vec{s_{i}}_{max}$ is selected; 2) then, for each finer level $i > 1$, the segment $\vec{s_{i}}_{max}$ corresponding to the most relevant segment belonging to $\vec{s_{i-1}}_{max}$ is chosen.

8.3.3 Variational auto-encoders

Images from the Aberdeen dataset are used to construct an explanation based on VAE encoding latent variables relevances. The VAE model was trained on an Aberdeen subset using the architecture suggested in [110] for the CelebA dataset. Then, an encoding of 10 latent variables is made using the encoder network for each test image. The resulting encodings were fed to the decoder network stacked on top of the trained VGG16. Next, the LRP algorithm was applied on the decoder top layer to compute the relevance of each latent variable.

8.4 Results

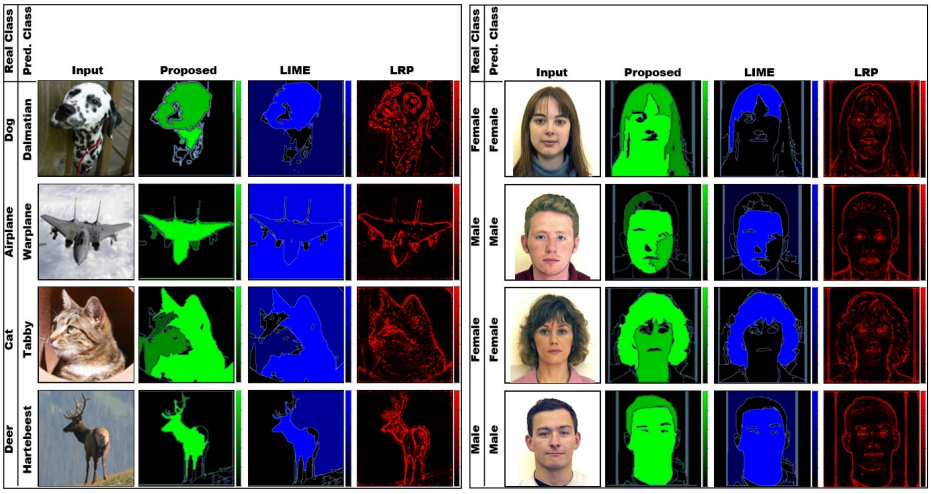
In this section is reported the evaluation assessment of the different realisation of the GMLF framework described in the previous section. For the evaluation both qualitative and quantitative results (see Section 8.4.5) are shown. In particular, in the first part of this section are reported some examples of explanations obtained using flat and hierarchical segmentation-based MLFs, and VAE-based MLFs. Thereafter, is shown an example of explanation using different types of MLFs. Finally, in Section 8.4.5 is reported a quantitative evaluation of the obtained results.

8.4.1 Flat Segmentation

In Figure 8.5 are shown some of the explanations produced for a set of images using the flat (non hierarchical) segmentation-based experimental setup described in Section 8.2.2. The proposed explanations are reported considering the first two more relevant segments according to the method described in Section 8.2.2. For each image, the real class and the assigned class are reported. From a qualitative visual inspection, one can observe that the selected segments seem to play a relevant role for distinguishing the classes.

8.4.2 Hierarchical Image Segmentation

In figures 8.6 and 8.7 is shown a set of explanations using the hierarchical approach described in Section 8.2.2 on images of the STL10 and the Aberdeen datasets. In this case, the hierarchical segmentation organisation is exploited to provide MLF explanations. In particular, for each image, a three layers decoder has been used, obtaining three different image segmentations S_1, cS_2 and S_3, from the coarsest to the finest one, which are hierarchically organised (see Section 8.2.2). For the coarsest segmentation (S_1), the two most relevant segments s_1^1 and s_2^1 are highlighted in the central row. For the image segmentation S_2 the most relevant segment s_1^2 belonging to s_1^1 and the most relevant segment s_2^2 belonging to s_2^1 are highlighted in the upper and the lower row (second column). The same process is made for the image segmentation S_3, where the most relevant segment s_1^3 belonging to s_1^2 and the most relevant segment s_2^3 belonging



(a) STL10 (b) Aberdeen

Figure 8.5. Explanations obtained by GMLF using the flat strategy (second columns), LIME (third columns) and LRP (fourth columns) for VGG16 network responses using images from STL10 (a) and Aberdeen datasets (b). In both (a) and (b), for each input (first columns) the explanation in terms of most relevant segments are reported for the proposed flat approach (second columns) and LIME (third columns). For better clarity, is reported a colormap where only the first two most relevant segments are highlighted both for MLRF and LIME.

to s_2^2 are shown in the third column. From a qualitative perspective, one can note that the proposed approach seems to select relevant segments for distinguishing the classes. Furthermore, the hierarchical organisation provides more clear insights about the input image's parts, contributing to the classifier decision.

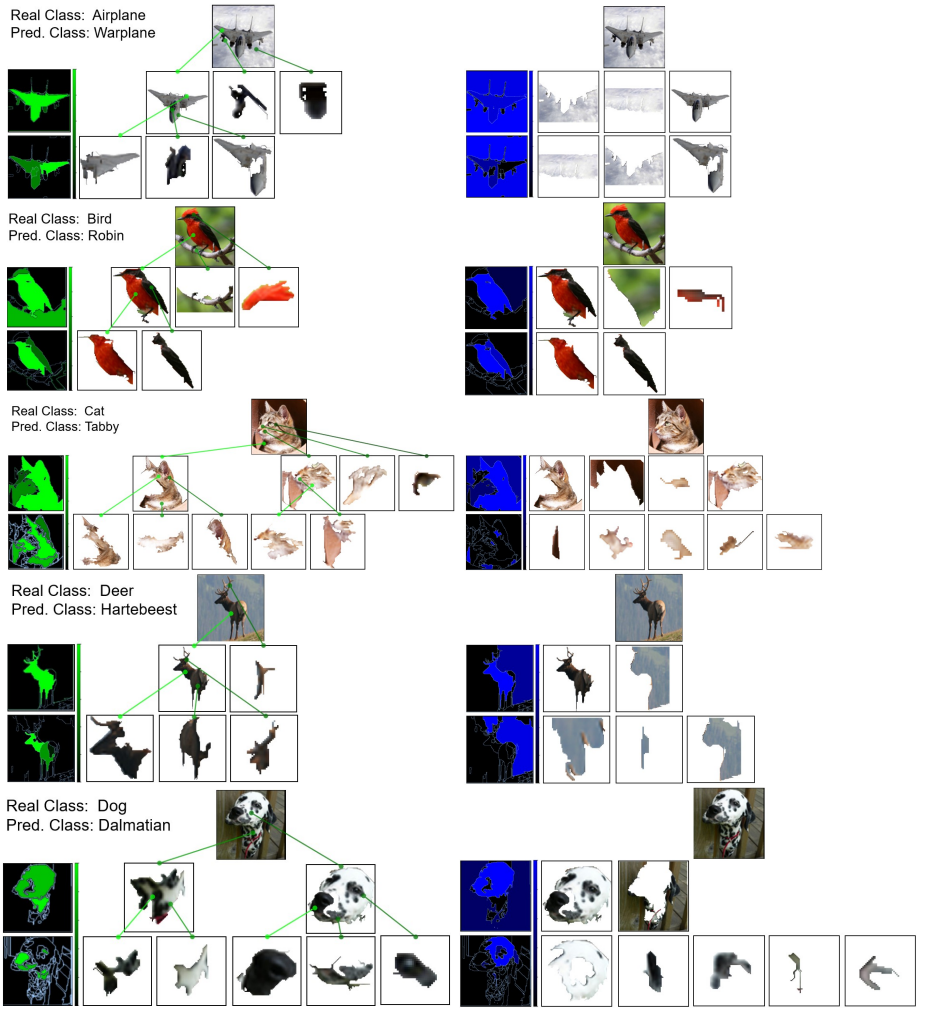
The usefulness of a hierarchical method can also be seen in cases of wrong classifier responses. See, for example, Figure 8.8 where a hierarchical segmentation MLF approach was made on two images wrongly classified: 1) a dog wrongly classified as a poodle although it is evidently of a completely different race, and 2) a cat classified as a bow tie. Inspecting the MLF explanations at different hierarchy scales, it can be seen that, in the dog case, the classifier was misled by the wig (which probably led the classifier toward the poodle class), while, in the other case, the cat head position near the neck of the shirt, while the remaining part of the body is hidden, could be responsible for the wrong classification.

8.4.3 VAE-based MLF explanations

In Figure 8.9 a set of results using the VAE-based experimental setup described in Section 8.3 is shown. For each input, a relevance vector on the latent variable coding is computed. Then, a set of decoded images are generated varying the two most relevant latent variables while fixing the other ones to the original encoding values. One can observe that varying the most relevant latent variables it seems that relevant image properties for the classifier decision are modified such as hair length and style.

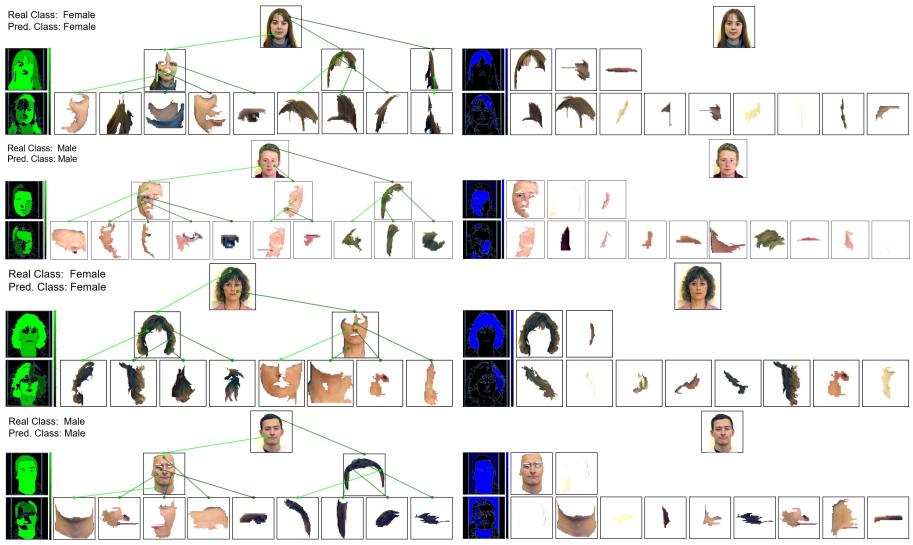
8.4.4 Multiple MLF explanations

For the same classifier input-output, the possibility to provide multiple and different MLF explanations based on the three types of previously mentioned MLFs is shown. In Figure 8.10, for each input, three different types of explanations are shown. In the firs row, an explanation based on MLFs obtained by a flat image segmentation is reported. In the second row, an explanation based on MLFs obtained by an hierarchical segmentation. In the last row, a VAE-based MLF explanation is showed. Notice that the three types of explanations, although based on different MLFs, seem coherent to each other.



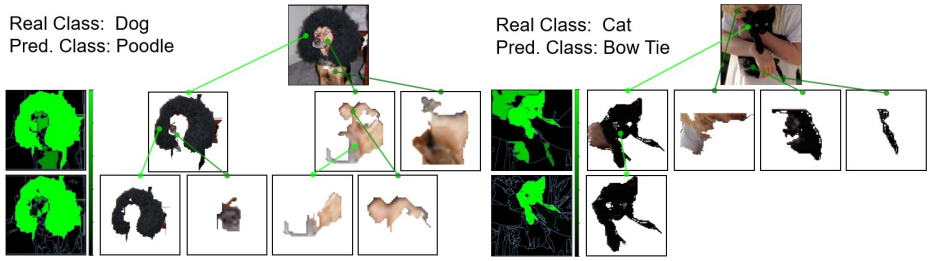
(a) Proposed approach (b) LIME

Figure 8.6. Examples of a two-layer hierarchical explanation on images classified as *warplane, tobby, hartebeest, dalmatian* respectively by VGG16. (a) First column: segment heat map. Left to right: segments sorted in descending relevance order. Top-down: the coarsest (second row) and the finest (third row) hierarchical level. (b) LIME explanation: same input, same segmentation used in (a).



(a) Proposed approach (b) LIME

Figure 8.7. Examples of a two-layer hierarchical explanation on images classified as *Female* and *Male* by VGG16. (a) First column: segment heat map. Left to right: segments sorted in descending relevance order. Top-down: the coarsest (second row) and the finest (third row) hierarchical level. (b) LIME explanation: same input, same segmentation used in (a).



(a) classified as poodle (b) classified as bow tie

Figure 8.8. Results obtained by Hierarchical MLF approach (described in Section 8.2.2) using VGG16 network on STL10 images wrongly classified by the model. (a) A dog wrongly classified as a poodle, although it is evidently of a completely different race. Inspecting the MLF explanations at different hierarchy scales, it can be seen that the classifier was probably misled by the wig (which probably led the classifier toward the poodle class), (b) A cat wrongly classified as a bow tie. Inspecting the MLF explanations at different hierarchy scales, it can be seen that the shape and the position of the cat head near the neck of the shirt, having at the same time the remaining of its body hidden, could be responsible for the wrong class.

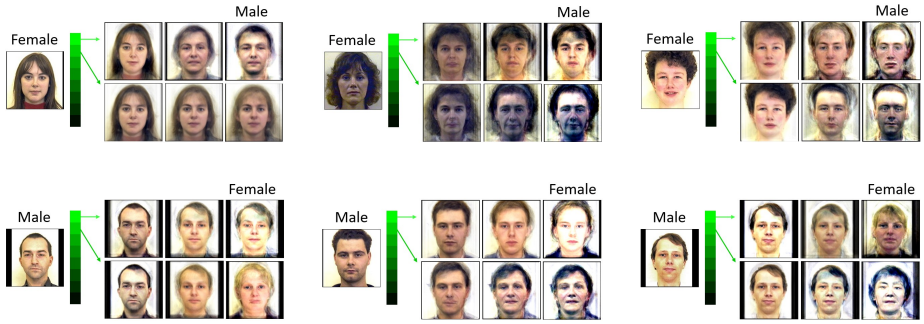


Figure 8.9. Results obtained by VAE MLF approach (described in Section 8.2.3) using a VGG16 network on Aberdeen image dataset. For each image, a VAE is constructed. For each input, the resulting relevance vector on the latent variable is computed. Then, decoded images are generated varying the two most relevant latent variables while fixing the other ones to the original values.

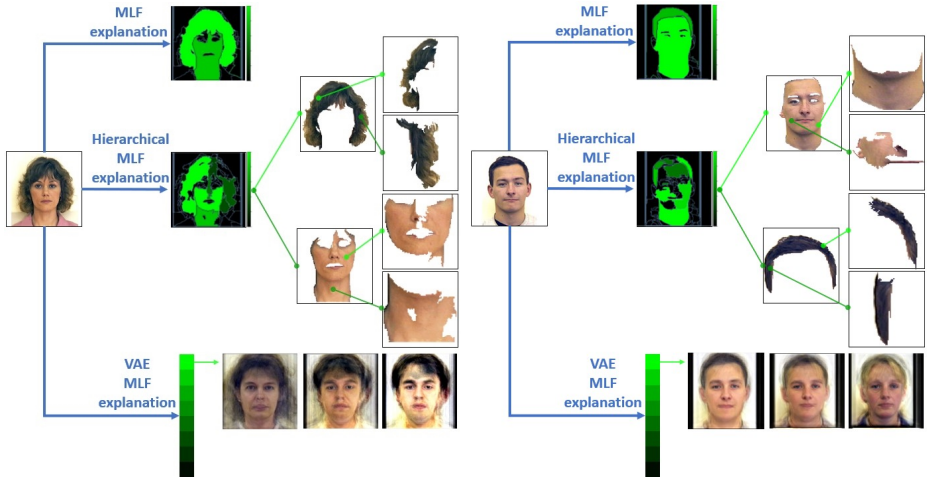


Figure 8.10. For each input, three different types of explanations obtained by GMLF approach are shown. In the first row, an explanation based on a flat image segmentation is reported. In the second row, an explanation based on an hierarchical segmentation. In the last row, a VAE-based MLF explanation is showed.

8.4.5 Quantitative evaluation

A quantitative evaluation is performed adopting the MoRF (Most Relevant First) and AOPC (Area Over Perturbation Curve) [40, 206] curve analysis. In this chapter, MoRF curve is computed following the *region flipping* approach, a generalisation of the *pixel-flipping* measure proposed in [40]. In a nutshell, given an image classification, image regions (in this case segments) are iteratively replaced by random noise and fed to the classifier, following the descending order with respect to the relevance values returned by the explanation method. In this manner, more relevant for the classification output the identified MLFs are, steepest is the curve. Instead, AOPC is computed as:

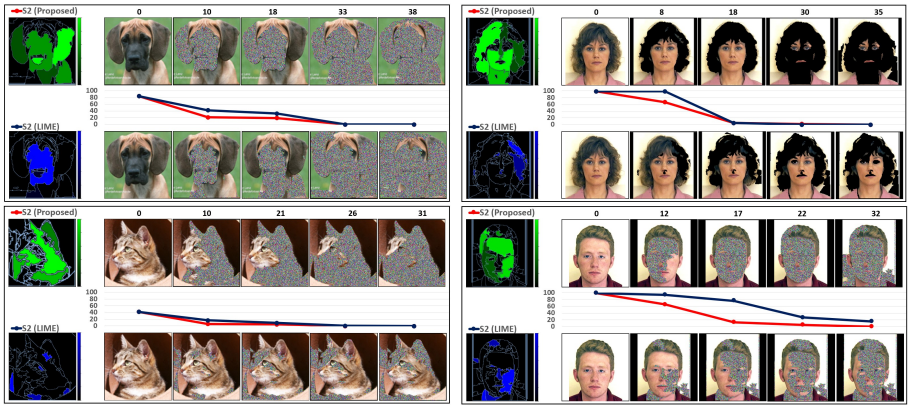
$$AOPC = \frac{1}{L+1} < \sum_{k=0}^{L} f(\vec{x}^{(0)}) - f(\vec{x}^{(k)}) >$$

where L is the total number of perturbation steps, $f(\cdot)$ is the classifier output score, $\vec{x}^{(0)}$ is the original input image, $\vec{x}^{(i)}$ is the input perturbed at step i, and $< \cdot >$ is the average operator over a set of input images. In this manner, more relevant for the classification output the identified MLFs are, greater the AOPC value is.

To evaluate the hierarchical approach with respect to the flat segmentation approach, at each step, MLFs were removed from the inputs exploiting the hierarchy in a topological sort depth-first search based on the descending order's relevances. Therefore, the MLFs of the finest hierarchical layer were considered. MoRF and AOPC are shown in Fig. 8.11 and 8.12. In Fig. 8.11 MoRF curves for some inputs are shown. It is evident that the MLFs selected by the proposed hierarchical approach are more relevant for the produced classification output. This result is confirmed by the average MoRF and average AOPC curves (Fig. 8.12), obtained averaging over the MoRF and AOPC curves of a sample of 100 and 50 random images taken from STL10 and Aberdeen respectively. To make an easy comparison between the proposed methods and summarising the quantitative evaluations, last iteration AOPC values of the proposed methods and LIME are reported in Tables 8.1 and 8.2 for STL 10 and Aberdeen dataset respectively.

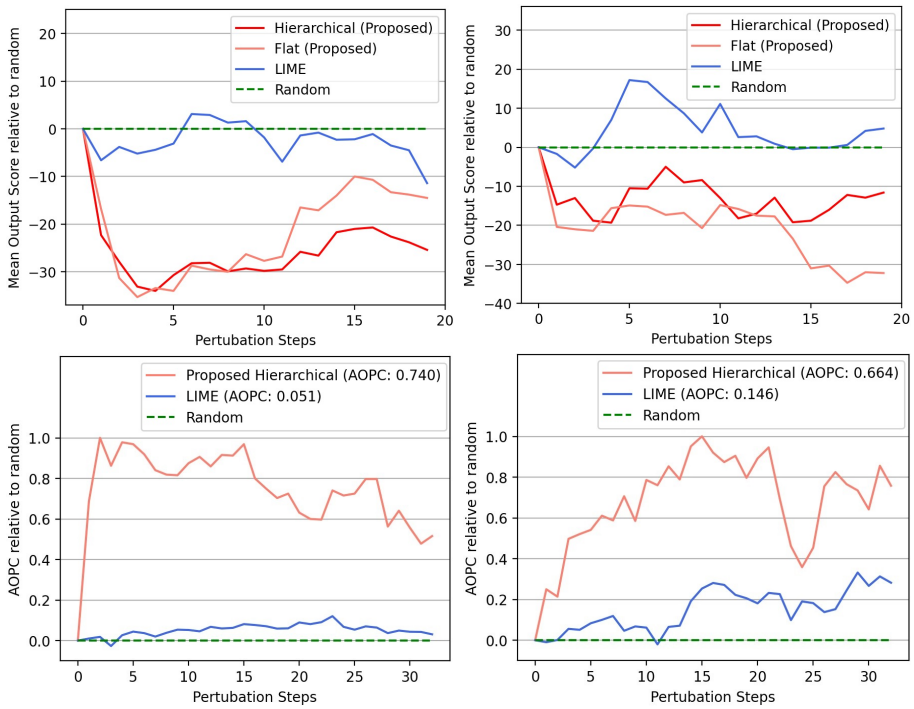
In Fig. 8.14, the same quantitative analysis using the VAE strategy is shown. Examples of MoRF curves using the VAE are shown in Fig. 8.13. As in the hierarchical approach, the latent features are sorted following the descending order returned by the relevance algorithm, and then noised in turn for each perturbation step.

Due to the difference between LIME and VAE MLFs (the former corresponds to superpixels, the latter to latent variables), no comparison with LIME was reported. In my knowledge, no other study reports explanations in terms of latent variables, therefore is not easy to make a qualitative comparison with the existing methods. Differently from perturbing the MLF of a superpixel-based approach where only an image part is substituted by noise, in a variational latent space perturbing a latent variable can lead changing in the whole input image. Therefore, classifiers fed with decoded images generated by different MLF types could return no comparable results, which may not be informative to make comparisons between MoRF curves.



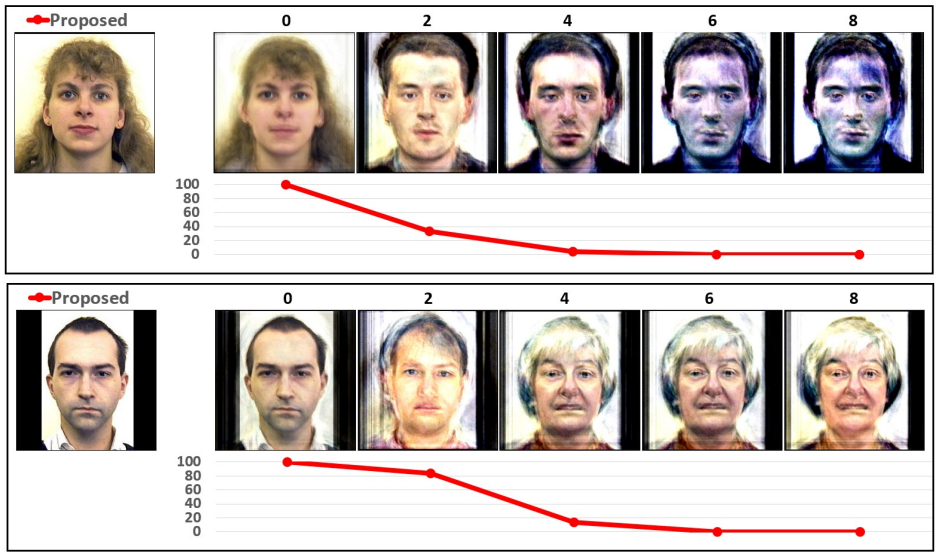
(a) STL10 (b) Aberdeen

Figure 8.11. A quantitative evaluation of the hierarchical GMLF approach on different input images. To evaluate the hierarchical GMLF approach respect to the LIME approach, a most relevant segment analysis is made using MoRF curves. MoRF curves computed with the proposed approach (red) and LIME (blue) using the last layer MLF as segmentation for both methods are shown. At each iteration step, a perturbed input based on the returned explanation is fed to the classifier. On the y axis of the plot, the classification probability (in %) of the original class for each perturbed input. On the x axis, some perturbation steps. For each input image, the figures in the first and the second row show the perturbed inputs fed to the classifier at each perturbation step for the proposed explainer system and the LIME explainer, respectively. More relevant for the classification output the identified MLFs are, steepest the MoRF curve is.



(a) STL10 (b) Aberdeen

Figure 8.12. average MoRF (first row) and AOPC (second row) computed on a sample of 100 and 50 random images sampled from STL10 (first column) and Abardeen (second column) respectively. Both the curves of the proposed hierarchical approach (red) and LIME (blue) are plotted using as baseline the removal of the Middle Level Features from the input images in a random order (green). More relevant for the classification output the identified MLFs are, steepest the MoRF curve is and greater the AOPC value is.



Aberdeen

Figure 8.13. A quantitative evaluation of the VAE GMLF approach on different input images. MoRF curves computed with the proposed approach (red) perturbing the VAE latent variables in the order given by the explainer are shown. At each iteration step, a perturbed input based on the returned explanation is fed to the classifier. On the y axis of the plot, the classification probability (in %) of the original class for each perturbed input. On the x axis, some perturbation steps. For each input image, the figures show the perturbed inputs fed to the classifier at each perturbation step for the proposed explainer system. More relevant for the classification output the identified MLFs are, steepest the MoRF curve is.

Table 8.1. average AOPC of the proposed methods and LIME obtained averaging over the last AOPC perturbation step on a sample of 100 random images taken from STL10 dataset. Flat and hierarchical proposal are compared with LIME, resulting better in both cases. Since the LIME MLFs structure is hardly different from VAE MLFs (the former corresponds to superpixels, the latter to latent variables), the AOPC reported has not to be compared with the other results.

	AOPC
LIME	0.042
Flat (proposed)	0.598
Hierarchical (proposed)	**0.732**
VAE (proposed)	0.595

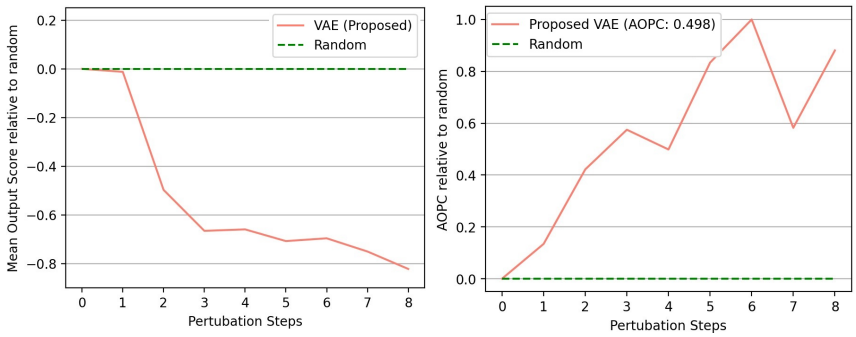
Table 8.2. average AOPC of the proposed methods and LIME obtained averaging over the last AOPC perturbation step values on a sample of 50 random images taken from Aberdeen dataset.

	AOPC
LIME	0.014
Flat (proposed)	0.571
Hierarchical (proposed)	**0.661**

8.5 Conclusion

A framework to generate explanations in terms of middle-level features is proposed in this chapter. With the expression *Middle Level Features* (MLF), (see Section 8, means input features that represent more salient and understandable input properties for a user, such as parts of the input (for example, nose, ears and paw, in case of images of humans) or more abstract input properties (for example, shape, viewpoint, thickness and so on).

This approach can be considered a general framework to obtain humanly understandable explanations insofar as it can be applied to different types of middle-level features as long as an encoder/decoder system is provided (for example image segmentation or latent coding) and an explanation method producing heatmaps can be applied on both the decoder and the ML system whose decision is to be explained (see Section 8.2.1). Consequently, the proposed approach enables one to obtain different types



average MoRF curve average AOPC curve

Figure 8.14. average MoRF (first column) and AOPC (second column) computed on a sample of 50 random images sampled from Aberdeen dataset. The curve proposed with the VAE approach (red) is plotted using as baseline the removal of the Middle Level Features from the input images in a random order (green). More relevant for the classification output the identified MLFs are, steepest the MoRF curve is and greater the AOPC value is.

of explanations in terms of different MLFs for the same pair input/decision of an ML system, that may allow developing XAI solutions able to provide user-centred explanations according to several research directions proposed in literature [204, 152].

The aim is to experimentally tested (see Section 8.3 and 8.4) this approach using three different types of MLFs: flat (non hierarchical) segmentation, hierarchical segmentation and VAE latent coding. Two different datasets were used: STL-10 dataset and the Aberdeen dataset from the University of Stirling.

The results were evaluated from both a qualitative and a quantitative point of view. The quantitative evaluation was obtained using MoRF curves [206].

Chapter 9

XAI: methods in EEG-based systems

Introduction

In this research thesis, the final aim is to experimentally investigate the performances of several well-known eXplainable Artificial (XAI) methods proposed in the literature in the context of Brain-Computer Interface (BCI) problems using EEG input-based Machine Learning (ML) algorithms to evaluate the possibility of alleviating the *Dataset Shift problem*. This is not a trivial issue as, differently from other signals, the non-stationarity of EEG signals makes them hard to analyse.

However, as stated also in chapter 3 one of the main defects of the EEG signal is that its statistical characteristics change over time. This implies that even under the same conditions and for the same task, significantly different signals can be acquired just as time passes. It is important to highlight that this phenomenon can also occur using the same stimuli-reaction (e.g., same emotions with the same stimuli) to the same subject at different times, leading to substantially different EEG signals even for the same subject. This problem is even more present among different subjects, who, given the same stimuli and emotions, can produce very different acquisitions between them. For these reasons, EEG is considered a non-stationary signal [122].

On another side, a sub-field of Artificial Intelligence, eXplainable Ar-

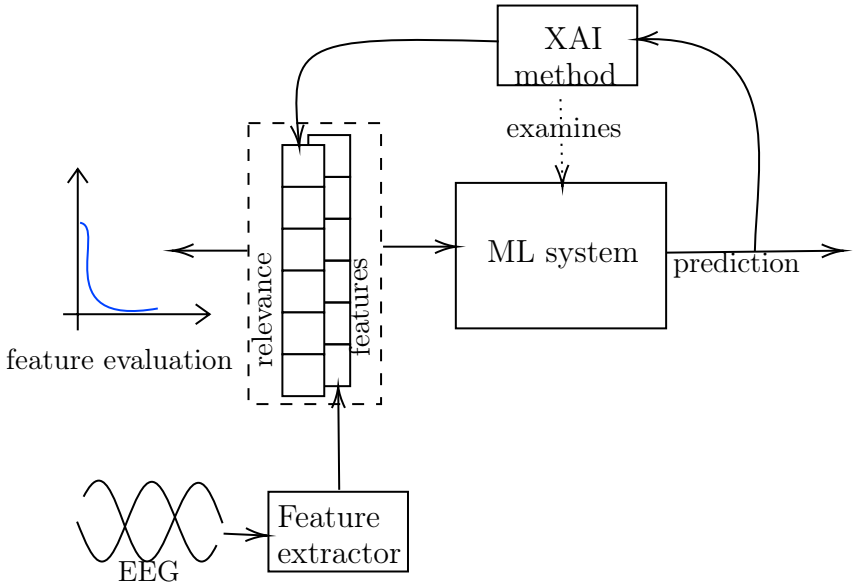


Figure 9.1. A general functional scheme of a Machine Learning (ML) architecture based on XAI methods to select and transform relevant input features with the aim of improving the performance of ML systems in the context of the dataset-shift problem.

tificial Intelligence (XAI), wants to explain the behaviour of AI systems, such as ML ones.

The general idea of this thesis is that outputs' explanations of a trained ML model on given inputs can help the setup of new models able to overcome/mitigate the dataset shift problem, in general, and to generalise across subjects/sessions in case of EEG signals, in particular.

More specifically, in this thesis, the goal is to focus on how several well-known XAI methods proposed in literature behave in explaining decisions made by an ML system based on EEG input features (Fig. 9.1). Notice that several current XAI methods are usually tested on datasets, such as image and text recognition datasets [202, 26], where the domain shift problem is slight or not present. Therefore, this thesis is a first step toward a long term goal consisting in exploiting explanations made by XAI methods to locate and transform the main characteristics of the input for each given output, and to build ML systems able to generalise toward different data coming from different probability distributions (in this context, sessions and subjects). To this end, in this thesis, the aim is to evaluate and analyse the explanations produced by a set of well-known XAI methods on an ML system trained on data taken from SEED [262], a public EEG dataset for an emotion classification task. The results obtained show, on oneside, that only some well-known XAI methods produce reliable explanations in the EEG domain in the analysed task. On another side, it is shown that the relevant components found in the training data can only be partially used on data acquired outside of the training stage. Notably, many relevant components found in the training data are still relevant across the sessions.

The chapter is organised as follows: In Section 9.1, a brief description of the related works is reported. In Section 9.2 the proposed evaluation framework is presented. In Section 9.3 the obtained results are discussed. Finally, in Section 9.4 is devoted to final remarks and future developments.

9.1 Related works

In general, Modern ML approaches, as Deep learning, are characterised by a lack of transparency of their internal mechanisms, making it not easy for the AI scientist to understand the real reasons behind the inner be-

haviours. In this case, the relationships of the classified emotion with the EEG input are often challenging to understand. In the EEG-based applications, works based on simple features selection strategies to choose the best EEG features are widely proposed in the literature, such as [243, 263]. These studies, however, are based on standard feature selection methods, without exploiting information given by XAI methods. XAI is a branch of AI concerned to "explain" ML behaviours. This is made providing methods for generating possible explanations of the model's outputs. XAI methods are gaining prominence in explaining several classification systems based on several inputs, such as images [202, 20], natural language processing [193], clinical decision support systems [211], and so on. To the best of my knowledge, however, the number of research works which attempt to improve the performance of ML models on the basis of XAI's methods is enough limited, especially in the context of bio-signal classification problems. For example, in [141, 214] feature selection procedures are carried out on biomedical data leveraging on Correlation-based Feature Selection and Chaotic Spider Monkey Optimization methods. In [117] the authors propose to use an occlusion sensitivity analysis strategy [250] to locate the most relevant cortical areas in a motor imagery task. In [198] the use of XAI methods to interpret the answer of Epilepsy Detection systems is discussed.

9.2 Methods

Taking in mind that the aim is to use the XAI method to alleviate the dataset shift problem in the BCI context, the target is to conduct a series of experiments having the following goals: 1) testing the capability of the selected XAI methods to find relevant components for this specific signal; 2) verifying how much relevant components are dependent on the single sample of the dataset where the relevance are computed; 3) how much relevant components can be considered shared among samples of the same session, and finally 4) how much relevant components can be considered shared between samples of two different sessions, where the data shift problem is typically present.

In the remaining of this section is reported the investigated XAI methods, the used data and model descriptions. Finally experimental assess-

ment and the evaluation strategy adopted are reported.

9.2.1 Investigated XAI Methods

In this thesis, the goal is to analyse XAI methods proposing explanations in terms of relevance of the input components on the output returned by a given classifier. More in detail, the following XAI methods are investigated: Saliency [218], Guided Backpropagation [225], Layer-wise Relevance Propagation (LRP) [41], Integrated Gradients [226], and DeepLIFT [216]. A description of these XAI methods can be found in section 7.2.

9.2.2 Dataset

The SEED dataset consists of EEG signals recorded from 15 subjects stimulated by 15 film clips carefully chosen to induce negative, neutral and positive emotions. Each film clip has a duration of approximately 4 minutes. Three sessions of 15 trials were collected for each subject. EEG signals were recorded in 62 channels using the ESI Neuroscan System[1]. During the experiments, the aim is to consider the pre-computed differential entropy (DE) features smoothed by linear dynamic systems (LDS) for each second, in each channel, over the following five bands: delta (1–3 Hz); theta (4–7 Hz); alpha (8–13 Hz); beta (14–30 Hz); gamma (31–50 Hz).

In this thesis, the relevant components of an EEG signal can be considered taking into account three different aspects of the signal: i) considering each single feature composing the input, ii) considering each single band composing the EEG signal, that are alpha, beta, theta, and delta, and iii) considering each single channel/electrode from which the input EEG signal was acquired. Cases ii) and iii) can be viewed as different aggregations of fixed features of the EEG signals. In the following of this thesis, the term "components" is referred to generically where it is not necessary to specify whether one is talking about features, bands or channels.

9.2.3 Experimental assessment

To achieve the goals defined at the beginning of this section, the following experiments are made: firstly, to evaluate the capability of the selected

[1] https://compumedicsneuroscan.com

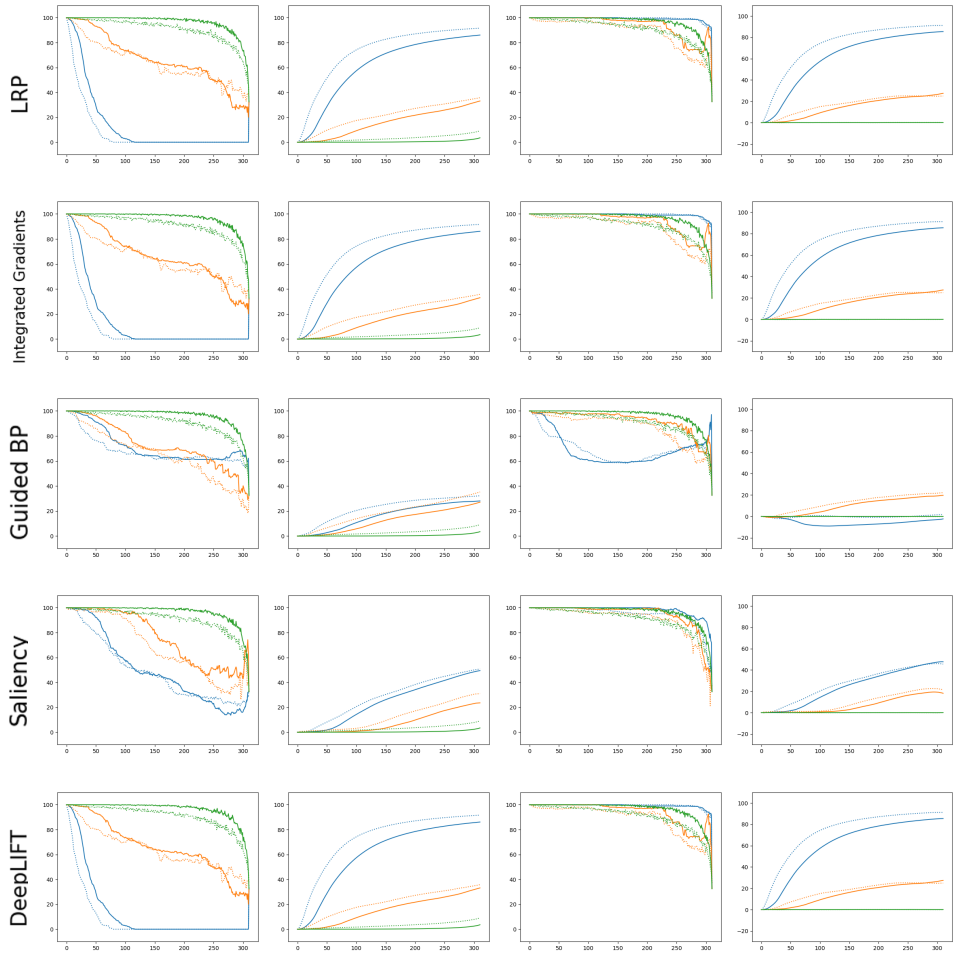


Figure 9.2. MoRF (first column), AOPC (second column), LeRF (third column), and ABPC (fourth column) curves using the tested XAI methods are reported for both intra-session (solid line) and inter-session (dotted lines) considering features as signal components. Results scoring the input components using effective relevance (blue lines) and averaged relevance computed on training data (orange lines) are reported for each case and compared with a random component scoring (green lines). On the x axis and y axis are reported the iteration step in the curve generation and the accuracy level reached, respectively.

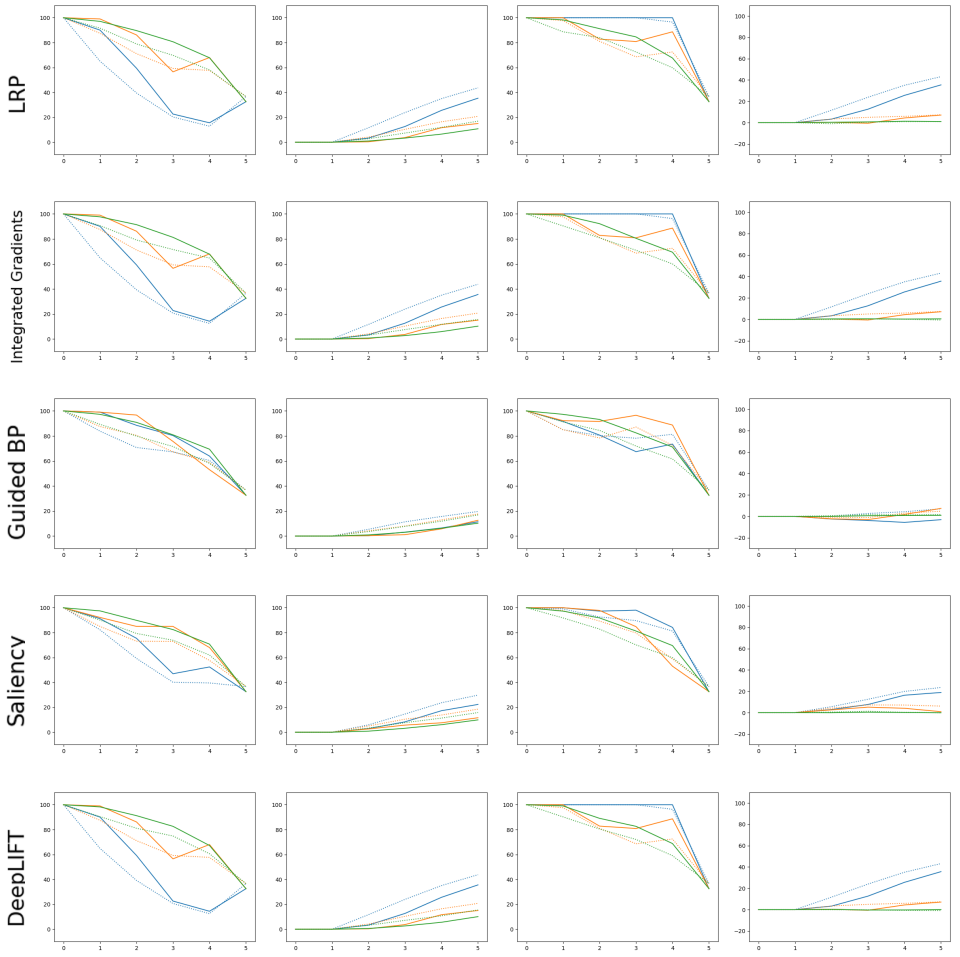


Figure 9.3. MoRF (first column), AOPC (second column), LeRF (third column), and ABPC (fourth column) curves using the tested XAI methods are reported for both intra-session (solid line) and inter-session (dotted lines) considering delta, theta, alpha, beta, gamma EEG bands as signal components. Results scoring the input components using effective relevance (blue lines) and averaged relevance computed on training data (orange lines) are reported for each case and compared with a random component scoring (green lines). On the x axis and y axis are reported the iteration step in the curve generation and the accuracy level reached, respectively.

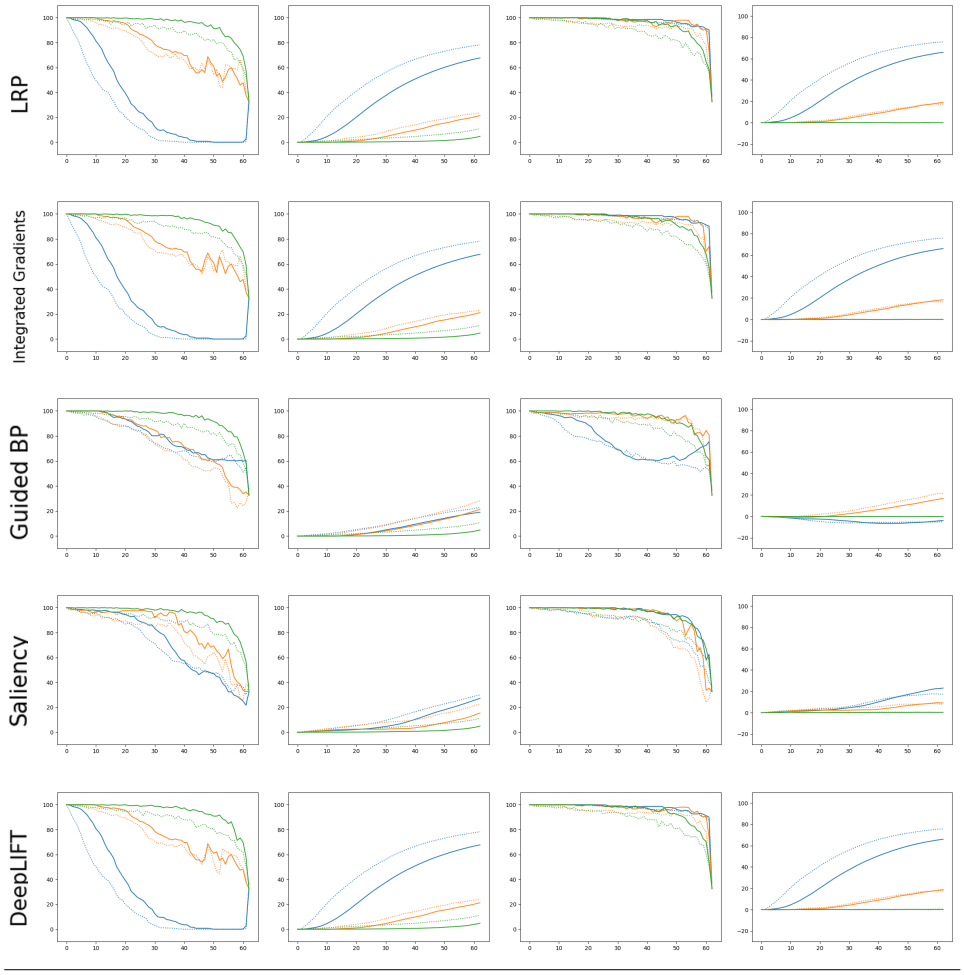


Figure 9.4. MoRF (first column), AOPC (second column), LeRF (third column), and ABPC (fourth column) curves using the tested XAI methods are reported for both intra-session (solid line) and inter-session (dotted lines) considering the acquisition electrodes as signal components. Results scoring the input components using effective relevance (blue lines) and averaged relevance computed on training data (orange lines) are reported for each case and compared with a random component scoring (green lines). On the x axis and y axis are reported the iteration step in the curve generation and the accuracy level reached, respectively.

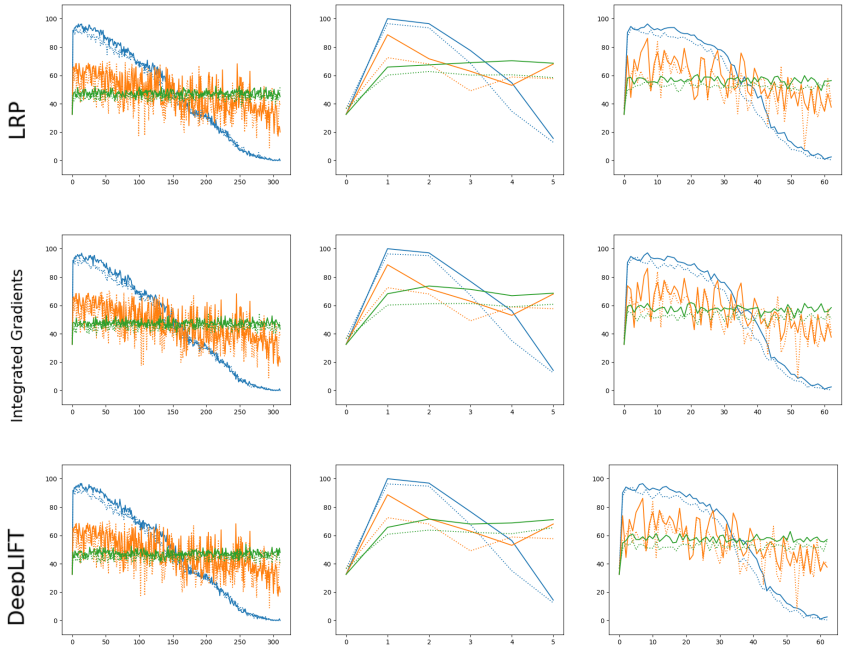


Figure 9.5. A first analysis of the discriminative power of the components alone. Signals composed of only one component following the relevance order given by the Explainer are fed to the ML system in an iterative manner. Results are reported for both intra-session (solid line) and inter-session (dotted lines) considering features (first column), bands (second column), and electrodes (third column) as signal components. Results scoring the input components using effective relevance (blue lines) and averaged relevance computed on training data (orange lines) are reported for each case and compared with a random component scoring (green lines).

XAI methods to find relevant components, explanations of model responses on data from the same session as the training data were analysed; then, in order to assess how relevant components could be considered shared between samples from the same session, explanations of model responses on data from a session other than the training session were analysed. Finally, to evaluate if relevant components can be considered shared between samples of two different sessions and how much relevant components are dependent on the single data sample where the relevance are computed, the components' average relevance of data coming from the training session are used as sorting score and select the components belonging to another session.

Summarising, the following cases are considered: i) intra-session case: given a model C trained on data coming from a session s_{tr}, explanations of the responses on input data belonging to the same session s_{tr} are built. ii) inter-session case: given a model C trained on data coming from a session s_{tr}, explanation of responses on inputs belonging to a sessions s_{te} different from s_{tr} are built. Each of these cases can be in turn evaluated considering two different relevance: a) real relevance: it is assumed that it is possible to compute the relevance of the input, since the classification output is known; b) presumed relevance: it is assumed that the relevance of the input is not available, because it is outside the training phase. In this case, the average of the relevance of the same components obtained on the training data is used as the component relevance.

9.2.4 Evaluation

For each case, the goal is to investigate the explanations returned by the XAI method in order to analyse if the explanations built can correctly identify the impact that they have on classification performance. Considering in terms of MLF (chapter 8): i) each input characteristic, MLFs are not used ii) each electrode as MLF and iii) each frequency band as MLF. To this aim, the relevance for each feature is considered the relevance score returned by the XAI method, for each electrode the mean relevance score of all the feature belonging to the electrode, and for each frequency bands the mean average score of all the features belonging to the frequency band. the relevance score returned by the XAI method is considered for each characteristic, for each electrode the average relevance score of all characteristics

belonging to the electrode and for each frequency band the average score of all characteristics belonging to the frequency band. Therefore, the following evaluation strategies are then adopted and repeated considering features, electrodes, and frequency bands as EEG components in turn: a) analysis of the MoRF (Most Relevant First) curve, proposed in [41, 206]. In case of evaluating the components relevance returned by the explanation method, the MoRF curve can be computed as follows: given a classifier, an input EEG signal $\mathbf{x}$ and the respective classification output $C(\mathbf{x})$, the EEG components are iteratively replaced by zeros, following the descending order with respect to the relevance values returned by the explanation method. In other words, performances were analysed by removing (i.e. setting to zero) components in a decreasing order of impact on the predictions supplied by the explanation. In this way, the expected curve is such that more relevant the identified components are for the classification output, steepest is the curve. Furthermore, the change in the AOPC (Area Over Perturbation Curve) value is reported for each MoRF iteration. AOPC is computed as

$$AOPC = \frac{1}{K+1}\langle \sum_{k=0}^{K} C(\mathbf{x}^{(0)}) - C(\mathbf{x}^{(k)}))\rangle$$

where K is the total number of iterations, $\mathbf{x}^{(0)}$ is the original input, $\mathbf{x}^{(k)}$ is the input at the iteration k, and $\langle\cdot\rangle$ is the average operator over a set of inputs. MoRFs and AOPCs are reported also considering channels and bands as characteristics to analyse.

b) the analysis of the LeRF (Least Relevant First) curve, proposed in [206]. Differently from the MoRF curve, in this case the EEG components are iteratively removed following the ascending order with respect to the relevance values returned by the explanation method. In the resulting curve, the classification output is expected should be very close to the original value when the less relevant components are removed (corresponding to the first iterations), dropping quickly to zero as the process goes toward the remotion of relevant elements. While the MoRFs report how much the classifier output is destroyed removing highly relevant components, LeRFs report how much the least relevant components leave the output intact. These indications can be combined in the ABPC (Area Between Pertur-

bation Curves, [206]) quantity, defined as:

$$ABPC = \frac{1}{K+1}\langle\sum_{k=0}^{K} C(\mathbf{x}_{MoRF}^{(k)}) - C(\mathbf{x}_{LeRF}^{(k)}))\rangle$$

where $\mathbf{x}_{MoRF}^{(k)}$, $\mathbf{x}_{LeRF}^{(k)}$ are the values of the MoRF and LeRF values obtained at the k-th iteration step. ABPC is an indicator of how good the XAI method is. The larger the ABPC value, the better the XAI method. LeRFs and ABPCs are reported also for channels and bands analysis.

c) an analysis of the discriminative power of each component alone is made. Signals composed of only one component following the relevance order given by the XAI method are fed to the ML system in an iterative manner, and the relative performance curves are plotted.

All the experiments were carried out only on correctly classified samples.

9.2.5 Classification model

The XAI methods are evaluated on a feed-forward fully connected multi layered neural networks. Hyperparameters were tuned through bayesian optimisation [222]: the number of layers was constrained to a maximum of 3; for each layer, the number of nodes was searched in the space $\{2^n | n \in \{4, 5, ..., 10\}\}$ having the ReLU as activation function. Each experiment was run having early stopping as convergence criterion with 20 epochs of patience. The 10 % of the training set was extracted using stratified sampling [179] on class labels and considered as validation set. Network optimisation was performed using Adam optimiser [129], whose learning rate that was searched in the space $\{0.1, 0.01, ..., 0.0001\}$.

As a result from the model selection stage, the best setting consisted in ANN having 3 layers with 128, 256 and 128 neurons respectively. The learning rate was set to 0.01, and reduced to its 10 % whenever the loss on validation set plateaus for 10 consecutive epochs.

9.3 Results & discussions

Since the behaviour of the explored XAI methods resulted in being similar across all the subjects, only the results obtained on just one subject are reported. In Fig. 9.2, 9.3, and 9.4 MoRF and LeRF curves using the tested XAI methods are reported for both intra-session and inter-session cases, considering as components to remove at each step features (Fig. 9.2), bands (Fig. 9.3), and channels (Fig. 9.4), respectively. Results related to the intra-session cases are reported with solid lines, while those regarding the inter-session case are marked with dotted lines. On the x axis and y axis are reported the iteration step in the curve generation and the accuracy level reached, respectively. With blue lines, results scoring the input components using effective relevance are reported; with orange lines, results scoring the components using averaged relevance computed on training data are reported; with green lines, results related to random choice.

All the curves were compared with the random curve obtained by removing the components in random order. Several interesting points can be highlighted:

1) In all the cases, LRP, IG and Deep LIFT resulted in being more reliable XAI methods with respect to Saliency and Guided BP. Indeed, MoRF curves of LRP, IG and Deep LIFT have high slopes, however similar to each other, differently from Saliency and Guided BP. In particular, the latter is the only method among those tested whose explanations do not always seem to capture the relevant components, especially in the case of intra-session. These considerations seem consistent with what is reported in LeRF, AOPC, and ABPC.

2) counterintuitively, in almost all the cases, explanations built in inter-session cases seem to be more reliable with respect to intra-session cases. This behaviour can be explained by a more significant "robustness" of the trained classifier toward data from the same training session. Instead, data coming from different sessions leads the classifier toward more borderline class scores, and minimum perturbation of the input data can lead to different classes, influencing the final performance.

3) Although the best XAI methods can locate relevant features/channels/bands for each input data sample, they don't seem able to locate a

set of relevant components for all the samples. In other words, the examined XAI methods fail to "generalise" to a set of general features/channels/bands relevant to the most significant part of the possible inputs. Indeed, removing the components following the average relevance (obtained in the training stage) in reverse order (MoRF orange curves) does not lead to a steep drop in performance, as in the other case (MORF blue curves). Even in some cases, such as using bands as a component to assign the relevance (Fig. 9.3), the obtained curves overlap with the random ones, highlighting that removing bands in random order is almost the same that following the relevance assigned by the XAI method. This is confirmed by the other evaluation metrics adopted, i.e. MeRF, AOPC and ABPC curves.

In Fig. 9.5 a first analysis of the discriminative power of the components alone is made. Signals composed of only one component following the relevance order given by the XAI method are iteratively fed to the ML system. The analysis is limited to only the best XAI methods identified in the previous phase: DeepLIFT, IG and LRP. From the obtained results, it is interesting to notice that the components considered most relevant for each sample fed to the classifier are enough to reach high performances. However, considering the average relevance detected during the training stage, the best components do not seem to lead toward similar performance, although they are still better than a random choice.

9.4 Conclusion

In this chapter, the performances of several XAI methods proposed in the literature in the context of Brain-Computer Interface (BCI) problems using EEG input-based Machine Learning (ML) algorithms are experimentally evaluated. The focus was on how much the relevant components selected by XAI methods be shared between different samples of the same dataset (in this case, same session) or samples of different datasets (in this case, different sessions). The final results show that the components considered most relevant for each sample fed to the classifier are enough to achieve high performances. However, the components detected considering the best average relevance during the training stage do not seem to lead toward performance returned by components scored according to their effective relevance returned by the XAI method.

Chapter 10

Conclusions

This thesis proposed several Machine Learning and eXplainable Artificial Intelligence methods to improve classification in EEG-based Brain-Computer Interface systems in terms of accuracy.
In the initial chapters, an overview was given on ML, chapter 2, highlighting that the dataset shift problem can lead to a decrease in performance; in chapter 3 the main properties of EEG signals were described.
In the part of the thesis dedicated to BCI, an introduction on the BCIs was made in chapter 4. Subsequently, two different BCI system problems were analysed.

In chapter 5, in a passive BCI, the experimental results showed how cognitive and emotional engagement enables the monitoring of stress levels and can help the automated rehabilitation platform with useful information to better adapt to the user's needs. In the described case, oversampling methods (such as KMeansSMOTE), standard ML algorithms (k-NN, SVM and ANN) and the combined use of feature extraction methods (CSP) enabled good performance on highly unbalanced datasets by class.

In chapter 6, regarding an example of an active BCI paradigm, four different ML-based algorithms were implemented to improve SSVEP classification, in terms of classification accuracy and temporal response. The first two ones, used on Augmented Reality-based datasets both in extracted feature format considering standard ML models and in raw format applying Deep Neural Networks, showed a significant performance improvement over the established CCA-based algorithm. The other two ones proposed

algorithms, custom ANN and EEGNet with Domain Adaptation methods, were tested on the Benchmark dataset. In particular, the combined use of DNNs and DA methods showed an improvement in performance even when using signal segmentations of a few seconds, an important component for an online approach. These results show a significant improvement over traditional state-of-the-art algorithms.

The second part of this thesis was devoted to XAI. In chapter 7, the main XAI methods in the literature were analysed.

In chapter 8 a framework to generate explanations in terms of Middle-Level Features for an image classification task is proposed. The use of middle-level features is motivated by the need to decrease the human interpretative burden in artificial intelligence explanation systems. The aim was to test this approach experimentally using three different types of MLF: flat (non-hierarchical) segmentation, hierarchical segmentation and latent VAE coding. The results are encouraging, both under the qualitative point of view, giving easily human interpretable explanations, and the quantitative point of view, giving comparable performances to the well known XAI method LIME. Furthermore, it is proved that a hierarchical approach can provide, in several cases, clear explanations about the reason behind classification behaviours.

In chapter 9, the performances of several XAI methods proposed in the literature in the context of BCI problems using EEG input-based ML algorithms are experimentally evaluated. Results show that many relevant components found by XAI methods are shared across the sessions and can be used to build a system able to generalise better. However, relevant components of the input signal also appear to be highly dependent on the input itself.

This thesis is the first step toward developing a BCI system able to exploit XAI methods to alleviate the dataset shift problem. However, in this thesis, only data belonging to different sessions but acquired from the same subjects are taken into account. In future work, the aim is to analyse the behaviour of XAI methods with inter-subject classifiers.

Several benefits can be obtained in the EEG-based BCI applications by the proposed solution. For example, a BCI system can work across different subjects without retraining the model on each new unseen subject, leading toward a subject-independent model. Furthermore, a better understanding

of the relationships between the system inputs and outputs provided by XAI explanations can lead to the developing and producing more effective EEG acquisition devices.

Bibliography

[1] Emotiv Epoc+ technical specifications. https://emotiv.gitbook.io/epoc-user-manual/introduction-1/technical_specifications.

[2] Society for Research in Child Development Ethical Principles and Standards for Developmental Scientists, 2021.

[3] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[4] Sarah N Abdulkader, Ayman Atia, and Mostafa-Sami M Mostafa. Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, 16(2):213–230, 2015.

[5] Reza Abiri, Soheil Borhani, Eric W Sellers, Yang Jiang, and Xiaopeng Zhao. A comprehensive review of eeg-based brain–computer interface paradigms. *Journal of neural engineering*, 16(1):011001, 2019.

[6] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[7] Arjun Akula, Shuai Wang, and Song-Chun Zhu. Cocox: Generating conceptual and counterfactual explanations via fault-lines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2594–2601, 2020.

[8] Ahmed Y Al Hammadi, Chan Yeob Yeun, Ernesto Damiani, Paul D Yoo, Jiankun Hu, Hyun Ku Yeun, and Man-Sung Yim. Explainable artificial intelligence to evaluate industrial internal security using eeg signals in iot framework. *Ad Hoc Networks*, 123:102641, 2021.

[9] Fares Al-Shargie, Usman Tariq, Omnia Hassanin, Hasan Mir, Fabio Babiloni, and Hasan Al-Nashash. Brain connectivity analysis under semantic vigilance and enhanced mental states. *Brain sciences*, 9(12):363, 2019.

[10] Brendan Z Allison, Stephen Dunne, Robert Leeb, José Del R Millán, and Anton Nijholt. *Towards practical brain-computer interfaces: bridging the gap from research to real-world applications*. Springer Science & Business Media, 2012.

[11] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[12] Antonio Andriella, Carme Torras, and Guillem Alenya. Cognitive system framework for brain-training exercise based on human-robot interaction. *Cognitive Computation*, 12(4):793–810, 2020.

[13] Leopoldo Angrisani, Pasquale Arpaia, Antonio Esposito, Ludovica Gargiulo, Angela Natalizio, Giovanna Mastrati, Nicola Moccaldi, and Marco Parvis. Passive and active brain-computer interfaces for rehabilitation in health 4.0. *Measurement: Sensors*, 18:100246, 2021.

[14] Leopoldo Angrisani, Pasquale Arpaia, Antonio Esposito, and Nicola Moccaldi. A wearable brain–computer interface instrument for augmented reality-based inspection in industry 4.0. *IEEE Transactions on Instrumentation and Measurement*, 69(4):1530–1539, 2019.

[15] Irshad Ahmad Ansari, Rajesh Singla, and Munendra Singh. SSVEP and ANN based optimal speller design for brain computer interface. *Computational Science and Techniques*, 2(2):338–349, 2015.

[16] A. Apicella, F. Isgro, R. Prevete, A. Sorrentino, and G. Tamburrini. Explaining classification systems using sparse dictionaries. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Special Session on Societal Issues in Machine Learning: When Learning from Data is Not Enough*, Bruges, Belgium, 2019.

[17] A. Apicella, F. Isgrò, R. Prevete, and G. Tamburrini. Contrastive explanations to classification systems using sparse dictionaries. In *International Conference on Image Analysis and Processing*, pages 207–218. Springer, Cham, 2019.

[18] A Apicella, F Isgro, R Prevete, and G Tamburrini. Middle-level features for the explanation of classification systems by sparse dictionary methods. *International Journal of Neural Systems*, 30(08):2050040, 2020.

[19] A. Apicella, F. Isgro, R. Prevete, G. Tamburrini, and A. Vietri. Sparse dictionaries for the explanation of classification systems. In *PIE*, page 009, Rome, Italy, 2019.

[20] A. Apicella, F. Isgrò, R. Prevete, A. Sorrentino, and G. Tamburrini. Explaining classification systems using sparse dictionaries. *ESANN 2019 - Proceedings, 27th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, page 495 – 500, 2019.

[21] Andrea Apicella, Pasquale Arpaia, Egidio De Benedetto, Nicola Donato, Luigi Duraccio, Salvatore Giugliano, and Roberto Prevete. Enhancement of ssveps classification in bci-based wearable instrumentation through machine learning techniques. *IEEE Sensors Journal*, 22(9):9087–9094, 2022.

[22] Andrea Apicella, Pasquale Arpaia, Mirco Frosolone, Giovanni Improta, Nicola Moccaldi, and Andrea Pollastro. Eeg-based measurement system for monitoring student engagement in learning 4.0. *Scientific Reports*, 12(1):1–13, 2022.

[23] Andrea Apicella, Pasquale Arpaia, Salvatore Giugliano, Giovanna Mastrati, and Nicola Moccaldi. High-wearable eeg-based transducer for engagement detection in pediatric rehabilitation. *Brain-Computer Interfaces*, 9(3):129–139, 2022.

[24] Andrea Apicella, Salvatore Giugliano, Francesco Isgrò, and Roberto Prevete. Explanations in terms of hierarchically organised middle level features. In *XAI.it - 2021 Italian Workshop on Explainable Artificial Intelligence, CEUR Workshop Proceedings*, 2021.

[25] Andrea Apicella, Salvatore Giugliano, Francesco Isgrò, and Roberto Prevete. A general approach to compute the relevance of middle-level input features. In *Pattern Recognition. ICPR International Workshops and Challenges*, pages 189–203, Cham, 2021. Springer International Publishing.

[26] Andrea Apicella, Salvatore Giugliano, Francesco Isgrò, and Roberto Prevete. Exploiting auto-encoders and segmentation methods for middle-level explanations of image classification systems. *Knowledge-Based Systems*, 255:109725, 2022.

[27] Andrea Apicella, Salvatore Giugliano, Francesco Isgrò, and Roberto Prevete. Exploiting auto-encoders and segmentation methods for middle-level explanations of image classification systems. *Knowledge-Based Systems*, 255:109725, 2022.

[28] Andrea Apicella, Francesco Isgrò, and Roberto Prevete. A simple and efficient architecture for trainable activation functions. *Neurocomputing*, 370:1–15, 2019.

[29] Bruno Apolloni, Ashish Ghosh, Ferda Alpaslan, and Srikanta Patnaik. *Machine learning and robot perception*, volume 7. Springer Science & Business Media, 2005.

[30] P. Arpaia, N. Moccaldi, R. Prevete, I. Sannino, and A. Tedesco. A wearable EEG instrument for real-time frontal asymmetry monitoring in worker stress analysis. *IEEE Transactions on Instrumentation and Measurement*, 69(10):8335–8343, 2020.

[31] Pasquale Arpaia, Luca Callegaro, Alessandro Cultrera, Antonio Esposito, and Massimo Ortolano. Metrological characterization of consumer-grade equipment for wearable brain-computer interfaces and extended reality. *IEEE Transactions on Instrumentation and Measurement*, pages 1–1, 2021.

[32] Pasquale Arpaia, Sabatina Criscuolo, Egidio De Benedetto, Nicola Donato, and Luigi Duraccio. A wearable ar-based bci for robot control in adhd treatment: Preliminary evaluation of adherence to therapy. In *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pages 321–324. IEEE, 2021.

[33] Pasquale Arpaia, Egidio De Benedetto, Lucio De Paolis, Giovanni D'Errico, Nicola Donato, and Luigi Duraccio. Highly wearable SSVEP-based BCI: Performance comparison of augmented reality solutions for the flickering stimuli rendering. *Measurement: Sensors*, 18:100305, 2021.

[34] Pasquale Arpaia, Egidio De Benedetto, Concetta Anna Dodaro, Luigi Duraccio, and Giuseppe Servillo. Metrology-based design of a wearable augmented reality system for monitoring patient's vitals in real time. *IEEE Sensors Journal*, 21(9):11176–11183, 2021.

[35] Pasquale Arpaia, Egidio De Benedetto, and Luigi Duraccio. Design, implementation, and metrological characterization of a wearable, integrated AR-BCI hands-free system for health 4.0 monitoring. *Measurement*, 177:109280, 2021.

[36] Pasquale Arpaia, Luigi Duraccio, Nicola Moccaldi, and Silvia Rossi. Wearable brain–computer interface instrumentation for robot-based rehabilitation by augmented reality. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6362–6371, 2020.

[37] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[38] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[39] N. Aznan, Stephen Bonner, J. D. Connolly, N. A. Moubayed, and T. Breckon. On the classification of SSVEP-based dry-EEG signals via convolutional neural networks. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3726–3731, 2018.

[40] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[41] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[42] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus A. F. Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview . *Bioinformatics*, 16(5):412–424, 05 2000.

[43] Tonio Ball, Markus Kern, Isabella Mutschler, Ad Aertsen, and Andreas Schulze-Bonhage. Signal quality of simultaneously recorded invasive and non-invasive eeg. *Neuroimage*, 46(3):708–716, 2009.

[44] Serena Barello, Stefano Triberti, Guendalina Graffigna, Chiara Libreri, Silvia Serino, Judith Hibbard, and Giuseppe Riva. ehealth for patient engagement: a systematic review. *Frontiers in psychology*, 6:2013, 2016.

[45] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[46] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[47] Chris Berka, Daniel J Levendowski, Michelle N Lumicao, Alan Yau, Gene Davis, Vladimir T Zivkovic, Richard E Olmstead, Patrice D Tremoulet, and Patrick L Craven. Eeg correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, space, and environmental medicine*, 78(5):B231–B244, 2007.

[48] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71, Barcelona, Spain, 2016. Springer.

[49] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[50] Laurent Bonnet, Fabien Lotte, and Anatole Lécuyer. Two brains, one game: design and evaluation of a multiuser bci video game based on motor imagery. *IEEE Transactions on Computational Intelligence and AI in games*, 5(2):185–198, 2013.

[51] Gianluca Borghini, Giovanni Vecchiato, Jlenia Toppi, Laura Astolfi, A Maglione, R Isabella, C Caltagirone, Wanzeng Kong, Daming Wei, Zhengchun Zhou, et al. Assessment of mental fatigue during car driving by using high resolution eeg activity and neurophysiologic indices. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6442–6445. IEEE, 2012.

[52] Sofiane Boucenna, Antonio Narzisi, Elodie Tilmont, Filippo Muratori, Giovanni Pioggia, David Cohen, and Mohamed Chetouani. Interactive technologies for autistic children: A review. *Cognitive Computation*, 6(4):722–740, 2014.

[53] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PLOS ONE*, 12(6):1–17, 06 2017.

[54] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, page 3121–3124, USA, 2010. IEEE Computer Society.

[55] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[56] Jessica Cantillo-Negrete, Ruben I Carino-Escobar, Paul Carrillo-Mora, David Elias-Vinas, and Josefina Gutierrez-Martinez. Motor imagery-based brain-computer interface coupled to a robotic hand orthosis aimed for neurorehabilitation of stroke patients. *Journal of healthcare engineering*, 2018, 2018.

[57] Ratanapat Chanubol, Parit Wongphaet, Napapit Chavanich, Cordula Werner, Stefan Hesse, Anita Bardeleben, and Jan Merholz. A randomized controlled trial of cognitive sensory motor training therapy on the recovery of arm function in acute stroke patients. *Clinical Rehabilitation*, 26(12):1096–1104, 2012. PMID: 22649162.

[58] David Charte, Francisco Charte, María J del Jesus, and Francisco Herrera. An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. *Neurocomputing*, 404:93–107, 2020.

[59] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[60] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[61] Hao Chen, Ming Jin, Zhunan Li, Cunhang Fan, Jinpeng Li, and Huiguang He. Ms-mda: Multisource marginal distribution adaptation for cross-subject and cross-session eeg emotion recognition. *Frontiers in Neuroscience*, 15, 2021.

[62] Huayu Chen, Shuting Sun, Jianxiu Li, Ruilan Yu, Nan Li, Xiaowei Li, and Bin Hu. Personal-zscore: Eliminating individual difference for eeg-based cross-subject emotion recognition. *IEEE Transactions on Affective Computing*, 2021.

[63] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[64] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[65] Xiaogang Chen, Yijun Wang, Shangkai Gao, Tzyy-Ping Jung, and Xiaorong Gao. Filter bank canonical correlation analysis for implementing a high-speed ssvep-based brain–computer interface. *Journal of neural engineering*, 12(4):046008, 2015.

[66] Xiaogang Chen, Yijun Wang, Masaki Nakanishi, Xiaorong Gao, Tzyy-Ping Jung, and Shangkai Gao. High-speed spelling with a noninvasive brain–computer interface. *Proceedings of the national academy of sciences*, 112(44):E6058–E6067, 2015.

[67] Yonghao Chen, Chen Yang, Xiaogang Chen, Yijun Wang, and Xiaorong Gao. A novel training-free recognition method for SSVEP-based BCIs using dynamic window strategy. *Journal of neural engineering*, 18(3):036007, 2021.

[68] Yu Mike Chi, Yu-Te Wang, Yijun Wang, Christoph Maier, Tzyy-Ping Jung, and Gert Cauwenberghs. Dry and noncontact eeg sensors for mobile brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(2):228–235, 2012.

[69] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[70] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[71] Mark Craven and Jude Shavlik. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 8, 1995.

[72] Xin Dang, Ran Wei, and Guohui Li. An efficient movement and mental classification for children with autism based on motion and eeg features. *Journal of Ambient Intelligence and Humanized Computing*, 8(6):907–912, 2017.

[73] Edward L Deci and Richard M Ryan. Conceptualizations of intrinsic motivation and self-determination. In *Intrinsic motivation and self-determination in human behavior*, pages 11–40. Springer, 1985.

[74] Elena Dell'Aquila, Gianpaolo Maggi, Daniela Conti, and Silvia Rossi. A preparatory study for measuring engagement in pediatric virtual and robotics rehabilitation settings. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 183–185, 2020.

[75] Li Deng and Xiao Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):1060–1089, 2013.

[76] Jürgen Dieber and Sabrina Kirrane. Why model why? assessing the strengths and limitations of lime. *arXiv preprint arXiv:2012.00093*, 2020.

[77] Francesco Donnarumma, Roberto Prevete, Domenico Maisto, Simone Fuscone, Emily M Irvine, Matthijs AA van der Meer, Caleb Kemere, and

Giovanni Pezzulo. A framework to identify structured behavioral patterns within rodent spatial trajectories. *Scientific reports*, 11(1):1–20, 2021.

[78] Derek Doran, Sarah Schulz, and Tarek R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.00794, 2017.

[79] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4829–4837, Las Vegas, USA, 2016.

[80] Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20, Oct 2018.

[81] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[82] D. Erhan, Y. Bengio, . Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[83] Ehsan T Esfahani, Shrey Pareek, Pramod Chembrammel, Mostafa Ghobadi, and Thenkurussi Kesavadas. Adaptation of rehabilitation system based on user's mental engagement. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2015.

[84] Chinedu Pascal Ezenkwu, Simeon Ozuomba, and Constance Kalu. Application of k-means algorithm for efficient customer segmentation: a strategy for targeted customer services. 2015.

[85] CWNFCW Fadzal, W Mansor, and LY Khuan. Review of brain computer interface application in diagnosing dyslexia. In *2011 IEEE Control and System Graduate Research Colloquium*, pages 124–128. IEEE, 2011.

[86] Muhamed Farooq and Omid Dehzangi. High accuracy wearable SSVEP detection using feature profiling and dimensionality reduction. In *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 161–164. IEEE, 2017.

[87] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

[88] Felipe Lemes Galvão, Silvio Jamil Ferzoli Guimarães, and Alexandre Xavier Falcão. Image segmentation using dense and sparse hierarchies of superpixels. *Pattern Recognition*, 108:107532, 2020.

[89] Hao Gao, Chi-Man Pun, and Sam Kwong. An efficient image segmentation method based on a hybrid particle swarm algorithm with learning strategy. *Information Sciences*, 369:500–521, 2016.

[90] Guido HE Gendolla. Self-relevance of performance, task difficulty, and task engagement assessed as cardiovascular response. *Motivation and Emotion*, 23(1):45–66, 1999.

[91] Davoud Gholamiangonabadi, Nikita Kiselov, and Katarina Grolinger. Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. *IEEE Access*, 8:133982–133994, 2020.

[92] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019.

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

[94] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[95] Milan Gnjatović. Therapist-centered design of a robot's dialogue behavior. *Cognitive Computation*, 6(4):775–788, 2014.

[96] Guendalina Graffigna, Serena Barello, Andrea Bonanomi, and Edoardo Lozza. Measuring patient engagement: development and psychometric properties of the patient health engagement (phe) scale. *Frontiers in psychology*, 6:274, 2015.

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

[98] Xiaowei Gu and Weiping Ding. A hierarchical prototype-based approach for classification. *Information Sciences*, 505:325–351, 2019.

[99] Christoph Guger, Brendan Z Allison, Bernhard Großwindhager, Robert Prückl, Christoph Hintermüller, Christoph Kapeller, Markus Bruckner, Gunther Krausz, and Günter Edlinger. How many people could use an ssvep bci? *Frontiers in neuroscience*, 6:169, 2012.

[100] Riccardo Guidotti, Anna Monreale, Stan Matwin, and Dino Pedreschi. Explaining image classifiers generating exemplars and counter-exemplars from latent representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13665–13668, 2020.

[101] Laurent Guigues, Jean Pierre Cocquerez, and Hervé Le Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.

[102] Silvio Jamil F Guimarães, Jean Cousty, Yukiko Kenmochi, and Laurent Najman. A hierarchical image segmentation algorithm based on an observation scale. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 116–125. Springer, 2012.

[103] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[104] Gido Hakvoort, Boris Reuderink, and Michel Obbink. Comparison of psda and cca detection methods in a SSVEP-based BCI-system. *Centre for Telematics & Information Technology University of Twente*, 2011.

[105] Ronan Hamon, Henrik Junklewitz, Ignacio Sanchez, Gianclaudio Malgieri, and Paul De Hert. Bridging the gap between ai and explainability in the gdpr: towards trustworthiness-by-design in automated decision-making. *IEEE Computational Intelligence Magazine*, 17(1):72–85, 2022.

[106] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[107] Angela R Harrivel, Daniel H Weissman, Douglas C Noll, and Scott J Peltier. Monitoring attentional state with fnirs. *Frontiers in human neuroscience*, 7:861, 2013.

[108] Bruno Miranda Henrique, Vinicius Amorim Sobreiro, and Herbert Kimura. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124:226–251, 2019.

[109] Judith H Hibbard, Eldon R Mahoney, Jean Stockard, and Martin Tusler. Development and testing of a short form of the patient activation measure. *Health services research*, 40(6p1):1918–1930, 2005.

[110] I. Higgins, Loïc Matthey, A. Pal, Christopher P. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[111] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[112] Mark Holmstrom, Dylan Liu, and Christopher Vo. Machine learning applied to weather forecasting. *Meteorol. Appl*, 10:1–5, 2016.

[113] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[114] Richard W Homan, John Herman, and Phillip Purdy. Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology*, 66(4):376–382, 1987.

[115] J-M Hopf and George R Mangun. Shifting visual attention in space: an electrophysiological analysis using high spatial resolution mapping. *Clinical neurophysiology*, 111(7):1241–1257, 2000.

[116] David Ibanez-Soria, Aureli Soria-Frisch, Jordi Garcia-Ojalvo, and Giulio Ruffini. Characterization of the non-stationary nature of steady-state visual evoked potentials using echo state networks. *PloS one*, 14(7):e0218771, 2019.

[117] Cosimo Ieracitano, Nadia Mammone, Amir Hussain, and Francesco Carlo Morabito. A novel explainable machine learning approach for eeg-based brain-computer interface systems. *Neural Computing and Applications*, 34(14):11347–11360, 2022.

[118] Chuan Jia, Xiaorong Gao, Bo Hong, and Shangkai Gao. Frequency and phase mixed coding in SSVEP-based brain–computer interface. *IEEE Transactions on Biomedical Engineering*, 58(1):200–206, 2010.

[119] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[120] William A Kahn. Psychological conditions of personal engagement and disengagement at work. *Academy of management journal*, 33(4):692–724, 1990.

[121] Daniel Kahneman and Amos Tversky. The simulation heuristic. Technical report, Stanford Univ Ca Dept Of Psychology, 1981.

[122] Alexander Ya Kaplan, Andrew A Fingelkurts, Alexander A Fingelkurts, Sergei V Borisov, and Boris S Darkhovsky. Nonstationary nature of the brain activity as revealed by eeg/meg: methodological, practical and conceptual challenges. *Signal processing*, 85(11):2190–2212, 2005.

[123] Yufeng Ke, Pengxiao Liu, Xingwei An, Xizi Song, and Dong Ming. An online SSVEP-BCI system in an optical see-through augmented reality environment. *Journal of neural engineering*, 17(1):016066, 2020.

[124] Eoin M Kenny and Mark T Keane. Explaining deep learning using examples: Optimal feature weighting methods for twin systems using post-hoc, explanation-by-example in xai. *Knowledge-Based Systems*, 233:107530, 2021.

[125] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[126] Sungchul Kim, Tao Qin, Tie-Yan Liu, and Hwanjo Yu. Advertiser-centric approach to understand user click behavior in sponsored search. *Information Sciences*, 276:242–254, 2014.

[127] Gillian King, Lisa A Chiarello, Laura Thompson, Matthew JW McLarnon, Eric Smart, Jenny Ziviani, and Madhu Pinto. Development of an observational measure of therapy engagement for pediatric rehabilitation. *Disability and Rehabilitation*, 41(1):86–97, 2019.

[128] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

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

[130] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[131] Li-Wei Ko, SSK Ranga, Oleksii Komarov, and Chung-Chiang Chen. Development of single-channel hybrid BCI system using motor imagery and ssvep. *Journal of healthcare engineering*, 2017, 2017.

[132] Toshiaki Koike-Akino, Ruhi Mahajan, Tim K Marks, Ye Wang, Shinji Watanabe, Oncel Tuzel, and Philip Orlik. High-accuracy user identification using eeg biometrics. In *2016 38th annual international conference of the*

IEEE engineering in medicine and biology society (EMBC), pages 854–858. IEEE, 2016.

[133] Zoltan J Koles, Michael S Lazar, and Steven Z Zhou. Spatial patterns underlying population differences in the background eeg. *Brain topography*, 2(4):275–284, 1990.

[134] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

[135] Steve W Kozlowski and Keith Hattrup. A disagreement about within-group agreement: Disentangling issues of consistency versus consensus. *Journal of applied psychology*, 77(2):161, 1992.

[136] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[137] N. Kumar and K. P. Michmizos. Machine learning for motor learning: Eeg-based continuous assessment of cognitive engagement for adaptive rehabilitation robots. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 521–526, 2020.

[138] Yatindra Kumar, ML Dewal, and Radhey Shyam Anand. Epileptic seizures detection in eeg using dwt-based apen and artificial neural network. *Signal, Image and Video Processing*, 8(7):1323–1334, 2014.

[139] Shinjini Kundu. Ai in medicine must be explainable. *Nature medicine*, 27(8):1328–1328, 2021.

[140] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[141] E Laxmi Lydia, CSS Anupama, and N Sharmili. Modeling of explainable artificial intelligence with correlation-based feature selection approach for biomedical data analysis. In *Biomedical Data Analysis and Processing Using Explainable (XAI) and Responsive Artificial Intelligence (RAI)*, pages 17–32. Springer, 2022.

[142] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. pages 2278–2324, 1998.

[143] Te-Won Lee. Independent component analysis. In *Independent component analysis*, pages 27–66. Springer, 1998.

[144] Anthony H Lequerica and Kathleen Kortte. Therapeutic engagement: a proposed model of engagement in medical rehabilitation. *American journal of physical medicine & rehabilitation*, 89(5):415–422, 2010.

[145] Eric C Leuthardt, Gerwin Schalk, Jarod Roland, Adam Rouse, and Daniel W Moran. Evolution of brain-computer interfaces: going beyond classic motor physiology. *Neurosurgical focus*, 27(1):E4, 2009.

[146] Bentian Li and Dechang Pi. Network representation learning: a systematic literature review. *Neural Computing and Applications*, pages 1–33, 2020.

[147] Jinyan Li, Lian-sheng Liu, Simon Fong, Raymond K Wong, Sabah Mohammed, Jinan Fiaidhi, Yunsick Sung, and Kelvin KL Wong. Adaptive swarm balancing algorithms for rare-event prediction in imbalanced healthcare data. *PloS one*, 12(7):e0180830, 2017.

[148] Y. Li, L. Wang, W. Zheng, Y. Zong, L. Qi, Z. Cui, T. Zhang, and T. Song. A novel bi-hemispheric discrepancy model for eeg emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1, 2020.

[149] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning. *Information Sciences*, 482:73–85, 2019.

[150] Yao Li and Thenkurussi Kesavadas. SSVEP-based brain-computer interface for part-picking robotic co-worker. *Journal of Computing and Information Science in Engineering*, 22(2):021001, 2021.

[151] Sheng-Fu Liang, Fu-Zen Shaw, Chung-Ping Young, Da-Wei Chang, and Yi-Cheng Liao. A closed-loop brain computer interface for real-time seizure detection and control. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4950–4953. IEEE, 2010.

[152] Brian Y Lim, Qian Yang, Ashraf M Abdul, and Danding Wang. Why these explanations? selecting intelligibility types for explanation goals. In *IUI Workshops*, 2019.

[153] Choon Guan Lim, Tih Shih Lee, Cuntai Guan, Daniel Shuen Sheng Fung, Yudong Zhao, Stephanie Sze Wei Teng, Haihong Zhang, and K Ranga Rama Krishnan. A brain-computer interface based attention training program for treating attention deficit hyperactivity disorder. *PloS one*, 7(10), 2012.

[154] Choon Guan Lim, Xue Wei Wendy Poh, Shuen Sheng Daniel Fung, Cuntai Guan, Dianne Bautista, Yin Bun Cheung, Haihong Zhang, Si Ning Yeo, Ranga Krishnan, and Tih Shih Lee. A randomized controlled trial of a

brain-computer interface based attention training program for adhd. *PloS one*, 14(5):e0216225, 2019.

[155] Bor-Shing Lin, Hsiao-An Wang, Yao-Kuang Huang, Yu-Lin Wang, and Bor-Shyh Lin. Design of ssvep enhancement-based brain computer interface. *IEEE Sensors Journal*, 21(13):14330–14338, 2021.

[156] Zhonglin Lin, Changshui Zhang, Wei Wu, and Xiaorong Gao. Frequency recognition based on canonical correlation analysis for ssvep-based BCIs. *IEEE transactions on biomedical engineering*, 53(12):2610–2614, 2006.

[157] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[158] Dixiu Liu, Danni Cai, Tom Verguts, and Qi Chen. The time course of spatial attention shifts in elementary arithmetic. *Scientific reports*, 7(1):1–8, 2017.

[159] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[160] Robert Gabriel Lupu, Danut Constantin Irimia, Florina Ungureanu, Marian Silviu Poboroniuc, and Alin Moldoveanu. Bci and fes based therapy for stroke rehabilitation using vr facilities. *Wireless Communications and Mobile Computing*, 2018, 2018.

[161] Bo-Qun Ma, He Li, Wei-Long Zheng, and Bao-Liang Lu. Reducing the subject variability of eeg signals with adversarial domain generalization. In *International Conference on Neural Information Processing*, pages 30–42. Springer, 2019.

[162] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[163] Jaakko Malmivuo, Robert Plonsey, et al. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA, 1995.

[164] Zijing Mao, Wan Xiang Yao, and Yufei Huang. Eeg-based biometric identification with deep learning. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 609–612. IEEE, 2017.

[165] David Marshall, Damien Coyle, Shane Wilson, and Michael Callaghan. Games, gameplay, and bci: the state of the art. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(2):82–99, 2013.

[166] Ignas Martišius and Robertas Damaševičius. A prototype SSVEP based real time BCI gaming system. *Computational intelligence and neuroscience*, 2016, 2016.

[167] Rytis Maskeliunas, Robertas Damasevicius, Ignas Martisius, and Mindaugas Vasiljevas. Consumer-grade eeg devices: are they usable for control tasks? *PeerJ*, 4:e1746, 2016.

[168] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[169] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[170] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[171] Francesco Carlo Morabito, Cosimo Ieracitano, and Nadia Mammone. An explainable artificial intelligence approach to study mci to ad conversion via hd-eeg processing. *Clinical EEG and Neuroscience*, page 15500594211063662, 2021.

[172] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[173] F. Mosteller and J Tukey. Data analysis, including statistics. In G. Lindzey and E. Aronson, editors, *Revised Handbook of Social Psychology*, volume 2, pages 80–203. Addison Wesley, 1968.

[174] Klaus-Robert Müller, Matthias Krauledat, Guido Dornhege, Gabriel Curio, and Benjamin Blankertz. Machine learning techniques for brain-computer interfaces. *Biomed. Tech*, 49(1):11–22, 2004.

[175] Gernot R Müller-Putz, Reinhold Scherer, Christian Brauneis, and Gert Pfurtscheller. Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of neural engineering*, 2(4):123, 2005.

[176] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[177] Michael S Myslobodsky and Jacob Bar-Ziv. Locations of occipital eeg electrodes verified by computed tomography. *Electroencephalography and clinical neurophysiology*, 72(4):362–366, 1989.

[178] Agata Nawrocka, Andrzej Kot, and Marcin Nawrocki. Application of machine learning in recommendation systems. In *2018 19th International Carpathian Control Conference (ICCC)*, pages 328–331. IEEE, 2018.

[179] Jerzy Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics*, pages 123–150. Springer, 1992.

[180] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. *ArXiv e-prints*, February 2016.

[181] Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. Borderline oversampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigm.*, 3(1):4–21, April 2011.

[182] Trung-Hau Nguyen and Wan-Young Chung. A single-channel SSVEP-based BCI speller using deep learning. *IEEE Access*, 7:1752–1763, 2018.

[183] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *sensors*, 12(2):1211–1279, 2012.

[184] Lloyd M Nirenberg, John Hanley, and Edwin B Stear. A new approach to prosthetic control: Eeg motor signal tracking with an adaptively designed phase-locked loop. *IEEE Transactions on Biomedical Engineering*, (6):389–398, 1971.

[185] Paul L Nunez, Ramesh Srinivasan, et al. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[186] L P. Devroye and T Wagner. Distribution-free performance bounds for potential function rules. In *Information Theory, IEEE Transactions on*, volume IT-25, pages 601 – 604, 10 1979.

[187] Wanjoo Park, Gyu Hyun Kwon, Da-Hye Kim, Yun-Hee Kim, Sung-Phil Kim, and Laehyun Kim. Assessment of cognitive engagement in stroke patients from single-trial eeg during motor rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(3):351–362, 2014.

[188] Gert Pfurtscheller, Christoph Guger, and Herbert Ramoser. Eeg-based brain-computer interface using subject-specific spatial filters. In *International Work-Conference on Artificial Neural Networks*, pages 248–254. Springer, 1999.

[189] Gert Pfurtscheller and Christa Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 2001.

[190] Gert Pfurtscheller, Christa Neuper, and Niels Birbaumer. 4 human brain–computer interface. 12 2004.

[191] Richard R Picard and R Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[192] Joshua J Podmore, Toby P Breckon, Nik KN Aznan, and Jason D Connolly. On the relative contribution of deep convolutional neural networks for ssvep-based bio-signal decoding in BCI speller applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):611–618, 2019.

[193] Kun Qian, Marina Danilevsky, Yannis Katsis, Ban Kawas, Erick Oduor, Lucian Popa, and Yunyao Li. Xnlp: A living survey for xai research in natural language processing. In *26th International Conference on Intelligent User Interfaces-Companion*, pages 78–80, 2021.

[194] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[195] Carolyn Ranti, Warren Jones, Ami Klin, and Sarah Shultz. Blink rate patterns provide a reliable measure of individual engagement with scene content. *Scientific Reports*, 10(1):1–10, 2020.

[196] Rajesh PN Rao and Reinhold Scherer. Brain-computer interfacing [in the spotlight]. *IEEE Signal Processing Magazine*, 27(4):152–150, 2010.

[197] Mamunur Rashid, Norizam Sulaiman, Anwar PP Abdul Majeed, Rabiu Muazu Musa, Bifta Sama Bari, Sabira Khatun, et al. Current status, challenges, and possible solutions of eeg-based brain-computer interface: a comprehensive review. *Frontiers in neurorobotics*, page 25, 2020.

[198] Prajakta Rathod and Shefali Naik. Review on epilepsy detection with explainable artificial intelligence. In *2022 10th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-22)*, pages 1–6. IEEE, 2022.

[199] Aravind Ravi, Nargess Heydari Beni, Jacob Manuel, and Ning Jiang. Comparing user-dependent and user-independent training of cnn for SSVEP BCI. *Journal of neural engineering*, 17(2):026028, 2020.

[200] Holger Regenbrecht, Simon Hoermann, Claudia Ott, Lavell Mueller, and Elizabeth Franz. Manipulating the experience of reality for rehabilitation applications. *Proceedings of the IEEE*, 102(2):170–184, 2014.

[201] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

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

[203] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

[204] Mireia Ribera and Agata Lapedriza. Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*, volume 2327, page 38, 2019.

[205] Audrey S Royer, Alexander J Doud, Minn L Rose, and Bin He. Eeg control of a virtual helicopter in 3-dimensional space using intelligent control strategies. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(6):581–589, 2010.

[206] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[207] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, and K.R. Muller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 5–22. Springer, 2019.

[208] Arthur L Samuel. Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, 11(6):601–617, 1967.

[209] Saeid Sanei and Jonathon A Chambers. *EEG signal processing*. John Wiley & Sons, 2013.

[210] Phattarapong Sawangjai, Supanida Hompoonsup, Pitshaporn Leelaarporn, Supavit Kongwudhikunakorn, and Theerawit Wilaiprasitporn. Consumer grade eeg measuring sensors as research tools: A review. *IEEE Sensors Journal*, 20(8):3996–4024, 2019.

[211] Tjeerd AJ Schoonderwoerd, Wiard Jorritsma, Mark A Neerincx, and Karel Van Den Bosch. Human-centered xai: Developing design patterns for explanations of clinical decision support systems. *International Journal of Human-Computer Studies*, 154:102684, 2021.

[212] Naeem Seliya, Taghi M Khoshgoftaar, and Jason Van Hulse. Predicting faults in high assurance software. In *2010 IEEE 12th international symposium on high assurance systems engineering*, pages 26–34. IEEE, 2010.

[213] Eric W Sellers, Theresa M Vaughan, and Jonathan R Wolpaw. A brain-computer interface for long-term independent home use. *Amyotrophic lateral sclerosis*, 11(5):449–455, 2010.

[214] R Pandi Selvam, A Sheryl Oliver, V Mohan, NB Prakash, and T Jayasankar. Explainable artificial intelligence with metaheuristic feature selection technique for biomedical data classification. In *Biomedical Data Analysis and Processing Using Explainable (XAI) and Responsive Artificial Intelligence (RAI)*, pages 43–57. Springer, 2022.

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

[216] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[217] Praveen Kumar Shukla, Rahul Kumar Chaurasiya, Shrish Verma, and G. R. Sinha. A thresholding-free state detection approach for home appliance control using p300-based bci. *IEEE Sensors Journal*, 21(15):16927–16936, 2021.

[218] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[219] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, Workshop Track Proceedings*, Banff, Canada, 2014.

[220] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[221] Rajesh Singla and BA Haseena. Comparison of ssvep signal classification techniques using svm and ann models for BCI applications. *International Journal of Information and Electronics Engineering*, 4(1), 2014.

[222] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[223] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3745–3753, 2016.

[224] T. Song, W. Zheng, C. Lu, Y. Zong, X. Zhang, and Z. Cui. Mped: A multi-modal physiological emotion database for discrete emotion recognition. *IEEE Access*, 7:12177–12191, 2019.

[225] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[226] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[227] D.S. Tan and Anton Nijholt. *Brain-Computer Interfaces: Applying Our Minds to Human-Computer Interaction*. 01 2010.

[228] A. Tedesco, D. Dallet, and P. Arpaia. Augmented reality (AR) and brain-computer interface (BCI): Two enabling technologies for empowering the fruition of sensor data in the 4.0 era. In *Proceedings of the AISEM 2020 Regional Workshop*, volume 753, pages 85–91, 2021.

[229] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[230] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[231] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[232] Marijn Van Vliet, Arne Robben, Nikolay Chumerin, Nikolay V Manyakov, Adrien Combaz, and Marc M Van Hulle. Designing a brain-computer interface controlled video-game using consumer grade eeg hardware. In *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, pages 1–6. IEEE, 2012.

[233] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[234] Rosanna Maria Viglialoro, Sara Condino, Giuseppe Turini, Marina Carbone, Vincenzo Ferrari, and Marco Gesi. Review of the augmented reality systems for shoulder rehabilitation. *Information*, 10(5):154, 2019.

[235] Peitao Wang, Jun Lu, Bin Zhang, and Zeng Tang. A review on transfer learning for brain-computer interface classification. In *2015 5th International Conference on Information Science and Technology (ICIST)*, pages 315–322. IEEE, 2015.

[236] Qiong Wang, Huan Wang, Chunxia Zhao, and Jingyu Yang. Driver fatigue detection technology in active safety systems. In *2011 International Conference on Remote Sensing, Environment and Transportation Engineering*, pages 3097–3100. IEEE, 2011.

[237] Xin Wang, Teng Cao, Boyu Wang, Feng Wan, Peng Un Mak, Pui In Mak, Mang I Vai, and Chaozheng Li. An online ssvep-based chatting system. In *Proceedings 2011 International Conference on System Science and Engineering*, pages 536–539. IEEE, 2011.

[238] Yijun Wang, Xiaogang Chen, Xiaorong Gao, and Shangkai Gao. A benchmark dataset for SSVEP-based brain–computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 25(10):1746–1752, 2016.

[239] Yijun Wang, T-P Jung, et al. Visual stimulus design for high-rate SSVEP BCI. *Electronics letters*, 46(15):1057–1058, 2010.

[240] Yijun Wang, Ruiping Wang, Xiaorong Gao, Bo Hong, and Shangkai Gao. A practical vep-based brain-computer interface. *IEEE Transactions on neural systems and rehabilitation engineering*, 14(2):234–240, 2006.

[241] Yu-Kai Wang, Shi-An Chen, and Chin-Teng Lin. An eeg-based brain–computer interface for dual task driving detection. *Neurocomputing*, 129:85–93, 2014.

[242] Adrian Weller. Transparency: Motivations and challenges, 2017.

[243] Agnieszka Wosiak and Aleksandra Dura. Hybrid method of automated eeg signals' selection using reversed correlation algorithm for improved classification of emotions. *Sensors*, 20(24):7083, 2020.

[244] Chung-Min Wu, Yeou-Jiunn Chen, Ilham AE Zaeni, and Shih-Chung Chen. A new SSVEP based BCI application on the mobile robot in a maze game. In *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pages 550–553. IEEE, 2016.

[245] Erwei Yin, Zongtan Zhou, Jun Jiang, Yang Yu, and Dewen Hu. A dynamically optimized SSVEP brain–computer interface (BCI) speller. *IEEE Transactions on Biomedical Engineering*, 62(6):1447–1456, 2014.

[246] Jun Yu, Di Huang, and Zhongliang Wei. Unsupervised image segmentation via stacked denoising auto-encoder and hierarchical patch indexing. *Signal Processing*, 143:346–353, 2018.

[247] Thorsten O Zander, Christian Kothe, Sabine Jatzev, and Matti Gaertner. Enhancing human-computer interaction with input from active and passive brain-computer interfaces. In *Brain-computer interfaces*, pages 181–199. Springer, 2010.

[248] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Cision*, pages 818–833, Zurich, Switzerland, 2014. Springer.

[249] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2018–2025, Barcelona, Spain, 2011. IEEE.

[250] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[251] Q. Zhang and S. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[252] Xiang Zhang, Lina Yao, Chaoran Huang, Tao Gu, Zheng Yang, and Yunhao Liu. Deepkey: an eeg and gait based dual-authentication system. *arXiv preprint arXiv:1706.01606*, 2017.

[253] Xiang Zhang, Lina Yao, Salil S Kanhere, Yunhao Liu, Tao Gu, and Kaixuan Chen. Mindid: Person identification from brain waves through attention-based recurrent neural network. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–23, 2018.

[254] Xiang Zhang, Lina Yao, Quan Z Sheng, Salil S Kanhere, Tao Gu, and Dalin Zhang. Converting your thoughts to texts: Enabling brain typing via deep

feature learning of eeg signals. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, 2018.

[255] Xiang Zhang, Lina Yao, Shuai Zhang, Salil Kanhere, Michael Sheng, and Yunhao Liu. Internet of things meets brain–computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity. *IEEE Internet of Things Journal*, 6(2):2084–2092, 2018.

[256] Xiaofeng Zhang, Yujuan Sun, Hui Liu, Zhongjun Hou, Feng Zhao, and Caiming Zhang. Improved clustering algorithms for image segmentation based on non-local information and back projection. *Information Sciences*, 550:129–144, 2021.

[257] Yue Zhang, Shane Q. Xie, He Wang, and Zhiqiang Zhang. Data analytics in steady-state visual evoked potential-based brain–computer interface: A review. *IEEE Sensors Journal*, 21(2):1124–1138, 2021.

[258] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *International Conference on Machine Learning*, pages 4091–4099. PMLR, 2017.

[259] Xingang Zhao, Yaqi Chu, Jianda Han, and Zhiqiang Zhang. SSVEP-based brain–computer interface controlled functional electrical stimulation system for upper extremity rehabilitation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):947–956, 2016.

[260] Xingyu Zhao, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. *arXiv preprint arXiv:2012.03058*, 2020.

[261] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017.

[262] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015.

[263] Xiangwei Zheng, Xiaofeng Liu, Yuang Zhang, Lizhen Cui, and Xiaomei Yu. A portable hci system-oriented eeg feature extraction and channel selection for emotion recognition. *International Journal of Intelligent Systems*, 36(1):152–176, 2021.

[264] Peixiang Zhong, Di Wang, and Chunyan Miao. Eeg-based emotion recognition using regularized graph neural networks. *IEEE Transactions on Affective Computing*, PP:1–1, 05 2020.

[265] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[266] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[267] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.

[268] André Zúquete, Bruno Quintela, and João Paulo da Silva Cunha. Biometric authentication using brain responses to visual stimuli. In *Biosignals*, pages 103–112, 2010.

Author's Publications

In this section the list of my scientific publications during PhD can be found.

- Apicella, A., Giugliano, S., Isgrò, F., & Prevete, R. (2022). **Exploiting auto-encoders and segmentation methods for middle-level explanations of image classification systems**. Knowledge-Based Systems, 255, 109725.
- Apicella, A., Arpaia, P., Giugliano, S., Mastrati, G., & Moccaldi, N. (2022). **High-wearable EEG-based transducer for engagement detection in pediatric rehabilitation**. Brain-Computer Interfaces, 9(3), 129-139.
- Apicella, A., Arpaia, P., Cataldo, A., De Benedetto, E., Donato, N., Duraccio, L., Giugliano, S, & Prevete, R. (2022). **Adoption of Machine Learning Techniques to Enhance Classification Performance in Reactive Brain-Computer Interfaces.** In 2022 IEEE International Symposium on Medical Measurements and Applications (MeMeA) (pp. 1-5). IEEE.
- Angrisani, L., Apicella, A., Arpaia, P., De Benedetto, E., Donato, N., Duraccio, L., Giugliano, S, & Prevete, R. (2022). **A ML-based Approach to Enhance Metrological Performance of Wearable Brain-Computer Interfaces.** In 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) (pp. 1-5). IEEE.
- Apicella, A., Arpaia, P., De Benedetto, E., Donato, N., Duraccio, L., Giugliano, S., & Prevete, R. (2022). **Enhancement of SSVEPs classification in BCI-based wearable instrumentation through machine Learning Techniques**. IEEE Sensors Journal, 22(9), 9087-9094.
- Apicella, A., Giugliano, S., Isgrò, F., & Prevete, R. (2021). **Explanations in terms of Hierarchically organised Middle Level Features.**

Italian Workshop on Explainable Artificial Intelligence. CEUR Workshop Proceedings (CEUR-WS.org) .

- Apicella, A., Giugliano, S., Isgrò, F., & Prevete, R. (2021). **A general approach to compute the relevance of middle-level input features.** In International Conference on Pattern Recognition (pp. 189-203). Springer, Cham.