Machine Learning to Elucidate Mechanisms of Human Cognition and Epilepsy

Dissertation

Zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Amirhossein Jahanbekam

aus

Schiraz, Iran

Bonn, July 2019

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Christian Bauckhage

2. Gutachter: Prof. Dr. Armin Cremers

Tag der Promotion: 05.05.2020

Erscheinungsjahr: 2020

"Only two things are infinite, the universe and human stupidity, and I'm not sure about the former."

Albert Einstein

Summary

Machine learning approaches, a branch of computer science based on the study of complex statistical algorithms, help researchers predict and discover facts about the outside world that may otherwise be too latent and sophisticated for more commonplace approaches.

Machine learning techniques are able to explore large amounts of data in a multivariate fashion, so that multiple factors comprising a phenomenon are analyzed simultaneously; a technique that human intelligence is not fully optimized for. Accordingly, machine learning is becoming a widely used assistive tool in many fields of science and technology. In the same vein, the current thesis aims to methodize two main scientific questions within the realm of the neurosciences using machine learning frameworks.

Here, machine learning is used to give a viable solution for decoding ongoing brain activities in cognitive studies using data obtained via intracranial electroencephalography (iEEG). IEEG data, represented as a $3D$ model is proposed, allowing the data to be broken down into distinct bins of information, and in addition, to be able to identify and discard non-informative components. Combining this data modeling approach with suitable machine learning algorithms, facilitates the procedure of interpreting brain activity and enables a traceable and plausible pattern classification solution.

Regarding the second scientific question, machine learning is implemented to aid epilepsy patients in tracking and recording their seizures. In order for patients with epilepsy to receive adequate counseling and treatment, accurate documentation of seizure activity is required, however research has shown that self-reporting of seizure activity is often fundamentally unreliable. In this thesis, extensive studies aiming to investigate this question were carried out and subsequently machine learning approaches are proposed to track and register the seizure activity of individuals with epilepsy based on bio-feedback signals.

Additionally, an introduction to the state-of-the-art deep artificial neural networks is given; in addition to discussing the applicability of deep learning on natural neural data.

Acknowledgements

I would like to express my deep gratitude to the advisor of my thesis Prof. Christian Bauckhage for his inspiration and support, before and during my Ph.D study, for his sustained advocacy and encouragements. His guidance has been always a life-saver for me and I have kept them in mind word by word.

Next, I would like to thank the rest of my thesis committee: Prof. Armin Cremers, Prof. Emmanuel Müller, and PD. Dr. Juergen Fell, for accepting to review my work and for their supportive comments and suggestions.

My candid appreciation also goes to Prof. Nikolai Axmacher, PD. Dr. Joergen Fell, and Prof. Rainer Surges, and Prof. Christian Elger who provided me a unique opportunity to join their teams in the university clinic Bonn at the epilepsy center in Bonn, and who gave me access to their laboratory and clinical research facilities. Without their precious support and supervision, it would not have be possible to conduct this research.

I would like to thank Boll foundation for its generous financial support in pushing forward my research studies.

Contents

List of Figures

List of Tables

Abbreviations

ANOVA	ANalysis Of VAriance
AI	Artificial Intelligence
CNS	Central Nervous System
CPS	Complex Partial Seizure
CPU	Central Processing Unit
CSI	Cardiac Sympathetic Index
CVI	Cardiac Vagal Index
DFT	Discrete Fourier Transform
EDA	Electrodermal Activity
ECG	Electrocardiography
ECoG	Electrocorticography
EEG	Electroencephalography
EMG	Electromyography
ERP	Event Related Potential
FFT	Fast Fourier Transform
fMRI	functional Magnetic Resonance Imaging
GPU	Graphics Processing Unit
GTCS	Generalized Tonic-Clonic Seizure
HRV	Heart Rate Variability
ICA	Independent Component Analysis
iEEG	intracranial Electroencephalography
MEG	Magnetoencephalography
PLV	Phase Locking Value
PNS	Peripheral Nervous System
PNS	Parasympathetic Nervous System

PPG	Photoplethysmography
PRV	Pulse Rate Variability
PSD	Power Spectral Density
ReLU	Rectified Linear Unit
RRI	R-Peak to R-Peak Intervals
RSA	Representational Similarity Analysis
RT	Reaction Time
STFFT	Short Time Fast Fourier Transform
SNS	Parasympathetic Nervous System
SPS	Simple Partial Seizure
SVM	Support Vector Machine

Dedicated to my thirsty land...

Chapter 1

Introduction

The brain is probably the most sophisticated natural system in the world. Some people consider it as the greatest mystery of our time. The building blocks of the brain are neural cells (*neurons*) and the brain consists of around 86 billion neurons interconnected to each other [53], eventually making trillions of neural connections. Yet adding to its complexity is not only the multitude of different permutations but also the plasticity of its neural networks. Ultimately, the brain as hardware and mind, the electrical interactions between neural ensembles, shapes what we are as humans and is responsible for an enormous array of voluntary and involuntary actions and reactions to the environment, as well as the brain itself. Huge numbers of scientists are trying diligently to discover the designation of bits and pieces of central nervous system to partially answer the big question: "how does the brain work?".

Consciousness, memory, emotion, language processing, perception, and cognition are various subfields in the realm of brain study. For a scientist to pose a scientific question and to explore various aspects of brain functioning, a scientific experiment should be designed and conducted. Answering to the questions and hypotheses such as how a particular section of the brain functions, how different parts of the brain interact and exchange information, how the brain processes the sensory information, how information transmits in the brain, how we imagine things, how the brain deals with emotion, how memory shapes, stores, transmits and consolidates in the brain, how the brain abilities can be manipulated by external actuators like drugs or sensory stimuli, and how to cure brain diseases and injuries, all require designing subtle experimental conditions and testing.

Scientific experiments for brain studies can be accomplished either in vitro or in-vivo. In an in-vitro experiment, a part of nervous system of the living animal is extracted and then studied precisely in a laboratory environment against electrical, chemical, or

biological simulations. A great portion of in-vitro studies aim to discover the functions of neurons or neural ensembles on the molecular and cellular level. Apart from its scientific significance, in-vitro studies are crucial, especially for medical purposes, in order to develop medications for neurological and psychological diseases.

In an in-vivo experiment, the brain activity of a living animal or human is tested by means of utilizing different measuring systems such as EEG, MRI, MEG, or even calcium-imaging. Typically in in-vivo experiments, an experimental paradigm is designed so that to perform the experiment in a controlled manner, e.g. to test the effect of a particular stimulation on some sections of the nervous system. For instance, which set of neurons in mice *amygdala* [1] will be activated when a mouse tastes something sweet, something sour, or something tasteless. While a living animal or human is carrying out the experiment, the brain signal is being recorded simultaneously in either of the mentioned methods of measuring brain activity. In the current thesis, I report some of the experiments conducted in order to study brain function, which are all in-vivo.

The data volume of recordings are typically huge, and are very complicated to interpret. Data science and analysis is an invaluable aspect of brain research in which, the signals recorded from brain activities will be transformed, statistically checked, categorized, and evaluated in order to be able to methodologically interpret the mechanisms underlying brain activities and functions.

In the past decades, the growth in computational capabilities of computers has had a huge impact on natural sciences. The physical, chemical, and biological information collected from vast numbers of studies could be processed quickly and efficiently by using the computational power of computers. An astronomer can use the computer aided image processing to discern asteroids, a chemist can use computer assisted mass-spectrometry tools to measure features of small particles and a biologist can use computer-based tools to feature different types of cells.

More recently, in the neurosciences, the role of neural computation has been greatly highlighted so that a new subfield of neuroscience has emerged. This has been termed *Computational Neuroscience* in which, the processing of information obtained from various neural organisms is of primary interest [13]. While computational neuroscience as an interdisciplinary field of research, apart from neural sciences itself, has vast implication in physics, psychology, electrical engineering, it has also a profound connection to computer science.

To tackle the difficulties of neural data analysis, computer scientists joined the neuroscience labs across the globe to accommodate the state-of-the-art machine learning and

[1] Amygdala is a structural component of medial temporal lobe responsible mostly for emotion processing, memory, and decision-making.

data analysis techniques to be able to explore and interpret neural information. Machine learning is being increasingly used in cognitive neuroscience to assist in yielding models of neural behaviors. In the same spirit, the current thesis deals with developing machine learning ideas in neural data domains. Hence, this thesis is intended to be presented in the field of computer science and machine learning since it poses machine learning questions regarding neural data, and it deals with the computational aspect of neural data.

1.1 Machine learning philosophy in neuroscience

Complex phenomena such as the French revolution or global warming could not possibly be solely influenced by a single solitary factor, but rather by variety of causes. In philosophy, this is known as the "fallacy of single cause", i.e. it is crucial to account for each individual factor, as it plays its smaller or larger role in giving rise to an event. In order to study such phenomena, what should be emphasized is that all factors should be considered simultaneously.

Cognitive neuroscience is a scientific field of study which focuses on investigating the biological and neural connections that are the basis for the brain cognitive states and memory perception in humans and animals [40]. Brain activities are the product of complex neural activities, and cognitive brain states by definition are high level brain activities. When dealing with higher levels of human cognitive states such as remembering or forgetting, being happy or sad, being stressed or confident, ... the cognitive states are the momentary states which are the end product of activities of neural ensembles at a particular time [90]. Similar to the examples given before, higher level neural activities are also not the result of a single actuator or cause.

In cognitive neuroscience, there are two approaches to interpret brains cognitive states. One is *univariate analysis* and the other is *multivariate analysis*. In univariate approaches of analyzing brain activity, the rule of a single variable is studied individually at a time and then the combinations of limited variables may be considered. Univariate analysis is a widely used approach to unveil brain states. In contrast, multivariate analysis deals with all variables of the study at the same time, and is able to embed numerous variables of the study simultaneously.

The main advantage of using multivariate analysis in neuroscience is that it can detect activities which are otherwise not detectable by univariate analysis. Some examples of multivariate analysis approaches are pattern classification [49], Representational similarity Analysis (RSA) [50] and clustering [80].

Machine learning (pattern classification) is the common ground between multivariate pattern analysis in neurosciences and computer science. In the current thesis, I aim at suggesting novel ways of applying multivariate analysis, and in particular, machine learning techniques on neural data recordings. My goal is to provide with ways to enhance the resolution of brain exploration on the one hand, and to assist neuroscientists and neurologists in detecting and recognizing neural activity which is more difficult to perceive while using univariate analysis, on the other hand.

1.2 Motivation

Pattern classification and machine learning approaches are widely practiced in different computer science disciplines such as data mining, computer vision and speech recognition. In more recent years, pattern classification and in other words, multivariate pattern analysis, is widely employed in neuroscience, enabling neuroscientists to enhance the quality of neural decoding [51].

Being wholeheartedly interested in applying skills which I leaned from the realm of computer science to the filed of neuroscience, to not only find the answers to my philosophical questions in theory of mind, but also to further extend the edge of machine learning applications in new territories, made me to search for such scientific laboratory.

In department of Epileptology in Bonn, I have found the chance to access unique neural data recordings such as *intracranial Electroencephalography* (iEEG), that is the recordings from patients with implanted electrodes in their head, in addition to *functional Magnetic Resonance Imaging* (fMRI) recordings, of healthy participants.

Moreover, there was an open access to seizure monitoring recordings of epileptic patients. Thanks to the department being one of the world's leading epilepsy clinics, the well-known problem of non-invasive seizure detection has been always a subject of research. I received the opportunity to access invaluable data recorded from epilepsy patients.

Accessing to such invaluable data opened two doors of research for me, in which machine learning could be broadly utilized. First, on iEEG and fMRI data, a scientific question or hypothesis in cognitive neuroscience could be posed and machine learning techniques could discover answers to those question.

Secondly, I found an opportunity to help neurologists in developing seizure detection systems for epilepsy patients, again with the help of machine learning techniques to improve their quality of life.

These two scopes are discussed below in more details:

1.2.1 Scope 1: iEEG and functional MRI in working memory

Working memory refers to the temporary retention, processing and manipulation of information in the brain [22], in contrast to the short-term memory which is generally referred only to the act of information retention [29]. To conduct a working memory study, it requires designing an experimental paradigm in which, patients who participate in the study are asked to perform certain memory tasks while their brain signals are being recorded simultaneously. To acquire accurate brain signals in spatial and temporal terms, the most promising method of recording brain activity is intracranial EEG, which has had the advantage of recording from the subdurally implanted depth electrodes in the brains of epilepsy patients.

Multivariate pattern classification analyses should be designed to promote identifying distributed activity patterns along the electrodes [101]. The goal is to aim for the highest classification accuracy which can satisfactorily and significantly distinguish between the different conditions of maintaining an item in memory. The result of the analysis can be used as a template for further possible inferences in the field of cognitive neuroscience.

Additionally, functional magnetic resonance imaging (fMRI), a brain imaging technique, could also be used to measures brain activity by sensing the variation in blood flow in different brain structures. MRI images can be seen as four dimensional images, depicting the neural activities.

The study required acquiring data from healthy volunteers who participated in the study. The goal of this study was to look through and propose an alternative machine learning approach for multivariate pattern analysis [49] for fMRI data to classify the intended brain states significantly above random level.

Subsequently, investigating some theories such as

- Pattern changes of brain activities during learning
- Restoring memory of learning activity during recall

were the follow up steps of this study.

1.2.2 Scope 2: non-Invasive seizure detection

Millions of people suffer from epilepsy all over the world [4]. Due to the risk of seizures and subsequent loss of consciousness, a large proportion of them are prohibited to engage in activities like biking or driving, and they are not allowed to take occupations that may

pose a possible danger to them or to their co-workers, let alone jobs such as airplane pilot or taxi driver. Automatic seizure detection systems assist both patients and physicians in tracking the kind and quantity of patients' seizures, and help physicians to diagnose and prescribe their patients with the most suitable course of treatment [94], to ultimately increase the quality of life for the patients.

Seizures are the result of sudden and abnormal neural activity which prevents the brain from having full control over the body and mind. The effect of having seizure can be also sensed and measured by other kinds of sensory systems such as *electroencephalography* (EEG), *electrocardiogram* (ECG) [5], *electrodermal Activity* (EDA) sensors, or *electromyography* (EMG) [8]. Since seizures can cause involuntary muscle movements, this movement of body parts can also be registered by *accelerometry* sensors [92]. It has been shown also that most of seizures affect the blood pressure too [48].

A significant need for epilepsy patients is to have their seizures tracked. Epilepsy societies around the globe are interested to have an automated seizure tracking system since self-reporting seizure information from patients is not a reliable option to track epilepsy patients in the clinical trial [58]. Building a non-invasive and user-friendly seizure detection system requires a multimodal sensory system to make a complex model for comparison. In addition, machine learning techniques can be widely employed to develop a generalized model to yield such an assistive system.

Developing such systems are rather expensive and resource intensive. It requires recruiting epilepsy patients in one hand and recording and analyzing their epileptic activities in the other, to be able to gather sufficient and proper data. Having epilepsy experts and epilepsy patients together in the Bonn epilepsy clinic, has been a great chance for me for data collection and data annotation processes.

In the following sections, an introduction to the basics of human nervous system is given. In order to have a better insight into the nature of the problems discussed in this thesis, general knowledge regarding the brain and neuroscience is presented. Additionally, some terms which are used in the proceeding chapters are explained.

1.3 Human nervous system

In recent centuries, we witnessed a paradigm shift in the way people link the mind, the soul, and the brain. In ancient ages, it was a common belief among people that the heart is a medium for perception (how we perceive the surrounding world), a medium for cognition (how we think), a container for the soul, and also an organ for the mind [90]. It has been a common practice among civilizations to purify the heart as it believed to

be where ideas and perception occurred. Aristotle (384-322 B.C.), one of the influential philosophers in the history of humankind was among those who strongly promoted the role of the heart in shaping the mind [45].

In the renaissance era and those succeeding it, the soul (often described as a fluid encompassing a person's body) and the location of mind, has been repositioned from the heart to the head. Leonardo da Vinci (1453-1519) was among those who believed that the mind and the soul are stored in the cavities of the brain, but not in the brain substances [24].

In 17th century, the idea of dualism, that the mind and the body are fundamentally separate entities, was propounded mostly by René Descartes (1596-1650). He believed that the mind and the soul originate from the same substance and that they aid in subjective understanding of external world. The body on the other hand, in Descartes' view, had the same functionality as animals' brain. Descartes' most famous statement, *"I think, therefore, I am"*, was also rooted in his belief that the subjective understanding of the world is something that has been brought together with the spirit. He then puzzled with the idea that *"how can the mind and the body possibly interact?"* and consequently he suggested that the brain structure, *pineal gland*, is possibly where the mind and the body could communicate with another. What Descartes overlooked at was the fact that other animals have also the pineal glands in their brain and they too, are able to have some extent of rational thinking [86].

In the modern view of neuroscience, the brain is unilaterally responsible for perception and cognition the mind is nothing but a biological interactions between different brain structures. *Thomas Willis* (1621-1675), a pioneer physician and neuroscientist and *John Locke* (1632-1704), a philosopher, were among some of the people who changed the antique views of how and where perception and cognition take place. Over time, neuroscientists discovered that distinct regions of the brain are specialized for particular tasks covering a range of cognitive processes including perception. [44, 86].

Analogous to other animals, the brain also has an evolutionary history. That is, the size of the brain compared to body mass in early primates, used to be considerably smaller. Hence, some parts of the brain (e.g. cortex) are historically younger than the other parts and therefore, our superiority in our cognitive abilities, in contrast to other animals is rooted in relatively newer, more developed segments of the brain [64].

Neuroscience is primarily the science of studying neurodegenerative and brain related diseases. In present era, neuroscience is primarily being referred also to answering philosophical and psychological questions focused on the mind and consciousness, especially

concerning the way these entities interact with the external world. Neuroscience investigates the functionalities of bits and pieces of the brain, and putting together the gathered knowledge thus far, gives us a consistent view of how mind and and body interact. Given the comprehensive advancements in different fields of neuroscience, the general answer to the mind-brain interaction dilemma is: the brain is responsible and it is the origin of our physical and mental activities [64, 90]. And interestingly, this view is wholly compatible with the evolution theory, which heretofore, could not be proven otherwise.

1.4 Terms and Definitions

In humans and especially in vertebrates, in general, the nervous system is generaly classified into *Central Nervous System* (CNS) and *Peripheral Nervous System* (PNS). The CNS constitutes the brain and the spinal cord, and the PNS consists of the widespread projection of nerve cells throughout the body. Neurons are the building blocks of the CNS and peripheral nerves, are the components of the PNS. Since the complexity of the PNS is significantly less than that of the CNS, the main focus of this section is on the CNS and the brain in particular.

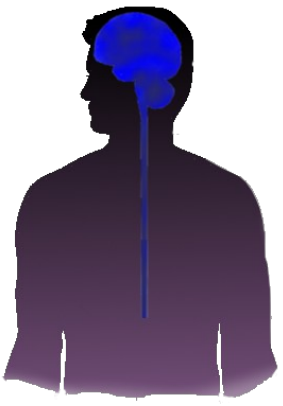


FIGURE 1.1: A schematic view of the central nervous system. The central nervous system is composed of brain and spinal cord. CNS is responsible for all voluntary and involuntary processes in term of actions and reactions to internal and external stimulations. Understanding the mechanisms of CNS is a great subject of research in the current century.

1.4.1 Terms used to describe the brain

Similar to our body which has two symmetric sides; our brain also has two symmetric halves. Each hemisphere of the brain contains structures similar to the other side. To address the brain regions, there are common terms which have been defined (which apply to both sides of the brain) to describe the brain and its structure.

Here are relative terms for describing the position of the brain regions:

- **Rostral**: towards the forehead (=anterior) vs.
- **Caudal**: towards the back of the head (=posterior)

- **Dorsal**: outside top to back of the brain vs.
- **Ventral**: inside underside of the head towards the base of the brain

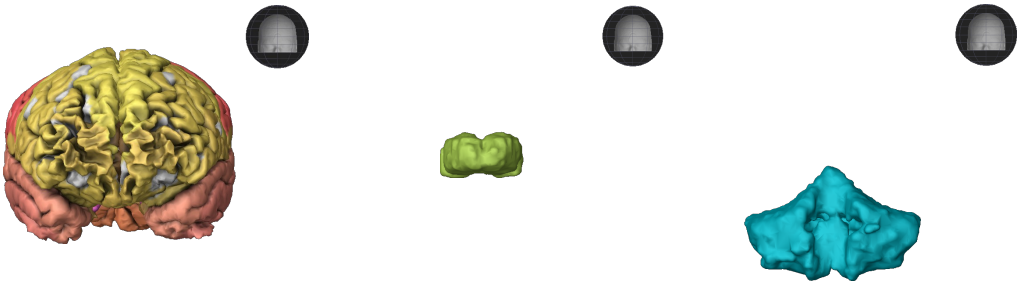
- **Medial**: towards the middle of the brain vs.
- **Lateral**: towards the outside of the middle brain

The brain can be largely divided into five subcategories from rostral to caudal:

- **Telencephalon**: is a large hemisphere that can be seen from the outside of the brain (most recently developed part of the brain). The outer part of the Telencephalon is called the *cortex* which is believed to host most of our voluntary actions.
- **Diencephalon**: is a segment of inner brain composed of the *thalamus* and *hypothalamus*. The thalamus literally means "waiting room", and functionally almost all of the information that goes to the cortex must pass first through the thalamus. The hypothalamus is responsible for some basic tasks such as keeping the temperature of the body (homeostasis etc).
- **Mesencephalon**: is the midbrain. One main function of midbrain is to generate reflexes such as closing eyes in sudden bright light.
- **Metencephalon**: is composed of the *cerebellum* and *pons*. The cerebellum is responsible for coordination of learned motor movements. The pones acts as a bridge to connect the cerebellum to the rest of the brain.

- **Myelencephalon**: is the connecting part of the spinal cord to the central nervous system. It is evolutionary, a very old part of the brain which is responsible for the communication between the brain and spinal cord.

Figure 1.2 depicts different levels of brain based on some of the described definitions.



(a) Telencephalon (b) Diencephalon (c) Metencephalon

FIGURE 1.2: Brain level classification from rostral to caudal. This classification is not the only way to classify the brain segments based on its functionalities, but it provides a rather historic classification of brain development. For each panel, there is a face direction indicator.

Here are some additional terms to describe parts of the brain:

- **Forebrain**: refers to the telencephalon together with the diencephalon and is the most recent part of the brain in its evolution.
- **Hindbrain**: caudal part of the brain containing the metencephalon and myelencephalon, and is involved in automatic actions such as heart rate and breathing.
- **Midbrain**: is basically referring to the mesencephalon segments.
- **Brainstem**: is the combination of mesencephalon, metencephalon, and myelencephalon. Structurally the forebrain encloses the brainstem.

The exterior surface of the brain is called *cerebral cortex*. Compared to other beings with central nervous system, cortex in humans is uniquely more developed, and it is responsible for all higher order processes such as language, memory and consciousness. The ridges that run over the cortex are called *gyri* and each valley on the surface of cortex is called a *sulcus*. Studies show that each gyrus on the cortex is specialized to process and execute certain type of tasks [64].

The cerebral cortex can be divided into four different segments (lobes) based on their functionalities.

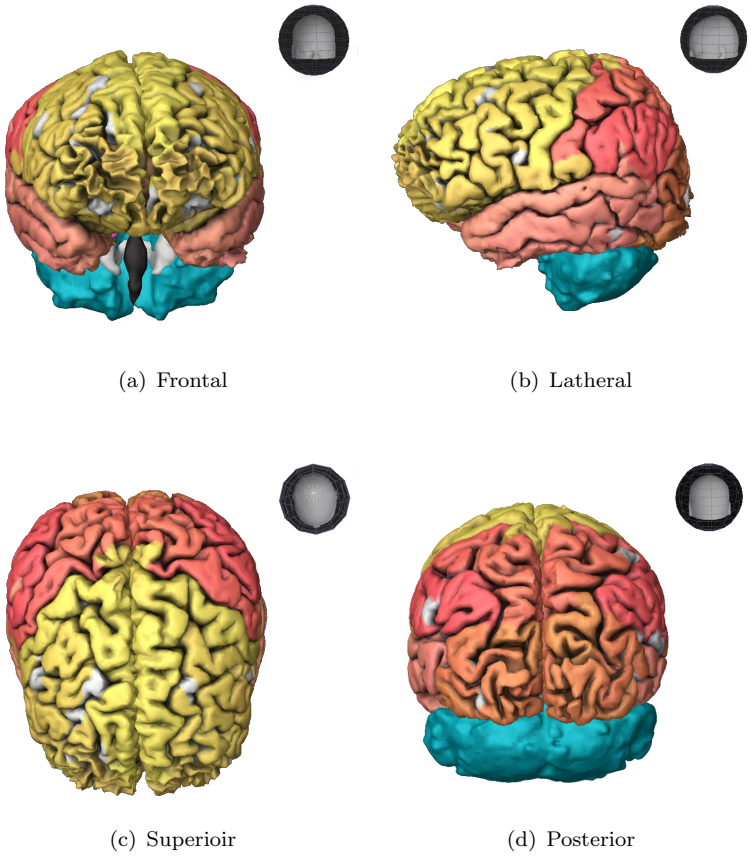
- **Frontal lobe**: is the anterior/rostral part of the brain and is related to higher level cognitive processes such as, rational thinking, short-term memory and voluntary motor movement. The frontal lobe is highly responsive to rewards.
- **Occipital lobe**: refers to the area at the back of the brain. Occipital lobe is dedicated to the processing of visual information. The visual input information collected from our eyes is projected to the gyri of visual cortex for object detection, from very basic to complicated shapes, and then higher level interpretations of visual information and stimuli.
- **Temporal lobe**: is positioned on the lateral and lower part of the brain and is primarily responsible for auditory and language processing. The *hippocampus* is a part of temporal lobe (medial temporal lobe). The hippocampus plays a crucial role in the formation and consolidation of short-term to long-term memories. One well-characterized structure of temporal lobe is the *fusiform gyrus* (FG), which is located on the edge of occipital cortex. A section of fusiform gyrus is called *fusiform face area* (FFA) and is responsible for human face detection.
- **Parietal lobe**: is located in the middle of the brain's surface. The parietal lobe is mainly responsible for tactile information processing such as pain, pressure, and touch. A section of parietal lobe is called the somatosensory cortex and is primarily in charge of processing the sense information received from the body.

A schematic view of the four lobes of the brain is shown in figure 1.3.

1.4.2 Building blocks of the Brain

If we cut the brain into thin slices, we will observe that some parts of the brain are darker than the others. The brighter parts are the parts that early neuroanatomists called *white matter* as opposed to the darker parts which were called *gray matter*. As other organs of human body, the brain is also comprised of cells. These speciallized cells are called the neurons (see image below 1.4).

A neurons is comprised of two primary parts: the *soma* or the cells body and the processes (*dendrites* and *axons*). Dendrites are the extensions of the cell body of a neuron. They receive electrochemical stimulation through numerous inputs from other neurons and conduct them to the soma. Dendrites play a critical role in integrating these inputs and in determining the extent to which action potentials are produced by the neuron. Given that the amount of collected input impulses reaches to a certain level, the electrical impulse can be traveled and projected through axons to other neurons and in general other parts of the brain.



(a) Frontal (b) Latheral

(c) Superioir (d) Posterior

FIGURE 1.3: A schematic view of central nervous system and lobes of the brain from four different perspectives. Different lobes are color coded: frontal:yellow, temporal:pink, parietal:red, occipital:brown, cerebellum:blue. Lobes are specialized to process higher order functions.

Networks of neurons will make neural nuclei and neural ensemble, to be able to compute and react to different stimuli. Brain is composed of numerous neural ensembles. A collection of cell bodies with their connected dendrites mostly yield what we see as the gray matter since it contain nuclei, and groups of axons shape the white mater regions.

1.5 Next chapters

Now after introducing the nature of our work and data, and also the terms which were used in the thesis, I will try to proceed with the main scientific questions we aimed to address with machine learning in this thesis.

In the next chapter, an introduction to machine learning and in particular, artificial neural networks is presented. The reason was two points: First, in recent years, artificial neural networks are the apple of the eyes of machine learning solutions. Second, since we

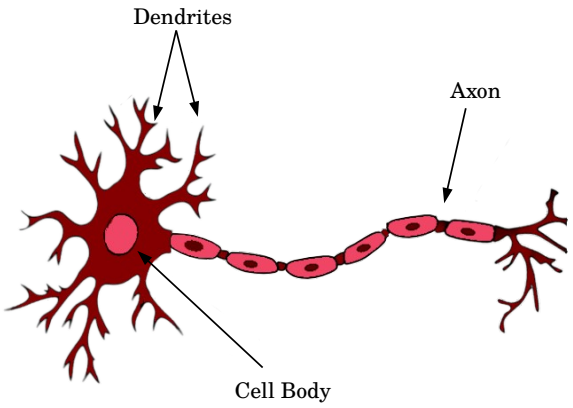


FIGURE 1.4: A schematic view of a neuron. Neurons are the building blocks of central nervous system.
The neuron has two main parts, the cell body and the processes (axons and dendrites). Dendrites detect electrical impulses in its neighboring surroundings. In case the amount of collected input impulses reached to a certain threshold, the electrical impulse can be projected through axons to other parts of brain.

deal with natural neural data, it makes sense to introduce artificial neural networks and briefly compare the way both function. Additionally, a comparison of the performance of classical machine learning methods versus artificial neural network is given in some later sections, to provide us with an idea, whether or not should one apply artificial neural networks on real neural data.

In chapter 3, through some machine learning studies which I have conducted on neural data (iEEG), I present a way in which machine learning can be used to gain an understanding of CNS functions. Two studies were presented in detail and the results are discusses accordingly.

In chapters 4 and 5, through performing multiple clinical studies, I demonstrate some machine learning methods to help epilepsy patients improve their quality of life. The data presented in these chapters were obtained from a lengthy and expensive clinical processes of patient recruitment and their brain recordings. By the end of this chapters, I discuss our achievements and also the limitations of the study.

The final part of each chapter holds a summary. They provide a concise and quick review of the main idea in case the reader intends to skip over technical details.

Chapter 2

Artificial neural networks

2.1 Introduction

Although we have used established machine learning steps throughout the thesis, by reemerging of neural networks in current years, additionally, I have developed and examined some versions of (artificial) neural network, to address our problems with novel state of the art solutions. The current chapter presents the core concept of neural networks and in particular *deep* neural networks. In the next two chapters, even though complete classical machine learning solutions were presented, a short reference to deep neural networks is also presented at the end to give an idea of how feasible the use of deep neural network is with respect to our neural data.

Artificial neural networks are among pioneering machine learning techniques. These techniques were inspired from the way our nervous system works and received very much recognition in 90s. Although we consider the neural networks as today's cutting edge technique, in the realm of machine learning, they had become obsolete to a great extent for almost two decades by other techniques such as SVM [19]. There were two main technical reasons involved. The first problem was the computational capacity of the computers at the time. In the training phase of neural networks compared to other established machine learning algorithms, more parameters needed to be learned. While the learning process of neural networks is very resource intensive, people could not afford having large and complex neural networks and have them trained. The second reason was the *over-fitting* problem. Later in this chapter, I present some fundamental concepts of neural network to address this issue but for now to explain this point, neural networks are/were very prone to fit exceedingly to the training examples and not being able to generalize well to unseen samples if no counteracting measure is used.

In the last few years, neural network is back to the market by addressing and solving those very two problems [25, 43, 54], caused it to surpass the accuracy of other machine learning techniques and even in some cases the human intelligence [33, 72]. Having clusters of powerful and parallel processing units such as CPUs and GPUs, solved the processing limitations of neural network training. Especially GPUs which were designed primarily for processing of graphical objects, are shown to be a perfect processing platform for matrix-wise computations which is highly needed for neural network training process. Having such platforms, complex networks can be trained in a reasonable time. The over-fitting problem is also solved by the introduction of some training tricks which prevented them from being overfit, by technically adding noise to the network while training! In the following, I explain how a basic neural network functions.

2.1.1 A basic neural network

A primary question in machine learning is to find out a line which can separate two kinds of data. Imagine we want to differentiate two class of variables, and we know two features of each type, x_1 and x_2. Then in a two dimensional Cartesian space, we should look for a line to split the two types best (see image 2.1).

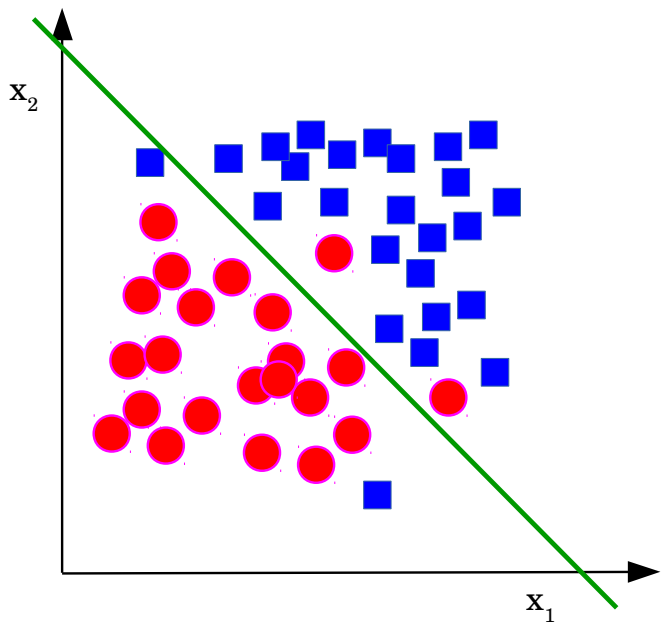


FIGURE 2.1: A simple line to separate two entities. X_1 and X_2 are two different features of an entity.

To find such line, we can formulate the problem as below:

$$w_1 x_1 + w_2 x_2 + b = 0 \tag{2.1}$$

where w_1 and w_2 are the weight of x_1 and x_2 respectively, and b is the intercept. To solve the problem, our goal is to discover good candidates for w_1, w_2, and b.

Given having those values obtained, by putting the dimensions of a test point in the equation, we get either a positive or a negative value. For y' as a test case, we have:

$$y' = \begin{cases} 0 \text{ (Red circle)} & \text{if } w_1 x_1 + w_2 x_2 + b \geqslant 0 \\ 1 \text{ (Blue square)} & \text{if } w_1 x_1 + w_2 x_2 + b < 0 \end{cases} \tag{2.2}$$

The mentioned equation is called *class boundary* and can be represented in neural networks by a perceptron, the building block of neural networks.

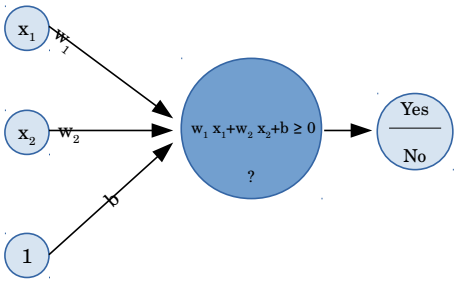


FIGURE 2.2: A two dimensional perceptron. Percepteron is the building block of neural networks. In this example, the perceptron represents a splitting line of two classes in a two dimensional space.

The structure of perceptron resembles the building block of our nervous system, the neurons. In principal, they both sense the input signal and map it to the output signal in a form of fire|not-fire (see figure 2.3).

The core of the perceptron as shown in figure 2.2, can be split into two processing segments: the summation of weighted inputs and the *step function* which is basically 0 if the summation result is less than 0, and is 1, if the summation result is greater than 0 (see figure 2.4). The step function is also called: *activation function*.

We can abbreviate the decision boundary equation by replacing $W = (w_1, w_2, ...)$ and $x = (x_1, x_2, ...)$. Therefore, we have:

$$Wx + b = 0 \tag{2.3}$$

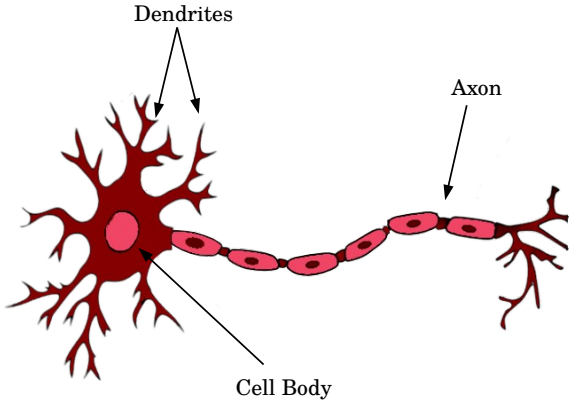


FIGURE 2.3: A schematic view of neuron. Dendrites are the input terminals of a neuron. Their summed output is projected along the axon if the input signal is strong enough to be relayed ahead.

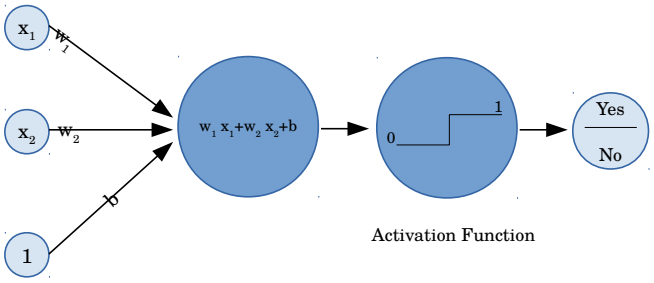


FIGURE 2.4: A two dimensional perceptron. The step function determines whether the output is mapped mapped to one or to zero.

and

$$y' = \begin{cases} 0 \text{ (Red circle)} & \text{if } Wx + b \geqslant 0 \\ 1 \text{ (Blue square)} & \text{if } Wx + b < 0 \end{cases} \tag{2.4}$$

where the term Wx is the dot product of vector W and vector x.

2.1.2 Perceptron algorithm

The perceptron algorithm is devised to direct a percetron to discover proper weights and a proper bias for its line equation. According to the perceptron algorithm, it starts with

initiating random weights and a random bias and then, drawing the line and classifying the points.

Next, it checks the misclassified points based on the drawn line. To correct the line for obtaining a better classification accuracy and to decrease the number of misclassified points, it checks the classified values. Given that the prediction for a misclassified point was 0, for each weight element in W, it updates each weight w_i by adding a fraction of its corresponding variable x_i to it. Otherwise for a misclassified point with 1 value, it subtracts the same amount from w_i.

The multiplied fraction to update the weights is know as *learning rate* and is represented by α. Having the learning rate value, causes the weights to be updated gradually. In this way, through iterations, the line comes gradually closer to the misclassified points and passes over them. Perceptron algorithm does not however update the wights and the bias for correctly classified points. The algorithm is summarized below in algorithm 1.

Algorithm 1: Perceptron algorithm.

Data: Multi dimensional data points
Result: Weights and bias for perceptron

$n \leftarrow$ dimension of input data;
$W \leftarrow$ Randomly generated vector real values of length n;
$b \leftarrow$ Randomly generated real number of length n;
$\alpha \leftarrow$ some fractional value e.g. 0.01
With W and b, make a perceptron and predict the class for input data.
for *every misclassified points* $x = (x_1, x_2, ..., x_n)$ **do**
    if *prediction = 0* **then**
        for $i = 1 : n$ **do**
            $w_i = w_i + \alpha.x_i$;
            $b = b + \alpha$;
        end
    end
    else
        for $i = 1 : n$ **do**
            $w_i = w_i - \alpha.x_i$;
            $b = b - \alpha$;
        end
    end
end

The process in which the parameters of perceptron is tuned up is called *learning* or *training*. Learning happens by comparing the true values of the data samples used for wights updating to the predicted values of the perceptron algorithm. Although the way the perceptron algorithm is designed, seems to find the optimum weights, in practice, we need a better solution. Since the output of the perceptron is discrete, $(0, 1)$,

it can easily happen that the weight correction process bounces back and forth but without any beneficial update. Therefore, we need a continuous function to be able to calculate a direction for weight updating. Thus, we should replace the step function with some continuous alternative. In a classical neural network, typically a *sigmoid* function, $\sigma(x) = \frac{e^x}{e^x + 1}$, is used as continuous activation function (see figure 2.5). The sigmoid function maps the input data to a range from 0 to 1 and therefore, it simulates also a probability function. Nonetheless, others such as *hyperbolic tangent*, *Rectified Linear Unit* (ReLU), ... can be employed as well. The activation function plays an important role in the generalization of neural network. More about it comes later.

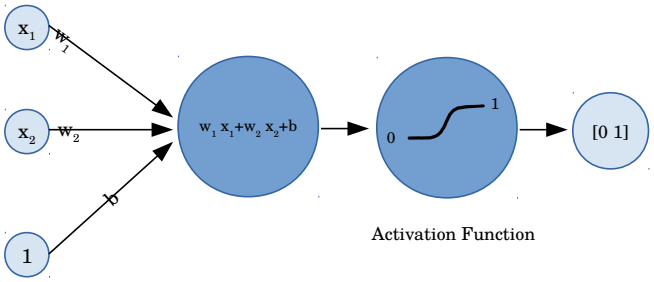


FIGURE 2.5: A two dimensional perceptron with sigmoid activation function. The output of the perceptron in contrast to figure 2.4 is a probability values in the range of [0 1].

To assess how good a perceptron with a continuous activation function predicts the actual values, it is essential to define a function so that it tells us how close the perceptron predicts the input data to the output data. This function is known as *error function*.

2.1.3 Error function

An error function during a leaning process indicates how far we are from an ideal solution. It then helps the leaner algorithm to find its path towards the solution, in a way in which the error decreases continuously. To find the way the learning algorithm should move (= the way to change the learning parameters), we can use the derivation of our error function of choice. Thus, the error function should be differentiable. The direction that "negative of the derivative of the error function" pinpoints, is the way to update the weights.

Let's start from the fact that from a learning algorithm, we expect the best predictions. This can be translated to obtaining the best aggregated probability for all learning samples. That is, the more all predictions are close to their actual classes, the better. One way to obtain the aggregated probability of all events is to measure the product of all probabilities resulted from feeding samples to the perceptron:

$$aggregated_{prob} = prob(sample_1) \times prob(sample_2) \times prob(sample_3) \times ... \tag{2.5}$$

and the goal is to find a setting in which the aggregated probability value is maximized. This concept is known as *maximum likelihood estimation* in literature [30]. The higher the aggregated probability can be, the better the model (perceptron) is for pattern classification.

$$max_{probablity} = max(all\ aggregated_{probs}) \tag{2.6}$$

Having multiplication in the formula, makes the aggregated probability very much prone to noise, as an extreme probability value can change the total product drastically. A method to escape the multiplication problem is to convert it to a summation. For this sake, we can calculate the *log* of products:

$$log(prob(sample_1) \times prob(sample_2) \times ...) = log(prob(sample_1)) + log(prob(sample_2)) + ... \tag{2.7}$$

The results will be a summation of small negatives values since the probabilities are in the ranges of [0 1]. Adding a minus sign to the formula turns it to a positive value. From it, we can introduce the following formula known as *cross-entropy* which is used widely in machine learning algorithms to evaluate the learning process:

$$\text{cross-entropy} = -\frac{1}{m}\sum_{i=1}^{m} y_i\, log(p_i) + (1 - y_i)\, log(1 - p_i) \tag{2.8}$$

where m is the number of samples, y_i is the true label of the sample, and p_i is the estimated probability of sample i. Please note that the above formula is meant to be for binary classifications cases. Please also pay attention to the fact that for event probabilities, we have to consider the probability of all possible events. Therefore, we consider the y_i ($y = 1$ and $y = 0$ for binary example) to cover all sample points from all classes. By measuring the cross-entropy, events with high probability have lower cross-entropy and events with lower probability have higher cross-entropy.

Corss-entropy provides the framework we look for an **error function**. It is firstly differentiable, and also a lower cross-entropy implies a better solution and being closer to the goal. The cross-entropy can be generalized to the following formula for multi-class learning cases:

$$\text{cross-entropy} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij}\, log(p_{ij}) \tag{2.9}$$

where m is the number of samples, and n is the number of classes. By using cross-entropy, our goal for maximizing the probability, had changed to minimizing the cross-entropy.

It should be noted that other error functions such as *mean squared error* can be equally used instead of cross-entropy. The main message here is to have a good candidate for error function to be able to assess the learning progress.

2.1.4 Weight updating and logistic regression

We have learned in the last section that the negative of the first derivative of the error function, shows the direction, the weights should be updated. That is, measuring the partial derivative of the error with respect to a particular weight, will be used to update that very same weight (increasing or decreasing the weight value). This technique is known as *Gradient Decent* in literature since in a hypothetical hyperplane of error with respect to weights, we move towards the valleys of the hyperplane. In this section, we introduce the *logistic regression* algorithm which is designed for the sake of weight updating. The code snippet 2 shows the logistic regression algorithm.

Algorithm 2: Logistic regression algorithm.

Data: Multi dimensional data points
Result: Weights and bias for perceptron

```
n ← dimension of input data;
W ← Randomly generated vector real values of length n;
b ← Randomly generated real number of length n;
α ← some fractional value e.g. 0.01 ;
Error ← cross-entropy of σ(Wx + b) ;
while Error is high do
    for every points x = (x_1, x_2, ..., x_n) do
        for i = 1 : n do
            w_i = w_i + α.∂E/∂w_i ;
            b = b + α.∂E/∂b ;
        end
    end
    Error ← cross-entropy of σ(Wx + b)
end
```

The primary difficulty in the algorithm 2 is to measure the partial derivative of error. Since for perceptron, our activation function is sigmoid function, $\sigma = \frac{e^x}{e^x + 1}$, we have:

$$\begin{aligned}
\sigma(x)' &= \frac{\partial}{\partial x}\frac{e^x}{e^x+1} \\
&= \frac{e^x(e^x+1)-e^x e^x}{(e^x+1)^2} \\
&= \frac{e^{2x}+e^x-e^{2x}}{(e^x+1)^2} \\
&= \frac{e^x}{(e^x+1)^2} \\
&= \frac{e^x}{e^x+1}\frac{1}{e^x+1} \\
&= \sigma(x)\frac{1}{e^x+1} \\
&= \sigma(x)(1-\sigma(x))
\end{aligned} \tag{2.10}$$

which is the derivative of the sigmoid function. Now, please remember again that our goals is to calculate the $\frac{\partial E}{\partial w_i}$ and also that the estimated value of perceptron's output is measured as $\hat{y} = \sigma(Wx+b)$. Thus, for measuring the $\frac{\partial}{\partial w_i}\hat{y}$, we have:

$$\begin{aligned}
\frac{\partial}{\partial w_i}\hat{y} &= \frac{\partial}{\partial w_i}\sigma(Wx+b) \\
&= \sigma(Wx+b).(1-\sigma(Wx+b)).\frac{\partial}{\partial w_i}(Wx+b) \\
&= \hat{y}.(1-\hat{y}).\frac{\partial}{\partial w_i}(Wx+b) \\
&= \hat{y}.(1-\hat{y}).\frac{\partial}{\partial w_i}(w_1x_1+...+w_ix_i+...+w_nx_n \\
&= \hat{y}.(1-\hat{y}).x_i
\end{aligned} \tag{2.11}$$

Now, we can go ahead and measure the $\frac{\partial E}{\partial w_i}$ by plugging in the cross-entropy formula:

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}(-ylog(\hat{y})-(1-y)log(1-\hat{y})) \\
&= -y\frac{\partial}{\partial w_i}log(\hat{y})-(1-y)\frac{\partial}{\partial w_i}log(1-\hat{y}) \\
&= -y\frac{1}{\hat{y}}\frac{\partial}{\partial w_i}\hat{y}-(1-y)\frac{1}{1-\hat{y}}\frac{\partial}{\partial w_i}(1-\hat{y}) \\
&= -y\frac{1}{\hat{y}}\hat{y}(1-\hat{y})x_i-(1-y)\frac{1}{1-\hat{y}}(-1)\hat{y}(1-\hat{y})x_i \\
&= -y(1-\hat{y})x_i+(1-y)\hat{y}x_i \\
&= -yx_i+y\hat{y}x_i+\hat{y}x_i-y\hat{y}x_i
\end{aligned} \tag{2.12}$$

and this is something very interesting. It tells us that the gradient of error for every coordinate x_i can be measured by a simple difference of the actual label and the predict label $(y - \hat{y})$. Equivalently for the bias, we can prove that $\frac{\partial E}{\partial b} = -(y - \hat{y})$. Then, for the gradient of error function, we will have:

$$\begin{aligned} \Delta E &= (-(y - \hat{y})x_1, ..., -(y - \hat{y})x_i, ..., -(y - \hat{y})x_n, -(y - \hat{y})) \\ &= -(y - \hat{y})(x_1, ..., x_i, ..., x_n, 1) \end{aligned} \tag{2.13}$$

Thus, we can now update the logistic regression algorithm as follows:

Algorithm 3: Logistic regression algorithm, updated.

Data: Multi dimensional data points
Result: Weights and bias for perceptron

```
n ← dimension of input data;
W ← Randomly generated vector real values of length n;
b ← Randomly generated real number of length n;
α ← some fractional value e.g. 0.01 ;
Error ← cross-entropy of σ(Wx + b) ;
while Error is high do
    for every points x = (x_1, x_2, ..., x_n) do
        for i = 1 : n do
            w_i = w_i + α.(y − ŷ).x_i ;
            b = b + α.(y − ŷ) ;
        end
    end
    Error ← cross-entropy of σ(Wx + b)
end
```

The logistic regression algorithm repeats the wight update process until the error would be tiny. Each repetition is called *epoch* in the jargon of machine learning.

Logistic regression algorithm similar to perceptron algorithm, is designed to update the weights and tune the decision boundary. However, it differs in a couple of point from perceptron. The output of logistic regression prediction is a value in the range of 0 and 1, whereas the perceptron's prediction which is a discrete value of either 0 or 1. Secondly, the weight updating in logistic regression is applied to all points in contrast to that in perceptron which applied only to the misclassified points.

2.1.5 Multi-layer perceptron

Although perceptron provides a solution for some classification problems, it is limited with its linearity, as it can not split the samples which are nonlinearly distributed. A

remedy to the perceptron linearity problem is to combine multiple perceptrons. This mimics the way our biological neural network functions. Neurons which can be thought of having a binary output, go/no-go, can be combined to shape a neural network in order to decide firing on highly non-linear decision processes. We can use a third perceptron to combine two individual perceptrons (see figure 2.6).

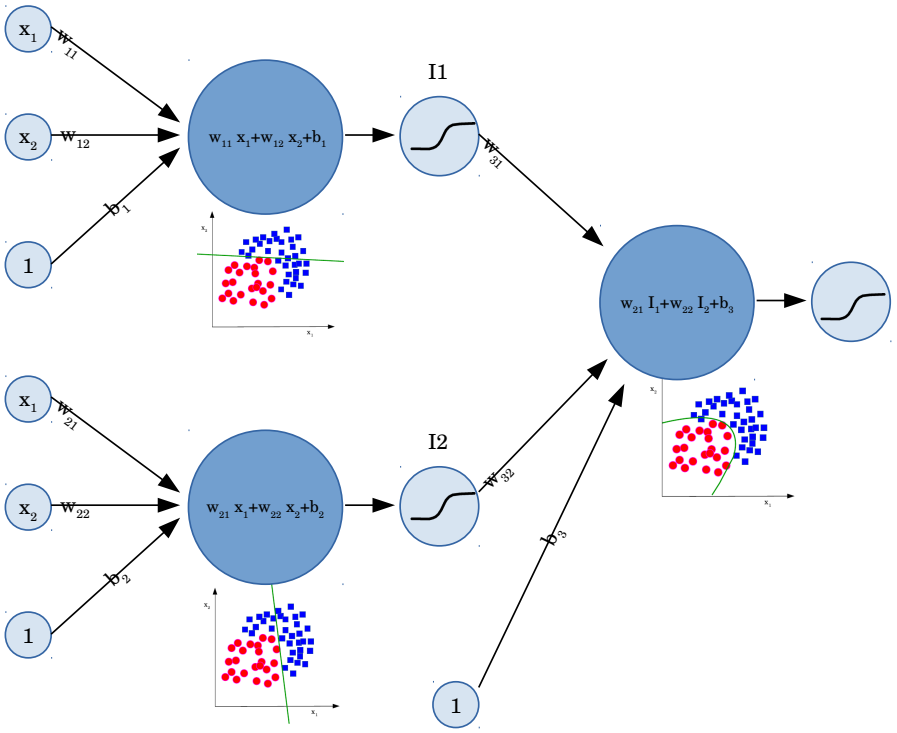


FIGURE 2.6: A simple network of perceptrons. A third perceptron can be used to combine two perceptrons by accepting their outputs as its input. The perceptrons on the left side are called input layer perceptrons and the one on the right is called first hidden layer perceptron.

The above representation of perceptron can be abbreviated in the following notion. To be able to achieve highly non-linear decision boundaries, we can add more columns of perceptrons to the network. Every column we add to the network is called a layer. The first column is known as *input layer*, those to its right are called *hidden layers* and the one to the far right is named *output layer*. These will shape a classical artificial neural network (see figure 2.7).

A question which may arise at this point is: what about the multi-class classification?". The answer is simple with applying a trick. For multi-class classification cases, the number of nodes in the output layer must be the same as the number of classes. That is, if we have four classes, we have also four nodes in the output layer. Then, for the training process, we transform the class labels from $0, 1, 2, 3, ...$ notion to a binary notion, in which for every class, we have a unique sequence of zeros and only a single one:

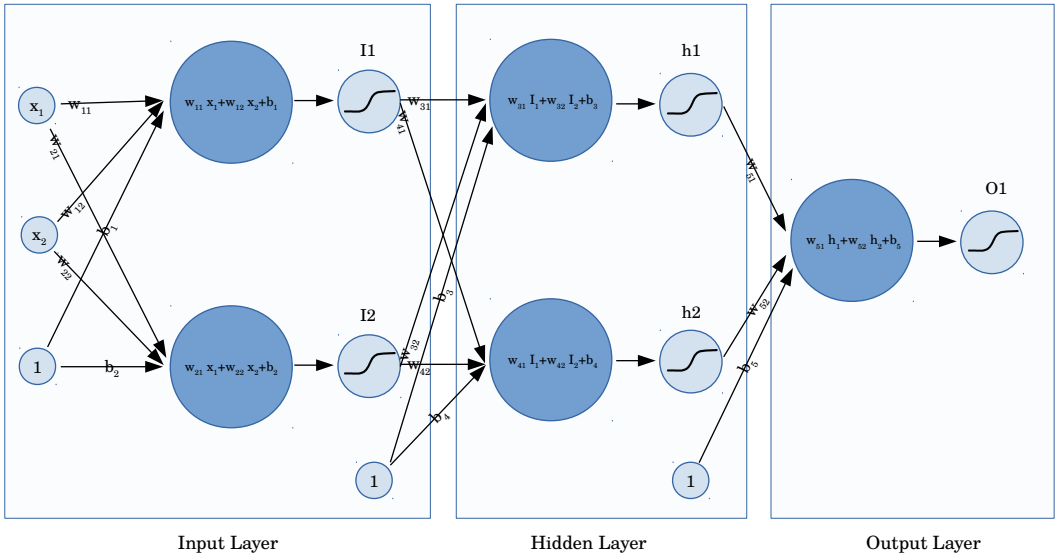


FIGURE 2.7: The input to the perceptron can be illustrated also as above. We can add more layers of perceptron to build up a network of peceptrons. Adding more layers, results in more non-linear decision boundary of the network. The first layer to the left is called input layer. The layers to the right of input layer are called hidden layer as it seems that they are hidden to the outside of the network. The rightmost layer is named output layer.

$$
\begin{array}{cccccc}
0 & \rightarrow & 0 & 0 & 0 & 1 \\
1 & \rightarrow & 0 & 0 & 1 & 0 \\
2 & \rightarrow & 0 & 1 & 0 & 0 \\
3 & \rightarrow & 1 & 0 & 0 & 0
\end{array}
$$

This technique is called *one-hot encoding*.

2.1.6 Feed-forward

Compared to perceptron, in a more complicated network of perceptrons, the input data must be fed to the input layer and its results must be gone through the entire network until reaching the output layer. The process of feeding data from input layer all the way up to the output layer is called *feed-forward*. As soon as the data reaches to the output nodes, the class probability can be measured and therefore, the data can be classified (see figure 2.8).

We can formulate the feed-forward procedure of the network as a sequence of linear model combinations and hence, we are able to depict it as a sequence of matrix multiplications:

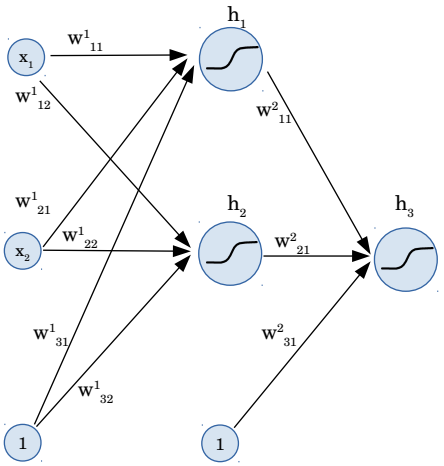


FIGURE 2.8: Feed-forward process. The data is fed to the input node and is propagated through the network until it reaches to the output nodes.

$$\hat{y} = \sigma\left(\left[\sigma\left(\begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix} \times \begin{bmatrix} w_{11}^1 & w_{12}^1 \\ w_{21}^1 & w_{22}^1 \\ w_{31}^1 & w_{32}^1 \end{bmatrix}\right), 1\right] \times \begin{bmatrix} w_{11}^2 \\ w_{21}^2 \\ w_{31}^2 \end{bmatrix}\right) \tag{2.14}$$

It is also possible to show the above equation as a sequence of dependent functions:

$$\hat{y} = \sigma \; o \; W^2 \; o \; \sigma \; o \; W^1(x) \tag{2.15}$$

For larger networks, we will have longer matrix multiplications. By performing feed-forward on a neural network, we can classify our test data but still we lack the ability to train and improve the weight for our network to achieve very low prediction error. Here again, similar to the single perceptron case, we require an error function to estimate how good we are doing with feed-forward process.

2.1.7 Back-propagation

To complete the training procedure, we need a complementary mechanism to measure the error for the entire network and update the weights. The error in the output layer can be easily measured as described for a single perceptron, and the weights can be updated by measuring the negative of the gradient of error with respect to the output weights.

For a multiple-layer perceptron network, we have the same error function, E, as before but here the $\hat{y}$ is a complex function:

$$\begin{aligned} &E(W) = -\tfrac{1}{m} \sum_{i=1}^{m} y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i) \\ &\hat{y} = \ldots \; \sigma \; o \; W^2 \; o \; \sigma \; o \; W^1(x) \end{aligned} \tag{2.16}$$

Thus, to discover the way the weight needed to be updated, the gradient should be calculated by measuring all partial derivatives of the error function E with respect to the weights:

$$\Delta E = (..., \frac{\partial E}{\partial w_j^i}, ...) \tag{2.17}$$

As mentioned above, we can simply measure the gradient of the error for the weights of output layer. Nonetheless, the gradient with respect to the other weights can also be calculated by employing the *chain rule* from mathematics. According to the chain rule, if $A = f(x)$ and $B = g(A) = gof(x)$, then for $\frac{\partial B}{\partial x}$ we have:

$$\frac{\partial B}{\partial x} = \frac{\partial B}{\partial A} \times \frac{\partial A}{\partial x} \tag{2.18}$$

We can use the same principle to measure the partial derivative of the error with respect to the weights in the other layers than the output layer. Since the output nodes can be written as a function of input and hidden nodes (as seen in feed-forward), we can also calculate the derivative of the error with respect to every wights in the hidden and input layer.

Back-Propagation is in principle the inverse of feed-forward, in which we spread the measured error from the output layer towards the input layer. As described above regarding the chain rule, we are able to measure the gradient of the error from the gradient measured in the previous layers to the right of it. Therefore, we would have:

$$\begin{aligned} &\vdots \\ &\tfrac{\partial E}{\partial w_{11}^2} = \frac{\partial E}{\partial \hat{y}} \times \frac{\partial \hat{y}}{h_3} \times \tfrac{\partial h_3}{\partial w_{11}^2} \\ &\vdots \\ &\tfrac{\partial E}{\partial w_{11}^1} = \frac{\partial E}{\partial \hat{y}} \times \frac{\partial \hat{y}}{h_3} \times \tfrac{\partial h_3}{\partial h_1} \times \tfrac{\partial h_1}{\partial w_{11}^1} \\ &\vdots \end{aligned} \tag{2.19}$$

The above calculations will provide all gradients of the error function with respect to the weights. Having them, we can perform the wights updating process similar to the logistic regression algorithm 3. And similarly, we iterate over the feed-forward and back-propagation sequence until the error is low.

With the so far described neural network (classic artificial neural network), it is possible to build a deep neural network and try to train it and classify the data. However, it has been shown theoretically and empirically in the last decades that the classification results of such neural networks has not been competitive with others (common knowledge). The reason is, there are a number tricks needed to be considered to train a deep neural network, and those tricks are the ones that gave rise to deep neural network.

2.2 Modern deep neural networks

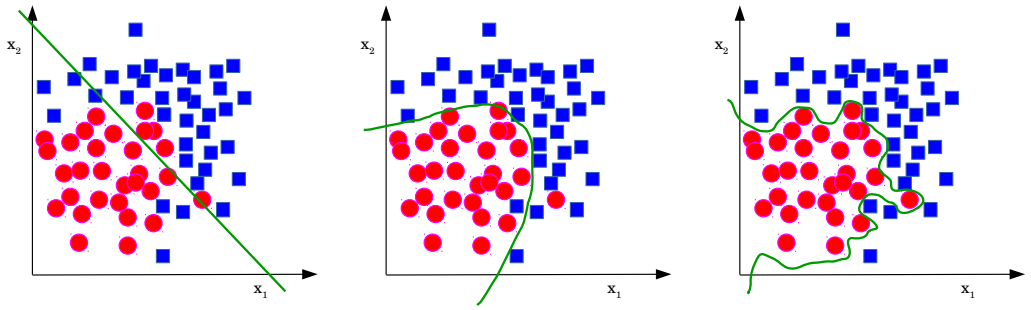
Classical artificial deep neural networks suffered from two main points. Initially, there was no capable computer in the past decades to bear the burden of calculating too many variables (weights) and carrying out many repetitions. A modern deep neural network may took years to be trained on older computers. The second reason was the problem of over-fitting which I describe in the following.

2.2.1 Overfitting

Appending more layers to a neural network is equal to shaping more complex decision boundaries. Complex and highly nonlinear decision boundaries work appropriately only for the data which the network is trained for. The goal of training a neural network is to be used for predicting unseen data. Given that the network has a very complex separating hyperplane, the test data could be easily misclassified. This problem is called overfitting problem and is a well-discussed problem in the realm of machine learning (see figure 2.9 for more details). The overfitting concept can remind us of the famous saying of British philosopher Bertrand Russell: "The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts."

In the same spirit, a good classification model is not the one that is shortsightedly stock to the information it has ever seen, but the one that has left some room for alternatives.

There are some tricks to avoid over-fitting of neural networks and we study some of them below.



(a) Underfit (b) Goodfit (c) Overfit

FIGURE 2.9: The Model to the left, split the feature space as simple as possible with multiple mistakes (underfitting). The model in the middle separates the point fairly good with minor mistakes. The model to the right does a perfect job to split the data points (overfitting). While it may look counter-intuitive, the model in the middle is much preferred compared to the other models. In the case of facing new data points, the overfit model fails to classify it better than the middle model since it is only optimized for the data which it has seen before. The underfit model also does a poor job in testing due to the lack of adequate complexity.

2.2.1.1 Early stopping

Learning is obtained through repetition in the neural networks. That is, in most of the cases, more repetition leads to more weight updates. While repetition is a crucial part of neural networks learning procedure, performing it for too many times causes overfitting. Therefore, we have to find an optimum number of repetition for any network we train. One way to tackle this problem is to use a testing set called *validation set*. The network should be trained on a handful of data points, *training set*, and must be tested with the validation set by keeping track of the number of repetitions. From such procedure, we can plot two error curves, one for the training set and the other for the validation set. This plot is called *model complexity graph* (see figure 2.10). From the model complexity graph, we can discover the optimum number of repetition in which the validation test error is on its minimum.

2.2.1.2 Regularization

A particular line or a hyperplane which is represented by a learning model can potentially have multiple similar formula (weights). For instance, the equation $x_1 + x_2 = 2$ and $10x_1 + 10x_2 = 20$ both depict the same line. However, if we measure the $\sigma(x_1 + x_2 = 2)$ and $\sigma(10x_1 + 10x_2 = 20)$ for a given input point, the first equation gives us a less confident prediction while the latter predicts the point with probabilities near to zero or near to one. Therefore, the latter equation is more prone to overfitting.

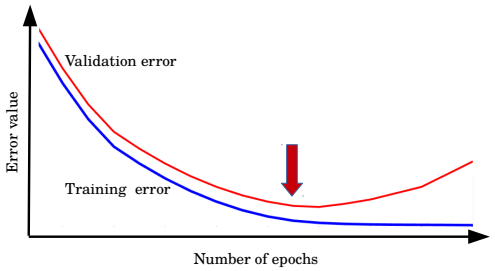


FIGURE 2.10: Model Complexity Graph. Two curves are plotted in the graph, the training error curve and the validation test error curve. They are plotted with respect to the number of training repletions. In this settings, we increase the number of repetitions and measure the classification error for both sets. We choose the optimum number of repetition as the point in which the validation test error is at its minimum. It is called early stopping since we stop right before we are encounter overfitting.

To avoid such scenarios, a technique known as *regularization* is devised to penalize and reduce the weights by adding a constant value λ to the error function. Two well-known types of regularization are L_1 and L_2 regularization and are defined as follows:

$$L_1 \rightarrow \ E(W) = -\frac{1}{m}\sum_{i=1}^{m} y_i\, log(\hat{y}_i) + (1-y_i)\, log(1-\hat{y}_i) + \boldsymbol{\lambda}(|\mathbf{w_1}| + |\mathbf{w_2}| + ... + |\mathbf{w_n}|) \tag{2.20}$$

$$L_2 \rightarrow \ E(W) = -\frac{1}{m}\sum_{i=1}^{m} y_i\, log(\hat{y}_i) + (1-y_i)\, log(1-\hat{y}_i) + \boldsymbol{\lambda}(\mathbf{w_1^2} + \mathbf{w_2^2} + ... + \mathbf{w_n^2}) \tag{2.21}$$

L_1 regularization tends to eliminate the features (weights) which are not important in the classification process and therefore, results in a sparse vector for weights. L_1 is used also as a tool for feature selection. In contrast, L_2 regularization reduces the weight uniformly and is more favorable for learning processes. Applying either of the regularization techniques will help to prevent overfitting caused by large weights.

2.2.1.3 Dropout

An effective technique to alleviate the overfitting dilemma is to use *dropout* technique. In dropout, during the learning process, some nodes of the network will be turned off. That is, with a particular probability (e.g. 0.15), a node in an epoch of learning will be deactivated. This will give a chance to the weaker paths to be traversed and therefore,

results in a more homogeneous network weights. The dropout technique will ultimately lead to better generalization and less overfitting.

The dropout can be also seen as a process for tricking the training process by adding noise to the network and avoiding early convergence, to find out a consistent generalized weights for the network. Dropout is used vastly in modern artificial neural networks.

2.2.1.4 Avoiding local minima

Since the error function is a progressive function and it acts based on its own previous steps, it can be the case that the error function has some local minima and the function traps in one of them while better minima are still available. To avert such scenarios, a trick is to start the error function from different points. Technically, this is equal to randomize the weights and train the network again and again. This, in turn, will help to bypass the local minima and trace other paths to the global minimum.

Another algorithm for avoiding local minima is called *momentum*. Momentum algorithm memorizes the gradient of the error in previous steps and then, penalizes the current gradient based on them. According to the momentum algorithm, the previous gradient values will affect the current gradient if it is getting close to zero, by forcing it to fly over local minima. Nonetheless, for the case of global minimum, the error function will eventually approach to it, even if it ever jumped over it.

2.2.1.5 Vanishing gradient problem

The error function of neural networks is measures based on the gradient of activation function. Provided that the gradient is small, then in case of having multi-layer networks, the product of small gradient functions will be exceedingly small. This is known as *vanishing gradient problem* in literature. Vanishing gradient causes the nodes of the first layers of the network to receive small value for the gradient of error function and consequently, having minute updates. To avoid this problem, other activation functions are introduced to substitute the sigmoid function.

2.2.1.6 Alternative activation functions

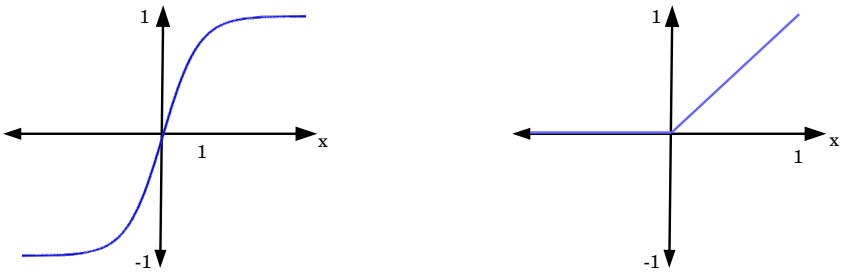
To address the vanishing gradient problem, other activation functions are introduced in recent years. Since sigmoid function has a gradient near zeros on its outer sides, kinds of activation functions are required that their gradient are larger compared to sigmoid. An alternative activation function is *hyperbolic tangent* and is defined as follows:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.22}$$

The other modern activation function is called *Rectified Linear Unit* (ReLU) and has a simple definition as below:

$$relu(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.23}$$

Unlike its simplicity, the ReLU function can easily construct very complex decision boundaries for neural networks. The pictorial representation of *tanh* and ReLU is shown in figure 2.11. Having gradient transformed more easily back to initial layers, it will be possible to develop more efficient deep networks, as the weights in all layers can be then significantly updated. As a consequence of ease of transferring information among layers, complex neural network with the ability of data abstraction can be yielded. Apart from the standards ReLU function, there are also other variants of it such as *Noisy ReLU* and *Leaky ReLU* in literature.



(a) Tanh (b) ReLU

FIGURE 2.11: Modern activation functions are emerged to address the problem of vanishing gradient. The error function of neural networks is measures based on the gradient of activation function. With sigmoid activation function, the measured gradient for the beginning layers (from back-propagation process) is very small. Therefore, the weights near the input layer do not get much updates from measuring the error function. By introducing newer activation functions, we are able to design more deep networks as the gradient can be transformed to the input layers much better than sigmoid function. The left panel represents the *tanh* activation function and the right panel show the ReLU activation function.

2.2.1.7 More tricks for large neural networks

In the process of neural network training, we feed the training data several times to the network. It is mentioned earlier that each one is called an epoch. For us to train a large neural network, it will be extremely exhausting to train it with all available

data. Instead, a technique called *stochastic gradient descend* is suggested in which for each epoch only a fraction of the whole dataset is fed to the network. Although with stochastic gradient descend, we end up with an estimation of the gradient that we are looking for, but in practice, it provides an effective way to deal with large amount of data using deep neural network.

Another trick is to control the speed of learning by having adaptive learning rates. That is, in the beginning of learning process, we have larger learning rate as compared to the end.

additionally, unsupervised learning could also be used to help the supervised neural network training. Unlabeled data which are easer to find than labeled data, can be used to *pre-train* the neural networks. This method which is known as *semi-supervised learning*, can improve the network weights much better than random weight initialization, and therefore, improving the classification results [33].

2.3 Modern types of deep learning algorithms

The fundamental concept of deep learning is discussed in the last sections of this chapter. This concept however, have been expanding so fast in the last few years such that, it branched and flourished much beyond its base concept. Currently, deep learning algorithms can be also classified into different families, each of which requires in-depth insight to comprehend their respective innovations and techniques.

In neuroscinece, artificial deep neural network helped modeling real biological neural networks in recent years [46, 104]. In the following, some of the well-known categorical branches of deep neural networks are introduced. Please be noted that only some of the core tricks and techniques are discussed in the following sections.

2.3.1 Convolution neural network (CNN)

CNN attained outstanding results in most of the state of the art problems of machine learning. From text classification to image classification [68], from natural language processing to human voice synthesiser [95], from artificial intelligence on gaming [2, 47] to computer vision [68], and even in language translation [41].

CNNs and MLPs are similar structurally. They are both composed of different layers, and use similar error functions. CNNs however, vary largely in the shape of input layer data and also in the types of hidden layers.

There are two subtle issues with MLPs. First, MLPs accept one dimensional data. Provided that the input data is multi-dimensional (like 2D or 3D images), MLPs require it to be transformed into a one-dimensional vector. Although this transformation makes it possible to feed multi-dimensional data to an MLP, it ruins the spacial interdependency of the elements of the input data. For instance, the pixels of an image are specially correlated, and by flattening it to a $1D$ vector, we would disentangle the local similarities, and in turn, leads to information loss. CNNs are especially designed to extract patterns from multi-dimensional data by preserving the internal coherence of data elements.

The second problem with MLPs is that it performs on fully connected layers. Fully connected layers can also be interpreted as having huge number of parameters to learn. With a normal pattern classification task, the complexity of the neural network can easily get out of control (see figure 2.12).

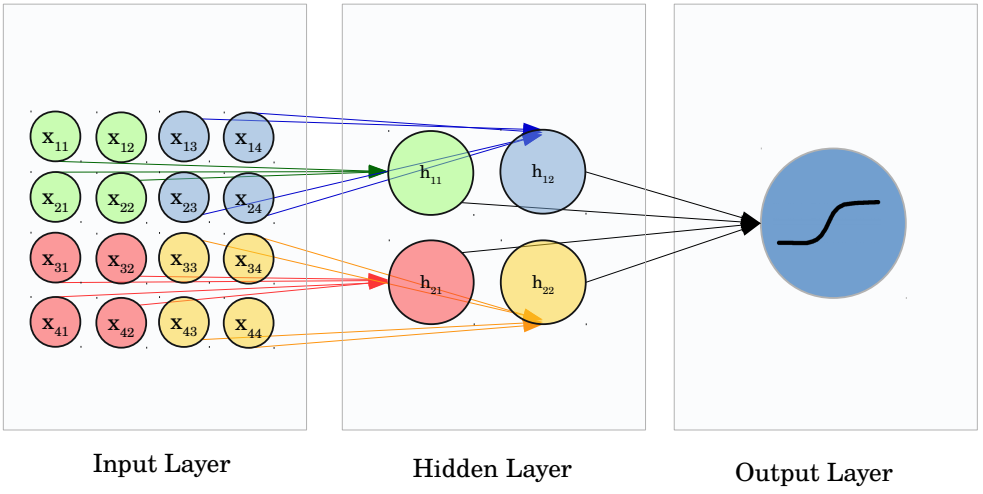


FIGURE 2.12: Convolutional neural network. The figure shows two main ideas behind the CNNs in contrast to MLPs. First, the structure of the input data is preserved. The adjacent elements of input data could be correlated. Therefore, the network is designed so that the nodes in hidden layers get input from certain regions of the input matrix. Second, the connections are sparse, and compared to MLPs, are not fully connected. This will mitigate the complexity problem of MLPs.

In the next sections, I explain the advancements introduced by CNNs as to address those mentioned issues.

2.3.1.1 Convolutional layers

The map from the data in the input layer to the hidden layer is not direct in CNNs. That is, as the name "convolutional" indicates, we convolve a filter to the input matrix to

obtain the elements for the next hidden layer. This resembles the concept of convolution in signal processing and also the concept of filtering in image processing. Thus, for a convolutional layer to be built, we slide a filtering window on the input data and measure the similarity of the overlapping part. By performing the convolution over the whole input matrix, we obtain a convolution matrix as a result, which represent the similarity of patches in the input matrix to the applied filter. Having tens or hundreds of filters, we can characterize the content of the input matrix. In case of the input being an image, this will grasp the image features and therefore, there is no need for a separate feature extraction phase in the image processing tasks while using CNNs.

The number of nodes in the hidden layers of CNNs is dependent on the number of filters used in it. The more filters applied, the more nodes in the second layer. Moreover, the size of a filter, the stride of a filter (the sliding step), and the padding for filtering all play a role in altering the size of the next hidden layer.

Convolutional layers normally take ReLU as their choice for an activation function.

2.3.1.2 Pooling layers

Having many filters will easily increase the dimensionality of the network, and having higher dimensional networks, in turn, increases the chance of overfitting. Hence in CNNs, a hidden layer called pooling layer is succeeding some or all of the convolutional layers in order to reduce the complexity of the network.

There are two established types of pooling layer. In the first type, *max pooling layer*, a window slides over the elements of convolutional layer output and takes the maximum value of the window to be used in the next layer. In this manner, the number of elements will be reduced significantly. The second method is *average pooling layer*, in which a window will slide over the output of the convolutional layer and takes the average value instead.

Pooling layers in fact convert multiple possibly infinitesimal nodes of the network to a single efficient one.

2.3.1.3 Stacking the hidden layers

Having a sequence of convolutional and pooling layers, one can make genuine deep neural networks. Having such structure, it is possible to represent the input data throughout the network, from general near the input layer to specific towards the output layer. That is, we can make hierarchy of patterns in which, a pattern like an image of the

human body can be represented by hierarchically combining its constitutive segments [74]. Amazingly, this resembles the way our brain sees and represents the visual objects [64]! A well-known example of pattern classification using CNN is shown in [68].

2.3.2 Recurrent neural networks (RNN)

MLPs are not suitable to capture the pattern in temporal data, the data in which one piece is temporally dependent on another. In simpler words, MLPs does not have memory. Recurrent neural networks (RNNs) are a class of artificial neural networks which are able to process and learn the time-based dependencies of the input data. As the word "recurrent" indicates, the network replicates the same task over and over again. Machine learning tasks such as machine translation, speech recognition, sentiment analysis, content analysis in video, street traffic analysis, stock market prediction, chatbots, gesture recognition in videos, and weather forecasting, can be extensively addressed by RNNs.

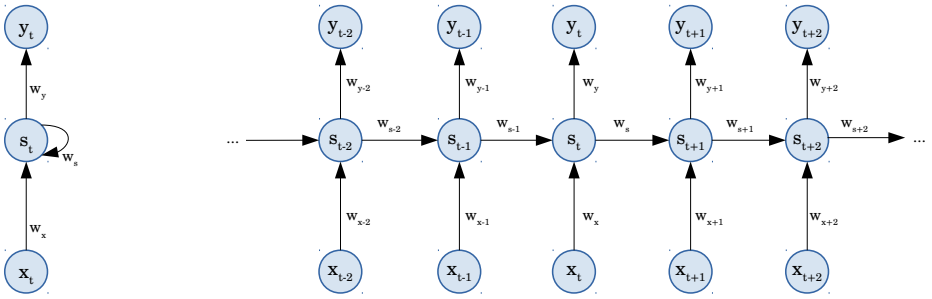
RNNs are designed to store the memory of the previous set of inputs. The memory segment in RNNs is called *state*. States are in fact the hidden layer outputs. The corresponding output for the current input in the RNNs is dependent on the input itself and also on the state variable(s) of the previous hidden layer.

Structurally, there are two intrinsic differences between MLPs and RNNs. First, for training RNNs, we do not feed a single element to the network but a *sequence* of elements. That is, the temporal dependency of the elements in the network must be captured and preserved. The second difference is the existence of the state variables, which are the output of the current input and used to be fed as input to the next layer. Therefore, the activation function will accept both the input variable and the state variable to measure the output:

$$\begin{aligned} y_t &= s_t w_s \\ s_t &= \sigma(x_t w_x + s_{t-1} w_s) \end{aligned} \tag{2.24}$$

A pioneer and simple instance of RNNs is known as *Elman network* [31]. Figure below illustrates how a simple RNN look like.

RNNs can accept one or multiple inputs variables and also can have one or many outputs. It is also possible to stack RNNs on top of each other by connecting the outputs of one to the inputs of the other.



(a) Folded (b) Unfolded

FIGURE 2.13: A simple representation of RNNs. Figure (a) shows a compact representation of RNNs known as folded. That is, for a time variable t, and input variable x_t, the output variable y_t is dependent on x_t and also the network outcome of previous times s_t. Figure (b) illustrates the stretched out version of figure (a) known as unfolded. The chart shows that how newer outputs of the network are dependent on the outcome of the previous steps.

To train RNN, we can apply the basic concept of feed-forward and back-propagation as described in the last sections with a slight change. Instead of normal back-propagation, we have an algorithm called *back-propagation through time*, which in that, we update the weights of the network based on the contribution of previous steps. That is, in measuring the partial derivative of the error with respect to a weight, we expand the calculation by adding the contribution of relevant previous steps (adding the partial derivative of the error with respect to previous steps).

RNNs perform adequately good if the number of time steps is not high (say 10 time point). Otherwise, due to problem of vanishing gradient, they are not able to act effectively. Here, is where a version of RNNs called *long short-term memory networks* (LSTM) [55], comes into play.

2.3.2.1 Long short-term memory (LSTM) networks

The primary motive behind developing LSTM was to acquire a unit to substitute the state in ordinary RNNs, which can decide what to memorize and what not to, on its own. Then, having recurrent networks of such would solve most of the problems such as vanishing gradient occurred by former RNN flavors. The new unit is itself a small neural network and is called *LSTM cell*.

As ordinary RNNs such as Elman network suffer from lack of a proper long-term memory, the LSTM cell is designed to posses a reliable long-term and short-term memory. In LSTM design, four functionalities is considered: *learning*, *forgetting*, *remembering*, and

using. Figure 2.14 shows a basic scheme of an LSTM cell [21]. As it can be observed, in contrast to the states in RNNs, LSTM cells has units known as gates to perform basic memory operations.

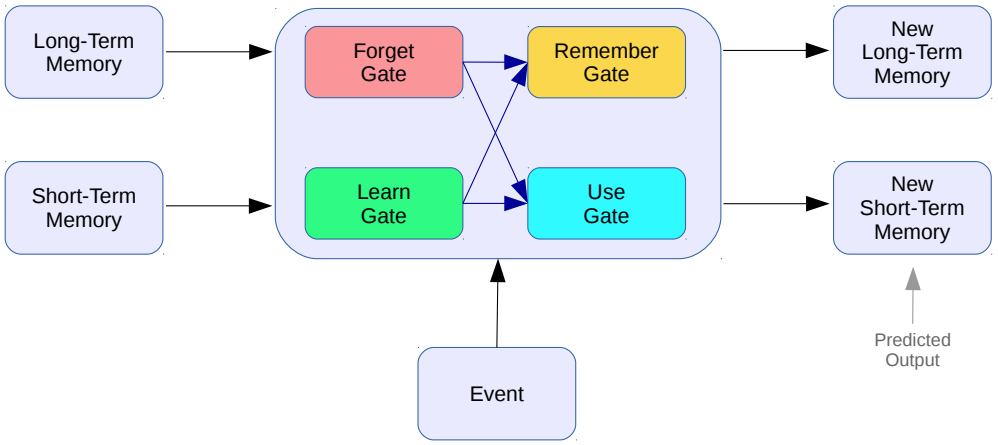


FIGURE 2.14: Basic LSTM cell concept. Four gates are considered to bring about the basic memory functions: learning, forgetting, remembering, and using. As an input event is fed to an LSTM cell, the short-term and long-term memory will be updated accordingly based on the internal gates of the unit. The output of the cell is the updated short-term memory.

The memory gates introduced above can be expressed mathematically as follows:

- Learn gate: The learn gate at time t, combines the short-term memory, STM, and the input event, E_t, and acquires new information, N_t. To update its state, it discards also a part of the new acquired information by multiplying it by a discarding factor d_t:

$$\begin{aligned} N_t &= tanh(w_n[STM_{t-1}, E_t] + b_n) \\ d_t &= \sigma(w_d[STM_{t-1}, E_t] + b_n) \\ Learn &= N_t \times d_t \end{aligned} \tag{2.25}$$

- Forget gate: The forget gate at time t, combines the short-term memory, STM, and the input event, E_t, through a small neural network and then multiplies the result, f_t, by the long-term memory, LTM:

$$\begin{aligned} f_t &= \sigma(w_f[STM_{t-1}, E_t] + b_f) \\ Forget &= LTM_{t-1} \times f_t \end{aligned} \tag{2.26}$$

- Remember gate: The remember gate updates the long-term memory by adding the collected information from the learn and forget gates:

$$\begin{aligned} LTM_t &= Learn + Forget \\ LTM_t &= (N_t \times d_t) + (LTM_{t-1} \times f_t) \end{aligned} \tag{2.27}$$

- Use gate: The use gate (output) is designed to update the short-term memory, STM, by combining two small neural networks. In the first neural network, it combines the forget gate and the long-term memory LTM. In the second neural network, it combines the short-term memory, STM, and the input event, E_t. Finally, it multiplies both networks output:

$$\begin{aligned} Net_1 &= tanh(w_{net_1} LTM_{t-1} \times f_t + b_{net_1}) \\ Net_2 &= \sigma(w_{net_2}[STM_{t-1}, E_t] + b_{net_2}) \\ STM_t &= Net_1 \times Net_2 \end{aligned} \tag{2.28}$$

At this point, we can put together different fragments of the LSTM cell to shape an LSTM cell. Figure 2.15 shows an LSTM cell wiring which contains the above mentioned gates.

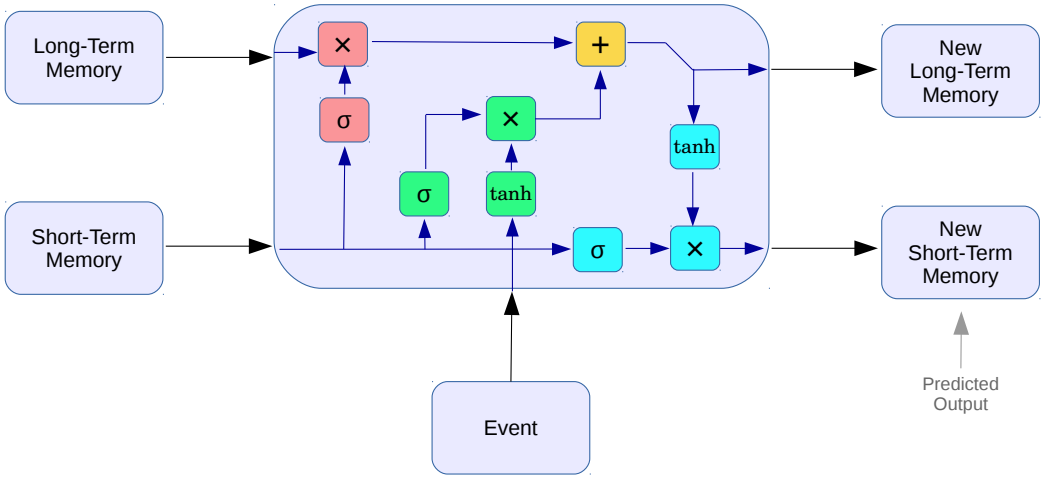


FIGURE 2.15: LSTM cell detailed architecture. In this LSTM cell example, the information gates (learn, forget, remember, and use) are combined to build an intelligent cell. This cell can replace the state variables designed in ordinary RNNs. The internal operations of LSTM, are color-coded with respect to figure 2.14.

The LSTM cells can replace the state variables in ordinary RNNs. Having such settings, each LSTM, individually, has an idea of the network thanks to its internal neural networks. By connecting several LSTM cells, we can make networks resembled to RNNs. However, unlike RNNs which were suffering from vanishing gradient problem and shallow architecture, we can build up rather deep networks with LSTMs (say 1000 layers). Figure 2.16 represents an LSTM network.

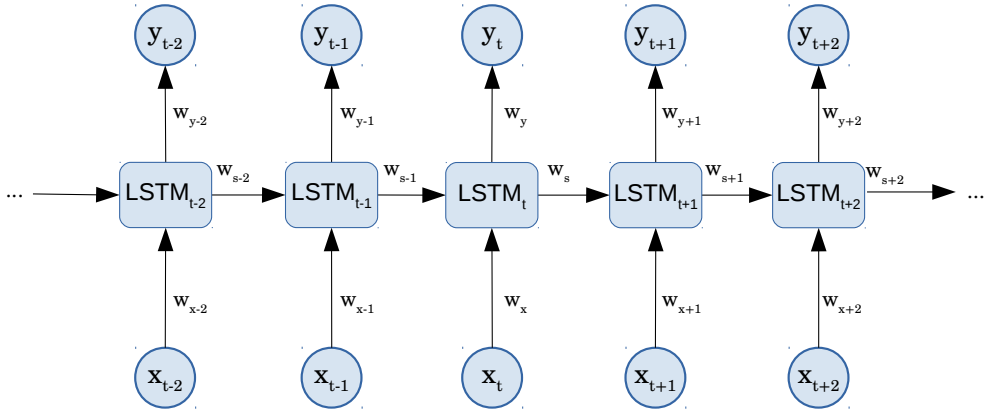


FIGURE 2.16: LSTM network. The LSTM network can overcome the limitations of RRNs. By means of every activation functions inside each cell, a cell could decide on letting the input to come to the cell, on retaining the information, and on letting the information being departed from the cell. Therefore, the network is more capable of storing memory and consequently, can be much more deep.

The presented LSTM cell is not the only LSTM cell in the market. The reason for its popularity is due to its practicality. However, one can think of different arrangements for LSTM cells. As far as LSTM cells can answer the need to deal with long-term and short-term memories in neural network effectively, other versions can be employed as well.

2.3.3 Other types of deep neural networks

In accordance with the type of data given in this thesis, which are mostly time series, the above presented deep neural network methods could be used to model the data. Nonetheless, deep learning is not limited to the mentioned categories. As long as classic machine learning and artificial intelligence (AI) methods are concerned, deep learning is widely employed and explored in those fields. Fields such as *deep generative adversarial networks* and *deep reinforcement networks* [83] drew much attention in the recent years.

Since these fields are not directly relevant to our work and our data, I am not going to present them in this work. However, in some cases, they can be used to address certain types of neural data, especially if a paradigm is designed to be modeled with AI methods.

To develop deep learning algorithms, one should not start from scratch. There are a set of good toolboxes available at the time in the market. They are basically designed to

handle much of the background difficulties such as sharing the computation processes with GPUs and resource managements. They offer also easy to learn user-interaction interfaces. TensorFlow [1], Theano [98], Keras [17], Caffe [62], ... are some examples of deep learning packages used in todays machine learning tasks.

2.4 Summary

In this chapter, I presented the core idea of artificial neural network, its basic building blocks and its development towards a complex machine learning approach. I went through its classical limitations and followed it to modern artificial neural networks, known publicly as deep neural networks. Modern branches of deep learning algorithms were also discussed, to the extent applicable to the natural neural data presented in this thesis.

I had three reasons to bring up deep learning in this thesis.

- First, it is a great chance to check the feasibility of applying deep learning on three classes of neural data (presented in this thesis), and conclude whether or not, it is useful to apply deep learning algorithms on "limited samples bulky time series".
- Second, classical machine learning approaches is addressed more broadly than deep learning on neural data. I found a room to evaluate deep leaning algorithms on the data I had at hand, to possibly extend the realm of deep learning in new environments.
- Third, since we deal with machine learning in general, the core concept of artificial neural networks, to a great extent, overlaps the main stream machine learning algorithms, and therefore, through describing basic of artificial neural networks, basics of machine learning is presented as well.

This chapter can be used also as a tutorial for people who are interested to start learning and working with artificial neural networks, and would like to know its strengths and limitations.

Chapter 3

Working memory, machine learning, and intracranial EEG

In cognitive neuroscience, discovering the mechanisms of *working memory* is one of the primary areas of research. Working memory in contrast to *short-term* and *long-term* memories, refers to briefly maintaining, processing, and manipulating of information in the brain, whereas short-term and long-term memories refer generally to preserving information for shorter or longer time intervals [22, 29].

With respect to what we discussed in chapter 1, one approach to decode brain activities is by classifying and modeling them with **machine learning** techniques [38]. In this chapter, the main focus is to present two working memory studies which are iEEG based recordings, and are addressed by machine learning techniques. Additionally, in the appendix A, an fMRI based working memory task is briefly presented, where again a machine learning solution is introduced to solve a cognitive neuroscience problem.

Given that we are aiming at studies with human subjects, and the fact that we are keen to understand how their brains function in macro levels, we can not control the study conditions without their consent and cooperation. On that account, for them to measure their brain activity, they should be put into certain condition and be asked to follow some procedures. This experimental design is called *paradigm*.

Paradigms are typically a computer program which interacts with people visually and/or audibly. From performing a paradigm, we get two sets of results. First, the behavioral data like the participant reaction time or the performance of participants in answering the questions. Second, we get the synchronous brain activity recording. Paradigms must be designed wisely to pose a scientific question and get the answers in a controlled

manner to be able to reveal the underlying brain activity and avoid confounding non-controllable variables.

Brain recordings can be sourced from a various recording methods such as EEG, fMRI, MEG, . . . , each of which has its own pros and cons. The EEG types of recording benefits from its high frequency resolution of recording as it measures the electrical pulses while it suffers from its poor spatial resolution. fMRI on the other hand, has a proper spatial resolution but has an ineffective time resolution.

In the proceeding paradigms, our data is a very special and hard to record type of EEG called iteracranial EEG (iEEG) which has superior spacial resolution compared to normal surface EEG recordings.

In epilepsy clinics across the world, epilepsy patients with unrecognized origin of their epilepsy in the brain, in expectational cases, through a sophisticated surgery, are being implanted with under surface electrode to discover the affected section of their brain. These electrodes can be subdural (on the outer surface of brain under the dura) or depth electrodes (penetrated in the brain tissue). The implanted electrodes are called intracranial since they are positioned inside the skull. The primary advantage of having intracranial electrode apart from better spatial resolution is to record the brain activities which are otherwise not accessible or easily recordable outside of the skull.

The recording signal of interacranial EEG (iEEG) is a set of time series, each of which is a recording of changes in electrical charges around the position where the iEEG electrode is implanted (see figure 3.1).

In the following, I introduce two iEEG studies by going through the corresponding paradigms (*Sternbeg* paradigm and *Face Direction* paradigm). Since the technical solutions of both paradigms share to a great extent with one another, I skipped the technicalities of the second and referred to the first.

3.1 Sternberg paradigm

The Sternberg paradigm is a classical working memory task and is named after the famous American psychologist Paul Sternberg. The design is meant to study how many individual items can a person sequentially keep in mind form a series of items. The Sternberg paradigm task was to ask participants to maintain a sequence of digits, one to nine as they presented sequentially on the screen and after a probe, rehearse the sequence of already seen digits. The length of the digit sequence could be either of one, three, five, or seven. This paradigm was designed and programed in the epilepsy clinic

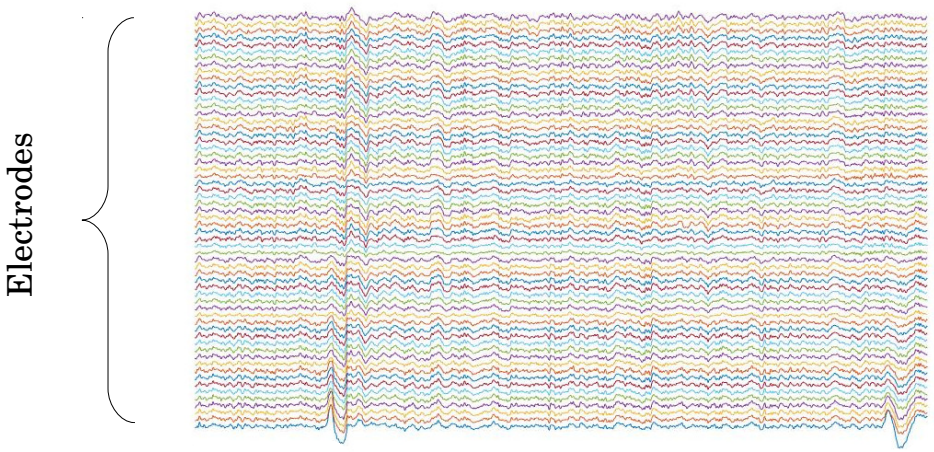


FIGURE 3.1: An example iEEG recording across multiple channels is show here. Different electrodes may record different regions on the brain surface or in depth tissue. The signal of different channels may look similar in the first glance and it is due to the very low frequency components of the recording. Splitting the signal into its constructive frequencies can reveal the differences among all electrodes for an ongoing brain activity.

as a part of cognitive study before I started my work at the clinic and was being recorded thereafter.

The idea behind this paradigm was to certify the well-established Sternberg paradigm with in-depth brain signal and realize the encoding and memorizing mechanism of brain while processing a sequence of items [76]. Figure 3.2 illustrates the steps of Sternberg paradigm visually.

The subject's tasks could be divided into three main phases. In the first phase, the *encoding* phase, the participants were presented with a sequence of digits, one, three, five, or seven. Each digit appears for 500 ms and after it, a fixation cross appears on the screen and the individual has to look at the sign for 1500 to 2000 ms. The duration is jittered to reject any time-based rhythmic memory consolidation theory.

In the second phase of the paradigm, the *maintenance* phase, the participant has 3000 ms time to maintain the already seen digits in mind and in the third step, the *retrieval* phase, they type the sequence of digits which they have already seen one by one on the presentation laptop.

While performing the task, the brain activity of the participants were recorded through the clinic iEEG recording machine. The recordings were synced to one millisecond resolution.

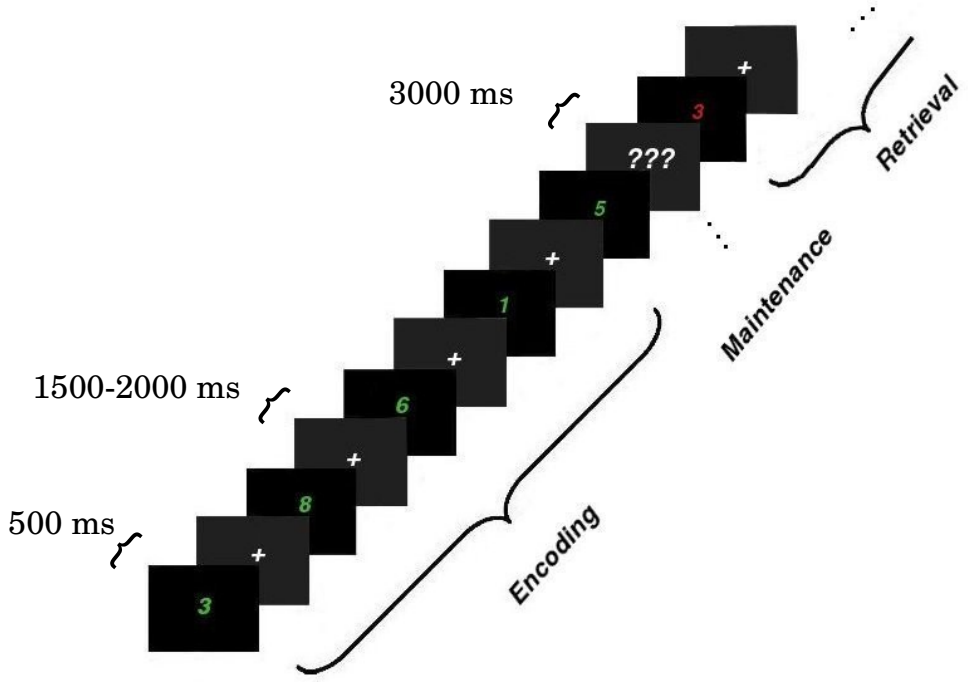


FIGURE 3.2: Sternberg paradigm. The details and task duration is presented. In this paradigm, the participants were asked to memorize a sequence of digits (one, three, five, or seven). This phase is called encoding phase. Right after it, the participant is given a time to maintain the already seen digits (maintenance). Then, a probe appears on the screen and the participant must remember the digit sequence by typing them in the paradigm software. This phase is called retrieval phase. During the task accomplishment, the brain activities were being recorded through clinical iEEG monitoring equipments across different channels.

Having this valuable dataset available, I posed a new question on it: **"Can we decode the way digits are represented in our brain?"** Since this paradigm was not designed initially for this task, I had to change the way the data was segmented (see segmentation later in this chapter).

For us to track the pattern of digits representation in the brain, instead of the sequence of digits, it was necessary to consider every digit at the presentation time or at the retrieval time, as separate instances of data. This is not however a trivial task since according to the paradigm design, the participant had to not only maintain the already seen digit in mind but at the same time rehearsing the sequence of previously seen digits. This fact would disturb a clear digits representation in the brain.

3.1.1 Methods

As mentioned before, to model the brain activity, it is planned to accomplish it by the means of machine learning and pattern classification techniques. In the following, the step-by-step technical solution for machine learning modeling is presented.

Similar to typical machine learning tasks, we should follow the following steps as suggested in [30]:

- Preprocessing
- Feature extraction
- Learning and classification

Before proceeding to the technicalities, let's talk first about the subjects and the data.

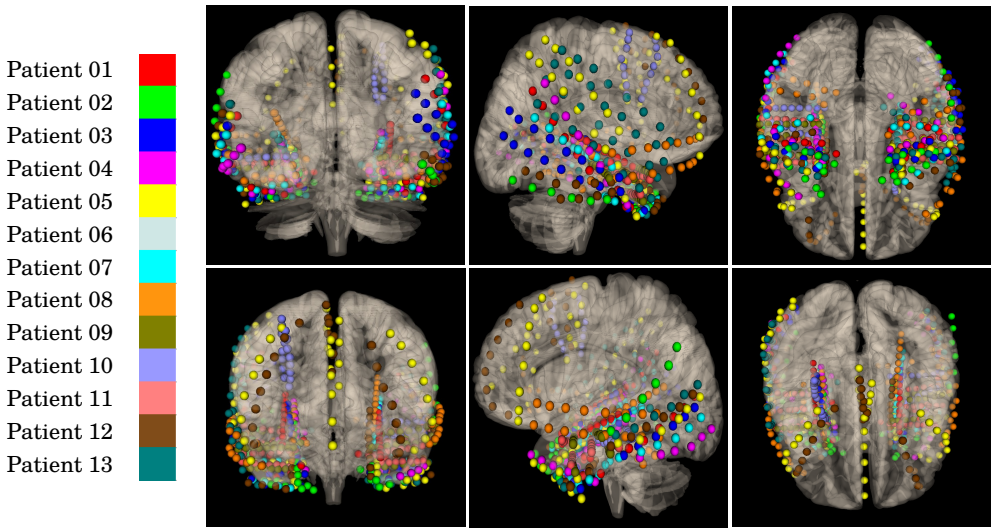
3.1.1.1 Subject and data

The subjects of this study were patients admitted in Epileptology Clinic Bonn for seizure monitoring. The subjects were implanted with electrodes to discover their brain lesions which in turn, cause the epileptic seizures. This opens a window of opportunity for groups like ours to use the chance and record from depth brain tissue which otherwise is not available for recording.

Recordings were acquired using stainless steel subdural strip or grid electrodes (contact diameter: 4 mm, center-to-center spacing: 10 mm) and intracranial depth electrodes (diameter of 1.3 mm, comprising 10 cylindrical platinum contact sites with a length of 2.5 mm and an inter-contact spacing of 4 mm). All data was sampled at 1000 Hz, referenced to linked mastoids and bandpass filtered [0.01 Hz (6 dB/octave) to 300 Hz (12 dB/octave)] using the digital EPAS system (Schwarzer, Munich, Germany) and Harmonie EEG software (Stellate, Montreal, Canada). To identify electrode positions, all

contacts were transferred into normalized MNI [1]space using the FSL (FMRIB Software Library).

Different patients have different number of electrodes and different distribution of electrodes. Figure 3.3 shows how different patients have different implantation pattern. The position of electrodes is plotted based on MNI coordinates.



1

FIGURE 3.3: Electrode distribution of Sternberg paradigm of all patients from different views. The position of electrodes in an MNI coordinate is plotted in color on a normalized brain image. Each color represents a patient. These images show that electrodes were merely implanted based on their diagnosis. It can be however seen that there are regions (e.g. medial temporal lobe) in which, most of patients have some implanted electrodes. The position of implanted electrodes for all patients are calculated based on their pre and post surgery MRI images with the help of *Pylocator* program [67], and then pooled together and plotted by the same program.

The data from intracranial and surface electrodes is recorded using *Stellate* ® EEG measurement system and is saved under Stellate file format (*.sig*, *.sts*). For our convenience, we have converted the data to *Brain-Vision* ® format using Brain-Vision Analyzer software. By doing this, we have obtained three files *.dat*, *.vmrk* and *.vhdr* file formats to represent raw signal, stimuli time event (trials) and electrode arrangement respectively. The data is then imported into Matlab ® and further analyses is primarily done in Matlab. In some section, Python ® code is also used to corroborate the analyses.

[1]MNI stands for Montreal Neurological Institute and Hospital, is a standards $3D$ coordinate system to address the coordinates of brain volume in millimeter precision.

3.1.1.2 Preprocessing

The preprocessing phase is required to prepare a noise-free and artifact-free data for the next phases of machine learning. Preprocessing is composed of the following steps:

- Artifact Rejection
- Filtering
- Signal Segmentation
- Baseline Correction

Artifact Rejection There are different types of artifacts in EEG data. Muscle artifacts, eye blinks, spike waves, and pathological high frequency artifacts. Here, I address the removing of spiky waves since they are more prominent source of artifacts in Intracranial EEG.

Intracranial EEG, is contaminated with not only the recording noise but also with artifacts resulting from pathological neural activities. These inherent neural activities are known as *spikes*. Spikes are rapid and high power activities of neurons. Figures 3.4 shows an spiky brain activity.

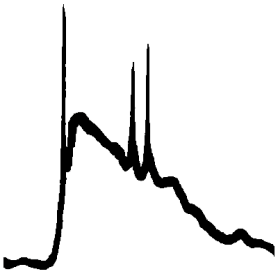


FIGURE 3.4: Three prominent spikes could be seen in the signal. Spikes are typically abrupt high magnitude rise of in the brain signal. They are normally broadband and can not be easily filtered.

Spikes are abrupt rise of the brain signal which are the result of overcharged cell ensemble activities. The activity of individual neural cells are typically spiky shape but since the cells are not sync necessarily, the recorded data on (i)EEG electrodes should not typically look like spikes. Spikes are in most of the cases the results of epileptic-like activities. Since spike are not the result of ordinary brain activity, they are considered to be artifacts in the brain studies and therefore, must be removed.

The process of removing spikes is a tedious task and has been done manually as a data preprocessing step in several epilepsy clinics. People show the spike activities of different

electrodes and decide whether or not to discard a part of data. A problem in the manual spike removal tasks is that different people have different views of spike definition. Some people remove the part of data which contains a spike if it happens in only one electrode and some people remove it if it happens synchronously across multiple electrodes.

Here I have developed an algorithm for automatic spike detection. The algorithm is developed by watching over the shoulder of three persons who were manually removing artifacts from three iEEG databases and recording their criteria. Based on the gained knowledge, new criteria for automatic spike detection was defined and the algorithm was developed.

In this algorithm, the average slope of signal is measured for an epoch of an electrode recording (e.g $500ms$). The epoch will be marked as spiky if the average slope is more than 75° (= persistent sharp signal rise) and/or the difference between maximum and minimum value is more than 6 times of the standard deviation of the signal in the 30 seconds window in signal's vicinity.

Ultimately, we consider an epoch to be spiky and noisy if at least 20% of the electrodes at that particular time are marked as artifact. The noisy marked part of the data will be remove from further study. In the code snippet 4 the algorithm for artifact detection is presented.

As mentioned, artifacts are usually occupying in a broad bands of frequencies. Hence, filtering those contaminated frequency bands would diminish a valuable instances of data. Alternatively, some people suggested using *independent component analysis* (ICA) to remove the spiky constitute of the signal instead of discarding it. It is however recommended by experts in the field to exclude those spiky artifacts from the whole study to make sure that no artificial contamination is made.

Filtering Brain as an extremely complex system with highly sophisticated interconnected subsystem, has an electrical medium for communication. Similar to the concept of radio waves and their frequency modulation, brain regions communicate to each other through different frequency channels. These are known as functional frequency bands of the brain. Table 3.1 shows a conventional classification of brain functional frequency bands.

TABLE 3.1: Functional frequency bands.

Frequency Name	Delta	Theta	Alpha	Beta	Gamma
Frequency Range	1-4 Hz	4-8 Hz	8-12 Hz	12-30 Hz	30-100 Hz

Algorithm 4: Detecting iEEG artifacts

Data: iEEG Raw Signal
Result: Detecting iEEG Artifacts

```
while not all channels selected do
    select an iEEG channel;
    resample to 1000 Hz sampling frequency;
    while not at the end of this signal do
        epoch ← take 30 seconds of iEEG signal;
        epocStd ← standard-deviation of epoch;
        while not at the end of epoch do
            window ← 500 ms of epoch;
            windowMax ← max(window);
            windowMin ← min(window);
            a ← measure window[1:end-1] - window[2:end];
            b ← arctan(a);
            c ← degree(b);
            d ← abs(c);
            windowDegree ← mean(d);
            if windowDegree > 75 or (windowMax-windowMin > epochStd × 6)
             then
                Mark the current time and channels as being artifact contaminated;
            end
        end
        move epoch/4 seconds forward;
    end
end
Search across channels along the time-line:
    if a time-point is marked as an artifact in 20% of channels then
    Mark the current time as noisy;
end
```

For instance, delta waves are known to be spread during sleep, alpha bands when you close your eyes, and gamma when performing cognitive and memory related activities. These bands are not however limited to the mentioned activities. Depending on the task, some particular frequency bands could be the medium of interest.

There are different ways that frequency filtering applied in the preprocessing step of (i)EEG recordings. Since EEG is an electrical recording in a microvolt or millivolt levels, any external source of electrical power in the recording environment, if not shielded, can affect the signal by inducing a magnetic field. This artifacts must be removed from the signal if they affect any frequency of interest in the study.

Since our iEEG recording is performed in Germany and the power line has a frequency of 50 Hz, it affects the Gamma power band. To get rid of this noise, we band-reject the signal for the [49.8 50.2] Hz interval by applying *notch-filter*.

Frequencies under 1 Hz and over 150 Hz also could be eliminated since they do not reflect any meaningful functional mechanism of the brain.

Technically, filtering can be done by different filtering tools such as FFT, wavelets, multitaper, Butterworth, Chebyshev, Notch, ..., each of which has its own use-case and also its own pro, and cons, and there exists no perfect filter. Filtering should be chosen based on the task itself, as some filters leave different edging effects and passband ripples than the others [96].

There are some tricks to improve the filtering quality such as improving the edging effect. For instance, if the signal is meant to be segmented and then filtered, people make the filtering first and then perform the segmentation last, to have more seamless filtered signal on the segmented edges. Another trick to avoid edging effect for already segmented signals is to reverse the signal at both ends of it and concatenate the reversed version to the original signal, perform the filtering on a rather continuous signal and the discard the concatenated parts.

It must be mentioned also that filtering from another perspective can be a part of feature extraction phase too. Thus, no wonder if I again come back to the filtering in the next section.

Signal Segmentation In a paradigm design, it is an important consideration to be able to track the onset and the offset of a specific brain activity. It can be determined from the paradigm design, when a mental task is started. Knowing the onset and the offset of the mental task, it is possible to segment a continuous data into task related sections. Each of these segments is called a *trial*. Figure 3.5 shows the frequency split phase as well as segmentation.

Baseline Correction Baseline correction is a technique to exclude ongoing brain activities before from a trial but with preserving the task related activities. Ongoing brain activities referred to the normal non-task related activities of the brain.

Since brain activities happen in different frequencies, to exclude the ongoing brain activities, for every frequency, the average signal of 700 ms to 200 ms before the beginning of the trial ($[-700\ -200]$), is subtracted from the amplitude of the signal within the trial. With respect to the segmented trials baseline correction can be applied (e.g. encoding phase or retrieval, etc).

Doing this, the amplitude of the signal within the trial would reflect the brain activity which is mainly task-related. This technique is called baseline correction since we remove the baseline brain activity from the signal.

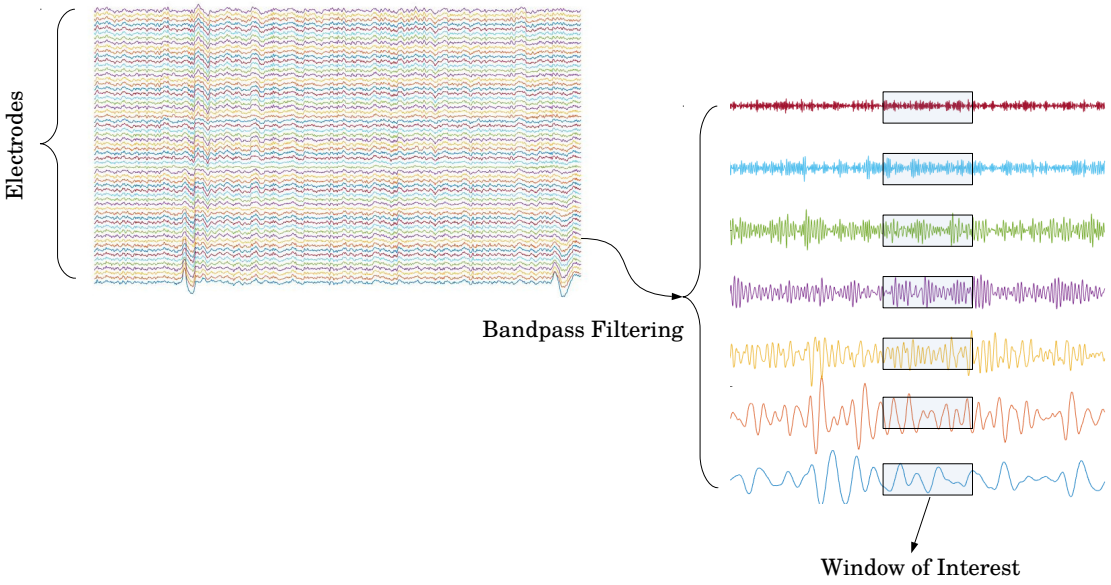


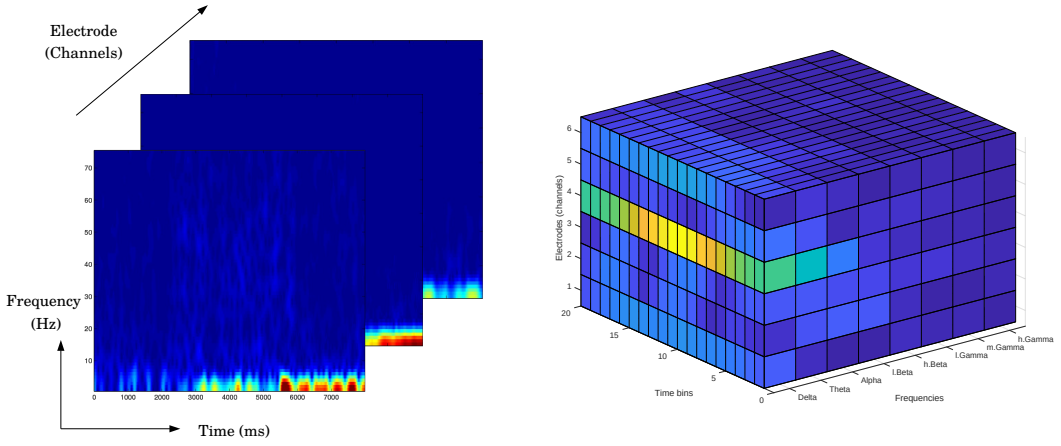
FIGURE 3.5: Frequency splitting and windowing. Splitting every channel to various frequency bands, to be able to measure features for a window of interest. Having obtained the onset and the offset of a brain activity, we are able to segment the data as windows, across different electrodes, and across different frequencies.

3.1.1.3 Feature extraction

Our raw data is changes in electrical charges resulted from the brain activity which is referenced to a reference electrode(typically the *mastoid* electrode near the ears of the patient). Since for each channel (electrode), there is a time variable, and we have multiple channels across the head, and also each channel can be split into several frequency bands, we will end up having a $3D$ matrix of raw data which each element of it is a value in a time-frequency-electrode map. Figure 3.6 depicts such a map. This way of representing brain data is called *time-frequency analysis* in the field.

In this concept, it is a common practice to bin the time axis data into 20 to 50 ms bins [38, 56]. The length of bins must be chosen so that the number of time points from raw features reduced significantly (by 20 to 50 times with 1000 Hz sampling frequency) and still being able to accommodate complete cycles of the high frequency gamma activity without loosing the time resolution. We call each of time-frequency-electrode bins a ***cell*** see figure 3.6(b)).

We talked so far about using raw signal directly for filtering and feature calculation, and it is justifiable to use it in some applications. However, in most of the cases, in order to obtain more prominent neural activities, we must perform some sort of *transformations* on the raw signal. Performing transformation alters the way a data can be seen and represented. The result is called *analytic signal*. Analytic signals are in the form complex



(a) Time-frequency-electrode plots

(b) Time-frequency-electrode cells

FIGURE 3.6: Time-Frequency-Electrode Map. Having this mapping (a) helps to disentangle constructive components of the raw signal and convert it into discrete buckets of information (b). Each bucket (cell) represents an electrode, a frequency, and a time period, in which features can be extracted from.

numbers and from an analytic signal, two secondary feature values can be extracted: the *power* and the *phase*. Power and phase are explained later in this chapter.

In the following, we describe three transformation methods which are extensively used in the realm of (i)EEG analysis to obtain analytic signal. They are aimed to convert the signal from time-domain to frequency domain.

- **Fast-Fourier-Transformation**

  Fourier transformation is proposed by french mathematician Joseph Fourier (1768-1830), suggesting that a continues signal can be represented by a combinations of sinusoids (sins and cosines). Sinusoid functions can be depicted by their amplitude, frequency and phase. That is, for every frequency, it is possible to show the phase and the amplitude of a particular signal and call it frequency representation of the signal. Fourier transformation can be formulated as:

$$f(\nu) = \int_{-\infty}^{+\infty} f(t) e^{-2\pi i \nu t} dt \tag{3.1}$$

  where t is the time, $f(t)$ is the amplitude of the signal in time t, ν represents a frequency, and e^{it} is the Euler's formula for representation of complex values. Euler's formula can be defined as:

$$e^{it} = cos(t) + isin(t) \tag{3.2}$$

where i indicates the imaginary part of a complex number. In this notion, a function of time is converted to a function of frequency. That is, we can decompose a signal from time domain to a collection of sinusoids in frequency domain. Each sinusoid is the representation of the original signal for a certain frequency in frequency domain. It should be noted that Fourier transformation assumes that the time signal is a periodic signal.

The *inverse Fourier transform* is also introduced to convert the signal from frequency domain to time domain:

$$f(t) = \int_{-\infty}^{+\infty} f(\nu) e^{2\pi i \nu t} d\nu \tag{3.3}$$

where the $f(\nu)$ contains the amplitude and the phase information for all frequencies. The inverse Fourier transform is in fact a collection of multiplications and summations to put together all frequency representations of a signal in frequency domain and convert them to a single signal in time domain.

The above notion of Fourier transform is dedicated to the transformation of continuous signals. In digital signal processing, normally, we deal with discrete and quantized signals. Therefore, our input signal will be a discrete sequence of data points ($n = 1...N$) and instead of all possible frequencies, limited number of frequency bands will be resulted. Therefore, the integral in the equation 3.1 will be replaced by a summation:

$$f(\nu) = \sum_{n=0}^{N-1} x_n \, e^{\frac{-2\pi i k n}{N}} \tag{3.4}$$

where X is the discrete signal in time domain and k represents the frequency (the number of cycles per N samples). This equation is called *discrete Fourier transform* (DFT).

Fast Fourier Transformation (FFT) is a well-known and efficient algorithm to measure DFT. FFT decomposes the DFT matrix into sparse factors and therefore reduces the calculation complexity. To fit the FFT into our cell concept, we require a flavor of FFT called short-time FFT (STFFT) in which the FFT is measured for a moving window on the signal and the window moves forward by some steps less than its length. With the help STFFT, it will be possible to characterize the neural activities both in terms of time and frequency. A good advantage of FFT is its simplicity of use. However, FFT is not the best transformation to pick up acute and abrupt signal changes [96].

- **Hilbert**

Hilbert transform is named after German mathematician David Hilbert (1862-1943) and provides us with an analytic signal. Hilbert transformation, technically, takes the frequency elements of Fourier transformation as input and rotates the complex value so that the real part maps onto the imaginary part ($\frac{\pi}{4}$ phase shift). Then, it performs the inverse Fourier transformation. Mathematically speaking, the Hilbert transform measures the convolution of $\frac{1}{\pi}$ with the input signal $u(t)$:

$$h(u)(t) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{u(t)}{t-\tau} d\tau \tag{3.5}$$

where t represents the time. The real part of Hilbert transformation is identical to the input data. The power and phase of Hilbert signal can be measured as explained later in this section.

A great advantage of Hilbert transformation can be revealed if it is applied on a bandpass filtered signal, as it can then signify the instantaneous power and phase in the signal much more prominently than STFFT.

Thus, we consider filtering the raw data into different frequency bands and then apply Hilbert transformation on the filtered signal. For the frequency band split definition, I stick to the functional classification of brain frequency signals as described above in the preprocessing step, but with slightly more frequency resolution for the beta and gamma frequency bands. Therefore, we end up having the signal bandpass filtered to the following 8 frequency bands:

TABLE 3.2: Frequency bands used for Hilbert transformation. Eight frequency bands were used to pass-band the input signal of each electrode, and to obtain analytic signal by means of Hilbert transformation.

Frequency Name	Delta	Theta	Alpha	Lower Beta	Higher Beta	Lower Gamma	Mid Gamma	Higher Gamma
Frequency Range (Hz)	1-4	4-8	8-12	12-20	20-30	30-50	50-75	75-100

For the choice of bandpass filtering, we opt second level *Butterworth filter* [2] as it has a monotonically falling of frequency response on the edging frequencies [96]. Figure 3.7 shows how Hilbert transformation on the filtered signal can represent the envelop of the signal and therefore, providing new information (analytical power). Please consult with "power and phase" section first if necessary.

[2] Butterworth filter is a classical type of analogue filters used widely in signal processing. It is named after British Engender Stephen Butterworth and is designed to have a very narrow frequency response to be used for the convolution procedure in filtering. A narrower frequency response causes more distinct frequency split with minimal distortion with neighboring frequencies.

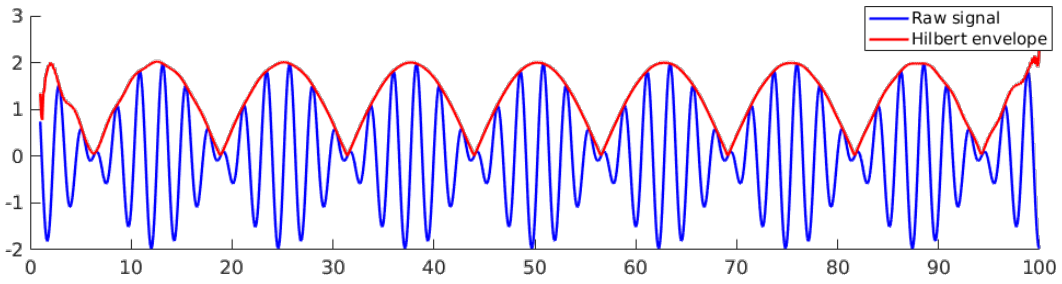


FIGURE 3.7: Hilbert transformation power. The red curve illustrates the amplitude (power) of Hilbert transformation. It makes an envelope around the signals which are in the same frequency vicinity and represents instantaneous power.

The phase of Hilbert transform is demonstrated in the figure 3.8 (analytical phase). The periodicity of the phase can be easily observed as it fluctuates between 0 and π. No clear pattern of periodicity can be observed in the phase value if the signal is not filtered to certain frequency bands.

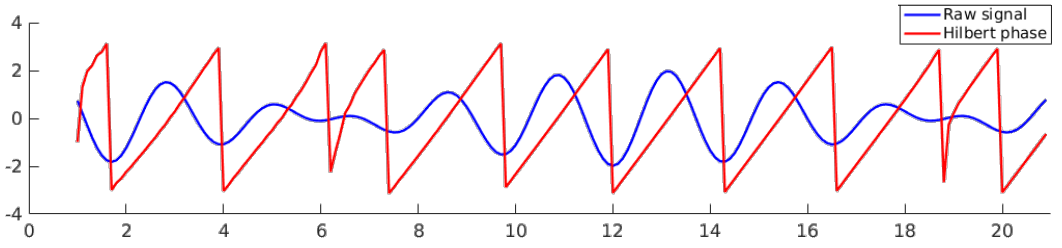


FIGURE 3.8: Hilbert transformation phase. The phase changes between 0 and π. The periodicity can only be observed if the signal is filtered.

- **Wavelets**

  Wavelets provide a way to analyze the frequency components of a signal. The plural word of wavelet**s** is to indicate that multiple waveform signals are used for the transformation. The wavelet transformation is basically performed by convolving the scaled and shifted version of another signal called *mother wavelet* with the original signal to obtain. The results of wavelet transform is a collection of similarity coefficient across time and frequency. Unlike Fourier transform, wavelet transform is designed to pick up the abrupt and non-regular changes in the signal.

  There are different types of wavelets. Mother wavelet is the most important characteristic of a wavelet transformation. The mother wavelet is a curve with zeros mean and finite length which has zero values in the beginning and at the end of the curve. Based on the shape and the design of mother-wavelet, different types of wavelets can be defined such as: *Morlet*, *Mexican Hat*, *Symlets*, *Biorthogonal*, *Haar*, ...

Similar to Fourier transform, *Continuous wavelet transform* and *discrete wavelet transform* are also two types of wavelet transformation, each of which can be used similarly for different data types, data resolution, and applications.

As mentioned, technically, scaling and shifting are two fundamental operations in wavelet transform.

- **Scaling**: is the act of stretching a wavelet in time domain with some scaling factor. Based on the scaling factor, the mother-wavelet can then represent various frequencies and can be used to reveal the frequency similarities compared to an input signal.
- **Shifting**: is to move the (scaled) wavelet forward on the on the signal to be able to measure the similarities all over the signal.

If we perform m times scaling and n times shifting in a wavelets analysis, we will end up obtaining $m \times n$ similarity coefficients, each of which represents the strength of similarity of the input signal to the scales-shifted wavelet (see Figure 3.9).

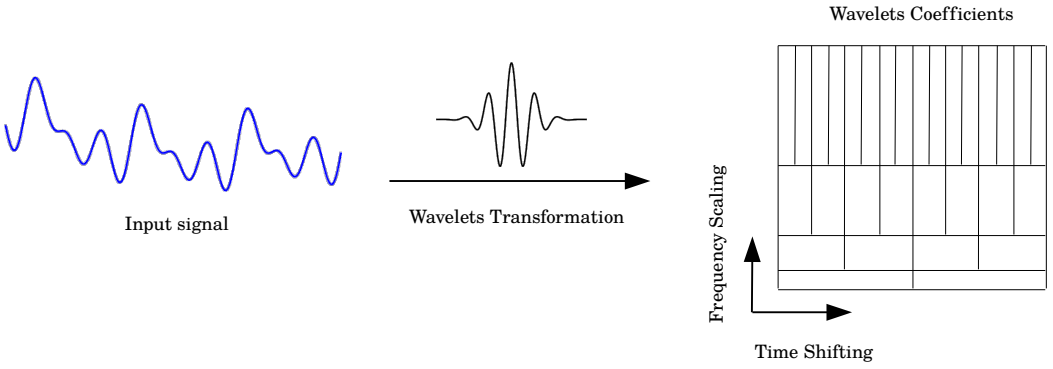


FIGURE 3.9: By performing the wavelets transformation on input signal, a set of wavelet coefficients are resulted (right panel). It is important to note that for lower frequencies, we have less wavelet coefficients but narrower frequency bands, and for higher frequencies, we have more wavelet coefficients but with wider frequency bands.

Since we deal with a complex value system, the analytical power and the phase for every time-frequency cell can be calculated the same way as with Hilbert transform (see below section for more details).

In neuroscience, people tend to use Morlet wavelets more than others as it best features the underlying neural activities. Morlet wavelets is stemmed from multiplying a sign wave and a Gaussian wave. Therefore, by applying it, there will be less concern about the edging effect in frequency domain.

Hilbert transform and wavelets transform are shown to produce accurate analytical power and phase, and both are extensively used in time-frequency analyses in neuroscience.

As explained before we can obtain phase and power values from signal transformation to frequency domain. in the following, the phase and the power are explained in detail.

Power and Phase Physical phenomena are measured in two fashions, as scaler units or else as wave forms. The weight of an apple renders a scaler unit whereas the color of the apple can be represented as oscillatory wave forms of light. In the waveform representation, a physical phenomenon is supposed to have variable values in different times while these values raise up and down with a certain speed and magnitude. Given that the speed and the harmony is constant in a physical signal, the signal is called a *periodic* signal since the content of the signal will be repeated after certain time interval.

Such periodic property of a signal is called Frequency. One of the simplest form of a periodic signal is the well-known function $sin(x)$ (see Figure 3.10). It is observable that after every 2π, the signal replicates its past. Hence, it is called periodic.

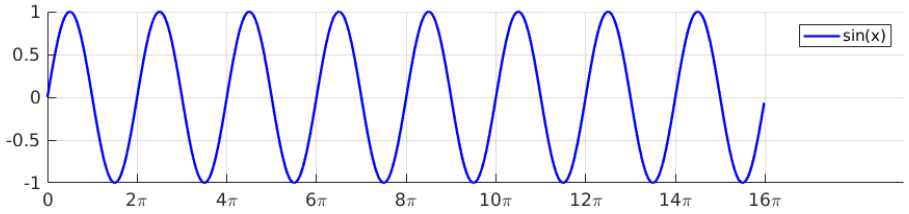


FIGURE 3.10: Sine signal. Sine signal is one of the simplest periodic signal. It repeats its past after a particular period, on and on.

Now consider the cosine function (see Figure 3.11). The $cos(x)$ looks the same as $sin(x)$ except that there is an offset between them. If x axis in the figure is the angle, then this shifting is the angular difference between two signals. In other words, the signals have different *phase*.

Apart from the phase attributes of a periodic signal, there is also another property, called *power*. The power of the signal represents the magnitude of the signal amplitude at a certain time/angle (y axis). Signal power can be the absolute value of the signal amplitude or square of it.

In the following, let's first review the phase property of the signal more closely.

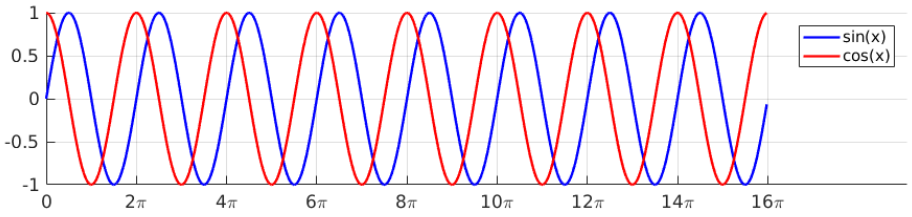


FIGURE 3.11: Sine wave vs. cosine wave. The cosine wave is the same replica of sine signal but with a time shift. Given that x axis in the figure represent angle, the difference between the sin wave and the cosine wave is called *phase* difference.

As mentioned above, the phase of a periodic signal represents the moment in which the signal is shifted from its oscillation origin. The position of a pendulum in an old fashioned clock can also be mapped to a phase value. In complex values representation of periodic signals, a complex number is illustrated on a $2D$ plot, in which the x axis is dedicated to the real part of the complex number and the y axis belongs to the imaginary part. Typically, a complex value has a notation like $a+ib$ where i indicates the imaginary part. Having complex values along two axes, one can address the angle between two values (phase).

In signal processing applications, generally, the periodic signals along the time-line, are being transformed to different frequencies to study their composite sub-signals more accurately. In other words, the signal in the time-domain is composed of several sub frequencies (sins & cosines waves). These sinusoids are represented as a form of complex numbers so that one value (i.e. $cos(x)$) represents the real coefficient of the complex number and the other (i.e. $sin(x)$) represents the coefficient of the imaginary part (to pack both phase and power in a single complex value).

Having both imaginary numbers obtained from typical signal transformation function (Hilbert, Fourier, Wavelets, Laplace, ...), one can compute the phase of the signal. The phase ϕ of a complex value $C = a + bi$ can be computed as

$$\phi = arctan(\frac{b}{a}) \tag{3.6}$$

and the power p can be measures as

$$= \sqrt{a^2 + b^2} \tag{3.7}$$

It has been shown that the phase of neural signals contains predominantly more information compared to the amplitude [56, 88]. Several phase-based-features can be measured

to further characterize a particular brain activity. Features such as *phase-difference*, *phase-shift*, *phase-locking*, *phase-synchronization*, *phase-amplitude-co- upling*, etc, can be measured.

- **Phase-difference**

  A simple way to compare two phases is to measure the difference between them. Nonetheless, since the phase is a circular value, a difference of 10° and 350°, both can be considered valid. In a method suggested in [85], to solve the above-mentioned problem, the phase difference can be obtained as followings:

$$Phase_{\,diff} = min(|\phi_1 - \phi_2|\,,\, 2\pi - |\phi_1 - \phi_2|) \tag{3.8}$$

  where ϕ_1 and ϕ_2 are the phase of two signals.

  Partially, the term **phase-lag** is also used to address phase difference. They can be however used interchangeably.

- **Phase-shift**

  Phase shift expresses the changes in phase difference of (typically) two signals along time-line. Two signals may have $\Delta\phi_1$ phase difference at the beginning of the measurement but $\Delta\phi_2$ at the end. Phase shift shows this difference. Then,

$$Phase_{\,shift} = \Delta\phi_2 - \Delta\phi_1 \tag{3.9}$$

- **Phase-reset**

  Phase reset is a phenomenon happening in neural system for neurons or neural ensembles to sync themselves with a broad ongoing brain rhythm. In the recorded (i)EEG signal, the phase reset can be observed when the phase value drops abruptly to zero. Measuring the times and the number of phase resets in a certain frequency can be considered as an informative feature since phase reset is a sign of inter-frequency communications.

- **Phase-locking**

  For measuring the phase similarities of two trials, *phase locking value* (PLV) is defined. Different definitions of PLV are given. In a work reported in[70], the PLV is defined as following:

$$PLV = \frac{|\sum_{t=1}^{n} e^{i\theta}|}{n} \tag{3.10}$$

where n is the number of phase values at time t, i stands for imaginary part of complex value, ϕ is the phase, $|$ denotes the absolute value and $\theta = \phi_1 - \phi_2$. The resulting value will be in the range of [0 1]. If this value is close to 1, it indicated lesser phase variations across trials and conversely, near to 0 if two trials are diverse in term of phase. This method of calculating PLV is also know as *phase-coherence* in the field.

PLV is measured however differently in works published by field experts in our clinic [35, 36]. Accordingly, the PLV can be defined as following:

$$PLV_t = 1 + \frac{\sum_{I=1}^{8} X_{i,t} \times log\, X_{i,t}}{log\, 8} \tag{3.11}$$

where for each time point t, the 360° phase values are split into 8 bins in the range of $[-180° \quad + 180°]$, and X_i represents the ith distribution among 8 and the core calculation is the normalized entropy of phase distributions. To obtain more uniform results, in a repetitive procedure, the splitting point of eight bands can be shifted by 1° and the PLV can be measured again; finally, average over all calculated PLV values. This PLV is useful to measure the phase similarities across two electrodes.

- **Phase-synchronization**

  Phase synchronization refers to the synchronous oscillatory behavior of two brain regions in certain frequencies. The level of synchronization between neural ensembles plays a decisive role in the behavior of the neurons [34, 63]. Regions which are distant from one another, may still communicate through phase synchronization [34]. Phase synchronization is arguably an important mechanism for memory formation. The concept of phase synchronization is similar to phase-locking but it is measured across electrodes (vs. across trials). Having instantaneous phase values extracted from a signal (by means of Hilbert or Wavelets, ... transformation), we can measure the phase synchronization:

$$PLV = \frac{|\sum_{t=1}^{n} e^{i(\theta_x(t) - \theta_y(t))}|}{n} \tag{3.12}$$

  where x and y refer to two different regions (signals).

- **Phase-amplitude coupling**

  In several EEG and MEG studies, it has been shown that the changes of phase in lower frequencies has correlates to the changes in amplitude in higher frequency of same or other electrodes [14, 34, 87, 105]. Similar to the concept of phase-synchronization, phase-amplitude couping (PAC) relates itself to memory

formation processes and therefore, can be employed as a feature. Phase-amplitude coupling can be calculated as:

$$PAC = \frac{||A(n)e^{i}(\theta(n)||}{max(A(n))} \tag{3.13}$$

where n is a time point and A shows the amplitude of one frequency and (θ represents the phase of another frequency.

The power of the signal on the other hand shows how strong is a signal in a certain frequency range and certain time. The power and the amplitude of a signal are related as both pinpoint the magnitude of the signal but amplitude is a signed measure and can be negative and positive. Power is always a positive entity and represents the absolute value of the amplitude or the square of it. The power can be measured either from raw signal or from the the complex value of transformed signal (analytical power).

By now, our toolbox for feature extraction has the required tools at hand, and features which could be extracted from the power and the phase of the signal can make time-frequency-electrode cell feature schema available for further analysis. In the current work, I report the results of power-based features. However, various combination of power and phase features has been extensively tested.

3.1.1.4 Classification and prediction

Machine learning and classification is discussed more extensively in the chapters 2 and 4. If not sure about the terms used here feel free to consult with the machine learning material which are presented there. For the current paradigm (digit recognition), I restrictedly reported the results of using features extracted from analytical power. The features were then used for the pattern classification step.

In the pattern classification and learning phase, we have several problems to tackle. First, we deal with a type of data which is recorded from a handful of participants but they do not have the same electrode number and positional distribution over the skull. Furthermore, every participant has run the paradigm for tens of trials. Therefore, to make a machine learning solution functional, we have to train our classifier of choice for every person individually.

Second, from feature extraction phase, we obtained huge number of features. Just imagine if the participant has 20 electrodes, a frequency split of 8 bands, time bins of 20 ms, then for a 2 second trial, we end up having $20 \times 8 \times 100 = 16000$ features which is way to high for this problem with a very low signal to noise ratio. Additionally,

theoretically, the number of features should not be significantly more than then number of trials and the following should not hold $num.features \gg num.trials$ but the reverse. If we put that number of features in any machine leaning algorithm, it would have difficulties dealing with numerous non-informative features and generalizing in higher dimensions. Thus, a solution for feature reduction should be thought.

Third, the classification results must be arranged so that we can relate it clearly to the theories of cognitive neuroscience or at least, we must be able to localize the features which make the classification feasible.

Fourth, since in this field, due to its complexity, perfect results like 99% accuracy can not be achieved, and above-random statistically significant effects are considered to be fair an acceptable, an statistical test should be devised to ensure the reliability of our classification results.

In the following classification and learning phase, I address the above concern.

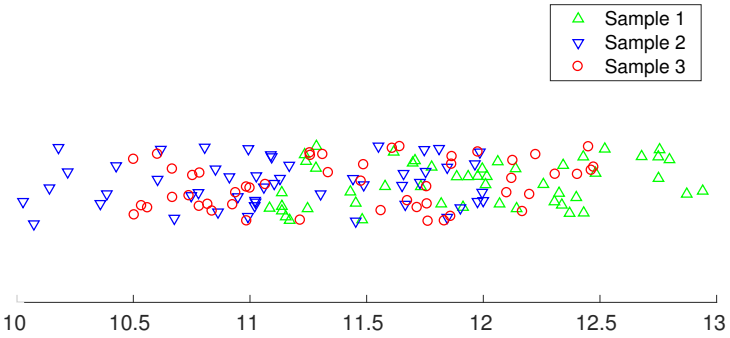
ANOVA and feature dimensionality reduction Earlier in this chapter, it is presented that we pick our features from a $3D$ tensor of time-frequency-electrode where each element of the tensor is a cell covering a time-interval, a frequency-band and an electrode. Therefore, the total number of features would be either equal to or multiple of the number of cells.

Let's now for the sake of simplicity, assume that we extract only one feature from a cell. This can be for instance the average power of the cell. Then, the total number of features will be equal to the total number of cells. We discussed previously that the number of features must be reduced to mitigate the classification process. This can be achieved by means of ANOVA (or analysis of variance).

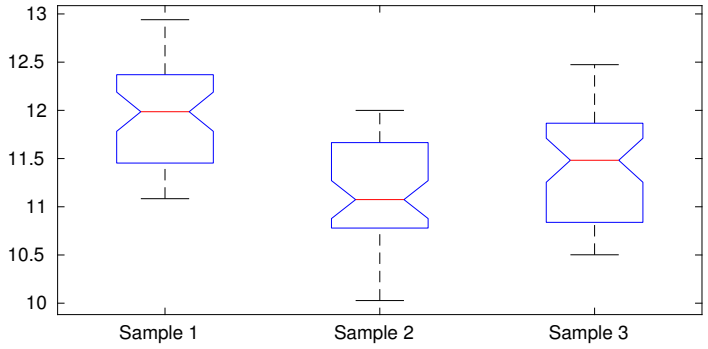
ANOVA is a test in statistics which measures how distant the "means" of samples of different classes are from each other. ANOVA checks the *null hypothesis*, whether two or more groups are randomly distributed, and reject the hypothesis if the groups are significantly differently distributed.

The test gives out two primary resulting values, F-value and p-values. The F-value tells how inter-class variance is larger than within-class variance and hence, the larger the F-value, more likable to reject the hypothesis. The p-value is a probability value and is in the range of [0 1] and tells about the likelihood of getting the F-value in a null hypothesis condition. If the p-value is less than $\alpha = 0.05$ then the test is considered to reject the hypothesis and we say "groups are significantly differently distributed".

With the help of ANOVA, we can rank the features based on their F-values and then keep the features with top F-values and discard the rest. To be able to achieve this goal, for each feature, we must first split the instances to their respective classes and feed them into ANOVA. That is, if we have 3 classes to classify, for each extracted feature, split the instances of that feature into 3 groups and then, test them with ANOVA. See Figure 3.12 for a hypothetical example.



(a) Sample data of 3 classes.



(b) ANOVA, analysing variance of classes.

Source	SS	df	MS	F	Prob>F	h
Columns	16.513	2	8.25652	24.37	7.23346e-10	1
Error	49.8032	147 0.3388				
Total	66.3162	149				

(c) ANOVA Table of statistics

FIGURE 3.12: ANOVA is a useful tool to test whether different groups (classes) of data are in fact different from each other. Plot (a) shows three groups of sampled data. For this sake, ANOVA examines the inter-class variability and compares it to the within-class variability. Figure (b) illustrated the mean and the standard deviation of each class. The result of ANOVA analysis is a table containing the test statistics (Figure (c)). The most straight forward result of the test is the h, having 0 for supporting the null hypothesis and 1 for rejecting it. That is, if the h is 1, it indicates that different groups in the testing population are distributed distinctively with different means. For h to be 1, the probability value p must be less than thresholding value $\alpha = 0.05$.

It should be clearly stressed here that the **ANOVA analysis applies only to the learning data**, to avoid leaking any information to the testing data set. After performing ANOVA test on all n features of the training set, n times F-value and p-value are obtained. In the next step, the features will be filtered based on their p-value and $\alpha = 0.05$ significance level threshold. Therefore, we end up having only significant features. Among them, we choose those with top 400 to 500 F-values.

It is noticeable that I put the p-value feature selection before F-value to make sure that only informative features are taken. That is, it is probable that a total number of features less than 400 were selected.

The number 400 is obtained empirically and proven to us to work best for different datasets. It also works for the mentioned dilemma of pf balancing between the number of trials and the number of instances.

There is however an alternative way for applying ANOVA in our work. Previously, at the time of shaping time-frequency-electrode cells, the idea of binning the data of 30 ms long was discussed. Instead, we can postpone the binning procedure to the phase after measuring the F-values. The latter trick showed to have slightly better performance than the former method and I mostly used the latter one.

Pattern classification schema As noted before, the classification should be accomplished for each participant individually since each of them have their own diverse electrode distribution and it will be hard to find a common number of features among them (as shown in figure 3.3). This does not mean that we wont be able at all to generalize over all patients later but it is more feasible and logical given our data.

In a typical pattern classification task, for a categorical data, the goal is to search within the data and discover a model to formalize the differences of different categories. This phase is called *learning* and is discussed more broadly in chapters 2 and 4. Since the classification performance must be always measured after the learning phase, there is a common procedure to split the data into two sets called *training set* and *test set*. Training set is used for the learning process and the test set is to evaluate the performance of the learned model.

Here, I applied a $5-fold$ *cross-validation* schema for splitting training and test sets, in which the data is shuffled and split into five subsets. Then for five time, one subset is considered as test and the rest four sets for training. Performing this procedure, five models could be obtained and therefore, five classification performance results. Ultimately, the final classification result would be the average of all obtained results.

To prepare the features selected from the last section for the classification phase, we stack all extracted features (power values in this report) of Time-Frequency-Electrode points of each trial into a single vector. In the training phase, these feature vectors are labeled with their original trial label, digits, and fed to the classifier.

We have employed a flavor of SVM [19] classifier called *Sequential Minimal Optimization* (SMO) [91] [3] to train a model from of the training samples. Next, to evaluate the accuracy of the classification phase, the already learned model was tested against test data. Comparing predictions and actual labels of the test data gives the classification accuracy.

Surrogate test One of our main concerns was to find a way to validate the significance of th classification results. Ergo, we need to prove methodologically that the classification results are not drawn from any random effect. For this sake, the *surrogate test* is devised.

The surrogate test algorithm takes the training data and shuffles the their corresponding labels. Then, by training the classifier with mislabeled data and testing it, a classifier accuracy performance can be obtained. Repeating this procedure for 100 times and acquiring 100 accuracy results, give us a distribution of random results. In the next step, we take the 95 percentile of the accuracy distribution and name it as *significance level*. Any accuracy above this level will be a reliable result. In the upcoming accuracy reports, the surrogate test results are presented too.

3.1.2 Results

In the followings, the results of Sternberg paradigm classification is presented. referring to the paradigm shown in figure 3.2, I conducted the machine learning approach for brain decoding on different segments of the data from encoding to retrieval.

As we discussed earlier about this paradigm, the task is classify the digit representation in the brain. The classification task is genuinely hard considering the design of the paradigm in which the participant had to rehearse the previously seen digits in mind. For memory encoding time interval, only one patient (patient $No.4$) showed an above random classification result. Checking any traceable effect in the memory retrieval interval, I found out 3 patients to have **number-effect** in their memory retrieval process.

Figure 3.13 depicts the classification results on the patients. Patient $No.4, 12, 13$ showed to have implanted electrodes which can trace the representation of numbers in their brain. The random level here is around 12.5% since due to a technical problem in the

[3] See chapter 4 for more details.

recording device, digit 1 could no be correctly recorded and hence, we have eight digits to classify.

In the figure, the surrogate test is also reflected. That is, the average surrogate test for each patient as well as 95 percentile is plotted. The results which are above the 95 percentile (yellow line) are considered to be significant results.

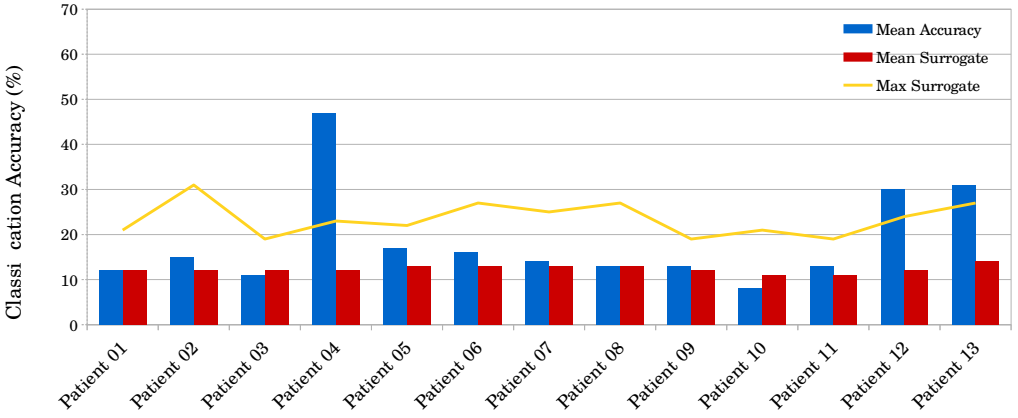


FIGURE 3.13: The classification results of Sternberg paradigm on the retrieval sections of the paradigm. The blue bar shows the classification results and the red bar shows the average surrogate accuracy level (= random level). The yellow line represents the 95 percentile accuracy of surrogate test. Any results above this level is considered as a significant result. Three patients show acceptable results in term of accuracy and it means, we are able to classify the number they thought of, well above random level.

Pattern classification is a tool to discover the patterns and thereupon, model them. The next step after modeling the brain activity in the field of cognitive neuroscience is to explain the causal effect of classification. Unfortunately, in most of the classification algorithms, it is really difficult to break down the classification effect into their pieces and determine their effects due to non-linearities of classification models (some tree-classifiers are able to do it though).

Fortunately, I designed our classification algorithm from the beginning to able able tell where the effects are coming from. I has been discuss earlier in this chapter about time-frequency-electrode cells and also about using ANOVA. Having these combinations, it will be possible to explore and back-trace the classification effects more. The time-frequency-electrode schema helps to localize the features on one hand and ANOVA allows only significant cells to take part in the classification process. That is, if some classification result is showing a significant effect, it is an indication that the underlying features were carrying some informative information and vice versa. Then, the selected

400 time-frequency-electrode cells could be used to address the causal effect. We empirically proved to ourselves that removing feature selection process will leave the classification result in the random level and therefore, proves the legitimacy of applying ANOVA on the features. Figure 3.14 shows two electrodes for each of the 3 patients which had the most prominent number-effect. The plots represent time-frequency-electrode cells in which features were selected.

It can be also understood from the plots that the selected features are mostly localized in few electrodes rather distributed across multiple electrodes considering the fact that we selected only 400 features. This would indicate the fact that most of the processing power of the brain used to solve the cognitive task in this paradigm is originated from local parts of the brain and is not distributed. Furthermore, The activities can be also cast across multiple frequencies as it can be observed in the plots.

Additionally, regarding the feature-times, where 0 indicates the time when the response is submitted by the patient, it can be noticed that the mental processes started before the answering action was physically performed. This effect can be observed since in some cases, we have features which are from negative time zones in the plots.

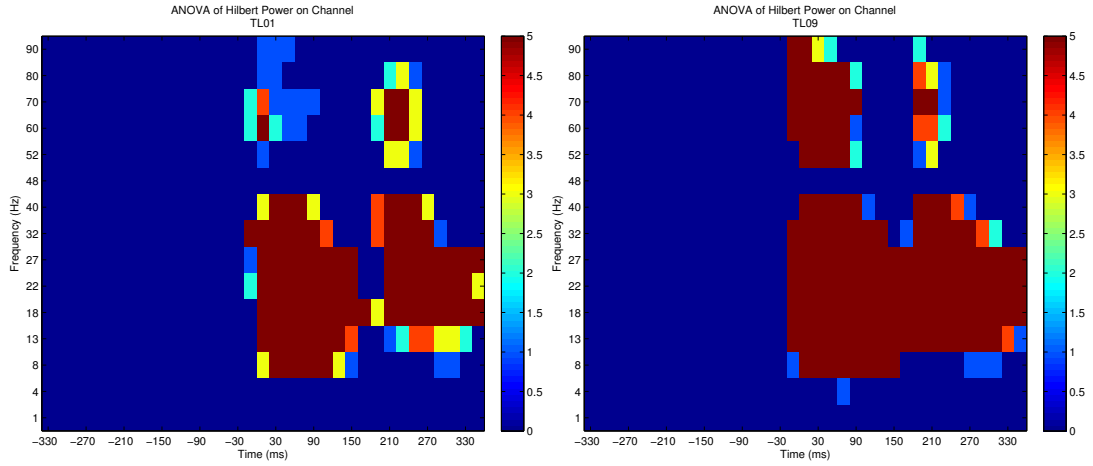
3.1.2.1 Discussion

The above result is by itself unique and valuable. However, by now we haven't published it since it does not show any generalized effect on all patients. We thought extensively about the possible causes and performed some further analyses. In some similar work published in [101], the task was to detect letters in an iEEG recording. They published their work with acceptable results. Therefore, we were interested to find out why this effect does not hold for all patients of our dataset. Finally, we visited the publishing group and acquire the data of two of their patients with good results. We then performed the following tasks.

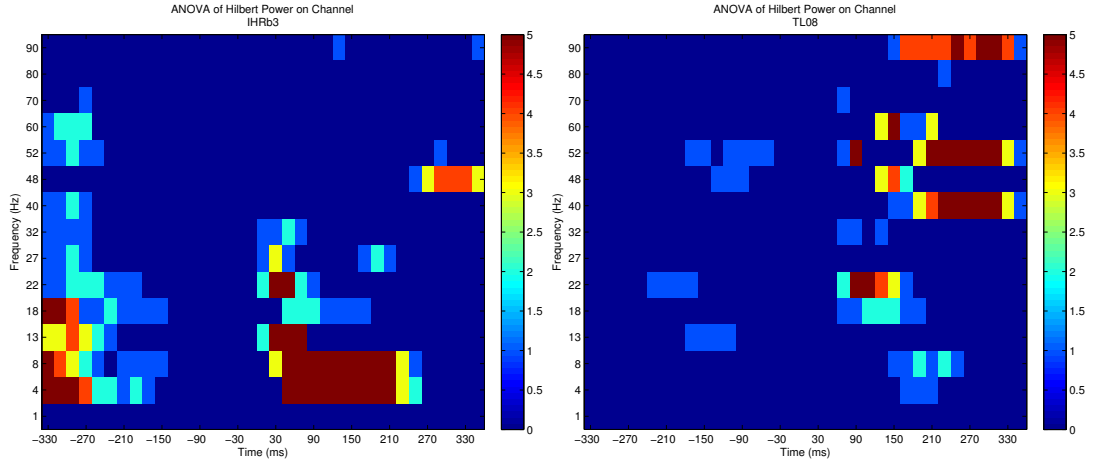
First, I applied our algorithm on their data and we got the same accuracy range as they published. The interesting point here was that most of discriminative features in their patients where rooted from regions near visual cortex. This will speak of classification based on brain visual features (Our patients did not have electrodes near those regions).

Second, I developed the algorithm of their work and applied it to our data as well as their data (elastic net logistic regression). The algorithm could detect the letters in their data but no digits from our data.

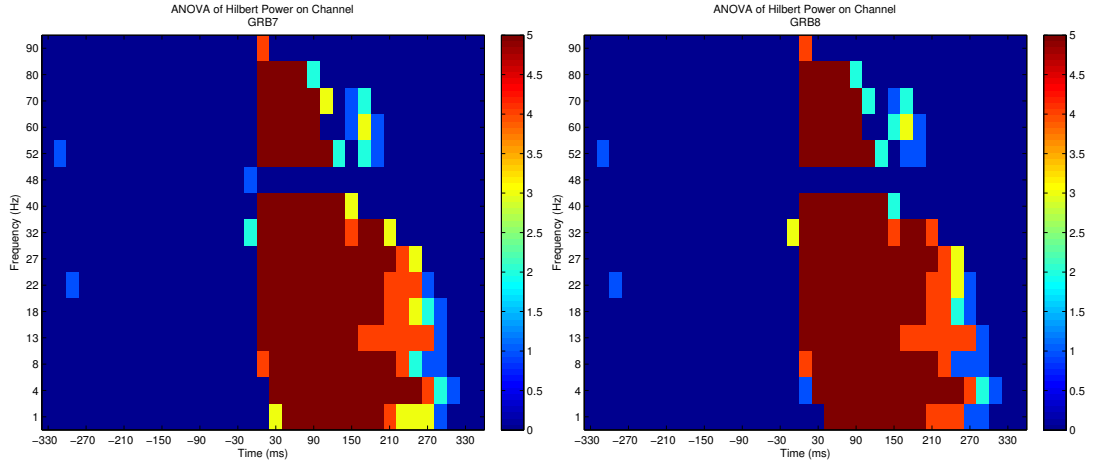
Accordingly, we came to the knowledge that two conditions should hold for us to be able to detect the digits. First, we have to have electrodes really close to the regions



(a) Patient 04



(b) Patient 12



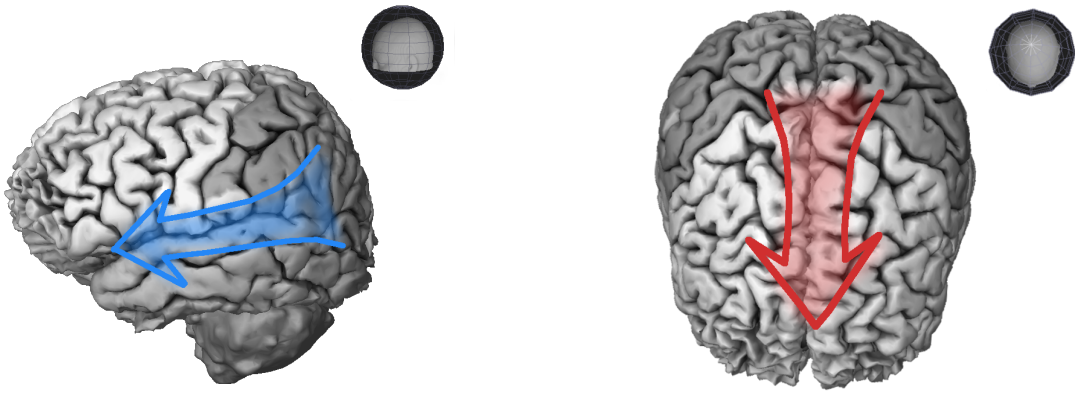
(c) Patient 13

FIGURE 3.14: Decisive feature cells. In this figure informative classification features of 3 patients are shown in panels (a), (b), and (c). The x axis shows the time. The time 0 indicates the time the patient's response registered in the computer. Negative time means the time before the response is submitted. For each patient, *two* electrodes with the most often selected features are shown. The time-frequency-electrode schema can be observed in the images. The dark blue color-code indicates that a cell is not selected for the classification and the other colors represent the F-value of ANOVA. Considering the fact that the total number of selected features are 400, it speaks of locality of selected features across the brain. These images indicate that the particular mental process are locally distributed. It can be also observed from the plots that the mental process of thinking about the digits stared before physically expressing them.

in the brain in which the person mentally processes the digit (like patient *No.*4) or we have electrodes near visual cortex. Since our epilepsy patients were implanted mostly in temporal lobe and frontal lobe, there was no chance for us to use the second possibility.

3.2 Face direction paradigm

In this section, the second iEEG study is presented. Visual data which are captured from either of our eyes are transmitted to the thalamus and then to the contra-lateral hemisphere. Referring to the chapter 1, the visual cortex is primarily dedicated to process the visual data from a basic edge detection to complex object detection and recognition. The processed visual data however is not meant to circulate merely within the visual cortex but to be transmitted to the other cortices to update the centers for memory, perception and attention. It is widely hypothesized that the visual information, after the elementary preprocessing steps falls into two primary data pathways to be processed in the form of working memory [42, 63, 75, 77]. In literature, these pathways are known as *ventral and dorsal data streams* (See figure 3.15). The ventral visual pathway is believed to transfer the data when the person wants to check the identity of a seen object. The ventral pathways is also known as "what pathway". In contrast, the dorsal visual pathway is assumed to convey the information when the features of a seen object is intended to be processed. The dorsal pathway is known as how/where pathway.



(a) Ventral visual stream

(b) Dorsal Visual Stream

FIGURE 3.15: Visual data streams. Figure (*a*) shows the ventral pathway of visual data stream. This pathway is hypothesized to convey information about the entity of a seen object. Figure (*b*) represents the dorsal visual stream pathway. Dorsal visual stream is meant to transmit the features of an observed object.

In the current study, a paradigm is designed to test the differences of brain visual data processing pathways. For this sake, we probe the epilepsy patients with the face of people in different conditions. Three testing cases were thought, *a.* identify the face of a person *b.* check the features of a persons' face *c.* see the face and do not process the face (control). Consequently, to make it possible for the patients to conduct the above conditions, for each trial of the paradigm, we present them with two consecutive images. The first is to show them a face and the second is to show them a comparison image to guide the participants to process the face information differently (delayed-matching-to-sample paradigm). The conditions are summerized below. The paradigm and the data is shared between this project and the work reported by Leszczynski et. al [77].

1. **Face identity** task: is to maintain the identity of a novel face presented and compare its identity to the second upcoming image. Here, we aim at checking for the ventral visual pathway or what pathway.
2. **Face direction/orientation** task: to process the orientation of the gaze of the presented face, match it with the position of a white square presented in second image (upward left, straight left, downward left, upward right, straight right, downward right). Here, we intend to direct the processing of visual information through the dorsal pathway or how/where pathway.
3. **Control/Gaze** task: not to maintain any item in the first image and tell the orientation of the gaze in the second image regardless of what has been shown in the first image.

In the Identity block, they had to press a left-hand button if the probe face matched the sample face and the right-hand button otherwise. In the Gaze direction block, they had to press the left-hand button if the position of the dot matched the gaze direction of the sample face and the right-hand button otherwise; and in the Control block, they had to press the left-hand button if the gaze of the probe face was to the left and the right-hand button otherwise. Thus, in the Identity block, participants had to maintain information about the identity of the face (but not on its gaze direction), while in the Gaze direction block, participants needed to remember the gaze of the sample face (but not its identity). In the Control block, participants did not have to maintain any information on the sample face. In each block, 50% of all trials were match trials and 50% were mismatch trials. There was no limitation in response time of the participant as the next trial started only when they had made a button press. After responding, participants received visual feedback on the accuracy of their response (by a green or a red). The feedback was presented for 500 ms followed by an inter-trial interval of 3000 ms duration. Blocks were interrupted by breaks of few minutes duration.

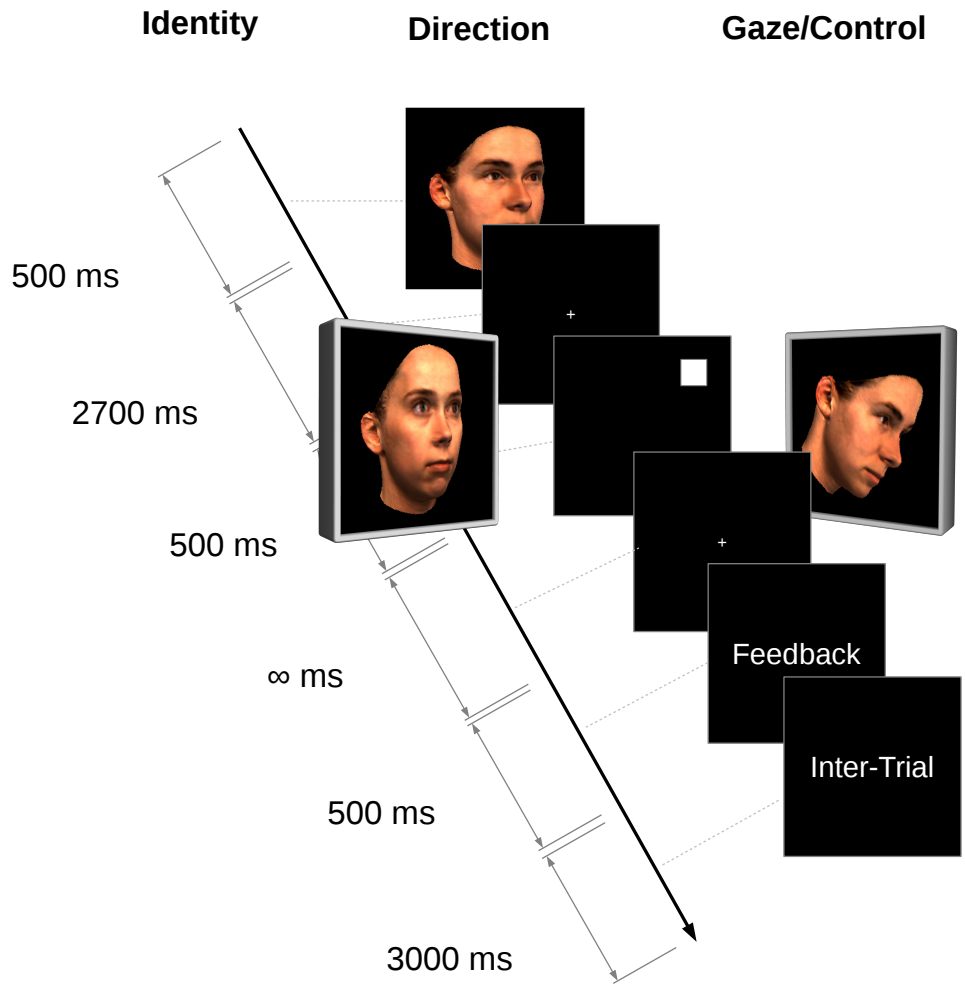


FIGURE 3.16: Face direction paradigm. The paradigm is composed of three blocks, "identity", "direction" and "control". In the "identity" phase, the participant is presented with a portrait image and asked to maintain the identity of the face. In face-"direction" task, the participant is asked to retain the orientation of the face. In "control" task, the participant is asked to tell the orientation of the gaze regardless regardless of the first image. For every trial, six events happens. 1. A face is shown for 500 ms. 2. An inter-stimulus time of 2700 ms time for maintaining/not-maintaining the seen face is given. 3. The second probe comes up for 500 ms and the participant has to check whether the first prob matches the second. 4. The participant has unlimited time to give his/her judgment. Meanwhile a fixation cross appears in the center of screen and the participant is asked to fix his/her gaze to the cross sign to avoid data contamination. 5. The correct answer appears on the screen for 500 ms. 6. An inter-trials interval of 3000 ms is given to dampen the memory process of the last trial.

3.2.1 Patients and data

Nineteen patients has been participated in the task. Due to data lost, iEEG artifacts and other technical recording issues, we excluded 5 patients. Therefore, we remained with 14 patients (8 female; age±SD: 35±11) and they were all recorded from 2009 to 2010. The patients were with pharmacoresistant epilepsy who had been implanted with intracranial electrodes for diagnostic purposes. Depending on the suspected ictal onset zone, patients had been implanted with subdural strip and/or grid electrodes. Different patients had different locations and number of electrodes (No. ContactsSD: 60 ± 17). In this report, the number of electrodes and the number of contacts are used interchangeably (see figure 3.17).

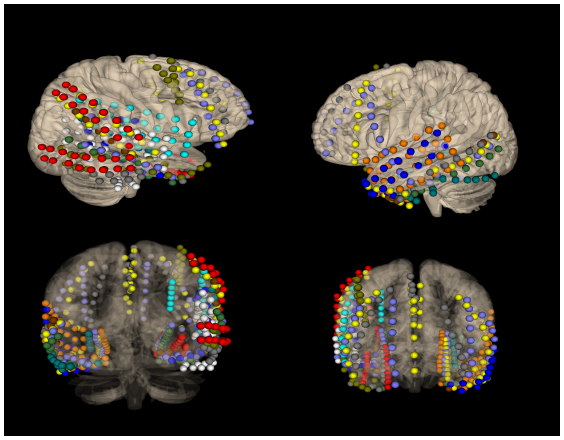


FIGURE 3.17: Electrode implantation of face-direction paradigm. Each color-coded ball represents an implantation position of a particular patient. The electrodes are localized by using pre- and post-implantation MRI images through Pylocator software.

The recording condition and the recording electrode and devices are the same as digit classification paradigm. Artifacts were automatically rejected using our Matlab script. Referring to our artifact rejection algorithm 4, trials containing more than one third of electrodes with epileptiform artifacts were removed from further analysis counting for propagating nature of epileptic activities.

3.2.2 Data analysis and classification

In order to obtain a model for brain's visual data streams in our paradigm, we labeled the trials of each block with the same category they belong to. Therefore, we end up with three classes of memory tasks: *Identity-Class*, *Direction-Class*, *Control-Class*. Multivariate pattern classification analyses were used to identify distributed activity patterns which differentiated between the three task conditions. The detail of classification phase

is identical to the previous paradigm and therefore, I skip it here. Nonetheless, the general settings is described below.

We excluded electrodes which are reported as pathological electrodes by clinical physicians. We applied artifact rejection algorithm to the data to exclude trials or electrodes covered with unusual spikes. To minimize the edging effects which occurs after the frequency filtering procedure, we segmented the data into relatively large time intervals of 2s before to 3s after the onset of the sample stimulus. Yet, we added a 1s flipped copy of the beginning of the signal to the beginning and a 1s flipped copy of the end of the signal to the end. The resulting data (7 second) were filtered using second order Butterworth filter in 8 common EEG frequency bands (delta [1-4Hz], theta [4-8Hz], alpha [8-12Hz], lower beta [12-20Hz], higher beta [20-30Hz], lower gamma [30-50Hz], middle gamma [50-75Hz], and higher gamma [75-110Hz]).

We tested a great range of feature choices to find the best feature which suits out model training and found out that power of Hilbert transform works best for our data. Next,we applied Hilbert transformation to the filtered signal and calculated the power. Afterward, we considered a five fold cross-validation schema in which we shuffled the data and split it to five folds, by considering each fold at a time as the test set and the rest as training set. We extracted frequency-specific power values by Hilbert transformation of stimulus and then disregarding both 3500 ms from the beginning and 1000 ms from the end. Empirically, we skipped over the baseline correction phase for this dataset since it showed no positive influence on the classification results which is also suggested in [59] to be occasionally legitimate.

We conducted a feature selection procedure by computing a one-way ANOVA of activity in the different blocks with block (Identity vs. Gaze direction vs. Control) and extracted F-values. Then, we averaged F-values in non-overlapping time bins of 30 ms for each of the 8 frequency bands, resulting in $83 \times 8 = 664$ values per trial per electrode. Time-bin intervals whose average p-value was above 0.05 were excluded. The total number of potential features was the number of electrodes times the remaining time-bin values per trial. Bins with higher averaged F-value selected as features. This resulted in 400 (SVM classifier) to 500 (Random-Forest classifier) values per participant chosen experimentally to pose less overfitting on the classifier of choice. In case of having less 400 bins, then we made use of all available non-zero bins. The selected bins were our informative clues and our features.

Selected electrodes are scattered across electrodes in middle and high gamma frequencies and concentrated in limited number of electrodes in lower frequencies. It is important to notice again that this was done in a 5-fold cross-validation schema in which 80% of the data served as a training set in the subsequent pattern classification analysis

and feature selected is first applied on the training set; then, the same selected feature indexes obtained from training set are selected from test set, to avoid any information sharing between training and test set.

In the next step for learning and classification: to make the features selected from the last phase prepared, we stacked all extracted power values of Time-Frequency-Electrode points of each trial in a single vector. In the training phase these feature vectors were labeled with their original trial label (*Identity-Class*, *Direction-Class*, *Control-Class*) and fed them to the classifier. We have employed Sequential Minimal Optimization (SMO) classifier for this paradigm too, to train a model out of the training samples. We made use of Weka [81] implementation of the classifiers with a wrapper for Matlab. In the next step, to evaluate the accuracy of our classification, we tested the already learned model against test data. Comparing predictions and actual labels of the test data brings out the accuracy.

3.2.3 Results

In the following, I present the classification accuracy results. A time period of 2500 ms after the offset of the first stimulus (encoding) is chosen as the main time period of interest which represents the memory maintenance time. Figure 3.18 illustrates the classification accuracy across three main conditions. Blue bars are the mean average accuracy of five folds of classification. To be able to estimate the significance of the classification, we have measured the accuracy of the surrogate data (as described in 3.1.1.4). According to the plot, all 14 subjects have significant classification result.

In addition to three conditions case, we can investigate the binary classification of two cases of above mentioned classes like (Identity Class/Direction Class), (Direction Class/-Control Class) and (Identity Class/Control Class) individually. The procedure of performing classification remains the same as mentioned above. In the following we report the classification accuracy of these cases (Figures 3.19, 3.20, 3.21).

3.2.3.1 Relevant electrodes and frequencies for classification using SVM

We extracted relevance (measured as a number of features selected) of each frequency and electrodes. In the following figures, the relevance of electrodes and frequency bands is highlighted by its size and color. The more the features were selected from a particular electrode in a certain frequency band, the bigger the ball becomes and more red-color is assigned. In this analysis , we considered the eight different frequency bands in our features selection process which are as explained above. (See figure 3.22). Based on the

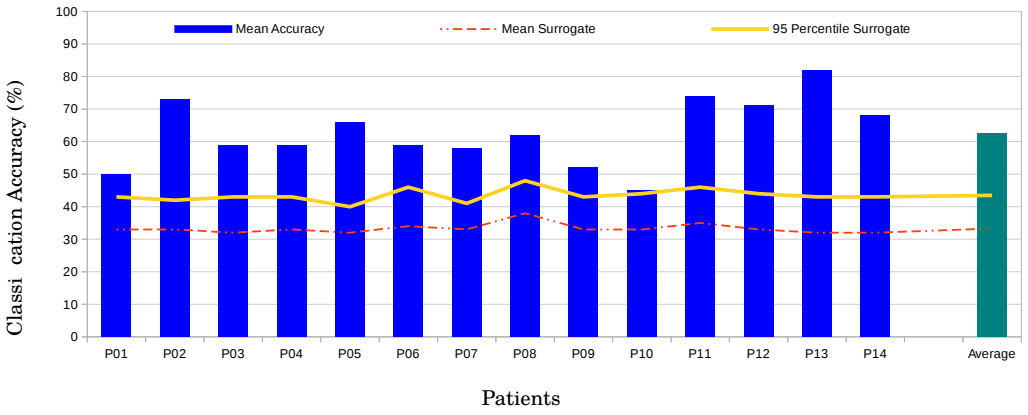


FIGURE 3.18: Classification results of 14 individuals for three conditions: face direction, face identity and control conditions. SMO classifier is used. The classification is applied to 2500 ms window after the first stimulus offset (maintenance phase). On the horizontal axis, the reference label of each participant is shown as well as the average classification accuracy. On the vertical axis, the classification accuracy (%) is illustrated. The blue bar in the plot shows the mean accuracy of empirical data classification across five folds for each participant. The yellow line represents the 95 percentile of surrogate classification results, indicating the significance level.

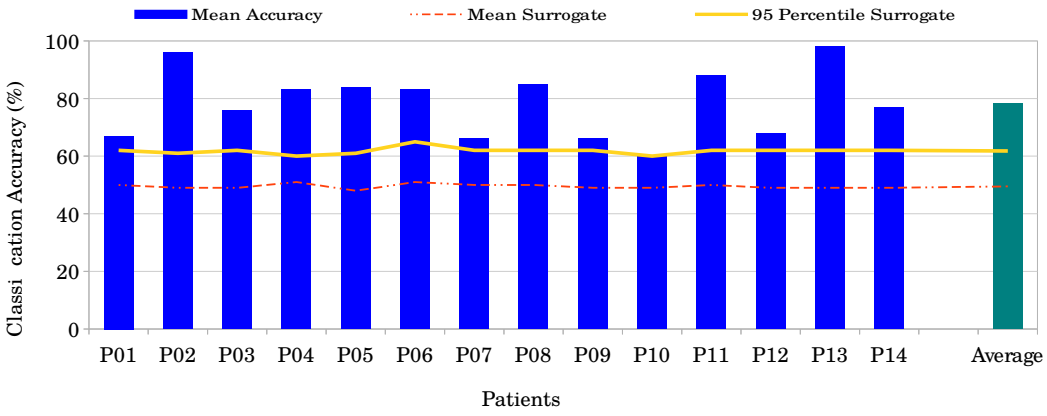


FIGURE 3.19: Classification results of 14 individuals for two conditions: face direction vs. face identity condition. The classification is applied to 2500 ms window after the first stimulus offset (maintenance phase). On the horizontal axis, the reference label of each participant is shown as well as the average. On the vertical axis, the classification accuracy (%) is illustrated. The blue bar in the plot shows the average accuracy of empirical data classification of 5 folds for each participant. The yellow line represents the 95 percentile of surrogate classification results, indicating the significance level.

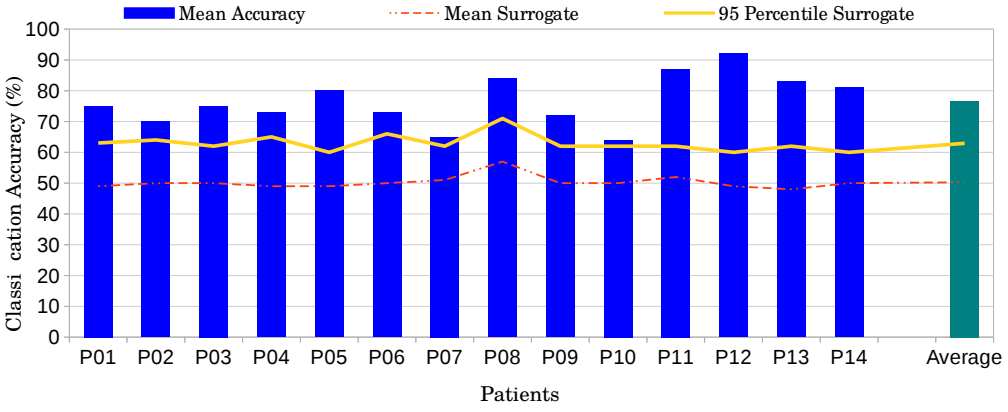


FIGURE 3.20: Classification results of 14 individuals for two conditions: face identity" vs. control condition. The classification is applied to 2500 ms window after the first stimulus offset (maintenance phase). On the horizontal axis, the reference label of each participant is shown as well as the average. On the vertical axis, the classification accuracy (%) is illustrated. The blue bar in the plot shows the average accuracy of empirical data classification of 5 folds for each participant. The yellow line represents the 95 percentile of surrogate classification results, indicating the significance level.

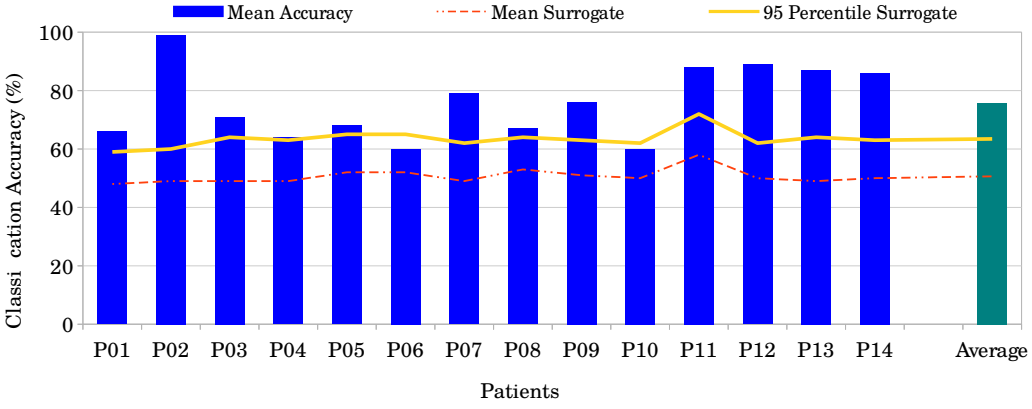


FIGURE 3.21: Classification results of 14 individuals for two conditions: face direction vs. control condition. The classification is applied to 2500 ms window after the first stimulus offset (maintenance phase). On the horizontal axis, the reference label of each participant is shown as well as the average. On the vertical axis, the classification accuracy (%) is illustrated. The blue bar in the plot shows the average accuracy of empirical data classification of 5 folds for each participant. The yellow line represents the 95 percentile of surrogate classification results, indicating the significance level.

images, delta and higher gamma frequencies play greater roles in modeling differences between ventral and dorsal visual pathways.

3.2.3.2 Checking feature importance using random-forest

By using a tree classifier, apart from the classification results, one can also obtain the importance of the features used in the classification. This is due to the way a tree classifier makes use of the features. In that, some features are randomly picked out of a set of all available features to take part in the classification process. Tree classifiers, step by step, split the samples based on each individual feature. Thus, during the training, the discriminability of each feature can be also obtained as a byproduct. This can help us to back-trace the classification effect to tell which parts were mostly associating in building the model.

Here, as a tree classifier of choice, I used random-forest classifier as it is believed to be a classifier with high classification accuracy in the field of machine learning and is known to have superb generalization abilities. After the training procedure in Random-Forest, every feature retains a ranking in the range of [0 1] for its importance in the classification process and consequently, can be used in our study to reveal the underlying brain activity. In the following, we report the results of pattern classification using Random-Forest. (see Figure 3.23).

The important features can be seen as those which survived the ANOVA feature reduction procedure. Based on the position of the electrodes and the frequencies in which the important features exist, we can plot them on the brain surface for each frequency band separately (see figure 3.24). The figure is spread into two pages. Corresponding to eight frequency bands. The figure in aligned in eight rows, each represents a frequency. At each row, two images are shown, each of them shows a brain hemisphere. On each image, red and black dots are shown. If a cell (frequency-electrode cell) was inactive, then the corresponding dot is black. If it is red, it means that the cell was active and the bigger the size of the cells are, the more important the cells are in the classification process.

3.2.3.3 Checking the importance of frequency bands

To test the importance of a single bands to the classification accuracy, I excluded, one frequency band at a time from the signal in each step, and then ran the 3-way classification as explained before. Bands were defined as: Delta (1-4 Hz), Theta(4-8 Hz),

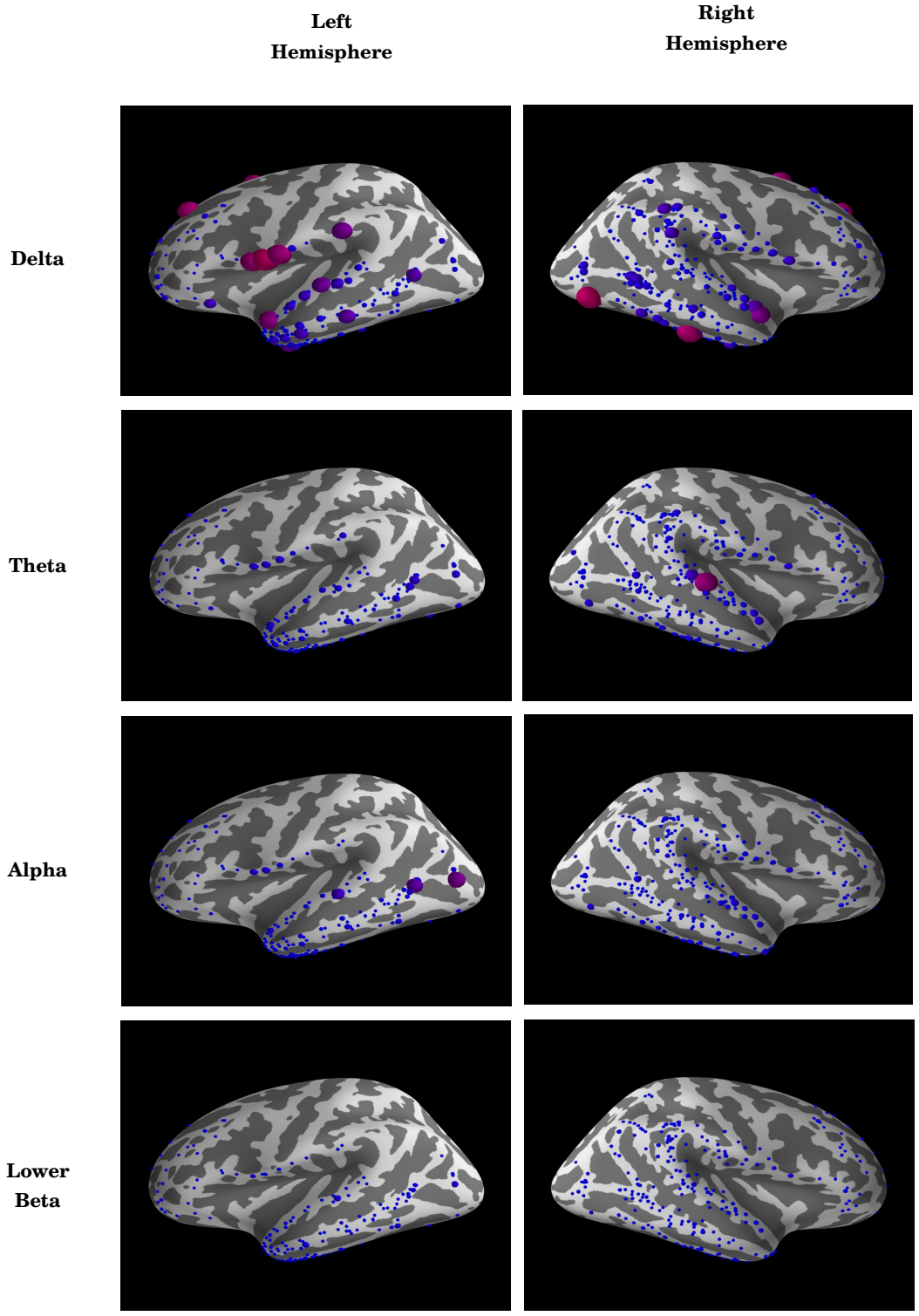


FIGURE 3.22: Continued in the next page ...

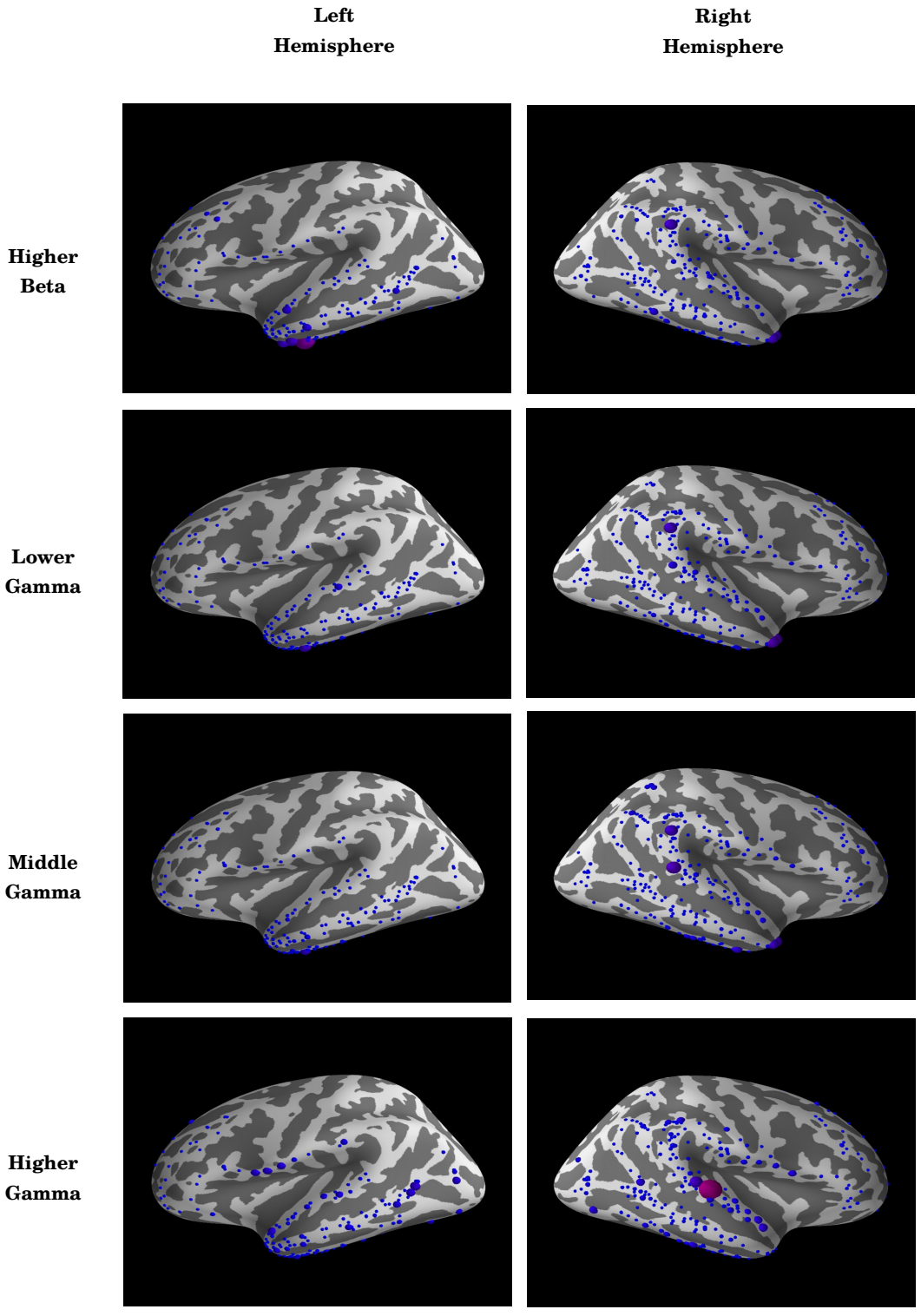


FIGURE 3.22: Pooled feature relevance plot. The features of all patients are pooled into the brain plots. The relevance of each selected feature is highlighted by the its color and size on the brain hemisphere across different frequencies. The bigger and more red a ball is, the more relevant that particular features was. It can be observed from the images that delta and higher gamma frequencies played greater roles in distinguishing visual pathways from another.

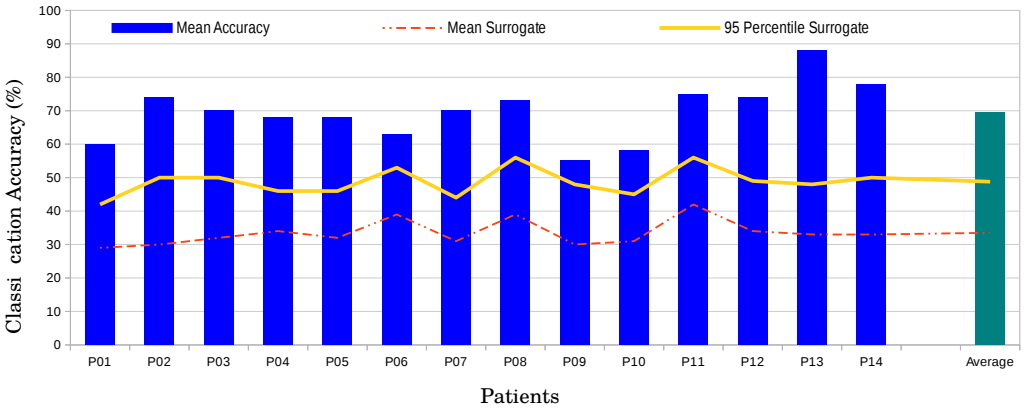


FIGURE 3.23: Classification results using random-forest for 14 individuals for three conditions: face direction, face identity and control conditions. The classification is applied to 2500 ms window of data after the first stimulus offset (maintenance phase). On the horizontal axis, the reference label of each participant is shown as well as the average. On vertical axis the classification accuracy (%) is illustrated. The blue bar in the plot shows the average accuracy of empirical data classification of 5 folds for each participant. The yellow line represents the 95 percentile of surrogate classification results, indicating the significance level.

Alpha(8-12 Hz), Beta(12-30 Hz), Gamma(30-110 Hz). Next, I compared the classification accuracies assuming that if a given band caries relevant information, its exclusion would result in accuracy drop. Table 3.3 summarized the results of t-test between all-inclusive frequencies case and one-band-out case. Excluding only the Gamma band (30-110 Hz) resulted in a significant ($p < 0.017$) drop (62.7 to 60) in accuracy while excluding other bands did not affect the accuracy significantly. This can indicate that other frequencies rather than gamma may share some information overlap but gamma contains some unique information which is not available in the other bands.

3.2.3.4 Checking the interplay of alpha vs. gamma frequency bands

Based on the work published by my colleague Dr. Marcin Leszczynski [77], we came up to the idea of testing the hypothesis of "interplay of alpha vs. gamma frequency bands" with machine learning techniques.

The maintenance of face identity has been proposed to be associated with increased gamma power over the medial temporal lobe (MTL) [63] and increased alpha power over dorsal stream. Alpha waves reflect inhibition of uninvolved brain area for the current task [66].

The opposite was also suggested for the maintenance of gaze direction. In the MTL, an increase in alpha power is observed and in the dorsal stream, an increase in gamma [63].

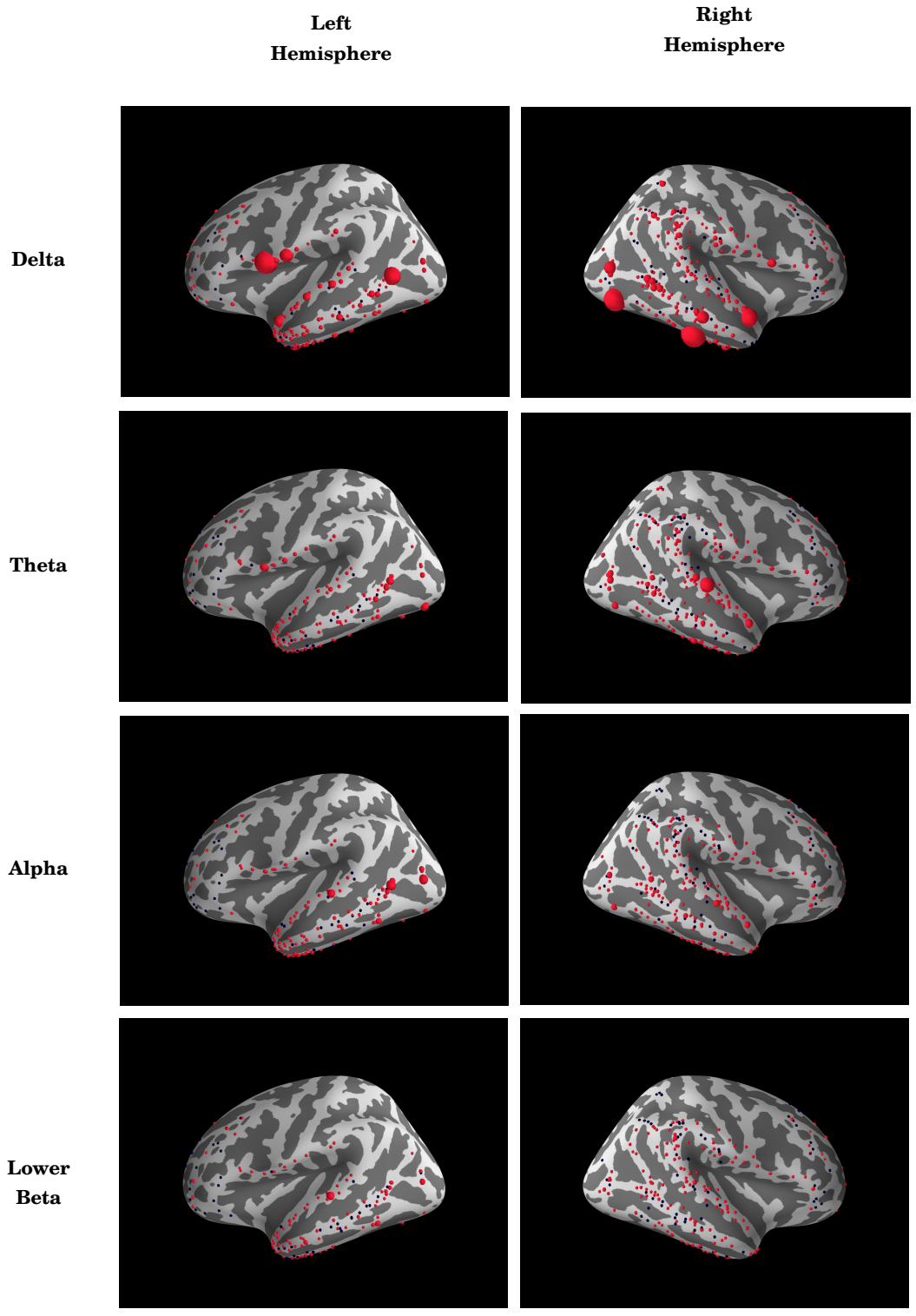


FIGURE 3.24: Continued in the next page ...

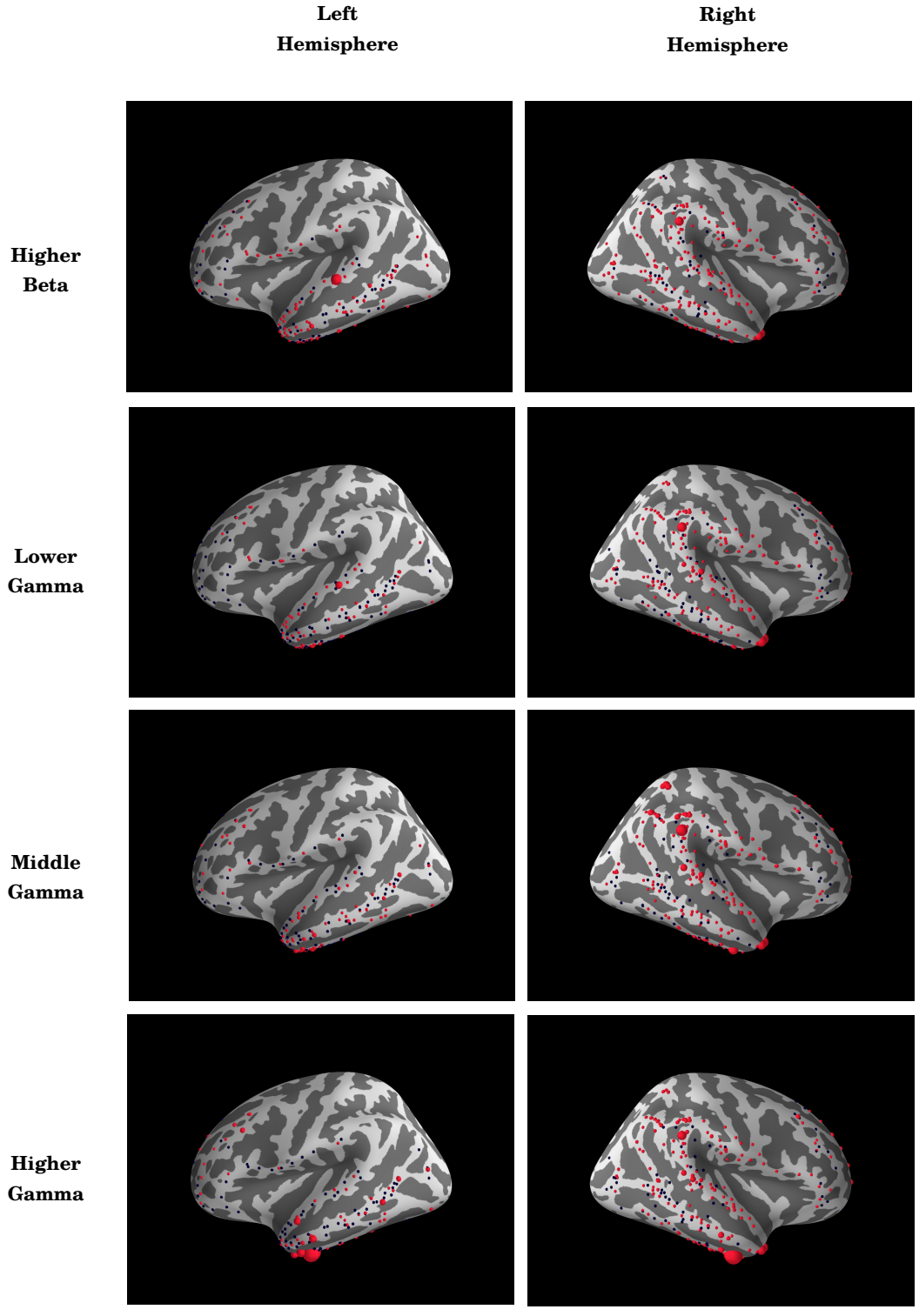


FIGURE 3.24: Feature importance plot. An importance metric for each feature in the classification process can be extracted from random-forest classifier. In the figure, the feature importance is pooled and plotted for all patients. Since the time-frequency-electrode schema is used, it is also possible to trace back the location of each important feature across different frequency bands and electrode positions. Having the MNI co-ordinations of electrode tips, we can localize the brain activity. Black dots indicate no-activity while red dots indicate that features from the an electrode is pariticipated in the classification performance. The bigger the size of the ball is, the more important that electrode is (higher F-value).

	All bands included		Delta band excluded		Theta band excluded		Alpha band excluded		Beta band excluded		Gamma band excluded	
Patient	**Mean Accuracy**	**95 Percentile Surrogate**	**Mean Accuracy**	**95 Percentile Surrogate**	**Mean Accuracy**	**95 Percentile Surrogate**	**Mean Accuracy**	**95 Percentile Surrogate**	**Mean Accuracy**	**95 Percentile Surrogate**	**Mean Accuracy**	**95 Percentile Surrogate**
P01	50	43	49	43	49	42	52	42	53	44	53	44
P02	73	42	78	43	70	44	73	42	73	42	73	42
P03	59	43	64	42	57	43	56	43	59	43	54	42
P04	59	43	53	42	59	46	60	42	58	43	59	44
P05	66	40	63	42	66	42	65	41	69	43	64	41
P06	59	46	55	45	64	45	58	45	60	45	56	45
P07	58	41	49	43	57	42	53	44	52	42	52	43
P08	62	48	55	43	62	42	64	48	62	51	62	48
P09	52	43	55	43	48	42	50	43	54	47	52	42
P10	45	44	42	42	47	42	44	41	41	43	41	43
P11	74	46	76	44	73	45	77	47	76	47	70	45
P12	71	44	70	42	75	44	73	44	67	43	67	44
P13	82	43	77	42	83	44	81	42	82	42	81	42
P14	68	43	65	41	71	42	72	42	70	40	68	41
Average	62.7142857	43.5	60.7857143	42.6428571	62.9285714	43.2142857	62.7142857	43.2857143	62.5714286	43.9285714	60.8571429	43.2857143
STD Error Mean	2.74047516	0.5522183	3.05657025	0.26945494	2.90218615	0.38055153	3.00078483	0.56867468	2.93278978	0.74468288	2.7968922	0.50740204

Classification Accuracy of Empirical Data

T-Test
All-Bands vs. Delta-Exclude

h	0	
p	0.1217	
ci	-0.5878	4.445

stats:	tstat:	1.6557
	df:	13
	sd:	4.3583

T-Test
All-Bands vs. Theta-Exclude

h	0	
p	0.7607	
ci	-1.7024	1.2738

stats:	tstat:	-0.3111
	df:	13
	sd:	2.5774

T-Test
All-Bands vs. Alpha-Exclude

h	0	
p	1	
ci	-1.4323	1.4323

stats:	tstat:	0
	df:	13
	sd:	2.4807

T-Test
All-Bands vs. Beta-Exclude

h	0	
p	0.8499	
ci	-1.4562	1.7419

stats:	tstat:	0.193
	df:	13
	sd:	2.7695

T-Test
All-Bands vs. Gamma-Exclude

h	1	
p	0.0169	
ci	0.392	3.3223

stats:	tstat:	2.7383
	df:	13
	sd:	2.5376

TABLE 3.3: Checking the importance of frequency bands. The upper table shows the details of classification accuracy for the cases of excluding one frequency band as well as surrogate test results. In the lower part, the results of t-tests for the case of band-reject vs. all inclusive frequency bands is presented. It can be understood that only by removing gamma activity, the classification accuracy will drop significantly and therefore, gamma should play an important role in distinguishing between three tasks.

Since most of the electrodes in our patients are localized in the MTL and also the fact that the MTL has been suggested to support the maintenance of face identity, we could test the whether the maintenance of face identity.

To this end, we performed some binary classifications: (Identity vs. Control) and (Direction vs.Control) and excluded alpha and gamma frequencies one by one [66]. We expected that excluding gamma frequency would deteriorate the classification performance between "Identity and Control" but keep intact the classification accuracy between Direction and Control. The opposite should be observed when we exclude alpha frequency band. Having only MTL Electrodes for two mentioned cases, we observed a significant drop in accuracy when classifying Identity vs. Control when excluding Gamma-band compared to all-inclusive case [$tstats = 2.5098,\ p = 0.0261$]. However, no significant change observed when excluding Alpha-band. On the other hand, we saw a

marginally significant drop in accuracy in classifying Direction vs. Control by removing Gamma-band [$tstats = 2.0334,\ p = 0.0629$] but not when rejecting Alpha-band.

Combining it with the results of previous section we can conclude:

1. Classifying by excluding one / including one frequency band: Only Gamma-exclude showed a significant drop
2. Classifying MTL Electrodes, by excluding one frequency band: Only Gamma-exclude showed a significant drop

Our results indicate also the inter-play between gamma and alpha while maintaining the identity of a face.

3.2.3.5 Electrode combinations

To check how the decoded information can be discovered all over the brain, we tried to find the optimum number of electrodes suits for the classification and extract ideal distances between their locations. To this end, in a very exhaustive analysis, for every participant, we classified the maintenance interval data of the three classes, by combining different number of electrode. All possible combinations of electrodes would lead to years or even decades of calculation. For instance, for 50 electrodes, there are $r = 1..50$ possible electrode numbers to choose, and for each number there would be $\frac{50!}{r!(50-r)!}$ combination, which leaves us with billions of combinations. To keep the number of combinations grow linearly, I designed an algorithm to alleviate the complexity, yet only partially sampling the space of all possible combinations. To summarize:

1. Calculate the classification accuracy for all single electrodes individually.
2. Among them, select 10% of the electrodes with the best classification accuracy and 10% of random electrodes from the remaining channels. Then, calculate the accuracy from all possible binary combinations.
3. Among them, select 10% of the combinations with the best classification accuracy and 10% randomly from the rest of electrodes which were not included in the chosen combinations. Then, calculate the accuracy of adding those newly selected electrodes one by one to survived combinations.
4. Continue the previous step until no other electrode remained to be added.

In this way we have excluded the non-optimum combinations from the total calculation procedure and the processing time for each participant has been scaled down to 1 month in average. For this calculation I made use of DZNE [4] institute data cluster with 192 cores. Figure 3.25, represents the maximum accuracy obtained from having different number of electrodes in the classification. It can be seen that the classification can be done even by having a single electrode. However, the best combination obtained when we have at least 3 or 4 electrodes used in the classification procedure. I then, ran some t-tests to check the effect of having more electrodes for iEEG classification. The difference of having one electrode compared to two is significant ($p < 0.00034$) and also the difference of having two electrodes vs. three is significant ($p < 0.003$) but the difference between having three electrodes to having four electrodes is not significant ($p > 0.63$).

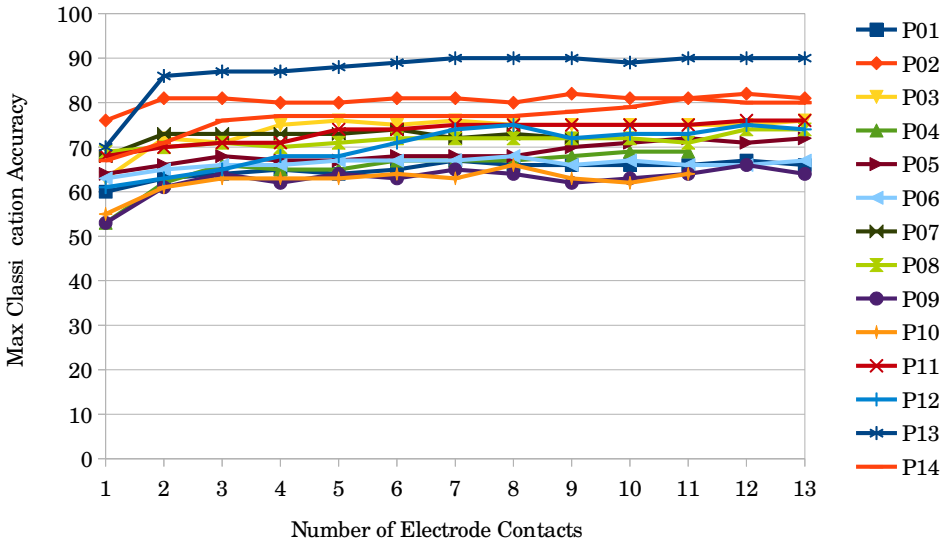


FIGURE 3.25: Combinations of electrodes used for classification. For each patient there is a maximum accuracy of having n electrodes. It can be seen that with one electrode, it is still possible to perform the classification above the chance level. However, having two and then three electrodes, the classification accuracy significantly increases. From having four electrodes and more on, adding extra electrodes does increase the maximum accuracy significantly.

3.2.3.6 Potential confound variables

The current design-block in which the conditions are arranged block-wise and away from each other in time, and with few minutes of inter-block intervals, one might argue

[4] Deutsches Zentrum fr Neurodegenerative Erkrankungen, DZNE, www.dzne.de

that the classification results could be affected by some time dependent (other than condition specific) external or physiological signal. To investigate the effect of potential confound variable in our data, we have checked the order of tasks in which the blocks were performed for every patient.

Block halves classification We divided each block into two halves. Then, a binary classification was performed on the halves of classes. To explore the effect of time on the classification accuracy (SVM/SMO), we performed classification between outer halves of the first and the second block, those halves which were temporally more distant. In another run, we ran classification between inner halves of the first and the second block, those halves which are closer/adjacent in time (see figure 3.26).

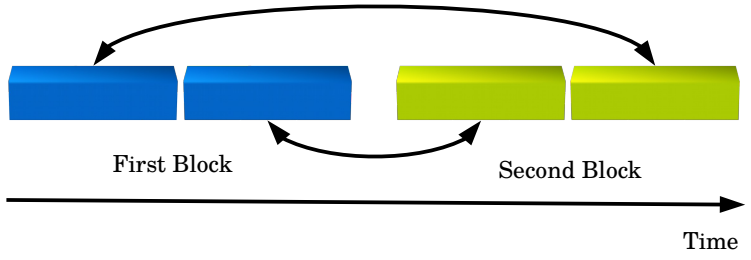


FIGURE 3.26: Classifying block halves. Each color (blue and green) represents a block. Trials of each block in the figure are divided into two halves, the earlier and later in time. The classification will be then proceed with two settings: inner halves, those are temporally close to each other and outer halves, those are temporally far from each other. Then, the classification accuracies of both settings are compared. In particular, we used machine learning and statistical tests to validate another machine learning task.

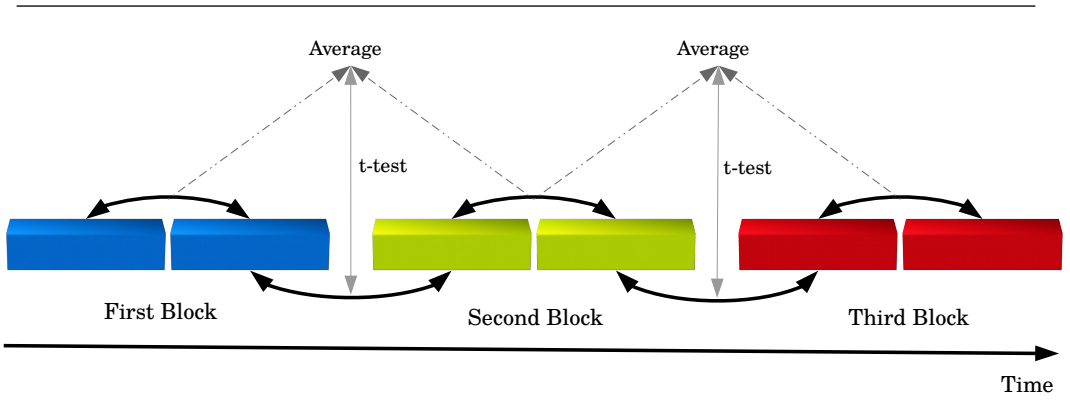
Having accuracy results of two classification cases, we can run a pairwise t-test to determine if the classification results of inner halves differ from the classification of outer halves. Assuming that some hidden time-dependent variables might contribute to classification accuracy, we expect a significant t-test result. Since there was a time gap between consecutive blocks, to have equal gaps between block halves and inter-block interval, I excluded some trials from the middle of each block equal to the length of inter-block gap. For this analysis, I excluded two patients who had done the task with inter-block time intervals more than half of the surrounding blocks. Testing 3 possible binary classification combinations, Identity vs. Direction, Direction vs. Control and Identity vs. Control once for inner halves and once for outer halves, there was no significant difference between inner and outer halves of all possible binary classifications. This suggests that the time dependent noise if at all does not contribute to the classification results significantly.

Between blocks vs. within blocks classification To further investigate the possible contribution of time dependent noises to our results, I aligned all blocks in their true temporal order and divided each block into two parts. Subsequently, I measured the classification accuracy of adjacent block halves, no matter if they were from the same block or from consecutive blocks (first part of the first block vs second part of first block; second part of first block vs. first part of second block, etc). As a result, 5 different classification results were obtained, 3 for block halves of the same blocks and 2 for between blocks. The reason here was to test blocks of same temporal distance that either come from different classes (between blocks classification) or come from the same class and only time dependent factor contributes.

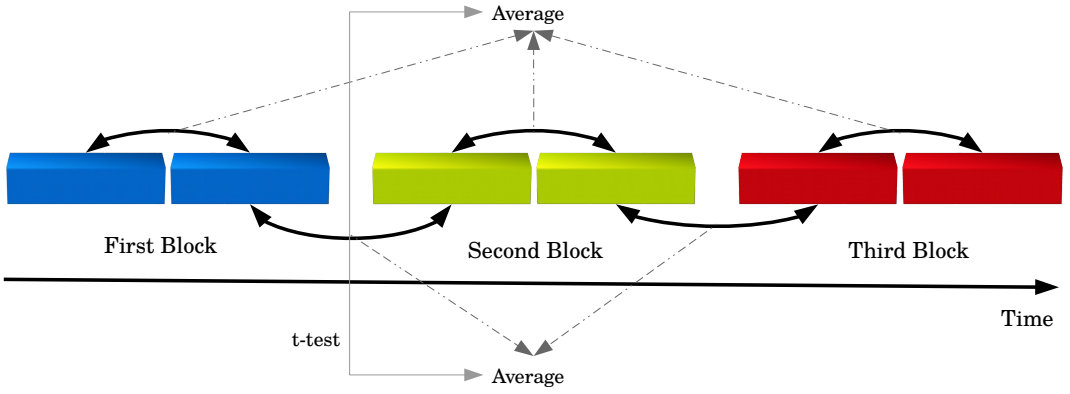
We expected to observe significantly higher classification accuracy for between block as compared to within block classification (task related brain activity). This would indicate that despite any potential time correlated noise, the difference between conditions can be reliably classified. We considered two settings. For the first setting, we averaged the accuracy of within block classification of first and second blocks and compared it with the classification accuracy of the second part of the first block vs. first part of the second block by doing t-test. We got $[t = +2.7354,\ P < 0.01]$. In addition, we compared the average accuracy of within block classification of the second and third blocks to the classification accuracy of the second part of the second block vs. the first part of the third block and we got $[t = -2.0995, P < 0.05]$ (see Figure 3.27(a)).

For the second setting, we separately averaged classification accuracy across three within-block halves (i.e. accuracy of the classification between halves of the first block, second and third block) and two across block halves (i.e. accuracy of classification between the second half of the first block and the first part of the second block, the second half of the second block and the first half of the third block). Next, we compared this aggregated accuracies with t-test and we got $[t = -2.2089,\ P < 0.05]$ (see Figure 3.27(b)). These results would support the idea that the main effect in classification is not driven from time based factors.

Distant vs. adjacent classification To further investigate the possible contribution of time-based causes in the classification results, I re-ordered the results of binary classification blocks temporally. As a result, we got class labels as: (First Block vs. Second Block), (Second Block vs.Third Block) and (First Block vs. Third Block). Next, I ran t-test among the combinations of mentioned classification results. It turned out that the classification accuracy of first vs. second and first vs. third is significantly better than the classification of second vs. third block $[t = +3.642,\ P = 0.003]$ and $[t = -.4.577, P = 0.0005]$. However, the classification accuracy of first vs. second



(a) Setting 1



(b) Setting 2

FIGURE 3.27: Comparing consecutive block halves of adjacent blocks. 5 classification sets obtained, 2 sets for between blocks and 3 sets for within block classification. Performing a t-test between these two conditions would indicate the existence of any prominent time-based noise.

compared to first vs. third is not significantly different. The interpretation of this result is complicated. On one hand, it speaks of the effect of the first block in the classification process and on the other hand, it indicates that there is no time-based noise in the late stages of the experiment. Consequently, it is difficult to take a side.

3.2.4 Discussion

In this study through an experimental paradigm, and with the help of pattern classification techniques, we examined a theory in the realm of cognitive neuroscience. The question I posed was to check for patterns in which the ventral and dorsal visual streams can be distinguished from another. According to our results, these patterns can be found mostly in delta and gamma frequency bands and gamma contributes significantly in the classification process. I proved that the classification results are significantly above the

random level and hence, reliable. I also found out that the power (or amplitude) of the signal, as a feature, can pinpoint the types of visual data streams activities. Additionally, these pattern can be extracted from a single electrode if it is positioned in the right place (e.g near medial temporal lobe) and with having three good electrodes, we can reach close to the best classification performance even if we do not have any electrode in the dorsal stream regions. This makes sense since the absence of power in some particular frequencies may indicate that the brain is processing in other regions.

The above results, corroborates some of the previous findings and adds to them. According to our tests to search for temporal confound variables, our analysis passed two of them but we could not draw a clear conclusion from the third one. Thus, we decided not to publish the work since it is not possible at this moment to prove/disprove the existence of temporal confound variables.

3.3 Applying deep learning to iEEG data

I have developed deep learning algorithms for two iEEG studies. In the first case, two algorithms were developed for Sternberg paradigm to classify digits. Secondly, an algorithm was developed on the data of an already published work from our group [26].

3.3.1 Deep learning on Sternberg paradigm

I have exhausted numerous deep learning algorithms on iEEG data of Sternberg paradigm, all under two branches of deep learning algorithms: modern multi-layer perceptron and convolution neural networks (CNN). The technical details of developing mentioned deep learning algorithms are covered in chapter 2 and also in chapter 4 in the deep learning section. Below are the short descriptions of the settings and results.

For the case with deep multilayer perceptron, the only main modification to the presented settings of this chapter was to replace the SMO classifier with the deep neural network. Various adjustment are tried, and it turned out that they do not perform better than our conventional classifiers. As with using SMO classifier, I could decode the brain activity of three patients above random level, with deep multilayer perceptron, it failed to perform that well (only patient no.4 was above random level, 20% accuracy). I have also tried to reproduce more samples (by data imputation, data augmentation), but they were of no further help to the classifier.

In a separate run, I have also developed a CNN network and fed the time-frequency-electrode plots as images to the network. This procedure was meant to resemble the

mainstream image classification tasks using CNNs in which, the feature extraction is performed automatically by the network. Performing this procedure with various layers, filters, and drop-out settings did not bring any results better than random.

These results can indicate the importance of applying feature extraction and feature selection in cases in which the number of features are significantly higher than the number of samples. Technical facts also support this reasoning. In our data, the number of features are really higher than the number of instances. Yielding a neural network from such data will build a wide network (as opposed to deep). In theory, wide neural networks are harder to train and, require more data than narrower ones. The amount of diversity reflected in the data is not sufficient enough to train such network; and this is the reason why our proposed framework work better.

3.3.2 Deep learning on Derner et al. data

A machine learning work from our group [26] has used the phase of iEEG signal to decode brain activities using SVM.

In the study, 27 epilepsy patients participated who were implanted with intracranial electrodes in the entorhinal cortex and hippocampus. The patients' task was to perform a word recognition paradigm, in which German nouns were presented. There, in two rounds, 450 words were presented, 150 words only once and 150 words with one repetition. The presented words were shown for 300 seconds and the inter-stimulus interval varied from $1600ms$, $2000ms$ or $2700ms \pm 200ms$ adjusted based on the initial performance of the patients. Individuals had to decide whether or not, the observed word is presented before. Two classes of data were defined: "remembered" and "forgotten" based on the performance of the patients in the second round of word presentation. The phase of iEEG data were extracted from different electrodes and were used in the pattern classification procedure.

I examined the corresponding dataset with a deep learning algorithm to see if any improvement can be achieved. In the published work, two different phase based time-frequency arrays were presented. In addition to them, I have added two power based time-frequency arrays and made a $4D$ data out of it. That is, every single trial was represented by a $4D$ image. This data can be put in a CNN (similar to image processing tasks) to classify the patterns into their corresponding classes (binary classification).

Trials of all two classes were upsampled (augmentation techniques) by making different combinations of images of the $4D$ data to 5000 samples for each class.

At the input level, the data were fed as an array of $100 \times 100 \times 4$ to the network. Three convolution layers were considered ($2D$ convolutional), each with a proceeding pooling layer:

- Convolutional layer 1 (num. filters=8, kernel size=2, activation function=ReLU)
- Max pooling layer (pool size=2)
- Convolutional layer 2 (num. filters=4, kernel size=2, activation function=ReLU)
- Max pooling layer (pool size=4)
- Convolutional layer 2 (num. filters=4, kernel size=2, activation function=ReLU)
- Max pooling layer (pool size=4)
- Dropout (20%)
- dense layer (activation=softmax)

By applying training the network and testing on it, no accuracy better than 55% is achieved which is not superior to the results reported in the publication. We can argue the same as with our previous deep learning analysis that the amount of diversity in the data is not sufficient to train the neural network better than other algorithms.

3.4 Summary

In this chapter, a machine learning based framework for analyzing (i)EEG data is introduced in which the following points are addressed and discussed in detail:

- How the data should be seen (time-frequency-electrode cell concept)
- How to remove artifacts from the signal
- How to transform the data to gain more information (from power and phase)
- How to measure features from power and phase
- How to reduce the dimensionality and complexity (ANOVA feature selection)
- How to train and evaluate the data
- What is the optimum number of electrodes and how to find it

- How to infer additional facts in post-classification phases by applying machine learning on different segments of data

The above mentioned point were presented through two dataset, "Sternbeg" and "Face direction" paradigms to show how far we can go with machine learning algorithms in analyzing and decoding (i)EEG brain data.

Finally, a short reference to deep learning algorithm is given, to check whether or not using deep learning is advantageous on these types of data.

Chapter 4

A multimodal, non-EEG based approach to detect epileptic seizures

4.1 Introduction

4.1.1 What is epilepsy?

Epilepsy is a common brain disorder among all nationalities. Typically, one percent of the population in every society experience epilepsy in their lives. People with epilepsy have epileptic seizures in which brain activity temporarily blocks the brain from carrying out its normal functions. If some part of the brain malfunctions during an epileptic seizure, it might be trapped in a synchronistic pattern with other neurons instead of carrying out its normal task. As a result, abnormal commands are then sent to segments of the brain and organs which are connected to that part of brain [4]. For instance, in case that an epileptic seizure attacks the movement area of the brain, the behavioral response would be involuntary muscle contraction or random limb movement. Different people may encounter with different types of epileptic seizures.

Various types of seizures have been classified based on their characteristics. One marker used to identify the seizures, is the *focality*. *Focal* (or *partial*) seizures are those which affect only a region of the brain, whereas *generalized* seizures which spread vastly in the regions of two hemispheres. Generalized tonic-clonic seizures (GTCS) are well-known types of generalized seizures which are involved with severe body movement. Focal

seizures can be *Simple Partial Seizures* (SPS) which occur with short loss of consciousness but with no significant irregular movement or behavior. In contrast, *Complex Partial Seizures* (CPS) affect only one region of the brain but are accompanied by seemingly meaningless types of behavioral activities such as smacking, licking leaps, or senseless laughing [4].

In addition, the origin of an epileptic seizure can also be viewed as a criterion for classification. As mentioned earlier, each hemisphere of the brain is segmented into four regions called lobes, namely (*temporal*, *frontal*, *occipital*, and *parietal*). In practice, it is possible to distinguish between seizures which are originated from frontal lobe and the seizures originated from temporal lobe.

The methods and medications used for treating epilepsy patients vary and are based on the type of epilepsy which has been diagnosed. One standard method for detecting the epilepsy origin is electroencephalography (EEG). In an EEG study, electrodes are either attached to the surface of the skull, implanted subdurally, or implanted deep in the brain tissue to record and track the electrical charges of the brain. If only certain electrodes sense neural abnormalities during a seizure, then the seizure is classified as focal and the origin of the seizure can be determined. Likewise, if signal abnormalities are detected throughout the brain, then the existence of a generalized seizure can be ascertained. Additionally, an electrode displaying early signs of signal irregularity would determine the origin of the seizure.

4.1.2 Seizure detection systems

In order to implement optimal drug treatment for controlling epileptic seizures, specialists must know the exact number and frequency of the seizures for each patient. To this end, epilepsy patients are advised to keep a diary and document their seizures as soon as they occur. However, many patients are unaware that a seizure has taken place or they simply forget to document them afterwards, thus rendering the seizure diary a very inaccurate and unreliable tool [37, 58, 65].

With assistive technology, patients can increase awareness of their bodies and their environment. To increase the performance of seizure counting, assistive systems have been developed [23, 78, 92, 100]. These assistive technologies try to trace and detect the seizure signs and symptoms and to differentiate between seizures and non-seizure events. The trend in recent years has been towards developing seizure detection systems for daily use which can be easily worn by the patient and are non-invasive.

EEG is probably the most reliable tool for detecting seizures. However, using EEG for recording and tracking seizures on a daily base is not optimal, even with the available mobile EEG systems [27, 28]. Attaching electrodes and wires to the head and then creating a recording which is susceptible to noise makes using a daily-based EEG recording system impractical. In addition, a portion of focal seizures can not be tracked with surface EEG. In recent years, there has been endeavors to develop non-invasive seizure detection systems which could be used daily and easily by patients. Moreover, other implantable intracranial EEG recording systems, despite their high signal-to-noise-ratio, are invasive and require surgery [52].

To develop non-EEG seizure detection systems, the effects of a seizure should be investigated using other biomarkers. Studies reveal that both the sympathetic nervous system (SNS) and parasympathetic nervous system (PNS) reflect some variation in their normal behavior during and after seizure events [79, 93, 99].

The physiological effects of epileptic seizures can be sensed in several regions of the patient's body. During a seizure, the heartbeat pattern deviates from its normal conditions. These changes are referred to as heart rate variability (HRV). Apart from HRV, the conductivity of the skin alters during a seizure. Tonic-clonic seizures can be easily observed as they involve involuntary movements. Muscle contraction can be also tracked during an epileptic seizure using electromyography (MEG).

Although automatic seizure detection devices have previously been proposed and tested especially within the context of predominant ictal motor signs [23, 78], our goal is to develop a wearable and portable multisensory-system for automatically detecting and registering all types of seizures. To this end, we used an ECG sensor alongside with three acceleration sensors all embedded in 3 comfortable-to-wear sensor units. In the final version of the application, to have the system prepared for the home usage, an application is developed for mobile phones to keep track of the sensor data online and to annotate and register the seizure-like events (see figure 4.1).

4.2 Related work

The theme of automatic seizure detection system has been the subject of many studies.

Dalton et al. [23] proposed a portable seizure monitoring system, a digital watch capable of connecting to Wi-Fi and an accelerometer sensor to track and register the motor seizures. They conducted their study in an epilepsy clinic to obtain the seizures' ground-truth by video-EEG monitoring. They asked patients to perform daily life activities during their residence in the clinic.

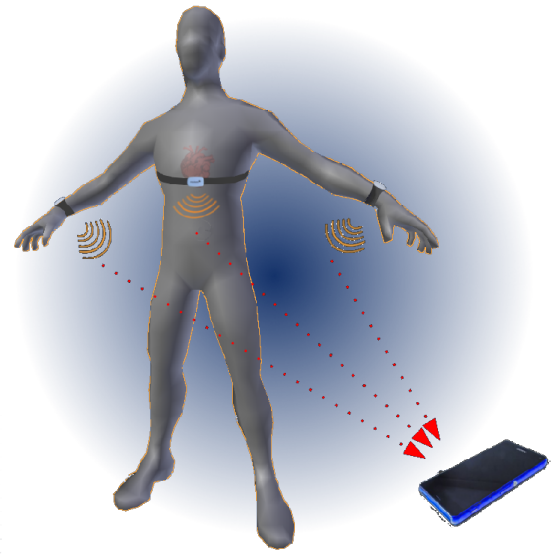


FIGURE 4.1: Wearable sensor units. The ultimate goal of the project was to develop a system in which, different sensors all along the body record the biometric information and send them to a mobile device for further analyses and seizure detection.

To differentiate seizures from non-seizures, they developed a dynamic time warping algorithm, a template matching algorithm, to check the similarities of the templates to seizure like patterns. They recorded from 5 subjects who had non-partial seizures. in total, they obtained 21 seizures with 91% sensitivity and 84% specificity and having 50 false-alarms. From their work, we can measure the precision of 27% and F_1 *score* of 42%, with having only GTCS (the evaluation metric are discussed in detail later in this chapter, section 4.3.6).

Lockman et al.[78] conducted a study for detecting and recording tonic-clonic seizures using a wrist-worn watch equipped with an accelerometer sensor. The device was developed to detect seizures with rhythmic and rapid movements. They admitted epilepsy patients in an epilepsy clinic and obtained the gold standard seizure onset/offset from the clinic technicians. If the watch detects a seizure like activity, it sends a signal to a remote computer through Bluetooth. Among the 40 patients they recorded, 6 of them happened to have tonic-clonic seizures. From those 6 patients, they acquired 8 seizure, from which 7 were detected as seizures. They have had however 204 false-alarm cases

which only one of them was during the sleep. From their work, we can calculate the sensitivity as 87% and the precision as 2%. Accordingly, the F_1 *score* would be 0.04%.

Cogan et al. [18] developed a wearable seizure detection system to detect both motor and non-motor seizures. The system is designed to alarm the occurrence of the seizure and build an electronic diary. To this end, they collected 330 hours of data from 10 patients in an epilepsy center. In total, they attained 26 seizures (31 initially, discarding 5 due to data missing in seizure times).

They realized that the idea of wearable sensors is useful for a handful of people but not all. Thus, they split the patients into two sets based on their bio response. If the patient seizure can be tracked with typical seizure pattern (heart rate and electrodermal activity increase and decrease in pulse oximetry (S_pO_2)), the patient would wear a wearable sensor and can be sent back home for remote monitoring. If it is otherwise, the patient will get access to a maximum 3 channels device EEG for being tracked with EEG.

They conducted their study into 3 stages. In their first stage, they looked at the heart rate, arterial oxygenation, and electrodermal activities, which could be tracked by a wearable device and be detected the seizures. In the second stage , they used a pattern recognition technique to classify the collected 3 bio-signals. In the next step, they employ a 3 channel EEG to detect the seizures in order to compensate for the missed seizures.

They collected the heart rate and pulse oximetry data from a finger cuff device, *Nonin*, and the electrodermal and accelerometry data from a wrist-worn device, *Affectiva Q*.

Their first stage could find all 11 seizures of 7 patients. Stage 2, could recognize the entirety of 10 seizures of 6 patients. In the stage 3, it detected two third of all seizures.

Velez et al. [103] studied the problem of seizure counting in a clinical environment using a wrist-worn watch (SmartMonitor ©) with accelerometry sensor. The study investigates the accuracy of Generalized Tonic-Clonic Seizures (GTCS) vs. non-GTCS. The watch is connected through Bluetooth to a tablet and via tablet through Wi-Fi to an online analyzing system. They recorded accelerometry and audio signals as well as video-EEG data available from the epilepsy clinic of the study. They recorded from 27 patients with 62 seizures from which 13 (21%) were GTCS and 49 (79%) were non-GTCS. They split the accelerometry signal into different frequency bands and measured the accumulated power within each band and performed the detection process based on the extracted features.

Based on their claims, 12 out of 13 GTCS seizures were detected by their system (92.3%). In total, the sensitivity of their seizures detection system for all types of seizures would

be 19.3%. They have reported of having 81 false-alarm from which 42 (51.8%) were canceled from watch interface by the patients. There was no clear info of how long the whole study took in their report but patients were recorded from 1 to 9 days.

Vandecasteele et al. [102] investigated the problem of ECG based seizure detection across three different recordings methods, namely clinical ECG, portable ECG, and also photoplethysmography (PPG). PPG is discussed in chapter 5. They have recorded from 11 patients a total of 701 hours recording and having 47 seizures. They aimed at fronto-temporal lobe epilepsy and they evaluated their clinical and mobile ECG seizure detection algorithms with 57% and 70% sensitivity and 1.92 and 2.11 false-alarms per hour respectively. The mentioned false-alarm ratio is equal to 46.8 and 50.64 false-alarms per day.

4.3 Methods

4.3.1 Multivariate analysis

Complex phenomena are resulted from multiple causes and their effects can also be investigated across multiple variables. Multivariate analysis is composed of methodologies to study the effect of multiple variables in a phenomenon simultaneously. While in univariate analyses we measure one variable in different conditions and infer the role of that variable, in multivariate analyses the effects of multiple variable are considered and measured at the same time. While the beneficial points of univariate analyses is the simplicity of the analyses, a great advantage of multivariate analyses is the ability to describe complicated events across multiple variables, which is hardly possible to examine in univariate analyses. For example, the problem of global warming can not be traced along a single cause such as fossil fuel usage in cars. On the other hand, the effect of global warming is not only the acceleration in melting icebergs around the globe.

In seizure detection systems, in most of the studies [61, 93, 99], the role of a single variable or few variables measured of a single sensor unit is taken into account. In our study, to be able to perceive maximum amount of biological effects of seizures, we study changes across multiple variables and multiple sensors.

As noted in chapter 1, an effective technical pathway towards understanding complex phenomena is to employ *machine learning* techniques. Machine leaning methods can learn from data and give us a mathematical benchmark for distinguishing different phenomena from one another. Thus, in this phase too, we have adopted some machine learning methods to solve the problem of seizure detection.

Before moving to the details of this project, it is worthwhile to sketch its technical outline. Figure 4.2 describes the sequence of steps required for developing a machine learning solution for seizure detection.

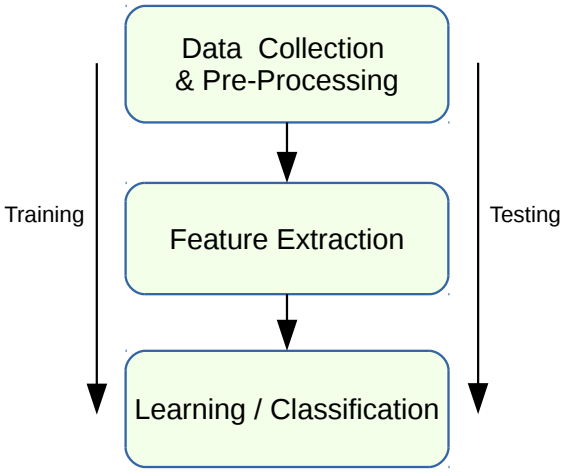


FIGURE 4.2: The general approach of using classification techniques to classify seizures from non-seizures. A model learned from train phase (pre-processing, feature-extraction, classification) will be used to classify unseen seizures from the testing phase.

In the following, the details of the outline shown in the diagram will be explained.

4.3.2 Subjects

Participants of our study are the patients who were admitted in the epilepsy clinic of Bonn for the sake of seizure monitoring. The patient monitoring includes video and EEG surveillance which can be recorded for later use (see Figure 4.3). The study was approved by the local medical ethics committee (Ethikkommission der Medizinischen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn, No. 140/13).

4.3.3 Sensors and data

Our seizure tracking study relied primarily on *Movisens* sensors [84]. Two types of Movisens sensors are available: one which integrates an ECG sensor and an acceleration sensor, and another which integrates an electrodermal sensor and an acceleration sensor (see Figure 4.4). The first type is attached to the chest of the patient/participant and

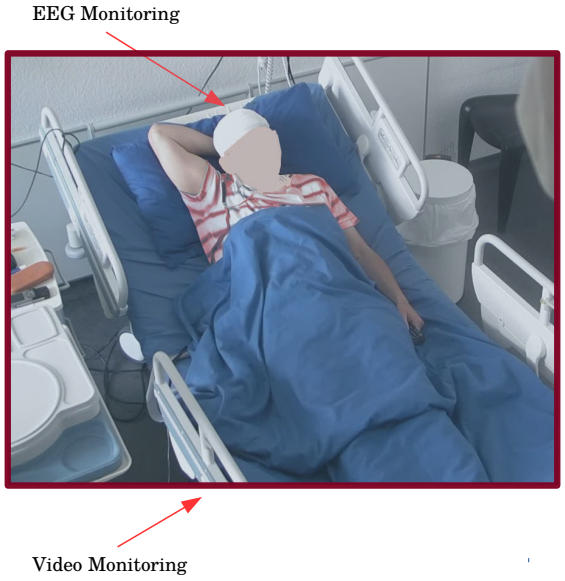


FIGURE 4.3: Video-EEG Monitoring. Patients are being recorded and monitored continuously during their residence in the clinic by video and EEG/ECG.

the second type is fastened to the wrists (one for each hand). Using these sensors, we were able to trace the seizure effects more effectively across three sensor units, with each unit measuring two modalities.

FIGURE 4.4: Movisens sensor units. Patients were equipped with three wearable devices provides by MoviSens GmbH (Karlsruhe, Germany) [84]. A device attached to the chest allowed recordings of ECG and acceleration of the body. Two more devices were attached to both wrists, allowing measurement of acceleration of arm movements and electrodermal skin response.

Each sensor unit could record for up to 24 hours without using a power supply. The sensor units could be charged prior to being worn and then programed and synchronized

with a PC in order to start recording at a certain time. After a day (or less) of recording, the sensors must be recharged and the data then transferred to the PC. The data of each sensor is presented as binary data. In addition, an XML file (meta-data) comes along with binary files of each sensor unit to augment the recording information with the recording time and the frequency of each sensor.

4.3.3.1 Synchronizing sensor data

The data recordings for the three different sensor units had to be synchronized due to the time drift happened in each sensor unit. Each sensor has its own clock oscillator and ticks not precisely in synchrony with the other units. Two methods for synchronizing could be considered. In the first method, at the beginning of the recording session, we placed all of the sensor units on top of each other and shook them so that the pattern of movement could be clearly reflected in the acceleration sensors. The start of a shaking pattern could be seen on each sensor, and the right offset for the synchronization was noted.

The second method which was to parse the XML file and check the timestamps of each sensor unit. To synchronize them, one can easily convert the absolute time of the recording of each sensor unit to the millisecond and then easily take the greater value among different sensors (the sensor which is lastly started) as the starting point of the recording. The second method is useful for synchronizing the beginning of the recording but not help resolving the clock drift problem.

Counting for time drift of sensors is crucial. Each sensor has an individual internal clock and therefore, at the end of the recording day, we can end up having 3 clocks pulsing differently, all different to the PC which they had been synchronized to. Fortunately, we noticed that in the worst case, the sensor clocks would be 10 seconds off of each other. This error is negligible due to the length of the seizures which are usually more than few minutes. Nevertheless, the signals could be also synchronized from the end of recording by resampling the signals. The end of recording however, needed to be determined again by putting all sensors on top of each other and shaking them.

4.3.3.2 Artifacts and ECG signal replacement

The ECG signal required several corrections. The ECG signals were recorded via two sensor contacts which were attached to the lower chest and the upper abdomen. ECG signals are vulnerable to movement artifacts. During strenuous activities or a seizure, the contacts can loosen or the signal can be contaminated by different muscle artifacts.

In order to address the artifact problems which occur during a seizure, I tried first to band-reject and filter out the noise. In case the signal cannot be reconstructed, the concurrent ECG recording from the monitoring room were used to replace the corrupt ECG signal of the seizure segments. When replacing the ECG of the Movisens sensors with the clinical one, two points should be kept in mind. First, the time of the seizure reported by the clinicians should be matched to the PC synchronization time for all of the sensors. Secondly, since the frequency of the two ECG systems does not match, a proper downsampling or upsampling process should be undertaken before replacement. Finally, the replaced signal should be compared against the original signal and inspected visually for its consistency for the neighboring segments of the replacement.

4.3.3.3 Visual inspection of ECG signal

For those noisy parts of the ECG signal which lacked clinical recording replacements, I visually inspected the entire signal. Sections of the signal in which no ECG relevant activity could be observed were excluded.

4.3.3.4 Annotating seizure time

Patients who were hospitalized in the Epileptology department of the university of Bonn medical center for seizure monitoring were under video and EEG/ECG surveillance. As soon as a seizure occurred, the seizure experts at the monitoring board examined the video and EEG/ECG recordings, identified the event, and responded accordingly. Since the event was recorded in its entirety, epilepsy experts could later comment on the exact onset/offset of the seizures. In order to synchronize the onset/offset of seizures in the clinical system with the sensor system, the rhythmic movement which we mentioned previously for signal synchronization, was recorded in detail on camera. There, it was possible to measure the time differences of the clinical system with the sensor system by checking the starting time of rhythmic movement in the accelerometer sensor versus the rhythmic movement action appearing in the video recording.

4.3.4 Feature extraction and multivariate analysis

In order to quantify alterations across different sensors, aside from the raw signal which is read out from sensors, we normally calculate more variables directly or indirectly from raw signals to unfold more aspects of an event which is difficult to observe solely by bare eyes. These variables are typically called *features* each of which can unveil a certain aspect of a phenomenon. For instance, while having the length and width of a

rectangle at hand, one can also calculate the area, diameter, and the circumference of that rectangle.

In statistical analyses and in pattern recognition terminology, features together shape a *Cartesian* multidimensional space called *feature space*. Each feature represents one dimension in the feature space and plays its smaller or larger role to identify an event or a phenomenon.

The data are measured from different sensor units: *Electrocardiogram* (ECG), *Acceleration*, and *Electrodermal* (EDA) sensors. In order to quantify the alterations for the different sensors, aside from the raw signal which is read out from sensors, we calculated more variables, directly and indirectly, from raw signals to explore more aspects of an event which is difficult to observe solely by raw signals. As mentioned earlier, these variables are called *features*, each of which can unveil a certain aspect of a phenomenon. Features together form a Cartesian multidimensional space called feature space. Each feature represents one dimension in the feature space and plays either a small or large role in identifying an event or a phenomenon.

Since our data is measured from different sensor units, the features should be calculated from those units separately.

4.3.4.1 Electrocardiogram (ECG)

ECG (or EKG) signals record heart activity via electrodes which are attached to the patient's chest. Electrical charges which are cast from the depolarization of heart muscles are transmitted to the skin and can be sensed and registered by ECG sensors, respectively. ECG signals are expressed as changes in voltage for a period of time. ECG signals can reveal information about the function and the structure of heart, as well as the pattern of the heart rate and rhythm.

ECG signals have a repetitive wave pattern (see figure 4.5). However, each wave form is comprised of smaller components, each of which is a part of a cardiac cycle. Primary components of cardiac events include the P-wave which reflects atrial depolarization, the QRS complex which renders the right and left ventricular depolarization, and T components which represent the ventricular repolarization (see figure 4.5).

The distance between consecutive R-peaks of QRS components is known as the RR-interval (or NN-interval). Studying the pattern of changes in RR-intervals is known as *heart rate variability* (HRV) and is a widely used tool for characterizing the physiological behavior of the heart.

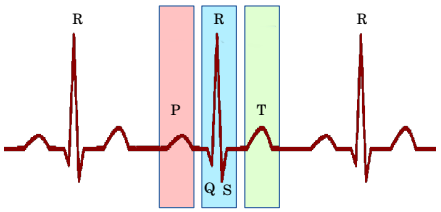


FIGURE 4.5: QRS complex of EEG. Each letter in the above signal, represents a piece of cardiac cycle. For seizure detection, the most important part of ECG signal is the distance between consecutive R-peaks. Most of relevant ECG features for seizure detection are extracted from the changes in the rhythm of R-to-R peaks intervals.

In order to detect seizures, the RR-intervals as an intermediate feature were used to further derive HRV-dependent features for qualifying the heart rate changes with respect to seizures. Detecting R-Peaks is an established method in the field [71, 89]. However, we realized that our ECG signal quality was dropping significantly throughout the day by having ECG contacts being loosely connected to the skin. Therefore, we needed a method to recover R-peaks from those with lower signal-to-noise-ratio recording segments. Consequently, I developed an algorithm to smartly detect R-peaks of ECG signal and measure RR-intervals:

- **Adaptive R-Peaks detection algorithm**

  To measure RR-Intervals, the first step is to detect R-peaks from ECG signal. Typically, ECG signal is contaminated with muscle and movement artifacts especially when an event like seizure is happening. Therefore, thresholding the peaks is not a proper way to reckon R-Peaks.

  Experimentally, I realized that by bandpass filtering the ECG signal from 10 Hz to 25 Hz, we will exclude most of the low and high frequency artifacts. Provided that the ECG electrodes were not completely detached from the subject during an event, by filtering the signal, we would normally end up with a cleaner signal which can systematically represent R-Peaks. Here, the algorithm, with an overlapping moving window over the signal, detects the R-peaks. The advantage of this algorithm is that it is not only resistant to noise but also detects R-peaks of different heights in cases in which, the ECG electrodes are loosely connected and therefore, their signal amplitude is significantly dropped (see algorithm 5).

  After bandpass filtering, by taking the absolute value of the filtered signal, R-Peaks will have always the greatest values along the signal. With an ordinary peak detector algorithm, it will be possible to detect R-Peaks accurately. The following criteria should also be taken into account for the peak-detector algorithm:

The minimum height threshold should be around 1 $\sim$ 1.7 times of the standard-deviation of the filtered signal and the minimum peak-to-peak distance should be one third of the ECG sampling frequency (e.g. $\sim$ 333 ms for 1000 Hz sampling-rate).

Algorithm 5: Detecting ECG R-peaks

Data: ECG Raw Signal
Result: Detecting ECG R-Peaks

```
while not at end of this signal do
    epoch ← take 5 seconds of ECG signal;
    a ← band-pass filter epoch [10-15 Hz, butterworth-2];
    b ← band-pass filter epoch [15-25 Hz, butterworth-4];
    c ← a .* b;
    d ← Carbox-filter(c);
    std ← standard-deviation(abs(d));
    find-peaks in d given:
        min peak height as std
        min inter-peaks distances as 1/3 of sampling-rate;
    if peaks with time overlapping then
        remove overlapping peaks;
    end
    move 2.5 seconds forward;
end
```

Figure 4.6 shows a part of ECG signal in blue and the absolute value of the filtered signal in green. The asterisks are the time points which were detected by the peak detector algorithm.

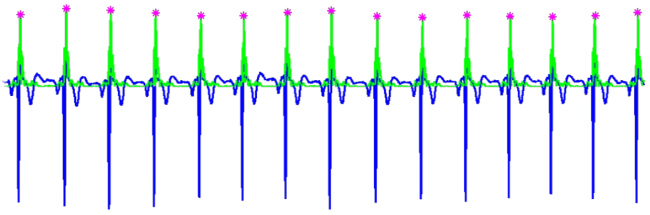


FIGURE 4.6: R-Peak detection algorithm. The algorithm is shown in 5. The asterisks represents the time points which were detected by the peak detector algorithm.

Having peaks detected, one can measure the distance between consecutive R-Peaks to get the RR-Intervals (RRI). To obtain the value in milliseconds, we multiply the distances by 1000 and divide the result by the sampling frequency.

$$RRI = abs(diff\ (peakLocations) \times (1000/Sampling\text{-}Frequency)$$

One should note that in cases which there are discontinuity in the detected peaks, we encounter having peak-to-peak intervals of longer than 1500 ms. Removing these values from RR-Intervals will fix the discontinuity problem by concatenating the isolated parts. On the other hand, values under 300 ms are not acceptable too, since heartbeat ratio more than 180 per minute is physiologically not much probable.

$$RRI = \{RRI_{>300ms}\ \&\ RRI_{<1500ms}\}$$

- **Feature: Heart-Beat:**

  Having R-Peaks detected, the heartbeat can be measured as the number of R-Peaks detected over the length of signal in seconds:

$$HeartBeat_{\forall sec} = \frac{NumOf(PeakLocations) \times SamplingRate(Hz)}{length(ECG))}$$

  The number of heartbeats per minute is also as following:

$$HeartBeat_{\forall min} = HeartBeat_{\forall sec} \times 60$$

- **Feature: Mean of RR-Intervals:**

  The average RR-Intervals indicates the first statistical moment of RR-intervals:

$$mean_{RRI} = mean(RRI)$$

- **Feature: STD of RR-Intervals (SDNN):**

  The standard deviation of RR-Intervals provides the second statistical moment of RR-intervals:

$$std_{RRI} = std(RRI)$$

- **Feature: Max of RR-Intervals:** The maximum value of RR-Intervals shows the longest RR-Interval:

$$max_{RRI} = max(RRI)$$

- **Feature: Min of RR-Intervals:** The minimum value of RR-Intervals represent the shortest RR-Interval:

$$min_{RRI} = min(RRI)$$

- **Feature: Root Mean Square of Successive Differences of RRIs (RMSSD):** RMSSD is a well-known measure in HRV analysis and functions principally as a high-pass filter in the time domain which can feature changes in autonomic vagal cardiac control and reveal respiratory sinus arrhythmia. It can also capture sympathetic activities in lower frequency variations [10]. RMSSD can be calculated as following:

$$RMSSD = sqrt(mean(diff(RRI)^2))$$

- **Feature: Shannon Entropy of RR-intervals:** Shannon entropy of RR-intervals reveals the uniformity of the data. Entropy will have a higher value if the fluctuation of the RR-interval values is high, and will have a lower value if the RR-interval values are almost uniform.

$$probablity_{RRI} = \frac{hist(RRI)}{length(RRI)}$$

$$entropy_{RRI} = -sum(probablity_{RRI} \times log2(probablity_{RRI}))$$

- **Power Spectral Density (PSD):**

  RR-interval features can be also featured in the frequency domain by measuring power spectral density (PSD). Measuring PSD of RR-intervals represents the frequency dependent analysis of RR-intervals. Certain ranges in RR-interval frequency spectrum convey particular physiological activities of the heart. It has been shown [5] that epilepsy patients have a different pattern of RR-Interval frequency especially during and after seizures compared to healthy individuals.

- **Feature: Very Low Frequency (VLF):**

  Very low frequency of RR-intervals, area under curve of PSD in frequencies from 0.003 Hz to 0.04 Hz. It can be measured as followings:

$$Samping\text{-}Freq = \frac{1000}{mean_{RRI}} \ , \ Num_{FFTs} = 2^{\ ceil(\ log_2(Fs \times 1000))}$$

$$Delta_{Freq} = \frac{2}{Num_{FFTs}}$$

$$PSD = P\text{-}Welch(\frac{Peak\text{-}Locations}{1000},\ 32,\ 24,\ Num_{FFTs},\ Samping\text{-}Freq)$$

$$f = FS/2 \times linspace(0,\ 1,\ \frac{Num_{FFTs}}{2} + 1)$$

$$PSD_{VLF} = \sum_{f=0.003_{Hz}}^{0.04_{Hz}} PSD \times Delta_{Freq}$$

One should note that to obtain accurate VLF features, the length of the recorded ECG signal must be more than 11 minutes, to be able to capture frequencies near 0.003 Hz (Nyquist-Shannon sampling theorem). The above mentioned PSD is estimated differently in [39] for measuring the sampling frequency. However, the results of both are almost identical.

- **Feature: Low Frequency (LF):** Sympathetic activity of the heart can be monitored by checking the RR-Intervals from 0.04 Hz to 0.15 Hz [93] . Similar to VLF, the area under the curve of frequency spectrum can be measured as:

$$PSD_{LF} = \sum_{f=0.04_{Hz}}^{0.15_{Hz}} PSD \times Delta_{Freq}$$

- **Feature: High Frequency (HF):** Similar to LF, parasympathetic activity [93] can be also represented by frequency of RRI PSD from 0.15 Hz to 0.4 Hz.

$$PSD_{VLF} = \sum_{f=0.15_{Hz}}^{0.4_{Hz}} PSD \times Delta_{Freq}$$

- **Feature: Total Power:** Total frequency power is the total area under the curve of PSD of RRI frequency.

$$PSD_{VLF} = \sum_{f=0.003_{Hz}}^{0.4_{Hz}} PSD \times Delta_{Freq}$$

- **Feature: Cardiac Vagal Index (CVI):**

  CVI is a sensitive measure to cardiac vagal activities [99]. RRI data point can be plotted as a form of *Lorenz* plot to be able to extract the values which are needed for measuring CVI. In Lorenz plot, every RR-Interval value must be plotted against its proceeding RR-interval value (see figure 4.7).

  Data points of Lorenz plot are normally distributed like an oval shape. By omitting outliers in the plot, two values can be extracted from Lorenz plot: L and T, the length and the width of the oval. CVI is defined a below:

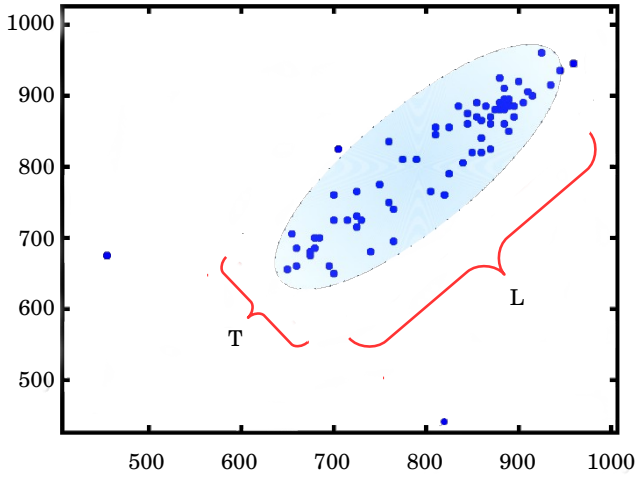


FIGURE 4.7: Lorenz-Plot. Every RRI value is plotted against its proceeding RRI value in time. The result will be an oval shape could of data. By discarding the extreme values, the diameters of the oval can be used to measure other HRV features.

$$CVI = Log_{10} L \times T$$

- **Feature: Cardiac Sympathetic Index (CSI):** This feature also inherits some parts of Lorenz plot (L and T) to show increase in pre-ictal and early ictal phases of a seizure[61, 99]. CSI is calculated as:

$$CSI = \frac{L}{T}$$

;

- **Feature: NN50:** NN50 is the number of successive RR-Intervals which differ more than 50ms:

$$diffRRI = abs(diff(RRI))$$

$$NN50 = Num.Of(diffRRI > 50)$$

- **Feature: pNN50:** pNN50 is the number of NN50 over the total number of RR-Interval changes:

$$diffRRI = abs(diff(RRI))$$

$$pNN50 = \frac{Num.Of(diffRRI > 50)}{Num.Of(diffRRI)}$$

- **Feature: ECG-Arousal:** ECG-Arousal measures the ratio of heartbeat increase in two consecutive minutes. Our current and former recordings showed that during most seizures (over 85%), a heartbeat increase of near 1.2 fold can be observed. ECG-Arousal can be easily measured as following:

$$ECG\text{-}Arousal = \frac{heart\ rate\ of\ first\ last\ minute}{heart\ rate\ of\ second\ last\ minute}$$

- **Statistical measures of raw ECG signal**

  Although it may look clinically pointless to check the statistical changes of raw ECG signal, these features could potentially express some of the seizure dependent changes of ECG signal. In the following, more features from ECG signal are extracted and included to our analysis. In ECG based seizure detection, the statistical features have not been normally used. However, I tested them and they improve the accuracy of seizure detection since they can feature the contaminated muscle contractions and movements noises during a seizure. Nonetheless, they are not employed in final classification model to be compatible with HRV-dependent measures.

- **Feature: Mean of signal:** The average signal amplitude of the ECG signal is also considered as a feature:

$$Mean\text{-}ECG = mean(ECG\text{-}Signal)$$

- **Feature: STD of signal:** The standard deviation of the ECG signal characterizes the deviations from the average amplitude of the signal (second statistical moment):

$$STD\text{-}ECG = std(ECG\text{-}Signal)$$

- **Feature: Skewness:** Skewness is also added to the set of features which captures the third statistical moment of the raw ECG signal.

$$Skewness\text{-}ECG = skewness(ECG\text{-}Signal)$$

- **Feature: Shannon Entropy:** Shannon entropy displays the randomness level of the signal. The more random the signal, the higher the Shannon entropy value.

$$probablity_{ECG} = \frac{hist(ECG)}{length(ECG)}$$

$$Entropy\text{-}ECG = -sum(probablity_{ECG} \times log2(probablity_{ECG}))$$

- **Feature: Frequency components of FFT:** Transforming the raw ECG signal to the frequency domain using fast Fourier transformation (FFT) results in several numeric components, each of which represents the magnitude of a certain frequency band in the original ECG signal. The power of the frequency components themselves can also be used as a feature for picturing the frequency specific features of the ECG signal. It should be noted that those frequency components of a certain length of signal should satisfy the Nyquist principle as mentioned before. In this study, the frequency components of 0.5 Hz and below have been tested.

$$Num_{FFTs} = \textit{ECG-Sampling-Freq} \times 20$$

$$\textit{ECG-Freq-Components} = fft(ECG, Num_{FFTs})$$

$$\textit{ECG-Freq-Power} = 2 \times abs(\textit{ECG-Freq-Components}(1 : \frac{Num_{FFTs}}{2} + 1))$$

$$\textit{ECG-Freq-Power-Cut} = \textit{ECG-Freq-Power}([0 : 0.5]Hz)$$

4.3.4.2 Accelerometry

Some types of seizure events (and tonic-clonic seizures, in particular) are associated with strong body movements and vibrations. Detecting and registering the movement pattern during a seizure helps improve the accuracy of seizure detection systems. In our study, three acceleration sensor units were embedded in three comfortable-to-wear sensor units. One was attached to the patient's chest with a belt to record body movements and the other two units were affixed to the patient's wrists in order to record hand movements. Same types of features for all accelerometry sensors were extracted. Therefore, what is described in the following for feature extraction of accelerometer, applies to all three sensors:

- **Feature: Average Displacement:** Since an acceleration sensor measures the acceleration along 3 axes, the displacement as the average norms of second level integral of all 3 acceleration axes along the timeline can be measured:

$$\overline{disp} = mean(\sqrt{(\sum_t \sum_t Acc_X)^2 + (\sum_t \sum_t Acc_Y)^2 + (\sum_t \sum_t Acc_Z)^2})$$

- **Feature: Standard Deviation of Displacement:** This feature indicates the displacement and can be calculated as the standard deviation of norms of second level integral of all 3 acceleration axes along the timeline:

$$\sigma_{disp} = std(\sqrt{(\sum_t \sum_t Acc_X)^2 + (\sum_t \sum_t Acc_Y)^2 + (\sum_t \sum_t Acc_Z)^2})$$

- **Feature: Average Velocity:** The velocity of the movement can be measured as the average norms of first level integral of all three acceleration axes along the timeline:

$$\overline{Velocity} = mean(\sqrt{(\sum_t Acc_X)^2 + (\sum_t Acc_Y)^2 + (\sum_t Acc_Z)^2})$$

- **Feature: Standard Deviation of Velocity:** The standard deviation of movement velocity can be measured as the standard deviation of the norms of first level integral of all 3 acceleration axes along the timeline:

$$\sigma_{Velocity} = std(\sqrt{(\sum_t Acc_X)^2 + (\sum_t Acc_Y)^2 + (\sum_t Acc_Z)^2})$$

- **Feature: Average Acceleration:** The acceleration of the movement can be measured as the average norms for all three acceleration axes along the time axis:

$$\overline{Acc} = mean(\sqrt{Acc_X^2 + Acc_Y^2 + Acc_Z^2})$$

- **Feature: Standard Deviation of Acceleration:** The standard deviation of the acceleration of the movement can be measured as the standard deviation of the norms for all three acceleration axes along the time axis:

$$\sigma Acc = std(\sqrt{Acc_X^2 + Acc_Y^2 + Acc_Z^2})$$

4.3.4.3 Electrodermal (EDA)

Electrodermal activity or skin conductance is the property of the human body that causes continuous variation in the electrical characteristics of the skin and is the result of changes in sweat glands of the skin. Sweat glands are more active when the activity of the sympathetic nervous system (SNS) is aroused. Consequently, the activity of sweat glands increases and this in turn causes increased skin conductivity. Changes in skin conductance reflect a degree of subliminal physiological or psychological activities.

To measure the skin conductance response, typically two electrodes will be attached to the palm of a person within $1 \sim 3$ centimeters distance from each other. As soon as a physiological or psychological event causes an arousal in sympathetic neural systems, the activity of sweat glands will alter the skin conductance. Changes in conductivity can be sensed by measuring the conductance level between two electrodes. Most of seizures change the electrodermal activities significantly [18, 92]. We included skin conductance sensors in our study to improve the versatility of seizure sensing.

Our analysis included the following EDA features that support the seizure detection systems with features which can potentially characterize arousal events of the sympathetic nervous system:

- **Feature: Mean of Electrodermal Signal:** This feature reflects the average of skin conductance signal amplitude, directly recorded from sensors.

$$\overline{EDA} = mean(EDA)$$

- **Feature: STD of Electrodermal Signal:** The standard deviation of the EDA conveys the changes of skin conductance signal amplitude.

$$\sigma_{EDA} = std(EDA)$$

- **Feature: Mean of First Derivative of Electrodermal Signal:** This feature reflects the average changes of skin conductance signal amplitude by measuring the first derivative of the signal.

$$\overline{EDA} = mean(EDA')$$

- **Feature: Standard Deviation of First Derivative of Electrodermal Signal:** This feature reflects the standard deviation of changes of skin conductance signal amplitude.

$$\sigma_{EDA} = std(EDA')$$

4.3.4.4 Characterizing the differences of features pre-ictal vs. post-ictal

In order to derive the physiological changes during a seizure, the features of pre-ictal phase vs. the features of post-ictal phase were compared. While each feature can individually convey seizure related information, the differences can prominently expose

the changes. In the following, the third level features, features to characterize differences between pre and post ictal are presented:

- **Feature: Change of heartbeat:**

$$\textit{Diff-}HR_{\textit{Pre-Post}} = HR_{Post} - HR_{Pre}$$

- **Feature: Change of average RR-Interval:**

$$\textit{Diff-Mean-}RRI_{\textit{Pre-Post}} = \overline{RRI_{Post}} - \overline{RRI_{Pre}}$$

- **Feature: Change of standard deviation of RR-Intervals:**

$$\textit{Diff-Std-}RRI_{\textit{Pre-Post}} = \sigma RRI_{Post} - \sigma RRI_{Pre}$$

- **Feature: Change of minimum of RR-Intervals:**

$$\textit{Diff-Min-}RRI_{\textit{Pre-Post}} = Min(RRI_{Post}) - Min(RRI_{Pre})$$

- **Feature: Change of maximum of RR-Intervals:**

$$\textit{Diff-Max-}RRI_{\textit{Pre-Post}} = Max(RRI_{Post}) - Max(RRI_{Pre})$$

- **Feature: Change of Root Mean Square of Successive Differences of RR-Intervals (RMSSD):**

$$\textit{Diff-RMSSD-}RRI_{\textit{Pre-Post}} = RMSSD(RRI_{Post}) - RMSSD(RRI_{Pre})$$

- **Feature: Change of entropy of RR-Intervals:**

$$\textit{Diff-Entropy-}RRI_{\textit{Pre-Post}} = \textit{Entropy}(RRI_{Post}) - \textit{Entropy}(RRI_{Pre})$$

- **Feature: Change of Cardiac Vagal Index (CVI):**

$$\textit{Diff-}CVI_{\textit{Pre-Post}} = CVI_{Post} - CVI_{Pre}$$

- **Feature: Change of Cardiac Sympathetic Index (CSI):**

$$\textit{Diff-}CSI_{\textit{Pre-Post}} = CSI_{Post} - CSI_{Pre}$$

- **Feature: Change of Power Spectrum Density (PSD) in very low frequency bands of RR-Intervals:**

$$\textit{Diff-PSD-VLF-RRI}_{\textit{Pre-Post}} = \textit{PSD-VLF-RRI}_{\textit{Post}} - \textit{PSD-VLF-RRI}_{\textit{Pre}}$$

- **Feature: Change of Power Spectrum Density (PSD) in low frequency bands of RR-Intervals:**

$$\textit{Diff-PSD-LF-RRI}_{\textit{Pre-Post}} = \textit{PSD-LF-RRI}_{\textit{Post}} - \textit{PSD-LF-RRI}_{\textit{Pre}}$$

- **Feature: Change of Power Spectrum Density (PSD) in high frequency bands of RR-Intervals:**

$$\textit{Diff-PSD-HF-RRI}_{\textit{Pre-Post}} = \textit{PSD-HF-RRI}_{\textit{Post}} - \textit{PSD-HF-RRI}_{\textit{Pre}}$$

- **Feature: Change of Power Spectrum Density (PSD) in all frequency bands of RR-Intervals:**

$$\textit{Diff-PSD-AF-RRI}_{\textit{Pre-Post}} = \textit{PSD-AF-RRI}_{\textit{Post}} - \textit{PSD-AF-RRI}_{\textit{Pre}}$$

- **Feature: Change in average displacement:**

$$\textit{Diff-Mean-Disp}_{\textit{Pre-Post}} = \overline{\textit{Disp}}_{\textit{Post}} - \overline{\textit{Disp}}_{\textit{Pre}}$$

- **Feature: Change in standard deviation of displacement:**

$$\textit{Diff-Std-Disp}_{\textit{Pre-Post}} = \sigma \textit{Disp}_{\textit{Post}} - \sigma \textit{Disp}_{\textit{Pre}}$$

- **Feature: Change in average velocity:**

$$\textit{Diff-Mean-Velocity}_{\textit{Pre-Post}} = \overline{\textit{Velocity}}_{\textit{Post}} - \overline{\textit{Velocity}}_{\textit{Pre}}$$

- **Feature: Change in standard deviation of velocity:**

$$\textit{Diff-Std-Velocity}_{\textit{Pre-Post}} = \sigma \textit{Velocity}_{\textit{Post}} - \sigma \textit{Velocity}_{\textit{Pre}}$$

- **Feature: Change in average acceleration:**

$$\textit{Diff-Mean-Acc}_{\textit{Pre-Post}} = \overline{\textit{Acc}}_{\textit{Post}} - \overline{\textit{Acc}}_{\textit{Pre}}$$

- **Feature: Change in standard deviation of acceleration:**

$$\textit{Diff-Std-Acc}_{\textit{Pre-Post}} = \sigma Acc_{Post} - \sigma Acc_{Pre}$$

- **Feature: Change in average EDA:**

$$\textit{Diff-Mean-EDA}_{\textit{Pre-Post}} = \overline{EDA}_{Post} - \overline{EDA}_{Pre}$$

- **Feature: Change in standard deviation of EDA:**

$$\textit{Diff-Std-EDA}_{\textit{Pre-Post}} = \sigma EDA_{Post} - \sigma EDA_{Pre}$$

- **Feature: Change in average first derivative of EDA:**

$$\textit{Diff-Mean-EDA}'_{\textit{Pre-Post}} = \overline{EDA'}_{Post} - \overline{EDA'}_{Pre}$$

- **Feature: Change in standard deviation of first derivative of EDA:**

$$\textit{Diff-Std-EDA}'_{\textit{Pre-Post}} = \sigma EDA'_{Post} - \sigma EDA'_{Pre}$$

- **Third level "difference" features extracted from raw ECG data** As mentioned earlier in this sub-section, the features of raw ECG signal contains information about seizure events. Again, even though the raw features were not used in the final evaluation, these features were measured. Similar to HRV features, the difference between pre-ictal and post-ictal features could be measured also for raw ECG data.

- **Feature: Change in PSD of raw ECG signal:** We considered three frequency bands as well as the total power changes using FFT:

$$\textit{Diff-Power-Cut-Freq-ECG}'_{\textit{Pre-Post}} = \textit{Power-Cut-Freq}_{Post} - \textit{Power-Cut-Freq}_{Pre}$$

4.3.4.5 Windowing over the data

A commonly accepted method for analyzing a time series is moving a sliding window along the timeline while computing the features for each window separately. This window slides stepwise. The number of steps should be so that each new window would overlap a section of the previous window. Having overlapping windows helps avoid massive changes from one window to the next, especially if the windows are long. Due to problems surrounding seizure detection, I separately characterize the signs and symptoms of a

seizure, both before and after the seizure occurred. Hence, as a time-point of interest (see figure 4.8), we considered three windows for feature computation:

1. Pre-ictal window: 5 minutes (300 sec) window assessing long term changes before the time-point of interest.
2. Post-ictal window: 5 minutes (300 sec) window assessing long term changes after the time-point of interest.
3. Ictal window: 10 sec window assessing momentary changes right at the exact time-point of interest.

The length of the windows has been determined according to the physiological effect of the seizures reported [57, 97] and our experimental conclusions. I chose the forwarding step to be 10 seconds equal to the size of ictal window and also similar to [92]. That is, the resolution of the seizure detection would be 10 seconds accordingly.

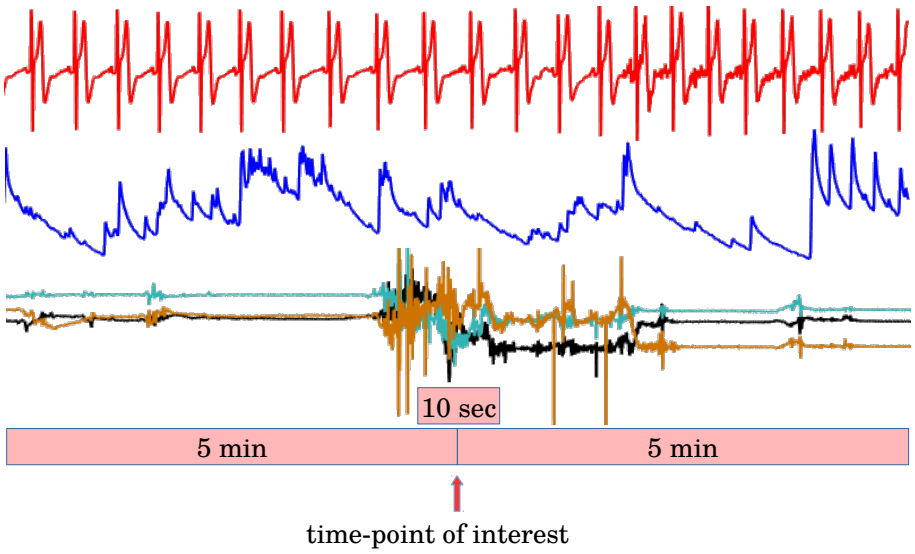


FIGURE 4.8: The Windowing concept and concurrent recording from different modalities. The upper time series is the ECG data. The middle signal is the electrodermal data. The bottom one is the accelerometry data from three axes. For any time-point of interest, a 10 second window, a 5 minutes window before the time-point and 5 minutes window after the time point is considered for feature extraction.

- **Concatenating windows and features** As mentioned above, for each time point of interest, three windows were examined and for each window, verities of features were computed. Features of each window together, shape a feature vector, and

feature vector of all three windows can be concatenated to form a longer feature vector. Linking the pre-ictal vs. post-ictal features, makes the feature vector even longer. This long feature vector represents the physiological phenomenon of the signal for any given time-point of interest as a quantified multidimensional measurement.

- **Multidimensional representation of feature vectors** Feature vectors that are composed of feature elements can be viewed as multidimensional points in Cartesian space. Each element of a feature vector can represent one value along an axis in their respective space. Having features aligned and represented in a multidimensional feature space, mitigates dealing with physical phenomena by utilizing algebraic operations and algorithmic methods.

4.3.4.6 Problem of unbalanced number of positive vs. negative examples

In activity detection scenarios, due to having too many negative examples (non-seizure feature windows) compared to positive cases (seizure windows), instead of classification accuracy, the performance of the system has to be measured based on sensitivity, specificity, and precision metrics. Nonetheless, since the number of negative examples outnumbers the positives drastically, a classification algorithm of choice would still have hard time to distinguishing negatives from positives. To ease this difficulty, Three approaches can be proposed.

1. **Event-based window selection**

   In this method, the goal is to find a feature/threshold which can filter out the majority of the non-seizure windows but still keeping absolute majority of seizure windows. A feature window is considered to be an event if the window can pass the threshold. To acquire event-based seizure detection, I searched over a separate seizure recording dataset and found a threshold for the event filtering. In accordance with the search, in absolute majority of the seizure cases, the heart rate (HR) increases by 1.2 folds in two consecutive minutes. Therefore, for event filtering, epochs in which the HR during a period of two minutes increased by at least a factor of 1.2, were exclusively considered. The feature "ECG-Arousal" from our feature vector can be used to reflect such event. By filtering the feature windows based on the event thresholding, the size of negative example was reduced significantly. The exact amount of reduction is dependent on the subjects' physical and physiological activities (more about it later in 4.4 section).

2. **Negative examples down-sampling**

The second way to reduce the negative examples for the classification task is to draw a meaningful amount of negative examples randomly from the negative pool and carry out the classification task and then, repeat the procedure for several time until a large proportion of the dataset is chosen for training and test.

3. **Positive examples up-sampling** The other way to alleviate the problem of unbalanced number of positive and negative examples is to increase the number of positive examples by duplicating the positive examples up and until the number of positive and negative examples will be the same. *Data imputation* and *data augmentation* are examples of up-sampling algorithms in literature.

4.3.5 Pattern classification

Feature extraction is normally a prerequisite for machine learning tasks. Feature extraction, either explicitly as a discrete step, or implicitly as with some modern machine learning techniques [68, 74], helps magnify more aspects of the data and reveals the latent pattern of different classes in complex scenarios. In order to properly categorize the patterns to their respective classes, a classifier is used. A classifier is the name assigned to different types of algorithmic, statistical, relational, or clustering algorithms or approaches in order to perform human independent prediction and classification of physical phenomena. In other words, it enables machines to learn from events and data and aids in the prediction or categorization of newly observed events and data.

In the realm of machine learning, there are two common ways of learning the procedure of pattern classification tasks: supervised learning and unsupervised learning. In supervised learning, the classifier knows the data as well as the data class. Based on the labeled data in supervised learning, the classifier derives a reference criterion for categorizing the data to their respective classes. The resulting criteria is called the *model* and the process of learning is known as *training*. To evaluate how a classifier would perform using unknown data, the classifier is provided with data which was not available during training. This procedure is called *testing*. One way of evaluating the performance of the classifier is to measure how many times the test data were correctly classified to their true labels and divide it by the total number of test cases.

In unsupervised learning however, the classifier has no access to labeled data and the goal is to categorize the unlabeled data based solely on their innate similarity. For instance, categorizing people in social media can be done by measuring their behavioral similarities and contrasts without knowing about their identity. Therefore, in unsupervised learning, training and testing phases are unified to one step. Unsupervised learning in literature is reffed to *clustering* too.

Supervised learning and unsupervised learning can also be combined in certain cases in which there are some unlabeled data used in the training phase alongside with labeled data. The *semi-supervised* classifier can outperform the supervised and unsupervised classifiers given that the data is distributed as clusters in their feature space [16]. In that case, the unsupervised learner can initially help the supervised learner to determine the class boundaries more accurately.

Apart from supervised and unsupervised learning, there is also another classification style in the machine learning called *reinforcement learning* in which the learner/classifier gradually learns how to react to different inputs by receiving feedback. This type of learning resembles the way people train a dog by giving it positive or negative rewards. Reinforcement learning is a used mostly in robotics.

Seizure detection problems can be investigated using supervised learning since our goal is to obtain an accurate function mapping of our input features to a binary output, and to determine whether those features represent a seizure or non-seizure case. Mathematically speaking, we are searching for a function h that maps our feature vector X to output Y:

$$h : X \rightarrow Y$$

Vector X is composed of n features $x_1 ... \ x_n$ and the output variable $y \in Y$ can be a binary case of seizure or non-seizure. Alternatively y can also be a probability value expressing the likelihood of an event being a seizure or non-seizure. The function h can be defined so that it initializes itself with some [random] parameters for mapping from X to Y, and measures the amount of misclassification (loss error), and then searches repeatedly for the best combination of mapping parameters which minimizes the loss error at most. The parameters of h is shown as θ in our modeling.

The solution to our problem can be formulated in the following manner:

$$P(y|x, \theta)$$

There are however two ways to measure it. One is *discriminative* approaches that $p(y|x)$ is measured directly while in the other, the *generative* approach, $p(x|y)$ (inverse probability) and $p(y|\theta)$ (prior probability) are combined by applying Bayes' rule [7, 12, 30].

Since in our problem, there is no reliable estimation of the real-world ratio of seizure cases to none-seizure cases, it is not possible to measure $p(x|y)$ accurately. Consequently,

I prefer to model the learning problem primarily using discriminative approach. Though, the generative approach is used partially in parallel with the discriminative classifiers. Classifiers such as SVM [19], Random-Forest [15], or KD-Tree (KDT) [9] are suitable tools to measure $p(y|x)$. There are varieties of classifiers in the real world which can be employed, each of which has its own pros and cons.

In our study, I found the random-forest (RF) classifier to outperform the others tested such as SVM, Logistic-Regression, etc. The RF classifier is a collection of decision tree classifiers, each of which has a limited number of features that are randomly selected and the decision boundary is decided by each tree individually. The final decision boundary of RF is a product of all decision boundaries of decision trees. Compared to decision trees, RFs insure more generalization and less overfitting. One of the reasons RFs can satisfactory classify the classes is due to our data distribution. Our data is unequally balanced and does not include the same number of instances for the different categories. RFs perform well with such data. Features of the class with smaller number are accumulated in some cells in their feature space and tree classifiers are suitable tools for finding class boundaries in such cases.

Besides, RF is a convenient choice when the collected features are not distributed seamlessly in the feature space. In such cases, features of instances of a particular class are sparsely distributed and hence, tree classifiers have significant edge over the conventional uniform distribution classifiers to model the data. In a work we published before, this fact has been investigated on a different dataset [60].

With RF, we can also rank the features based on their level of importance for classification. This will help untangle the associated roles of parameters used for classification which is of significant importance to medical societies. Moreover, RF classifiers can internally account for missing data. If some portions of the signal are missing or corrupted, their respective feature in the feature vector has zero/nan values. However, RFs can learn from the available data to compensate it.

4.3.5.1 Probability thresholding

In detection systems, it is often important to detect positive examples, even if this means increasing the prospect of a false positive alarm. A classifier typically estimates the probability of data belonging to different classes. A class with higher probability is selected as a classification result. For this study, a lower threshold (20%) was utilized instead of the common 50% probability threshold. With this adjustment, the classifier now identifies cases with a classification probability of 20% or more as seizures. This, in

turn, causes cases with few seizure similarities to be included by increasing the chances of incorrectly identifying seizures.

4.3.5.2 Early fusion vs. late fusion

In cases that the source of data is from different modalities, there will be two perspectives to conduct a machine learning task: *early fusion* and *late fusion*. In early fusion, features from different modalities (ECG, ACC, EDA in our case) will be extracted and pooled together and a single classifier will be trained on them. In late fusion approach however, for each modality , a separate classifier will be trained and the classification results of all classifier will be pooled together to decide on a testing example. I can not address a definitive rule to say which one performs better but in my view people tend to prefer the late fusion. Our classification results could be also reported based on early or late fusion. However, I did not observe any significant difference in using either of them in our data.

4.3.6 Evaluation

The collected database of seizures was then fed into a classifier. In a common classification task, a segment of data which is typically a collection of labeled feature vectors is used initially by the classifier to learn and train a model. The resulted model can then be used as a benchmark for predicting/classifying the data which were not previously presented to the classifier. Accordingly, a portion of the data which included feature vectors of both seizure and non-seizure cases was fed to the classifier. Next, the performance of the classification task by testing the rest of data which were not presented to the classifier against the already learned model was evaluated.

The subsequent question at this phase would be which metric should be used to evaluate the classification performance. Knowing that we have nonequivalent amount of seizure and non-seizure cases, we can not evaluate the performance of the system based on classification accuracy because it can potentially bring very high accuracy which is influenced by the class with absolute majority. In such cases, a conventional way to assess the performance is to consider some measurements such as *sensitivity* and *specificity* and *precision* in parallel.

Before defining sensitivity, specificity, and precision, the following terms should be defined:

- Positive (P): Number of seizure events

		Ground Truth	
		P	N
Predicted	P	TP	FP
	N	FN	TN

TABLE 4.1: Detection table. This table shows different conditions of binary classification results.

- Negative (N): Number of non-seizure events
- True-Positive (TP): Number of positive examples detected truly as positive
- False-Positive (FP): Number of positive examples detected falsely as positive (type one error)
- True-Negative (TN): Number of negative examples detected truly as negative
- False-Negative (FN): Number of negative examples detected falsely as negative (type two error)

The table below helps to better comprehend the above mentioned terms:

sensitivity (or *recall*) measures how accurate a system is tuned to detect positive cases (seizures). In other words, it measures how many of seizure cases were detected truly as seizure. Sensitivity can be measured as following:

$$Sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Specificity shows how accurate a system is to detect negative cases (non-seizure). Specificity measures how many of non-seizure cases were truly detected as non-seizure. In other words, a highly specific system, does not detect too many of negative examples falsely as positive (false-alarm). Specificity can be measured as following:

$$Specificity = \frac{TN}{N} = \frac{TN}{TN + FP}$$

Precision (or *positive predictive value*) calculates the probability of the cases which were detected as seizures, being really seizures. Precision is measured as below:

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

A perfect detection system aims at maximizing all of the above mentioned metrics. There are measurements to combine two metrics among sensitivity, specificity, and precision in order to provide a single quantity, and F_1 *score* [11] is one of them. F_1 *score* is measured as followed:

$$F_1\ score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$$

F_1 *score* is in fact a harmonic mean of sensitivity and precision. In the context of seizure detection, sensitivity and precision are more important than the specificity since the great number of negative examples helps naturally in getting higher amounts of specificity.

Additionally, the *Area Under the Curve* (AUC) of *Receiver Operating Characteristic curves* (ROC) is another method to summarize the performance of a binary classifier in the from of a single measurement. To plot an ROC curve, two elements are needed to be measured: *true positive rate* (TPR) and *false positive rate* (FPR). True positive rate is the number of all detected positive cases over the total number of positive cases. Hence, TPR is the same as sensitivity. False positive rate is the number of negative cases falsely detected as positive over the total number of negative cases. Therefore, FPR is equal to 1-specificity. In an ROC curve, by changing the classification threshold to a ranges of possible values, we will be able to obtain a curve resulted from plotting TPR against FPR. Figure 4.9 illustrates an ROC curve.

The area under ROC curve is an indication of the system performance and it ranges from 0.5 (random guess) to 1 (perfect classification). The larger AUC of ROC indicates a reliable balance between sensitivity and specificity.

A common practice for evaluating a classifier is to use an *n-fold cross-validation scheme*, in which the training and testing procedures are repeated for n times and then, the final result is the averaged result of all folds. Using n-fold cross validation helps average out possible biases in the classification results. For the upcoming results, I used a 5-fold cross validation schema. That is, 80% of the feature vectors were randomly drawn for train a model and then the remaining 20% feature vectors used for testing. Repeating this for 5 times and then averaging the results gave the final classification evaluation. Alternatively, I used a *leave-one-out cross-validation* schema. This is a special case of n-fold cross-validation wherein one sample is for testing and the remaining are for training. This is repeated until all samples in the dataset have been used once in testing.

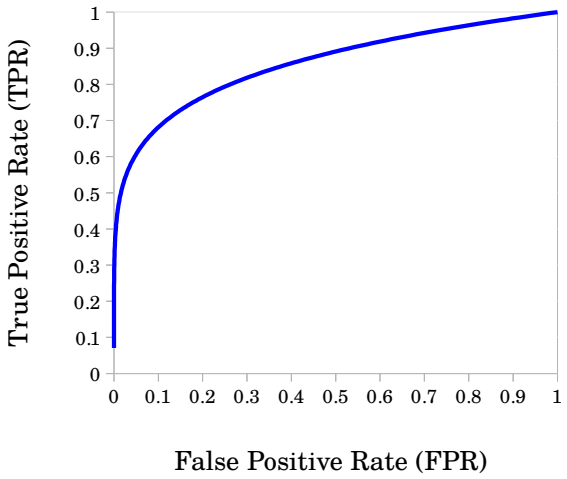


FIGURE 4.9: The area under the curve of ROC is used to measure the performance of a detection system. TPR and FPR are the axes of the plot. In an ideal case the area AUC of ROC is close to 1. AUC of ROC being near 0.5 indicates random level detection results.

Afterwards, the results are averaged. Because we have a limited number of seizure cases (compared to non-seizure cases), I performed the splitting portion of cross-validation based solely on the number of positive examples.

4.4 Results

In this thesis, the data of 42 first patients who were participated in our study were examined. The data from the first 6 patients was excluded because they did not have accelerometry sensors attached to one hand and the ECG signal frequency was inconsistent with the other patients. Patient No.15 was also excluded because the data was lost. Therefore, we continued the analysis with the remaining 35 patients.

For the remaining 35 patients, the total recording amounted to 52 days(days could be less than 24 hours) with each patient having been admitted to the clinic from $1 \sim 6$ days. While some patients did not have any seizures during their stay, others had one or more seizures. In total, we compiled 33 seizure cases (4 simple partial, 24 complex partial and 5 generalized tonic-clonic seizures, see image below) from 24 patients (mostly with temporal lobe epilepsy; age 39 ± 14 years, and having an epilepsy duration of 18 ± 13.5 years).

Since the number of seizures per patient was not sufficiently high to aim at seizure prediction per patient, I sought for a model by which it is possible to generalize over various types of seizures of different patients. Hence, I made a dataset which pools over all seizure and non-seizure cases of all patients. At this stage, each feature vector

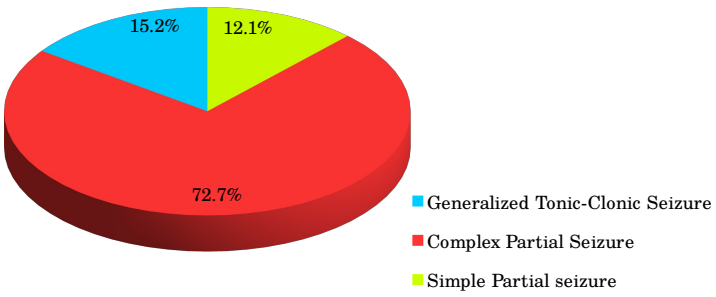


FIGURE 4.10: Proportion of 33 recorded seizures.

was labeled as seizure or as a non-seizure case based on the ground truth provided by epilepsy experts in the epilepsy clinic Bonn.

The primary goal of our system was to detect and register seizures and then count them. For technical consistency, I performed a windowing approach to extract the features. However, the final evaluation had to be based on physiological events and not the windows. To tackle this problem, a maximum of 6 consecutive windows ($6x10Seconds = 1minute$) was considered as an event if they could survive the early filtering. In this setting, for an event, a class with a majority of votes from the classified windows will be declared as seizure or non-seizure.

For later analyses, it is important to report the distribution of feature windows round the clock to be able to evaluate the results based on daily time precisely. The table 4.2 shows the distribution of feature windows along time.

In the following, I evaluated and reported the performance of the system based on various settings, including changes in the classifier of choice and its parameters, the combination sensor data, and event thresholding, etc. Different metrics like sensitivity, specificity and precision, area under the curve of ROC as well as F_1 score has been used.

Earlier in this chapter, I referred to two machine learning arrangements towards solving the current seizure detection problem, with event-filtering and without event-filtering. As a quick recap, in event-filtering, we train the classifier merely on limited number of windows. Those windows must possess certain characteristic to be included in the training process. If a window has that distinguishing characteristic, it will be considered to represent an event. In contrast, in a regular training process, (theoretically) all windows will be taking part in the training process.

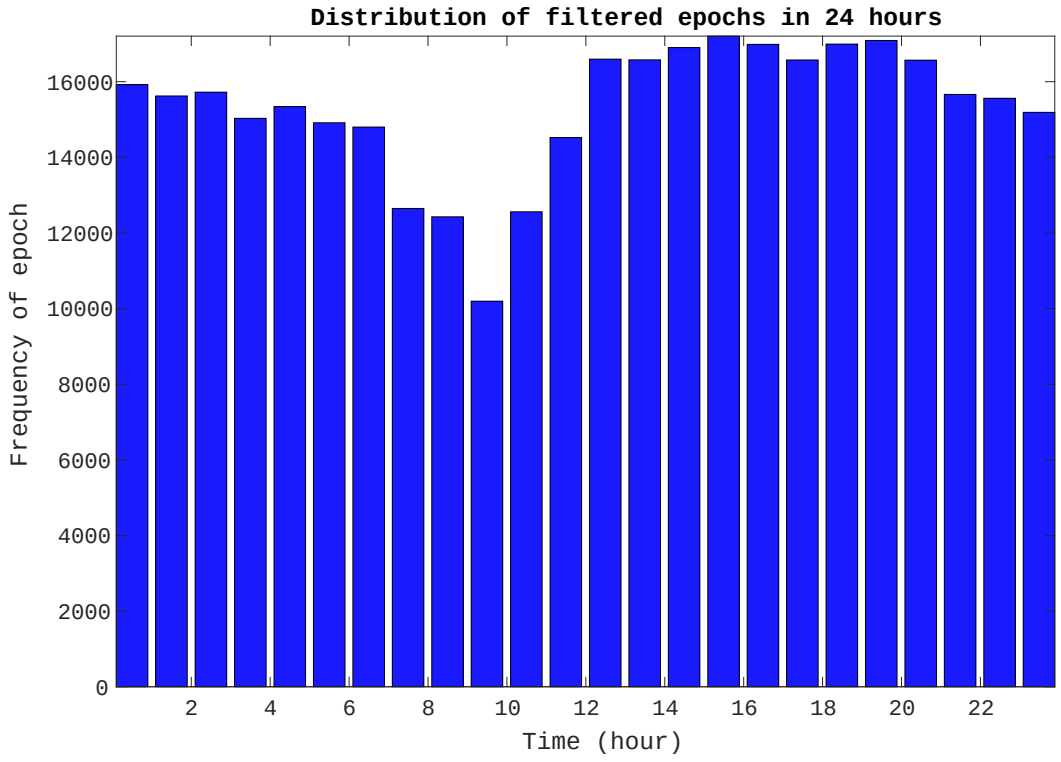


TABLE 4.2: Epoch Distribution over 24h. The total number of analyses windows is shown in this example. The windows are almost uniquely distributed except the times around 9AM in which the sensors where usually replaced or being recharged.

4.4.1 Event filtering approach

As shortly mentioned before, I chose the event filtering criterion to be the ratio of increase in average heart rate during two consecutive minutes by some fold (e.g. 1.2 fold increase). This criterion was considered as one of the features too. This thresholding criterion has to be able to include vast majority of positive examples and only a handful of negative examples.

A question that might rise is that event-filtering could compromise the fairness of the classification since obtaining the criterion needs to search all over the dataset. To escape this dilemma, I searched for such filtering criterion in a separate dataset collected some years before the start of our study in Bonn epilepsy center.

In the following, I present the inclusion rate of the event filtering based on positive and negative examples (seizures and non-seizures). Figure 4.11 and table 4.3 show how many of positive and negative events have the heart rate increase of $1.1, 1.15 ... 1.6$. As mentioned, we have learned this effect from one of our previous studies in the clinic and here too, most of the positive events (seizures) have significant heart rate increase while only limited number the negative events (non-seizure cases) show a significant heart rate

increase. A huge difference between positive and negative examples is observable in the plot. For instance, 90% of seizures have a window with atleast 1.2 increase in heart rate whereas only 4% on non-seizure events contain a window with 1.2 heart rate increase.

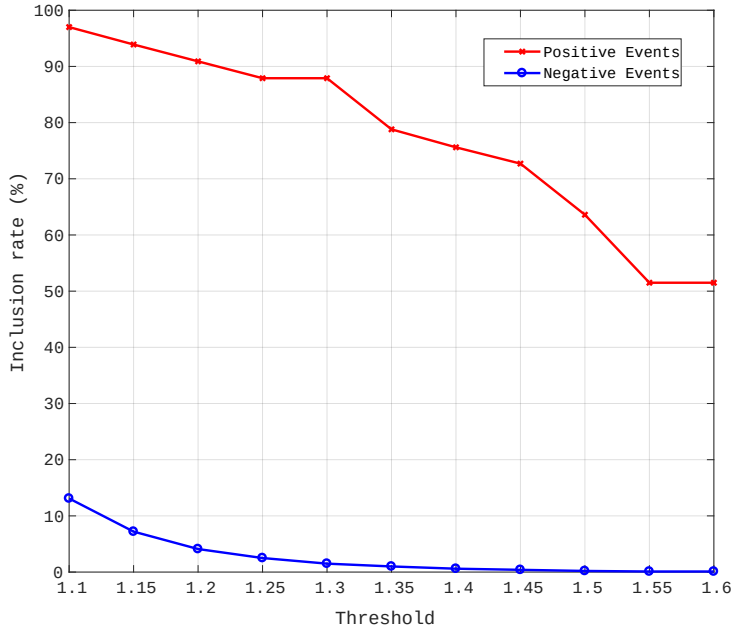


FIGURE 4.11: Event inclusion rate. Positive and negative examples are threshold based on the heart rate increase during two windows of one minute long. The results show that positive examples have decidedly different behavior in term of hear rate increase.

TABLE 4.3: Event inclusion ratio

threshold	seizure events (%)	non-seizure events (%)
1.10	97.0	13.1
1.15	93.9	7.2
1.20	90.9	4.1
1.25	87.9	2.5
1.30	87.9	1.5
1.35	78.8	1.0
1.40	75.6	0.6
1.45	72.7	0.4
1.50	63.6	0.2
1.55	51.5	0.1
1.60	51.5	0.1

Event filtering is used in our classification settings as a preliminary step to the machine learning phase. Having event filtering schema by itself has a big advantage in real world. As mentioned before, to acquire higher sensitivity, I tweak the classifier threshold and set it to lower values. This will increase the number of false alarms. With the event filtering setting compared to direct classification, the number of assessed events will

reduce significantly, and accordingly, the number of false alarms. Then in real world scenarios, a patient feedback design can assist to lower the number of false alarms (see chapter 5). Hence, choosing a threshold for the event-filtering phase will be a decisive step to the whole seizure detection task.

In the following, the result of two-step classification is presented by changing the filtering threshold. First, the classification performance of two-step classification with all features and also by applying 50% classification threshold is presented. In table 4.4, the classification performance is extensively investigated. Please note that the performance is presented in 3 levels: *event-filtering level*, *classification level* and *combined level*. Additionally, a 5-fold cross-validation is used in the proceeding results. Therefore, the results could change minutely if the random sampling is changed.

My classifier choice for the following results was a random-forest and I used the Java implementation of it provided by the Weka machine learning tool [81] using its default settings. The toolbox was then imported in Matlab to be combined with the rest of code.

According to table 4.4, the best final sensitivity,(52.5%) resulted from the thresholding value of 1.35. The best final precision was 98.4% and came from the thresholding value of 1.2. The best final F_1 score arose from thresholding with 1.45 and was 67.7%.

Two points should be mentioned. First, the precision is computed according to the total number of false alarms for all patients. Thus, we do not have a good estimation of the number of false alarms per day. However, in this setting, we have used a total of 1030 **hours** = 43 days of recording and the amount of precision should be considered accordingly. Thus, for example, 50% of precision and 30 seizures, would mean roughly 0.7 false alarms per day (30/43) which is a good record (compared to related work). Later, I will change the setting to measure the false alarms per day for different recording sessions separately. Second, although the precision is drastically high, the sensitivity in this setting is rather low. In detection systems such as these, typically the main priority is to detect whatever positive examples exist, even at the cost of lowering the specificity and precision.

On that account, to increase the sensitivity, I dropped the classification probability threshold from 50% to 10%. That means, if the classifier indicates that an event has a probability of 10% or more for being a seizure, that event will be marked as a seizure. The other possible thresholds could be used too. Nevertheless, according to our extensive trials of alternative thresholds, 10% seems to meet our expectations.

Table 4.5 illustrates the classification performance of the upper mentioned settings but with tuning the classification probability thresholding to 10%.

From the table, we can infer that the best sensitivity, 79.1%, is with an outcome of 1.25 event thresholding and the highest precision and F_1 score resulted from a thresholding value of 1.6, which are 58.2% and 54.6% respectively.

To take this one step further, we can lower the classification probability thresholding value to achieve a higher sensitivity.

One way to check the effect of different modalities on the classification performance is to exclude one modality and compare the resulting performance with other cases. Since we have three main modalities in our work, ECG, ACC, and EDA, we can exclude them one by one and repeat the classification procedure.

Table 4.6 represents the classification results of the case using only HRV features. In contrast to the case of having ECG raw features included, having only HRV features makes it more feasible to systematically match up the seizure related effects on heart rate rhythms with other clinical studies. Although raw ECG features could potentially help distinguishing seizures from non-seizures, they are mostly contaminated with movement and muscle artifacts and accordingly difficult to discern a conclusive seizure pattern.

The best sensitivity resulted from the event thresholding of 1.2 and 1.3 (76.2%). The topmost precision derived from the case with 1.6 thresholding value (55.2%) and the highest F_1 score obtained from 1.45 thresholding case (55.6%).

comparing this HRV features with the features of raw ECG and HRV combined using paired T-test, we observe a significant improvement in sensitivity ($p < 0.0005$) but insignificant increase in precision and F_1 score:

TABLE 4.7: Checking the effect of the raw ECG features classification using paired T-test. Two cases are compared: the case using raw ECG and HRV features and the case using only HRV features.

	Sensitivity	Precision	F_1 score
h	1	0	0
p	4.3583e-04	0.26	0.12
tstst	5.14	1.19	1.67

This shows us that for a given threshold, having raw ECG features could help improving sensitivity of the system but also increasing the number of false alarms.

In the next analysis, the classification of having HRV features and accelerometry is targeted (see Table 4.8).

The best sensitivity value (79.1%) is resulted from applying 1.25 threshold value. The top precision (53.5%) derived from using 1.6 threshold value for event filtering and

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	48.5	100	96.0	67.3	51	47.0	96.0	63.1
1.15	93.9	50.0	100	96.7	69.5	52.5	46.9	96.7	63.2
1.20	90.9	51.6	100	98.4	69.3	54.1	46.8	98.4	63.4
1.25	87.9	56.6	99.9	95.0	71.3	58.9	49.7	95.0	65.3
1.30	87.9	56.6	99.8	96.0	73.6	62.2	49.7	96.0	65.5
1.35	78.8	66.6	99.5	93.3	76.2	66.5	52.5	93.3	67.2
1.40	75.6	61.5	99.6	90.0	74.8	62.5	46.5	90.0	61.3
1.45	72.7	72.0	99.3	96.0	79.6	72.5	52.3	96.0	67.7
1.50	63.6	72.7	98.9	93.2	85.4	80.2	46.2	93.2	61.8
1.55	51.5	77.8	96.2	90.0	84.9	79.6	40.0	90.0	55.4
1.60	51.5	83.3	90.0	83.0	84.3	79.1	42.9	83.0	56.7

Settings: All modalities 2 step classi cation Classi er: Random-Forest Probability Threshold = 0.5

TABLE 4.4: The classification performance using event filtering and all modalities. The probability threshold is set to 50%. The results show the event filtering on left side of table, machine learning in the middle, and combined performance on the right.considering different event filtering thresholds. This table and the next tables give us parameters of a seizure detection system to achieve a particular detecting performance.

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	78.7	98.9	38.8	84.1	49.6	76.3	38.8	51.4
1.15	93.9	81.2	98.2	43.6	88.2	55.1	76.2	43.6	55.5
1.20	90.9	83.9	96.0	37.9	87.7	49.1	76.3	37.9	50.6
1.25	87.9	90.0	93.9	38.9	92.2	54.1	79.1	38.9	52.1
1.30	87.9	86.6	91.1	39.8	90.2	54.7	76.1	39.8	52.3
1.35	78.8	92.5	87.5	40.1	90.8	55.9	72.9	40.1	51.7
1.40	75.6	96.1	77.3	37.4	87.1	53.8	72.6	37.4	49.4
1.45	72.7	96.0	67.4	39.0	82.0	55.4	69.9	39.0	50.0
1.50	63.6	100	68.9	46.7	84.4	63.1	63.6	46.7	53.8
1.55	51.5	100	62.0	48.6	81.1	65.1	51.5	48.6	50.0
1.60	51.5	100	62.1	58.2	80.7	72.8	51.5	58.2	54.6

Settings: All modalities 2 step classi cation Classi er: Random-Forest Probability Threshold = 0.1

TABLE 4.5: The classification performance using event filtering and all modalities. The probability of the classifier is threshold to 0.1 to force for higher sensitivities

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	72.7	98.7	33.9	83.0s	43.8	70.5	33.9	45.8
1.15	93.9	78.1	97.7	35.5	85.8	46.8	73.3	35.5	47.8
1.20	90.9	83.9	95.9	36.0	88.9	49.5	76.2	36.0	49.0
1.25	87.9	83.3	93.9	35.9	86.3	48.5	73.2	35.9	48.2
1.30	87.9	86.7	91.1	37.7	87.6	51.8	76.2	37.7	50.4
1.35	78.8	88.9	87.5	40.1	86.4	53.8	70.0	40.1	51.0
1.40	75.6	92.3	81.7	42.1	87.7	57.0	69.8	42.1	52.6
1.45	72.7	92.0	74.0	44.7	83.6	60.2	66.9	44.7	55.6
1.50	63.6	90.9	68.7	46.4	80.4	60.9	57.8	46.4	51.5
1.55	51.5	94.4	58.4	45.9	76.6	61.1	48.6	45.9	47.2
1.60	51.5	94.4	61.8	55.2	78.1	69.1	48.6	55.2	51.7

Settings: HRV features 2 step classi cation Classi er: Random-Forest Probability Threshold = 0.1

TABLE 4.6: The classification performance using event filtering and only **HRV** features. The probability of the classifier is threshold to 0.1 to force for higher sensitivities.

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	75.8	98.7	32.2	83.0	43.2	73.5	32.2	44.8
1.15	93.9	78.1	98.0	39.5	89.2	52.5	73.3	39.5	51.3
1.20	90.9	80.6	96.2	39.6	89.0	52.5	73.2	39.6	51.4
1.25	87.9	90	93.4	38.4	92.0	52.9	79.1	38.4	51.7
1.30	87.9	83.3	85.4	35.6	88.6	50.1	73.2	35.6	47.9
1.35	78.8	85.2	86.0	35.4	85.6	49.6	67.1	35.4	46.3
1.40	75.6	92.3	79.0	38.9	86.4	54.7	69.8	38.9	50.0
1.45	72.7	92.0	68.1	40.4	80.7	55.6	66.9	40.4	50.4
1.50	63.6	95.5	69.8	46.7	82.9	62.2	60.7	46.7	52.8
1.55	51.5	94.4	62.2	47.8	78.5	63.5	48.6	47.8	48.2
1.60	51.5	94.4	56.4	53.5	75.6	67.5	48.6	53.5	50.9

Settings: HRV + ACC features 2 step classification Classifier: Random-Forest Probability Threshold = 0.1

TABLE 4.8: The classification performance using event filtering and only **HRV** and **accelerometry** features. The probability of the classifier is threshold to 0.1 to force for higher sensitivities.

the peaking value for F_1 score (51.7%) resulted from 1.25 thresholding value. We are interested to check the cumulative effect of accelerometry features on HRV features. T-tests reveal that there is no classification effect of adding accelerometry features to HRVs whatsoever. In term of sensitivity, precision, and F_1 score, the changes are highly insignificant. Please see table 4.9 for the detailed information.

TABLE 4.9: Checking the effect of using accelerometry features in classification using paired T-test. Two cases are compared: the case using only HRV features and the case using **HRV** together with **accelerometry** features. The test rejects the hypothesis.

	Sensitivity	Precision	F_1 score
h	0	0	0
p	0.9521	0.8574	0.6865
tstst	-0.0609	0.1820	0.4096

Additionally, we were interested in discovering the effect of seizures in electrodermal activities. In table 4.10, the classification results of using HRV and electrodermal activity features are shown. The results reveal that the topmost overall sensitivity is derived from the 1.25 thresholding value case (79.1%), the best precision is from the 1.55 thresholding value (52.2%), and the highest amount of F_1 score reported from the 1.4 thresholding case amounted to 55.7%.

To check the effect of adding electrodermal features, I ran another set of T-tests. Similar to the previous case (adding accelerometry features), there was no incremental effect on the evaluation metrics. All of the tests reported highly insignificant effects. Please see table 4.11 for more details.

TABLE 4.11: Checking the effect of using electrodermal features in classification using paired T-test. Two cases are compared: the case using only **HRV** features and the case using **HRV** together with *EDA* features. It can be observed that EDA did not have any significant added effect on the classification performance (the test rejects the hypothesis).

	Sensitivity	Precision	F_1 score
h	0	0	0
p	0.9145	0.8620	0.8277
tstst	-0.1088	-0.1761	-0.2205

Next, we checked the classification performance of using all modalities: HRV, accelerometry and electrodermal features. Table 4.12 shows the detail classification performance of combining all mentioned modalities.

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	72.7	98.9	35.8	83.1	46.4	70.5	35.8	47.5
1.15	93.9	78.1	97.6	35.0	85.8	46.7	73.3	35.0	47.4
1.20	90.9	77.4	96.2	35.5	84.5	45.4	70.4	35.5	47.2
1.25	87.9	90.0	93.4	35.6	89.6	49.9	79.1	35.6	49.1
1.30	87.9	83.3	90.9	39.4	87.6	52.3	73.2	39.4	51.2
1.35	78.8	88.9	89.1	44.5	90.1	59.1	70.0	44.5	54.4
1.40	75.6	92.3	79.0	39.7	86.4	55.4	69.8	39.7	50.6
1.45	72.7	92.0	67.4	38.7	80.4	54.4	68.9	38.7	49.0
1.50	63.6	95.5	73.2	51.4	84.6	66.6	60.8	51.4	55.7
1.55	51.5	100	66.2	52.2	83.1	68.2	51.5	52.2	51.8
1.60	51.5	94.4	56.1	51.1	75.6	65.6	48.6	51.1	49.8

Settings: HRV + EDA features 2 step classi cation Classi er: Random-Forest Probability Threshold = 0.1

TABLE 4.10: The classification performance using event filtering and only **HRV** and **electrodermal** features. The classification probability is threshold to 0.1 to allow for higher sensitivities.

Filtering		Machine Learning Performance					Overall Performance		
Threshold	Positive Inclusion Ratio	Sensitivity	Specificity	Precision	AUC Of ROC	F_1-Score	Sensitivity	Precision	F_1-Score
1.10	97.0	81.8	98.8	38.2	86.1	49.4	79.3	38.2	51.6
1.15	93.9	81.3	97.9	40.4	90.3	53.9	76.3	40.4	52.8
1.20	90.9	83.9	96.4	39.2	89.1	52.4	76.3	39.2	51.8
1.25	87.9	83.3	93.2	35.5	88.3	49.2	73.2	35.5	47.8
1.30	87.9	86.7	91.9	41.8	90.7	56.4	76.2	41.8	54.0
1.35	78.8	81.5	86.2	35.6	84.3	49.4	64.2	35.6	45.8
1.40	75.6	88.5	77.3	37.0	84.0	51.9	66.9	37.0	47.6
1.45	72.7	96.0	67.4	40.0	82.0	56.2	69.8	40.0	50.9
1.50	63.6	100	64.4	42.8	82.3	59.6	63.6	42.8	51.2
1.55	51.5	94.4	52.2	43.6	73.8	59.1	48.6	43.6	46.0
1.60	51.5	100	48.2	48.6	74.4	65	51.5	48.6	50.0

Settings: HRV + ACC + EDA features 2 step classi cation Classi er: Random-Forest Probability Threshold = 0.1

TABLE 4.12: The classification performance by applying event filtering and using **HRV**, **accelerometry**, and **electrodermal** features. The classification probability is threshold to 0.1 to allow for higher sensitivities.

The best sensitivity (79.1%) is obtained with 1.1 as a threshold; the best record for precision came from setting the threshold to 1.6 with 48.6%, and the highest F_1 score is resulted from 1.3 thresholding case amounted to 54.0%.

Similar to previous cases, I ran several T-tests to check for possible improvement of combining modalities. In this case, it can be realized once again that adding excessive features from other modalities to HRV does not provide us with any advantage. All T-tests reported highly insignificant changes in either direction.

TABLE 4.13: Checking the effect of using both accelerometry and electrodermal features in classification by applying paired T-test. Two cases were compared: the case using only HRV features and the case using HRV together with accelerometry and electrodermal features.

	Sensitivity	Precision	F_1 score
h	0	0	0
p	0.7603	0.6717	0.9217
tstst	-0.3093	0.4301	0.0995

4.4.1.1 Day and night classification

I have also investigated the difference of day vs. night classification. The data between $10pm$ to $6am$ is considered to be the night time as the patients normally sleep during this time. Otherwise, the recording times between $6am$ to $10pm$ were considered to be the day-time recording.

Next, I have performed a leave-one-patient-out cross validation and performed an ECG based classification. I chose however different classification probability thresholds for the day and the night time data.

In this setting, apart from random-forest, I made use of another classifier, *Naive-Bayes-Updateable*, concurrently. The reason was that the Naive-Bayes-Updateable classifier uses a generative approach to find a model, as we discussed in section 4.3.5, and we can give it a chance to help the random-forest classifier to achieve higher sensitivity.

Table 4.14 shows the overall performance of our seizure detection system as well as days and nights split performance.

It can be observed that the day classification is significantly more reliable than night's even though we have used lower classification probability threshold for the night data to force for higher sensitivity. One known reason to this is the *sleep arousal* effect, in which the heart rate increases abruptly around 6 to 10 times in an hour during the sleep. This

Time	Overall (%)	Day (%)	Night (%)
Sensitivity	61.8	71.7	28.6
Precision	51.2	51.5	28.6
F_1 score	56.0	59.9	28.6

TABLE 4.14: Day vs. night results. It can be seen that the classifier performs better during the days compared to the nights.

can be easily mistaken with seizures. The difference between the performance of seizure detection classification in days and nights is discussed widely in [11].

4.4.1.2 Important features for classification

I have conducted a feature search approach based on t-test to check for the information every feature carries. That is, for every feature, an individual t-test is conducted to check the amount distribution difference between seizure cases and non-seizure cases (performed on training-data). Based on the results of t-tests, the 15 top most informative features are listed below:

Feature Name	Description
diffHeartBeatPrePost	Difference of heart-rate, pre-ictal vs. post-ictal
diffMeanRRIntervalPrePost	Difference of average RRI, pre-ictal vs. post-ictal
diffEntropyRRIntervalPrePost	Difference of entropy of RRI, pre-ictal vs. post-ictal
diffCVIPrePost	Difference of CVI, pre-ictal vs. post-ictal
diffCSIPrePost	Difference of CSI, pre-ictal vs. post-ictal
diffRRIPSDVLFPrePost	Difference of PSD of RRI in VLF, pre-ictal vs. post-ictal
diffRRIPSDLFPrePost	Difference of PSD of RRI in LF, pre-ictal vs. post-ictal
diffRRIPSDHFPrePost	Difference of PSD of RRI in HF, pre-ictal vs. post-ictal
diffRRIPSDTotalPowerPrePost	Difference of PSD of RRI in all freq., pre-ictal vs. post-ictal
diffRRIPSDPowerRatioPrePost	Difference of PSD of RRI in power ratio, pre-ictal vs. post-ictal
diffNN50PrePost	Difference of NN50, pre-ictal vs. post-ictal
diffPNN50PrePost	Difference of PNN50, pre-ictal vs. post-ictal
ecgArousal	Relative heart-rate fold change
meanNormDisplMidPost	Average displacement
stdNormDisplMidPost	Standard deviation of displacement

It is worth to notice that the most important features are those which characterize the difference between pre-ictal and post-ictal and extracted from ECG.

4.4.1.3 Summarizing the results of event filtering approach

The event filtering approach is an ideal method for seizure detection systems specially if they are built based on limited number of seizures. The reason is that most of the potential processing load is excluded by an early and easy event filtering system and

therefore, it could be efficiently running on a small processor. The other advantage of using the event filtering approach is to reduce the number of false alarms and it is due to filtering out the cases which are not similar seizures physiologically in the first place.

Here, I conducted a two-step arrangement. In the first step, windows with seizure-like patterns were selected for the second step, and in the second step, machine learning was performed (the criteria for the event filtering phase was extracted from a previous study). The criterion in particular was the following: the heart rate increases e.g. 1.2 times in average within two consecutive minutes.

Random-forest was our main classifier of choice since it showed best performance in most of the settings. Additionally, it was convenient with random-forest to perform for the classification probability thresholding. Moreover, random-forest could also be used to trace the relevance and importance of the features used in the classifier. From the results, we can infer the following points:

- We can extract features from the raw ECG signal and those features may help increase the sensitivity of the system significantly. However, raw ECG features also raise the number of false alarms and as a result, the precision and F_1 score does not improve significantly.
- Heart Rate Variability (HRV) features are sufficiently informative for seizure detection. Although other modalities such as accelerometry and electrodermal activity features reveal some seizure-related activities, based on our data, they do not provide any added information, which in turn does not improve the performance.

4.4.2 Direct classification approach

In the next step, I developed a direct machine learning approach by skipping the event filtering phase and sticking to the conventional machine learning procedure. Since we do not have any event filtering, the number of negative examples (non-seizure) outnumbers the positive examples drastically. Thus, to reduce the complexity of the classification, a random portion of negative examples for the training phase can be used. Nonetheless, in the testing phase, the whole negative and positive samples can be chosen to get an accurate evaluation of classification metrics.

An important point here is that splitting the windows to training and test sets will end up having samples in training and test sets which are timely correlated. This will ease the classification procedure and will produce biases leading to outstandingly good results. To avoid it, I split the testing based on the sessions of recording. In this way, there wont be any sample in the test sets which is timely correlated to the training set.

Training and testing classification models on non-event-based data, we have obtained acceptable sensitivity but with extremely poor precision.

Table below shows the best result of non-event-based classification. It can be understood from the results that this approach is not as efficient as event-filtering approach for seizure detection.

Performance metric	Direct classification
Sensitivity	86.2 %
Precision	2.5 %
F_1 score	4.8 %
False-alarm per day	24.0

TABLE 4.15: Direct classification performance. In this approach we skipped over the event filtering approach. The results reveal the importance of event filtering approach as the number of false-alarms rose significantly, and consequently, reducing the precision and F_1 score.

With direct classification approach, a higher sensitivity can be achieved but it is not giving us a reasonable precision and F_1 score. Therefore, it could be used in systems in which the sensitivity is more important than the other factor.

4.5 Prospective evaluation part 1, mobile EEG/ECG

One important result of our previous project was proving that the most significant modality for detecting seizures was the heart rhythm. It was also demonstrated that ECG is sufficient for seizure tracking purposes. In this phase of study, I aimed at validating the algorithm developed from the last sections by new collected data and to see how our trained model would function in a more realistic scenario. For this purpose, we collected another set of patients with (mobile) ECG recordings obtained in the Epileptology Department at the University of Bonn Medical Center.

4.5.1 Patients, sensors, data

Epilepsy patients who were hospitalized in epilepsy clinic were asked to carry a portable recording device with themselves during their stay. We have recorded a total of 30 patients, 8 females, 22 males (ages: 38.6 ± 15.7).

All data of patients in this group were collected during standard clinical care, so that additional informed patient consensus was not required, as approved by the local medical ethics committee (No. 352/12). All patients signed a consent letter for their participation in the study prior to recordings.

Epilepsy patients who were hospitalized in the Epileptology department were asked to carry the (portable) recording device with them during their stay. We recorded a total of 30 patients: 8 females and 22 males (ages: 38.6 ± 15.7). There were a total of 33 recording sessions, each lasting 1 to 2 days. In total, we recorded for 758.44 hours, registering 49 seizures from patients.

The distribution of seizures is rather different to what we had in the previous phase and it adds to the difficulty of correctly detecting seizures based on the model learned previously (see figure 4.12 for more details).

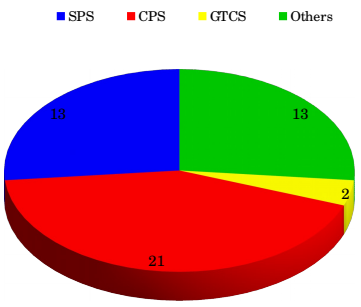


FIGURE 4.12: Distribution of seizures in Mobile-EEG recording. This image indicates that the distribution of seizure types is different to the first study (apart from the recording condition). This difference can be used to determine how our developed seizure detection algorithm would generalize on a new set of data.

This time however, instead of Movisens sensor or in-bed ECG recording, we collected our data with another mobile device provided from Micromed company (see image 4.13)

FIGURE 4.13: Mobile Micromed EEG/ECG recording device. The patients could freely move within the clinic and perform their daily routine (walking, sitting, watching TV, etc. while being recorded by the device. The idea was to record patient in a different environmental and physiological condition.)

The device was called *micromed mobile EEG* which was able to record and register ECG signals alongside EEG recordings. The patient could freely move inside the clinic while carrying the device with them.

4.5.2 Methods

In this phase of study, the technicalities is not outlined. The same methodology applied in the last project, applies here too, but with a minor changes. Here, we limited ourselves to HRV features since we previously discovered that they are the most informative. Therefore, neither accelerometry nor electrodermal measurement is involved.

Altogether, 65 HRV features were measured from the ECG signals. As previously stated, our main goal was to confirm that the previous results are not biased or have confounded variables. To do this, the data obtained in this phase was blindly tested against the model from the last project. That is, our model was trained using only the Movisens ECG dataset and then checked for the classification performance of the trained model against the new (mobile EEG/ECG) dataset. This setting has some advantages in that it helps us recognize whether or not our trained model is susceptible to noise or other changes in patient activity. This way of evaluating the model is know as *prospective evaluation* in the realm of medicine as opposed to *retrospective evaluation*.

4.5.3 Results

For each threshold from 1.1 to 1.45, I trained a model based on Movisens data and tested it against Mobile-EEG dataset with the same threshold. While training, I chose 10 best discriminative features among all 65 features and train the model accordingly. In the test phase, the same sets of features is selected for testing. In the following table the results of testing on Mobile-EEG is presented (table 4.16).

From the results, it can be realized that the detection rates are not as good as previous dataset even though they are lying in an acceptable range. The reason could be the followings. First, the seizures presented in the new dataset do not have the same distribution as previous dataset and accordingly is not optimized fully for it. Second, the new data set contains lots of aura seizures and short time focal seizures (SPS) compared to the last dataset. These types of seizures do not carry the typical physiological signs and symptoms of the seizures. The third reason is that 10 out of 49 seizures have heart-rate increase of less than 1.15 during seizures (20%).

All in all, the results of this phase reveals the difficulties and short-comings of detecting all types of features in different conditions.

Threshold	Sensitivity	Precision	F_1 Score	Total num of False-Alarms	False Alarm Per Day
1.1	53.06	5.53	10.01	444	14.4
1.15	59.18	6.69	12.03	404	12.78
1.2	50.00	7.57	13.15	293	9.27
1.25	45.83	8.66	14.57	232	7.34
1.3	44.44	13.42	20.61	129	4.08
1.35	35.71	16.30	22.38	77	2.43
1.4	44.73	14.28	21.65	102	3.22
1.45	31.42	22.91	26.50	37	1.17

TABLE 4.16: Results of classification on Mobile-EEG and with different thresholds. This table also gives us a map to opt for a desired seizure detection settings. To aim for higher sensitivities, a lower thresholding value is needed. To go for a smaller false-alarm and higher precision, a higher thresholding value is needed. In term of F_1 score, the higher threshold produces the best result.

4.6 Prospective evaluation part 2, Epitect ECG

In addition to the last prospective evaluation, we have validated the ECG based detection algorithm on a third dataset called Epitect. The Epitect dataset collected over the course of two years in epileptology center in Bonn. The project was composed of three concurrent recordings, the clinical ECG and two wearable devices to record photoplethysmography (see chapter 5). The following results are obtained by applying my algorithm to the first 97 patients of the Epitect study. The work has conducted in collaboration with my colleague Dr. Jan Baumann [6].

4.6.1 Patients and data

A total of 97 patients participated in the study. These epilepsy patients were admitted in Bonn epilepsy center and underwent Video-EEG monitoring. Among them, 255 seizures were recorded.

All patients gave informed consent for their participation prior to recordings and the study has been approved by the local medical ethics committee (No. 355/16).

As opposed to patients in the mobile EEG/ECG study, patients of this study had similar conditions with the main study as they were all recorded in bed in non-walking conditions but with standard clinical ECG recording developed by Micromed company.

4.6.2 Results

We split the patients of this study to 50 first patients to be used for the training phase and the rest 47 patients for testing.

This algorithm largely replicated the same performance of the main algorithm:

Performance metric	Epitect ECG
Sensitivity	39 %
Precision	73 %
F_1 score	51 %
False-alarm per day	0.4

Together with the other validation groups, we can conclude that the algorithm works satisfactorily well for patients with less physical activity and works moderately good for the moving patients. This is however a good news for epilepsy patients as one of the most dangerous threats to them is Sudden Unexpected Death in EPilepsy (SUDEP) [3, 97]. SUDEP happens as a consequence of an epileptic seizure. The danger of SUDEP is much larger during the sleep, as in most of the cases, patients have a small chance of having people around them to take care of the situation.

The algorithm developed in this section would be of great benefit to the epilepsy patients especially during sleep. An on the fly alarming system can be developed upon it to inform the care givers, and therefore, rescue the patients.

4.7 Developing deep learning algorithms on ECG data

Aside from applying conventional machine learning techniques, I have also investigated two deep learning methods to detect seizures in our data. For this sake I used the Epitect dataset which contains fairly high amount of recording and seizures considering the fact that the classification performance on this dataset were significantly good using conventional machine learning methods.

4.7.1 Method 1, Convolutional Neural Networks (CNN)

Referring to chapter 2, since CNNs are predominantly designed to learn from $2D$, $3D$ or $4D$ data such as images and videos, to be able to use their full strength, an idea was to feed the ECG data in the form of images to the network. For this sake, I measured a $2D$ map of power spectral density (PSD) of continuous wavelets transformation (CWT)

from ECG data (see image 4.14). This map represents the power spectrum of ECG. The representation of R-Peaks of ECG should be also theoretically embedded in some time-frequency regions of the maps. Other heart-rate related activities should be also reflected in the map. Image 4.14 show an instance of PSD resulted from CWT on ECG.

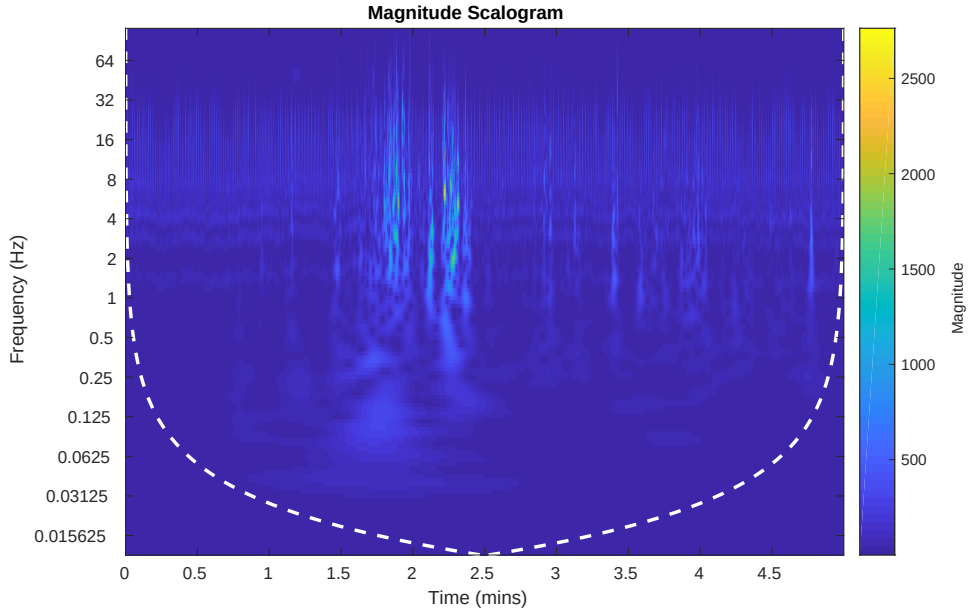


FIGURE 4.14: Contineous Wavelets Transformation of ECG. The map show a time-frequency representation of the power of ECG signal. Changes in the ECG signal should be reflected in the map across different frequencies.

A five minutes windows is considered $(300s \times 256Hz)$. The resulted transformation image is a 134×76800 in size due to the frequency resolution and time resolution. This image should be resized to a moderate image size before putting it in a CNN. The long image is then resized to 128×512. Multiple variations and settings were tested to build a CNN. Below I present a network of 4 layers for this time-frequency image. Image 4.15 represents the network structure pictorially. The network is composed of 3 convolutional layers and one dense layer. Every convolutional layer is accompanied with a max-pooling layer. The dense layer is preceded with a dropout section to prevent possible overfitting.

Although I have tested the ECG time-frequency images against multiple number of CCNs, the networks were never good enough to distinguish seizures from non-seizures. I have also applied techniques for data imputation and data augmentation to increase the number of positive examples, that was of no help to the networks.

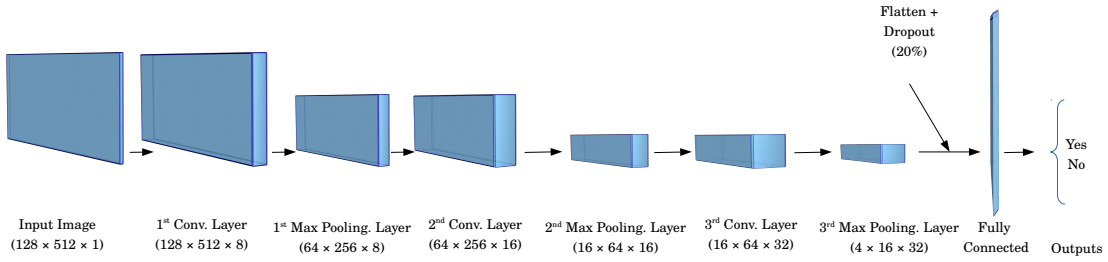


FIGURE 4.15: Convolutional neural network design on ECG time-frequency image. The network is supposed to extract the features out of the image and decide whether the heat-map represents a seizure or not. The network is composed of 4 layers, 3 convolutional layers and one dense layer. Each convolutional layer is proceeded by a max-pooling layer to reduce the degree of freedom. The size of each layer is labeled in the figure. The dense layer (fully connected) is preceded by a drop-out section in which 20% of the paths are dropped randomly to account for overfitting.

Next, I have also used our old trick of event filtering, to use only images with significant associated heart-rate increase. That did not have any impact on the networks neither, and the classification results did not improve from random chance.

4.7.2 Method 2, Deep Multi-Layer Perceptron (dMLP)

Since using CNN did not show to have any effect on the ECG classification, I developed a modern deep multi-layer perceptron (dMLP) network which benefited from ReLU activation function and drop-out, to check weather this can improve our results. Here, instead of the images used for developing CCN , I used our extracted ECG feature (65 features, 1.35 thresholding value) explained broadly in this chapter. Training on the first half of patients in Epitect dataset and testing on the second half, reveals superior results than CNN:

Performance metric	dMLP
Sensitivity	55.7 %
Precision	42.6 %
F_1 score	48.2 %

The classification performance is in all of its terms inferior to of the random-forest and it shows that tree-classifiers tend to perform better on our datasets.

4.7.3 Conclusion

The above results indicate that perhaps the amount of information carried by such datasets are not uniform enough to converge a neural network. The reason tree classifiers work best in such scenarios is that the data is irregularly scattered in the feature space and therefore, tree classifiers like random-forest can model the exceptions much easier naturally since they search for discrete combinations of features which can distinguish the classes.

The other reason can be the size of the dataset. While the number of negative examples are huge, the number of positive examples are significantly low. And artificially increasing the number of positive examples does not improve the amount of explorable patterns for neural networks.

The third reason why algorithms like CNNs are not optimal in such situations is referring to the data itself. Wider neural networks in theory are harder to converge than narrower ones. A big image would cause too many degrees of freedom in each layer and therefore, cumbersome to train. A way to tackle such problems is to feed more patterns with diversity, which is not easily possible to obtain in our case.

These facts may imply that applying deep learning techniques does not always lead to the best results. Deep learning techniques are a powerful tool in the machine learning toolbox but should be used only if profitable.

4.8 Summary

To give people with epilepsy a proper counseling, seizure diaries were assigned to patients to record and report their seizures as they take place. Although seizure diaries are considered to be the gold standard for monitoring seizures in the clinical trials, multiple studies have shown that these reports are markedly inaccurate [37, 58, 65].

A solution to this problem was to develop **machine learning algorithms** to be used on portable and wearable devices to track and record and register seizures based on biofeedback signals such as EEG, ECG, AAC, EMG, EDA. EEG is the gold-standard for seizure detection but it is not a practical every-day use solution. Devices for detecting primary motor seizures evolved significantly in the last years [8, 23, 78] and are mostly dependent on movement sensors. However, there is a gap for seizure detection devices to detect all types of seizures. In this chapter, I aimed at bridging this gap and developing a system for detecting all types of seizures (seizures with/without movement involved).

For this sake, I presented three datasets in this chapter. I have exhausted the first dataset to develop machine learning algorithms for seizure detection (Movisens dataset). The second two datasets (Mobile EEG/ECG and Epitect datasets) were merely used to evaluate the algorithm from the first group prospectively (as suggested also in [8]).

Three windows, pre-ictal, ictal, and post-ictal windows were considered. For each window, I calculated features across all modalities. Altogether 225 features were calculated. This shaped a feature vector representing an event. Among all feature vectors, feature vectors with abrupt heart-rate changes were pre-selected, and a machine learning setting which suits this data was developed to search among the feature vectors and yield a model for seizure detection. Other machine learning approaches such as classes of deep learning, have also been tested.

The work in this chapter compared to the work reviewed in [11, 69] provides a practical approach for seizure detection to help patients with epilepsy in detecting **all** types of seizures. This should help them in better monitoring their epilepsy in their clinical trial and ultimately improving their quality of life. The performance is superior to other reported ECG related work. Compared to human level report, the system performs better in sensitivity, a bit inferior in precision but showing slight performance increase in term of $F_1\,score$. With higher sensitivity than humans, it is a good candidate to prevent sudden unexpected death caused by epilepsy (SUDEP) during sleep.

The results also suggest that among tested modalities, only algorithms based on ECG data can be used for the clinically meaningful automated detection of all seizure types and that the use of further biosignals does not seem to bring any relevant improvement.

The study also highlights the importance of the conditions under which such algorithms are developed and highlights the limitations of automated detection technologies for all types of seizures.

Chapter 5

Photoplethysmography towards portable seizure tracking

After testing and proving the abilities of HRV-based seizure detection, we searched for an easy-to-use and affordable solution for putting our developed seizure detection system into effect. Our goal was to search for devices that patients could easily wear and perform day-to-day activities without difficulty.

A recent technological advance called *photoplethysmography* (PPG) has emerged in hand-held devices and sport/smart watches which measures heart rate. Unlike ECG which tracks and reflects the electrical pulses resulting from heart activities, PPG can detect the minute momentary changes in color by using either photosensors or a camera. As blood is pumped from the heart, it circulates throughout the entire body. The effect of the blood flow can be observed via skin color. Therefore, PPG helps determine the time at which the skin becomes darker and measures the time between color peaks. In this manner, the heart rate can be estimated.

However, detecting peaks in PPG is not a trivial task as multiple peaks might be close together and consequently, it can be harder to ascertain the true heart rhythm. Ordinarily, the heart rate is extracted via the fast Fourier transform (FFT). FFT helps PPG estimate the correct heart rate while also diminishing the time resolution of the detection. Accordingly, using FFT in PPG is always a compromise between a frequency domain resolution and a time domain resolution. This will be addressed in the next section.

Next, we tried to find the most reliable and optimum PPG device currently available on the market. The potential PPG product had to fulfill certain criteria:

1. It must be able to measure the signal continuously for nearly 24 hours.
2. If the data analysis is not functioning within the device, then it should be able to transmit the data actively and continuously to a mobile device.
3. The measured heart rate must be available in real-time.
4. The device should be affordable.

5.1 Related work

Laázaro et. al. [73] proposed using PPG for heart rate tracking in a study for tracking sleep-apnea in children. They used PPG to classify normal heart rate activities vs. apnea cases. It turned out in their work that the amplitude of PPG signal is used as an informative feature for detecting apnea cases.

Van Andel et. al. [100] compared how effective a PPG recorder (Mio Global alpha [82]) is compared to ECG for seizure recording. They recorded 7 epilepsy patients during sleep and wakefulness and realized that there is no significant difference between Mio PPG recording and clinical ECG during seizure and non-seizure times. For matching evaluation, they measured the root-mean-square of differences between those signals. According to their work, the best match of PPG and ECG pulse however happens during sleep due to lack of movement artifacts.

In a work similar to our settings, Vandecasteele et. al. [102] studied the pattern of heart rate changes across three different recording means: clinical ECG, a wearable ECG and also wearable PPG devices. In previous chapter, more broadly, we have discussed their ECG results and their algorithm. Regarding their PPG work, they have collected 47 seizures in total over the course of 701 hours, recording from 11 participants using Empatica E4 PPG recorder [32]. In their algorithm to detect seizures, they followed the method suggested in [73] in which the pulse rate variability (PRV) instead of HRV is used. PRV in fact is the differences of the locations in medium amplitude points of PPG signal. They initially used an adaptive thresholding to discover the rapid changes in pulse and then selecting sections with such features. Next step is to obtaining PRV and measuring 3 features: heart rate peak, 60s average heart rate before heart rate increase, and 60s standard-deviation of heart rate before heart rate increase. Thereafter, they used SVM to classify the cases. They achieved 32% sensitivity with 1.8 false-alarm per hour (43.2 per day).

Jan Baumann [6] in our group conducted a study to track the heart rate of epilepsy patients based on video data. That is, the video of the patients in the monitoring room

could be used to extract the heart rate activities. He realized that the green channel in RGB images of the skin is affected most by heart activities and can be used to extract the heart rate rhythms, and consequently monitoring the epilepsy patients. There were however two points to be considered to have a good recording. First, the skin should not be covered with make up. Second, the skin patch of interest should not be moving much in the video. This method requires color image and hence, the images recorded in infrared spectrum can not be applied.

5.2 Wrist-worn PPG

Based on our mentioned required points, we performed a search to find the most suitable candidates. For smart watches, we chose the following candidates.

We chose the ® Samsung Gear $S3$ watch as it has a longer battery life for 24-hour recording and streaming, and it also provides direct access to PPG sensor values (Tizen ® operating system). The other Android-based watches provided features almost within the same range. The Apple watch was also a good candidate, but not optimal for instantaneous heart rate availability. We also studied sport watches that were available at the time of study on the market.

We found two good candidates: Empatica ® $E4$ and Mio ® Alpha 2. Both watches could measure and stream the data for a long period of time and their recordings were instantaneously available at the receiving end. Mio is advantageous in that it can record for more than 24 hours (vs. 20 hours) and is significantly more cost-effective.

To select the more reliable watch, we conducted an experiment (pilot study) at the clinic. We wanted to determine which watch measured the PPG signal more accurately and near the ECG level. We invited five people to participate in the study and asked them to wear both devices on both wrists as well as a portable ECG recording device to acquire the ground-truth ECG. The participants were asked participate in a $\sim$ 20-minute experiment in which their heart rate was measured under different conditions: sitting, standing still, walking, running, and climbing the stairs with full power. We also recorded the patients with a PPG in-ear recorder which was employed in another concurrent study at the clinic (see it later in this chapter).

The results of our pilot study were as follows: under normal conditions (i.e. sitting and slowly walking), all of the sensors performed as accurately as the ECG. However, in the stress cases, the Mio watch was able to keep the heart rate trends in most cases (4 out of 5). The Samsung watch and in-ear sensor displayed the same result in stress situations (2 out of 5). Hence, according to our study results, we chose the Mio watch for our

study. Also, another seizure recording study was published at the time using the Mio watch as a recording medium [100].

FIGURE 5.1: Mio pulse sensor [82]. The watch, compared to available watches in the market provides more accurate heart rate and in better time resolution ($1Hz$). The watch could record for the minimum of $24h$ and and was easy to wear. The recorded PPG based heart rate were transmitted on the fly to a phone through low energy Bluetooth.

5.2.1 Patients, Data, Recording Media

In this study, we recorded 100 patients, typically for one day or more. In total, we obtained 5650 hours of recording.

The Mio watch was equipped with a low energy Bluetooth module which transmits the PPG measured heart-rate recordings using a scale of 1 second to a receiving device. For this study, we paired a phone with the Mio watch and used Wahoo app [100] to record the PPG data. We could then transfer the PPG data as a CSV file format along with their respective times.

Image 5.2 shows a sample Mio HR recording of 24 hours. The upper panel shows the HR values and the red points indicate the seizure times while the lower panel displays the corresponding heart rate changes for two consecutive minutes for each of the recording points. One can see that the seizure detection pattern in this case is not a trivial task due to the large number of non-seizure values which have heart rate increased cases even much higher than seizure cases.

Image 5.3 illustrates what a magnified version of a seizure recorded by Mio watch might look like. The figure is a cutoff of figure 5.2 during seizure time. The red area in its upper panel indicates seizure times. In the lower panel, the measured minute-to-minute changes in heart rate are shown.

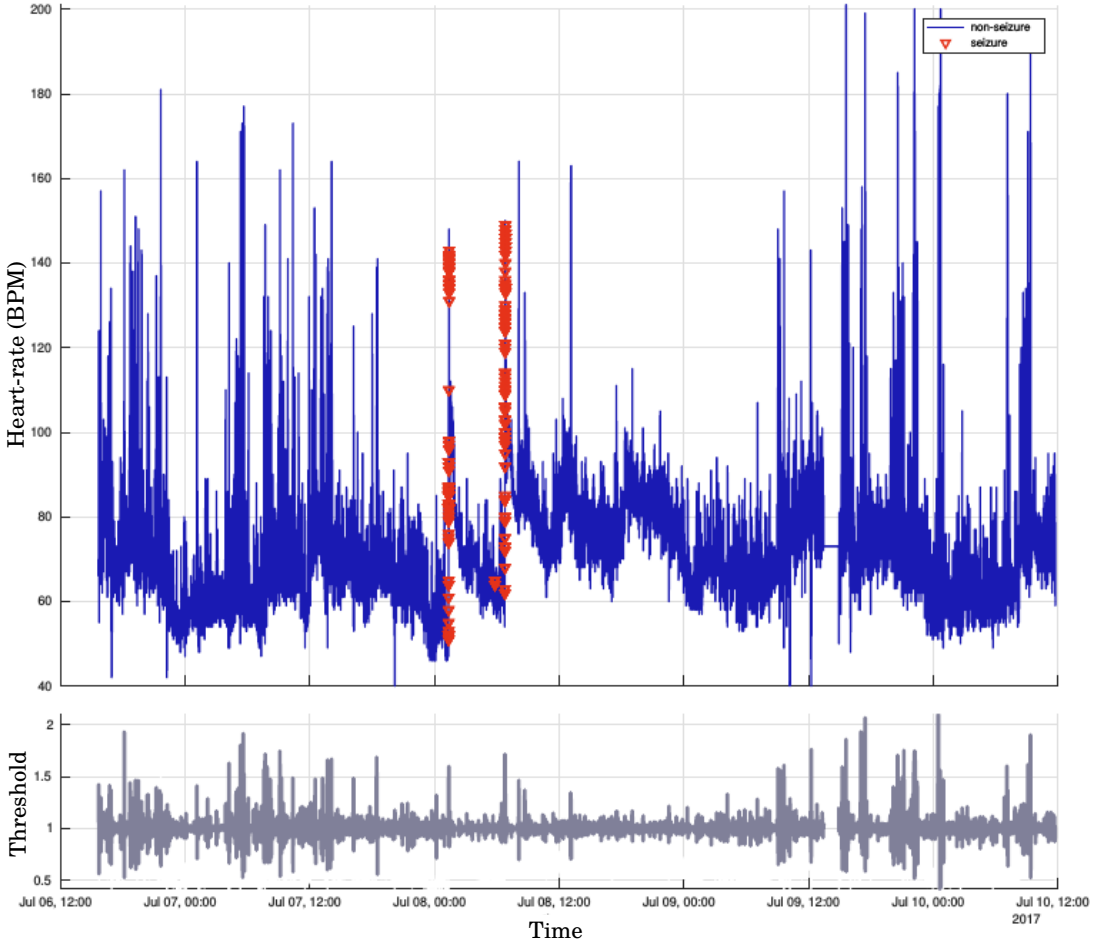


FIGURE 5.2: Mio recording and seizures. In the upper panel, the heart rate recording signals of Mio sport watch is shown. The red signs indicate seizure times. In the lower panel, the momentary heart-rate fold change during consecutive minutes is shown.

5.2.2 Method

Our goal was to record a seizure from an epilepsy patient using the Mio watch. In order to draw a solid conclusion regarding PPG technology, we recorded as many patients as possible with Mio in order to express statistically whether PPG technology is a dependable means for seizure detection.

Therefore, we simultaneously recorded our patients using PPG technology through Mio sport watch but also profiting from another concurrent study using a PPG based in-ear sensor (next section). This set-up was advantageous because we could finally perform a head-to-head comparison between our recordings and similar recording techniques and then weigh the pros and cons.

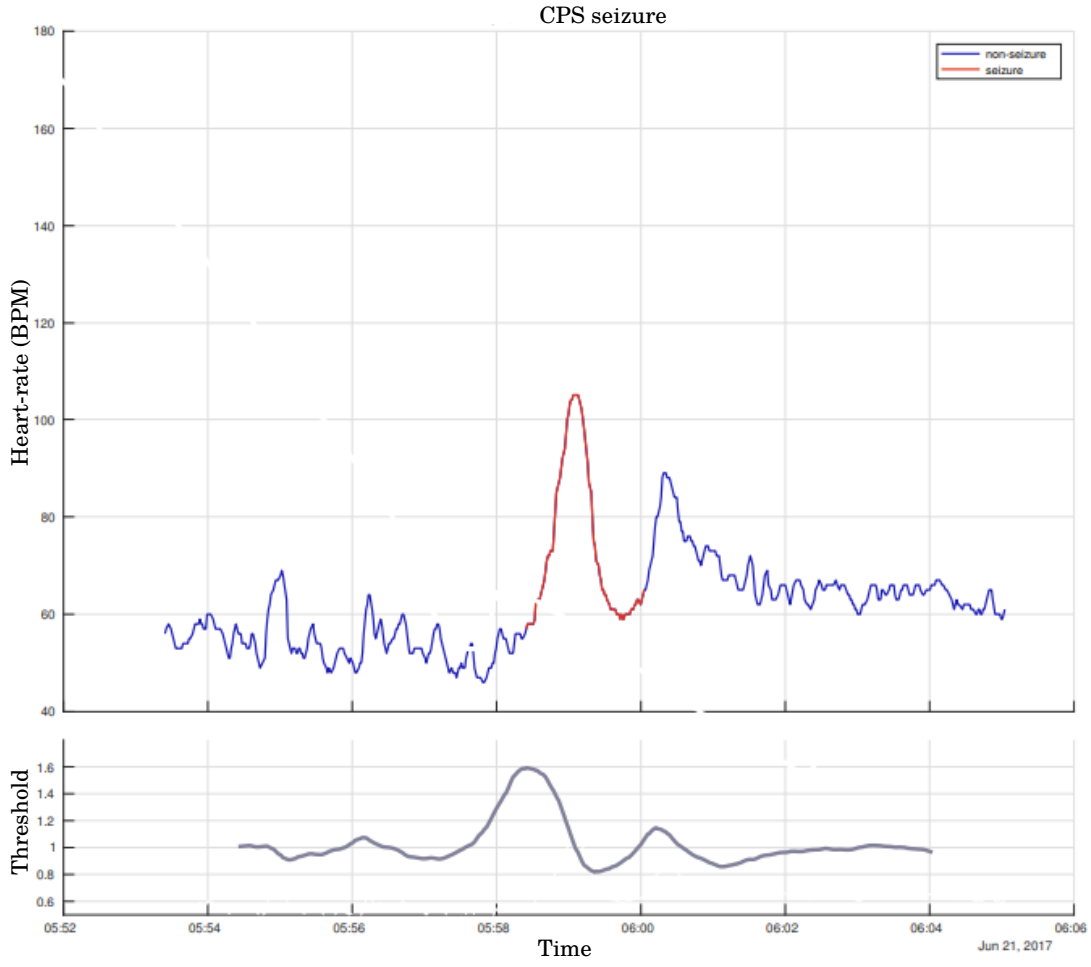


FIGURE 5.3: Mio recording and seizures zoom-in. The upper panel represents an instance of heart-rate during seizure time. The lower panel corresponds to consecutive fold change of heart-rate of the upper panel's data.

As in the previous phases, we obtained our data from patients who were registered at the Epileptology Department at the University of Bonn Medical Center and were recorded using clinical EEG and ECG. There are two main reasons for this. First, the ground-truth ECG had to be determined for checking the accuracy of the PPG recording. Second, the exact onset/offset of the seizures could be obtained from the EEG recordings and with the assistance of epilepsy experts at the clinic. Having a fairly large number of seizure recordings, I was then able to construct a machine learning model for PPG data in the same manner as the ECG in the previous chapter.

The collected data can then be used as raw material for feature extraction. Unlike ECG, in order to measure HRV features, the RRIs should be reconstructed from the heart-rate (HR). That is, for any particular time point for which there is a heart rate, the data must be interpolated so that we have a HR value for every single second. Then, the

time required to go from an R peak to the other can be measured as:

$$RRI = (\frac{HR}{60\,sec}) \times 1000\,ms$$

It should be stressed that the measured RRIs are not as accurate as ECG since PPG is the result of applying FFT in a time window. Hence, abrupt changes in actual RRI may not have been reflected accurately in PPG-based RRI.

As with other phases, I aimed at building a reference model for seizure/non-seizure events. To do so, I once again utilized machine learning techniques. For a machine learning task, typically, measuring features from raw data is required. From this point on, I measured the HRV features the same way that I measured them for HRV on ECG in chapter 4. Similar to ECG, 65 HRV features in total were calculated from PPG data.

Next, I modeled the data with an RF classifier using the first half of data (from patient 1 to 50). I then tested the remaining 50 patients against the trained model.

5.2.3 Results

Table 5.1 shows the performance of using Mio sport watch to monitor seizures. It can be observed easily that the performance of the PPG sensor shows a significant drop compared to the ECG study. It should be mentioned here that the ECG recording of the same group of patients with same classification settings (reported in chapter 4) shows a significant difference.

Performance metric	Mio PPG	ECG
Sensitivity	12.66 %	39 %
Precision	9.30 %	73 %
F_1 score	10.72 %	51 %
False-alarm per day	6.2	0.4

TABLE 5.1: MIO watch PPG seizure detection performance. Compared to the results of ECG in chapter 4, a significant drop in performance can be observed. It is main due to the nature of PPG pulse, which averages out abrupt heart rate changes from the signal.

This fact led us to check for an alternative PPG based seizure detection system with patients' feedback. Ergo, we extended the study in two directions (see them later in sections: App development; Apple watch extension).

5.2.4 In-ear sensor

As mentioned earlier in this chapter, we have run our PPG sport watch study in parallel with another PPG based study called *Epitect*. The difference was that in Epitect, an in-ear PPG sensor was used to get the heart rate activity (see image 5.4). The sensor is developed in a company called Cosinus [20] in Munich Germany. The sensor was able to record and transmit the raw PPG signal (not only the pulse) as well as temperature and accelerometery data to a mobile device through Bluetooth. The main advantage of this sensor was to obtain raw PPG signal compared to Mio watch which only pulse data was available. Raw PPG signal contains also information about blood pressure. It has been shown in a work from our group that blood pressure changes significantly during seizure time [48]. The disadvantage of the sensor is its small battery life (6 hours) and wearing comfort problems (in patients view).

FIGURE 5.4: Cosinuss one in-ear sensor [20]. The sensor can measure PPG, accelerometry and temperature and transmit it in real-time to an external receiver. The advantage of this device was to provide raw PPG, which contained SPO_2 information (oxygen saturation). The sensor had but limitations in battery life as it had to be changed every 6 hours.

The details of this project has not yet been published but according to preliminary results, it appears to have better seizure detection performance compared to the Mio watch study but still inferior to its corresponding ECG recording. I have developed and shared some codes for feature extraction for the developers of this project in the early stage of the project and is partially being used in their current work.

5.2.4.1 App development

As shown above, the results of Mio watch PPG study are not as promising as what we observed in the ECG analysis. We could still increase the sensitivity, but at the cost of lowering precision. One way to tackle this problem is to involve the patients in the process. Therefore, I developed an app for Android smart phones to alert the patient as soon as a suspicious change in heart rate activity occurs. The patient receives a notice

after the incident and then indicates whether or not a seizure actually occurred. The real seizures can then be collected and electronically transmitted to a medical repository.

Below are screenshots of the developed Android app:

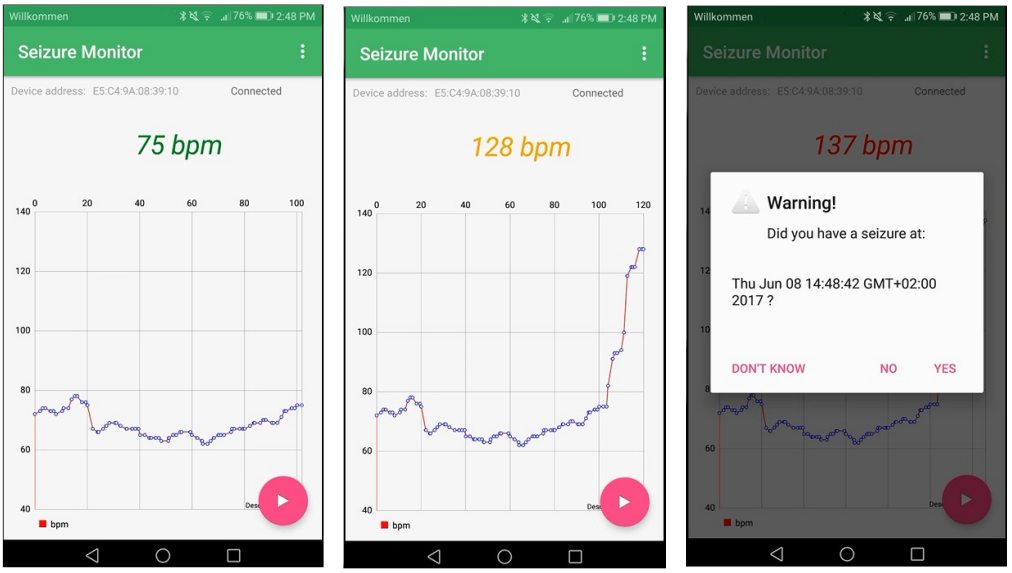


FIGURE 5.5: Mio watch [82] monitoring Andoird app. The app runs in background of the cell phone and pops up with an visual and auditory alarm as early as a suspicious heart rate changes is observed. The patients has a chance to answer whether he/she has had a seizure or not. In case the patient is not conscious or forgets to answer the question, the warning will pop up more until it is answered. The app also has an adjustable threshold to adopt the events based patient's physiology and activities.

We performed a pilot study in the clinic to check the feasibility of this solution. We realize that some patients prefer not to be disturbed by the alarm especially during the sleep.

The setting is a ready to use as an at-home solution for patient with epilepsy.

5.3 Apple watch extension

With the emergence of Apple Watch 3 and reports showing cutting edge performance in the field of communication, battery life and PPG, we broadened our project to test whether the new Apple device has an edge over former PPG recordings such as Mio. The concept, however, is a bit different than the Mio project. Since the Apple watch is a smart watch, it can be programmed internally and react to changes in heart rate [almost] autonomously (see figure 5.6).

In order for it to function, we have let developed an app to track the patients heart rate changes. As with the other concepts, this app must not only monitor, but also inform

FIGURE 5.6: Apple watch. Since Apple watch could be used as a portable mean to interact with a patient, we have employed it check for patients' feedback, to express whether or not a seizure happened. The watch could record continuously for 21 hours.

the patient of irregular activity and instantly request seizure confirmation. Since the app benefits from the Apple Health Kit, we developed five different modes of heart rate activity for evaluation:

- Stationary mode
- Walking mode
- Running mode
- Cycling mode
- Driving mode

Figure 5.7 represents screenshots of the Apple watch app :

The system works based on individual predefined thresholds for heart rate changes in the above mentioned modes. That is, for every activity (stationary, walking, running, cycling, driving), a threshold for heart rate fold change can be set. As soon as the heart rate change exceeds the threshold, an alarm will pop on the screen of the smart watch, asking the patient to answer whether or not, he or she experienced a seizure. There is also a third option, "I do not remember", which if answered, the alert message will pop again to ask for confirmation. The report of seizure activity is then transmitted through a paired iPhone to an email address of a caregiver.

We could not unfortunately employ any machine learning technique to help the system. The reason was because of Apple's health kit which does not give any internal access to the PPG sensor. The heart rate will be delivered to the app based on the preference of health kit and could be delayed for a long time (10 to 15 seconds). On that account,

FIGURE 5.7: Apple watch epilepsy app. This app was developed to popup and ask patients about the occurrence of a seizure as soon as a suspicious heart rate change is detected. Due to very low time resolution of the signal ($0.1Hz$), machine learning could not be applied for detecting suspicious activities. Instead, five different threshold values were used to set an alarm for different user activities. (stationary, walking, running, cycling, driving). The patient then responds to the alarm by telling whether or not a seizure took place. The patient's report could be sent to a caregivers' email by though the accompanied app from the linked iphone.

we could not calculate any HRV feature and therefore, we had to rely solely on the thresholds and the patient's feedback.

Table 5.2 represents the performance of the seizure detection system using Apple watch 3. It can be observed that even though the system is based on human feedback, it still lags behind ECG based seizure detection systems. Please note that in the results, instead of the number false alarm per day, the total number of alarms is reported.

Performance metric	Value
Sensitivity	14.05 %
Precision	24.28 %
F_1 score	17.80 %
~~False~~ alarm per day	11.4

TABLE 5.2: Apple watch PPG seizure detection performance. Even though the patient were involved in the process, the results are not promising. Inadequate attention of patients to the alarm caused a significant drop to precision. In addition, setting higher threshold values, decreased the sensitivity.

5.3.1 Conclusion

The idea behind using Apple watch was to rely on human feedback instead of machine learning. Theoretically, to decrease the number of false alarms human feed back could be plugged in. Our results shows however the reverse. We have investigated the reasons and came to the following points:

- The patients did not like it to be disturbed every other minute, and consequently, the thresholds had to be set higher for their convenience (1.3 or 1.4). Higher threshold means missing most of the seizures and therefore, lower sensitivity.
- Although the the threshold was set to a higher value, the watch will frequently pops alarms (1 or 2 time per hour). This still annoyed the patients and in return, they did not payed adequate attention to the shown messages, only presses randomly on the screen to shut down the watch.

The results of Apple watch study corroborates the work of [37, 58, 65], in which the inaccuracy of the patients self seizures reporting is widely discussed.

Table be low present a performance comparison of all five seizure detection studies presented in chapter 4 and 5. It can be observed that ECG based seizure detection have a clear edge over PPG based seizure detection systems.

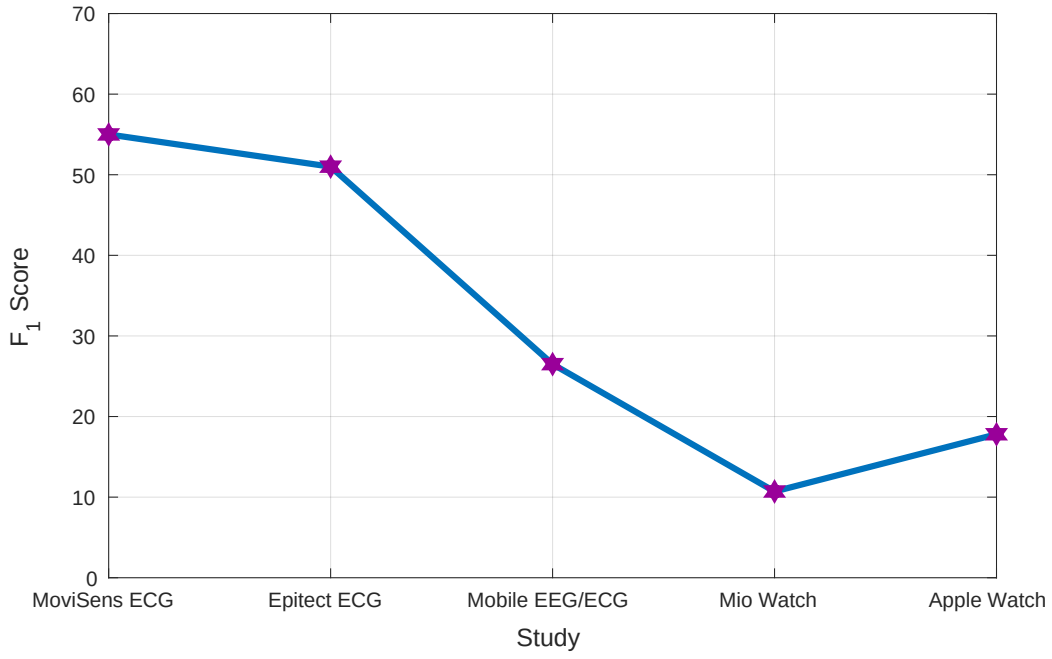


FIGURE 5.8: All seizure detection studies performance comparison. The performance of each seizure detection dataset is presented based on F_1 score. It can be observed that ECG based seizure detection systems showed better performance compared to PPG based seizure detectors.

This brought us to the conclusion that using PPG should be accompanied with machine learning to reduce the number of false alarms. For using machine learning, we require the raw PPG data on hand-held wearable devices. On this account, we started off a new

project to investigate how a PPG recording with raw PPG signal can help detecting the seizures. The recording was going on at the time of this writing.

5.4 Summary

In chapter 4, a framework for detecting seizure based on biomarkers and in particular based on ECG was developed. We have learned from the work that ECG could be used for clinically meaningful seizure detection.

Although portable ECG devices are available in today's market, we were interested to look for more affordable and easy to wear solutions for the problem of seizure detection. In this chapter I addressed the feasibility of using photoplethysmography (PPG) available on sport/smart watch for detecting seizures. PPG is used to track an individual's pulse.

For this sake, we have collected multiple recording dataset with different PPG recording devices. First we have collected data from 100 patients with Mio sport watch recording. I have split the dataset into two halves of 50 patients and applied the principals concluded in the last chapter for ECG seizure detection on PPG. The analyses and the results showed that PPG carry less information compared to ECG and therefore, is less accurate than ECG to detect all types of seizures.

We have continued the work by employing Apple watches. Apple watch is a smart watch and gives us a portable and wearable platform for seizure tracking. However, Apple watch could not provide any PPG recording with reasonable time resolution. For this reason machine learning is not applicable on it. We have chosen the Apple watch to give people with epilepsy an interface to decide on seizures themselves. PPG was used to set an alarm for the patients, and they responded accordingly to it. We designed this phase to compare the performance of patients self-verification with the performance of automatic PPG based seizure detection in Mio watch recording.

All in all, we have learned that that machine learning techniques should be used to improve the performance of seizure detection systems, either on ECG or on raw PPG to be able to track and record and report all types of seizure, and ultimately, increase the quality of life for patients with epilepsy.

Appendix A

Working memory and fMRI

As described in chapter 1 and chapter 3, machine learning could also be applied on fMRI data in order to decode complex brain activities.

In the course of this thesis, I have also applied machine learning techniques on functional Magnetic Resonance Imaging (fMRI) data. Similar to the paradigms presented in chapter 3, performing a memory task in the MRI machine requires also a paradigm design. The paradigm is presented to the participant while lying in the MRI scanner through an optical fiber monitor and air buttons (no metal object is allowed inside the scanner as it otherwise poses a life danger).

The participants were presented with eight classes of images (see image A.1) and they had to give their judgment in two upcoming sessions, whether or not, an image were previously presented. The first session is called the *learning/encoding* session and the next two sessions are called *retrieval* sessions.

1. Beers 2. Roadsigns 3. Leaves 4. Cheeses 5. Dogs 6. Faces 7. Door Knobs 8. Houses

FIGURE A.1: fMRI classes. Images from eight categories were presented to the participant and later on two occasions, they were probed again with images asking to tell whether they have seen an image before or not.

In fMRI recordings, the activity of the brain is measured spatially. The blood oxygen in a magnetized environment can be oriented and be senses in magnetic field receiver sensors. More blood oxygen level indicates more brain activity.

Similar to the concept of pixels in $2D$ images, in fMRI recording, we have the concept of *voxels*. Voxels are the minimum separable buckets of data in the $3D$ images resulted from fMRI recording. In fMRI data recording, the recording machine is programed to scan the whole brain sequentially, and therefore, it gives a sequence of $3D$ images, each of which reflects the brain active voxels at a particular time ($4D$ in total).

Different people have different brain volume. To be able to compare the brain activities of different people, the $3D$ brain recordings of the participants must be mapped and registered to a standard size brain. This standard size is a conventional brain size obtained from the average MRI images of the brain of numerous people.

The time resolution of fMRI images is much lower than of EEGs ($\sim 1Hz$). This is due to the time it takes for the scanner to scan multiple surfaces of a $3D$ image.

Having the $4D$ data preprocessed, I applied machine machine learning techniques to check for the pattern differences between all of the observed stimuli images. Initially, I trained and tested on the encoding session, when the learning take place. Image A.2 shows the classification results of the object classification of brain data. There were near a million voxels in an image of the brain. Voxels could be used directly as feature. Similar to the techniques presented in chapter 3, I used ANOVA to preselect the voxels which show best separability (8 way ANOVA). One thousand voxels were selected and then fed to an SMO classifier.

Alternatively, the classifier could be trained on the encoding phase and then tested on the retrieval session. Image A.3 shows the classification accuracy of such settings tested on the first retrieval sessions. This test can show whether or not, same network of voxels used for the encoding and the retrieval phases.

Given that a network of voxels pass the pattern classification significance test, voxels can be mapped into their belonging structure in the brain to investigate the structural relations between brain activities and brain organs.

The learned model could ultimately be used to classify random patterns during resting state, a phase in which no particular task is given to the participant and their mind is let free. Their mind can subconsciously rehearse the recently seen objects. Any activation similar to the modeled brain activity can indicate a task relevant memory activity.

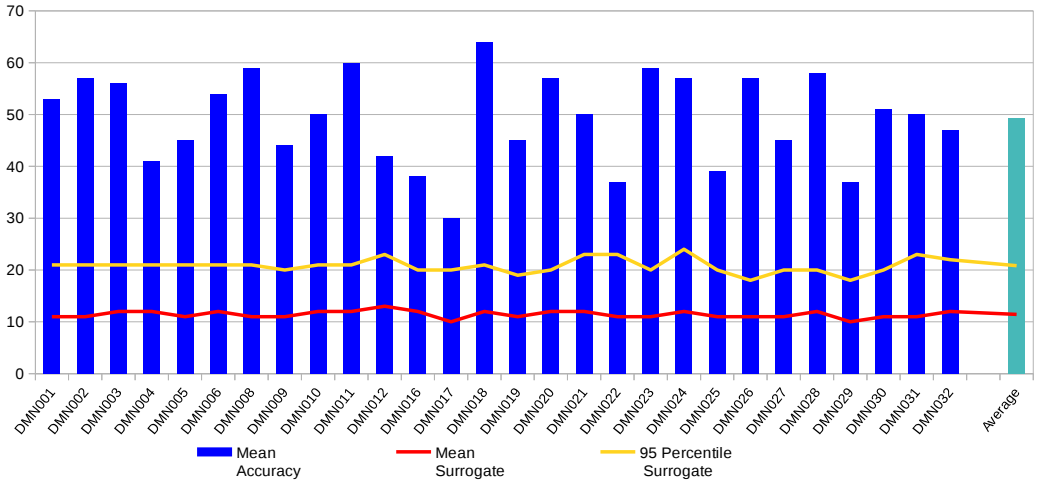


FIGURE A.2: fMRI classification results on encoding. fMRI classification results on encoding. The data after preprocessing is gone through the feature selection procedure, in which 1000 voxels chosen among one million. The selected voxels were then fed to the SMO (SVM) classifier. The blue bars show the classification accuracies of 28 participants. The red line represents the average accuracy and the yellow bar shows the 95 percentile of the surrogate and in another word the *confidence level*.

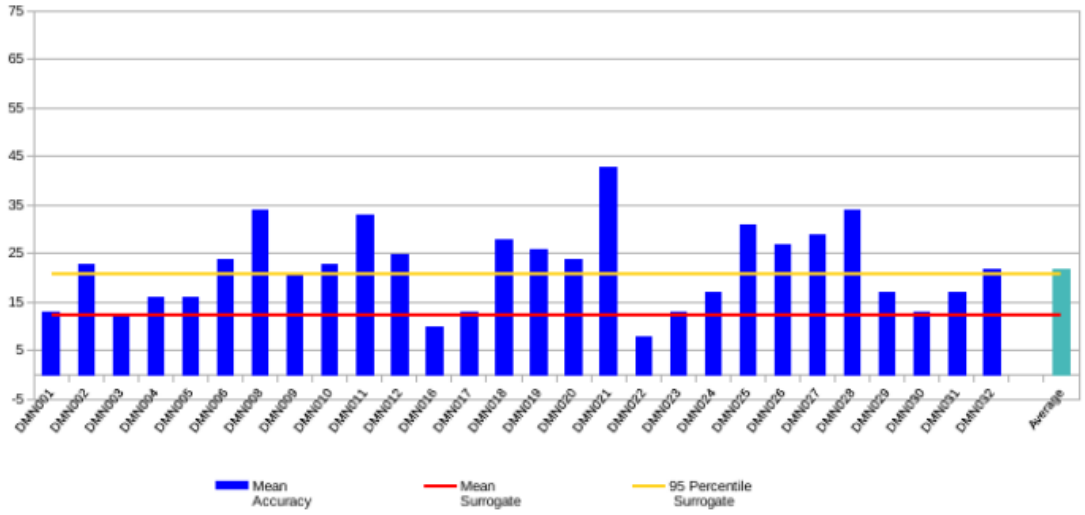


FIGURE A.3: fMRI classification results 2. fMRI classification results 2.

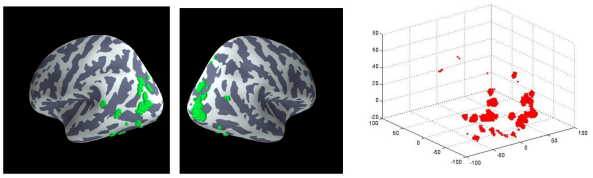


FIGURE A.4: fMRI active voxel . Those voxels in the fMRI which shows task related activity, can be mapped to their corresponding brain structure to investigate the relation between brain segments and types of mental processing.

Appendix B

Poster presented at OHBM conference.

This poster is presented in the "Organization of Human Brain Mapping" (OHBM) conference in 2014, Hamburg, Germany.

Multivariate analyses on intracranial EEG task characteristics during human working memory

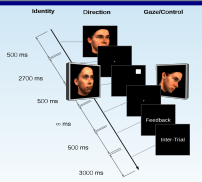
Amirhossein Jahanbekam[1,2], Marcin Leszczynski[1], Jürgen Fell[1], Nikolai Axmacher[1,2]
[1] *Dept. of Epileptology, University of Bonn, Germany*
[2] *German Centre for Neurodegenerative Diseases (DZNE), Bonn, Germany*

Introduction

- Different types of visual information are processed in different pathways. While object identity is represented in the ventral visual stream, spatial information is represented in the dorsal visual stream [1].
- However, representations may not only rely on such local processing but also on distributed networks that can be investigated using multivariate pattern classification algorithms [2].
- Here, we used intracranial EEG recordings to investigate the time-frequency characteristics underlying representations of object identity and spatial direction in a delayed matching to sample paradigm.

Material & Methods: Paradigm

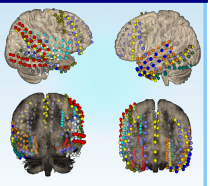
- The paradigm consisted of three consecutive blocks ("Identity" block, "Gaze direction" block, and "Control" block), which were presented in a random order [3].
- Either the identity or the gaze direction or no information of a face stimulus had to be maintained.



Delayed matching to sample paradigm composed of 3 blocks: Maintaining face identity, face direction, and control condition.

Material & Methods: Subjects & Data

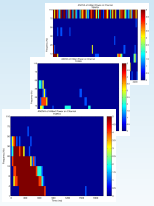
- 12 patients (6 female; mean age±SD: 34±12.5 years) had been implanted with intracranial electrodes for diagnostic purposes. Depending on the suspected ictal onset zone, patients had been implanted with subdural strip, grid electrodes and/or depth electrodes.
- All data was sampled at 1000Hz, referenced to linked mastoids.
- Electrodes from the seizure onset zones and noisy electrodes were removed from further analysis.



Implantation schema. Each color represents an individual patient.

Material & Methods: Multivariate Analyses

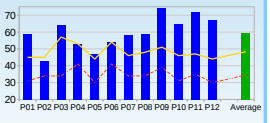
- Multivariate pattern classification analyses were used to identify distributed activity patterns which differentiated between the three task conditions.
- The trial data were segmented and filtered in 8 common EEG frequency bands (delta [1-4Hz], theta [4-8Hz], alpha [8-12Hz], beta1 [12-20Hz], beta2 [20-30Hz], gamma1 [30-50Hz], gamma2 [50-75Hz], gamma3 [75-110Hz]).
- We extracted frequency-specific power values by Hilbert transformation. Informative features at time-frequency-electrode points (30ms bins) were extracted using ANOVA and fed into either SVM or random forest classifier [5]. Classification was evaluated using 5-fold cross-validation.



Feature selection in time-frequency-electrode space.

Results: Pattern Classification Performance

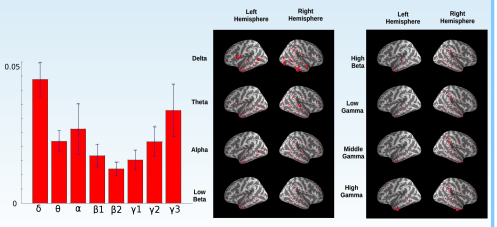
- Classification performance compared between empirical data and label-shuffled surrogate data.
- Classification result of 12 patients:
for 10 subjects classification performance was significantly above chance level and for 2 it was marginally significant.



Classification results during the maintenance period using random forest classifier across the 3 conditions.

Results: Important Features Distribution

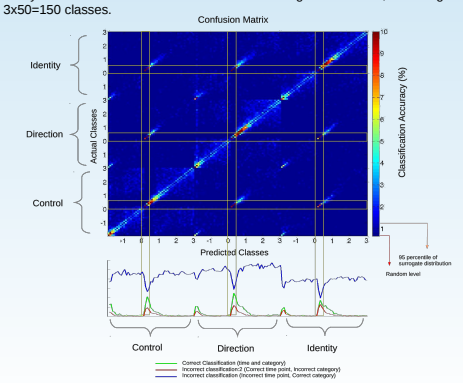
- Using random forest classifier, we are able to rank the importance of features in the classification process.
- The frequency specific feature importance is shown for all 8 frequency bands.



Frequency specific feature importance.

Results: Time Resolved Classification

- To investigate classification across time, a temporal classification schema is considered.
- Every 100ms time bin is considered as a single condition, resulting in 3x50=150 classes.



Confusion matrix illustrating time resolved classification across different conditions.

Summary

- We found that different task demands in a visual working memory task are associated with significantly altered distributed representations of (mainly) delta and high gamma-band activity across the brain.
- Time-resolved classification revealed highest accuracy during stimulus presentation. Incorrect classifications had different causes during stimulus presentation as compared to during the maintenance period.

References

(1) Goodale MA, Milner AD (1992). Separate visual pathways for perception and action. Trends Neurosci. 15 (1): 20–5.
(2) van Gerven M, Chao Z, Heskes T (2012). On the decoding of intracranial data using sparse orthonormalized partial least squares. J Neural Eng. 9(2):026017.
(3) Jokisch D, Jensen O (2007). Modulation of gamma and alpha activity during a working memory task engaging the dorsal or ventral stream. J Neurosci 27(12):3244-51.
(4) Lee SH, Kravitz DJ, Baker CI (2013). Goal-dependent dissociation of visual and prefrontal cortices during working memory, Nature Neuroscience 16(8): 997–999.
(5) Breiman L, Schapire E (2001). Random forest. Machine Learning 45(1): 5-32.

Amirhossein.Jahanbekam@ukb.uni-bonn.de

FIGURE B.1: Poster presented at Organization of Human Brain Mapping (HBM) conference.

Appendix C

Poster presented at DGfE conference.

This poster is presented at "Deutschen Gesellschaft für Epileptologie" (DGfE) conference in 2015, Dresden, Germany.

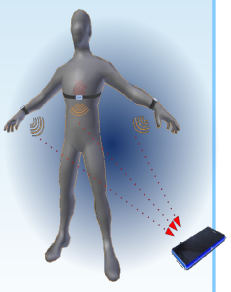
universitätsklinikumbonn

A multimodal, non-EEG based approach to detect epileptic seizures

Amirhossein Jahanbekam[1], Jan Baumann[1], Christian Bauckhage[2], Christian E. Elger[1], Rainer Surges[1]
[1] *Dept. of Epileptology, University of Bonn, Germany*
[2] *Fraunhofer Institute for Intelligent Analysis and Information Systems, Sankt Augustin, Germany*

Introduction

- Epilepsy patients are advised to keep a diary documenting all seizures as soon as they take place. A considerable proportion of patients, however, is not aware of their seizures or forgets them, so that the seizure diaries are very unreliable tools [1]. Automatic seizure detection devices have previously been tested especially in the context of predominant ictal motor signs [2,3].
- Here, we aim at developing a wearable multisensory-system to automatically **detect and register all types of seizures**. To this end, we used an ECG sensor alongside with 3 acceleration sensors all embedded in 3 comfortable-to-wear sensor units [4].



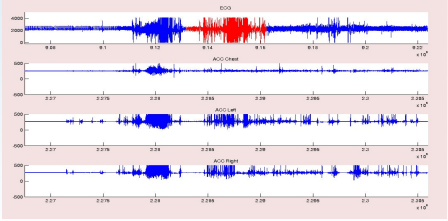
Material & Methods: Subjects, Sensors, and Data

- We have investigated people with refractory focal epilepsy undergoing presurgical video-EEG monitoring at the Department of Epileptology in Bonn. Patients were asked to wear 3 sensor units [2], one attached to the chest and the others fixed on both wrists.
- Each sensor could record the acceleration along 3D axes, while the chest sensor was also able to measure ECG signals.
- After synchronizing the signals across different sensor units, we extracted 180 features for epochs of 10 minutes long across all sensors. Each window was labeled as seizure/non-seizure based on expert-review of electroclinical data. Recordings being compromised by movement or EMG artifacts were excluded after visual inspection.

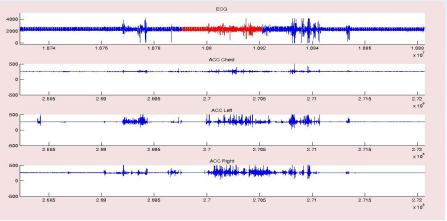
Sensor units. One attached to the chest and the other two on both wrists.

Material & Methods: Multivariate Analyses and Pattern Classification

- We have monitored 35 patients (mostly with temporal lobe epilepsy; age 38±14 years) during a period of 68 days. A total of 43 seizures (5 simple partial, 33 complex partial and 5 generalized tonic-clonic seizures) were included in this study.
- To acquire event-based seizure detection, we exclusively considered epochs in which the heart rate (HR) during a period of 2 minutes increased rapidly and significantly (at least a 1.2-fold increase in HR), which finally provided a total number of about 6000 event windows. Furthermore, to train a classifier on the data, we considered a 5-fold cross-validation scheme in which windows of the same seizure were used either only in training-phase or only in test-phase.
- Two examples of online seizure-detection events are illustrated here:



Truly detected as a seizure case



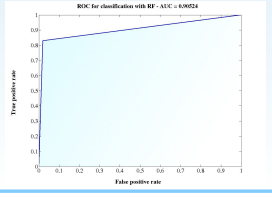
Truly detected as a non-seizure case

Results: Sesitivity & Specificity

- We evaluate the performance of the system by measuring the sensitivity and specificity:

Sensitivity	**86 %**
Specificity	**97.5 %**

- The performance can also be depicted as "area under the curve" in ROC. The closer the value to 1 is, the better the system performance in both terms of sensitivity and specificity.
- The **ROC** of our approach amounts to **0.905**.



Summary

- Our preliminary data suggest that the combination of various ECG and accelerometry features allows automatic detection even of complex-partial seizures with a promising sensitivity and specificity. This may lead to the development of a wearable tool that improves accuracy of seizure counting.
- In a second step, the sensors' recordings will be transferred to a smart phone to allow interaction with the patient to further improve the performance of the seizure detection. The result will be kept and used for medical purposes.

References

[1] Hoppe C, Poepel A, Elger CE. Arch Neurol 2007;64:1595-9.
[2] Lockman J, Fisher RS, Olson DM. Epilepsy Behav 2011;20:638-41.
[3] Dalton A, Patel S, Chowdhury AR, et al. IEEE Trans Biomed Eng 2012;59:3204-11.
[4] http://www.movisens.com/de/produkt

Contact::
Amirhossein.Jahanbekam@ukb.uni-bonn.de or
Rainer.Surges@ukb.uni-bonn.de
This project is funded by the Marga und Walter Boll-Stiftung, Kerpen

FIGURE C.1: Poster presented at DGfE conference.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] AlphaGo. www.alphagomovie.com, 2017.

[3] Dirk Matthias Altenmüller, Andreas Schulze-Bonhage, Christian E. Elger, and Rainer Surges. Local brain activity persists during apparently generalized postictal EEG suppression. *Epilepsy and Behavior*, 62:218–224, 2016.

[4] Ulrich Altrup, Christian E Elger, and Markus Reuber. *Epilepsy Explained, a book for people who want to know more about epilepsy.* Medicine Explained Publishing, 2005.

[5] Hanna Ansakorpi, J T Korpelainen, H V Huikuri, U Tolonen, V V Myllylä, and J I T Isojärvi. Heart rate dynamics in refractory and well controlled temporal lobe epilepsy. *Journal of neurology, neurosurgery, and psychiatry*, 72(1):26–30, jan 2002.

[6] Jan Baumann. Reconstructing human motion. *Bonn University, PhD thesis*, (HBZ-ID:HT019733402), 2018.

[7] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society (London)*, 53:370–418, 1763.

[8] Sándor Beniczky, Isa Conradsen, and Peter Wolf. Detection of convulsive seizures using surface electromyography. *Epilepsia*, 59(December 2017):23–29, 2018.

[9] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, sep 1975.

[10] Gary G Berntson, David L Lozano, and Yun-Ju Chen. Filter properties of root mean square successive difference (RMSSD) for heart rate. *Psychophysiology*, 42(2):246–52, mar 2005.

[11] Jonathan Bidwell, Thanin Khuwatsamrit, Brittain Askew, Joshua Andrew Ehrenberg, and Sandra Helmers. Seizure reporting technologies for epilepsy treatment: A review of clinical information needs and supporting technologies. *Seizure*, 32:109–17, nov 2015.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] James M Bower. *20 Years of Computational Neuroscience*. Springer Series in Computational Neuroscience, 2013.

[14] Bob Bramson, Ole Jensen, Ivan Toni, and Karin Roelofs. Cortical Oscillatory Mechanisms Supporting the Control of Human SocialEmotional Actions. *The Journal of Neuroscience*, 38(25):5739–5749, 2018.

[15] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[16] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*, volume 1. MIT Press, 2006.

[17] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[18] Diana Cogan, Javad Birjandtalab, Mehrdad Nourani, Jay Harvey, and Venkatesh Nagaraddi. Multi-Biosignal Analysis for Epileptic Seizure Monitoring. *International journal of neural systems*, 26(6):1650031, 2016.

[19] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[20] Cosinuss. Cosinuss One in-Ear sensor. *https://www.cosinuss.com/de/produkte#cosinuss-one*, 2017.

[21] Coursera. Deep-learning, nano degree. Onlie lecture, 2017.

[22] Nelson Cowan. What are the differences between long-term, short-term, and working memory? Nelson. *NIH Public Access*, 6123(07):323–338, 2009.

[23] Anthony Dalton, Shyamal Patel, Atanu Roy Chowdhury, Matt Welsh, Trudy Pang, Steven Schachter, Gearóid Ólaighin, and Paolo Bonato. Development of a body sensor network to detect motor patterns of epileptic seizures. *IEEE Transactions on Biomedical Engineering*, 59(12 PART2):3204–3211, 2012.

[24] Rolando Del Maestro. Leonardo da Vinci: The search for the soul. *Journal of Neurosurgery*, (December 1998), 1998.

[25] Li Deng and Dong Yu. *Deep Learning: Methods and Applications*, volume 7. 2013.

[26] Marlene Derner, Amirhossein Jahanbekam, Christian Bauckhage, Nikolai Axmacher, and Juergen Fell. Prediction of memory formation based on absolute electroencephalographic phases in rhinal cortex and hippocampus outperforms prediction based on stimulus-related phase shifts. *The European journal of neuroscience*, 47(7):824–831, apr 2018.

[27] Emotiv EEG Device. Emotiv EPOC Headset, 2009.

[28] MindWave EEG Device. MindWave, NeuroSky EEG Bio Marker, A Mobile Headset to Experience Brainwave, 2014.

[29] Adele Diamond. Executive functions. *Annual review of psychology*, 64:135–168, 2013.

[30] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[31] Jeffrey L Elman. Finidng structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[32] Empatica Inc. www.empatica.com/en-eu/research/e4/. *Empatica E4*, 2018.

[33] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542:115, jan 2017.

[34] Juergen Fell and Nikolai Axmacher. The role of phase synchronization in memory processes. *Nature reviews. Neuroscience*, 12(2):105–118, 2011.

[35] Juergen Fell, Rüdiger Köhling, Thomas Grunwald, Peter Klaver, Thomas Dietl, Carlo Schaller, Albert Becker, Christian E Elger, and Guillén Fernández. Phase-locking characteristics of limbic P3 responses in hippocampal sclerosis. *NeuroImage*, 24(4):980–9, feb 2005.

[36] Juergen Fell, Eva Ludowig, Timm Rosburg, Nikolai Axmacher, and Christian E Elger. Phase-locking within human mediotemporal lobe predicts memory formation. *NeuroImage*, 43(2):410–9, nov 2008.

[37] Victor Ferastraoaru, Daniel M Goldenholz, Sharon Chiang, Robert Moss, William H Theodore, and Sheryl R Haut. Characteristics of large patient-reported outcomes: Where can one million seizures get us? *Epilepsia open*, 3(3):364–373, sep 2018.

[38] Lluís Fuentemilla, Will D Penny, Nathan Cashdollar, Nico Bunzeck, and Emrah Düzel. Theta-coupled periodic replay in working memory. *Current biology : CB*, 20(7):606–12, apr 2010.

[39] Koichi. Fujiwara, Miho. Miyajima, Toshitaka. Yamakawa, Erika. Abe, Yoko. Suzuki, Yuriko. Sawada, Manabu. Kano, Taketoshi. Maehara, Katsuya. Ohta, Taeko. Sasai-Sakuma, Tetsuo. Sasano, Masato. Matsuura, and Eisuke. Matsushima. Epileptic Seizure Prediction Based on Multivariate Statistical Process Control of Heart Rate Variability Features. *IEEE Transactions on Biomedical Engineering*, 63(6):1321–1332, 2016.

[40] Chael S Gazzaniga. *The Cognitive Neurosciences*. 1956.

[41] Jonas Gehring and Yann N Dauphin. Convolutional Sequence to Sequence Learning. 2016.

[42] M. a. Goodale and a. D. Milner. Separate visual pathways for perception and action. [Review] [61 refs]. *Trends in Neurosciences*, 15(I):20–5, 1992.

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[44] Patrick Grim. Philosophy of Mind: Brains, Consciousness, and Thinking Machines. *The Great Courses (TTC)*, 2012.

[45] CHARLES G. GROSS. The Brain. *The Neuroscientist*, 1:245–250, 1995.

[46] Umut Güçlü and M. a. J. van Gerven. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream. *Journal of Neuroscience*, 35(27):10005–10014, jul 2015.

[47] Xiaoxiao Guo, Honglak Lee, and Richard Lewis. Action-Conditional Video Prediction using Deep Networks in Atari Games. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2015.

[48] Kevin G. Hampel, Amirhossein Jahanbekam, Christian E. Elger, and Rainer Surges. Seizure-related modulation of systemic arterial blood pressure in focal epilepsy. *Epilepsia*, 57(10):1709–1718, 2016.

[49] James V Haxby. Multivariate pattern analysis of fMRI: the early beginnings. *NeuroImage*, 62(2):852–5, aug 2012.

[50] James V Haxby, Andrew C Connolly, and J Swaroop Guntupalli. Decoding Neural Representational Spaces Using Multivariate Pattern Analysis. *Annual review of neuroscience*, jun 2014.

[51] James V Haxby, M. Ida Gobbini, Maura L. Furey, Alumit Ishai, Jennifer L. Schouten, and Pietro Pietrini. Distrubuted and Overlapping Representations of Face and Objects in Ventral Temporal Cortex. *Science*, 293(5539):2425–2430, sep 2001.

[52] C. Henle, Markus Raab, Joacir Graciolli Cordeiro, S. Doostkam, A. Schulze-Bonhage, T. Stieglitz, and J. Rickert. First long term in vivo study on subdurally implanted Micro-ECoG electrodes, manufactured with a novel laser technology. *Biomedical Microdevices*, 13(1):59–68, feb 2011.

[53] Suzana Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3:31, 2009.

[54] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006.

[55] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[56] Marlene Höhne, Amirhossein Jahanbekam, Christian Bauckhage, Nikolai Axmacher, and Juergen Fell. Prediction of successful memory encoding based on single-trial rhinal and hippocampal phase information. *NeuroImage*, 139:127–135, 2016.

[57] Christian Hoppe, Mieke Feldmann, Barbara Blachut, Rainer Surges, Christian E. Elger, and Christoph Helmstaedter. Novel techniques for automated seizure registration: Patients' wants and needs. *Epilepsy and Behavior*, 52:1–7, 2015.

[58] Christian Hoppe, Annkathrin Poepel, and Christian E Elger. Epilepsy: accuracy of patient seizure counts. *Archives of neurology*, 64(11):1595–9, nov 2007.

[59] Anna Jafarpour, Aidan J. Horner, Lluís Fuentemilla, W. D. Penny, and Emrah Duzel. Decoding oscillatory representations and mechanisms in memory. *Neuropsychologia*, 51(4):772–780, mar 2013.

[60] Amirhossein Jahanbekam, Christian Bauckhage, and Christian Thurau. Age recognition in the wild. *Proceedings - International Conference on Pattern Recognition*, pages 392–395, 2010.

[61] Jesper Jeppesen, Sandor Beniczky, Peter Johansen, Per Sidenius, and Anders Fuglsang-Frederiksen. Using Lorenz plot and Cardiac Sympathetic Index of heart rate variability for detecting seizures for patients with epilepsy. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2014:4563–4566, jan 2014.

[62] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[63] Daniel Jokisch and Ole Jensen. Modulation of gamma and alpha activity during a working memory task engaging the dorsal or ventral stream. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(12):3244–51, mar 2007.

[64] Eric R. Kandel, J. H. Schwartz, and Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill Medical, 4th edition, July 2000.

[65] Frank Kerling, Sonja Mueller, Elisabeth Pauli, and Hermann Stefan. When do patients forget their seizures? An electroclinical study. *Epilepsy & behavior : E&B*, 9(2):281–285, sep 2006.

[66] Wolfgang Klimesch. Alpha-band oscillations, attention, and controlled access to stored information. *Trends in Cognitive Sciences*, 16(12):606–617, 2012.

[67] Thosten Kranz. PyLocator, Application for Localization of EEG-electrodes from MRI-volumes. *http://pylocator.thorstenkranz.de/*, 2014.

[68] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012.

[69] Abhinav V Kurada, Tarun Srinivasan, Sarah Hammond, Adriana Ulate-Campos, and Jonathan Bidwell. Seizure detection devices for use in antiseizure medication clinical trials: A systematic review. *Seizure*, 66:61–69, mar 2019.

[70] J P Lachaux, E Rodriguez, J Martinerie, and F J Varela. Measuring phase synchrony in brain signals. *Human brain mapping*, 8(4):194–208, jan 1999.

[71] P. Laguna, R. Jané, S. Olmos, N. V. Thakor, H. Rix, and P. Caminal. Adaptive estimation of qrs complex wave features of ecg signal by the hermite model. *Medical and Biological Engineering and Computing*, 34(1):58–68, Jan 1996.

[72] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science (New York, N.Y.)*, 350(6266):1332–38, 2015.

[73] Jesús Lázaro, Eduardo Gil, José María Vergara, and Pablo Laguna. Pulse rate variability analysis for discrimination of sleep-apnea-related decreases in the amplitude fluctuations of pulse photoplethysmographic signal in children. *IEEE Journal of Biomedical and Health Informatics*, 18(1):240–246, 2014.

[74] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, 2009.

[75] Sue-Hyun Lee, Dwight J. Kravitz, and Chris I. Baker. Goal-dependent dissociation of visual and prefrontal cortices during working memory. *Biochim Biophys Acta*, (4):395–401, 2013.

[76] Marcin Leszczynski, Juergen Fell, and Nikolai Axmacher. Rhythmic Working Memory Activation in the Human Hippocampus. *Cell Reports*, 13(6):1272–1282, 2015.

[77] Marcin Leszczynski, Juergen Fell, Ole Jensen, and Nikolai Axmacher. Alpha activity in the ventral and dorsal visual stream controls information flow during working memory. *Division, Translational Neuroscience*, pages 1–34, 2017.

[78] Juliana Lockman, Robert S Fisher, and Donald M Olson. Epilepsy & Behavior Detection of seizure-like movements using a wrist accelerometer. *Epilepsy & Behavior*, 20(4):638–641, 2011.

[79] Paulo a Lotufo, Leandro Valiengo, Isabela M Benseñor, and Andre R Brunoni. A systematic review and meta-analysis of heart rate variability in epilepsy and antiepileptic drugs. *Epilepsia*, 53(2):272–82, feb 2012.

[80] Eric Maris and Robert Oostenveld. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, 164(1):177–190, 2007.

[81] Ian H. Witten Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann. The WEKA Data Mining Software: An Update. 11(1), 2009.

[82] Mio Global. Mio Alpha 2 Haertrate Watch. *https://www.mioglobal.com/en-us/Mio-ALPHA-2-Heart-Rate-Sport-Watch/Product.aspx*, 2017.

[83] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei a. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, feb 2015.

[84] Movisens GmbH. Movisens, 2015.

[85] Benedict Shien Wei Ng, Nikos K Logothetis, and Christoph Kayser. EEG phase patterns reflect the selectivity of neural firing. *Cerebral cortex (New York, N.Y. : 1991)*, 23(2):389–98, feb 2013.

[86] Jeanette Norden. Understanding the Brain. *The Great Courses (TTC)*, 2012.

[87] Carina R. Oehrn, Simon Hanslmayr, Juergen Fell, Lorena Deuker, Nico A. Kremers, Anne T. Do Lam, Christian E. Elger, and Nikolai Axmacher. Neural Communication Patterns Underlying Conflict Detection, Resolution, and Adaptation. *The Journal of Neuroscience*, 34(31):10438–10452, 2014.

[88] A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, May 1981.

[89] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, March 1985.

[90] Steven Pinker. *How the Mind Works*. New York, NY: W. W. Norton & Company, 1997.

[91] John C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Advances in kernel methods*, pages 185 – 208, 1998.

[92] Ming-zher Poh, Tobias Loddenkemper, Claus Reinsberger, Nicholas C Swenson, Shubhi Goyal, Mangwe C Sabtala, Joseph R Madsen, and Rosalind W Picard. Convulsive seizure detection using a wrist-worn electrodermal activity and accelerometry biosensor. 53(5):93–97, 2012.

[93] Athi Ponnusamy, Jefferson L B Marques, and Markus Reuber. Comparison of heart rate variability parameters during complex partial seizures and psychogenic nonepileptic seizures. *Epilepsia*, 53(8):1314–21, aug 2012.

[94] Pariya Salami, Maxime Lévesque, Jean Gotman, and Massimo Avoli. A comparison between automated detection methods of high-frequency oscillations (80500 Hz) during seizures. *Journal of Neuroscience Methods Neurosci Methods*, 211(2):265–271, 2012.

[95] Karen Simonyan, Sander Dieleman, Andrew Senior, and Alex Graves. WaveNet: a generative model for raw audio. pages 1–15, 2016.

[96] Steven W. Smith. *Digital signal processing*. 1999.

[97] Rainer Surges and Matthew C. Walker. Peri-ictal heart rates depend on seizure-type. *Seizure*, 19(7):453, 2010.

[98] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[99] Motomi Toichi, Takeshi Sugiura, Toshiya Murai, and Akira Sengoku. A new method of assessing cardiac autonomic function and its comparison with spectral analysis and coefficient of variation of RR interval. *Journal of the Autonomic Nervous System*, 62(1-2):79–84, jan 1997.

[100] Judith van Andel, Constantin Ungureanu, Ronald Aarts, Frans Leijten, and Johan Arends. Using photoplethysmography in heart rate monitoring of patients with epilepsy. *Epilepsy & Behavior*, 45:142–145, 2015.

[101] Marcel A.J. van Gervena, Eric Marisa, Michael Sperlingb, Ashwini Sharanc, Brian Littd, Christopher Andersond, Gordon Baltuche, and Joshua Jacobsf. Decoding the memorization of individual stimuli with direct human brain recordings. *NeuroImage*, 70:223–232, 2013.

[102] Kaat Vandecasteele, Thomas De Cooman, Ying Gu, Evy Cleeren, Kasper Claes, Wim Van Paesschen, Sabine Van Huffel, and Borbála Hunyadi. Automated epileptic seizure detection based on wearable ECG and PPG in a hospital environment. *Sensors (Switzerland)*, 17(10):1–12, 2017.

[103] Mariel Velez, Robert S. Fisher, Victoria Bartlett, and Scheherazade Le. Tracking generalized tonic-clonic seizures with a wrist accelerometer linked to an online database. *Seizure*, 39:13–18, 2016.

[104] Panqu Wang, Vicente Malave, and Ben Cipollini. Encoding Voxels with Deep Learning. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 35(48):15769–71, dec 2015.

[105] Takufumi Yanagisawa, Okito Yamashita, Masayuki Hirata, Haruhiko Kishima, Youichi Saitoh, Tetsu Goto, Toshiki Yoshimine, and Yukiyasu Kamitani. Regulation of motor representation by phase-amplitude coupling in the sensorimotor cortex. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 32(44):15467–75, oct 2012.