A machine learning toolbox for the development of personalized epileptic seizure detection algorithms

Guillaume Saulnier-Comte

Master of Science

Computer Science

McGill University

Montréal,Québec

July 2013

A thesis submitted to McGill University in partial fulfilment of the requirements of the degree of Master of Science

DEDICATION

To my parents, Sylvie and Pierre, my brother, Julien, and my significant other, Solveig, with whom I share these wonderful relations.

ACKNOWLEDGEMENTS

I gracefully thank my supervisor, Joelle Pineau, for the opportunity to get a taste of research early during my undergraduate studies. Her insights, guidance, support and trust during the academic years leading to this work made the experience even more edifying. I am grateful of the support provided by Dr. Avoli's laboratory at the Montréal Neurological Institute, more specifically to Maxime Lévesque, who provided me with the electrocorticographic recordings of rats presented in this thesis. Moreover, I had the honour of being part of the Reinforcement Learning Laboratory, a wonderful workplace source of endless knowledge.

The generous support of the Natural Sciences and Engineering Research Council of Canada and the Tomlinson fellowship enabled me to focus on my research without worrying about my subsistence.

Finally, special thanks go to my family, Solveig and my really good friend Alexandre, for keeping me sane.

ABSTRACT

Epilepsy is a chronic neurological disorder affecting around 50 million people worldwide. It is characterized by the occurrence of seizures; a transient clinical event caused by synchronous and/or abnormal and excessive neuronal activity in the brain. This thesis presents a novel machine learning toolbox that generates personalized epileptic seizure detection algorithms exploiting the information contained in electroencephalographic recordings. A large variety of features designed by the seizure detection/prediction community are implemented. This broad set of features is tailored to specific patients through the use of automated feature selection techniques. Subsequently, the resulting information is exploited by a complex machine learning classifier that is able to detect seizures in real-time. The algorithm generation procedure uses a default set of parameters, requiring no prior knowledge on the patients' conditions. Moreover, the amount of data required during the generation of an algorithm is small. The performance of the toolbox is evaluated using cross-validation, a sound methodology, on subjects present in three different publicly available datasets. We report state of the art results: detection rates ranging from 76% to 86% with median false positive rates under 2 per day. The toolbox, as well as a new dataset, are made publicly available in order to improve the knowledge on the disorder and reduce the overhead of creating derived algorithms.

ABRÉGÉ

L'épilepsie est un trouble neurologique cérébral chronique qui touche environ 50 millions de personnes dans le monde. Cette maladie est caractérisée par la présence de crises d'épilepsie; un événement clinique transitoire causé par une activité cérébrale synchronisée et/ou anormale et excessive. Cette thèse présente un nouvel outil, utilisant des techniques d'apprentissage automatique, capable de générer des algorithmes personnalisés pour la détection de crises épileptiques qui exploitent l'information contenue dans les enregistrements électroencéphalographiques. Une grande variété de caractéristiques conçues pour la recherche en détection/prédiction de crises ont été implémentées. Ce large éventail d'information est adapté à chaque patient grâce à l'utilisation de techniques de sélection de caractéristiques automatisées. Par la suite, l'information découlant de cette procédure est utilisée par un modèle de décision complexe, qui peut détecter les crises en temps réel. La performance des algorithmes est évaluée en utilisant une validation croisée sur des sujets présents dans trois ensembles de données accessibles au public. Nous observons des résultats dignes de l'état de l'art: des taux de détections allant de 76% à 86% avec des taux de faux positifs médians en deçà de 2 par jour. L'outil ainsi qu'un nouvel ensemble de données sont rendus publics afin d'améliorer les connaissances sur la maladie et réduire la surcharge de travail causée par la création d'algorithmes dérivés.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Epilepsy

Epilepsy is a chronic neurological disorder affecting around 50 million people worldwide [51]. These epileptic syndromes are characterized by the occurrence of seizures; a transient clinical event caused by synchronous and/or abnormal and excessive neuronal activity in the brain. The clinical manifestations of seizures may affect the sensory, motor and autonomic functions of the body, as well as the consciousness, memory, cognition and behaviour of the patient [7]. In many cases, the aetiology of the syndrome is unknown, varying from genetic pre-dispositions, head traumas, brain tumours, infections, etc.

The diagnosis of epilepsy is based on the observation of epileptic seizures in conjunction with the analysis of the neuronal brain activity through electroencephalographic (EEG) recordings. The epileptologists observe the EEG recordings to find special characteristics of the ictal phase and interictal epileptiform discharges (IEDs) that occur during and between seizures. IEDs are short events lasting between 20 and 200 milliseconds caused by neurons firing synchronously [37]. These characteristics of the EEG can help confirm the diagnosis of epilepsy, the type of epilepsy syndrome, the type of seizures, etc.

The treatment of epilepsy can be achieved through medication and/or surgery; the goal being for the patients to be seizure-free. However, seizures are refractory to anti-epileptic drugs in 20% to 30% of patients [30]. Moreover, most of the drugs have common side-effects caused by their action on the central nervous system like tiredness, fatigue, unsteadiness, cognitive impairment, blurry vision, visual field loss, etc. These side-effects are sometimes related to the dosage of the drugs and, in most cases, are reversible [36]. As for surgery, patients are free from seizures impairing awareness in 58% of the cases [50]. Surgery is considered a safe operation as only 2% of the patients with temporal lobe epilepsy have clinically important consequences. Nonetheless, neuropsychological outcomes are relatively frequent and depend on which side the brain is operated on. Surgery on the left side causes verbal memory loss in 44% of the patients, visual memory loss in 23% and naming reduction in 34%. As for the patients operated on the right side of the brain, verbal memory loss occur in 30% of the patients and visual memory loss in 21% [44]. Even though there exist treatments that are effectively controlling seizures, they are not without consequences.

The quality of life of patients suffering from epilepsy can be greatly hindered by the syndrome. Indeed, some patients suffer from the constant fear and uncertainty of having seizures. Even patients for which anti-epileptic drugs are successfully controlling seizures can have their quality of life deteriorated significantly by adverse effects caused by medication [36]. Also, stigmata are still present and have a substantial impact on the quality of life,

both by reducing the social interactions quality and self-esteem of epileptic patients [20]. Furthermore, people diagnosed with epilepsy often have lower job and income levels compared to the general population [49]. Finally, in many countries, laws ban or restrict the possibility of epileptic patient to obtain a driving license. Not only the syndrome and the medication affects the quality of life of people suffering from epilepsy, but the social and economical repercussions also play a major role in its reduction.

1.2 Objectives

The objective of this thesis is to design a toolbox to automate the creation of efficient personalized seizure detection algorithms that operate in real-time. Seizure detection is the process of identifying and reporting the occurrence of ictal (seizure) events present in an EEG recording. The algorithms should monitor the EEG recordings in real-time and raise alarms during ictal events, i.e. after their beginning, but before their end. The generated algorithms are to be tailored to a specific patient in order to improve efficiency; measured by the expected number of successful and false alarms raised over a fixed period of time. Moreover, the toolbox must be usable without having to optimize the parameters on a patient specific basis. Therefore, it needs to be general enough to cover a large set of patient and seizure characteristics, and be able to focus on those relevant to a given patient.

In the short term, such tools could help alleviate the task of annotating EEG recordings during the diagnosis of the syndrome in new patients by

indicating points of interest to the epileptologist. Furthermore, nursing care facilities could benefit from such systems in order to alert nurses or relevant authorities that the patient is currently suffering from a seizure. Finally, this toolbox could help during the analysis of new large databases of EEG recordings taken from epileptic patients.

In the long run, these detection algorithms will help to create prediction algorithms. Indeed, once the automated detection of seizures is possible, one can explore the feature space and other algorithms in order to search for recurrent patterns occurring before the start of seizures. It could also help the development of early seizure detection systems that are capable of issuing alarms at the early beginning of an ictal event. These tools could help design new treatments in the form of prevention or early seizure abortion techniques, through electro-stimulation, automated chemical release mechanisms, etc. Lastly, discoveries made while studying EEG recordings for characteristics intrinsic to ictal events and the syndrome in general could help understand the mechanisms responsible for epilepsy.

1.3 Challenges

Creating such a toolbox is a complex procedure as it requires multiple components to be well designed in order to achieve good performance. The sources responsible for this complexity come from different areas, which are covered in the following paragraphs.

First, epileptic syndromes are quite complex and vary greatly across patients. In recent years, research has provided multiple different pathophysiological mechanisms, or combinations thereof, capable of explaining why seizures occur in different patients suffering from epilepsy [42]. However, these only cover a small fraction of the unsettling amount of possible mechanisms that could explain epileptic seizures. To help describe the characteristics of a patient epileptic syndrome, the International League Against Epilepsy (ILAE) published a report proposing a new classification scheme for both seizure types and epilepsy syndromes [5]. The scheme contains 40 different seizure types defined according to their pathophysiologic mechanisms, responses to anti-epileptic drugs, affected neuronal structures, patterns generated in EEG recordings, etc. The epilepsy syndromes are classified into 30 categories using properties such as the presence or absence of seizure types, age of onset, generated interictal EEG patterns, pathophysiologic mechanisms, etc. It is important to understand that these categorizations have been made in order to improve the quality of communication, diagnosis and research about epilepsy. We can see that a patient epileptic syndrome is a complex system of intertwined mechanisms that cause the occurrence of different types of seizures. Moreover, the classification schemes created by the ILAE express the heterogeneity of the syndromes across patients. Therefore, we need a toolbox that is able to manage the large diversity of possible mechanisms, syndromes and seizures.

Second, the data provided by EEG recordings is quite useful, but subject to some limitations. Standard EEG recordings are performed using electrodes that are placed on the scalp and sampled at a high frequency, providing a good temporal resolution. This procedure is relatively inexpensive and easy to perform [37]. On the other hand, the electrical activity recorded comes principally from neuronal cells close to the scalp surface, preventing the analysis of deeper brain structures. In order for the electrodes to capture changes in polarity, a large area of neurons, approximately 6 cm^2, must fire synchronously. Moreover, the electrical activity is blurred by the three layers separating the neurons from the electrodes: the cerebrospinal fluids, the skull and the scalp. As a result, the spatial resolution of scalp EEG recordings is poor [48]. In rare cases, mostly patients preparing for epilepsy surgery, electrocorticograms (ECoG) are available. These recordings are made with electrodes placed under the scalp, on the surface or inside the brain. The spatial resolution of these recordings is better locally and it is possible to look a deeper brain structures, but a very small portion of the brain is analyzed. Both types of recordings measure the neuronal activity caused by a sea of unknown brain mechanisms operating simultaneously. Therefore, the toolbox needs to extract information related to the epilepsy syndrome of a patient from the complex EEG signal that is recorded.

Third, we are subject to computational and machine learning constraints. Since EEG recordings are sampled at high frequencies using multiple electrodes, the throughput of data to analyze is quite substantial. Therefore, we

need fast computational tools to be able to extract and treat the information in real-time in order to detect seizures. The same high throughput of data is responsible for the large size of the datasets used to validate the performance of algorithms. Indeed, the three datasets used in this study weighed 249 GB in text format, but only contained data coming from 48 different patients, with an average of 36 hours of EEG recording per individual. This further emphasizes the importance of using fast and efficient algorithmic tools. Also, the fact that seizures are sparse events of relatively short durations, less than five minutes, prohibits the use of some standard machine learning techniques without some modifications [19]. Effectively, this property of seizures makes the datasets highly imbalanced as the time spent during ictal events is much smaller than the one spent in the interictal period. It is thus important to consider these algorithmic constraints when designing a toolbox, for it to be useful and efficient.

Finally, the lack of publicly available datasets containing EEG recordings from epileptic patients hinders the research on the syndrome. Indeed, the fact that most research is conducted using private datasets makes the comparison and reproducibility of results difficult. Moreover, to improve the quality of the experiments, one might need to test his findings on different epileptic syndromes that might be absent from his datasets. Therefore, an effort should be made to anonymize the medical data and make it publicly available. This would increase the pool of available EEG recordings and enable a better coverage of the syndromes. To the best of our knowledge, there exist only two

large freely available datasets. The first dataset is from the Epilepsy Center of the University Hospital of Freiburg, in Germany, and contains ECoGs of patients suffering from medically intractable epilepsy [8]. The second dataset is from the Children Hospital of Boston (CHB), conjointly with the Massachusetts Institute of Technology (MIT), and contains EEG recordings from pediatric patients suffering from epilepsy [45]; it is freely available on Physionet [12]. In 2012, an European effort, the EPILEPSIAE project [22], came to fruition and will soon provide the world's largest database of EEG recordings from epileptic patients, under some conditions. We ought to use and encourage these wonderful initiatives in order to improve the quality of the research performed on epilepsy.

1.4 Seizure Detection and Prediction

As explained in Section 1.2, the task of real-time seizure detection is to raise an alarm once an ictal event begins, but before it ends. This problem as been tackled by researchers since the early 1980's [13]. Nowadays, multiple other detection algorithms have been developed using more advanced computational techniques. They extract complex features from the raw EEG signals and use either a simple threshold or machine learning tools, such as neural networks, in order to detect the occurrence of seizures. Some of these algorithms are explained in more details in Chapter 5. Many of the limitations of the methods developed for seizure detection stem from design decisions or from their performance analysis. Indeed, the features are often hand selected or created by the authors, requiring prior knowledge about the

patients conditions. Also, the feature configurations are sometimes tailored to the dataset used in the experiments. Moreover, the performance is almost always evaluated using private datasets, making the comparison of different algorithms difficult. Finally, some datasets are pre-processed in order to remove EEG artifacts, overestimating the performance that the methodology would have if used in real clinical settings.

As for the task of seizure prediction, the goal of the algorithm is to forecast the occurrence of a seizure in the near future, from a few seconds to a few minutes in advance. These algorithm work in a similar way, analyzing features extracted from the EEG signals in order to detect particular changes in the pre-ictal period present before an ictal event. It is important to note that even if a lot of work as been done in this area, it is still unknown if seizures are predictable [32]. This is mainly the result of the poor evaluation methodologies and lack of rigorous statistical analysis of the performance of these algorithms.

1.5 Contributions

We have developed a toolbox that automates the creation of personalized seizure detection algorithms by gathering and combining knowledge from multiple different research areas. It is general and proficient enough to generate efficient algorithms for a large proportion of epileptic syndromes and seizures without the need to input any prior knowledge about a patient's characteristics. It was designed with modularity in mind, so that its components may be used independently or conjointly. Figure 1–1 depicts a

flowchart of the toolbox and its components. We now describe the modules as well as the reasons behind their design decisions.

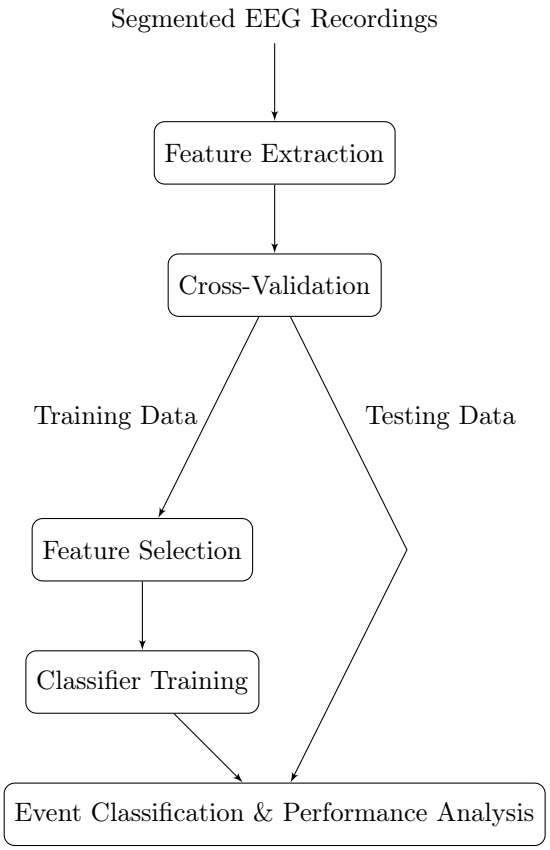


Figure 1–1: Flowchart describing the steps taken by the toolbox. The modularity of the toolbox enables the interchangeability of any subcomponents.

The first module is the feature extraction component. We implemented a large set of features created and validated by the seizure detection/prediction community. Most features have parameters, enabling them to cover a large set of epileptic syndromes and seizures. Moreover, their validation implies that they gather relevant information about the epileptic syndrome and are

able to make abstraction of the other brain processes. We also provide a complexity analysis of each feature to guide the choice of features to extract if constrained by computational costs.

The second module partitions the data for cross-validation. This technique creates an environment that simulates real experimental conditions in the sense that we train on a subset of the data and validate the performance on another, unseen, subset. The results obtained when using cross-validation are closer to the ones to be expected if the algorithm was to be used on a patient in real-time (i.e. on unseen data).

The third module is the feature selection component. Its role is to select a subset of features that are relevant to a patient's epileptic syndrome and seizures in order to reduce the computational cost of the feature extraction step. Indeed, this smaller subset enables us to compute the features in real-time. Furthermore, the selected subset must not hinder the efficiency of the automated seizure detection algorithm. We evaluate the performance of two distinct feature selection methods: a basic score selection technique based on the area under the receiver operating characteristic (ROC) curve and a model-based selection using a $\ell 1$-regularized logistic regression.

The fourth module is the classification component, enabling the training of a classifier. The detection of ictal events can be done using a $\ell 1$-regularized logistic regression, or for a richer hypothesis class, extremely randomized trees (Extra-Trees). Extra-Trees are useful for multiple reasons: they are fast to train and query, and are able to capture complex changes between

classes of events (in our case interictal and ictal data). The logistic regression is used as a simple model to validate the use of more complex classifiers.

The fifth, and last, module evaluates the performance of the toolbox. It measures the detection rate, false positive rate, and detection latency of the algorithms generated for the patients.

We then validate the performance of the toolbox on three datasets. We used the datasets provided by the University Hospital of Freiburg and the CHB-MIT. The third dataset was provided by the Montréal Neurological Institute (MNI) and contains ECoG recordings of Sprague-Dawley rats where status epilepticus was induced by injection of pilocarpine. The use of this data is justified by the similarity of the model to human epilepsy [4], the ease of long-term collection compared to human ECoG recordings and the fact that it enables the possibility to explore different control mechanisms which could either prevent or abort seizures. Both the MNI dataset and the toolbox are made publicly available at `www.cs.mcgill.ca/~gsauln` under the Apache License, Version 2.0.

CHAPTER 2
Technical Background

2.1 Machine Learning

In this section, we introduce important notions about machine learning that enable us to understand the design decisions that were taken throughout this project. We first give an introduction to supervised learning and a few examples of classification methods. Then, we explain the bias/variance trade-off, useful when considering classifiers and estimating their testing error. Finally, we cover the importance of cross-validation, a technique used to measure the expected testing error correctly.

2.1.1 Supervised Learning

In order to define supervised learning, we need to introduce some notation. Let X be the space of *input* variables and Y be the space of *output* variables. We have that $\mathbf{x}_i \in X$ is an instantiation of an input variable represented as a vector, with $\mathbf{x}_{ij}$ representing the value at its j^{th} position. Moreover, $y_i \in Y$ is the output variable corresponding to $\mathbf{x}_i$. We assume that there exists a true function $f : X \to Y$ that models the system we are observing such that given an input variable $\mathbf{x}_i \in X$, we have $f(\mathbf{x}_i) = y_i$. In other words, $f(\mathbf{x}_i)$ defines the output variable, or label, y_i. Unfortunately, f is often unknown and we would like to be able to determine y_i given a new unseen instance $\mathbf{x}_i \in X$. Thus, we want an approximation function $\widehat{f} : X \to Y$

that yields approximate classifications $\widehat{f}(\mathbf{x}_i) = \widehat{y}_i$ such that $\widehat{f}(\mathbf{x}_i)$ is as close as possible to $f(\mathbf{x}_i)$ most of the time. In supervised learning, we are given a set of inputs $\mathbf{X}$ and the corresponding set of outputs $\mathbf{Y}$ and we are asked to find a good approximation function $\widehat{f}$ using this known data [18]. Note that both $\mathbf{X}$ and $\mathbf{Y}$ are in fact multisets: sets in which we allow multiplicities. When the output value Y is categorical, the task is called classification, and when the output is quantitative, the task is called regression.

2.1.2 Classification Methods

In order to approximate f, multiple different methodologies of varying complexities have been defined by the machine learning community, where the principal ones are presented in [24]. These methods often map the samples $\mathbf{x} \in X$ into a feature space defined by some function ϕ. This function enables to extract relevant information about $\mathbf{x}$. As an example, if $\mathbf{x}$ is an individual taken from a population X, $\phi(\mathbf{x}) = \{age, weight, heigth, sex\}$ could be the set of properties helpful in differentiating overweight versus healthy people. Amongst the simplest classification methods, linear classifiers will try to capture linear relations between the properties of the samples and their corresponding labels. The $\ell 1$-regularized logistic regression, a linear classifier, is described in Section 3.4.2. More complex classifiers, such as decision trees, can exploit the feature space in order to capture complicated relations between properties. Extra-Trees, a variant of decision trees, are presented in Section 3.5.2. While only two different classifiers are used in this thesis, many others exist, such as k-nearest neighbours, artificial neural network,

support vector machines, etc. and are available through libraries such as Weka [15].

2.1.3 Bias/Variance trade-off

When training a classifier $\widehat{f}$, it is important to take into account the bias/variance trade-off. Indeed, the average error of a classifier $\widehat{f}$ on a set of testing data (i.e. unseen data) is related to the sum of the bias and variance errors. We can think of training $\widehat{f}$ as selecting the right function from a family of functions $\mathcal{F}$. The bias error of a function $\widehat{f}$ is caused by the inability of $\mathcal{F}$ to correctly approximate the true function f. As for the variance error, it is correlated to the variance of the average error of $\mathcal{F}$ on a given test set, when trained on different sets sampled from the same distribution.

The complexity of the family of functions $\mathcal{F}$ affects both the bias and variance errors: the bias is inversely proportional to the complexity of the family $\mathcal{F}$ and the variance is proportional to it. Indeed, if we have a family $\mathcal{F}$ which has a low complexity compared to f, we will never be able to model f correctly, inducing a high bias, but variations in the training data will not affect the choice of $\widehat{f} \in \mathcal{F}$ by much, inducing lower variance. On the other hand, a very complex family of functions $\mathcal{F}$ will be able to model the training data perfectly, inducing a low bias, but will make many errors predicting unseen data, as the chosen function $\widehat{f} \in \mathcal{F}$ is tailored only according to the training data used. Figure 2–1 depicts the standard behaviour of the bias, variance and testing error with respect to complexity of $\mathcal{F}$.

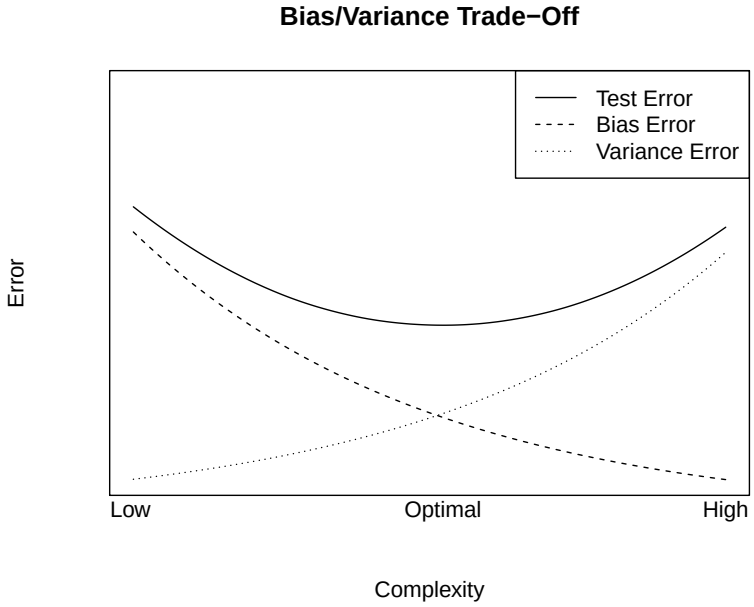


Figure 2–1: Standard behaviour of the bias, variance and test error with respect to complexity of a family of function $\mathcal{F}$. The optimal complexity lies at the intersection of the bias error and variance error curves.

2.1.4 Cross-Validation

Since we are now aware of the bias and variance errors related to a family of function $\mathcal{F}$, we can now explain the importance of using cross-validation to validate the experimental results. If we train our algorithm using all the available data, and then test it using the same data, the error calculated is only related to the bias. Indeed, there are now no differences between the testing set and training set, therefore no variance error is present. Notice from Figure 2–1 that the bias error goes to zero as the complexity of the classifier increases. Therefore, we could optimize $\widehat{f}$ to have the lowest possible training error by increasing its complexity. The problem is that this classifier would

have a fairly poor performance when used on new unseen data. This validates the importance of using the test error as the true efficiency of a classifier.

In an optimal world, we would be able to separate the data randomly into two large sets, one used for training and the other used to validate the performance of our classifier. Unfortunately, this is not always possible. As in the problem of creating automated seizure detection algorithms, the data is often scarce. Indeed, for some patients, only 3 seizures are available. This would imply either training on one and testing on two, or the inverse, training on two and testing on one. In both cases, we can see that the reliability of the results would probably be poor, as either there is not enough data to successfully train the classifier or validate its performance accurately.

Luckily for us, even when the available data is scarce, we can compute the expected test error easily using k-fold cross-validation. This methods takes the available training data and separates it uniformly into k different sets. Then, all but one set are used in the training of the classifier. The performance of this classifier is evaluated on the remaining set, not contained in the training data. This process is repeated until all sets have been used as test sets. Finally, the overall performance of the classifier is calculated by averaging the performance on each testing sets. This method directly estimates the expected testing error of the classifier [18].

2.2 Seizure Detection

The task of an automated seizure detection algorithm is to analyst the EEG recording in real-time and raise alarms during ictal events. In order

to do so, at a fixed time interval, the algorithm extracts features from the EEG recording and feeds them to a classifier. Using the output of the latter, an alarm may be raised. The interested reader can refer to Figure 2–2 for a flowchart of the procedure. The objective is to only raise alarms during ictal events, without missing any.

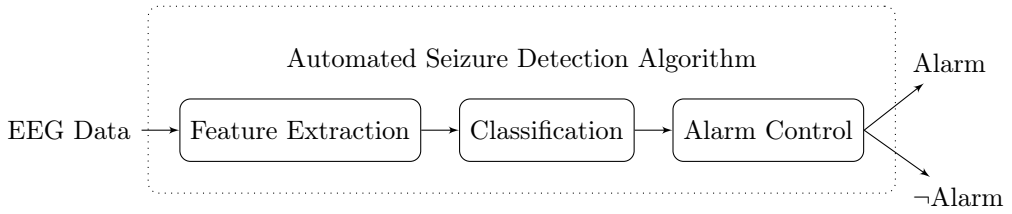


Figure 2–2: Flowchart describing the steps taken by an automated seizure detection algorithm.

2.3 EEG Recordings

EEG recordings consist of multivariate time-series data where each measured variable comes from a different electrode. The electrodes are usually sampled between 200Hz to 2000Hz and downsampled to 256Hz for computational reasons. The univariate time-series sampled by those electrodes are called EEG channels. The analog data is often digitized to 16 bits of resolution.

To be able to train a classifier, each EEG recording needs to be analyzed by an epileptologist in order to define the beginning and end of each ictal events. These cut points enable us to label each EEG sample as either *ictal* or *interictal*. Figure 2–3 shows an example of a scalp EEG recording and

Figure 2–4 shows an example of an ECoG recording, in which seizures have been annotated.

Using the machine learning terminology introduced in Section 2.1, the set $Y = \{ictal, interictal\}$ corresponds to the two output variables that we are trying to approximate with our classifier. The set of input variables is defined as $X \subseteq \mathbb{R}^d$, where d corresponds to the number of features extracted from the EEG recording. Each vector of X corresponds to the possible feature values an EEG could have at a single point in time.

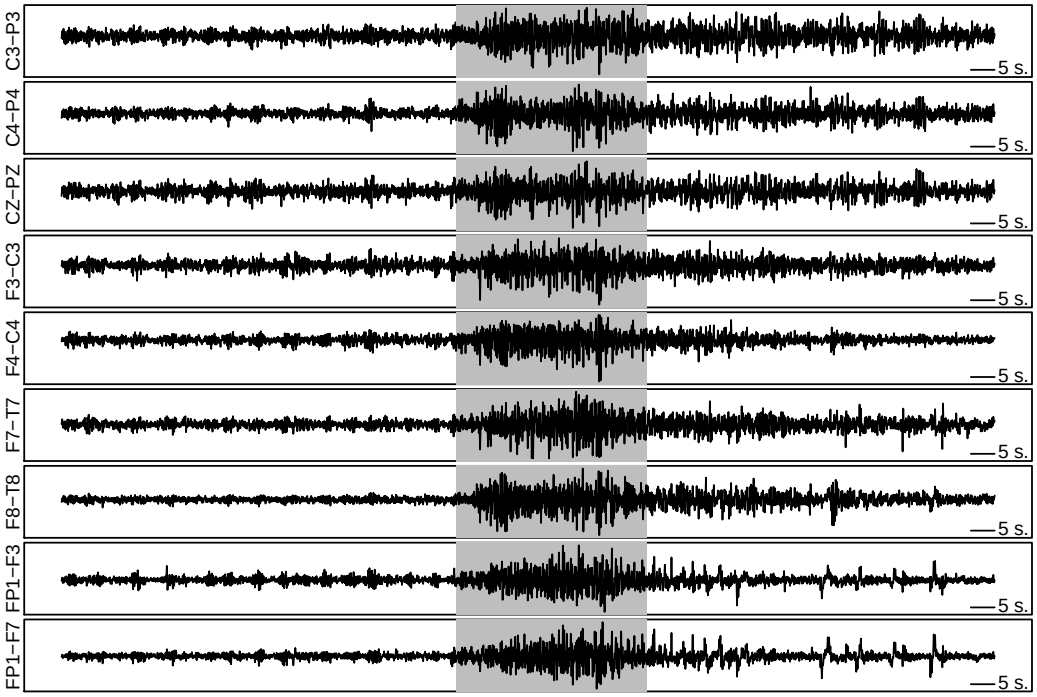


Figure 2–3: Multi-channel EEG recording containing a seizure delimited by the gray areas. The signal is less defined compared to ECoG recordings.

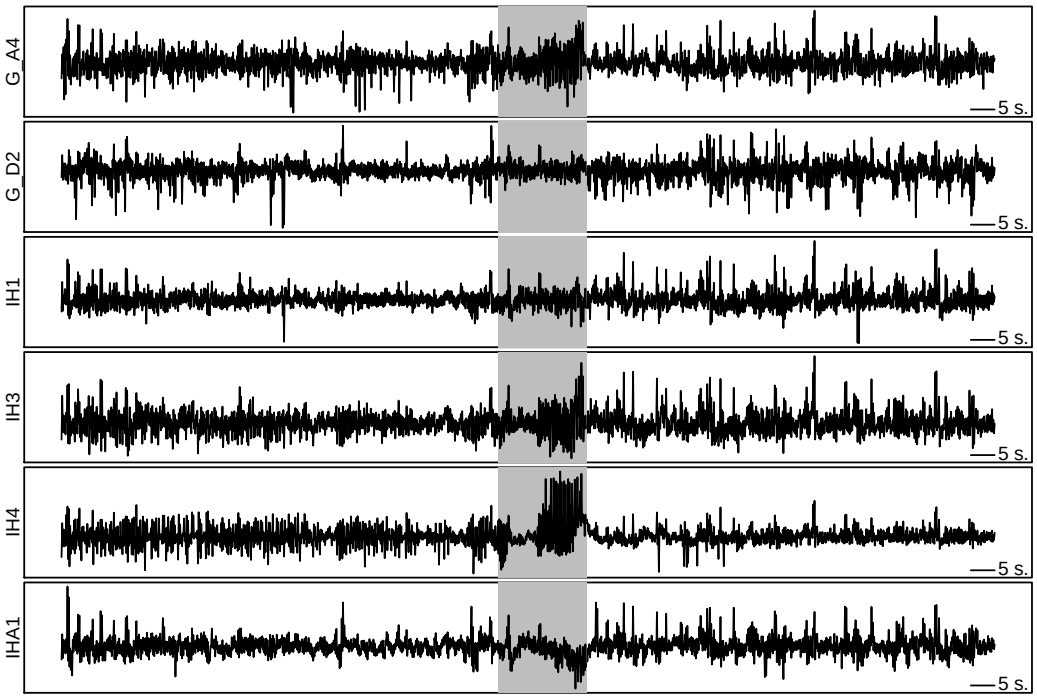


Figure 2–4: Multi-channel ECoG recording made with three focal and three extra-focal electrodes containing a seizure delimited by the gray areas. The signal is more defined compared to scalp EEG recordings.

CHAPTER 3
A Toolbox for Personalized Seizure Detection Algorithms

This chapter presents the core contribution of thesis: the toolbox implemented to generate personalized epileptic seizure detection algorithms that operate in real-time. We cover the different components of the toolbox depicted in Figure 1–1. Unless specifically mentioned, everything was implemented directly in the toolbox.

3.1 Data Description

To ease the pre-processing of datasets, the toolbox uses a simple text format. An EEG recording is defined by a unique folder containing a text file for each of its channels. Each of these files has one value per line, corresponding to the amplitude of the signal recorded by a given electrode. The files must all start and end at the same time, as well as possess the same sampling rate. Therefore, the n^{th} line of a channel file must exists in all of them, and correspond to the same recording time. A simple XML file encapsulates all the information required to process the multivariate time-series: the EEG name, the sampling rate, the name and filenames of each channel file and the labels of the different events present in the recording. These labels are defined by the first and last samples present in the contiguous event, combined with a numerical value corresponding to its type.

3.2 Feature Extraction

The feature extraction module is responsible to extract relevant information about a patient's epileptic syndrome and seizures from his EEG recordings. In order to do so, a large set of univariate features, computed on a unique EEG channel, and bivariate features, computed on pairs of EEG channels, has been implemented. The toolbox provides a direct access to a well-tested implementation of many features proposed in the literature to achieve seizure detection and/or prediction ([32], and references therein). It is worth noting that most of the features depend on some parameters (e.g. window length over which the feature is calculated, type of finite impulse response filter used, etc.). The toolbox is capable of extracting them over a wide range of parameter values. This range can be modified if desired, by reducing the range to accelerate computation, or to possibly improve performance by extending the range. Our assumption is that the feature set, with its possible configurations, is large enough to capture relevant information from the EEG recordings of many different epileptic syndromes and seizures.

To account for the non-stationarity of the EEG recordings and to detect changes in the brain state, the features are extracted over windows of EEG data that are moved in time. Multiple windows lengths are used to get different time resolutions. Indeed, when moved by a small amount of time, a feature value will vary quickly in short windows, but slowly in longer windows, making the former more sensitive and the latter more stable to changes in the brain activity. The toolbox is able to extract any features

on a set of window length $\mathbf{W}$ with a spacing of δ samples between the end of two consecutive windows. These windows are synchronized at their end, such that in a real-time setting, all features are computed using the latest available data.

We now define the mathematical notation that will be used to describe the features. Let E represent an EEG recording and let $c \in E$ be a channel present in the EEG. The vector $\mathbf{c}$ represents a time-series consisting of the recordings of the electrode across time. We will denote by $\mathbf{c}_k$ the value of the k^{th} sample of the time-series $\mathbf{c}$. A window of length w ending with sample k will be written as $\mathbf{c}_k^w$. Thus, $\mathbf{c}_{kl}^w$ corresponds to the l^{th} value of $\mathbf{c}_k^w$ for $0 \leq l < w$. Note that all indices are starting at 0. An example of two consecutive sets of moving windows with $\mathbf{W} = \{200, 400, 1000\}$ and $\delta = 200$ is illustrated in Figure 3–1. A complete feature vector $\mathbf{x} \in X$ consists of all the features computed on all the channels (or pair of channels) over a given window set, defined by $\mathbf{W}$ and k.

To be concise with the machine learning notation introduced in Section 2.1, we will denote the set of feature vectors extracted from the EEG data by $\mathbf{X}$ and its corresponding set of labels by $\mathbf{Y}$. The label $y_i \in \mathbf{Y}$ assigned to a feature vector $\mathbf{x}_i \in \mathbf{X}$ corresponds to the label that is in majority across the samples $\{k - \delta + 1, k - \delta + 2, \ldots, k\}$ of the EEG recording.

We use the notation $(\mathbf{a} \circ \mathbf{b})_i = \mathbf{a}_i \mathbf{b}_i, \forall i$ to denote pointwise multiplication of the vectors $\mathbf{a}$ and $\mathbf{b}$.

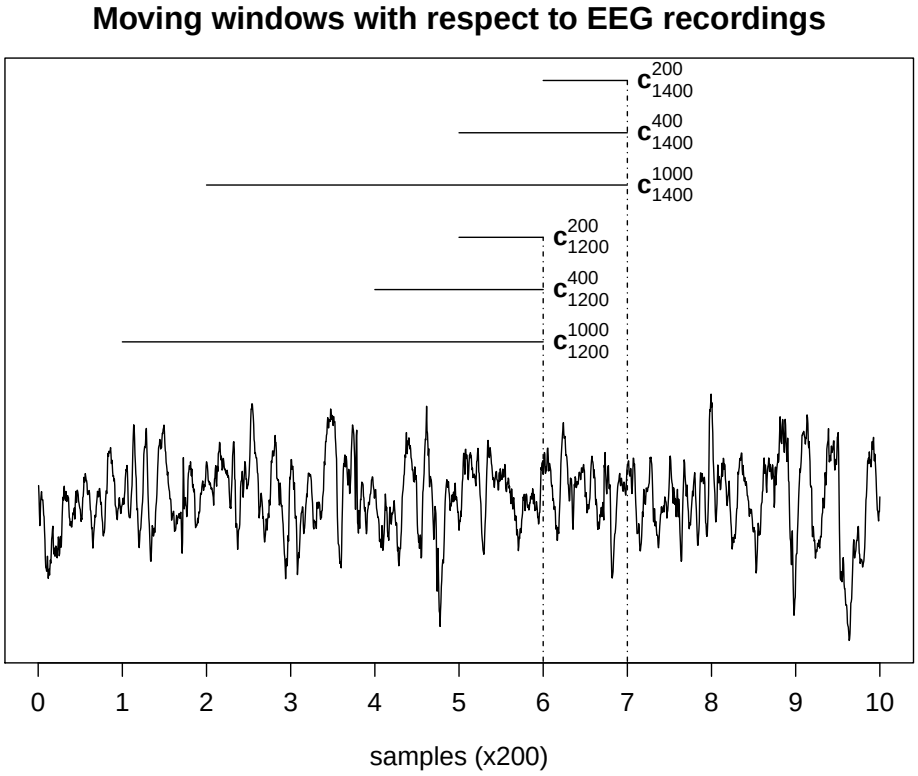


Figure 3–1: Two consecutive sets of moving windows with lengths $\mathbf{W} = \{200, 400, 1000\}$ and $\delta = 200$ are illustrated in the above figure. The horizontal lines correspond to the segment of data on which the features are extracted and the dashed lines show the synchronization of the endpoints of a set of windows. The signal shown corresponds to $c \in E$ and is sampled at 200 Hz.

3.2.1 Univariate Features

Univariate features are extracted from a single channel $c \in E$ at a time.

- *Mean* (μ):

$$\mu(\mathbf{c}_k^w) = \frac{1}{w} \sum_{l=0}^{w-1} \mathbf{c}_{kl}^w. \tag{3.1}$$

The mean captures shifts in the base line of the EEG recordings.

- *Variance* (σ^2):

$$\sigma^2(\mathbf{c}_k^w) = \frac{1}{w-1}\sum_{l=0}^{w-1}(\mathbf{c}_{kl}^w - \mu(\mathbf{c}_k^w))^2. \tag{3.2}$$

The variance is positively correlated to the amplitude of the measurements made by the electrodes.

- *Line-Length* ($\mathcal{L}$):

$$\mathcal{L}(\mathbf{c}_k^w) = \sum_{l=1}^{w-1}\left|\mathbf{c}_{kl}^w - \mathbf{c}_{k[l-1]}^w\right|, \tag{3.3}$$

where $|\cdot|$ denotes the absolute value. The line-length was introduced in [34] and is positively correlated to the high-frequency components contained in the signal and the signal's amplitude.

- *Fast Fourier Transform* (FFT):

The spectral characteristics of the brain activity are a natural component to extract from EEG recordings. A windowing function $\mathbf{h}$ such as Hann or Hamming [17] is applied on the window $\mathbf{c}_k^w$ to reduce the creation of artifacts due to the non-periodicity of the signal inside that window. Let $\mathbf{h}$ be a windowing function such that $|\mathbf{h}| = w$. We denote the FFT of $(\mathbf{h} \circ \mathbf{c}_k^w)$ as

$$\mathsf{FFT}(\mathbf{h} \circ \mathbf{c}_k^w)_l = \sum_{j=0}^{w-1}(\mathbf{h} \circ \mathbf{c}_k^w)_j \cdot e^{\frac{-i2\pi lj}{w}}, \tag{3.4}$$

with $l \in \{1, 2, \ldots, \frac{w}{2}\}$ and i representing the imaginary number. Given the sampling rate f_s of the EEG recording and a frequency f with $0 \leq f \leq \frac{f_s}{2}$, the linear weighted average of the magnitude of the spectrum between $l_1 = \left\lfloor \frac{f \cdot w}{f_s} \right\rfloor$ and $l_2 = \left\lceil \frac{f \cdot w}{f_s} \right\rceil$ is returned as the amplitude of f. Note that only frequencies up to $\frac{f_s}{2}$ are considered because of the Nyquist-Shannon sampling theorem [43]. The analysis of the spectral components of an EEG recording is a standard procedure used as the brain activity is often characterized in wave bands.

- *Mean of the Squared Convolution* (MSC):

  Let $\mathbf{g}$ be a finite impulse response (FIR) filter. The convolution of $\mathbf{g}$ and $\mathbf{c}_k^w$ is defined as

$$(\mathbf{g} * \mathbf{c}_k^w)_l = \sum_{j=0}^{w+|\mathbf{g}|-1} \mathbf{g}_{|\mathbf{g}|-l+j-1} \mathbf{c}_{kj}^w, \tag{3.5}$$

  where $l \in \{0, 1, \ldots, w+|\mathbf{g}|-1\}$ and the convolution operator is denoted by $*$. The values for out of bounds indices are assumed to be 0. The mean of the squared values of the convolved signal is defined as

$$\mathsf{MSC}(\mathbf{c}_k^w, \mathbf{g}) = \frac{1}{w + |\mathbf{g}| - 1} \sum_{l=0}^{w+|\mathbf{g}|-1} \left| (\mathbf{g} * \mathbf{c}_k^w)_l \right|^2. \tag{3.6}$$

  The MSC feature corresponds to the power, or energy per sample, of the convolved signals. This feature was taken from [35].

  The FIR filters used with the MSC feature for automated seizure detection are constructed from the Daubechies 4 wavelets. We first describe

these wavelets as well as the discrete wavelet transform in order to explain how the filters are constructed. The Daubechies 4 wavelets consists of two FIR filters:

$$\mathbf{D}^1_{high} = \left(\frac{1-\sqrt{3}}{4\sqrt{2}}, \frac{-3+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{-1-\sqrt{3}}{4\sqrt{2}} \right), \tag{3.7}$$

$$\mathbf{D}^1_{low} = \left(\frac{1+\sqrt{3}}{4\sqrt{2}}, \frac{3+\sqrt{3}}{4\sqrt{2}}, \frac{3-\sqrt{3}}{4\sqrt{2}}, \frac{1-\sqrt{3}}{4\sqrt{2}} \right), \tag{3.8}$$

which are depicted in Figure 3–2.

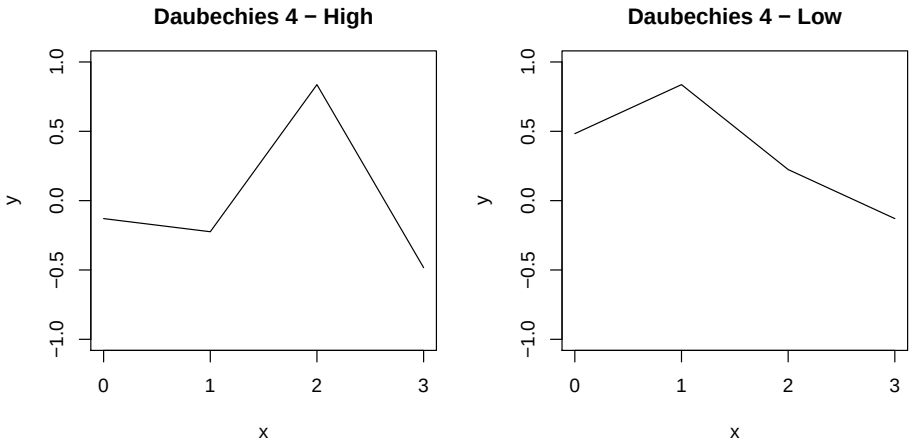


Figure 3–2: The Daubechies 4 High and Low band filters.

To get the coefficients of the discrete wavelet transform of a signal $\mathbf{s}$ at level n, we need to follow the operations on the path from $\mathbf{s}$ to level n in the filter bank depicted in Figure 3–3.

It is however possible to obtain the coefficients of the signal $\mathbf{s}$ at level n by performing a single convolution with the filter $\mathbf{D}^n_{high}$ and down-sampling the result by a factor of 2^n:

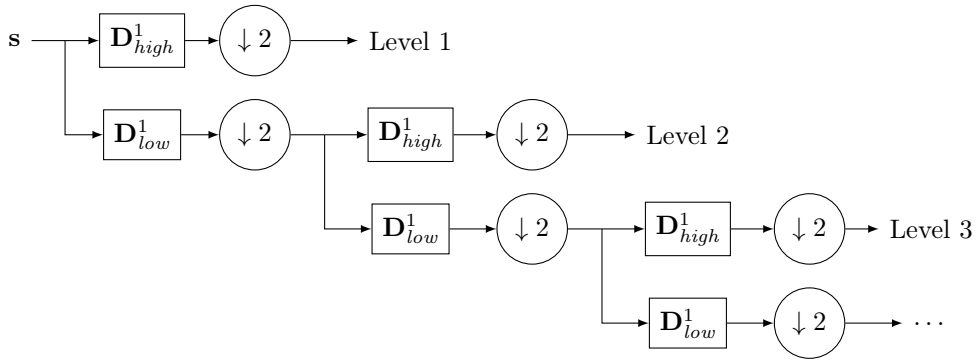
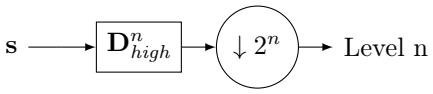


Figure 3–3: Filter bank describing the discrete wavelet transform. To get the coefficients at level n, one needs to follow the operations present on the path from $\mathbf{s}$ to that level. The boxes represent convolutions and the circles represent sampling operations such that $\downarrow 2$ is a downsampling by a factor of two.



We used the filters $\mathbf{D}^n_{high}$, with different values of n, as our FIR filters in the MSC feature. To construct the filter $\mathbf{D}^n_{high}$, one needs to follow the operations presented in the filter bank depicted in Figure 3–4. The use of these filters was suggested in [35]. Figure 3–5 shows $\mathbf{D}^n_{high}$ and its frequency response for different values of n.

3.2.2 Bivariate Features

Bivariate features are extracted over a pair of EEG channels $c, d \in E$ in order to measure the relations between them. The features are extracted over windows of the same size at the same time location, i.e. $\mathbf{c}^w_k, \mathbf{d}^w_k$.

- *Linear Coherence* (LC):

  Let $\mathbf{g}$ be a weighted average filter of size $|\mathbf{g}| \ll \frac{w}{2}$. For larger $|\mathbf{g}|$, the

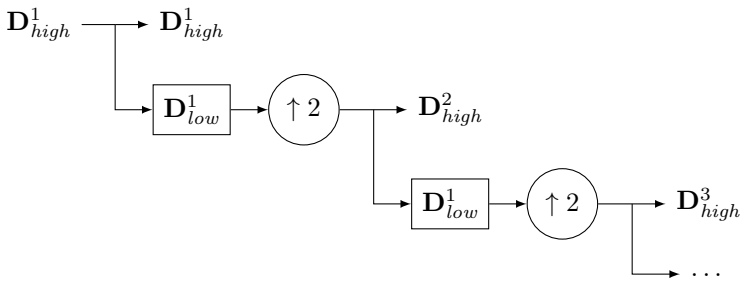


Figure 3–4: Filter bank describing the construction of $\mathbf{D}^n_{high}$. To get the filter at level n, one needs to follow the operations present on the path from the original $\mathbf{D}^1_{high}$ to that level. The boxes represent convolutions and the circles represent sampling operations such that $\uparrow 2$ is a upsampling by a factor of two (adding a zero between each sample).

statistical significance of the measure is increased at the cost of spectral distortion [25]. Let

$$G(\mathbf{c}^w_k, \mathbf{d}^w_k, \mathbf{g}, \mathbf{h}) = \mathbf{g} * [\mathsf{FFT}(\mathbf{h} \circ \mathbf{c}^w_k) \circ \mathsf{FFT}(\mathbf{h} \circ \mathbf{d}^w_k)^*], \tag{3.9}$$

be the sample cross-spectrum of $\mathbf{c}^w_k$ and $\mathbf{d}^w_k$ convolved with $\mathbf{g}$ where a^* denotes the complex conjugate of a. This estimation of the spectrum is shown to be roughly equivalent to the Welch method with $M = |\mathbf{g}|$ when $\mathbf{g}$ is uniform [25]. We then define the linear coherence as

$$\mathsf{LC}(\mathbf{c}^w_k, \mathbf{d}^w_k, \mathbf{g}, \mathbf{h})_l = \left| \frac{G(\mathbf{c}^w_k, \mathbf{d}^w_k, \mathbf{g}, \mathbf{h})_l}{\sqrt{G(\mathbf{c}^w_k, \mathbf{c}^w_k, \mathbf{g}, \mathbf{h})_l \cdot G(\mathbf{d}^w_k, \mathbf{d}^w_k, \mathbf{g}, \mathbf{h})_l}} \right|, \tag{3.10}$$

where $l \in \{|\mathbf{g}| - 1, |\mathbf{g}|, \ldots, \frac{w}{2}\}$. Then, for a given sampling rate f_s and a frequency f, with $0 \leq f \leq \frac{f_s}{2}$, the value of the measure is located at index $l = \mathsf{round}\left(\frac{f \cdot w}{f_s}\right) + |\mathbf{g}| - 1$. The value at l gives the linear coherence

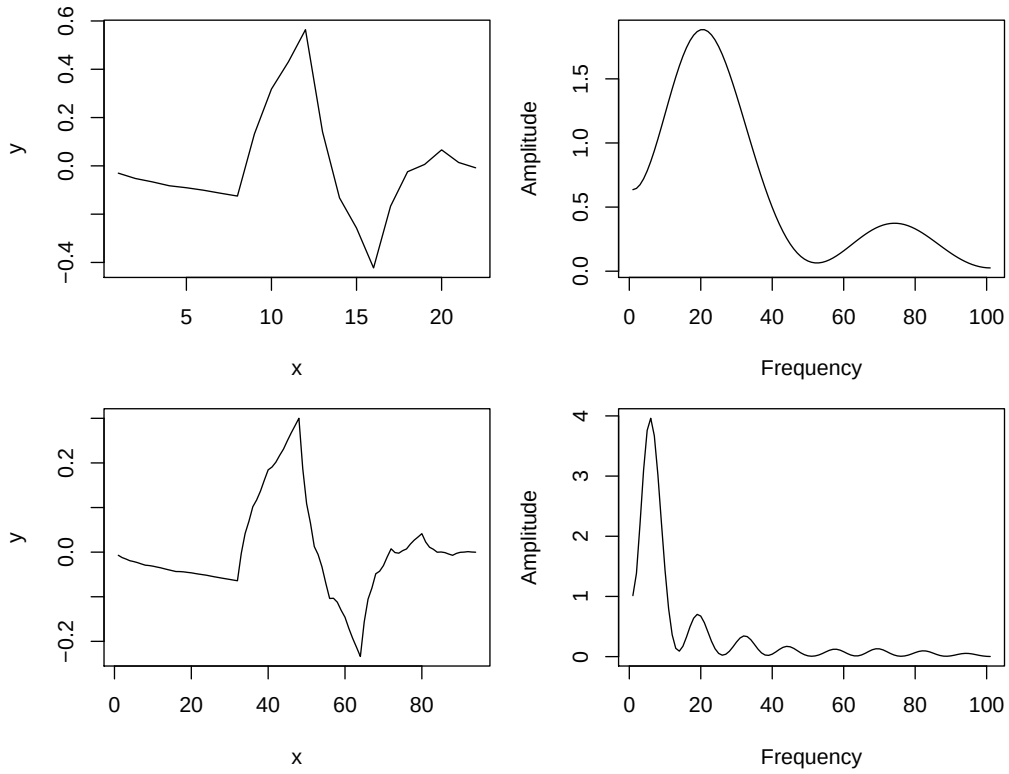


Figure 3–5: $\mathbf{D}^n_{high}$ filters and their frequency response for $n \in \{3, 5\}$. $\mathbf{D}^3_{high}$ and $\mathbf{D}^5_{high}$ are represented on the first and second rows, respectively. The first column corresponds to the filter values and the second column corresponds to the amplitude of the frequency response of the filter on a signal sampled at 200Hz.

between $\mathbf{c}^w_k$ and $\mathbf{d}^w_k$ at the frequency band defined by $\mathbf{g}$ centered at f. The linear coherence has a value of 1 when there is a perfect linear synchronization and 0 when there is no synchronization. This feature as been used in [38].

- *Maximum Cross-Correlation* (MCC):

  Let

$$C(\mathbf{c}_k^w, \mathbf{d}_k^w, \tau) = \begin{cases} \frac{1}{w-\tau} \sum_{j=0}^{w-\tau-1} \mathbf{c}_{k(j+\tau)}^w \mathbf{d}_{kj}^w & \tau \geq 0 \\ C(\mathbf{d}_k^w, \mathbf{c}_k^w, -\tau) & \tau < 0 \end{cases}, \tag{3.11}$$

  define the standard linear cross-correlation. Then,

$$\mathsf{MCC}(\mathbf{c}_k^w, \mathbf{d}_k^w, r, t) = \max_{\tau \in R(r,t)} \left| \frac{C(\mathbf{c}_k^w, \mathbf{d}_k^w, \tau)}{\sqrt{C(\mathbf{c}_k^w, \mathbf{c}_k^w, 0) \cdot C(\mathbf{d}_k^w, \mathbf{d}_k^w, 0)}} \right|, \tag{3.12}$$

  measures the maximum normalized lag synchronization of the two signals in the range defined by $R(r,t) = \{-r, -r+t, -r+2t, \ldots, r\}$. The cross-correlation measures the linear similarity of the amplitude of two signals as a function of lag. Therefore, the maximum cross-correlation returns the highest similarity obtained across the lags contained in $R(r,t)$. This feature as been used in [38].

- *Non-Linear Interdependence* (NI):

  Let

$$\psi(\mathbf{c}_k^w, d, \tau)_l = (\mathbf{c}_{kl}^w, \mathbf{c}_{k[l+d\tau]}^w, \ldots, \mathbf{c}_{k[l+(d-1)\tau]}^w), \tag{3.13}$$

  be the time delay embedding of $\mathbf{c}_k^w$ in d dimensions with lag τ. Note that valid values for l range from 0 to $w-(d-1)\tau-1$. We define the set of indices corresponding to the r nearest neighbours of l in $\psi(\mathbf{c}_k^w, d, \tau)$ as

$$\mathsf{NN}(\mathbf{c}_k^w, d, \tau, r)_l = \underset{\substack{A\subseteq\{0,\ldots,w-(d-1)\tau-1\}:\\ |A|=r}}{\operatorname{argmin}} \sum_{j\in A} \|\psi(\mathbf{c}_k^w, d, \tau)_l - \psi(\mathbf{c}_k^w, d, \tau)_j\|_2, \quad (3.14)$$

where $\|\mathbf{v}\|_2$ denotes the Euclidean norm (length) of the vector $\mathbf{v}$ and $\|\mathbf{v}\|_2^2$ naturally denotes the square of $\|\mathbf{v}\|_2$. If we let

$$R(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r)_l = \frac{1}{r} \sum_{j\in \mathsf{NN}(\mathbf{d}_k^w, d, \tau, r)_l} \|\psi(\mathbf{c}_k^w, d, \tau)_l - \psi(\mathbf{c}_k^w, d, \tau)_j\|_2^2, \quad (3.15)$$

and define $N = |\psi(\mathbf{c}_k^w, d, \tau)| = w - (d-1)\tau$, one can compute the following measures of non-linear interdependence

$$\mathsf{NI}_S(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r) = \frac{1}{N} \sum_{l=0}^{N-1} \frac{R(\mathbf{c}_k^w, \mathbf{c}_k^w, d, \tau, r)_l}{R(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r)_l}, \quad (3.16)$$

$$\mathsf{NI}_H(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r) = \frac{1}{N} \sum_{l=0}^{N-1} \log \frac{R(\mathbf{c}_k^w, \mathbf{c}_k^w, d, \tau, N-1)_l}{R(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r)_l}, \quad (3.17)$$

where higher values imply higher degrees of non-linear interdependence. These features measure a notion of generalized synchronization that captures an asymmetric relation of dependence between $\mathbf{c}_k^w$ and $\mathbf{d}_k^w$. They were suggested in [1].

- *Phase Synchrony* (PS):

  Let $a \in [0, 1]$ and h be a width parameter, we define

$$\sigma(h, a) = \frac{h}{\sqrt{2} \cdot \operatorname{erf}^{-1}(a)}, \quad (3.18)$$

where

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \tag{3.19}$$

is the error function. Then,

$$\mathcal{G}(f, h, a, f_s)_l = \frac{1}{f_s} e^{2\pi i \frac{l}{f_s}(f + \pi i \frac{l}{f_s}[\sigma(h,a)]^2)}, \tag{3.20}$$

is a complex valued function with a frequency response corresponding to a normal distribution centered at f where a of the area of its probability density function (pdf) is located between $[f - h, f + h]$. Note that f_s is the sampling rate of the EEG recording and $l \in \mathbb{Z}$.

Let $\mathbb{1}_{expr}$ be the indicator function; having a value of one when the *expr* is satisfied and zero otherwise. Then, we can define the Hilbert transform in the time domain. For w even, we define

$$\mathcal{H}(w)_j = \mathbb{1}_{j=\lfloor \frac{w}{2} \rfloor} + \frac{i}{w} \left(\cot\left(\frac{(j - \lfloor \frac{w}{2} \rfloor)\pi}{w} \right) - \frac{\cos\left((j - \lfloor \frac{w}{2} \rfloor)\pi\right)}{\sin\left(\frac{(j - \lfloor \frac{w}{2} \rfloor)\pi}{w} \right)} \right), \tag{3.21}$$

when $j - \lfloor \frac{w}{2} \rfloor$ is even and

$$\mathcal{H}(w)_j = 0, \tag{3.22}$$

when $j - \left\lfloor \frac{w}{2} \right\rfloor$ is odd. For w odd, the Hilbert transform is defined as

$$\mathcal{H}(w)_j = \mathbb{1}_{j=\left\lfloor \frac{w}{2} \right\rfloor} + \frac{2i}{w} \sin^2 \left(\frac{(j - \left\lfloor \frac{w}{2} \right\rfloor)\pi}{2} \right) \cot \left(\frac{(j - \left\lfloor \frac{w}{2} \right\rfloor)\pi}{w} \right), \quad (3.23)$$

for all j's. The real values of $\mathcal{H}(w) * \mathbf{c}_k^w$ correspond to $\mathbf{c}_k^w$ and the imaginary values correspond to the Hilbert transform of $\mathbf{c}_k^w$. This representation of the Hilbert transform is useful as we can now use either $\mathcal{G}$ or $\mathcal{H}$ in the same manner to calculate the phases of a signal. Indeed, choosing $\mathbf{g}$ to be either $\mathcal{G}$ or $\mathcal{H}$, we can compute the phases of a signal $\mathbf{c}_k^w$ by

$$\phi(\mathbf{c}_k^w, \mathbf{g})_j = \arctan \left(\frac{\mathrm{Im}[(\mathbf{g} * \mathbf{c}_k^w)_{j+\left\lfloor \frac{|\mathbf{g}|}{2} \right\rfloor}]}{\mathrm{Re}[(\mathbf{g} * \mathbf{c}_k^w)_{j+\left\lfloor \frac{|\mathbf{g}|}{2} \right\rfloor}]} \right), \quad (3.24)$$

in which $j \in \{0, \ldots, w-1\}$ and $*$ denotes the convolution operator defined earlier. The phase difference between $\mathbf{c}_k^w$ and $\mathbf{d}_k^w$ is defined as

$$\Delta(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_j = \phi(\mathbf{c}_k^w, \mathbf{g})_j - \phi(\mathbf{d}_k^w, \mathbf{g})_j. \quad (3.25)$$

Three different features can be extracted from the recordings. They are measures of synchronization that do not depend on the amplitude of the signals, but instead are related to their respective phases. The first is the mean phase synchrony,

$$\mathsf{PS}_\mu(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g}) = \left| \frac{1}{w} \sum_{j=0}^{w-1} e^{i \cdot \Delta(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_j} \right|, \quad (3.26)$$

which represents the average phase difference between the two signals. Alternately, an index based on the conditional probability (PS_{cp}) and another index based on the Shannon entropy (PS_{se}) can be computed.

To define these, we first need to separate the interval $[0, 2\pi]$ into equidistant bins. The number of such bins is given by

$$L(w) = \left\lceil e^{0.626+0.4\ln(w-1)} \right\rceil, \tag{3.27}$$

as defined in [40]. Let

$$M(\mathbf{c}_k^w, \mathbf{g})_l = \left\{ j : \phi(\mathbf{c}_k^w, \mathbf{g})_j \in \left[\frac{l}{L(w)} 2\pi, \frac{l+1}{L(w)} 2\pi \right] \right\}, \tag{3.28}$$

where $j \in \{0, 1, \ldots, w-1\}$, l is a bin index and $M(\mathbf{c}_k^w, \mathbf{g})_l$ is an index set containing the indices of the elements of $\phi(\mathbf{c}_k^w, \mathbf{g})$ that fall into the l^{th} bin. Then, for each bin, we compute the following value

$$\lambda(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_l = \frac{1}{|M(\mathbf{c}_k^w, \mathbf{g})_l|} \sum_{j \in M(\mathbf{c}_k^w, \mathbf{g})_l} e^{i\phi(\mathbf{d}_k^w, \mathbf{g})_j}, \tag{3.29}$$

which enables us to calculate the index based on conditional probability,

$$\mathsf{PS}_{cp}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g}) = \frac{1}{L(w)} \sum_{l=0}^{L(w)-1} |\lambda(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_l|. \tag{3.30}$$

This index is related to the probability that $\phi(\mathbf{d}_k^w, \mathbf{g})_j$ has a certain value given that $\phi(\mathbf{c}_k^w, \mathbf{g})_j$ fell in a certain bin. As for the index based on the Shannon entropy, we bin the phase differences as

$$P(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_l = \frac{\left| \left\{ j : \Delta(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_j \in \left[\frac{l}{L(w)} 2\pi, \frac{l+1}{L(w)} 2\pi \right] \right\} \right|}{n}, \tag{3.31}$$

where $j \in \{0, 1, \ldots, w-1\}$. Then the index is computed as

$$\mathsf{PS}_{se}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g}) = 1 + \frac{1}{\ln[L(w)]} \sum_{l=0}^{L(w)-1} P(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_l \ln[P(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})_l]. \quad (3.32)$$

The PS_{se} index represents the Shannon entropy of the binned phase differences between the two signals.

3.2.3 Computational Complexity

In this section, we analyze the computational complexity of each feature. This will enable a user to choose which feature configurations to use in the training phase of the algorithm by accounting for the time available to compute them.

We now introduce the Big-Oh notation $O(\cdot)$. We say that a function $f(n)$ is $O(g(n))$ if there exist constants $c > 0$ and $n_0 \geq 0$ such that for all $n \geq n_0$, we have $f(n) \leq cg(n)$ [23]. Note that the Big-Oh notation only provides an upper bound on the growth rate of a function.

The bounds derived for the features are not necessarily tight with the toolbox implementation, since multiple tricks have been used to speed up computations. Nevertheless, these are still useful to get an idea of the computational cost of a feature. For the sake of the calculations, we assume that $|\mathbf{g}| \leq w$, with $w \in \mathbf{W}$, for all possible configurations.

The computational costs of each univariate features on a single $\mathbf{c}_k^w$ are shown in Table 3–1. To account for the total cost of computing the feature for a given feature vector, we need to add the costs for each window size

present in $\mathbf{W}$ and then multiply by the number of channels present in the EEG recording, $|E|$.

Table 3–1: Computational cost of univariate features on a single channel $\mathbf{c}_k^w$.

Univariate Feature	**Computational Cost**
$\mu(\mathbf{c}_k^w)$	$O(w)$
$\sigma^2(\mathbf{c}_k^w)$	$O(w)$
$\mathcal{L}(\mathbf{c}_k^w)$	$O(w)$
$\mathsf{FFT}(\mathbf{h} \circ \mathbf{c}_k^w)$	$O(w \log w)$
$\mathsf{MSC}(\mathbf{c}_k^w, \mathbf{g})$	$O(w \log w)$

The computational cost of each bivariate feature on a pair $(\mathbf{c}_k^w, \mathbf{d}_k^w)$ is given in Table 3–2. Since bivariate feature are computed on all pairs of channels, the total cost of computing them is equal to the summation of the costs for each window size present in $\mathbf{W}$ multiplied by the total number of pairs of channels present in the EEG recording, i.e. the binomial coefficient $\binom{|E|}{2} \sim O(|E|^2)$.

Table 3–2: Computational cost of bivariate features for a single channel pair $(\mathbf{c}_k^w, \mathbf{d}_k^w)$.

Bivariate Feature	**Computational Cost**
$\mathsf{LC}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g}, \mathbf{h})$	$O(w \log w)$
$\mathsf{MCC}(\mathbf{c}_k^w, \mathbf{d}_k^w, r)$	$O(rw)$
$\mathsf{NI}_S(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r)$	$O(dw^2 + w^2 \log w)$
$\mathsf{NI}_H(\mathbf{c}_k^w, \mathbf{d}_k^w, d, \tau, r)$	$O(dw^2 + w^2 \log w)$
$\mathsf{PS}_\mu(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$	$O(w \log w)$
$\mathsf{PS}_{cp}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$	$O(w \log w + wL(w))$
$\mathsf{PS}_{se}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$	$O(w \log w + wL(w))$

From our experimentation with the toolbox, any complexities containing terms higher or equal to $O(w^2)$ are excessive for values of $w \sim 1000$, i.e. 5

seconds sampled at 200Hz. Moreover, the constant terms present in the linear coherence implementation's complexity prohibits its computation when more than 15 channels are present in the EEG recordings.

3.3 Cross-Validation

The cross-validation module is responsible for the separation of the data into training and testing sets in order to correctly estimate the efficiency of the created automated seizure detection algorithms. It does so by separating the ictal events and interictal segments uniformly across the folds. All the other modules of the toolbox are carried independently on each different training set.

3.4 Feature Selection

In the feature extraction phase, we extracted a large amount of features by using multiple different configurations in order to cover a large set of epileptic syndromes and seizures. The computational cost of extracting this ample pool of features was high. However, our goal is to create personalized seizure detection algorithms that run in real-time. Therefore, given a patient, we only need the subset of features that is relevant to his particular epileptic syndrome and seizures. Selecting this smaller subset of features will reduce the computational cost of the feature extraction when performed in real-time as well as reduce the amount of irrelevant features fed to the classifier, possibly increasing its efficiency. The combination of the extraction of the large pool of features with the automated feature selection enables the

toolbox to create personalized seizure detection algorithms without the need for external input on the patient's conditions.

The machine learning literature provides a large number of methods for automatic feature selection [14]. Most of these feature selection techniques can be categorized as filter, wrapper or embedded methods. A common advantage of these methods is that regardless of their simplicity, a rich function class can still be considered when building a classifier. We now discuss both the filter and embedded methods, as techniques from those two feature selection schemes are implemented in the toolbox.

The filter methods consist of ranking each feature individually according to a scoring function and selecting the highest ranked subset. These methods have some attractive properties: they are simple, scalable and have been shown to work well on multiple problems [14]. Indeed, the computational complexity is linear with respect to the cost of computing the scoring function, as a single score has to be calculated per feature. Furthermore, these methods are robust against overfitting [18]. On the other hand, filter methods cannot capture complex relations between features and they are subject to select multiple highly correlated ones. Standard scoring functions are often based on mutual information, correlation or other measures such as the area under the receiver operating characteristic (ROC) curve, where the latter is explained in Section 3.4.1.

As for the embedded methods, they perform feature selection while constructing a classifier. A standard approach in statistical analysis is to impose

a complexity constraint when training a simple classifier; selecting only the features with non-zero weights [18]. This complexity constraint creates a trade-off between the number of selected features and the overall classifier performance. Thus, the features selected are only those that improve the performance of the classifier substantially. The downside of embedded methods is that they often have a higher computational complexity than filter methods. An example of an embedded method, the $\ell 1$-regularized logistic regression, is described in Section 3.4.2.

3.4.1 Area under the ROC curve (AUC)

The area under the receiver operating characteristic (ROC) curve, denoted AUC, is a useful filter method to assign a score for each feature independently when the final task is to perform binary classification. We first need to introduce the necessary concepts in order to understand the ROC curve.

Given a classifier $\widehat{f}$ and a new sample $\mathbf{x}_i$ to classify, there are four possible outcomes considering the true label $f(\mathbf{x}_i)$: true positive, false positive, true negative and false negative. Table 3–3 enumerates these outcomes as well as the conditions for them to occur.

Table 3–3: The possible outcomes of the classification of a new sample $\mathbf{x}_i$ using a classifier $\widehat{f}$ with respect to the true labels.

	$f(\mathbf{x}_i) = 1$	$f(\mathbf{x}_i) = 0$
$\widehat{f}(\mathbf{x}_i) = 1$	True Positive	False Positive
$\widehat{f}(\mathbf{x}_i) = 0$	False Negative	True Negative

From these outcomes, we can define the true positive rate (TPR) and false negative rate (FPR) of a classifier $\widehat{f}$ over a set of samples $\mathbf{X}$ as,

$$TPR(\widehat{f}, \mathbf{X}) = \frac{\left|\{\mathbf{x}_i \in \mathbf{X} : \widehat{f}(\mathbf{x}_i) = 1, f(\mathbf{x}_i) = 1\}\right|}{|\{\mathbf{x}_i \in \mathbf{X} : f(\mathbf{x}_i) = 1\}|}, \tag{3.33}$$

and

$$FPR(\widehat{f}, \mathbf{X}) = \frac{\left|\{\mathbf{x}_i \in \mathbf{X} : \widehat{f}(\mathbf{x}_i) = 1, f(\mathbf{x}_i) = 0\}\right|}{|\{\mathbf{x}_i \in \mathbf{X} : f(\mathbf{x}_i) = 0\}|}, \tag{3.34}$$

where $|A|$ denotes the size of set A. These two measures define the two dimensional ROC space, shown to the right of Figure 3–6. The point located at the origin, $(0, 0)$, corresponds to a classifier that classifies everything as a negative, i.e. $\widehat{f}(\cdot) = 0$. At the other extreme, we have the point $(1, 1)$, which corresponds to a classifier that classifies everything as positive, i.e. $\widehat{f}(\cdot) = 1$. The optimal point is located at $(0, 1)$, where we have a perfect TPR and FPR, i.e. $\forall \mathbf{x}_i \in \mathbf{X}, \widehat{f}(\mathbf{x}_i) = f(\mathbf{x}_i)$. The diagonal line $y = x$ (the dashed line to the right of Figure 3–6) corresponds to random classifiers. Indeed, if we have a classifier that randomly classifies a sample as positive with probability p, its corresponding point in the ROC space would be (p, p).

Now consider a function $g : X \to \mathbb{R}$ instead of $\widehat{f} : X \to Y$. For g to be helpful in determining the class of new samples, its distributions of values must be different when the class is positive or negative. Figure 3–6 to the left depicts an example of the densities of the output of a function g according to the different classes. We can observe that the two distributions overlap, but are not completely the same. Therefore, the function g is capable of partially differentiating the classes. We can create points in the ROC space for the

function g by choosing a threshold t and saying that every value to the left will be classified as negative, and every value to the right as positive, or vice-versa. In fact, we are converting g into a simple classifier called a decision stump. As an example, the threshold depicted to the left of Figure 3–6 for the function g correspond to the ROC point in the right figure. To approximate the full ROC curve of g shown to the right of Figure 3–6, we can vary the threshold so that all the values between $\min(g(\cdot))$ and $\max(g(\cdot))$ are covered.

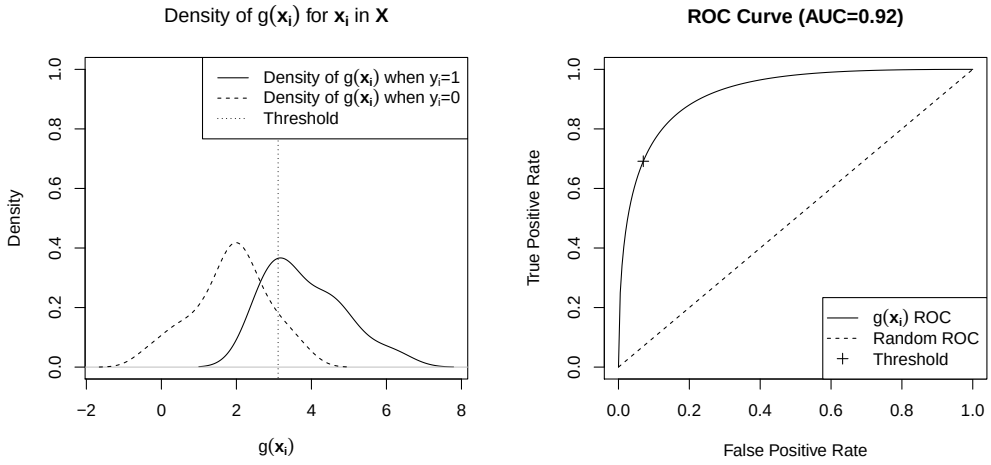


Figure 3–6: Left - Density distribution of the function g according to the different classes. If we consider everything to the left of the threshold as negative and everything to the right as positive, it yields the ROC point denoted in the right figure. Right - ROC space with the ROC curves of both g and a random classifier. The AUC of g is reported.

Once we have a complete ROC curve, we can compute our scoring function: the AUC. Since the curve is embedded into the unit square, we know that AUC of g is less than or equal to 1. We also know that a random classifier has an AUC of 0.5. This ranking measure has an important statistical

property: it is equivalent to the probability that $g(\mathbf{x}_i) \geq g(\mathbf{x}_j)$ for a randomly chosen sample $\mathbf{x}_i \in \mathbf{X}$ such that $f(\mathbf{x}_i) = 1$ and a randomly chosen sample $\mathbf{x}_j \in \mathbf{X}$ such that $f(\mathbf{x}_j) = 0$ [6]. In other words, the higher the AUC, the more g is able to distinguish between the two classes. Moreover, this measure in robust against class imbalances present in the data. Indeed, it is easy to see that given any quantity of samples from both classes, the *TPR* and the *FPR* will stay the same for a given classifier. Note that the AUC is closely related to the Gini coefficient [16].

We can compute the AUC of a feature by considering it as the output of some function g. Since we do not know the true function, we can approximate its density distributions according to each class by using the feature values over all the labelled training data. This enables us to compute its AUC. Once a score is assigned to each feature, we can select a subset of features by picking the best n ranked ones.

3.4.2 $\ell 1$-Regularized Logistic Regression

We now describe the $\ell 1$-regularized logistic regression as an embedded feature selection method. This approach is natural in statistical learning [18]. It uses the logistic function $\sigma_\theta : X \to [0, 1]$ as a simple model. For $\mathbf{x} \in X$, the logistic function is defined as

$$\sigma_\theta(\mathbf{x}) = \frac{1}{1 + e^{\theta^T \mathbf{x}}}, \tag{3.35}$$

where θ are the parameters of the function. It can be shown that given the right set of parameters θ for a model $f(\cdot)$, such that $f(\mathbf{x}) = y$ with $y \in \{0, 1\}$,

$$\sigma_\theta(\mathbf{x}) \approx p(y = 1|\mathbf{x}). \tag{3.36}$$

Indeed, if we let

$$\theta^T \mathbf{x} \approx -\ln\left(\frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x}|y=0)P(y=0)}\right), \tag{3.37}$$

we obtain

$$\begin{aligned}
\sigma_\theta(\mathbf{x}) &= \frac{1}{1 + e^{\theta^T \mathbf{x}}} && (3.38)\\
&\approx \frac{1}{1 + e^{-\ln\left(\frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x}|y=0)P(y=0)}\right)}} && (3.39)\\
&\approx \frac{1}{1 + \frac{P(\mathbf{x}|y=0)P(y=0)}{P(\mathbf{x}|y=1)P(y=1)}} && (3.40)\\
&\approx \frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x}|y=1)P(y=1) + P(\mathbf{x}|y=0)P(y=0)} && (3.41)\\
&\approx \frac{P(\mathbf{x}|y=1)P(y=1)}{P(\mathbf{x})} && (3.42)\\
&\approx \frac{P(\mathbf{x}, y=1)}{P(\mathbf{x})} && (3.43)\\
&\approx P(y=1|\mathbf{x}), && (3.44)
\end{aligned}$$

as required. The set of optimal parameters θ can be estimated from a set of training samples $\mathbf{X}$ and corresponding set of labels $\mathbf{Y}$ by maximizing the log likelihood of the labels given the data with respect to the parameters, i.e.

$$\underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{x}_i \in \mathbf{X}} -\log[p(y_i|\mathbf{x}_i; \theta)]. \tag{3.45}$$

Unfortunately, the set of parameters found during this optimization might contain only non-zero entries; using all the available features in $\mathbf{x}$.

To remedy this problem, we can use an interesting complexity constraint called the $\ell 1$ norm:

$$\|\theta\|_1 = \sum_{\theta_j \in \theta} |\theta_j|, \tag{3.46}$$

which measures the density of θ. We modify the optimization function of θ to include this complexity constraint,

$$\underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{x}_i \in \mathbf{X}} -\log[p(y_i|\mathbf{x}_i;\theta)] + \lambda\|\theta\|_1, \tag{3.47}$$

where λ is a parameter controlling the complexity penalty. The optimization function is now $\ell 1$-regularized. The parameter λ indicates how much to penalize the parameter vector θ according to its $\ell 1$ norm. Indeed, if $\lambda = 0$, no penalization is applied, and we retrieve the original optimization function. On the other hand, for larger values of λ, the regularization parameter has more influence during the optimization. Therefore, sparser θ vectors are preferred, i.e. vectors with few non-zero entries.

The $\ell 1$-regularized logistic regression has two main advantages. First, it is efficient at selecting a small subset of useful features from a pool containing a large proportion of irrelevant ones [33]. Second, highly correlated features are often not selected during the optimization, providing a more diverse feature set.

Note that iterative methods are often used to solve the regularized optimization because of the large number of features for which we want to learn

the set of parameters θ [27]. In our case, the $\ell 1$-regularized logistic regression was implemented with the use of the Vowpal Wabbit (VW) toolbox [26]. Vowpal Wabbit was designed for computational efficiency when training on large datasets, using a modified online gradient descent algorithm for learning.

3.5 Classification

The classification module is responsible for creating classifiers using the selected features and the training data. The goal is to train a classifier $\widehat{f}$ using the selected features and use it to get good approximations of the labels $\widehat{y} = \widehat{f}(\mathbf{x})$ for new samples $\mathbf{x} \in X$. We implemented two types of classifiers able to represent different families of functions. The first one is a $\ell 1$-regularized logistic regression [18], able to represent linear functions, and the second one is a forest of Extra-Trees [11], able to represent a much more complex family of functions. We describe the two methods as well as a way to define thresholds in order to account for the data imbalance present in EEG recordings.

3.5.1 $\ell 1$-Regularized Logistic Regression

The $\ell 1$-regularized logistic regression was explained in Section 3.4.2. It is important to note that for the right choice of λ, the complexity constraint prevents the optimization function from overfitting the training data. Indeed, it reduces the complexity of the model created by reducing the number of non-zero entries in the set of parameters θ. Even though the family of functions

this classifier is able to represent is fairly simple, it is really useful in order to validate the use of more complex classifiers.

3.5.2 Extra-Trees

Extra-Trees are a type of extremely randomized binary decision trees introduced by Geurts et al. in [11]. A binary decision tree is a binary tree where each inner node contains a test and each leaf determines a label $y \in Y$. A test is a function $T : X \to \{\text{True}, \text{False}\}$ defined by a threshold t and a feature j such that given a feature vector $\mathbf{x} \in X$,

$$T(\mathbf{x}_i) = \begin{cases} \text{True} & \text{if } \mathbf{x}_j \geq t \\ \text{False} & \text{otherwise} \end{cases} . \tag{3.48}$$

Using a decision tree, we can classify a sample $\mathbf{x} \in X$ by running tests on $\mathbf{x}$ until we reach a leaf node. The path taken by the sample $\mathbf{x}$ from the root to the leaf is determined by the outcome of the tests contained in its inner nodes: if the output of a test is false, we go into the left subtree and if it is true we go into the right subtree. The label assigned to sample $\mathbf{x}$ is the one contained in the leaf node it reached.

Assuming the features contain significant information about the class label, the usefulness of a decision tree resides in the way that we define its tests at each inner node. These are often chosen by optimizing an information theoretic criterion: the information gain. Since Extra-Trees use a particular normalization of the information gain, we will only introduce the modified version. Let $\mathbf{X}$ be our set of training samples with corresponding set of labels

$\mathbf{Y}$. We first define the multiset $T(\mathbf{X})$ to be

$$T(\mathbf{X}) = \{T(\mathbf{x}_i) : \mathbf{x}_i \in \mathbf{X}\}, \tag{3.49}$$

containing each outcome of T on the samples of $\mathbf{X}$. The scoring function $\mathcal{S}(T, \mathbf{X}, \mathbf{Y})$ is defined as

$$\mathcal{S}(T, \mathbf{X}, \mathbf{Y}) = \frac{2\mathcal{I}(T(\mathbf{X}); \mathbf{Y})}{H(T(\mathbf{X})) + H(\mathbf{Y})}, \tag{3.50}$$

where $H(T(\mathbf{X}))$ is the log entropy of $T(\mathbf{X})$, $H(\mathbf{Y})$ is the log entropy of $\mathbf{Y}$ and $\mathcal{I}(T(\mathbf{X}); \mathbf{Y})$ is the mutual information between $T(\mathbf{X})$ and $\mathbf{Y}$. Let $p(a)$ be the probability of an event a, $B = \{\text{True}, \text{False}\}$ be the set of outcomes of a test, $\mathbf{x}$ be a random element of $\mathbf{X}$ and $\mathbf{y}$ be a random element of $\mathbf{Y}$, then

$$H(T(\mathbf{X})) = -\sum_{b \in B} p(T(\mathbf{x}) = b) \log[p(T(\mathbf{x}) = b)], \tag{3.51}$$

$$H(\mathbf{Y}) = -\sum_{y \in Y} p(\mathbf{y} = y) \log[p(\mathbf{y} = y)], \tag{3.52}$$

and finally

$$\mathcal{I}(T(\mathbf{X}); \mathbf{Y}) = \sum_{b \in B, y \in Y} p(T(\mathbf{x}) = b, \mathbf{y} = y) \log \left[\frac{p(T(\mathbf{x}) = b, \mathbf{y} = y)}{p(T(\mathbf{x}) = b) p(\mathbf{y} = y)} \right]. \tag{3.53}$$

Standard decision trees are created using a top down recursive approach. We first consider the root node, with the full training set $(\mathbf{X}, \mathbf{Y})$. We select a test T by maximizing the scoring function over all possible thresholds and

features. Then, we split the training data according to the output of the test:

$$L(\mathbf{X}) = \{\mathbf{x}_i \in \mathbf{X} : T(\mathbf{x}_i) = \text{False}\}, \tag{3.54}$$

$$R(\mathbf{X}) = \{\mathbf{x}_i \in \mathbf{X} : T(\mathbf{x}_i) = \text{True}\}. \tag{3.55}$$

Finally, we repeat the procedure for the left subtree using $L(\mathbf{X})$ as the training data and for the right subtree using $R(\mathbf{X})$. This procedure continues until all the labels in the remaining training data are the same, where we create a leaf and assign it that label.

Extra-Trees differ in the way they are created. Indeed, during the top down recursion, not all possible thresholds and features are considered. In fact, K tests are created by randomly picking a non-constant feature and a threshold uniformly at random across that feature range. Then, each test is scored according to $\mathcal{S}$ and the highest ranked is selected as that inner node test. As with the standard trees, the training data is then split according to the test and the procedure is carried out on the two subtrees. However, the stopping criteria is also different: we stop when all the training data labels are the same or when the size of the training set reaches a lower limit $n_{\min}$.

It is interesting to consider the effects of the parameters K and $n_{\min}$ when creating an Extra-Tree. The parameter K controls the randomization of the tree, where lower values of K induce higher randomization. This increased randomization causes the tree to have a higher bias error, but a lower variance error. The parameter $n_{\min}$ averages the noise present in the output variables. Indeed, higher values induce an earlier stopping criteria

where the label of the leaf is decided across a larger set of training example. The resulting tree is also shorter, increasing the bias error and lowering the variance error.

The main source of prediction error in standard decision trees is caused by the variance error. Effectively, it is principally the result of overfitting the inner tests to the training data [10]. To consider the prediction error of an Extra-Tree, we first need to look at the default parameters defined in the original paper: $K = \sqrt{|F|}$ and $n_{\min} = 2$, where $|F|$ is the number of features present in each training point [11]. These default parameters cause an Extra-Tree to have higher variance and a slight increase in bias compared to standard decision trees, making their prediction error worse. A natural question comes to mind: why use an Extra-Tree?

The trick resides in the cause of the variance of an Extra-Tree: the randomization of the tests. Instead of using a single Extra-Tree, we create an ensemble of M trees, called a forest. To classify a new sample, we ask each tree in the forest to label the sample and return the proportion labelling the example for each possible class. By choosing the label to be the one in highest proportion, the forest is actually averaging out the variance error of each Extra-Tree [11]. For higher values of M, we have a higher reduction in the variance error. Note that for randomized methods, increasing M will never cause the ensemble to overfit the data, as the expected prediction error is a monotonically decreasing function with respect to M [2]. The resulting forest still has a slightly higher bias error than a standard decision tree, but

its variance error almost completely disappears. Therefore, forests of Extra-Trees, or simply Extra-Trees, usually outperform standard decision trees as their prediction error is much lower, thanks to a much lower variance error.

Another advantage of Extra-Trees over standard decision trees is the computational cost to create them. Both methods have a cost of $O(n \log(n))$, assuming balanced trees and n training samples. However, it is important to consider the constant factor induced by the choice of tests at each inner node. We expect it to be much lower for Extra-Trees as only K tests are considered, versus the full spectrum for standard decision trees. These trees are also fast to query, requiring $O(n \log(n))$ operations when balanced. This enables their use in real-time settings.

3.5.3 Threshold Selection

The two classifiers that can be created by the toolbox have real valued outputs. Indeed, the logistic function is defined as $\sigma_\theta : X \to [0, 1]$ and the Extra-Trees, E, return the proportion of the forest that labels an example as each class, i.e. $E : X \to [0, 1]^{|Y|}$. Since we are considering binary classification, the output of the Extra-Trees can be denoted as $E : X \to [0, 1]$, where the returned value is the proportion of trees that label the sample as ictal.

When designing seizure detection algorithms, we are faced with the class imbalance problem [19]. It occurs when the proportion of training samples between the two classes that we are trying to learn differ by a large amount. Indeed, the number of samples present during ictal events is magnitudes lower than the number present during interictal data. To help remedy to the

problem, we pick a non-standard threshold to apply on our classifiers in order to define the final classification. Normally, for balanced classes, a standard threshold with value 0.5 is used. Indeed, a logistic regression combined with this threshold will return the class that is most probable and the Extra-Trees will return the class that is in majority across the forest. In our case, this threshold may be suboptimal as the classes are highly imbalanced. For logistic regression, ictal samples have a lower influence on the optimized set of parameter θ, because they are in much lower proportion compared to interictal samples. As for the Extra-Trees, it is much more difficult for a leaf to contain a majority of ictal samples, as these are dispersed amongst a sea of interictal samples. By choosing a different threshold, we help the classifier by putting more importance on ictal samples.

Our methodology to find this non-standard threshold is rather simple. Consider a classifier $g : X \to Y$ and a set of training data $\mathbf{X}$. Let

$$\mathbf{X}^- = \{\mathbf{x}_i \in \mathbf{X} : \, g(\mathbf{x}_i) = 0\}, \tag{3.56}$$

$$\mathbf{X}^+ = \{\mathbf{x}_i \in \mathbf{X} : \, g(\mathbf{x}_i) = 1\}. \tag{3.57}$$

We select the threshold t_g to be

$$t_g = \frac{\mu(g(\mathbf{X}^-)) + \mu(g(\mathbf{X}^+))}{2}, \tag{3.58}$$

where

$$\mu(g(\mathbf{X})) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x}_i \in \mathbf{X}} g(\mathbf{x}_i), \tag{3.59}$$

is the mean of the output of g on samples in $\mathbf{X}$. We can then define the final classifier $\widehat{f}_g$ as

$$\widehat{f}_g(\mathbf{x}) = \begin{cases} \textit{ictal} & \text{if } g(\mathbf{x}_i) \geq t_g \\ \textit{interictal} & \text{otherwise} \end{cases} . \tag{3.60}$$

This final classifier is more sensible to ictal events as instead of using an absolute threshold of 0.5, it picks the midpoint between the mean output of the classifier over the two classes. It can be thought of as the equivalent of a 0.5 threshold, but for shifted distributions. The output of these classifiers will be used in order to raise alarms for the automated detection of ictal events.

3.6 Performance Analysis

The performance analysis module is responsible for the computation of the metrics used to validate the efficiency of the personalized automated seizure detection algorithms generated by the toolbox. Recall that a complete automated seizure detection algorithm consists of a combination of a feature extraction module, a classification module and an alarm control module, as depicted in Figure 2–2. The first two components have been described in sections 3.2 and 3.5, respectively. We will now cover the alarm control module as well as the performance metrics used to validate the efficiency of the automated seizure detection algorithms generated by the toolbox.

3.6.1 Alarm Control

Recall that the output of a classifier $\widehat{f}$ on a feature vector $\mathbf{x} \in X$ is a label of *ictal* or *interictal* as defined in Section 2.3. Since the types of classifiers considered in the toolbox do not take into account their previous classifications when classifying a new feature vector, their output might be highly sensitive to small changes in the EEG recordings. Therefore, raising an alarm for each *ictal* classifications would cause a high false positive rate. To remedy this problem, an alarm is issued only after a minimum number of consecutive *ictal* classifications are made by the classifier and is maintained until the first *interictal* classification. We call the minimum number of classifications the *minimum trigger length* (MTL) and it is a tunable parameter of the toolbox. We can increase the MTL until a maximum acceptable false positive rate is achieved and then observe the corresponding detection rate of the algorithm. It is important to note that an alarm has a starting time and a duration for which it is on. Figure 3–7 shows an example of the output of a classifier combined with MTL of 5 on a segment of EEG recording containing a seizure.

3.6.2 Performance Measures

To evaluate the performance of the generated algorithms, we consider the following three definitions:

i. *Detection:* An alarm issued by the algorithm is deemed a true detection if it overlaps an ictal event and the first *ictal* classification occurred no

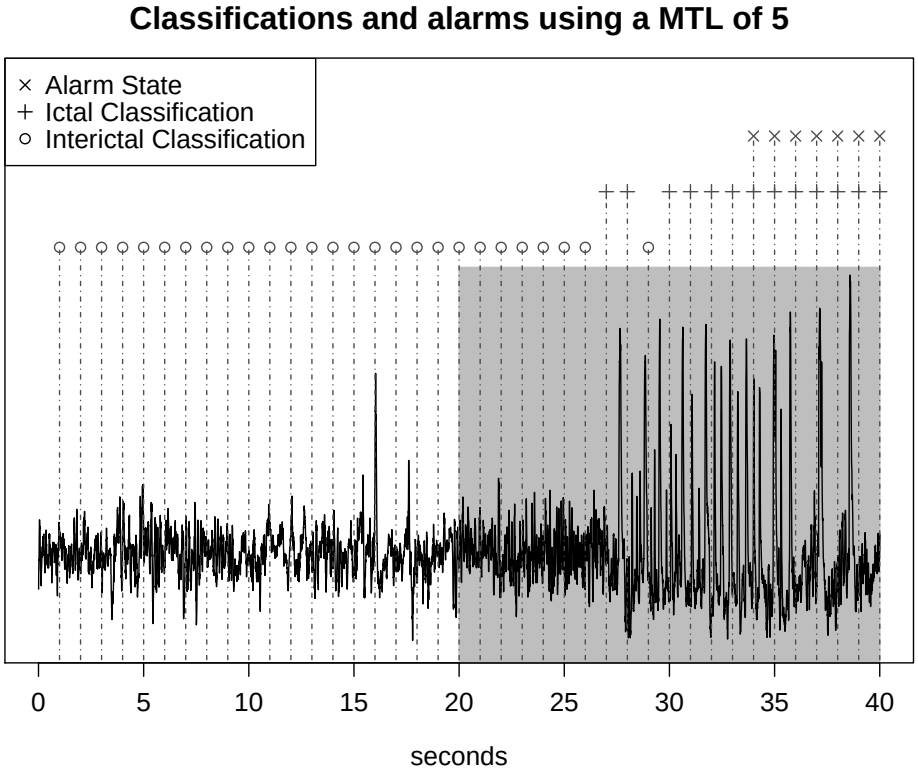


Figure 3–7: Example of the effect of a minimum trigger length of 5 on a segment of EEG recording containing a seizure in the greyed out area. The alarm is only raised after the 34^{th} feature vector is classified, when the 5^{th} consecutive *ictal* classification is made. For simplicity, only a single channel of the EEG is displayed.

more than 30 seconds before the start of the ictal event. If a single alarm overlaps 2 ictal events, only the first one is deemed detected.

ii. *False Positives:* Each alarm present in an EEG recording segment containing no ictal events counts as a false positive.

iii. *Latency:* The latency is the time between the start of the ictal event and the beginning of the earliest alarm that successfully detects this event.

Note that if the alarm begins before the start of the seizure, the latency is considered to be 0.

These metrics are evaluated for each personalized automated seizure detection algorithms generated by the toolbox. For a given patient and for each different fold of the cross-validation, these quantities are measured over every EEG segment not included in the training data. The final performance measures of an algorithm are then averaged across each fold of the cross-validation, as is described in Section 2.1.4.

3.7 Discussion

In this chapter we described each of the modules present in the toolbox in details. First, we covered the simple data format required by the toolbox, facilitating its use across different datasets. Then, the features implemented were defined in details and a computational cost analysis for each of them was provided. Both the definitions and analysis will help the community choose which features to use if constrained by computational reasons during the training phase. Two different feature selection techniques were presented in order to compare a simpler filtering method with interesting properties, the AUC, to a more complex embedded method, the $\ell 1$-regularized logistic regression. The classification module contained two classifiers, the simpler logistic model as well as a more complex classifier, the Extra-Trees. The trees have multiple interesting properties: they are fast to train and query, and are robust against overfitting. Moreover, we presented a technique to choose a threshold that helps account against the class imbalance present

between the ictal events and the interictal data. Also, the minimum trigger length, a tunable parameter, was introduced to enable the specification of a maximum acceptable false positive rate in order to observe the corresponding algorithm performance. Finally, standard performance metrics calculated by the toolbox were presented.

The toolbox could be improved by implementing an even broader set of features defined by the seizure detection/prediction community. Indeed, increasing the diversity of the features would enable to cover a larger set of epileptic syndromes and seizures, improving the performance of the generated algorithms. Moreover, temporal information could be incorporated in the feature vectors by concatenating parts of previously computed vectors. The current strategy of the toolbox is to increase the size of the windows on which the features are extracted in order to capture information present in a longer history of EEG data. This method is more stable with respect to changes present in the EEG signals whereas the concatenation would provide the same higher sensitivity that is present in shorter windows, but with a temporal evolution. This would help reduce the number of false positives caused by artifacts and by interictal epileptiform discharges as well as possibly increase the sensitivity of the generated seizure detection algorithms.

Other classifiers, such as artificial neural networks, support vector machines, etc. could be incorporated in the toolbox. These would enable a broader set of classification techniques to be used in order to find which are

more resilient to the difficulties present when performing automated seizure detection.

Finally, if datasets containing more seizures per patient were available, it would be interesting to modify the toolbox such that the minimum trigger length parameter is selected during the training phase of the cross-validation. This modification would help us better estimate the true performance of the generated algorithms in a real clinical environment.

CHAPTER 4
Validation and Results

This chapter will cover the experimental methodology and the toolbox configuration used to evaluate the toolbox performance. The metrics are measured on the personalized epileptic seizure detection algorithms generated for subjects present in three different datasets. One of these datasets is newly introduced and made publicly available. Finally, the performance of the toolbox is presented for each dataset independently.

4.1 Experimental Methodology

This section will describe in details how the experiments were conducted using the toolbox.

4.1.1 Dataset Preparation

First, every EEG recording is split into non-overlapping two minutes segments (Figure 4–1-b). Then, any ictal event split across multiple segments is merged back together by joining the corresponding segments (Figure 4–1-c). Finally, any segment containing an ictal event is padded with two extra segments (4 minutes) at the beginning and at the end, when possible (Figure 4–1-d). See Figure 4–1 for a detailed explanation of the procedure. This segmentation procedure was performed for two reasons: it lowers the memory requirements during the feature extraction phase of the toolbox and

it helps balancing the amount of *ictal* and *interictal* data contained in the training folds of the cross-validation.

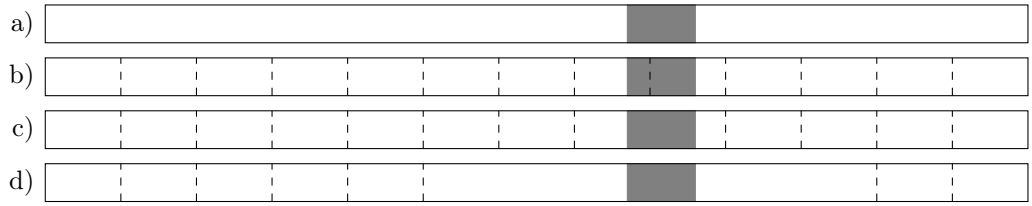


Figure 4–1: Description of the data preparation procedure used for the experimentation. Segments are separated by the dashed horizontal lines. a) An EEG recording containing a seizure marked in gray. b) Segmentation of the EEG into two minutes segments. c) Merging of segments containing ictal events. d) Two segments (four minutes) padding before and after segments containing ictal events.

4.1.2 Feature Extraction

Let f_s be the sampling frequency of the EEG recordings. Features were extracted over a set of windows lengths $\mathbf{W} = \{f_s, 2f_s, 5f_s\}$ (1,2,5 seconds) with a delay of $\delta = f_s$ (1 second) between the end of each consecutive set of windows. The following feature configurations were used:

- $\mu(\mathbf{c}_k^w)$
- $\sigma(\mathbf{c}_k^w)$
- $\mathcal{L}(\mathbf{c}_k^w)$
- $\mathsf{FFT}(\mathbf{h} \circ \mathbf{c}_k^w)_l$ where $\mathbf{h}$ is a Hann window and $l \in \{1, 2, \ldots, 100\}$.
- $\mathsf{MSC}(\mathbf{c}_k^w, \mathbf{g})$ where $\mathbf{g}$ is a finite impulse response filter corresponding to the Daubechies 4 wavelet decomposition at levels $\{1, 2, \ldots, 5\}$.
- $\mathsf{LC}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g}, \mathbf{h})_l$ where $\mathbf{g} = (1, 4, 6, 4, 1)$, $\mathbf{h}$ is a Hann window and $l \in \{1, 2, \ldots, 100\}$.
- $\mathsf{MCC}(\mathbf{c}_k^w, \mathbf{d}_k^w, r, t)$ where $r = f_s$ and $t = 5$.

- $\mathsf{PS}_{\mu}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$,

  $\mathsf{PS}_{cp}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$,

  $\mathsf{PS}_{se}(\mathbf{c}_k^w, \mathbf{d}_k^w, \mathbf{g})$, where

$$\mathbf{g} \in \left\{ \begin{array}{l} \mathcal{H}(w) \\ \mathcal{G}(2, 2, 0.95, f_s) \\ \mathcal{G}(5.5, 1.5, 0.95, f_s) \\ \mathcal{G}(10, 3, 0.95, f_s) \\ \mathcal{G}(14, 1, 0.95, f_s) \\ \mathcal{G}(22, 8, 0.95, f_s) \\ \mathcal{G}(37, 7.5, 0.95, f_s) \\ \mathcal{G}(72.5, 27.5, 0.95, f_s) \end{array} \right\}$$

The phase synchrony (PS) features were only extracted using windows of length f_s. For the CHB-MIT dataset, presented in Section 4.2.3, we did not extract the linear coherence (LC) feature, because of the high number of channels present in the EEG recordings.

4.1.3 Cross-Validation

To correctly measure the performance of the toolbox, we followed a 3-folds cross-validation scheme for each patient. Patients with less than three seizures were dropped from the experimentation as not enough seizures were present to create three folds. For the other patients, we randomly separated the segments containing ictal events across three sets. Then, we selected uniformly at random an amount of interictal segments equal to twice the total duration of the ictal segments. We also separated these segments across

the three sets. The training data used for each fold consisted of the segments present in a pair of sets, such that all possible pairs were considered. Figure 4–2 shows an example of a 3-folds cross-validation performed on a fictitious patient. Again, all the following phases of the toolbox were carried independently on each fold of each patient.

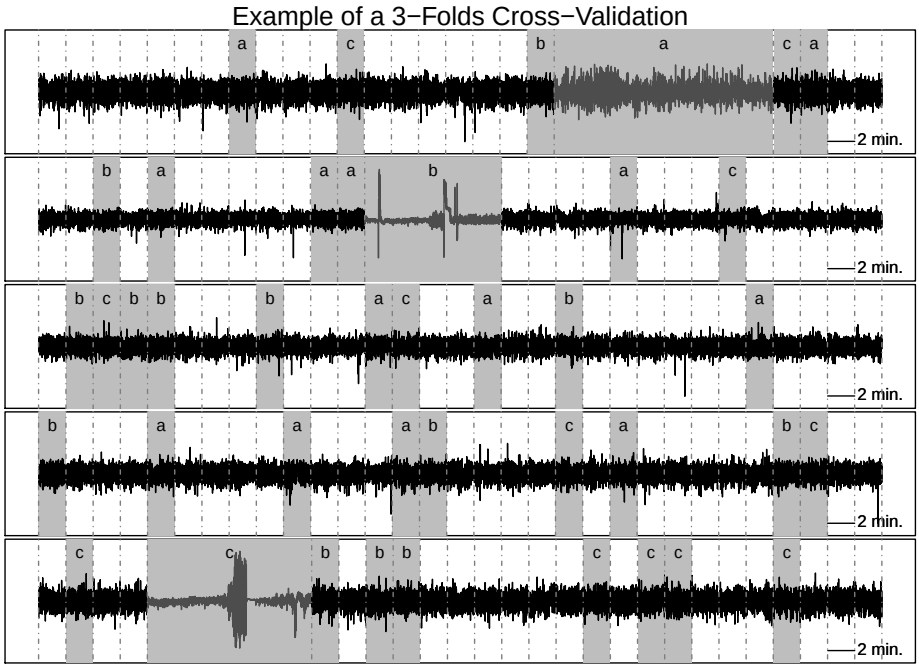


Figure 4–2: Example of a 3-folds cross-validation on a fictitious patient. For simplicity, a single channel of the EEG recording is shown. Segments are separated by dashed lines, and they are placed in a random ordering. The signal of segments containing an ictal event is gray. Segments with a black signal and a shaded background were selected uniformly at random across the ones not containing an ictal event. Each class $\{ictal, interictal\}$ of shaded segments were distributed uniformly across the tree sets $\{a, b, c\}$. The training was performed on all segments contained in a pair of sets and the performance was measured on all the segments not contained in these two sets. This process was repeated three times, in order to consider all possible pairs of training sets.

4.1.4 Feature Selection

- AUC: We computed the value of the AUC for each feature. Then, we selected the top 200 ranking features as the feature set.
- $\ell 1$-Regularized Logistic Regression: We configured the Vowpal Wabbit toolbox using the following set of parameters:

```
vw --loss_function logistic --exact_adaptive_norm
--passes 10 --learning_rate 5 --decay_learning_rate 0.90
--l1 1.0E-5
```

  These parameters tell the Vowpal Wabbit to optimize a $\ell 1$-regularized logistic regression with the complexity parameter $\lambda = 10^{-5}$. The use of the exact adaptive norm improves the quality of the learned parameters when many features are present. The passes argument tells the toolbox to go over all the training data up to ten times, in order to get a better fit. The decay learning rate parameter decreases the original learning rate by a factor of 0.90^{n-1} at the n^{th} pass. The features corresponding to parameters of non-zero weights were selected as the feature set.

4.1.5 Classification

- $\ell 1$-Regularized Logistic Regression: We used the same configuration as for the feature selection, see Section 4.1.4.
- Extra-Trees: We built forests of $M = 200$ Extra-Trees using the default parameters $K = \sqrt{|F|}$ and $n_{\min} = 3$, where $|F|$ corresponds to

the number of features used. At each fold, three forests were built: one using all the features that were extracted, another using only the features selected by the AUC, and the last one using the features selected by the $\ell 1$-regularized logistic regression.

4.2 Datasets

To validate the performance of the personalized automated seizure detection algorithms generated by the toolbox over a large variety of epileptic syndromes and seizures, we used three publicly available datasets. We now describe their characteristics into details.

4.2.1 Dataset 1: Montréal Neurological Institute (MNI)

A rat pilocarpine model of temporal lobe epilepsy was used to collect ECoG data [28, 29]. This animal model of epilepsy is highly isomorphic to human epilepsy [4]. A one hour status epilepticus (continuous stage 5 seizures [39]) was induced in six Sprague-Dawley rats (250-300g) by intraperitoneal injection of pilocarpine (380mg/kg). Then, three days after, surgery was performed to place intracranial bipolar electrodes in the CA3 region of ventral hippocampus, the medial entorhinal cortex, the ventral subiculum, and the dentate gyrus for rats 45-5, 46-5, and 50-9. For rats 38-5, 39-3 and 39-8, electrodes were placed in both the left and right regions of the CA3, the medial entorhinal cortex and the amygdala. All procedures were approved by the Canadian Council of Animal Care and all efforts were made to minimize the number of animals used and their suffering. The recordings were

downsampled to 200Hz without filtering and digitized using a 16 bit analog-to-digital converter. Random recordings between the 4^{th} and 15^{th} day after injection were used, for a total of 475 hours of data containing 137 seizures (see Table 4–1 for the data distribution per rat). This new dataset is freely available at `www.cs.mcgill.ca/~gsauln/files/mni_dataset/`.

Table 4–1: Distribution of ECoG data (MNI).

Patient	**Ictal Segments**	**Inter-ictal Segments (2 min. each)**
rat38-5	18	2953
rat39-3	20	4908
rat39-8	47	4769
rat45-5	7	262
rat46-5	31	620
rat50-9	14	42
Total	137	13554

4.2.2 Dataset 2: Freiburg

The Freiburg epilepsy dataset [8] consists of patients suffering from medically intractable focal epilepsy at the Epilepsy Center of the University Hospital of Freiburg, in Germany. Recordings from three focal and three extra-focal intracranial electrodes from subjects undergoing presurgical monitoring are available. The ECoG signals are sampled at 256Hz and digitized using a 16 bit converter. No filters were applied to the recordings. We analyzed data from 18 of the 21 available patients due to a lack of seizures for patients 002, 008 and 013 (less than 3 seizures). A total of 450 hours of ECoG recordings containing 79 seizures were processed for the experiments (see Table 4–2 for the data distribution).

Table 4–2: Distribution of ECoG data (Freiburg).

Patient	**Ictal Segments**	**Inter-ictal Segments (2 min. each)**
pat001	4	720
pat003	5	720
pat004	5	720
pat005	5	720
pat006	3	720
pat007	3	738
pat009	5	716
pat010	4	732
pat011	4	720
pat012	4	743
pat014	4	714
pat015	4	720
pat016	5	720
pat017	5	721
pat018	5	746
pat019	4	731
pat020	5	767
pat021	5	718
Total	79	13086

4.2.3 Dataset 3: CHB-MIT

The CHB-MIT dataset [45], freely available on Physionet [12], contains EEG recordings from 24 pediatric patients from the Children's Hospital of Boston. Non-invasive scalp recordings were made during the monitoring of patients after their withdrawal from anti-seizure medication in order to characterize their seizures and assess the possibility of surgical intervention. The signals were sampled at 256Hz with a 16 bit resolution. For some patients, the set of available electrodes changed between recordings. Therefore we chose the largest subset of electrodes common to a maximum number of recordings

for the analysis. On average, 21 channels were used per patient. A total of 824 hours of data containing 185 seizures were analyzed (see Table 4–3 for the data distribution).

Table 4–3: Distribution of EEG data (CHB-MIT).

Patient	**Ictal Segments**	**Inter-ictal Segments (2 min. each)**
chb01	7	1017
chb02	3	990
chb03	7	930
chb04	4	4357
chb05	5	1020
chb06	10	1225
chb07	3	1739
chb08	5	450
chb09	4	1746
chb10	7	1080
chb11	3	959
chb12	27	330
chb13	12	750
chb14	8	570
chb15	20	780
chb16	10	390
chb17	3	540
chb18	6	900
chb19	3	810
chb20	8	660
chb21	4	869
chb22	3	840
chb23	7	526
chb24	16	278
Total	185	23756

4.3 Results

The effect of the minimum trigger length on the trade-off between the detection rate and the false positive rate for all 3 datasets is illustrated in

Figures 4–3, 4–5 and 4–7. As expected, an increase of the MTL decreases both the false positive rate and the detection rate. Effectively, increasing the MTL requires the algorithm to make *ictal* classifications for a longer period of time before raising an alarm. By fixing the MTL according to a maximum acceptable false positive rate, we are able to obtain the corresponding detection rate of the algorithm. In Figures 4–3, 4–5 and 4–7, the marks correspond to the smallest MTL such that the median false positive rate is less than 2 per day, or 0.08 per hour.

Figures 4–3, 4–5 and 4–7 also demonstrate the superior performance of the Extra-Trees, both for the detection and false positive rate, compared to the logistic regression. We can also compare the effectiveness of the different feature selection methods.

4.3.1 Dataset 1: MNI

The Extra-Trees using the AUC feature selection and a MTL of 13 are able to detect 118 out of the 137 seizures, which represents a 86.1% sensitivity. The corresponding median false positive rate is 0.071 per hour or 1.704 per day. The average latency of the detections is 24.9 seconds. A per rat analysis of this configuration is provided in Figure 4–4. A perfect sensitivity was achieved for rat39-3 and rat45-5, and no false positives occurred for rat45-5 and rat50-9. The AUC feature selection picked 5.96% of the 3354 available features and the $\ell 1$-regularized logistic regression picked 2.06% on average.

4.3.2 Dataset 2: Freiburg

The Extra-Trees using the AUC feature selection and a MTL of 6 are able to detect 60 out of the 79 seizures, which represents a 75.9% sensitivity. The corresponding median false positive rate is 0.053 per hour or 1.272 per day. The average latency of the detections is 15.6 seconds. A per patient analysis of this configuration is provided in Figure 4–6. A perfect sensitivity was achieved for 8 of the 18 patients and no false positives occurred for 6 of them. A low sensitivity (≤ 0.5) was obtained for 4 patients and a high false positive rate (≥ 0.5) for another. Removing these patients, as the algorithm performs under our expectations for them, would yield a sensitivity of 89.5% and an average and median false positive rate per hours of 0.092 and 0.060, respectively. The AUC feature selection picked 2.82% of the 7089 available features and the $\ell 1$-regularized logistic regression picked 1.52% on average.

4.3.3 Dataset 3: CHB-MIT

The Extra-Trees using the AUC feature selection and a MTL of 7 are able to detect 144 out of the 185 seizures, which represents a 77.8% sensitivity. The corresponding median false positive rate is 0.076 per hour or 1.824 per day. The average latency of the detections is 14.2 seconds. A per patient analysis is provided in Figure 4–8. A perfect sensitivity was achieved for 15 of the 24 patients and no false positives occurred for 5 of them. A low sensitivity (≤ 0.5) was obtained for 3 patients and a high false positive rate (≥ 0.5) for 5 of them. Removing these patients, as the algorithm performs under our expectations for them, would yield a sensitivity of 91.4% and an average and

median false positive rate per hours of 0.101 and 0.068, respectively. The AUC feature selection picked 1.17% of the 17061 available features and the $\ell 1$-regularized logistic regression picked 1.65% on average.

4.3.4 Feature Selection

Figures 4–9 depicts the average proportion of features selected across patients by both feature selection methods for the MNI, Freiburg and CHB-MIT datasets. The horizontal lines present in those figures represent the maximum proportion achievable assuming 200 features are selected. The AUC feature selection picks principally FFT and MSC features in all datasets. The $\ell 1$-regularized logistic regression selects mostly FFT, MSC and LC features for the MNI and Freiburg datasets. As for the CHB-MIT dataset, the method selects mainly FFT and PS features. More than 90% of the features selected by the AUC method are univariate features, whereas the $\ell 1$-regularized logistic regression selects under 60% of them. Table 4–4 contains the proportions of univariate and bivariate features selected by both methods across the three datasets.

Table 4–4: Average proportions of univariate and bivariate features selected for both feature selection methods across all three datasets.

Dataset	**Feature Selection**	**Univariate** (%)	**Bivariate** (%)
MNI	AUC	100.0	0.0
	$\ell 1$-regularized logistic regression	59.6	40.4
Freiburg	AUC	90.8	9.2
	$\ell 1$-regularized logistic regression	52.3	47.7
CHB-MIT	AUC	95.7	4.3
	$\ell 1$-regularized logistic regression	35.0	65.0

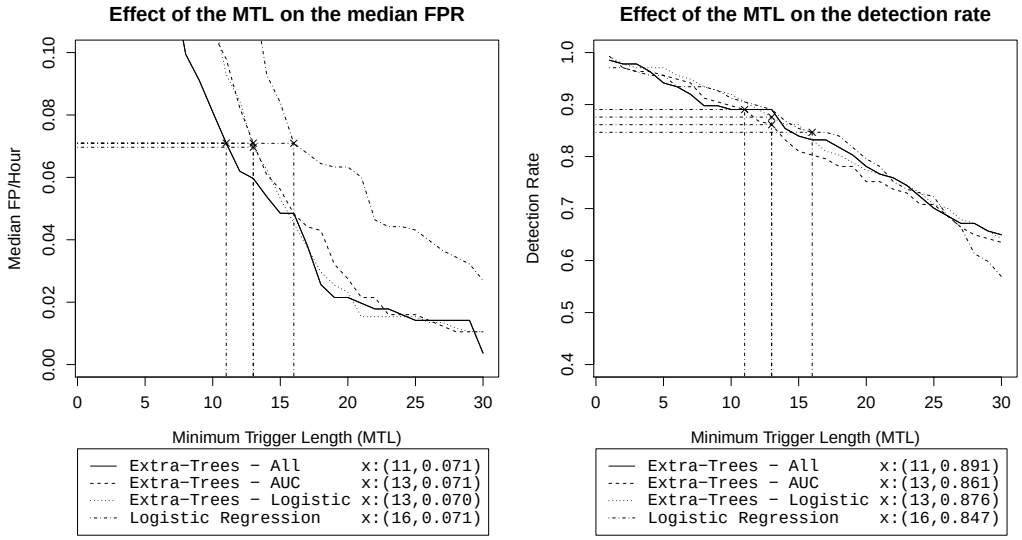


Figure 4–3: MNI: Effect of the Minimum Trigger Length. The curves in the two graphs correspond to the evolution of the median false positive rate and the detection rate with respect to the increase of the MTL. The marks show the smallest MTL such that a false positive rate of less than 2 per day is achieved.

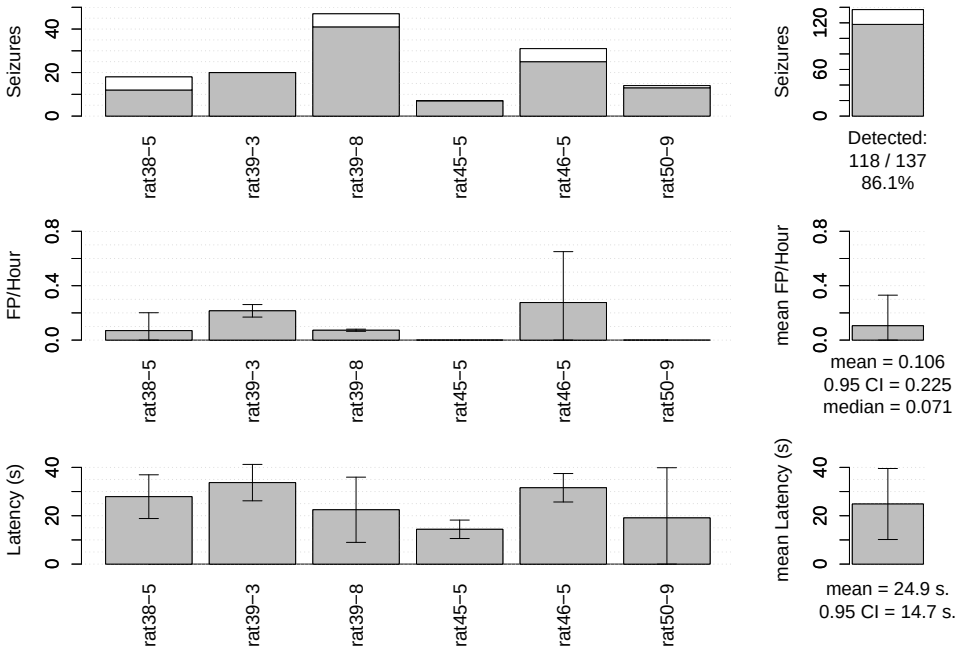


Figure 4–4: MNI: Performance of Extra-Trees trained on features selected by their AUC, using a MTL of 13. A per patient analysis is presented on the left and the overall statistics across all patients is presented to the right. The error bars correspond to the 95% confidence interval.

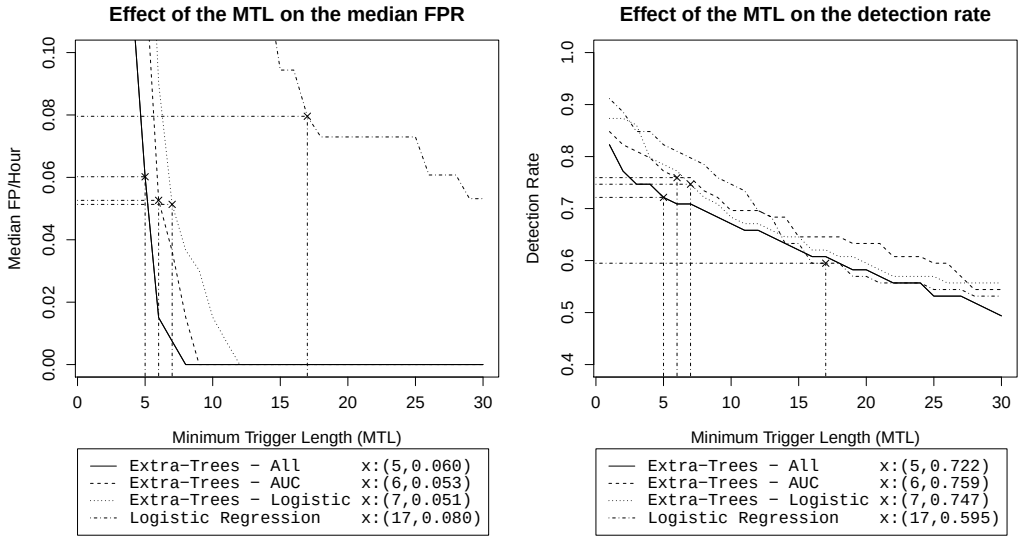


Figure 4–5: Freiburg: Effect of the Minimum Trigger Length. The curves in the two graphs correspond to the evolution of the median false positive rate and the detection rate with respect to the increase of the MTL. The marks show the smallest MTL such that a false positive rate of less than 2 per day is achieved.

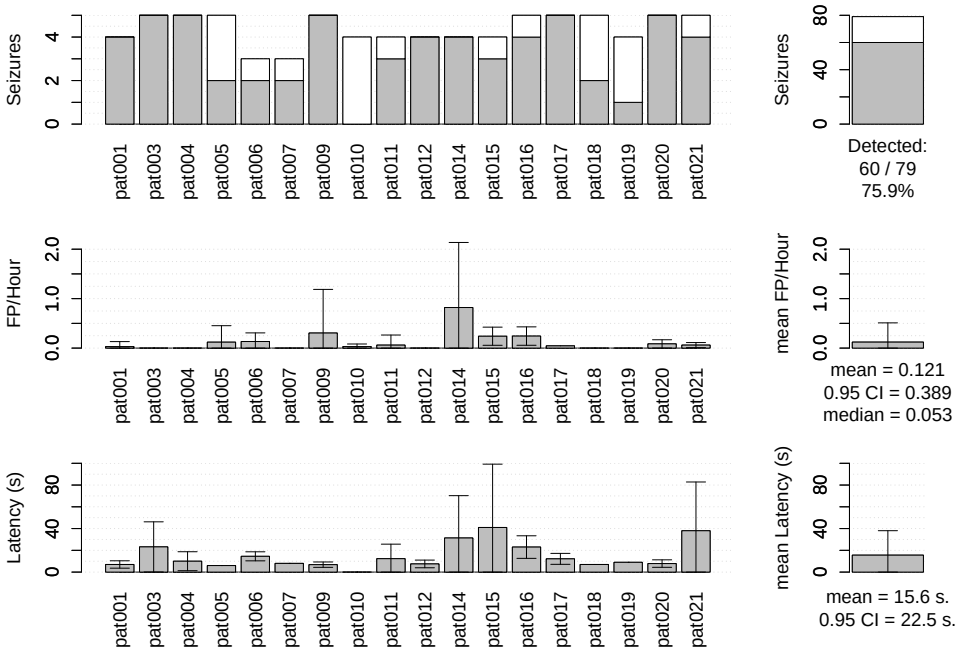


Figure 4–6: Freiburg: Performance of Extra-Trees trained on features selected by their AUC, using a MTL of 6. A per patient analysis is presented on the left and the overall statistics across all patients is presented to the right. The error bars correspond to the 95% confidence interval.

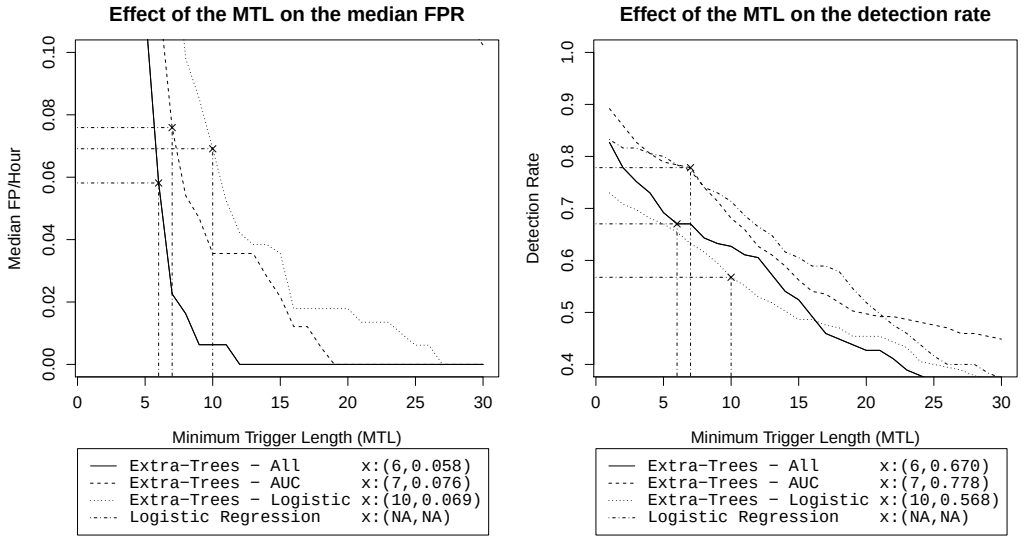


Figure 4–7: CHB-MIT: Effect of the Minimum Trigger Length. The curves in the two graphs correspond to the evolution of the median false positive rate and the detection rate with respect to the increase of the MTL. The marks show the smallest MTL such that a false positive rate of less than 2 per day is achieved.

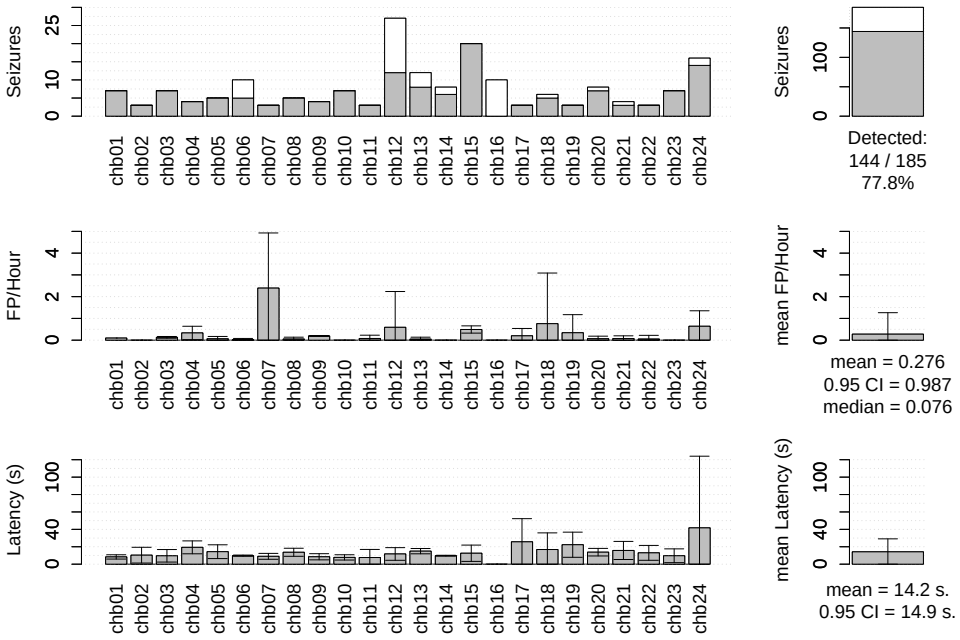


Figure 4–8: CHB-MIT: Performance of Extra-Trees trained on features selected by their AUC, using a MTL of 7. A per patient analysis is presented on the left and the overall statistics across all patients is presented to the right. The error bars correspond to the 95% confidence interval.

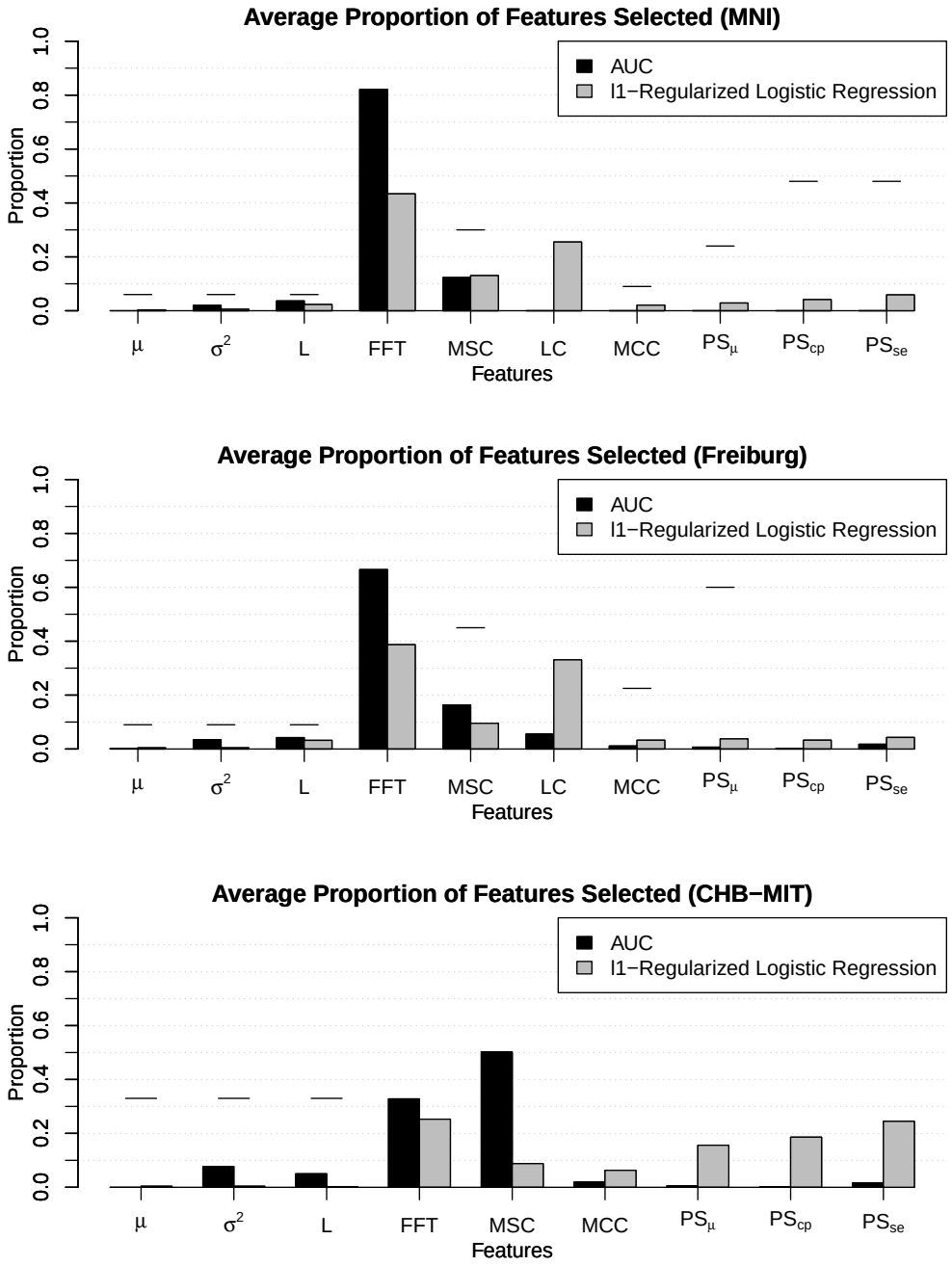


Figure 4–9: Average proportions of features selected by the AUC and $\ell 1$-regularized logistic regression for the MNI, Freiburg and CHB-MIT dataset. The horizontal bars represent the maximal proportion achievable assuming 200 features are selected.

CHAPTER 5
Analytical Comparison of Related Work

We show that the toolbox provides state of the art results in seizure detection by comparing it to other published methods. Table 5–1 contains relevant information about the statistics of the datasets and the performance of these algorithms. Note that the comparison of different methods is difficult as only the work by Shoeb & Guttag was performed on a freely available dataset (CHB-MIT). We will first consider work developed using ECoG recordings and finish with algorithms developed for scalp EEG recordings.

Chan et al. [3] used spectral and temporal features combined with a support vector machine (SVM) in order to localize seizure onset times. They obtained a 89.4% sensitivity with an average of 0.69 false positives per hour using ECoG recordings. The sensitivity obtained for the MNI dataset (86.1%) is similar, but our average false positive rate per hour (0.106) is 6.5 times lower. Reducing the MTL to 5, we obtain a comparable average false positive rate per hour of 0.651 with a sensitivity of 95.6%. For the Freiburg dataset, our sensitivity is lower at 75.9%, but our false positive rate per hour is 5.7 times lower at 0.121. We can lower the MTL at 3 to obtain a 81.0% sensitivity with an average false positive rate per hour of 0.425. However, these results are still worse than the one obtained by Chan according to the sensitivity. The sensitivity of our method could be improved if more seizures were

available per patient. Indeed, to train their algorithm, 10 seizures were used per patient, which is much more that the 2 to 4 that were available in our case (recall that a third of the seizures are always withheld for performance evaluation in the cross-validation).

Gardner et al. [9] used energy-based features combined with a one-class SVM for abnormal activity detection in ECoG recordings. They obtained a 97.1% sensitivity with an average false positive rate per hour of 1.56. They have a higher sensitivity than both the MNI (86.1%) and Freiburg (75.9%) datasets, but their average false positive rate is 14.7 and 12.9 times higher in comparison, respectively. We can reduce the MTL for the MNI dataset to 5 and obtain a similar sensitivity of 95.6%, with an average false positive rate of 0.651, which is still 2.4 times lower. Even when reducing the MTL for the Freiburg dataset, we are unable to obtain a similar sensitivity. However, it is important to note that the 5 patients used in their study all suffered from temporal lobe epilepsy and that the ECoG recordings were hand selected by experts to insure the absence of artifacts. On the other hand, their algorithm was trained using only interictal data, making it harder to differentiate between ictal events and abnormal activity that could result from other processes. The advantage of this methodology is that no seizures need to be present in the training data; at the cost of a higher false positive rate.

Comparing the techniques developed for scalp EEG recordings, we see that the performance of our algorithm on the CHB-MIT dataset is on par with Saab & Gotman, obtaining a sensitivity of 77.8% and an average false

positive rate per hour of 0.276 compared to their 76% sensitivity and 0.34 average false positive rate [41]. Their algorithm used wavelet decomposition combined with a Bayesian model for seizure detection. Our sensitivity is also similar to the one obtained by IdentEvent (79.5), however their average false positive rate per hour (0.09) is 3.1 times lower [21]. The IdentEvent algorithm is not patient specific, but it was trained using 141 seizures, taken from 47 patients, with a total of 3653 hours of data. This is a lot more data than what we used for each patient present in the CHB-MIT dataset.

Zandi et al. [52] implemented a new feature called the combined seizure index and monitored its increase using a robust statistic in order to raise alarms. They obtained a 90.5% sensitivity with an average false positive rate of 0.51 per hour using scalp EEG recordings [52]. Their sensitivity is 12.7% higher than ours (77.8%), but our average false positive rate per hour is 1.85 times lower at 0.276. Their algorithm did not use any machine learning techniques for seizure detection. However, they optimized the algorithm configuration using their testing data. As an example, they chose the Daubechies 6 wavelets as they had the best performance after comparing Daubechies, Coiflets and Simlets wavelets at different orders. Moreover, they show that the algorithm parametrization was close to optimal for their dataset. To get a better idea of the true sensitivity and false detection rate of their algorithm, experiments should be ran on a held out dataset.

Shoeb & Guttag [47] designed a new type of feature vector that accounted for both the spectral and spatial information of a patient's disorder characteristics and used a SVM with a non-linear kernel to detect ictal events. Their work provides much better results than ours, with a sensitivity of 96% and a median false positive rate of 0.08 per hour on the CHB-MIT dataset [47]. The large difference in sensitivity could be partly explained by the way their algorithms were trained and performance measured. They performed a full leave-one-out cross-validation for each patient, training on all the data except for a single one hour long segment in each fold. They assessed the performance of the algorithm on the left out segment, measuring the sensitivity and latency when the segment contained ictal events and the false positive rate when the segment was seizure free. Doing so, the algorithm was able to train on almost all the data (on average 96.8%) available per patient at each fold, therefore lowering the chances of committing errors. As a comparison, our algorithm was trained using on average 9.08% of the available data per patient at each fold.

Table 5–1: Related Work. Medians are indicated by (m) and averages by (μ). All authors used private datasets with the exception to Shoeb, who used the MIT dataset.

Method	**Type**	**Patients**	**Seizures**	**Duration (h)**	**Sensitivity**	**False Detection Rate (h^{-1})**	**Latency (s.)**
Saab & Gotman [41]	Scalp	16	69	360	76	0.34 (μ)	10 (m)
IdentEvent [21]	Scalp	102	287	4853	79.5	0.09 (μ)	-
Shoeb et al. [46]	Scalp	36	139	60	94.2	0.25 (μ)	8 (μ)
Shoeb & Guttag [47] (CHB-MIT)	Scalp	24	173	916	96	0.08 (m)	4.6 (μ)
Zandi et al. [52]	Scalp	14	63	75.8	90.5	0.51 (μ)	7 (m)
Chan et al. [3]	iEEG	6	1792	166.6	89.4	0.69 (μ)	(Offline)
Gardner et al. [9]	iEEG	5	29	≥ 200	97.1	1.56 (μ)	-7.58 (μ)
Extra-Trees using AUC selection - MNI	iEEG	6	137	475	86.1	0.071 (m), 0.106 (μ)	24.9 (μ)
Extra-Trees using AUC selection - Freiburg	iEEG	18	79	450	75.9	0.053 (m), 0.121 (μ)	15.6 (μ)
Extra-Trees using AUC selection - CHB-MIT	Scalp	24	185	824	77.8	0.076 (m), 0.276 (μ)	14.2 (μ)

CHAPTER 6
Discussion of Results

We now discuss the results presented in Chapter 4 with respect to the different design decisions made while creating the toolbox. Some ideas of future work are also presented at the end of the chapter.

6.1 Feature Extraction

From the overall good performance of the toolbox on the majority of the patients present in the three datasets, we can confirm the effectiveness and extended coverage of the features implemented with respect to the different epileptic syndromes and seizures. Indeed, for each patients, the large pool of feature configurations enabled a subset of features to be useful in identifying ictal events while disregarding the presence of artifacts, interictal epileptiform discharges and other brain processes contained in the EEG recordings.

6.2 Feature Selection

The main concern when using such a large set of different features is the computational complexity, which translates into a prohibitively long feature extraction phase. We remedied this problem by adding a feature selection module. We considered a filtering method based on the AUC of individual features and an embedded method using a $\ell 1$-regularized logistic regression.

The filtering method proved to be more useful, as the performance of the Extra-Trees using the selected subsets was superior. This can be explained

by the fact that the AUC is robust against class imbalances. Effectively, only 5% of the training data consisted of feature vectors computed during ictal events. The AUC feature selection picked a total of 200 features per fold from the available set, using only 2.06%, 1.52% and 1.65% of the total number of features present in the MNI, Freiburg and CHB-MIT datasets, respectively. This small proportion of features can be extracted in real time from EEG recordings using a standard personal computer. Another interesting consequence of the use of this method is its small computational footprint when selecting features during the training phase of the toolbox.

Even though the reduction in features used in the Extra-Trees was substantial, their performance was not affected for the MNI dataset. The variations in sensitivity could be explained by the random property of the trees. For the Freiburg dataset, selecting a subset of features increased the performance of the algorithm by a small margin, discarding irrelevant features. As for the CHB-MIT dataset, the features selected using the AUC improved the sensitivity of the trees by 11.8% compared to using all of them. This major improvement can be explained by two factors: the selected features are highly sensitive to signal differences present in ictal events, and by choosing a smaller subset of features, we force the trees to explore the separation boundaries between ictal events and interictal data at a finer level. On the other hand, the features selected by the $\ell 1$-regularized logistic regression reduce the performance of the trees by 10.2% compared to when all of them

are used. If we look at the performance of the $\ell 1$-regularized logistic regression as a classifier, we observe that it is unable to obtain a false positive rate per hour under 0.08 with MTLs in $\{1, 2 \ldots, 30\}$. The blurred electrical activity recorded from large areas of neurons on the brain surface combined with the presence of artifacts in scalp EEG recordings may cause the decision boundary between ictal events and interictal data to be highly non-linear, making it impossible for the regression to correctly model the data. Indeed, the maximum sensitivity of the regression over the training data, obtained with a MTL of 1, is around 80%, whereas for the MNI and Freiburg datasets, it is close to 98%. This indicates that the regression is unable to capture the complex differences between the two classes of data. Moreover, most of the features selected by the regression are irrelevant, as reflected in their AUC scores. The median AUC across all folds is 0.60, implying that half have a behaviour close to random. The latter could explain why the Extra-Trees are unable to correctly separate the two classes using the features selected by the $\ell 1$-regularized logistic regression: not enough relevant information about the epileptic syndrome is present in the selected features.

6.3 Classification

Although the feature selection module has some impact on the performance of the toolbox, the choice of classifier is even more important. As expected, the performance of the Extra-Trees, a high level machine learning classifier, is much better than the simpler $\ell 1$-regularized logistic regression. Indeed, the Extra-Trees obtained good detection rates with much lower false

positives rates on most of the patients compared to the $\ell 1$-regularized logistic regression. It is no surprise, as the trees are able to represent much more complex classification boundaries than the linear model. It is important to note that even if the trees are more complex, the time used for their creation and during classification is still small. Moreover, we would like to highlight the low data requirements to train the classifier: a small number of hand labelled seizures (e.g. 3) and a few segments of interictal data.

The higher performance of the Extra-Trees compared to the logistic regression is clearly reflected across both the Freiburg and CHB-MIT datasets. In both cases, the high amount of false positives generated by the logistic regression force the need of higher MTL values, lowering the sensitivity of the algorithm under 60%. The better performance of the $\ell 1$-regularized logistic regression on the MNI dataset could be explained by the larger amount of seizures available during training (median of 12.5). As a comparison, the median number of seizures present in the training folds of the Freiburg dataset is 3, and it is 4 for the CHB-MIT dataset. Also, we suspect the epileptic syndrome generated in the rats to be simpler than the ones existing in human patients. When using human data, the more complex brain processes and the lower amount of available ictal events during the training phase could be responsible for the lower performance of the Extra-Trees compared to the one obtained for rats.

Finally, we would like to highlight the necessity of using more complex classification models, such as Extra-Trees, in order to obtain good detection

and low false positive rates. Indeed, only the Extra-Trees were able to provide good results on the majority of the patients. The $\ell 1$-regularized logistic regression had high positive rates, which induced low detection rates. It is important to note that a single feature combined with a threshold is equivalent to training a logistic regression using only this feature. Therefore, since the $\ell 1$-regularized logistic regression, using more than one feature, fell short on performance, the simple feature/threshold model would be even worse.

6.4 Minimum Trigger Length

As with most classifiers, we cannot specify a trade-off between sensitivity and false positive rate directly in the training of the Extra-Trees. The introduction of the MTL parameter enables us to control this trade-off to some extent, by choosing an acceptable false positive rate for the task at hand. Tasks demanding lower false positive rates can use a higher MTL, whereas tasks that can tolerate a higher false positive rate can lower the MTL, improving the sensitivity.

6.5 Default Toolbox Parameters

An interesting characteristic of the toolbox is its default set of parameters. Indeed, the default parametrization of the feature extraction, feature selection and classifier training was used on all the patients present in the three datasets, with good results. This essentially makes the algorithm "parameter-free" if desired, while enabling the toolbox to be used under different environments, such as seizure prediction, by tweaking the parameters if need be.

The principal set of parameters that would require modifications reside in the feature extraction phase. They could be tailored to capture relevant information occurring in the pre-ictal phases of the EEG recordings. The Extra-Trees could be trained to detect the pre-ictal phases, therefore predicting the occurrence of seizures. In this study, about half of the AUC scores of the computed features are under 0.60. We could speed up the initial feature extraction phase by reducing the range of features extracted from the data such that most of them have a score above this threshold. In order to do so, we would need to verify if a subset of features performs constantly poorly across all the patients.

The parameters of the Extra-Trees could also be modified. Recall that increasing the value of M, the number of trees present in the ensemble, will only increase the performance of the classifier as it will reduce the variance error. The value of $n_{\min} = 3$ provided trees that were deep enough to capture the complex differences between ictal events and interictal data. The default parameter of $K = \sqrt{|F|}$ provided enough randomization in the trees, achieving good results when used in an ensemble.

6.6 Modularity

The separation of the toolbox into distinct modules that can be used separately or jointly facilitates the development of new algorithms for seizure detection or prediction. As an example, the output of the feature extraction/selection modules can readily be used by other machine learning toolboxes such as Weka [15].

6.7 Future Work

It would be interesting to see if any correlations exists between the different types of epileptic disorders and seizures, and the features used to detect ictal events in the EEG recordings. If such relations exist, they could help elucidate the potential mechanisms responsible for the seizures. Moreover, these correlations could be detected in a patient automatically in order to refine the parameters of the features extracted from the EEG recording, improving the quality of the information retrieved.

Also, a long-term online in vivo study would help validate the performance of the toolbox in real world settings. Indeed, such a study would capture other properties of the disorders like their evolution over time, the different manifestations of seizures, etc.

CHAPTER 7
Conclusion

We presented a toolbox that automates the creation of efficient personalized seizure detection algorithms that operate in real-time. Its default set of parameters provide a good performance for the majority of the patients, eliminating the necessity to adjust them on a per patient basis. The data requirements of the toolbox are modest, requiring a small number of seizures (e.g. 3) and a few EEG segments containing interictal data.

The toolbox is capable of extracting relevant information about a patient's epileptic syndrome and seizures from the complex EEG recordings by calculating a large number of features and then selecting a small subset representative of the patient's condition. It was shown to be robust against the class imbalances present between ictal events and interictal data. Also, the computational complexity of each module is low, enabling the toolbox to process large amounts of data.

To our knowledge, this is the first toolbox/algorithm to be analyzed using a sound methodology, such as cross-validation, on multiple freely available datasets. We want to highlight the importance of testing new techniques on publicly available datasets in order to ease the comparison of methods for seizure detection and prediction. For this reason, the MNI

dataset, containing ECoG recordings of rats, is made publicly available at `www.cs.mcgill.ca/~gsauln/files/mni_dataset/`.

The existence of such a toolbox could alleviate the task of annotating EEG recordings during the diagnostic of new patients. Moreover, it could help the analysis of large EEG databases. Nursing care facilities could also benefit from such a system, alerting relevant authorities when a patient suffers from a seizure. In the long run, good detection algorithms can help design early seizure detection as well as seizure prediction algorithms. From these, new treatments could be developed in order to prevent the occurrence of seizures or cause them to abort early.

On a side note, it is important to mention that the toolbox can be used on a wide variety of time-series signals, such as accelerometer data, as was performed by Moghaddam et al. [31]. As of now, the toolbox contains few classification algorithms: linear regressions and Extra-Trees. We would like to add classifiers such as support vector machines.

Finally, we release the toolbox for public use under the Apache License, Version 2.0, at `www.cs.mcgill.ca/~gsauln/` and we encourage others to do the same in order to improve the knowledge on the disorder as well as reduce the overhead of creating derived algorithms.

References

[1] J. Arnhold, P. Grassberger, K. Lehnertz, and C.E. Elger. A robust method for detecting interdependences: application to intracranially recorded EEG. *Physica D: Nonlinear Phenomena*, 134(4):419–430, 1999.

[2] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[3] A. M. Chan, F. T. Sun, E. H. Boto, and B. M. Wingeier. Automated seizure onset detection for accurate onset time determination in intracranial EEG. *Clinical Neurophysiology*, 119(12):2687–2696, 2008.

[4] G. Curia, D. Longo, G. Biagini, R. S.G. Jones, and M. Avoli. The pilocarpine model of temporal lobe epilepsy. *Journal of Neuroscience Methods*, 172(2):143–157, 2008.

[5] J. Engel. Report of the ILAE classification core group. *Epilepsia*, 47(9):1558–1568, 2006.

[6] T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27:861–874, June 2006.

[7] R. S. Fisher, W. van E. Boas, W. Blume, C. Elger, P. Genton, P. Lee, and J. Engel. Epileptic seizures and epilepsy: Definitions proposed by the international league against epilepsy (ILAE) and the international bureau for epilepsy (IBE). *Epilepsia*, 46(4):470–472, 2005.

[8] Freiburg University. The Freiburg EEG database. http://epilepsy.uni-freiburg.de/freiburg-seizure-predictionproject/eeg-database, 2012.

[9] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, and B. Litt. One-class novelty detection for seizure analysis from intracranial eeg. *J. Mach. Learn. Res.*, 7:1025–1044, December 2006.

[10] P. Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification.* PhD thesis, University of Liège, Belgium, May 2002.

[11] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[12] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[13] J. Gotman. Automatic recognition of epileptic seizures in the eeg. *Electroencephalography and Clinical Neurophysiology*, 54(5):530 – 540, 1982.

[14] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction, Foundations and Applications*. Series Studies in Fuzziness and Soft Computing. Springer, 2006.

[15] M. Hall, M. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. In *SIGKDD Explorations*, volume 11, 2009.

[16] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.

[17] F.J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, January 1978.

[18] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[19] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.*, 21(9):1263–1284, September 2009.

[20] A. Jacoby and J. K. Austin. Social stigma for adults and children with epilepsy. *Epilepsia*, 48:6–9, 2007.

[21] K.M. Kelly, D.S. Shiau, R.T. Kern, J.H. Chien, M.C.K. Yang, K.A. Yandora, J.P. Valeriano, J.J. Halford, and J.C. Sackellares. Assessment of a scalp EEG-based automated seizure detection system. *Clinical Neurophysiology*, 121(11):1832–1843, 2010.

[22] J. Klatt, H. Feldwisch-Drentrup, M. Ihle, V. Navarro, M. Neufang, C. Teixeira, C. Adam, M. Valderrama, C. Alvarado-Rojas, A. Witon, M. Le Van Quyen, F. Sales, A. Dourado, J. Timmer, A. Schulze-Bonhage, and B. Schelter. The epilepsiae database: An extensive electroencephalography database of epilepsy patients. *Epilepsia*, 53(9):1669–1676, 2012.

[23] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[24] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering*, pages 3–24. IOS Press, 2007.

[25] L. Kristensen and P. Kirkegaard. *Sampling problems with spectral coherence*. Roskilde, Denmark : RisOo National Laboratory., 1986.

[26] J. Langford, L. Li, and A. Strehl. Vowpal Wabbit, 2012. `http://hunch.net/~vw`.

[27] S.I. Lee, H. Lee, P. Abbeel, and A.Y. Ng. Efficient l1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, pages 401–401. AAAI Press; MIT Press; 1999, 2006.

[28] M. Levesque, A. Bortel, J. Gotman, and M. Avoli. High-frequency (80-500 Hz) oscillations and epileptogenesis in temporal lobe epilepsy. *Neurobiology of Disease*, 42(3):231–241, 2011.

[29] M. Levesque, P. Salami, J. Gotman, and M. Avoli. Two seizure-onset types reveal specific patterns of high-frequency oscillations in a model of temporal lobe epilepsy. *Journal of Neuroscience*, 32(38):13264–13272, 2012.

[30] W. Loscher. Animal models of intractable epilepsy. *Progress in Neurobiology*, 53(2):239–258, 1997.

[31] A.K. Moghaddam, J. Pineau, J. Frank, P. Archambault, F. Routhier, T. Audet, J. Polgar, F. Michaud, and P. Boissy. Mobility profile and wheelchair driving skills of powered wheelchair users: Sensor-based event

recognition using a support vector machine classifier. In *Engineering in Medicine and Biology Society (EMBC), 2011 Annual International Conference of the IEEE*, pages 7336–7339, September 2011.

[32] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz. Seizure prediction: the long and winding road. *Brain*, 130(2):314–333, 2007.

[33] A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 78–. ACM, 2004.

[34] D. E. Olsen, R. P. Lesser, J. C. Harris, W. R. S. Webber, and J. A. Cristion. Automatic detection of seizures using electroencephalographic signals. Patent, 05 1994. US 5311876.

[35] I. Osorio, M. G. Frei, and S. B. Wilkinson. Real-time automated detection and quantitative analysis of seizures and short-term prediction of clinical onset. *Epilepsia*, 39(6):615–627, 1998.

[36] P. Perucca and F. G. Gilliam. Adverse effects of antiepileptic drugs. *The Lancet Neurology*, 11(9):792 – 802, 2012.

[37] J. Pillai and M. R. Sperling. Interictal eeg and the diagnosis of epilepsy. *Epilepsia*, 47:14–22, 2006.

[38] R. Quian Quiroga, A. Kraskov, T. Kreuz, and P. Grassberger. Performance of different synchronization measures in real data: A case study on electroencephalographic signals. *Phys. Rev. E*, 65:041903–041903, Mar 2002.

[39] R. J. Racine. Modification of seizure activity by electrical stimulation: Ii. motor seizure. *Electroencephalography and Clinical Neurophysiology*, 32(3):281–294, 1972.

[40] M. Rosenblum, A. Pikovsky, J. Kurths, C. Schäfer, and P.A. Tass. Chapter 9 phase synchronization: From theory to data analysis. In F. Moss and S. Gielen, editors, *Neuro-Informatics and Neural Modelling*, volume 4 of *Handbook of Biological Physics*, pages 279–321. North-Holland, 2001.

[41] M.E. Saab and J. Gotman. A system to detect the onset of epileptic seizures in scalp EEG. *Clinical Neurophysiology*, 116(2):427–442, 2005.

[42] H. E. Scharfman. The neurobiology of epilepsy. *Curr Neurol Neurosci Rep*, 7(4):348–354, July 2007.

[43] C. E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, January 1949.

[44] E. M. S. Sherman, S. Wiebe, T. B. Fay-McClymont, J. Tellez-Zenteno, A. Metcalfe, L. Hernandez-Ronquillo, W. J. Hader, and N. JettÃľ. Neuropsychological outcomes after epilepsy surgery: Systematic review and pooled estimates. *Epilepsia*, 52(5):857–869, 2011.

[45] A. Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment*. PhD thesis, Harvard University–MIT Division of Health Sciences and Technology., Massachusetts Institute of Technology, 2009.

[46] A. Shoeb, H. Edwards, J. Connolly, B. Bourgeois, S. T. Treves, and J. Guttag. Patient-specific seizure onset detection. *Epilepsy & Behavior*, 5(4):483–498, 2004.

[47] A. Shoeb and J. Guttag. Application of Machine Learning To Epileptic Seizure Detection. In *International Conference on Machine Learning (ICML)*, 2010.

[48] R. Srinivasan and P.L. Nunez. Electroencephalography. In Editor in Chief: V.S. Ramachandran, editor, *Encyclopedia of Human Behavior (Second Edition)*, pages 15 – 23. Academic Press, 2012.

[49] W. H. Theodore, S. S. Spencer, S. Wiebe, J. T. Langfitt, A. Ali, P. O. Shafer, A. T. Berg, and B. G. Vickrey. Epilepsy in north america: A report prepared under the auspices of the global campaign against epilepsy, the international bureau for epilepsy, the international league against epilepsy, and the world health organization. *Epilepsia*, 47(10):1700–1722, 2006.

[50] S. Wiebe, W. T. Blume, J. P. Girvin, and M. Eliasziw. A randomized, controlled trial of surgery for temporal-lobe epilepsy. *New England Journal of Medicine*, 345(5):311–318, 2001.

[51] World Health Organization. WHO - Epilepsy, 2012. `http://www.who.int/mediacentre/factsheets/fs999/en/`.

[52] A.S. Zandi, M. Javidan, G.A. Dumont, and R. Tafreshi. Automated real-time epileptic seizure detection in scalp EEG recordings using an algorithm based on wavelet packet transform. *Biomedical Engineering, IEEE Transactions on*, 57(7):1639–1651, 2010.