ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

BIWI Computer Vision

Prediction of epileptic seizures using EEG data

Semester Thesis

Maurice Gonzenbach
Department of Mathematics

Advisors: Dr. Martin Jaggi, Data Analytics Lab
Dr. Valeria De Luca, Computer Vision Laboratory
Supervisor: Prof. Dr. Gábor Székely, Computer Vision Laboratory

August 18, 2015

Abstract

Seizure forecasting systems have the potential to improve the quality of life for patients with epilepsy. In this work a framework is proposed to predict epileptic seizures, as part of an open challenge sponsored by the American Epilepsy Society (kaggle.com). 16 to 24 channel intracranial electroencephalogram (EEG) streams of dogs and patients in normal state and prior to an epileptic seizure are used as raw data. Therefrom, around 1000 features are extracted, based mainly on discrete wavelet transform. Linear binary classifiers are trained on these features. The quality of the prediction is evaluated using cross validation, which delivers an area under the curve of response operator characteristics (ROC AUC) ranging in [0.77, 0.99] when counting the best classifier result for each subject. The findings in this report show that simplicity often beats complexity, thus favoring linear classifiers, and highlight the importance of robust regularization techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Setting

Currently over 1 % of the world population are affected by spontaneous seizures that can be traced back to epilepsy [14] [10]. Epilepsy is caused by genetic predisposition or events, such as strokes, tumors or drug misuse. Although there is medication available for preventing seizures, only approximately 70 % of the patients respond to these drugs. Other treatments, e.g. surgery, neurostimulation and change of diet are not always / often successful. Besides the health issues due to epilepsy, many patients suffer from persistent anxiety of experiencing further seizures. They also have to cope with many restrictions in their daily lives, such as the inability to drive a car or other social and financial disadvantages.

Reliable predictions of upcoming seizures would greatly benefit many patients. First applications would probably be situated in clinical environments, where appropriate measurement equipment and trained personnel is readily available. Knowledge of the probability for a seizure within a certain time frame could help doctors and nurses to fit the treatment methods better to the patient and allow them to plan staffing requirements (e.g. due to expected alarms) more efficiently.

Further ahead, the technology might also enable patients to anticipate seizures themselves using portable devices, which would allow them to prepare if a seizure is imminent or perform potentially dangerous activities when no seizure is expected.

1.2 Goal of this Thesis

The goal of this thesis is to predict seizures based on intracranial EEG data. The brain activity of epileptic patients can be classified into four states, namely:

- **Interictal:** between seizures, baseline
- **Preictal:** prior to seizure (usually up to around 1 h before seizure)
- **Ictal:** during seizure
- **Post-ictal:** after seizure

Using a machine learning approach, the presented algorithm tries to discriminate between **preictal** and **interictal** classes of EEG segments, which can be translated into a binary classification problem. The data used in this work was provided by the American Epilepsy Society through the machine learning competition website kaggle.com [10]. The challenge provided EEG data from seven patients, totaling at around 680 h, which corresponds to almost 120 GB of binary data. Besides aiming for a high prediction correctness – measured as *area under the curve* (AUC) of *response operator characteristics* (ROC) – performance therefore was another issue that had to be kept in mind. Future applications should obviously run in (near) real-time, thus making it a necessity that at least the classification routine is sufficiently fast.

Finally, the results of the developed algorithm are benchmarked on the *American Epilepsy Society Seizure Prediction Challenge* [10] competition, which allows an estimation about the performance compared to others.

1.3 Proposed Approach

The approach suggested in this thesis combines well known signal analysis and machine learning approaches. First of all, a Principal Component Analysis (PCA) is separately performed on all interictal and all preictal segments, obtaining two new coordinate systems. Thereby the different channels of the recordings (each of them corresponding to one sensor located on the brain) are treated as the dimensions and the samples in time as observations. However, the data is *not yet* transformed into these spaces, but rather kept in its original representation.

A sliding window with overlap is used to further divide the 10 min segments into pieces of roughly 30 s. Two new channels are then added to the original ones, by projecting the data onto the previously computed PCA spaces; therefrom two scalar measures are calculated, which describe how well every sample "fits" into each of the spaces, leading to the additional channels mentioned above.

Each channel subsequently is divided into 7 frequency bands (lying between 0 Hz and 200 Hz) using Discrete Wavelet Transformation (DWT).

Finally, some basic statistical features are computed on all frequency bands, e.g. mean, standard deviation, signal energy and mean of peak prominence. In summary we now have 17 to 25 channels

(corresponding to sensors on the brain), every one of them being divided into 7 frequency bands on which in turn 8 basic features are computed. This leads to around 1000 features per window.

Various classifiers have been tested on the resulting features, with a focus on linear discriminant analysis (LDA), linear regression (LR) and support vector machines (SVM). A pitfall hereby lay in developing a suitable cross validation scheme, in particular finding a meaningful split between the sets. Special attention was also paid to the regularization of the classifiers, as for some subjects only very few sets were available for training.

Chapter 2

Related Work

In the past decade, several publications focused on the automatic characterization and classification of EEG signals from epileptic patients. Approaches and aims varied significantly. Pioneers in this research topic were Adeli et. al (2003) [2]. They proposed the use of wavelet transform for analyzing epileptic discharges on scalp EEG data. While their approach was very successful in helping characterize different types of discharges, they did not apply an automatic classification algorithm. Follow-up works included the automatic detection of epileptic seizures using a Levenberg-Marquardt backpropagation neural network [7] and a Multi-Spiking Neural Network [6], both applying basic statistical features computed from wavelet transforms. On their respective test sets both methods claim an accuracy of over 90 %. In a further publication the authors also tackle the classification of other diseases, e.g. ADHS (*Attention deficit hyperactivity disorder*) [3]. Subasi et al. (2010) [13] suggested the use of PCA, independent component analysis (ICA) and LDA after applying a DWT, to reduce dimensionality of the data and obtain features which can be fed to an SVM classifier. On their test set LDA outperformed the other methods, with an accuracy of 100 %.

While most of these authors explicitly address the problem of *detecting* epileptic seizures, their approaches are generic enough to easily allow adaption to another two-class problem – the *prediction* of epileptic seizures, yet probably resulting in a lower classification accuracy. An interesting method (although similarly presented in earlier publications already) is suggested by Guo et al. (2011) [8]: after computing an initial feature set, they apply a genetic programming algorithm to select a subset of the original feature space, transform the selected features and combine them using simple algebraic and transcendental functions (e.g. additions, multiplications, logarithms). A good overview of past and current research is given in the review of automated EEG analysis of epilepsy by Acharya et al. (2013) [1], suggesting that nonlinear features promise the best results, while no decisive statement can be made on the exact type of features. Classifiers delivering good results include SVMs, Gaussian Mixture Models, Fuzzy classification and decision trees.

Other solutions to the problem stated in this report, using also the same data as here, can be found on the forum of the seizure prediction challenge on kaggle.com. The winner of the challenge followed an approach not too different to the one presented in this work [11]: as features he used the per channel and average energy in different frequency bands, correlation of energy in different frequency

bands and a some more basic statistical measures. These were then non-linearly transformed using logarithmic, square root and power functions to obtain around 900 to 1500 features per segment. As classifier he applied a highly regulated, iterative LS Ensemble, which was tuned using one-out cross validation. A lot of emphasis was put on the regularization of the classifier, as he identified over-fitting to be the biggest challenge in the process .

Chapter 3

Materials and Methods

3.1 Data

The data used in this thesis was provided by the American Epilepsy Society through the machine-learning website kaggle.com [10]. They contain EEG recordings from five dogs and two human patients, parts of which are labelled for training purposes. For every subject, a few dozen to over a thousand 10 min segments were given, each being either preictal or interictal. The segments again consisted of 15 to 24 channels, corresponding to the number of sensors applied to the subject's brain. Details on the raw data are summarized in table 3.1.

Table 3.1: Characteristics of the raw data provided by kaggle [10]

	Dog 1	*Dog 2*	*Dog 3*	*Dog 4*	*Dog 5*	*Patient 1*	*Patient 2*	*Total*
Segments total	1'006	1'542	2'419	1'891	671	263	210	8'002
Segments labeled preictal	**24**	**42**	**72**	**97**	**30**	**18**	**18**	**301**
Segments labeled interictal	**480**	**500**	**1'440**	**804**	**450**	**50**	**42**	**3'766**
Ratio interictal / preictal	**20**	**12**	**20**	**8**	**15**	**3**	**2**	**n/a**
Segments unlabeled (test)	502	1'000	907	990	191	195	150	3'935
Independent preictal phases	**4**	**7**	**12**	**17**	**5**	**3**	**3**	**51**
Sampling frequency [Hz]	400	400	400	400	400	5'000	5'000	n/a
Channels	16	16	16	16	15	15	24	n/a
File size [GB]	7.7	11.8	18.6	14.5	4.8	25.2	31.0	113.6

While the sum of all recordings was more than 680 h, corresponding to almost 120 GB of data, the distribution of classes among the subjects varies drastically: all subjects have substantially fewer

preictal segments, with the ratio of interictal to preictal ranging between 2 and 20. This poses an additional challenge to the classifier, which has to take these differences in distribution into consideration. It can further be observed, that the number of *independent* preictal phases is even lower, ranging from only 3 to 17. The reason for the mutual dependence of some segments is given by the nature of the recordings: preictal segments occur from 5 min to 1 h 5 min before a seizure (see figure 3.1). Therefore, six segments of 10 min actually originate from the same 1 h period. It is important to consider this fact when performing cross validation, as checking the classifier against segments from the same period delivers over-optimistic results.

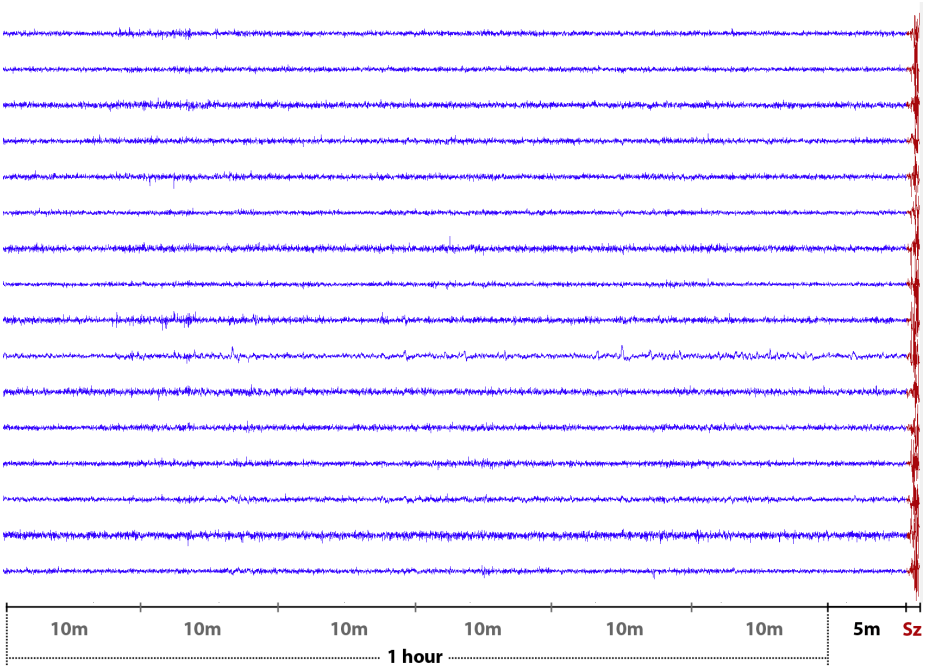

Figure 3.1: Illustration of a preictal EEG signal [10]

3.2 Software Environment and Libraries

All development and computation was done using Matlab®. In particular we used the *Statistics and Machine Learning Toolbox* for classification algorithms, as well as the *Signal Processing Toolbox* for signal analysis and feature generation. Some basic code snippets were taken from the *Getting Started Code* by Elliot Dawson (2014) [4]. The linear models were trained and evaluated using *LIBLINEAR*, developed by Fan et al. (2008) [5].

3.3 Processing Pipeline

The processing pipeline can be divided into four basic phases: *(i)* preprocessing, where some general measures are taken to make the data comparable and extract basic characteristics; *(ii)* feature generation, where higher-level features are extracted from the raw data, drastically reducing data dimensionality and sampling rate; *(iii)* classification, where a classifier is trained on the generated features using labeled data, enabling it to make predictions for unlabeled segments; *(iv)* post-processing, which was tailored to the kaggle competition to make the results for the different subjects more consistent. Figure 3.2 shows the aforementioned phases, which we will describe next.

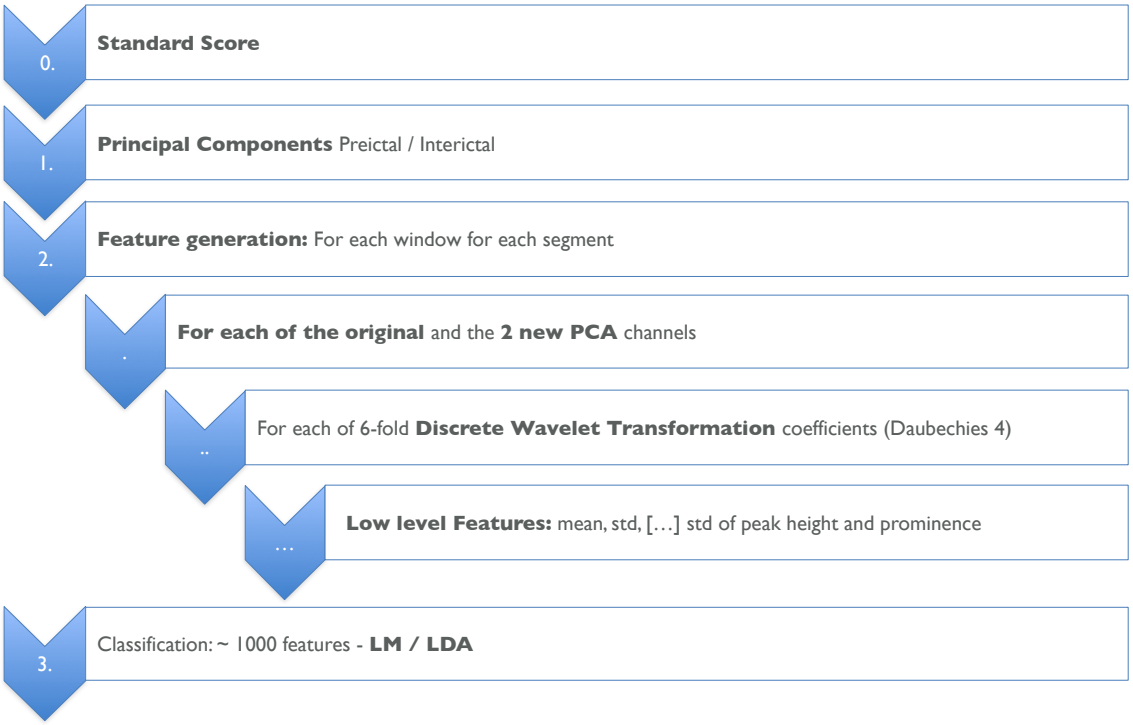

Figure 3.2: Overview of the processing pipeline

3.3.1 Preprocessing

First of all, the data is normalized per subject, using the *preictal* mean and standard deviation. It is important to standardize all three segment "types" (i.e. preictal, interictal and test) by the same measure, to avoid loss of information. From the normalized data we compute two matrices, the principal component space for all interictal segments and the principal component space for all interictal segments:

Given the normalized *preictal* data matrix $\mathbf{X}_p \in \mathbb{R}^{n \times m}$ (concatenation of all preictal segments), where n is the number of observations and m is the dimension (i.e. number of channels), we compute its covariance matrix $\mathbf{C}_p \in \mathbb{R}^{m \times m}$:

$$\mathbf{C}_p = \frac{1}{n-1} \mathbf{X}_p^\top \mathbf{X}_p \tag{3.1}$$

on which a subsequent eigenvalue decomposition is performed, leading to

$$\mathbf{V}_p^{-1} \mathbf{C}_p \mathbf{V}_p = \mathbf{D}_p \tag{3.2}$$

with $\mathbf{D}_p \in \mathbb{R}^{m \times m}$ being a diagonal matrix of all eigenvalues and $\mathbf{V}_p \in \mathbb{R}^{m \times m}$ being a matrix with the corresponding eigenvectors as columns. After sorting the columns of $\mathbf{V}_p$ with respect to the value of their eigenvalue in descending order, these new basis vectors are stored for later use. The same procedure is carried out for the normalized *interictal* data matrix $\mathbf{X}_i$, thus obtaining two different basis representations.

3.3.2 Feature Generation

Sliding Windows

To split up the given 10 min segments, a sliding window approach was used. By applying a grid-search performed on the Brutus Cluster of ETH, appropriate values for the segment length and overlap were determined. All other components of the system were kept constant. We considered a window length of 30 s and an overlap for sequential windows of 5 s or 16.6 %. For example, a 600 s segment resulted in 24 windows.

Introduction of PCA Channels

We introduced two additional channels to the 15 to 24 ones from the EEG data. These are produced using the PCA basis vectors computed beforehand:

When looking at a specific time t, we can interpret the measurements of all channels as a m dimensional vector $\boldsymbol{x_t} \in \mathbb{R}^m$, whereby m is the number of channels. This vector is then projected into both the preictal and the interictal PCA space, leading to:

$$\begin{aligned} \boldsymbol{p_t} &= \mathbf{V}_p \boldsymbol{x_t} \\ \boldsymbol{q_t} &= \mathbf{V}_i \boldsymbol{x_t} \end{aligned} \tag{3.3}$$

For each of these two vectors, a measure indicating how well they "fit" in the PCA space is calculated according to:

$$\boldsymbol{x_t}^{(m+1)} = \frac{\|\boldsymbol{p_t}^{(1:8)}\|_2}{\|\boldsymbol{x_t}^{(1:m)}\|_2} \tag{3.4}$$

$$\boldsymbol{x_t}^{(m+2)} = \frac{\|\boldsymbol{q_t}^{(1:8)}\|_2}{\|\boldsymbol{x_t}^{(1:m)}\|_2} \tag{3.5}$$

, where the notation $\boldsymbol{x}^{(k)}$ denotes the k-th entry of $\boldsymbol{x}$, and $\boldsymbol{x}^{(k:l)}$ denotes entries k to l of $\boldsymbol{x}$. The idea behind this measure is, that if we assume the sample $\boldsymbol{x}$ to be *preictal*, then most of its information should be contained in the first few dimensions of its correspondence $\boldsymbol{p}$ in the *preictal* PCA space. Experiments showed, that over 90 % of the signal energy was contained in the first 8 dimensions, which is why the measure was defined accordingly. Thus the ratio shown in equation 3.4 should be pretty high, ideally close to 1. The ratio of its correspondence $\boldsymbol{q}$ in the *interictal* PCA space as shown in equation 3.5, however, is expected to be rather low. For an interictal sample these properties are obviously just the opposite.

Discrete Wavelet Transformation

Many publications (e.g. Adeli et. al 2003 [2], Guo et al. 2011 [8], Subasi et al. 2010 [13]) suggested that both temporal and frequency domains carry relevant information. Discrete wavelet transformation (DWT) allows to retain temporal resolution, while also getting information about specific frequency bands. Based on the results of published research papers, the method of choice turned to be Daubechies Wavelets of order 4 [9].

Two parameters have to be defined in order to apply the DWT: *(i)* the frequency range which is interesting at all and *(ii)* the "resolution" of the frequency bands (i.e. the range of frequencies summarized per band).

To answer *(i)*, the signal energy in dependence of the frequency for some sample segments was examined (see figure 3.3). The power decreases with higher frequencies, yet a significant difference between the preictal (dashed) and interictal (solid) segments can be observed up to 200 Hz. Therefore all available frequencies from 0 to 200 Hz were used. It is important to note here that the shown signal energies only represent a small sample of all recordings (4 channels of 10 segments). The discrepancy between the preictal and interictal segments clearly visible in these samples can *not* be generalized to all of the segments.

The signals from the human patients, which were sampled at the much higher frequency of 5000 Hz, were subsampled after applying a low-pass filter, to achieve the same sampling frequency for all subjects. For parameter *(ii)* it was decided to use a higher resolution for lower frequencies, as apparently more information is contained in these. This fact was also empirically shown while developing the DWT setup, by evaluating different frequency bands on separately trained classifiers.

In total 6 consecutive DWTs were performed, respectively halving time domain and doubling frequency domain. Two sets of coefficients resulted from every transformation, together describing the lower half of the original frequency domain: The approximation coefficients A_i represented the lowest quarter and the detail coefficients D_i represented the second lowest quarter of the original frequency domain. The frequency bands used for the further computations are listed in table 3.2.

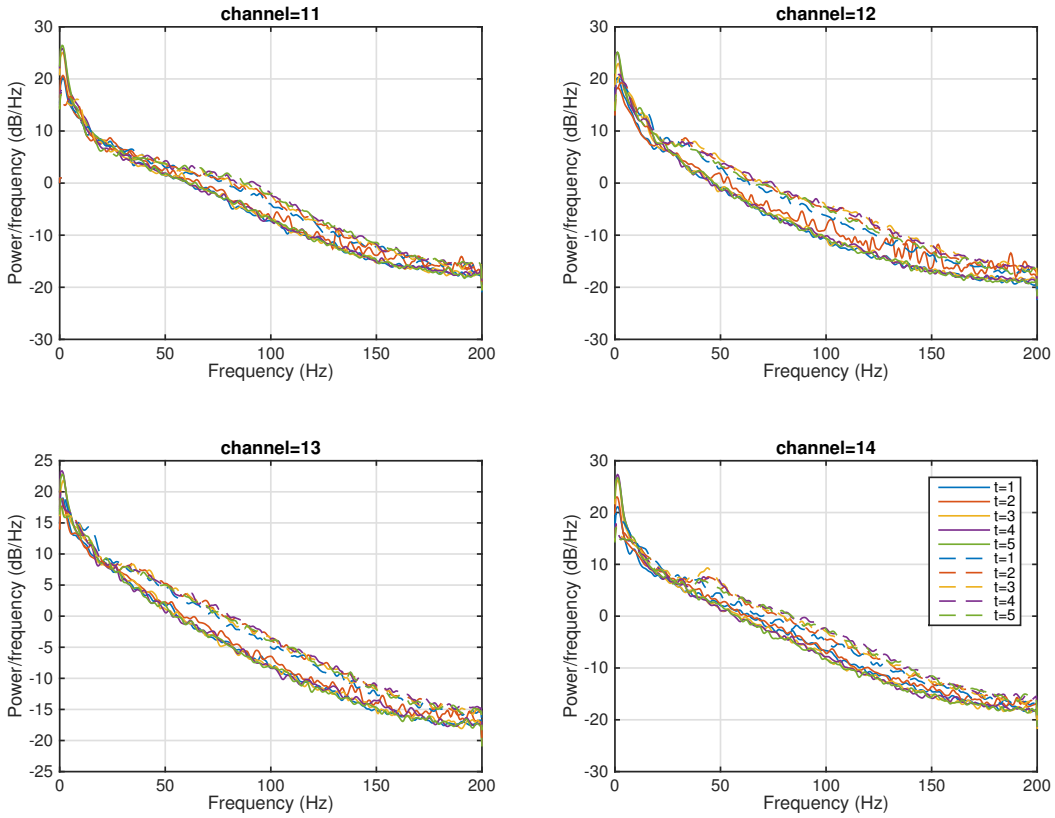

Figure 3.3: Power of signal depending on frequency, shown for 4 channels of 10 sample segments. Dashed lines denote preictal segments, solid lines denote interictal segments. Sampling rate of original data was 400 Hz.

Table 3.2: Frequency bands of DWT decompositions used for feature generation. Note that the decomposition level corresponds to the factor by which the time domain is reduced.

Coefficients	*Frequency band* [Hz]	*Decomposition level*
D1	100-200	1
D2	50-100	2
D3	25-50	3
D4	12.5 - 25	4
D5	6.25 - 12.5	5
D6	3.125 - 6.25	6
A6	0 - 3.125	6

Low Level Statistical Features

For each frequency band of each channel of each window, the remaining time resolution (between 0.5 s and 15 s) was condensed into a scalar using a set of statistical features. The complete list of low level statistical features considered in this step and their definition can be found in table 3.3. In summary we now have 17 to 25 channels, every one of them being divided into 7 frequency bands on which in turn 8 basic features are computed. This leads to around 1000 features per window.

Table 3.3: Low level features and their definition used during feature generation. The original signal is denoted by x_t, with $t \in [0, n]$. Definitions partly from Guo et al. 2011 [8]. and MathWorks Documentation 2014 [12]

Name	*Definition*
Mean μ	$\frac{1}{n}\sum_{t=0}^{n} x_t$
Standard deviation σ	$\sqrt{\frac{1}{n}\sum_{t=0}^{n}(x_t-\mu)^2}$
Signal energy	$\frac{1}{n}\sum_{t=0}^{n} x_t^2$
Curve length	$\sum_{t=0}^{n-1} \lvert x_{t+1} - x_t \rvert$
Skewness	$\frac{1}{n}\sum_{t=0}^{n}\left(\frac{x_t-\mu}{\sigma}\right)^4 - 3$
Standard deviation of peak height Standard deviation of peak prominence Mean of peak prominence	*A local peak is defined as a data point, which is larger than the neighboring samples. Prominence is defined as the vertical distance between a peak and the lowest contour line encircling it and no higher peak.*

3.3.3 Classification and Validation

Different methods have been evaluated for the classification of same features. This guaranteed a meaningful comparison between the different feature extraction approaches tried, although a certain mutual dependence between the performance of the feature generation method and the classifier cannot be ruled out. More complex models than the ones described below were tested, but performance results clearly favored linear models.

To assess the classifiers' results, leave-one-out cross validation was applied. Special attention had to be given to choosing meaningful splits between training and test sets. During development it became clear, that consecutive segments often have a strong correlation, leading to over-optimistic results if one of them was in the training and the other one in the test set. For the results presented

in the next chapter, it was ensured that all preictal and interictal segments belonging to the same one hour phase (see figure 3.1) of recordings were in the same set. The number of sets for each patient was chosen to be the number of *independent preictal phases* as specified in table 3.1.

Linear Least Squares

A linear regression model was applied to the training data, with a response of 1 for windows belonging to a preictal and 0 for those belonging to an interictal segment. The coefficients $\boldsymbol{\beta}$ are obtained using the non-regularized linear least squares formulation

$$(\mathbf{X}^\top \mathbf{X})\boldsymbol{\beta} = \mathbf{X}^\top \boldsymbol{y} \tag{3.6}$$

whereby $\mathbf{X}$ denotes the features matrix and $\boldsymbol{y}$ the response of the observation.

Linear Discriminant Analysis

The LDA classifier is a fast and relatively simple model for two-class problems. Geometrically, LDA can be interpreted as projecting the features vector onto a weights vector and then determining on which side of a hyperplane this point comes to lie, thus specifying to which class it most probably belongs. The problem consists of finding the optimal weights vector $\tilde{\boldsymbol{w}}$, being defined as the weights vector $\boldsymbol{w}$, which maximizes the distance between the projected means $\tilde{\boldsymbol{\mu}}_i$ of our classes $i \in \{1, 2\}$ and minimizes the standard deviations $\tilde{s}_i$ within the projected classes. To summarize we want to find $\tilde{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} J(\boldsymbol{w})$, where

$$J(\boldsymbol{w}) = \frac{|\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{|\boldsymbol{w}^\top(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)|^2}{\sum_{\boldsymbol{x}\in C_1}(\boldsymbol{w}^\top(\boldsymbol{x} - \boldsymbol{\mu}_1))^2 + \sum_{\boldsymbol{x}\in C_2}(\boldsymbol{w}^\top(\boldsymbol{x} - \boldsymbol{\mu}_2))^2} \tag{3.7}$$

, μ_i is the mean of all features belonging to class i and C_i is the set of all observations x belonging to set i. For this project the routine `fitcdiscr` included in the Matlab® *Statistics and Machine Learning Toolbox* was used.

Regularized SVM

SVM classifiers aim at finding a weights vector, onto which the observations are projected, which maximizes the margin of a hyperplane dividing the two class sets. Specifically, it tries to achieve a hyperplane that has the largest distance to the nearest training data point of any class. The implementation used in this paper is LIBLINEAR [5], a library for the minimization of the following loss function with respect to the weights vector $\boldsymbol{w}$:

$$J(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + c\sum_{i=1}^{n} max(1 - y_i \boldsymbol{w}^\top \boldsymbol{x_i}, 0)^k \tag{3.8}$$

where $\boldsymbol{x_i}$ denotes the i-th observation, y_i its label and $c > 0$ a user-given penalty parameter. The exponent $k \in \{1, 2\}$ determines whether a L1 error function or L2 error function is applied.

3.3.4 Post Processing

As the described classifiers were trained on single windows, results have to be combined to obtain a result for an entire segment. Many options were evaluated for that (average of sigmoided output, majority vote, etc.) but we ultimately considered the mean of all windows in a segment, as it proved to be the most effective solution.

Although not relevant for the prediction of epileptic seizures in general, some further post processing was performed to optimize the results for the kaggle competition. This included normalizing the results per subject and shifting them by a subject specific value.

Chapter 4

Experiments and Results

All results were produced using the processing pipeline and parameters presented in the previous chapter.

4.1 Computational Performance

As described in section 3.3.2, a grid search was performed to find the optimal values for some parameters, such as window length and number of wavelet transformations. These computations and also the entire feature generation process, were performed on the Brutus Cluster of ETH Zurich. Features for each subject were computed individually on a 12-core AMD Opteron 6174 node with up to 256 GB RAM, of which 60 GB were actually used. An overview of the time needed to compute the features for each subject can be found in table 4.1.

Table 4.1: Timing of the feature generation pipeline on the Brutus Cluster (top; in minutes (min)); raw data size and recording time for scale (bottom; in minutes or gigabytes (GB)).

	Dog 1	*Dog 2*	*Dog 3*	*Dog 4*	*Dog 5*	*Patient 1*	*Patient 2*
Preictal segments [min]	1.0	1.6	5.2	6.7	2.4	2.0	3.2
Interictal segments [min]	16.5	33.7	97.2	54.1	30.3	4.5	6.2
Test segments [min]	17.2	69.3	60.4	67.0	12.5	15.1	20.3
Total [min]	**35**	**105**	**163**	**128**	**45**	**22**	**30**
Raw data file size [GB]	7.7	11.8	18.6	14.5	4.8	25.2	31.0
Raw data rec. len. [min]	10060	15420	24190	18910	6710	2630	2100
Ratio comp. time / rec. len. [1]	0.002	0.004	0.002	0.004	0.002	0.006	0.010

4.2 Classification Performance

Classification results are shown for all three methods described in section 3.3.3, namely LR, LDA and SVM. The quality of the results was estimated using n fold leave-one-out cross validation, whereby n ranged from 3 to 17. As measure of quality the *area under the curve* (AUC) of the *receiver operating characteristics* (ROC) was measured, which is the integral of the true positive rate (TP) against the false positive rate (FP) depending on the threshold t:

$$AUC = \int_0^1 \mathrm{TP}\,(\,\mathrm{FP}(t)\,)\,d\mathrm{FP}(t) \tag{4.1}$$

An optimal classifier would deliver an AUC of 1, meaning that there exists a threshold which yields 100 % TP and simultaneously 0 % FP.

A separate classifier was trained for each subject, as the data from the study subjects varied in numerous parameters, e.g. numbers of channels. Moreover the sensors were placed at different locations on the patients' brains. Thus the raw data is not directly comparable and a global classifier not sensible. The combined performance of all subjects can be assessed in two different ways:

1. **Average AUC:** The most meaningful measure, given one is interested in good results per subject, is to simply take the average of the AUC of the individual classifiers.

2. **Global AUC:** If aiming for a global classifier, the AUC can also be computed as a function of a *global* threshold, which is applied to all subjects' results in the same way. This punishes results where the optimal thresholds of the per-subject classifiers differ significantly. The score on kaggle.com is evaluated in this way, with an unknown weighting scheme.

The value for the AUC of all *individual* classifiers (from cross validation), the *average* of the individual classifiers (from cross validation), the *global* AUC with uniform weights (from cross validation) and the result given by *kaggle.com* (from unlabeled test data) using the proposed methods are summarized in table 4.2. LR delivered AUC scores between 0.2 and 0.99, LDA between 0.51 and 0.99 and regularized SVM between 0.56 and 0.91. For four subjects LR would be the method of choice, for one it would be LDA and for two the SVM model. If only the best scores per subject are considered, the results range from 0.77 to 0.99. Detailed ROC plots for all subjects are shown in section 4.2.1.

Table 4.2: Cross validation (CV) AUC results for all individual classifiers (top); Average AUC, global AUC computed locally using CV and global AUC of test data evaluated on kaggle.com (bottom). The best result for each subject is printed in bold.

	LR	*LDA*	*Regularized SVM*
Dog 1	**0.78**	0.63	0.56
Dog 2	**0.99**	0.99	0.87
Dog 3	0.91	**0.92**	0.78
Dog 4	**0.86**	0.85	0.84
Dog 5	**0.98**	0.96	0.87
Patient 1	0.63	0.77	**0.91**
Patient 2	0.20	0.51	**0.77**
Average AUC (local CV)	0.78	**0.80**	**0.8**
Global AUC (local CV)	0.84	**0.87**	0.80
Global kaggle.com AUC (test data)	0.55	**0.69**	0.60
kaggle.com rank [1]	331^{th}	**75th**	230^{th}

4.2.1 ROC Results per Subject

The following figures show the results of the best classifier for each subject. The following measures are shown:

- **Top left:** Receiver operator characteristics (TP vs FP rate).
- **Top right:** TP rate depending on the threshold value.
- **Bottom left:** FP rate depending on the threshold value.
- **Bottom right:** TP rate minus FP rate depending on the threshold value. The highest point of this function marks a possibility for an objective "optimal" threshold value.

[1] Hypothetical rank out of the 504 competing teams / individuals that would have been achieved if the submission had been made before the end of the competition. The actual final rank of the author as the competition ended in November 2014 was 99^{th}.

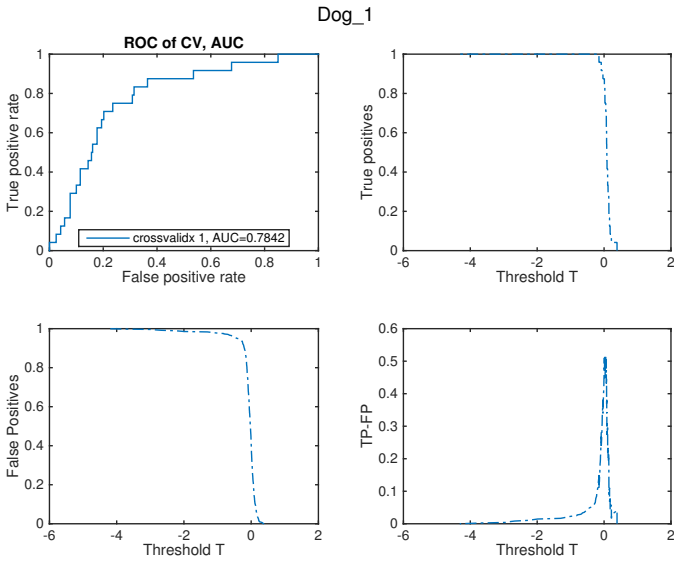

Figure 4.1: Receiver operator characteristics for Dog 1 with *linear least squares* classifier.

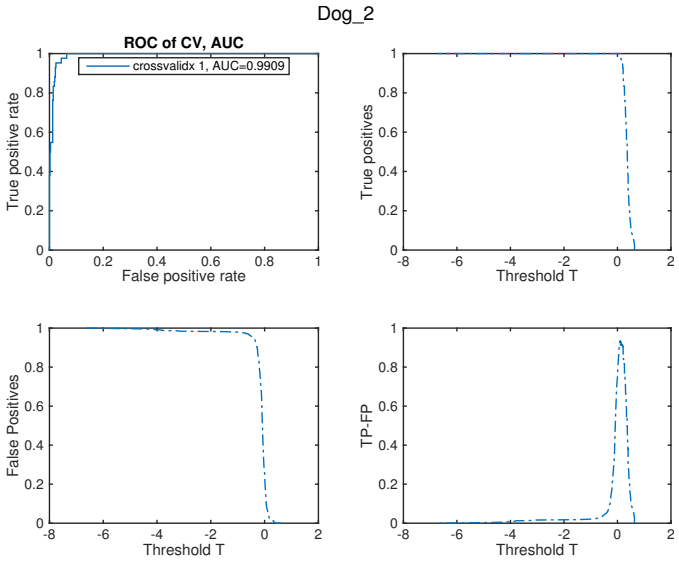

Figure 4.2: Receiver operator characteristics for Dog 2 with *linear least squares* classifier.

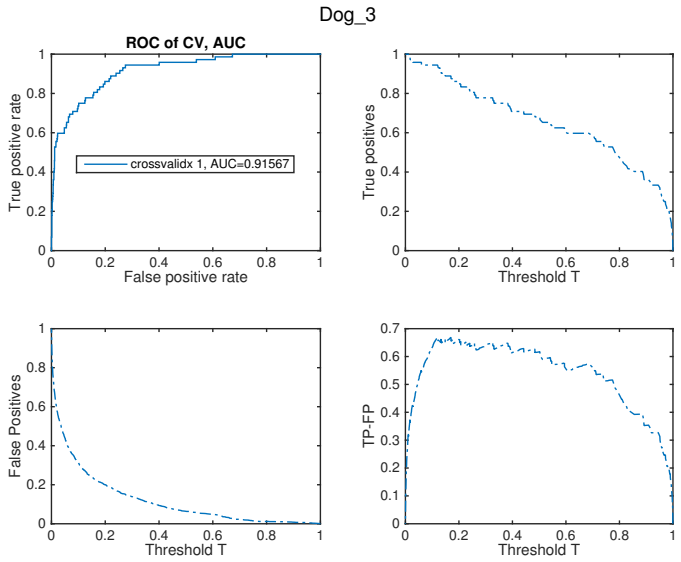

Figure 4.3: Receiver operator characteristics for Dog 3 with *linear discriminant analysis* classifier.

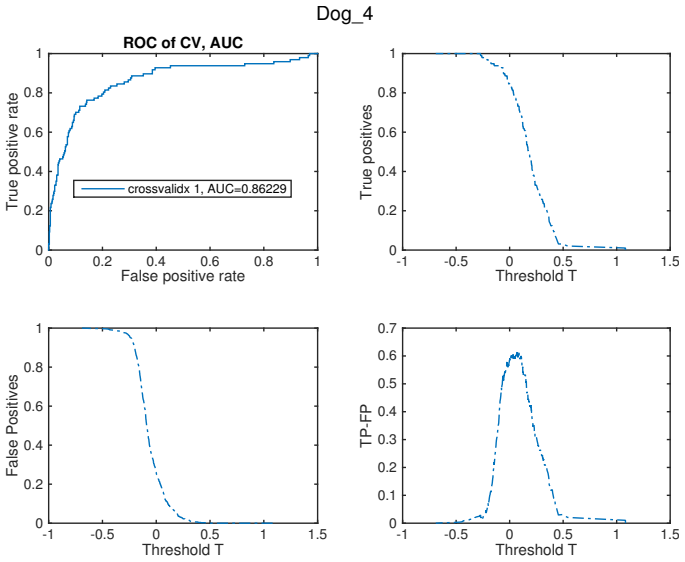

Figure 4.4: Receiver operator characteristics for Dog 4 with *linear least squares* classifier.

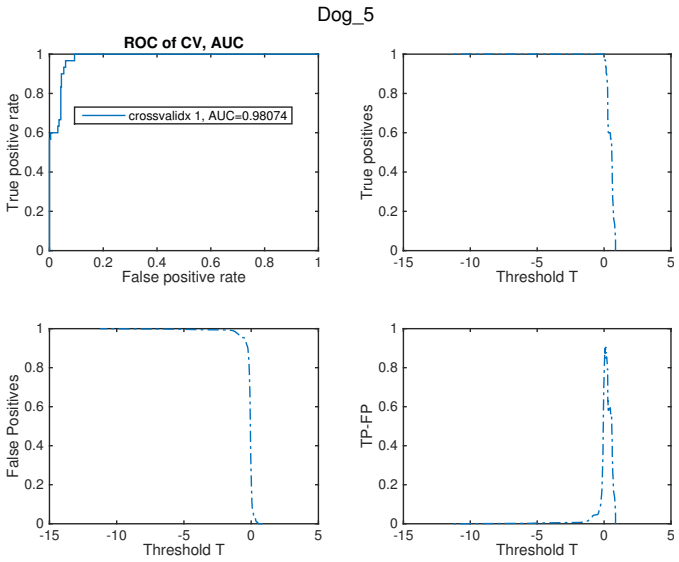

Figure 4.5: Receiver operator characteristics for Dog 5 with *linear least squares* classifier.

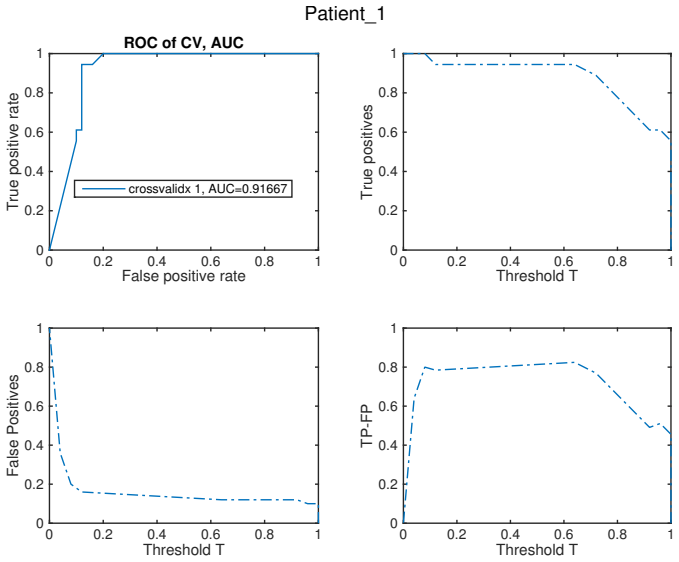

Figure 4.6: Receiver operator characteristics for Patient 1 with *regularized SVM* classifier.

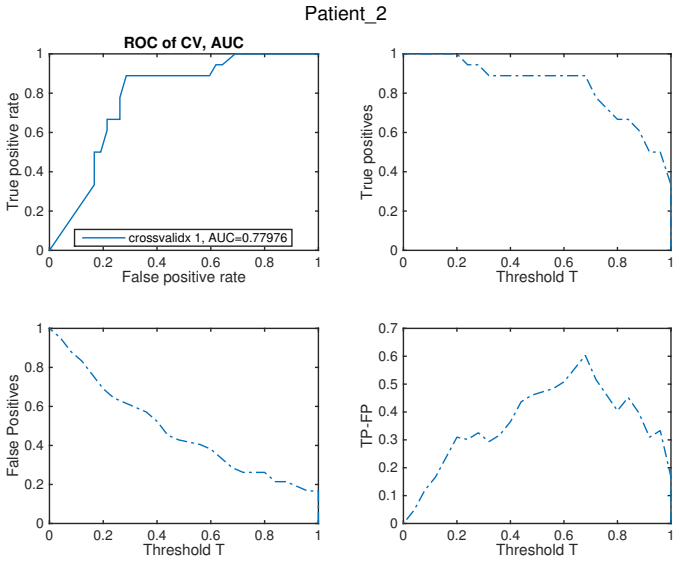

Figure 4.7: Receiver operator characteristics for Patient 2 with *regularized SVM* classifier.

Chapter 5

Discussion

5.1 Computational Performance

Although the computations are indeed very time consuming in absolute terms and were executed on high-performance hardware, the ratio of execution time over recording time is still very low, ranging from 10^{-3} to 10^{-2}. This allows the conclusion, that future applications in real time (i.e. where computation time is $\leq$ recording time) would very well be possible using the proposed method.

Performance could be improved substantially, if the code was written in a lower level language (e.g. C++) and be memory and cache optimized for the given problem.

5.2 Classification Performance

5.2.1 Comparison between tested Classifiers

The results vary drastically between the different subjects and classifiers. There is no general answer to which is the most suited classifier. Numerically, AUC ranged from 0.20 for the worst classifier / subject combination, which is worse than random, and 0.99 for the best one, which is very solid. Concerning the dogs, for whom in general much more training data was provided, taking simple linear least squares works best. Regularized SVM is well suited for the human patients, where regularization is very important due to small training sets. Globally, the linear discriminant analysis delivered the highest score locally as well as on kaggle.com.

Nevertheless, there remains a huge discrepancy between the globally computed AUC using cross validation and the AUC given by kaggle.com. Part of that divergence may lie in a different weighting scheme applied by kaggle.com, which might scale the influence of subjects with many test segments down and such with few test segments up. Hence, the score for the linear least squares classifier could definitely be improved by normalizing it in a reasonable way with information from cross validation, such that all subjects have a similar "optimal" threshold. The reason that the least squares classifier still shows a reasonably good global AUC score from cross validation is that the dogs – for which it performs well – make up almost 97 % of the total number of segments. However, this explanation

is not sufficient to explain the mentioned gap and further investigations would be needed to explore the cause thereof. The publication of the labels for the test segments would greatly facilitate this analysis. Unfortunately, up-to-date the organizers of the kaggle.com challenge have not responded to this request.

5.2.2 Comparison to other Publications

Directly comparing the classification accuracy to published research papers proves to be very difficult, as they are all based on other data sets. Therefore this section is limited to the comparison to other teams having participated in the kaggle.com challenge.

The three best performing teams were the only ones to achieve an AUC of over 0.8. The winner, Nir Kalkstein, who is a professional researcher at Israel based *MedialResearch*, reached a score of 0.84. His general approach is not too different from the one presented in this report, but developed from-scratch in native C [11]. As features he used the per channel and average energy in different frequency bands, correlation of energy in different frequency bands and some more basic statistical measures. These were then non-linearly transformed using logarithmic, square root and power functions to obtain around 900 to 1500 features per segment, which is fairly similar to the number suggested by the author. As classifier he applied a "highly regulated, iterative LS ensemble", which was tuned using one-out cross validation. A lot of emphasis was put on the regularization of the classifier, as he identified over-fitting to be the biggest challenge in the process. Some further post-processing was applied to the results of the classifiers, e.g. finding segments which are similar, thus probably belonging together, and taking the maximum of their results.

The findings of Mr Kalkstein align well with the ones of this report, suggesting that improving regularization of the classifiers is crucial for better overall results.

Chapter 6

Conclusion

The results of the proposed processing pipeline vary drastically between the different subjects. Therefore, the classifiers were optimized per subject, although this optimization can at least partially be automated using cross validation results. Taking the best classifier for each subject leads to an AUC range of between 0.77 and 0.99, which is a decent result compared to the other participants of the competition.

Besides the specific combination of features proposed in this work, the idea of adding new channels by projecting the original data onto PCA spaces computed from training sets is a new approach. This definitely generates valuable characteristics for later classification, however special attention needs to be paid to the portability to other subjects. In cases where no labeled training data is available for a patient – such that a classifier trained with data from another subject has to be used – more generic approaches might lead to better results.

Performance wise, the suggested algorithm is computationally demanding. Yet it has proven to be efficient enough to allow for real-time processing.

Future work heavily depends on two factors:

- **Amount of training data:** Assuming customized classifiers per patient, there is a strong probability that only little training data is available. Thus strong emphasis should be put on the development of highly regularized classifiers, whereby simplicity beats complexity, which favors linear classifiers.

- **Computing power:** Depending on the amount of processing capacity available, different strategies can be pursued. If operating on strong machines, then a tree like approach can be taken, considering different classifiers and optimizing every one of them using cross validation. Otherwise, the decision which classifier to use and which parameters to apply has to be made in advance.

Regardless of that, further features could be computed from the original data and existing ones could be combined in new ways, e.g. using non-linear functions. In particular the correlation between

different channels would be interesting to explore in depth, as many other competitors made use thereof.
The proposed methods should also be tested with data from different sources, to assess their robustness and genericness.

Bibliography

[1] U. R. Acharya, S. Vinitha Sree, G. Swapna, R. J. Martis, and J. S. Suri. Automated EEG analysis of epilepsy: A review. *Knowledge-Based Systems*, 45:147–165, 2013. doi:10.1016/j.knosys.2013.02.014.

[2] H. Adeli, Z. Zhou, and N. Dadmehr. Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123:69–87, 2003. doi:10.1016/S0165-0270(02)00340-0.

[3] M. Ahmadlou and H. Adeli. Wavelet-synchronization methodology: a new approach for EEG-based diagnosis of ADHD. *Clinical EEG and neuroscience : official journal of the EEG and Clinical Neuroscience Society (ENCS)*, 41:1–10, 2010. doi:10.1177/155005941004100103.

[4] E. Dawson. Getting Started Code by Elliot Dawson, 2014. URL: https://www.kaggle.com/c/seizure-prediction/forums/t/10312/getting-started-code-by-elliot-dawson.

[5] R. Fan, K. Chang, and C. Hsieh. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning*, 9:1871–1874, 2008. URL: http://dl.acm.org/citation.cfm?id=1442794, doi:10.1038/oby.2011.351.

[6] S. Ghosh-Dastidar and H. Adeli. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural networks : the official journal of the International Neural Network Society*, 22:1419–1431, 2009. doi:10.1016/j.neunet.2009.04.003.

[7] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr. Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection. *IEEE Transactions on Biomedical Engineering*, 54:1545–1551, 2007. doi:10.1109/TBME.2007.891945.

[8] L. Guo, D. Rivero, J. Dorado, C. R. Munteanu, and A. Pazos. Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications*, 38:10425–10436, 2011. doi:10.1016/j.eswa.2011.02.118.

[9] C. Heil. Ten Lectures on Wavelets (Ingrid Daubechies), 1993. doi:10.1137/1035160.

[10] Kaggle. American Epilepsy Society Seizure Prediction Challenge, 2015. URL: http://www.kaggle.com/c/seizure-prediction.

[11] N. Kalkstein. Description of solution by Medrr, 2014. URL: http://www.kaggle.com/c/seizure-prediction/forums/t/11024/our-solution.

[12] MathWorks. Signal Processing Toolbox: Documentation of findpeaks, 2014. URL: http://ch.mathworks.com/help/signal/ref/findpeaks.html.

[13] A. Subasi and M. I. Gursoy. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, 37:8659–8666, 2010. doi:10.1016/j.eswa.2010.06.065.

[14] Wikipedia. Wikipedia - Epilepsy, 2015. URL: https://en.wikipedia.org/wiki/Epilepsy.