CHALMERS
UNIVERSITY OF TECHNOLOGY

Machine learning for detection of epileptic seizures

Evaluation of features and machine learning methods for acceleration based classification of generalized tonic-clonic seizures

Master's thesis in Systems, Controls and Mechatronics and Biomedical Engineering

MADELEINE CZARNECKI

NICLAS GUSTAFSSON

Department of Signals & Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Master's thesis EX028/2015

Machine learning for detection of epileptic seizures

Evaluation of features and machine learning methods for acceleration based classification of generalized tonic-clonic seizures

MADELEINE CZARNECKI
NICLAS GUSTAFSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals & Systems
Division of Signal processing and Biomedical engineering
Chalmers University of Technology
Gothenburg, Sweden 2015

Machine learning for detection of epileptic seizures
Evaluation of features and machine learning methods for acceleration based classification of generalized tonic-clonic seizures
MADELEINE CZARNECKI
NICLAS GUSTAFSSON

Supervisor: Fredrik Ohlsson, Acreo Swedish ICT AB, Gothenburg
Examiner: Tomas McKelvey, Signals & Systems, Chalmers

Master's Thesis EX028/2015
Department of Signals & Systems
Division of Signal processing and Biomedical engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2015

Machine learning for detection of epileptic seizures
Evaluation of features and machine learning methods for acceleration based classification of generalized tonic-clonic seizures
MADELEINE CZARNECKI
NICLAS GUSTAFSSON
Department of Signals & Systems
Chalmers University of Technology

Abstract

This thesis evaluates the use of machine learning for classification of generalised tonic-clonic (GTC) epileptic seizures from acceleration data. The data used stems from measurements at Sahlgrenska Academy where patients undergoing diagnosis of epileptic seizures are observed. Acceleration measurements are performed on patients arms and sternum, with one triaxis accelerometer on each measurement location. A broad literature study of previous theses and research articles on epileptic seizure detection as well as machine learning literature is performed. Features that can be extracted from the accelerometer data and used to distinguish seizure events from non-seizure activity are identified in literature and studied further to enable construction of feature spaces that provide good classification performance. Machine learning methods are then applied to the extracted features and their performance evaluated by cross-validation. The methods evaluated are linear- and logistic regression, quadratic discriminant analysis (QDA), k-nearest neighbors (KNN), support vector machines (SVM) and random forest. The ability of the machine learning methods to correctly classify seizures when trained and tested on data from the same patient is evaluated. Training and testing on data from different patients is also performed to provide insight into how well methods generalise for multiple patients. The robustness of the machine learning methods against everyday activities that involve seizure-like movements such as toothbrushing, dishwashing and playing sports is tested. Results show that when training and testing on data from the same patient, all methods evaluated are able to perform well and correctly classify all seizures contained in test data. When tested on data sets from multiple patients, the machine learning methods generalise well, and are able to detect all seizures with the exception of linear- and logistic regression. Linear regression, KNN, SVM and random forest are found to be insensitive to seizure-like activities tested in this thesis, other methods suffer varying reductions in classification performance.

Keywords: Machine learning, epilepsy, GTC, generalized tonic-clonic seizure, accelerometer, gyroscope, classification

Acknowledgements

We want to thank our supervisor at Acreo Swedish ICT AB, Fredrik Ohlsson, for his involvement in the project and the support he gave us throughout. Also, we want to thank our examiner at Chalmers, Tomas McKelvey, for his technical advice during the project. A special thanks to Swedish Foundation of Strategic Research for the financial support of this project.

Madeleine Czarnecki & Niclas Gustafsson, Gothenburg, June 7

Division of work

Content produced together

- Feature extraction
- Preprocessing of data
- Background and literature study

Madeleine Czarnecki

- Decision layer
- Feature evaluation
- Evaluation of classification performance of ML methods
- Evaluation of gyroscope

Niclas Gustafsson

- Cross-validation evaluation
- Dimensionality reduction techniques
- Theory

Contents

1

Introduction

At Sahlgrenska Academy studies on epileptic seizures are made to broaden the understanding of the origin and evolution of the seizure in the brain. Epileptic seizures can manifest differently, where some involve large limb movements while others are less severe and exhibit no motor symptoms. To obtain quantitative measures of the seizures, the patients movements are recorded with accelerometers on patients that are undergoing clinical investigation. To improve the amount of relevant measurements it is desirable to be able to collect data from seizures occurring outside the clinic by means of wearable sensors [1].

It is desirable to use accelerometers with the purpose of collecting more data in environments outside the clinic. This type of seizure data will be of great help for both the clinicians, that can study the seizures occurring during the patients daily activities, and the patients, since they do not need to be comitted to the hospital as often. The use of accelerometers, to aid in diagnosis of epilepsy, has previously been investigated since EEG measuring that is normally used can be inflexible and uncomfortable for the patients [2], [3].

Acreo Swedish ICT AB is part of an SSF project concerning this type of system where they are aiming to develop a garment with built-in sensors to improve and simplify the diagnosis and treatment of neurodegenerative diseases such as epilepsy. An integral part of a system that can identify and classify the different types of epileptic seizures is the application of techniques based on Machine Learning (ML).

1.1 Purpose

The purpose of this project is to perform both classification and detection of generalized tonic-clonic (GTC) type epileptic seizures by using ML methods. The ML methods and features will be chosen and developed based on measured accelerometer data from patients at the observation unit at Sahlgrenska University Hospital where clinicians have distinguished seizures from normal movement activities using EEG. The machine learning methods and features should, after development, be able to provide clinicians with quantitative and relevant data for treatment and research on epilepsy.

1.2 Aim

The aim of the master's thesis project is to identify and evaluate ML algorithms, which are applicable to the problem of detecting and characterizing epileptic seizures of GTC type, and to derive a set of suitable sensor features for the application of the selected algorithms. After evaluation the project aims to investigate the performance of the evaluated features and methods in classifying GTC seizures in real patient data. The resulting classification methods are supposed to act as supplementary diagnostic data for physicians when diagnosing patients with epilepsy. This means that the classification will be performed offline, after conducting measurement sessions with patients. Evaluating and deriving the classification methods and combinations of features that provide the highest patient specific performance is the main focus of this thesis. How well the methods generalize for data obtained from several patients will also be investigated in this project for the purpose of future research. Also, the project includes evaluation of the classification performance of the ML methods when also gyroscope measurements are used in addition to the accelerometer measurements.

The measurements that will be used are acquired from patients that are mobile and performing a limited amount of everyday tasks. These tasks are somewhat restricted by what the patients are able to do while in the hospital, meaning that they do not fully mirror the movements that would be measured in day-to-day life. The resulting detection method must be robust against recorded movements that are similar to a seizure but stem from everyday activities. Since the available data does not fully represent daily movements, additional data from a selection of everyday activities such as tooth brushing, dishwashing and playing sports is recorded. This data is used to evaluate the robustness of the proposed classification methods.

1.3 Epilepsy

Epilepsy is the fourth most common neurological disorder characterized by its unpredictable seizures. The source of the seizures is the brain where electrical events can affect other parts of the body. There exists different types of seizures varying in intensity and duration, where an atonic seizure is short lived with muscles losing strength, compared to a generalized tonic-clonic seizure (GTC) that lasts longer and is more intense with forceful muscle contractions. The type of the seizure depends on several factors, e.g. where in the brain the seizure originates, how it spreads, which parts of the brain are affected and the duration of the seizure. All these factors influence the seizure characteristics. Epilepsy can also cause other health problems and is often a safety risk since people suffering from the disorder can fall helplessly to the ground with risk of injury to themselves. Some type of seizures can result in fatal complications, e.g. heart failure [4].

1.3.1 Treatments

A person is diagnosed with epilepsy after having two unprovoked seizures occurring at least 24 hours apart [5]. Knowing if a seizure is related to epilepsy or not can be difficult since there exists a range of other disorders that induce seizure-like behaviors. Deciding on treatment is difficult since an accurate diagnosis is not easily obtained. The goal of treating a patient with epilepsy is to reach a state where the seizures can be controlled or even completely stopped from occurring. There exist different types of treatments where anti-seizure medication is the most common. Other solutions could be trying different diets or even surgery in some cases. The patients from which the measurements used in thesis are obtained are all in the planning process of surgical treatment at Sahlgrenska Academy. The chosen treatment is based on what happens during the seizures, both the movements of the body and the brain activity and since a seizure rarely occurs when the patient is hospitalized it can be difficult for clinicians to administer the right type of care. Therefore there exists a need to gather information about the seizures outside of the hospital environment [4].

1.4 Prior work

There exists a number of researchers that have been exploring the possibility to classify epilepsy from different kinds of data, mostly acceleration and EEG with different approaches and results. A paper was published in 2005 by Nijsen et al. where they investigated the possibility of using a three-dimensional accelerometer for detecting motor seizures by evaluating the measures obtained by the sensors. The sensors where placed on the wrists, the ankles and the chest of the patients. They noticed that motor seizures easily could be visually detected in the accelerometer signal [6]. Nijsen et al. also published papers with their following work on the subject but with different focus and approaches. In the paper published in 2007 they wanted to find all motor activity that the accelerometer data contains by using machine learning methods. The features used are the change in acceleration and variance and the method used to classify is a linear threshold function [7]. These two papers serves as a basis for understanding how different seizures are registered by accelerometers depending on the different seizure motor characteristics and how machine learning can be used in a basic classification manner with satisfactory results.

In a paper published in 2008 Nijsen et al. begun work on distinguishing activity generated by tonic seizures from normal movements [3]. The signal is processed by filters to extract the slow movements generated by the tonic seizure together with the fast components since tonic seizures often are accompanied by a subtle tremor. Different features are calculated to distinguish the two classes, both from the low and high motion components and by using features that distinguishes tonic seizures from other motor seizures, by using continuous wavelet transform on the other seizures. The results give a sensitivity of approximately 0,80 and a positive

predicted value of 0,35. They explain that the low PPV comes from that some of the false positives are seizures but not of tonic type and therefore it is an expected value [3].

An additional research paper by Nijsen et al. published in 2010, time domain and frequency domain methods are evaluated when classifying myoclonic seizures (short, shock-like jerks of a muscle) from accelerometer data. The methods used are short-time Fourier transform (STFT), the Wigner distribution (WD), the continuous wavelet transform (CWT) and model-based matched wavelet transform (MOD). Since myoclonic seizures are brief they can be matched by a wavelet in an easy way, the signal also contains higher frequency contents for a seizure than for normal movements. The results for classifying myoclonic seizures show that wavelets get higher sensitivity for time-frequency methods but get more false positives which lowers the specificity [8].

Another paper published in the subject of classifying convulsive seizures, e.g. GTC seizures, with the use of electrodermal activity and accelerometer sensors, is written by Poh et al. They used Support Vector Machine for classification, which is a supervised learning algorithm, together with 19 features, where 16 of them were calculated from the data extracted by the accelerometers. Some features are derived from the frequency domain of the data. To evaluate the estimated model double cross-validation was used, which is a type of cross-validation that divides the data into two samples and uses both as training and testing sets. They managed to detect 15 of 16 seizures from different patients with a total of 130 false alarms, approximately one per 24 h. When only the features from the accelerometer were used for classification the number of seizures detected was the same but with a higher false alarm rate, approximately 1.5 per 24 h. In the discussion it is mentioned that many of the false alarms were triggered by activities containing rapid and forceful movements, e.g. dice shaking, juggling and motion-controlled gaming. Increasing the number of seizures correctly classified and lowering the rate of misclassifications by using training data containing more seizures was not investigated in the paper due to a limited amount of available data. The amount of data containing normal activity gave a good estimation on the false alarm rate [2].

Lockman et al. investigates the use of a watchlike, accelerometer based device to detect tonic-clonic seizures. The study covers 40 patients of which 6 suffered from tonic-clonic seizures. The device contains a triaxis accelerometer and the signal from each axis is used as a feature in detection of seizure activity. The detection algorithm incorporated in the device uses both time domain and frequency domain properties in conjunction with several thresholds and conditions specified by epileptologists to detect seizures. The study found that using acceleration data for detection of tonic-clonic seizures was possible, however there were issues of false alarms caused by rythmic everyday activities [9].

The possibility of detecting GTC seizures with a Nintendo Wii© remote is evaluated by Shulc et al. by implementing a threshold based algorithm. The algorithm uses different measures associated with the vector magnitude(VM) of the acceleration

signal together with thresholds for the duration and intensity of the movements that exceeds threshold values. The frequency content of the acceleration signal was studied and found to be above 2 Hz for GTC seizures. The proposed algorithm produced sensitivity of 100% and specificity of $\geq 88\%$ for the three patients involved in the study, which is deemed to be encouraging for future research [1].

In a paper published in the subject of detecting GTC seizures using a wrist-worn accelerometer the decision making for detecting a seizure was done in real-time based on a fixed threshold that was set to detect high acceleration. All patients were able to move freely in the facilities which led to false alarms when patients brushed their teeth or participated in activities involving rhythmic movements [10].

Acreo Swedish ICT AB has already been a part of two Master's Theses regarding sensors attached on patients having epilepsy which were performed by two students, J. Wipenmyr and A. Hildeman [11], [12]. Both theses measured the acceleration by having three sensors connected to the patients wrists and chest. The seizures that they aim to classify are not only of GTC characteristics but also tonic seizures. Also, A. Hildeman tried to expand the number of measurements by simulating seizures by attaching sensors on a healthy person and mimic a seizure with small jerkings decreasing in frequency and increasing in amplitude. He concluded that the simulations did not resemble a true seizure. The main focus of both projects has been the data acquisition, how to measure and how to process the data into usable files. The measurements were then used to calculate different features, both in time-domain and in frequency domain. A. Hildeman evaluates different clustering methods, which are types of unsupervised learning that creates clusters of data having similar characteristics. For classification he uses linear and quadratic discriminant analysis (QDA) and k-nearest-neighbors (KNN). Also a transition matrix containing information of a seizure's order and duration was developed. To avoid curse of dimensionality two methods where used, principal component analysis (PCA) and partial least squares discriminant analysis (PLS-DA). The evaluation methods that were used to examine the performance of the different classification techniques are resubstitution and cross-validation, where the latter gives a more accurate estimation of the performance.

Tests that were performed on the different methods and features were: best performance of the classification method, best parameters for each patient, discriminatory information in features and comparing the results with J. Wipenmyr's classifier [11]. The best performance was achieved with KNN, producing a sensitivity and a specificity of 0,927 and 0,835, respectively, for a patient with GTC seizures and 0,718 and 0,317, respectively, for a patient having other type of seizure. By using KNN with $k = 5$ nearest neighbors the best trade off between sensitivity and specificity was achieved compared to QDA that gave high sensitivity but lower specificity. By reducing the features concerning the magnitude of the acceleration, the performance is increased. Also, it seems that the number of sensors used affects the performance of the classification. The method used by J. Wipenmyr gave lower sensitivity and specificity than the KNN method used by A. Hildeman. The low evaluation measures for the patient with seizure not having GTC characteristics could depend on

the tonic seizures, which involve less rapid motions than GTC seizures which the other patient suffers from. He also concluded that a generalization of the classification method seems possible between different patients having same type of seizures [12].

2

Background

2.1 Seizures

The type of seizures that this project aims at classifying is GTC seizures since their characteristics should be measurable with an accelerometer and also separable from most normal movements. From a clinical point of view, patients having GTC seizures are prioritized for treatments since they are the most severe and dangerous. This project is an important first step towards a more general classification where a wider range of different seizure characteristics can be classified in addition to GTC seizures.

2.1.1 Generalized Tonic-Clonic seizures

Generalized Tonic-Clonic seizures are characterized by two phases starting with all muscles contracting making the body stiff, which is the tonic phase. The stiffness stems from the muscles contracting at such a high frequency that they do not have time to relax between contractions. Air is forced past the vocal cords due to the abdominal muscles contracting, which results in a typical groan or cry. After some time the muscles contract less and less frequently. This results in high frequency motions with small magnitudes which transition into low frequency motions with high magnitudes. This transient phase is referred to as the clonic phase. This type of seizure lasts 1 to 3 minutes in general. In Figure 2.1 the measurements from an accelerometer placed on the wrist during a GTC seizure is shown.

After a seizure, consciousness slowly returns, and the patient is normally drowsy, confused and exhausted. It is not uncommon that bladder or bowel control is lost as the body relaxes after a seizure. GTC seizures are sometimes called grand mal seizure due to their dramatic nature.

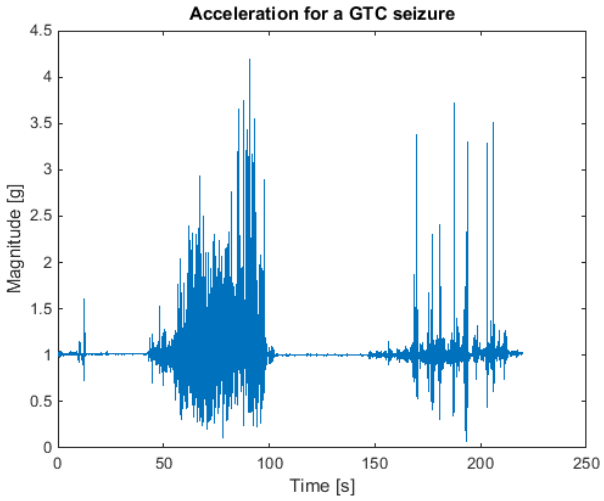

Figure 2.1: Acceleration for a GTC seizure followed by normal activity.

2.2 Measurements

The measurement data that this work is based on was collected from patients undergoing diagnosis and observation at Sahlgrenska University Hospital. The measurements are performed with three tri-axis accelerometers with one on each wrist and the third placed on the sternum. The sensors stream their measurements to a nearby PC via radio communication where the data is recorded. Acceleration data is collected at a sample rate of 50 Hz or 102,4 Hz.

Visual inspection of the measurement data received from the sensor placed on the sternum revealed large segments of missing samples caused by connection issues between the sensor and the recording PC. The measurement data from the sternum sensor is of such low quality that it is excluded from further analysis. During the measurement sessions the patients are also being monitored with EEG (electroencephalography) and video. The EEG and video recordings are reviewed by physicians which annotate the recorded data with the times where confirmed GTC seizures start.

2.3 Patients

The collected data comes from different patients and therefore the characteristics between the measurements can differ. All the patients presented in this chapter have had GTC seizures. A seizure registered by the hospital staff may not always look like a GTC seizure and that could be due to faults in the sensors, that the arms of the patient where stuck and could not move freely, that the hospital staff have intervened during the seizure or that it was incorrectly registered.

2.3.1 Patient 7

Two separate measurement occasions have been performed with this patient.

2.3.1.1 Measurement A

For this measurement 12 seizures annotated by physicians where recorded and one of them does not seem to have the GTC characteristics although the clinical log marks it as a GTC seizure. Access to video recordings and EEG measurements were restricted during this project which prevented confirmation of the true nature of this seizure and therefore this seizure is not used during tests to not compromise the reliability and comparability of the results. All other measurements during seizures behave similarly but one of the seizures also have missing data where the sensors have lost the contact with the recording computers. A total of 51 hours of measurement exist. In Figure 2.2 the accelerometer data and the energy in frequency bands are shown for a typical seizure. The sampling rate is 50 Hz.

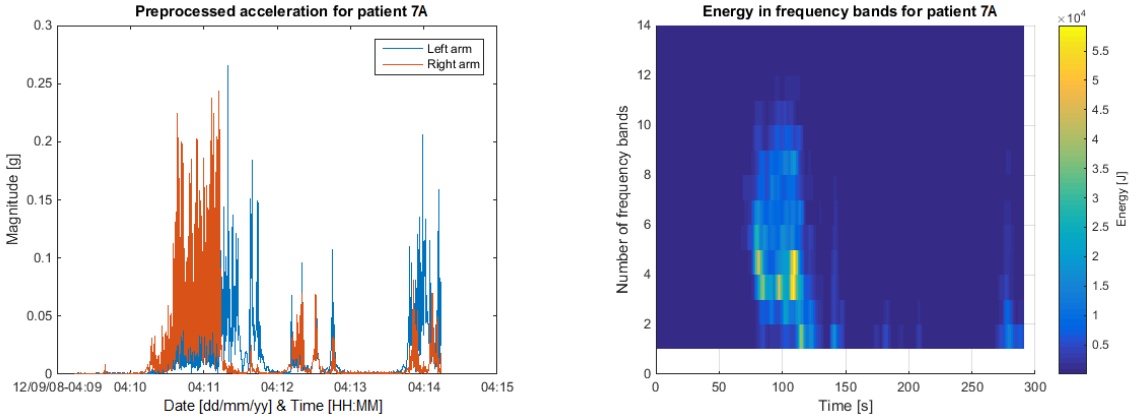

Figure 2.2: Acceleration and energy in frequency bands for a GTC seizure in Patient 7, measurement A. The frequency bands 1-14 in the wright plot are equally spaced between 0.75-11.25 Hz. The seizure starts at approximately 04:10:10 and ends at 04:11:30. The seconds in the right figure indicates the duration from the start time in the left figure which is 04:09:15.

2.3.1.2 Measurement B

The difference between this measurement and the first one is that the patient got intracranial EEG electrodes operated in to the head between the two occasions. This measurement contains 13 seizures where one of them has lower magnitude and does not follow the same pattern as for the other seizures. Also, one of the sensors has a significant amount of missing data during this seizure and therefore it is not used in further testing. All other measurements during seizures appear to behave similarly. A total of 182 hours of measurement exists. In Figure 2.3 the accelerometer data

and the energy in frequency bands are shown for a typical seizure. The sampling rate is 50 Hz.

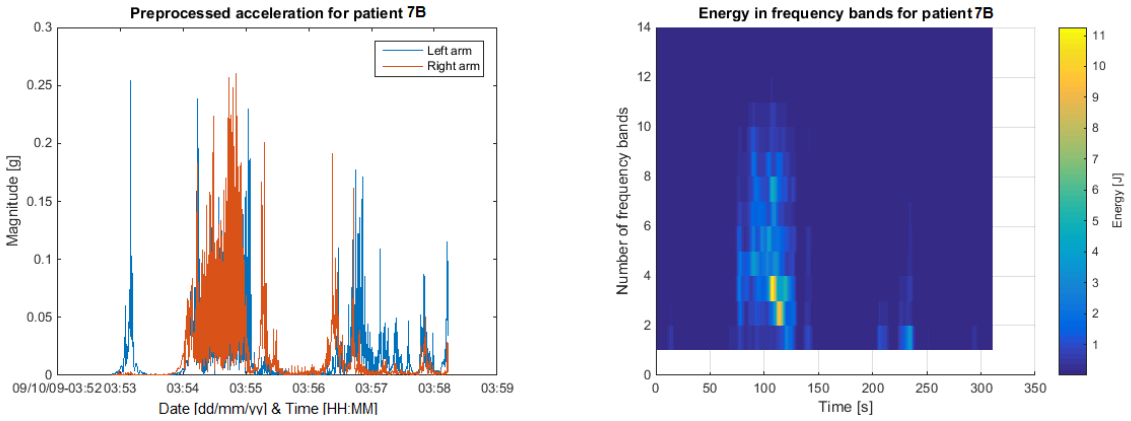

Figure 2.3: Acceleration and energy in frequency bands for a GTC seizure in Patient 7, measurement B. The frequency bands 1-14 in the wright plot are equally spaced between 0.75-11.25 Hz. The seizure starts at approximately 03:53:55 and ends at 03:55:05. The seconds in the right figure indicates the duration from the start time in the left figure which is 03:52:50.

2.3.2 Patient 48

This patient has two seizures, both having similar behaviour in the measurements. A total of 96 hours of measurements exists. In Figure 2.4 the accelerometer data and the energy in frequency bands are shown for a typical seizure. The sampling rate is 50 Hz.

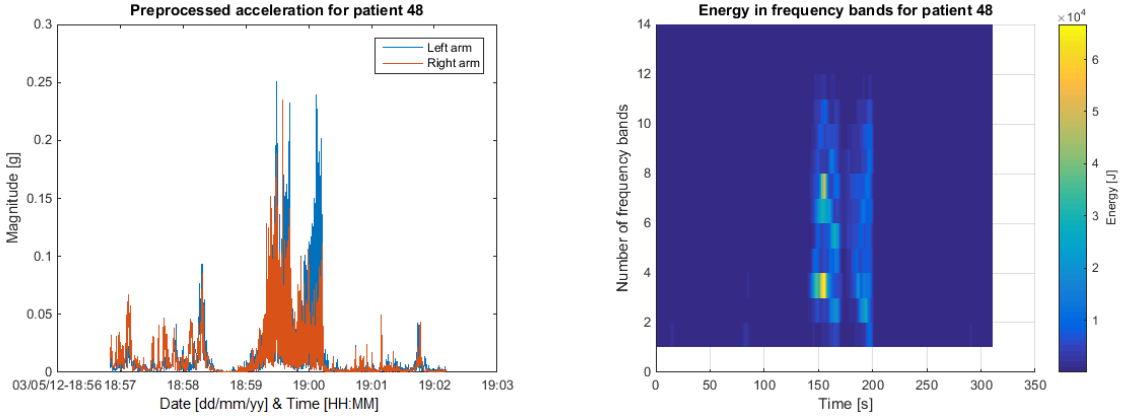

Figure 2.4: Acceleration and energy in frequency bands for a GTC seizure in Patient 48. The frequency bands 1-14 in the wright plot are equally spaced between 0.75-11.25 Hz. The seizure starts at approximately 18:58:45 and ends at 19:00:20. The seconds in the right figure indicates the duration from the start time in the left figure which is 18:58:50.

2.3.3 Patient 55

This patient has two GTC seizures, both seizures having the same behaviour. Unlike the other patients, only two sensors were used during measurements, placed on the upper arms and the data was locally stored for later extraction. This patient also has measurements from a gyroscope. A total of 25 hours of measurement exists. In Figure 2.5 the accelerometer data and the energy in frequency bands are shown for a typical seizure. The sampling rate is 102,4 Hz. Patient 55 is a more active individual compared to the other patients and performed various physical exercises during measurements. This is important to note since the data from this patient contains higher activity levels that could possibly lead to misclassifications if some movements produce measurement data similar to a seizure. If the classification methods can correctly classify data from this patient, it indicates that the proposed features and methods are robust against high activity levels in non-seizure related measurements.

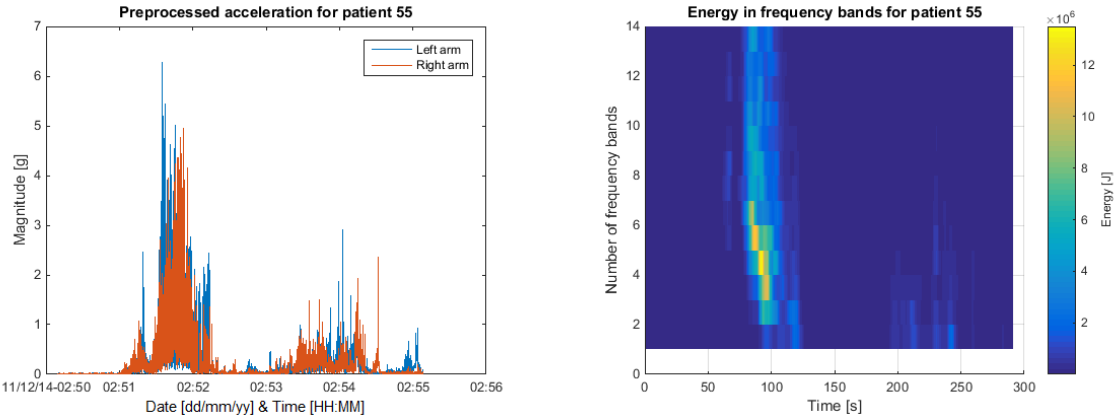

Figure 2.5: Acceleration and energy in frequency bands for a GTC seizure in Patient 55. The frequency bands 1-14 in the wright plot are equally spaced between 0.75-11.25 Hz. The seizure starts at approximately 02:51:00 and ends at 02:52:10. The seconds in the right figure indicates the duration from the start time in the left figure which is 02:50:10.

2.3.4 Summary of patient characteristics

Each patient has rapid movements during the seizures and in most cases the majority of the energy content of the signal falls into frequencies above 2 Hz. It is visible that even though the maximum energy in measurement B from Patient 7 is not placed in bands over 2.25 Hz that a majority of the energy content is in the higher frequency bands during seizures. The seizure characteristics observed in the measurement data is consistent with characteristics described in previous studies.

3

Theory

In this chapter the theories and concepts behind statistical methods used in this thesis are explained. The implementation and usage of the contents of this chapter is explained in Chapter 4.

3.1 Feature extraction

An essential step to perform, prior to classification, is to extract features from the measured data. Finding the right type of features is as important as finding the best ML method since the features contain the information that the methods use for building their models. The features serve to enable the classification methods to distinguish seizures from measured normal, daily activities that are not seizure-related. The computed features are therefore designed in such a way that the characteristic pattern of a GTC seizure will become easily detectable. A feature containing information not distinguishing the seizure from normal activity can deteriorate other features performance and therefore the feature combination in feature spaces also may be important.

All features are calculated to provide a value for each second in the time series. A Hamming window, w_i, is used in order to reduce the impact of edge effects. Since a GTC seizure can last for 1 to 3 minutes, some features are to be computed using an overlapping, sliding window over 10 seconds. The overlap serves to capture feature information over a longer period of time than one second. Some of the used features are based on previous work and some are newly designed, based on knowledge of the characteristics of GTC seizures.

3.1.1 Vector Magnitude

Vector magnitude measures the overall activity of a sensor based on the vector magnitudes in combination with a window function.

$$VM = \frac{\sum_i (\sqrt{x_i^2 + y_i^2 + z_i^2}) \cdot w_i}{\sum_i w_i}$$

3.1.2 Signal Magnitude Area

SMA uses the mean value of the absolute values for the three axes per sensor in combination with a window function.

$$SMA = \frac{\sum_i (\frac{|x_i| + |y_i| + |z_i|}{3} \cdot w_i)}{\sum_i w_i}$$

3.1.3 Energy content in frequency band

This feature calculates the energy in the sensor signals for chosen frequency intervals. In previous work the width of the frequency bands were wide and to achieve higher frequency resolution the bands are narrowed. The frequency bands used in this project are ranged from 0.75 to 11.25 Hz in increments of 0.75 Hz which creates 14 bands. Signal energy is not calculated for frequency bands higher than 11.25 Hz since it is reasoned that no human motion activity can realistically be present at higher frequencies [2]. Excluding higher frequencies in the energy calculations also serves to minimize the impact of high frequency disturbances that can be caused by interference in the radio communications between sensor and receiver [12].

In order to accurately calculate the energy content of a segmented signal, discrete time STFT is used. The transform uses a Hamming window in conjunction with overlapping signal segments to reduce artifacts at the boundaries between segments [13]. The total energy of the signal in a band of frequencies is then calculated for each sensor as the sum of squared absolute values of the transform.

This feature enables distinction between different activities based on the amount of energy they contain and at what frequency they are performed. The energy content in each band is computed as a spectrogram, so that the energy content of the signal can be studied for each time instance and over a time series. In the current implementation the spectrogram is calculated for each second with an overlap that corresponds to the desired window length of ten seconds.

3.1.4 Frequency Peak

The frequency peak feature is extracted by calculating in which of the frequency bands from the previous feature the sensors have their maximum energy content.

This feature is calculated per second and sensor. After calculating in which band the energy peak lies, a moving average filter is used over the time axis to smooth the resulting curve. The idea behind this feature is to clearly be able to separate seizure activity from other activities, as seizure activities contain energy in higher frequencies. Voluntary movements normally contain frequency content below 2 Hz, the involuntary movements a patient exhibits during a GTC seizure be of higher frequency than 2 Hz [2].

3.1.5 Root mean square, mean value and accumulated acceleration

Based on previously research performed by Poh et.al, the mean value and root mean square value for the acceleration signal are calculated for each second and used as features. The magnitude of each axis is summed over an epoch of 10 seconds for each sensor providing a feature that represents the accumulated acceleration over a longer period of the time series [2]. These features are calculated after the measurement data has been pre-processed which removes the gravitational DC-component and non-activity data. The features are meant to serve as measures of general activity.

3.1.6 Variance and standard deviation

The variance and standard deviation of each sensor is calculated with a 10 second overlap in order to capture the transient behavior of a seizure and suppress the influence of short jerks. Both variance and standard deviation are evaluated as possible features since both measures have been used in previous research [2], [3].

3.1.7 Correlation

The correlation between sensors is used as a feature with the purpose of capturing simultaneous movements during a seizure. Non-seizure related activities will not be evident in this feature since such activities are less likely to involve simultaneous movement of the sensors. The correlation coefficient is calculated by treating each sensor's three-axis resultant vector as a random variable and calculating the sample based correlation between them.

$$CORR(X,Y) = \frac{1}{N}\sum_{i=1}^{N}(X - \mu_X)(Y - \mu_Y) \tag{3.1}$$

where the sample based mean, μ, values used in the correlation calculations are computed for each second, producing a correlation coefficient for each second in the time series. X and Y denote the three-axis resultant data vector of each sensor, N is the number of samples obtained during each second.

3.1.8 Entropy

Entropy is, in general, seen as a measure of information content. Its application to this work is the idea that when estimating the information content of the measured acceleration data, an estimate of the overall activity of the sensors is obtained. In theory the difference in activity levels between seizure and non-seizure events will contain significantly different levels of entropy, enabling good class separation of the feature data.

The entropy of the acceleration signal is computed each second by producing a histogram over a 10 second epoch centered at the current time instance. The histogram function in **Matlab** [1] is used to estimate the probability density function $f(x)$ for each computed histogram:

$$f(x) = \frac{h_k(x)}{\sum_k(h_k(x))} \tag{3.2}$$

where h_k represents the counts for each bin k = 1,2,...,30. The entropy can then be approximated as:

$$H(X) = -\sum_k(f(x)\log f(x)) \tag{3.3}$$

[14]. The number of bins for the histogram is 30, this number of bins is chosen to provide sufficient resolution of the information content in the acceleration signal.

3.2 Dimensionality reduction techniques

Following feature extraction the feature data span a n-dimensional space, where n is the number of features. This space is referred to as the feature space. For large data sets and large amounts of features, the feature space becomes increasingly dense, which increases the computational effort needed when fitting models to the feature data. However the main issue with high dimensional feature spaces is the curse of dimensionality. Intuition may make it seem reasonable that the more features used in training a model, the better the fit. This intuition falters for high dimensions since the sampling density is proportional to $N^{\frac{1}{p}}$, where p denotes the dimension of the feature space and N is the sample size [15]. Since measurement data is not infinitely abundant, trying to span a high dimensional feature space with limited data leads to a sparsely populated space that does not accurately describe the distribution of

[1] The built-in histogram function in **Matlab** calculates the histogram over a specified number of bins.

data throughout the feature space. Trying to fit models to the sparse feature space will lead to poor fit and in turn poor classification performance.

To reduce the performance degradation caused by the curse of dimensionality, there exists methods which serve to describe the quality of the data in terms of how well it describes the classes and how much variance the data contains. Low quality data can further degrade performance since features that do not provide enough information to differentiate classes will not make for good training data.

3.2.1 Principal component analysis

Principal component analysis, referred to as PCA from hereon, is a method used to linearly transform the feature set X into orthogonal components and score them by how much of the variance of the complete set they contain. This is done with the goal of using only a few components to describe a large part of the variance in X. To achieve this, the eigenvectors of the sample covariance matrix P are analyzed. The eigenvectors of P are the principal components of X and their eigenvalues represent the significance in terms of how much of the variance of X the principal components describe. The most significant components of X can then be kept and used as a new feature set in classification [16].

3.2.2 Partial least squares regression

One drawback with PCA is that only the variance in data is considered, regardless of how correlated the variance is with the response classes. The goal of the partial least squares method is to reduce the feature set with respect to both variance and correlation with the response classes. The components are calculated using both the feature space X and the class vector y, directions in X that maximize both variance and correlation with y are then sought. PLS components are then in theory more suited for classification purposes since the components kept for classification are guaranteed to contain information related to the response classes, and not only have high variance [15].

3.3 Classification methods

In this chapter all ML methods used in the project is stated together with a brief explanation of their properties and underlying theory. Classification is the act of applying a model of a classification method to a data set X, and obtaining the predicted classification $\hat{y}$ of the classes y contained in X. The data set X is also referred to as the feature set. For the purpose of this thesis, X is a matrix where rows correspond to time instances and columns represent each feature. Before using

a machine learning method for classification, the model intended to be used must be fit to a specific feature set, this feature set is referred to as the training set, X_{train}. After training a model to X_{train}, it can be applied to the data set X_{test} that is to be classified, this set is referred to as the test set. When evaluating machine learning methods it is important to train and test them on separate feature sets. If a method is trained on data that it is later tested with, it may perform very well. However if the training and testing is performed with the purpose of evaluating how suitable a machine learning method is to separate a distinct selection of classes, as is the case in this thesis, training and testing must be performed with separate feature sets to give a correct picture of how the method performs.

3.3.1 Linear regression

Linear regression is a simple approach to classification where the boundary between classes is approximated as linear functions in feature space. For each feature in a training set X_{train} of size Nxp, a coefficient β is estimated. The estimation is performed by minimizing the residual sum of squares(RSS). The residuals are the difference between the true class variables in $\mathbf{y}$ and the class prediction $\hat{\mathbf{y}} = f(x_i)$:

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - f(x_i))^2 = \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \tag{3.4}$$

The RSS is minimized with least squares and gives a vector of coefficients $\boldsymbol{\beta} = (\beta_0,\beta_1,...,\beta_p)$ that are used for classification. Classification with linear regression is performed by multiplying the set of coefficients with the test set X_{test}

$$\hat{\mathbf{y}} = X_{test}\boldsymbol{\beta} \tag{3.5}$$

The scores in $\hat{\mathbf{y}}$ are the predicted classifications of the classes in $\mathbf{y}$ [15], [17]. Since $\hat{(y)}$ is a continuous variable, it needs to be translated into a discrete variable that corresponds to the two classes in order to correctly determine the predicted classes.

3.3.2 Logistic regression

Logistic regression is a model based classification method that enables modelling of the classes posterior distributions by means of linear functions. Furthermore the logistic regression model ensures that the linear functions used in modelling the posterior probabilities sum to one and remain in the range [0,1] [15].

Models are fit by maximum likelihood and for binary classification of two classes

the logistic regression model has the following general form:

$$\begin{aligned}
\log\frac{P(G=1|X=x)}{P(G=K|X=x)} &= \beta_{10} + \beta_1^T x \\
\log\frac{P(G=2|X=x)}{P(G=K|X=x)} &= \beta_{20} + \beta_2^T x \\
&\vdots \\
\log\frac{P(G=K-1|X=x)}{P(G=K|X=x)} &= \beta_{(K-1)0} + \beta_{K-1}^T x
\end{aligned} \tag{3.6}$$

where G denotes the class vector, K represents the number of classes. In the case of binary classification (two-classes), $K = 2$ the model only consists of a single linear function. Since the model is fit using maximum likelihood to specify $P(G|X)$ which in turn completely specifies the conditional distribution, the multinomial distribution is suitable. For N observations the log-likelihood is

$$l(\theta) = \sum_{i=1}^{N} \log(p_{g_i}(x_i;\theta)) \tag{3.7}$$

where $\theta = \{\beta_{10}, \beta_1^T, ..., \beta_{(K-1)0}, \beta_{K-1}^T\}$ and $p_k(x_i;\theta) = P(G = k|X = x_i;\theta)$. For the case where y contains two classes, the log-likelihood can be written

$$\sum_{i=1}^{N} \{y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})\} \tag{3.8}$$

To find the optimal β-parameter, the log-likelihood function needs to be maximized. The derivatives are set to zero:

$$\frac{\Delta l(\beta)}{\Delta \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i;\beta)) = 0 \tag{3.9}$$

Solving for the maximum is then achieved with the Newton-Raphson algorithm, resulting in a β-parameter that describes the optimal model of the decision boundary between the two classes [15].

3.3.3 K-Nearest Neighbors

The nearest neighbor algorithm is one of the simplest machine learning algorithms, as it is memory based and does not require a model to be fit to the data. For every data point $x_0 \in X_{train}$ in the feature space, the training points in each corresponding

row in X_{train}, $x_{(r)}$,$r = \{1,...,k\}$ closest in distance to x_0 are found. Classification of x_0 is then performed by a majority vote between the k neighbors. The distance between neighbors is normally measured as Euclidean distance. This requires that all features are normalized before classification since they may be measured in different units. The normalization serves to make the features comparable in distance. The number of neighbors used in classification can be used as a tuning parameter for increasing performance. In general, the higher the number of neighbors used, the smoother the obtained decision boundary will be. A low number of neighbors can result in a noisy decision boundary, which can be useful in some cases where data is not easily separable or neatly clustered [15].

3.3.4 Quadratic Discriminant Analysis

Discriminant analysis methods are based on the presumption that the posterior probability density for each class can be modelled as a multivariate Gaussian density, and the classes can be described by applying Bayes theorem [15]. In a two-class ($K = 2$) problem where the classes are $G = k$ or $G = l$, the class-conditional density of X, $f_k(x)$, describes the class-conditional density of X in the class $G = k$. The prior density of class k is π_k. Achieving optimal classification is then dependent on obtaining $\Pr(G|X)$. Application of Bayes theorem gives

$$Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l} \tag{3.10}$$

The class $G = k$ is modelled as a multivariate Gaussian density:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma_k}|^{1/2}} e^{\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma_k^{-1}} (x-\mu_k)} \tag{3.11}$$

When computing the decision boundary for linear discriminant analysis, the classes are assumed to have a common covariance matrix $\mathbf{\Sigma_k}$. This assumption is not made for quadratic discriminant analysis. In order to estimate the posterior density of each class with quadratic discriminant analysis, each class needs to have its covariance matrix estimated. The quadratic discriminant analysis function for the class $G = k$ is defined as:

$$\delta_k(x) = -\frac{1}{2}\log|\mathbf{\Sigma_k}| - \frac{1}{2}(x - \mu_k)^T \mathbf{\Sigma_k^{-1}} (x - \mu_k) + \log\pi_k \tag{3.12}$$

In order to perform classification with 3.12 the following decision rule is formed:

$$\hat{G}(x) = \arg\max_k \delta_k(x) \tag{3.13}$$

The decision rule in 3.13 finds the class k that maximizes 3.12 [18].

Quadratic discriminant analysis is a popular method that provides good results for a variety of data sets. The decent overall performance of this method is thought to not be attributed to the data sets being approximately Gaussian, but to the fact that the Gaussian models provided by the method are stable. Many classification problems can be solved by the simple decision boundaries provided by quadratic discriminant analysis, making it a popular choice [15].

3.3.5 Support vector machine

The difficulty level of a classification problem depends on many factors, an important one is how the classes are distributed in the feature space. If the classes are separable and do not overlap, the classification problem can most likely be solved with a simple decision boundary. In the case of classes being nonseparable, the optimal decision boundary is more difficult to produce with simple methods. For this case, a support vector machine (SVM) can provide good classification performance [15].

When classifying with the basic SVM classifier, a linear boundary is first defined in feature space, similarly to linear regression. A region surrounding the boundary is then specified, wherein the classes are allowed to overlap. Within the region, linear functions that separate the overlapping data points are computed. These additional linear boundaries are regarded as support vectors, and the decision boundary is adjusted to achieve optimal separation of the classes based on the support vectors. The optimization problem that is solved to obtain the support vector classifier is formulated as:

$$\min||\beta|| \text{ subject to } \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi \ \forall i, \\ \xi \geq 0, \sum \xi_i \leq C \end{cases} \tag{3.14}$$

where the β-parameters specify the hyperplane that acts as the decision boundary. The ξ-variables are called slack variables, and they describe the distance between overlapping data points and the edge of the region specified around the hyperplane. The size of the region in which the support vectors are constructed is specified by a distance metric C, referred to as the box constraint. The box constraint is a hyperparameter and needs to be tuned to achieve the optimal decision boundary since its value depends on the distribution in feature space [15]. How the box constraint C affects the optimization problem in 3.14 is more evident when it is re-expressed as:

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N} \xi_i \tag{3.15}$$

The support vector classifier is then computed by solving 3.15 for $\hat{\beta}$,$\hat{\beta}_0$ and forming

the decision function:

$$\hat{G}(x) = sign[x^T\hat{\beta} + \hat{\beta}_0] \tag{3.16}$$

[15].

The SVM classifier described until now uses linear hyperplanes as basis for its decision boundary, however there exists so called kernel functions, that alter the way the constructed hyperplanes are shaped. The simplest one is the trivial kernel described so far that constructs hyperplanes with linear functions and therefore the method is called linear SVM. There is a multitude of kernel functions that can be used to describe the underlying shape of the hyperplane, three common kernel functions are:

$$d\text{th-Degree polynomial} : K(x,x') = (1 + (x,x'))^d \tag{3.17}$$

$$\text{Radial basis function (RBF)} : K(x,x') = exp(-\gamma||x - x'||^2) \tag{3.18}$$

$$\text{Neural network (sigmoid/hyperbolical kernel)} : K(x,x') = tanh(\kappa_1(x,x') + \kappa_2) \tag{3.19}$$

The choice of kernel function is dependent on how the feature data is distributed. The radial basis function is closely related to Gaussian distributions since it is based on the exponential function [15]. The neural network kernel is based on the *tanh* function, in reality it behaves similarly to the radial basis function and does in general not provide any performance advantage over the radial basis function [19]. For the purpose of this thesis, the polynomial and radial basis function kernels will be evaluated as well as the trivial kernel.

3.3.6 Random Forest

Classical decision trees are efficient and capable of constructing complex decision boundaries in feature data [15]. While decision trees can provide good classification performance for a wide range of problems, they have notoriously high variance between predictions. In order to reduce the variance but retain good classification performance, a number of trees are fit to random bootstrap aggregated samples of the feature set. An average of the output of the fitted trees is calculated and the resulting forest of models is then used in classification. The goal of random bootstrap samples of the feature set is to reduce the correlation between decision trees without increasing variance significantly, which means that each tree is not fitted to all features but a random selection of features. The number of trees needed to grow a low-variance Random Forest model varies depending on the distribution of the classes in feature space [15].

A Random Forest model of B trees is fitted to training data by, for every tree b, drawing a bootstrap sample $Z(b)$ of size N from the training data. The B trees are

then individually grown from their bootstrap sample $Z(b)$ to the desired node size n by randomly selecting m features from the p features in the training data, picking the best split-point among the m features that performs the best classification and splitting that node into two daughter nodes. This is done recursively until the desired node size is achieved where the forest of B tree models T^B is obtained [15]. For a new data point x, classification equals to:

$$\hat{C}_{rf}^{B}(x) = majority\ vote\ \hat{C}_{b}^{B}(x)_1 \tag{3.20}$$

where $\hat{C}_b(x)$ is the class prediction of the bth Random Forest tree from T^B [15].

In general Random Forest models perform better the larger the forest is, but for every classification problem there is a point where increasing the size of the forest gives diminishing increase in prediction performance. The number of trees required to properly predict the boundary between classes varies and need to be tuned for every application. The Random Forest model is popular in literature since it provides stable models with overall good performance in many applications [15].

3.4 Evaluation methods

3.4.1 Cross-Validation

Cross-validation estimates the prediction error of a model $\hat{f}(X)$ when it is applied to an independent test sample. To estimate the error, data is randomly split into equally sized parts, referred to as folds. One fold is withheld to test $\hat{f}(X)$ which is fitted to the remaining folds. This is repeated for every fold and the test error estimate from every test is accumulated, producing the cross-validated estimate of the test error. Literature recommends five- or tenfold cross-validation to avoid issues with too small or too large folds, issues that can result in over- or underestimation of the test error [15]. In this thesis, tenfold cross-validation will be the standard measure of the cross-validated prediction error.

With the number of folds defined as $k = 1,..,K$ the cross-validated prediction error can be derived as

$$\mathrm{CV}(\hat{f}) = \frac{1}{k}\sum_{i=1}^{k}(\frac{1}{N}\sum_{i=1}^{N} L(y_i,\hat{f}^{-k(i)}(x_i))) \tag{3.21}$$

which is the average of the loss function L for all models $\hat{f}^{-k(i)}(x_i)$ that have been fitted and tested on the k folds according to the algorithm described above.

3.5 Performance measures

To evaluate the performance of each method and to compare them with each other, measures affected by the classification are extracted. The classification done by the methods are for each second and since a seizure last in 60-180 seconds it will consist of a sequence of seconds to be predicted. A decision layer is therefore added after the classification process to convert the predicted classification for each time instance to a prediction for each seizure event. Different measures are extracted depending on if the outcome of the classification is per seizure event or time instance. For the purpose of time instance classification, a true positive is an time instance correctly classified as a seizure and a true negative is a time instance correctly classified a non-seizure activity. When classifying seizure events, a true positive represents a correctly classified seizure event. True negatives are not obtained from the seizure event classification since the developed decision method does not provide symmetrical classification in the same manner as the time instance based classification.

3.5.1 Sensitivity

Sensitivity, in some cases referred to as true positive rate, is the rate that the correct positive classifications are made with. In this application, sensitivity represents the percentage of time instances that contain seizure activity that are correctly classified. Sensitivity is defined as:

$$SEN = \frac{TP}{(TP + FN)} \tag{3.22}$$

where TP represents true positives and FP denotes false negatives.

3.5.2 Specificity

Specificity, in some cases referred to as the true negative rate, gives information about the rate with which non-seizure related activity is correctly classified. Specificity is defined as:

$$SPC = \frac{TN}{(FP + TN)} \tag{3.23}$$

TN denotes true negatives and FP represents false positives for each time instance.

3.5.3 Prediction error

Cross-validation produces an estimate of the average prediction error. In this thesis, the average prediction error is used as an overall measure of how suitable a machine learning method is for the task of classifying GTC seizures. The least squared loss function L is defined by Equation 3.24 and is used in this project to estimate the error.

$$L(y_i,\hat{f}(x_i)) = (y_i - \hat{f}(x_i))^2 \tag{3.24}$$

3.5.4 True positives, false positives and false negatives

The true positives, false positives and false negatives are used as measures when the result is going to be based on the classification of seizure events and not seizure time instances. True positives are classified seizures, false positives are normal activities classified as seizures and false negatives are misclassified seizures. These measures are generated by a decision layer explained in Section 4.3 that evaluates the classification of a sequence of seconds and converts it to an event of seizure or normal activity. The decision is then checked against the true classification to get the measures.

4

Method

This chapter details how the previously described theories and concepts will serve the purpose of this thesis. This chapter also aims to describe the consecutive steps required to perform classification on patient data.

An important aspect of the literature study is to identify features that can be extracted from the measurements and used to achieve high performance in classifying seizures. New features are created based on seizure studies both from literature and visual inspections of the measurements since identifying potentially useful features is just as important as identifying the most suitable classification method. The optimal combination of features and classification method will provide the best performance.

Different evaluation methods will be implemented to investigate the performance of features and classification methods by looking at prediction error, sensitivity and specificity. The evaluation method for ML methods is 10 fold cross-validation.

4.1 Pre-processing of data

The raw measurement data from the accelerometers will be processed in order to enable accurate feature extraction and classification. The data is filtered with a bandpass filter to remove the gravitational DC component of the signal and the high frequency components that are not of interest. The filter used is a Butterworth filter with passband between 0.25 and 13 Hz. Data in higher frequencies is discarded since the movements being studied do not occur with such high frequencies [2]. During measurements the sensors have, in some cases, lost contact with the radio receiver and data for these time instances are missing when it is extracted from the sensors [12]. To ensure that no incorrect information about the recorded movements is included in testing and evaluation, the missing data segments that last past one second are identified and these periods of time are not used in further calculations, due to the fact that feature calculations are made per second. These segments could be marked with 'no decision' since no information of the classes are known in these time instances.

Each series of measurements will contain loss of data in the initialization and in the end when synchronizing the sensors due to different battery life. The measurement series are cut to equal length to ensure that each sensor contains valid measurements.

Since the data sets can be collected over several days it is desired to minimize the size of the data sets by removing non-motion data since it will not contain seizures. These time instances containing movements under 0.1 g are therefore not used in further calculations.

4.2 Classification

Two classes are used for classifying different data sets, seizure and no seizure, which are marked as 1 and 0, respectively. The classification is made by using one of the ML methods explained in Section 3.3 and train a model with a feature space extracted from a measurement together with the true classification vector of the measurement. The model is then used to predict the classification of other measurements. To get the training error, the same measurement set is used for both training and testing the model while the prediction error is derived by using different data sets.

4.3 Decision layer

The wanted outcome of the classification application is to be able to detect the seizures as distinct events but not necessarily all the seconds during a seizure. Therefore the prediction obtained as output from the classification model is processed through a decision layer. The decision layer is composed of two sublayers that together generates the true positives, false positives and false negatives in terms of number of seizures. The number of seconds that the layers use for classification are tuned by empirical studies.

Layer 1
The first layer works as a type of median filter intended to remove seizure classified seconds not surrounded by others. The filter uses a moving window of a fixed length of seconds over the classification vector. For a second surrounded by the window to be classified as a seizure, the number of seizure marked seconds inside the window needs to be at least larger than a given threshold. The length of the window and the threshold are tuned by empirical studies.

Layer 2
The second layer determines if the series of seconds classified as a seizure by the first layer is a seizure or not. If a window of a fixed length of seconds only contain seconds classified as seizures, then that time instance will be marked as the start of a seizure and the next window to check will be 120 seconds ahead since a normal

time span for a GTC seizure is approximately 1-2 minutes. The window length is tuned by empirical studies. By using the frequency peak feature the second layer is improved by adding a condition that the maximum energy content in the observed window should lay in frequencies above 2.25 Hz, since a GTC seizure generally has rhythmic movements above 2 Hz [2].

4.4 Evaluation and testing

A series of tests are performed to be able to evaluate the proposed features and ML methods. The training can be a time consuming task depending on which method is used and therefore the collected data from each patients measurements is restricted to only be the consecutive period of 24 hours containing all the seizures. These hours also contain an amount of normal activity data and the percentage of missing data in each patients measurements are between 3,9-7,6%. For tests based on data from all patients, the train and test sets contain a mix of seizures and normal activities from each patient data set. The methods that are evaluated are KNN, SVM, linear regression, logistic regression, QDA and Random Forest which are described in Section 3.3. The neighbors evaluated with KNN are: $k = \{2,3,4,5,10,15,30\}$. For Random Forest the forest sizes $B = \{5,10,15,20,25,30\}$ are evaluated. Results for Random Forest are presented for the choice of B that achieves the highest sensitivity. The box constraints used for SVM are $b = \{0.1,1,10,100,1000\}$, however it is not presented in the result section to not mislead readers into thinking that it is a parameter that is chosen based on Test 2. The box constraint is a hyperparameter and different values need to be evaluated for each application of an SVM classifier. When dividing data into training and test sets, both the time instances and the seizure events are divided according to the desired division.

4.4.1 Test 1: Feature evaluation

The features extracted from the measurements are evaluated by creating different feature spaces containing various combinations of the features. Features that are similar are tested against each other and features extracted in the spectral domain are evaluated to discern if they provide significant information in addition to the features extracted in the time domain. The different feature spaces (FS) used for evaluation are stated in Table 4.1 where FS1 is the default feature space. Each feature space is evaluated by using 75% of all patient data for training and the remaining 25% for testing and the tests are then compared by extracting the measures sensitivity, specificity and prediction error. All the methods mentioned in Section 4.4 are tested.

Table 4.1: Different feature spaces for evaluating features

Feature Spaces:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
VM	•	•		•	•	•	•	•	•	•	•	•	•	•	
SMA	•		•												
Variance	•	•	•		•	•	•	•	•	•	•	•	•	•	
Standard deviation	•			•											
Mean	•				•										
RMS	•							•							
Accumulated sum	•							•							
Entropy	•	•	•	•	•		•	•	•	•	•	•	•	•	
Correlation	•	•	•	•	•	•		•	•	•	•	•	•	•	
Freq. 0.75-2.25 Hz	•	•	•	•	•	•	•	•		•			•		•
Freq. 2.25-6 Hz	•	•	•	•	•	•	•	•		•		•		•	•
Freq. 6-9 Hz	•	•	•	•	•	•	•	•		•		•			•
Freq. 9-11.25 Hz	•	•	•	•	•	•	•	•		•			•		•
Frequency peak	•	•	•	•	•	•	•	•			•	•	•	•	•

4.4.2 Test 2: Evaluation of ML methods and dimensionality reduction techniques

This test serves two purposes, obtaining statistical measures on how suitable each ML method is for classification of GTC seizures and investigating the effect on classification performance when dimensionality reduction techniques are applied to feature spaces. For each method the feature space resulting in the highest sensitivity from Section 4.4.1 is used. Testing is performed by 10 fold cross-validation for each method and corresponding feature space with measurement A from Patient 7. Measures obtained from this test are the mean and standard deviation of the prediction error, sensitivity and specificity across the 10 folds. These measures are extracted for each method since they provide a basis with which to compare the different methods in terms of best classification performance, signified by the mean and in terms of stability for different data sets, signified by the standard deviation.

In order to evaluate potential increase of classification performance with dimensionality reduction techniques, each methods feature space is gradually reduced in terms of components with both PCA and PLS. The number of components used for each test is determined by how much of the total variance in the feature space data they describe. The lowest number of components used for evaluation is set when the contained variance is below 5%. The different amounts of components analyzed for each method is determined by visualising how the variance content varies and empirically selecting components that seem to provide significant increase in variance. Following cross-validation the training error for each method is calculated on data from Patient 7 with measurement A in order to make sure that the cross-validation results correctly describe each methods performance. If a method scores well in the cross-validation but receives a high training error it may not be suitable for

classification of GTC seizures.

4.4.3 Test 3: Decision layer parameters

To decide the length of the windows inside the decision layers tests are performed by using measurement A for Patient 7 with cross-validation. The cross-validation uses four folds since a common division of the data is 75% for training and 25% for testing [15]. The lengths tested are independently varying in the range of 1-25 seconds and compared with the results generated without a decision layer. The methods used for constructing the decision layer thresholds are determined by the results from 4.4.2. They will be chosen based on their performance in sensitivity and specificity, which should be poor, with a restriction that the sensitivity needs to be at least 50% to be sure that there exists enough correctly classified seconds to have a possibility of classifying seizure events using the windows in the decision layers. Methods that perform poorly compared to others are used for parameter evaluations to obtain a robust decision layer, based on the idea that if the decision layer performs well for methods with poor performance, it will function for methods that perform better. The results are measured in true positives, false negatives and false positives.

4.4.4 Test 4: Testing the classification performance of ML methods

This test is meant to test how well the different methods and eventual reduction techniques perform when classifying seizure events. For this test the model parameters that were identified in 4.4.2 will be used together with decision layer parameters from 4.4.3. Classification performance of the methods in this test will be judged on how accurate they are when classifying seizure events. In test 1 and 2 the performance measures are related to how many individual seconds that are correctly classified, but for the concluding evaluation done by this test, the ML methods will be compared to how accurate they are in classifying entire seizures.

This test is performed as follows:

1. Pre-process train and test data
2. Get classification of both training and test data
3. Feature extraction of the training and test data
4. (opt.) Feature space dimension reduction of both data sets
5. Train desired classification model
6. Apply model to test data and obtain predicted classification

7. Decision layer

8. Extract performance measures

The performance measures that are extracted and used for comparison between methods are the prediction error, sensitivity and specificity after the median filter and number of correctly classified seizures. In order to gain as much information as possible about the methods performance, different feature sets are used for training and testing. The methods are first and foremost trained and tested on separate data from the same patient, so as to evaluate how well they perform on each patient. Additional tests are performed with data from different patients and serves to gain information on how well different methods generalise for feature sets with mixed patients. Data from measurement A from Patient 7 is only used as training data in this stage of testing since it was used to identify model parameters. Since the data from measurement A has influenced the choice of model parameters, if it is used as test data the results will be skewed and not correctly reflect how well a method would perform on unseen data. The combinations of training and test sets are shown in Table 4.2.

Table 4.2: The different combinations of training and test sets used in this stage of testing

Data comb.:	**1**	**2**	**3**	**4**
Train set:	75% pat.7B	100% pat.7A	100% pat.7B	100% pat.48
Test set:	25% pat.7B	100% pat.7B	100% pat.48 & 55	100% pat.22

Since the data from Patient 48 and 55 only contain two seizures each, these are not considered large enough to perform any statistical analysis of their performance if they were used individually for training and testing. Therefore they are used in conjunction with other sets of data when evaluating how well the methods generalise between different patients. The data from Patient 55 is sampled at 102,4 Hz as opposed to the 50 Hz sample rate of all other data. The higher sampling frequency results in higher energy estimates than for data sampled in 50 Hz when using the spectrogram function in **Matlab**. The difference in energy estimates between data sets is adjusted by rescaling the frequency domain feature data from Patient 55 to match other frequency feature sets. The scaling factor is based on the ratio between sampling frequencies squared and calculates to: $(\frac{102{,}4\text{Hz}}{50\text{Hz}})^2 = 4{,}19$. Seeing as Patient 7 has two measurements performed at different occasions, data combination 2 in Table 4.2 is comparable to data combination 1 since they both concern training and testing on the same patient. The test with data combination 1 is evaluated with fourfold cross-validation with decision layer applied on the predicted classification for every test fold.

4.4.4.1 Testing the robustness in models against seizure like activities

An important part of evaluating how suitable different ML methods are for classifying GTC seizures is to investigate how robust they are against activities that involve seizure like movements. When testing the robustness of the methods, data sets containing high frequency activities collected during a game of badminton, toothbrushing and dishwashing are added to the data used for testing. The test will provide information on how robust the methods are against activities that are not seizures but contain movements that are similar to the movements displayed by patients having GTC seizures.

4.4.5 Test 5: Evaluating gyroscope measurements

Patient 55 contains measurements from two three-axis gyroscopes, located in the same units as the accelerometers, in addition to the acceleration. The test is requested by Acreo Swedish ICT AB and will not affect the choice of classification methods but can be an important step for future work. The test is performed by training and testing on the same data to generate a training error. The gyroscope measurements are rescaled to have the same magnitude as the accelerometer data. The same feature extraction used for accelerometer data is applied on the measurements generated by the gyroscope where the features are not tailored to fit velocity. The feature spaces used are the ones produced by test 1 for each method. Training errors are generated for measurements collected by the gyroscope and the gyroscope in combination with the accelerometer data to be compared with the results generated with only accelerometer data. This is done to estimate if the gyroscope will give any additional information to the accelerometer measurements.

5

Results

The results generated by each test are presented in this chapter.

5.1 Test 1: Feature evaluation

The results in this section are based on the test described in 4.4.1 with the feature spaces stated in Table 4.1 and the entire result can be seen in Appendix B.1. The features that increased the performance of the classification method where evaluated together and the feature space that performed well for each method can be seen in Table 5.1 which are the ones used in further testing. The measures for these feature spaces can be seen in Table 5.1. Features that did not increase the performance of the majority of the methods were correlation, variance, frequency peak, RMS and acculumated sum. The features that did increase the performance were VM, SMA, standard deviation, entropy and the frequency bands where the smallest and the largest bands worked better together than the bands in between.

Table 5.1: The feature spaces generated for each method

	VM	SMA	Variance	Standard deviation	Mean	RMS	Accumulated sum	Entropy	Correlation	Freq. 0.75-2.25 Hz	Freq. 2.25-6 Hz	Freq. 6-9 Hz	Freq. 9-11.25 Hz	Frequency peak
Linear regression	●			●				●		●			●	●
Logistic regression	●			●				●		●			●	●
QDA	●				●		●	●			●			
KNN $k = 2$	●		●		●			●		●			●	
KNN $k = 3$	●			●				●		●			●	
KNN $k = 4$		●		●						●			●	
KNN $k = 5$		●		●						●			●	
KNN $k = 10$		●		●	●			●		●	●	●	●	
KNN $k = 15$		●		●	●			●		●	●	●	●	
KNN $k = 30$		●		●				●		●	●	●	●	
SVM linear	●			●				●		●			●	●
SVM RBF	●			●		●	●	●	●					
SVM poly 2		●		●	●			●			●			●
SVM poly 3	●		●					●			●			
Random Forest n = 30	●		●					●		●			●	

Table 5.2: Measures for each method with the feature spaces in Table 5.1

	Sensitivity	Specificity	Error
Linear regression	0,7238	0,9993	0,0096
Logistic regression	0,8122	0,9965	0,0095
QDA	0,9779	0,9184	0,0796
KNN $k = 2$	0,9503	0,9828	0,0182
KNN $k = 3$	0,9171	0,9919	0,0105
KNN $k = 4$	0,9171	0,9904	0,0120
KNN $k = 5$	0,9171	0,9934	0,0091
KNN $k = 10$	0,8950	0,9954	0,0079
KNN $k = 15$	0,8785	0,9963	0,0075
KNN $k = 30$	0,8674	0,9970	0,0071
SVM linear	0,8508	0,9928	0,0118
SVM RBF	0,5028	0,9922	0,0236
SVM poly 2	0,9061	0,9889	0,0138
SVM poly 3	0,9392	0,9823	0,0191
Random Forest n = 30	0,9171	0,9926	0,0098

5.2 Test 2: Evaluation of ML methods and dimensionality reduction techniques

The results presented in this section are based on the test described in Section 4.4.2. The mean and standard deviation values shown in figures in this section are based on the numerical results listed in Appendix C.

5.2.1 Linear regression

From Figure 5.1 and Figure 5.2 the number of components used in cross-validation for evaluation of linear regression with PCA and PLS are selected.

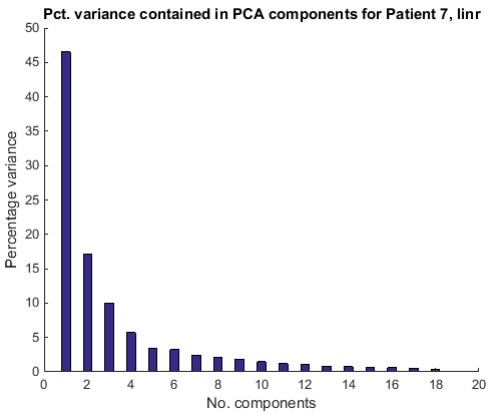

Figure 5.1: PCA components

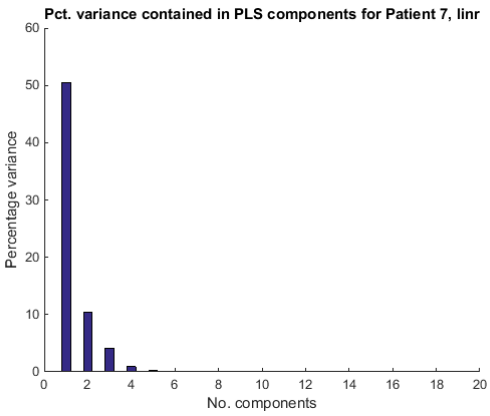

Figure 5.2: PLS components

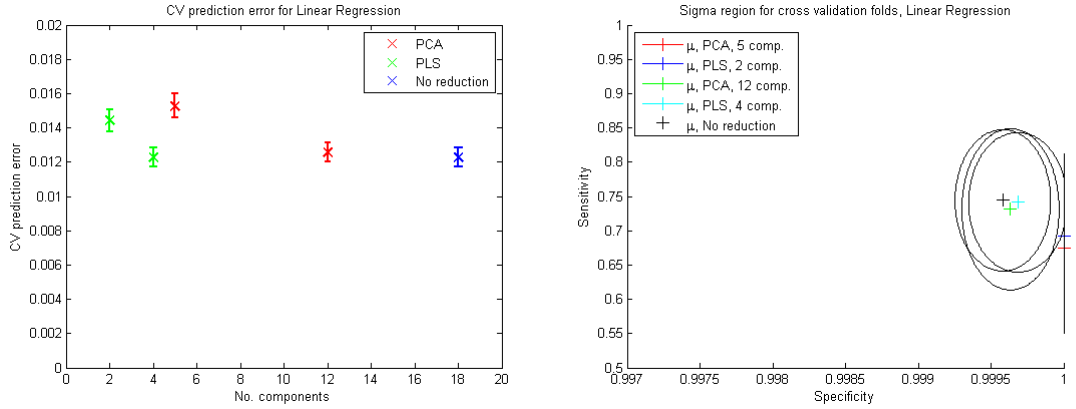

Figure 5.3: Prediction error, sensitivity and specificity for 10 fold cross-validation of linear regression. Prediction error is presented with error bars. Ellipsoids represent the standard deviation, σ, of sensitivity and specificity centered at the mean, μ. These are referred to as sigma regions.

When comparing the prediction error of the unreduced feature space with the different PCA and PLS components, it is apparent that none of the reductions offer a decrease in prediction error. Sensitivity decreases with reduction but specificity increases for some component choices. For 5 PCA- and 2 PLS components the classification becomes unstable.

5.2.2 Logistic regression

From Figure 5.4 and Figure 5.5 the number of components used in cross-validation for evaluation of logistic regression with PCA and PLS are selected.

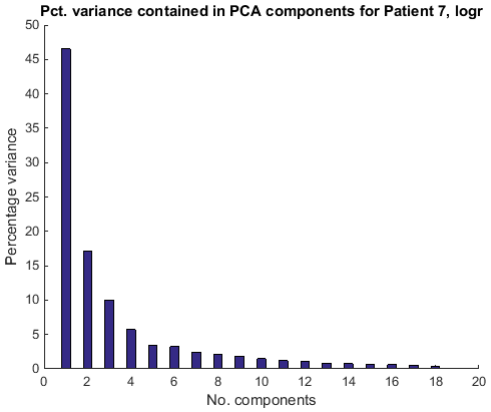

Figure 5.4: PCA components

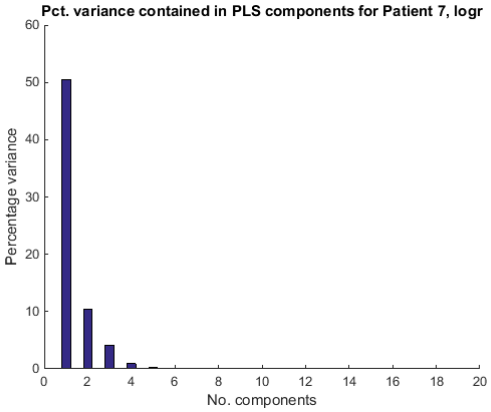

Figure 5.5: PLS components

The prediction error of logistic regression behaves similar to linear regression with

reduced feature spaces. The exception is for PCA with 12 components, where a slight decrease in error from 0,0107 to 0,0103 can be noted. Using 12 PCA components also increases both sensitivity and specificity with an increase in mean from 0.8157 to 0,8162 and 0,9982 to 0,9987 respectively. Complete results can be found in Table C.2. From Figure 5.7 and Figure 5.8 the number of components used in cross-validation for evaluation of QDA with PCA and PLS are selected.

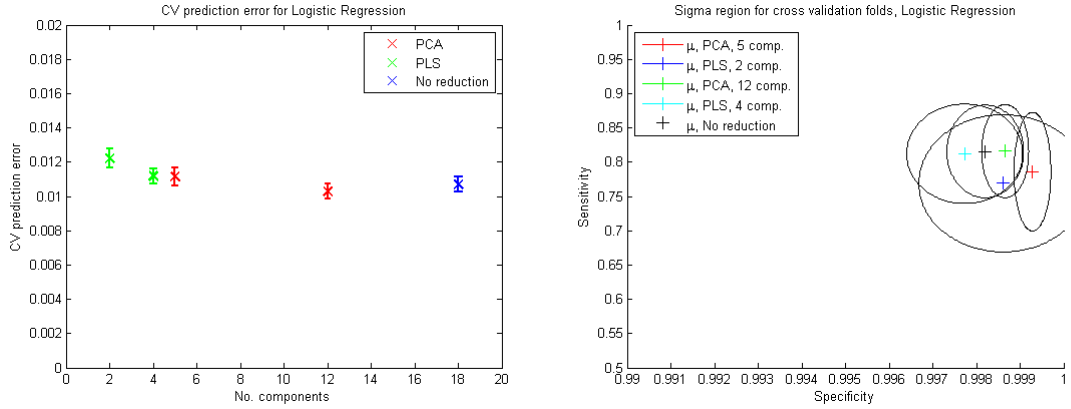

Figure 5.6: Prediction error, sensitivity and specificity for 10 fold cross-validation of logistic regression. Prediction error is presented with error bars. Ellipsoids represent the standard deviation, σ, of sensitivity and specificity centered at the mean, μ. These are referred to as sigma regions.

5.2.3 Quadratic discriminant analysis

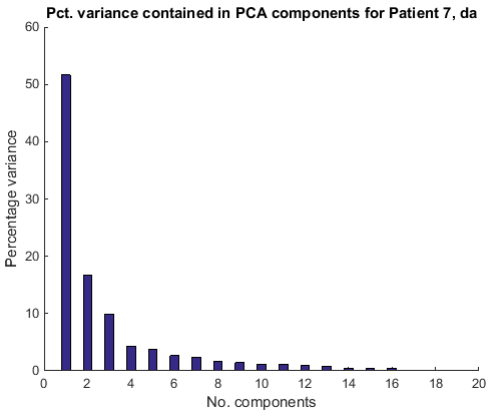

Figure 5.7: PCA components

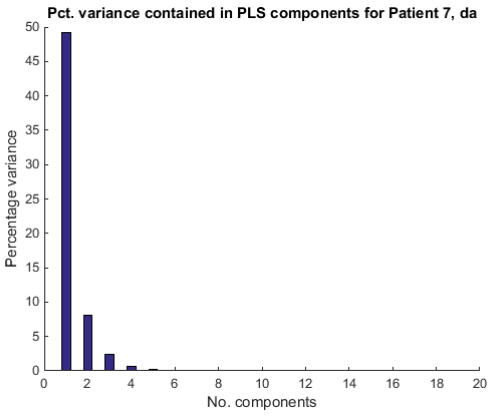

Figure 5.8: PLS components

Fig.5.9 shows that reducing the feature space with PCA decreases the prediction error when using 10 components. PLS reduction has no positive effect on the prediction error for QDA since it increases with reduction. Sensitivity decreased for all reductions. Using 10 PCA components increases the specificity however.

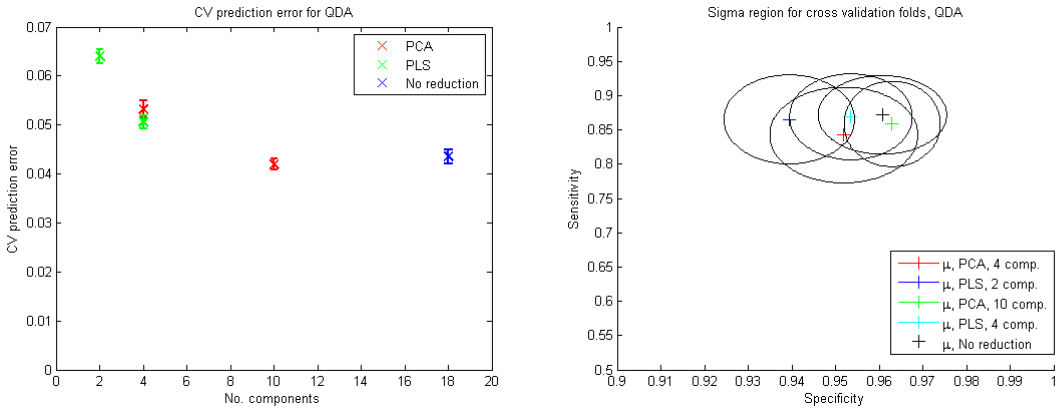

Figure 5.9: Prediction error, sensitivity and specificity for 10 fold cross-validation of QDA. Prediction error is presented with error bars. Ellipsoids represent the standard deviation, σ, of sensitivity and specificity centered at the mean, μ. These are referred to as sigma regions.

5.2.4 KNN

From Figure 5.7 and Figure 5.8 the number of components used in cross-validation for evaluation of KNN with PCA and PLS are selected. The variance in the feature sets for each k is similar for both PCA and PLS, therefore the same number of components are used for each choice of k. Since the KNN classifier produces a different model with each selection of amount of neighbors, each component selection is evaluated for all desired neighbors.

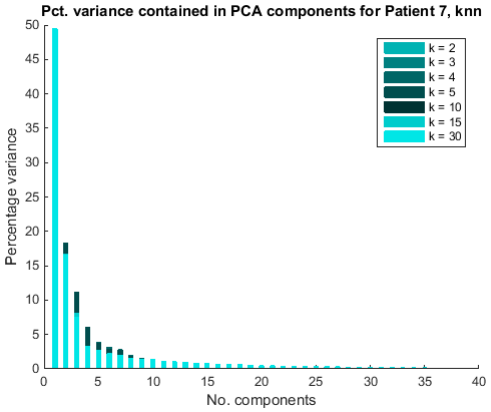

Figure 5.10: PCA components

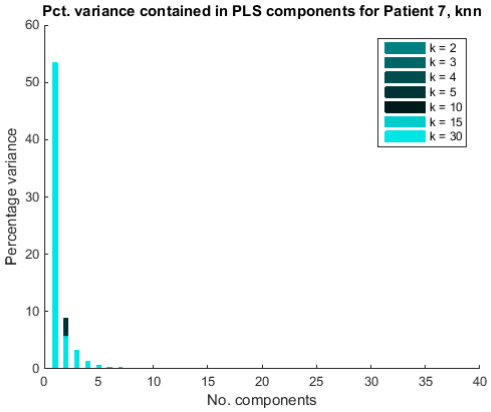

Figure 5.11: PLS components

The prediction error shown in Figure 5.12 can be decreased with both PCA and PLS. There is however no increase in sensitivity for any of the tested reductions. For 2 PLS components there is an increase in the standard deviation of both sensitivity and specificity shown in Figure 5.14, which suggests that the method becomes

unstable for low numbers of PLS components. Reducing the feature space to 10 PCA components does not impact performance severely, which can be observed in Figure 5.13, making this a possible reduction to use if minimizing prediction error would be a priority.

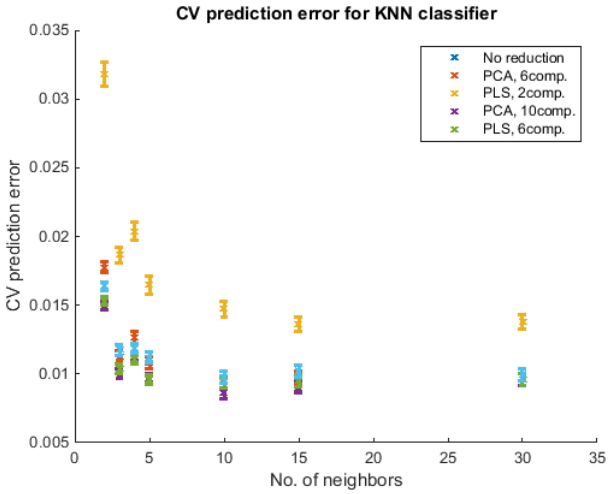

Figure 5.12: Prediction error for unreduced feature space, and different component choices for PCA and PLS

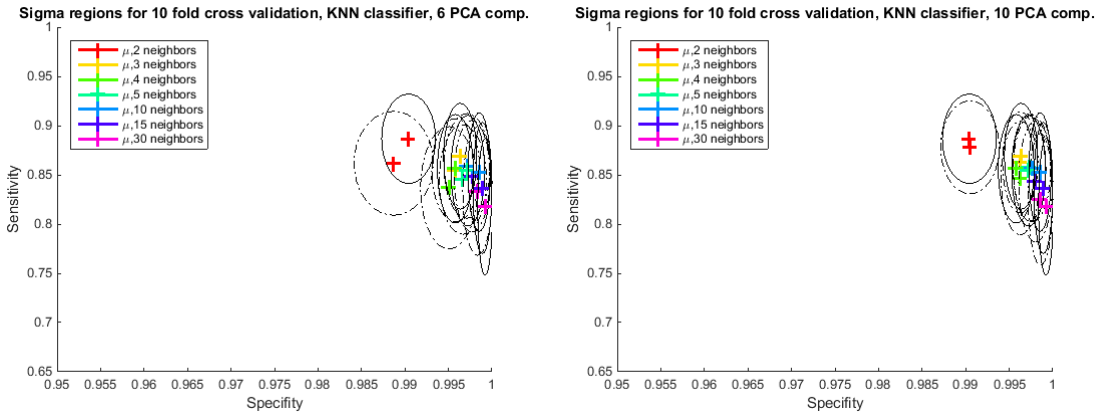

Figure 5.13: PCA reduction of KNN feature spaces. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

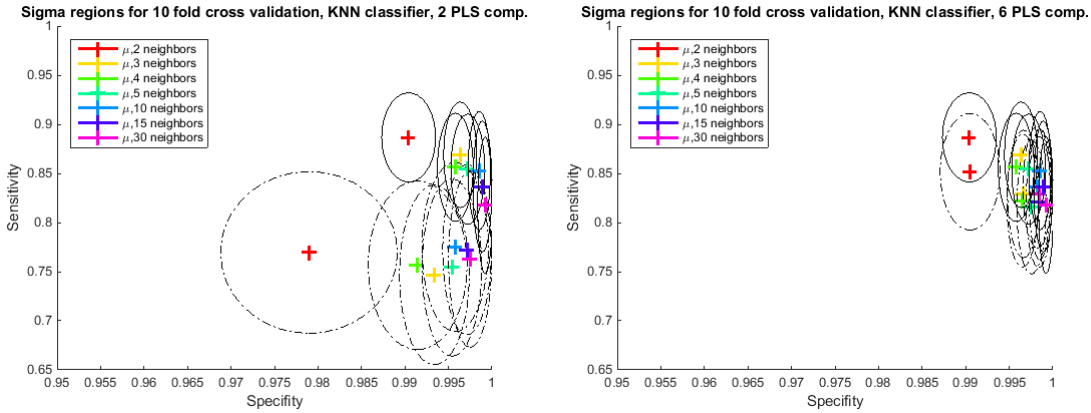

Figure 5.14: PLS reduction of KNN feature spaces. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.5 SVM

The results from support vector machine with the different kernels evaluated are presented in this section.

5.2.5.1 Linear SVM, trivial kernel

From Figure 5.15 the number of PCA and PLS components are determined. For some box constraints the prediction error in Figure 5.16 can be slightly decreased when reducing the feature space to 6 PLS components. This corresponds with Figure 5.15 where the variance in the feature set seems to be completely described by 6 PLS components. Comparing the results in Figure 5.17 to Figure 5.18 shows that PLS reduction of the feature space can increase sensitivity but it comes at a cost of decreased specificity.

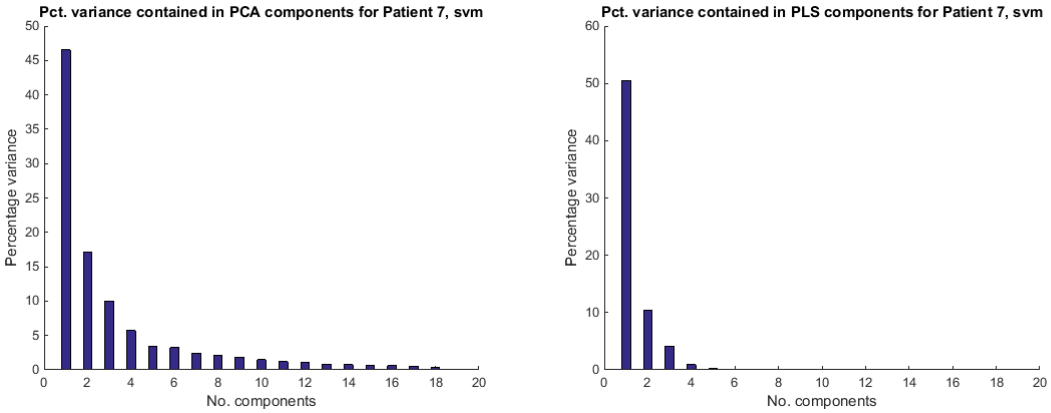

Figure 5.15: Variance in PCA and PLS components for the feature space used with linear SVM, trivial kernel.

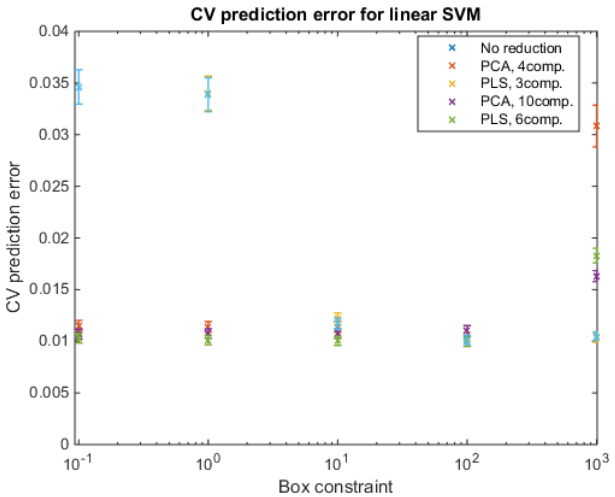

Figure 5.16: Prediction error for unreduced feature space, and different component choices for PCA and PLS

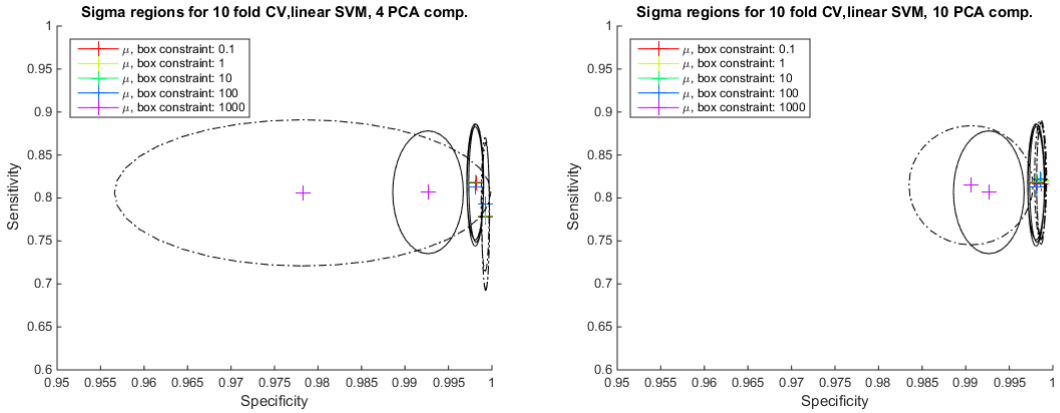

Figure 5.17: PCA reduction of linear SVM, trivial kernel feature space. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

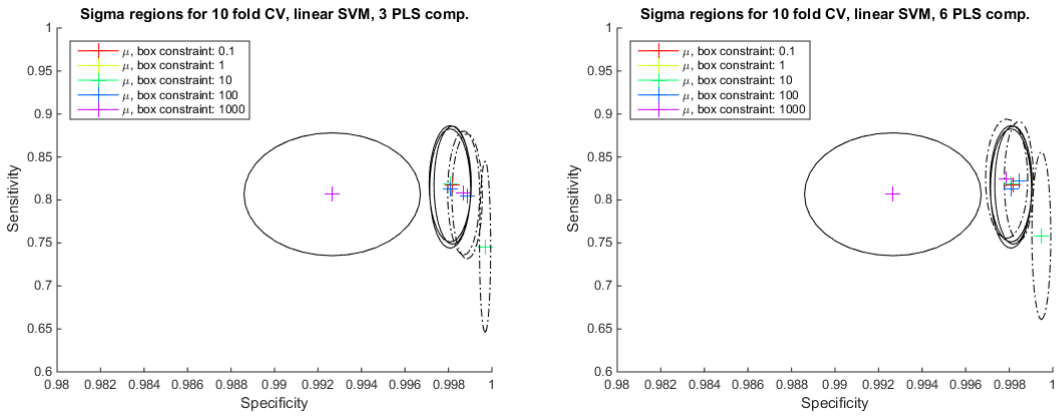

Figure 5.18: PLS reduction of SVM linear kernel feature space. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.5.2 Polynomial kernel, order 2

From Figure 5.19 the number of PCA and PLS components to evaluate the reduction methods for are chosen. The prediction error for the component selections is shown in Figure 5.20. There is no decrease in error noted for any reduction and the prediction error increases notably with 4 PCA components. PCA reductions shown in Figure 5.21 makes the method perform worse both in terms of the mean and standard deviation of sensitivity and specificity. Reduction by PLS shown in Figure 5.22 seems to offer more stable performance than PCA since the standard deviation is not increased as much as in Figure 5.21. Although more stable than PCA, the PLS components do not offer any increase in sensitivity.

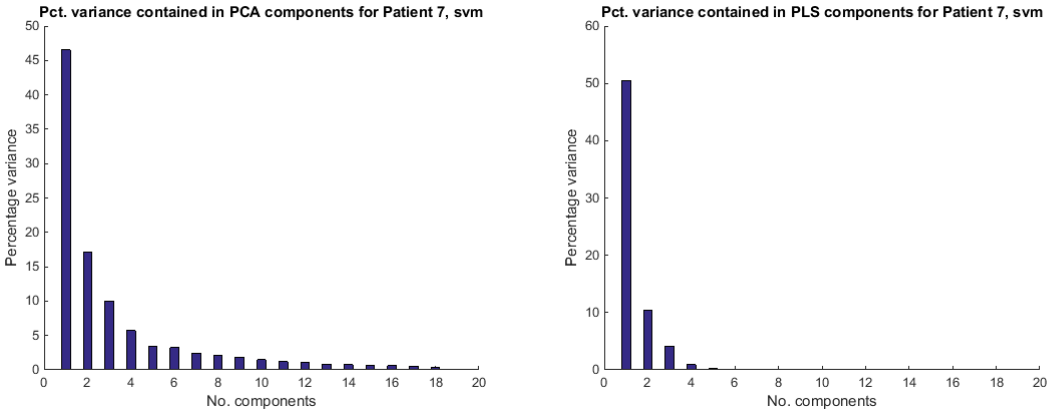

Figure 5.19: Variance in PCA and PLS components for the feature space used with SVM poly. order 2 kernel.

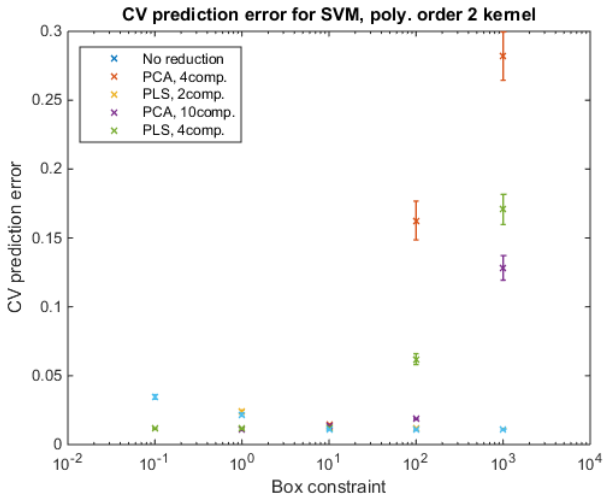

Figure 5.20: Prediction error for unreduced feature space, and different component choices for PCA and PLS

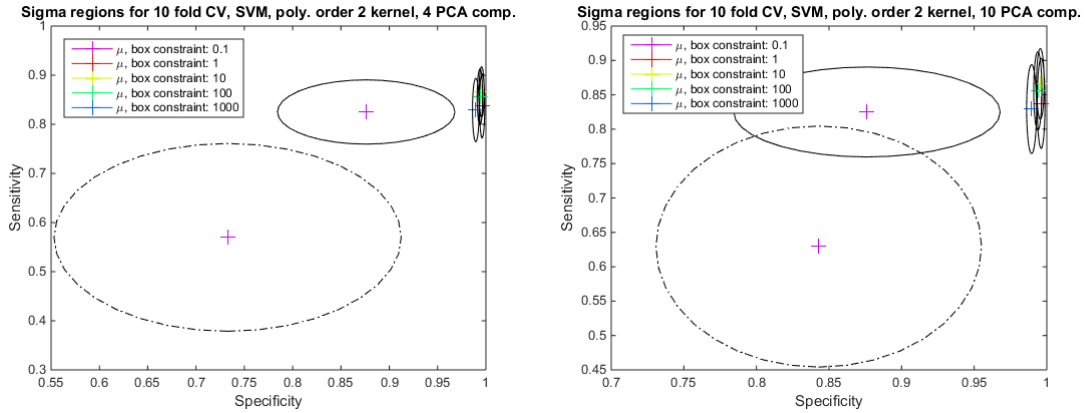

Figure 5.21: PCA reduction of feature space for SVM with second order polynomial kernel. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

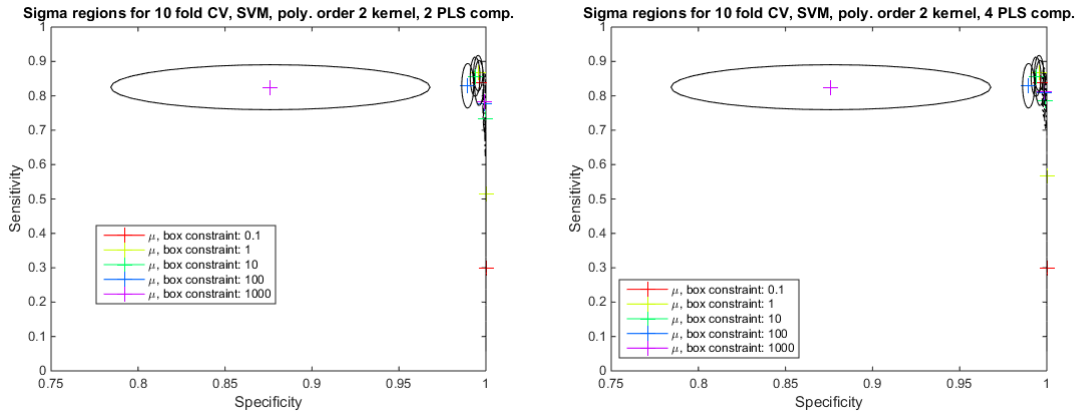

Figure 5.22: PLS reduction of feature space for SVM with second order polynomial kernel. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.5.3 Polynomial kernel, order 3

From Figure 5.23 the number of PCA and PLS components to evaluate the reduction methods for are chosen. The performance of the polynomial SVM kernel of order 3 behaves similarly to the polynomial kernel of order 2 when reduced. There is no decrease compared to the unreduced feature space in Figure 5.24. Sensitivity is decreased for both reduction techniques as displayed in Figures 5.25-5.26. There is however an increase in specificity to be gained when using PLS together with this kernel.

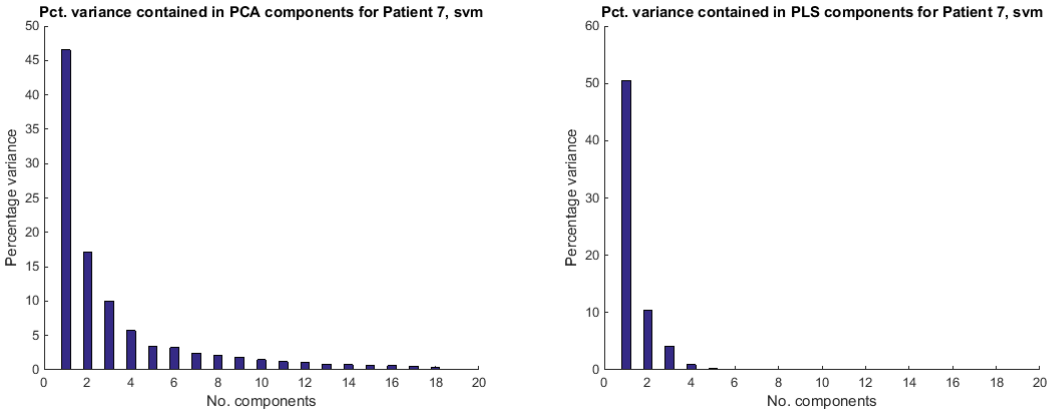

Figure 5.23: Variance in PCA and PLS components for the feature space used with SVM polynomial of order 3 kernel.

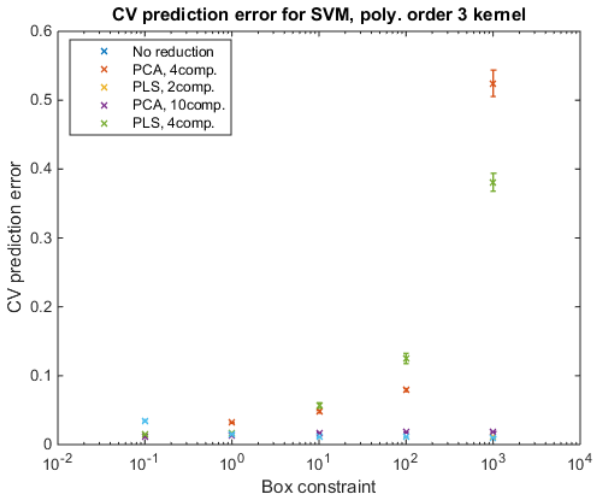

Figure 5.24: Prediction error for unreduced feature space, and different component choices for PCA and PLS

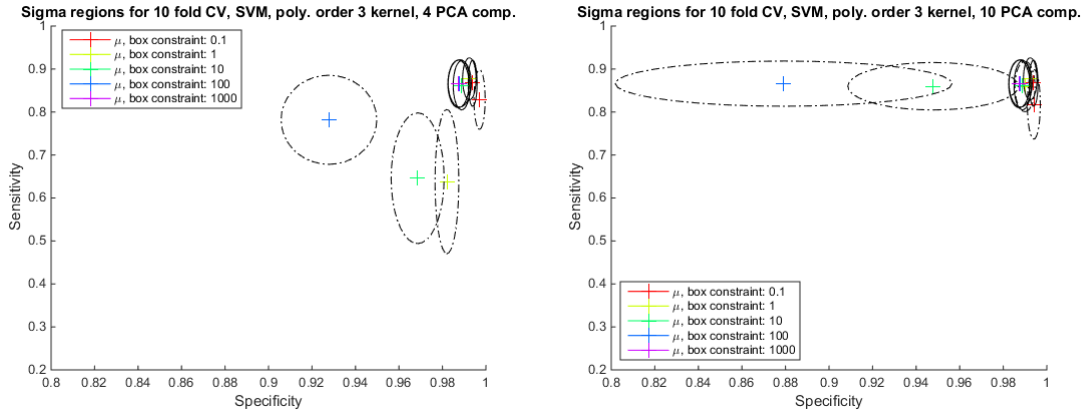

Figure 5.25: PCA reduction of feature space for SVM with third order polynomial kernel. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

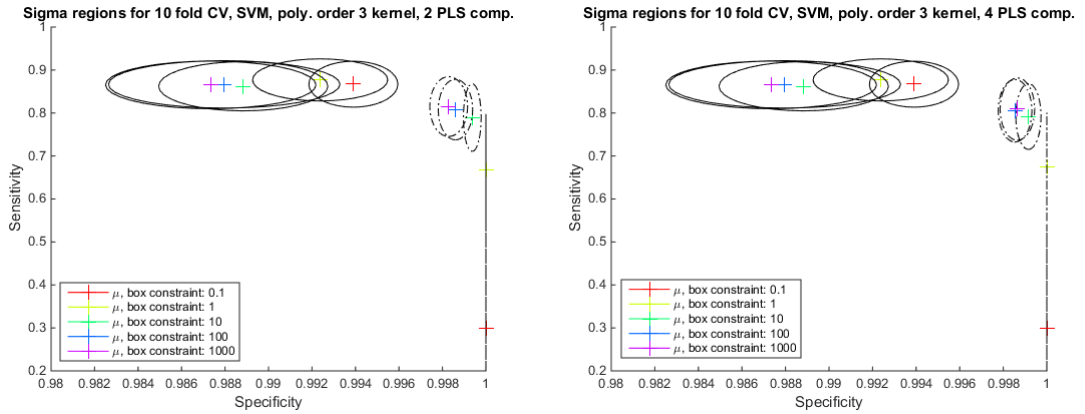

Figure 5.26: PLS reduction of feature space for SVM with third order polynomial kernel. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.5.4 Radial basis function kernel

When using the RBF kernel with SVM, the prediction error can be decreased with 10 PCA components, which can be seen in Figure 5.28. When taking sensitivity and specificity shown in Figure 5.29 into account this PCA reduction causes a decrease in sensitivity for all box constraints that are tested. The PLS reductions in Figure 5.30 increases specificity but lowers sensitivity.

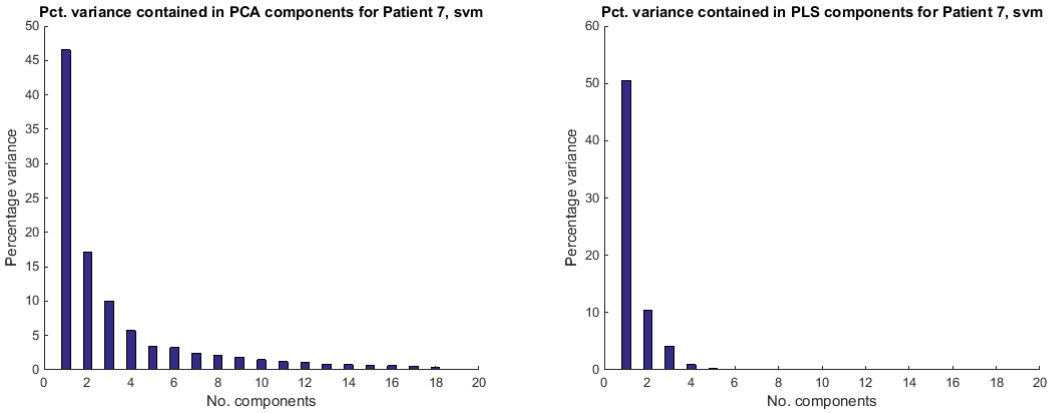

Figure 5.27: Variance in PCA and PLS components for the feature space used with SVM, radial basis function kernel.

From Figure 5.23 the PCA and PLS components are selected.

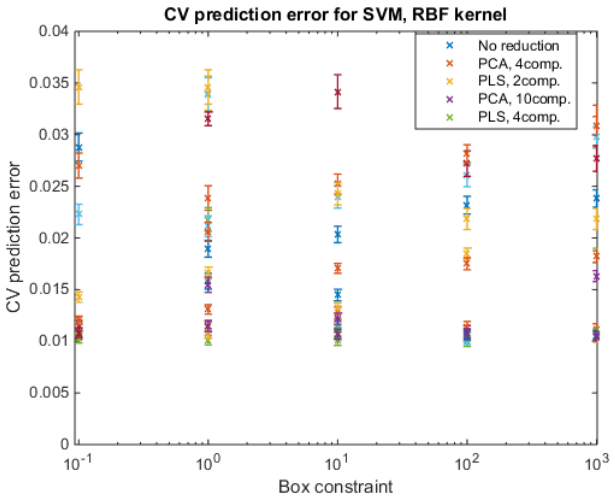

Figure 5.28: Prediction error for unreduced feature space, and different component choices for PCA and PLS

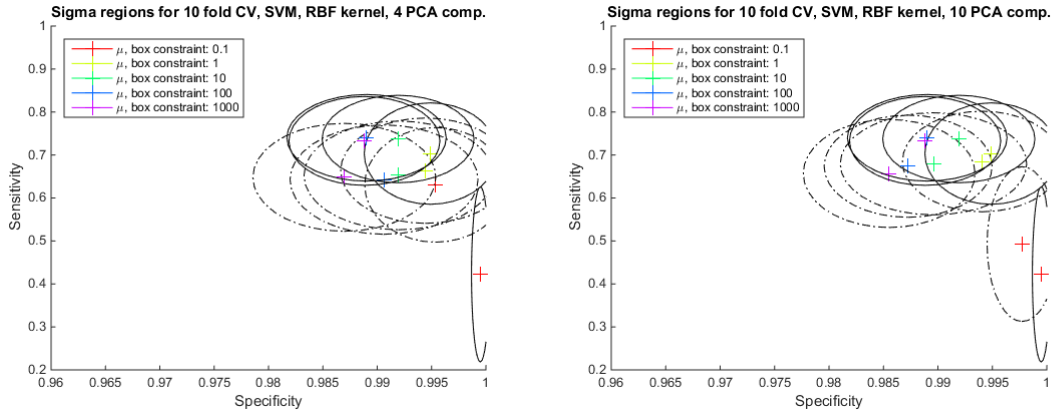

Figure 5.29: PCA reduction of SVM, RBF kernel feature space. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

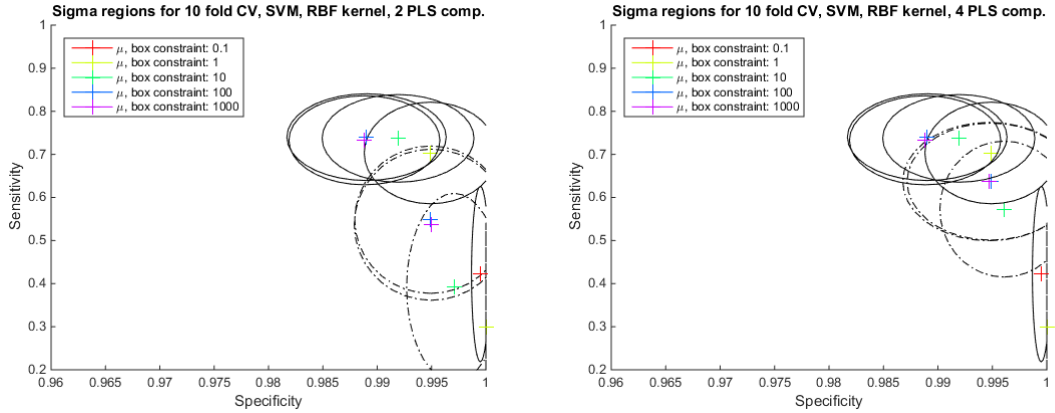

Figure 5.30: PLS reduction of SVM, RBF kernel feature space. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.6 Random Forest

The prediction error for cross-validation of Random Forest in Figure 5.32 is decreased when using 10 PCA components or 4 PLS components. There is no increase of sensitivity provided by any of the component choices shown in Figures 5.33-5.34.

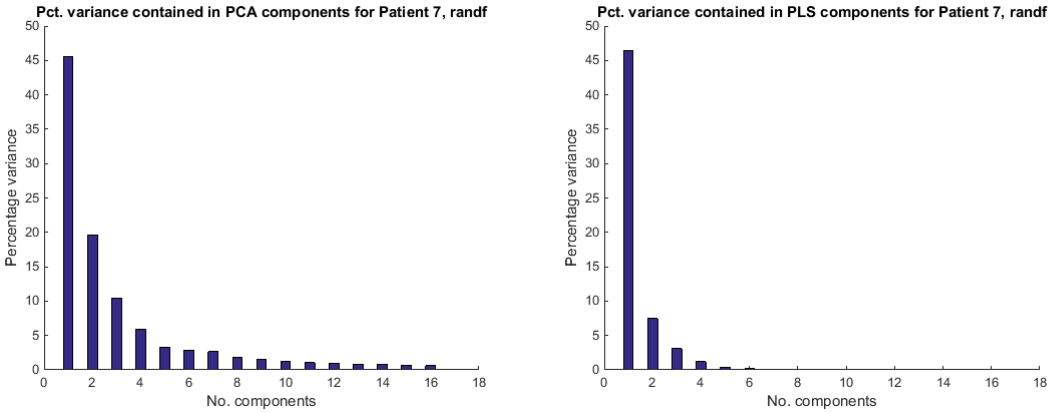

Figure 5.31: Variance in PCA and PLS components for the feature space used with Random Forest.

The PCA and PLS components used for evaluation are selected from Fig.5.31.

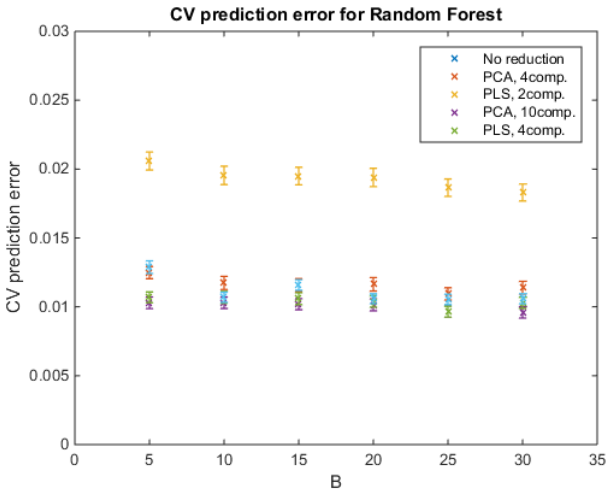

Figure 5.32: Prediction error for unreduced feature space, and different component choices for PCA and PLS

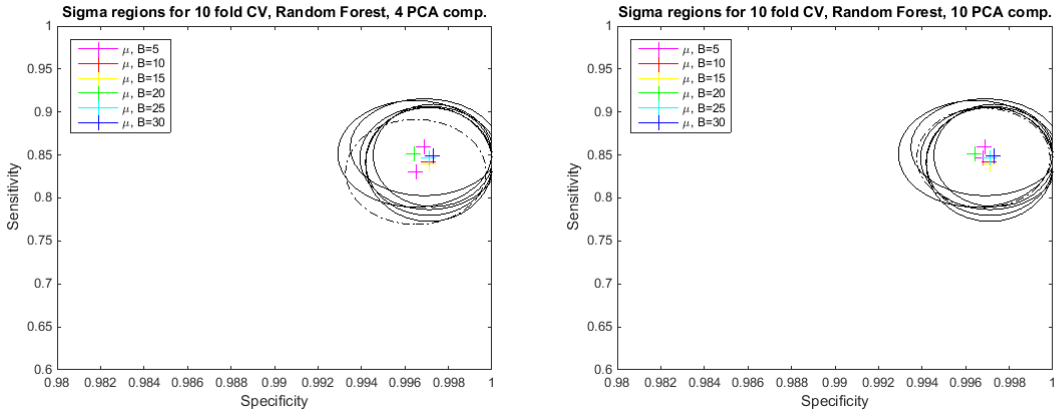

Figure 5.33: PCA reduction of feature space used with Random Forest. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

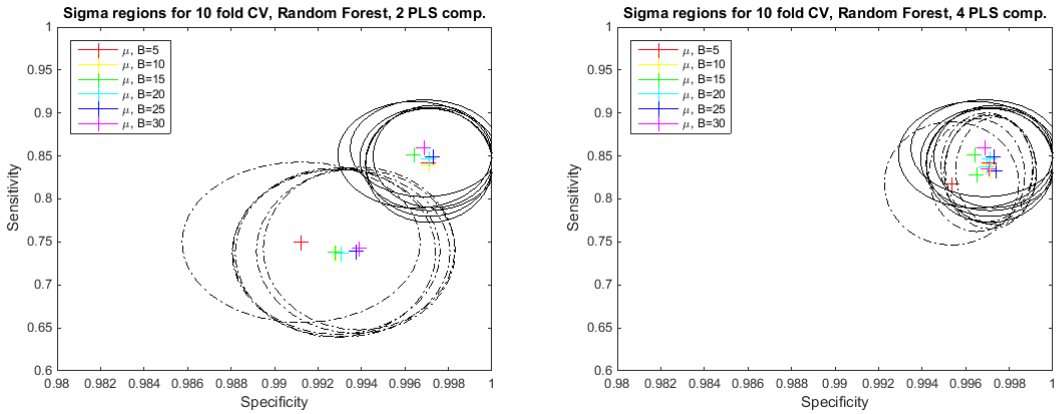

Figure 5.34: PLS reduction of feature space used with Random Forest. Ellipsoids represent the mean, μ, and standard deviation, σ, of sensitivity and specificity. Solid lines are unreduced feature space, dashed lines represent reduced feature space.

5.2.7 Summary of test 2

Cross-validation results in Figure 5.3 show that for some methods there is an increase in specificity for certain reductions of the feature space. However there are no reductions that offer significant increases in sensitivity, which is the prioritized measure. This suggests that the manual feature selection in 4.4.1 serves its purpose and the features chosen contain the information needed to make each method perform well. Due to time constraints an exhaustive investigation of all component choices is not performed so there is a possibility that there may exist component selections that could offer an increase in classification performance. For further testing no feature space reduction will be made and the results in Table 5.3 will be

used to compare how suitable the methods are for this classification problem. The training error in Table 5.4 serves as confirmation of the cross-validation results. A high training error suggests that a method is less suited for classification of GTC seizures since it then is unable to correctly classify all seizure instances even under optimal circumstances.

Table 5.3: Summary of performance measures from the cross-validation of the ML methods. All measures in this table are calculated with feature spaces selected from 4.4.1.

Method:	**Lin.reg.**	**Log.reg.**	**QDA**	**KNN,** $k = 2$	**KNN,**$k = 3$
Pred. err., μ:	0,0123	0,0107	0,0435	0,0150	0,0100
Sensitivity, μ:	0,7443	0,8157	0,8718	0,8867	0,8689
Specificity, μ:	0,9996	0,9982	0,9607	0,9904	0,9964
Pred. err., σ:	0,0011	0,0009	0,0028	0,0006	0,0007
Sensitivity, σ:	0,2059	0,1351	0,1145	0,0908	0,1077
Specificity, σ:	0,0007	0,0018	0,0295	0,0062	0,0038

Method:	**KNN,** $k = 4$	**KNN,**$k = 5$	**KNN,**$k = 10$	**KNN,**$k = 15$
Pred. err., μ:	0,0112	0,0096	0,0086	0,0090
Sensitivity, μ:	0,8562	0,8542	0,8529	0,8367
Specificity, μ:	0,9958	0,9972	0,9985	0,9989
Pred. err., σ:	0,0007	0,0007	0,0008	0,0008
Sensitivity, σ:	0,1104	0,1123	0,1206	0,1334
Specificity, σ:	0,0041	0,0039	0,0026	0,0022

Method:	**KNN,** $k = 30$	**SVM,linear**	**SVM,poly.order 2**
Pred. err., μ:	0,0096	0,0107	0.0107
Sensitivity, μ:	0,8175	0,8188	0.8659
Specificity, μ:	0,9992	0,9981	0.9955
Pred. err., σ:	0,0009	0,0009	0.0009
Sensitivity, σ:	0,1390	0,1353	0.1029
Specificity, σ:	0,0017	0,0019	0.0053

Method:	**SVM,poly.order 3**	**SVM,RBF**	**RF,** $B = 30$
Pred. err., μ:	0.0131	0,0232	0,0096
Sensitivity, μ:	0.8770	0,7400	0,8590
Specificity, μ:	0.9924	0,9890	0,9969
Pred. err., σ:	0.0009	0,0017	0,0008
Sensitivity, σ:	0.0979	0,2013	0,1129
Specificity, σ:	0.0062	0,0147	0,0068

Table 5.4: Highest training error recorded for each method

Method:	**Lin.reg.**	**Log.reg.**	**QDA**	**KNN, k=30**
Training error:	0,011	0,0095	0,0401	0,0083

Method:	**SVM (RBF)**	**R.F, B=30**
Training error:	0,0339	0,0002

5.3 Test 3: Decision layer parameters

To decide if a sequence of seconds should be classified as a seizure a decision layer is made with parameters tuned by an empirical study applied on two methods, SVM with RBF kernel and QDA. These methods gave results with low sensitivity and specificity, respectively, which can be seen in Figures 5.29 and 5.9. In appendix the Tables D.1 and D.2 shows the mean sensitivity, specificity and prediction error for cross-validation after they have been processed through a median filter together with the number of seizures classified and misclassified by the decision layer. Four tests showed good results where all seizures where found for both methods which can be seen in Table 5.5. Test 3.1 had the lowest prediction error and detected all seizures for both methods. The chosen parameter set for the decision layer then is the median filter with window length 25 seconds, median filter threshold of 10 seconds and minimum decision threshold of 20 seconds. To reduce the false positives the frequency peak condition is added to the decision layer for Test 3.1 and both methods find all seizures. A reduction in false positives together with one misclassified non-seizure event is observed for QDA.

Table 5.5: The tests with the decision layer parameters that gave the best results for SVM with RBF and QDA. The window lengths are presented in the order: median filter window length, median filter threshold and minimum decision threshold.

SVM RBF												
Test:	3.1			3.2			3.3			3.4		
Thresholds [s]:	10	5	15	10	4	15	25	10	20	15	5	25
Sensitivity:	0,6500			0,7171			0,7448			0,7865		
Specificity:	0,9982			0,9961			0,9989			0,9959		
Pred. error:	0,0114			0,0100			0,0054			0,0068		
TP:	11			11			11			11		
FP:	0			0			0			0		
FN:	0			0			0			0		

QDA												
Test:	3.1			3.2			3.3			3.4		
Thresholds [s]:	10	5	15	10	4	15	25	10	20	15	5	25
Sensitivity:	0,8446			0,8710			0,9017			0,9017		
Specificity:	0,9656			0,9506			0,9727			0,9473		
Pred. error:	0,0084			0,0147			0,0044			0,0088		
TP:	11			11			11			11		
FP:	12			14			12			7		
FN:	0			0			0			0		

5.4 Test 4: Testing the classification performance of methods with parameters identified in Test 2

In this section the results from tests performed used for evaluation of patient specific use of methods, generalisation of methods and robustness against high-frequent motions are showed.

5.4.1 Patient specific test results

The results from evaluating the methods with the data combination 1 in Table 4.2, training and testing on Patient 7 with measurement B with fourfold cross-validation, can be seen in Table 5.6. All methods find all 12 seizures for measurement B except SVM with RBF kernel that misclassified one of them. QDA misclassified four non-

seizures but finds the 12 true seizures.

Table 5.6: The results from four-folds cross-validation on Patient 7 with measurement B with the measures mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,7342	0,9993	0,0144	12	0	0
Logistic regression	0,8399	0,9991	0,0084	12	0	0
QDA	0,8954	0,9355	0,0656	12	4	0
KNN $k = 2$	0,8724	0,9989	0,0070	12	0	0
KNN $k = 3$	0,8189	0,9990	0,0096	12	0	0
KNN $k = 4$	0,8635	0,9990	0,0076	12	0	0
KNN $k = 5$	0,8174	0,9989	0,0097	12	0	0
KNN $k = 10$	0,8060	0,9992	0,0102	12	0	0
KNN $k = 15$	0,7753	0,9992	0,0120	12	0	0
KNN $k = 30$	0,7281	0,9994	0,0145	12	0	0
SVM linear	0,8436	0,9991	0,0083	12	0	0
SVM RBF	0,5029	0,9998	0,0251	11	0	1
SVM poly 2	0,8479	0,9989	0,0081	12	0	0
SVM poly 3	0,9310	0,9972	0,0058	12	0	0
Random Forest B = 15	0,8454	0,9991	0,0086	12	0	0

In Table 5.7 the results are shown from evaluating the methods by using data combination 2: training on Patient 7 with measurement A and testing on measurement B. All methods find all seizures without any misclassification, except SVM with RBF kernel which did not correctly classify any seizure.

Table 5.7: The results from training on Patient 7 with measurement A and testing on measurement B with the measures mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,4956	0,9998	0,0269	12	0	0
Logistic regression	0,6491	0,9983	0,0202	12	0	0
QDA	0,7472	0,9522	0,0587	12	0	0
KNN $k = 2$	0,5799	0,9983	0,0238	12	0	0
KNN $k = 3$	0,5371	0,9988	0,0256	12	0	0
KNN $k = 4$	0,5572	0,9989	0,0245	12	0	0
KNN $k = 5$	0,5220	0,9987	0,0265	12	0	0
KNN $k = 10$	0,5346	0,9995	0,0251	12	0	0
KNN $k = 15$	0,5270	0,9998	0,0252	12	0	0
KNN $k = 30$	0,5132	0,9998	0,0259	12	0	0
SVM linear	0,6629	0,9970	0,0207	12	0	0
SVM RBF	0,0025	0,9995	0,0532	0	0	12
SVM poly 2	0,6629	0,9970	0,0207	12	0	0
SVM poly 3	0,6629	0,9970	0,0207	12	0	0
Random Forest B = 15	0,6616	0,9970	0,0207	12	0	0

5.4.2 Generalisation of methods

The results from using Patient 7 with measurement B for training and Patients 48 and 55 for testing can be visualised in Table 5.8. All methods correctly classify all seizures, except SVM with RBF kernel that does only detect one seizure out of four. Linear regression, logistic regression, QDA, KNN with two and three neighbors, SVM with second and third polynomial kernel and Random Forest have varying number of false positives occurring in both Patient 48 and 55.

Table 5.8: The results from training on Patient 7 with measurement A and testing on Patients 48 and 55 with the measures mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,8000	0,9964	0,0045	4	2	0
Logistic regression	0,8820	0,9971	0,0035	4	2	0
QDA	0,9213	0,9620	0,0382	4	7	0
KNN k = 2	0,7902	0,9942	0,0068	4	1	0
KNN k = 3	0,8262	0,9917	0,0091	4	1	0
KNN k = 4	0,8230	0,9949	0,0060	4	0	0
KNN k = 5	0,8295	0,9933	0,0075	4	0	0
KNN k = 10	0,7869	0,9946	0,0065	4	0	0
KNN k = 15	0,7836	0,9946	0,0064	4	0	0
KNN k = 30	0,7770	0,9946	0,0065	4	0	0
SVM linear	0,8885	0,9981	0,0025	4	0	0
SVM RBF	0,0820	0,9938	0,0107	1	1	3
SVM poly 2	0,8361	0,9892	0,0116	4	4	0
SVM poly 3	0,8590	0,9474	0,0530	4	5	0
Random Forest B = 25	0,8918	0,9909	0,0095	4	1	0

The results when using Patient 48 for training and Patient 7 with measurement B for testing are shown in Table 5.9. SVM with RBF kernel does not detect any of the 12 seizures and linear regression only classifies two of them. All other methods correctly classifies all seizures and QDA also has one false positive.

Table 5.9: The results from training on Patient 48 and testing on Patient 7 with measurement B with the measures mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,0830	1,0000	0,0486	2	0	10
Logistic regression	0,6138	0,9995	0,0209	11	0	1
QDA	0,8704	0,9824	0,0235	12	1	0
KNN k = 2	0,6855	0,9996	0,0171	12	0	0
KNN k = 3	0,6717	0,9996	0,0177	12	0	0
KNN k = 4	0,6868	0,9994	0,0171	12	0	0
KNN k = 5	0,6604	0,9995	0,0185	12	0	0
KNN k = 10	0,6201	0,9998	0,0203	12	0	0
KNN k = 15	0,5937	0,9998	0,0217	12	0	0
KNN k = 30	0,5711	0,9999	0,0228	12	0	0
SVM linear	0,7119	0,9995	0,0157	12	0	0
SVM RBF	0,0000	0,9988	0,0541	0	0	12
SVM poly 2	0,6541	1,0000	0,0183	12	0	0
SVM poly 3	0,5937	0,9902	0,0308	12	0	0
Random Forest B = 25	0,6528	0,9995	0,0189	12	0	0

5.4.3 Adding high-frequent normal activities to measurements

In this section tests are done by adding distortions to the measurements to provoke misclassifications. In Table 5.10 the results from training on Patient 7 with measurement A and testing measurement B with one hour of badminton added to the measurement can be seen. QDA, linear SVM and SVM with third polynomial kernel get some false positives. When using SVM with RBF kernel all seizures are undetected. All other methods correctly classifies all events.

Table 5.10: The results from training on Patient 7 with measurement A and testing on Patient 7 with measurement B together with one hour of badminton measurements. The measures are mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,6113	0,9994	0,0171	12	0	0
Logistic regression	0,7321	0,9950	0,0163	12	0	0
QDA	0,8302	0,9091	0,0942	12	10	0
KNN k = 2	0,7031	0,9910	0,0212	12	0	0
KNN k = 3	0,6830	0,9993	0,0142	12	0	0
KNN k = 4	0,6943	0,9987	0,0143	12	0	0
KNN k = 5	0,6755	0,9978	0,0159	12	0	0
KNN k = 10	0,6780	0,9993	0,0144	12	0	0
KNN k = 15	0,6767	0,9993	0,0144	12	0	0
KNN k = 30	0,6629	0,9994	0,0150	12	0	0
SVM linear	0,7748	0,9880	0,0211	12	1	0
SVM RBF	0,0000	1,0000	0,0426	0	0	12
SVM poly 2	0,7195	0,9921	0,0195	12	0	0
SVM poly 3	0,6906	0,9411	0,0696	12	2	0
Random Forest B = 30	0,7610	0,9982	0,0120	12	0	0

In Table 5.11 the results is shown for using Patient 7 with measurement A for training and the measurement collected during dishwashing in addition to measurement B for testing. Logistic regression, QDA, KNN with two and three neighbors and SVM with third polynomial kernel classifies parts of the dishwashing as seizures. SVM with RBF kernel does not find any of the seizures. All other methods correctly classifies all events.

Table 5.11: The results from training on Patient 7 with measurement A and testing on measurement B together with measurement from dishwashing in fifteen minutes. The measures are mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,6088	0,9993	0,0203	12	0	0
Logistic regression	0,7522	0,9598	0,0507	12	6	0
QDA	0,8642	0,9187	0,0840	12	8	0
KNN k = 2	0,7157	0,9943	0,0197	12	1	0
KNN k = 3	0,6893	0,9962	0,0193	12	1	0
KNN k = 4	0,6994	0,9981	0,0170	12	0	0
KNN k = 5	0,6767	0,9983	0,0179	12	0	0
KNN k = 10	0,6692	0,9992	0,0174	12	0	0
KNN k = 15	0,6642	0,9993	0,0176	12	0	0
KNN k = 30	0,6528	0,9993	0,0181	12	0	0
SVM linear	0,7296	0,9993	0,0142	12	0	0
SVM RBF	0,0000	1,0000	0,0503	0	0	12
SVM poly 2	0,7472	0,9997	0,0130	12	0	0
SVM poly 3	0,7031	0,9696	0,0438	12	1	0
Random Forest B = 30	0,7560	0,9975	0,0147	12	0	0

In Table 5.12 the results are shown for training on Patient 7 with measurement A and testing on measurement B together with measurements on toothbrushing with electrical and normal toothbrush. All methods performed well except QDA that classified three false positives and SVM with RBF kernel that misclassified all seizures.

Table 5.12: The results from training on Patient 7 with measurement A and testing on measurement B together with measurement from toothbrushing with electrical and normal toothbrush. The measures are mean sensitivity, specificity and prediction error after median filter and TP, FP and FN after decision layer with frequency peak condition.

	Sensitivity	Specificity	Pred. Error	TP	FP	FN
Linear regression	0,64277	0,99931	0,01928	12	0	0
Logistic regression	0,76855	0,99626	0,01561	12	0	0
QDA	0,86667	0,91621	0,08637	12	3	0
KNN k = 2	0,71950	0,99599	0,01843	12	0	0
KNN k = 3	0,68931	0,99744	0,01863	12	0	0
KNN k = 4	0,70692	0,99744	0,01771	12	0	0
KNN k = 5	0,67925	0,99723	0,01935	12	0	0
KNN k = 10	0,67170	0,99758	0,01941	12	0	0
KNN k = 15	0,66415	0,99799	0,01941	12	0	0
KNN k = 30	0,64654	0,99841	0,01994	12	0	0
SVM linear	0,77484	0,99536	0,01613	12	0	0
SVM RBF	0,00000	1,00000	0,05214	0	0	12
SVM poly 2	0,67296	0,99426	0,02249	12	0	0
SVM poly 3	0,73836	0,99806	0,01548	12	0	0
Random Forest B = 10	0,73333	0,99834	0,01548	12	0	0

5.5 Test 5: Evaluating gyroscope measurements

Three tests where performed on Patient 55 to evaluate if the gyroscope adds any additional information to the classification methods. In Table 5.13 the results from only using the measurements extracted from the accelerometer is used for classification. Two methods, linear regression and KNN with 30 neighbors, classifies too few seizure time instances for the decision layer to detect both seizures.

Table 5.13: Test results when training and testing on accelerometer data from Patient 55

	Sensitivity	Specificity	Train Error	TP	FP	FN
Linear regression	0,3279	1,0000	0,0018	1	0	1
Logistic regression	0,5984	1,0000	0,0011	2	0	0
QDA	0,8033	0,9877	0,0128	2	0	0
KNN k = 2	1,0000	0,9997	0,0003	2	0	0
KNN k = 3	0,8361	1,0000	0,0004	2	0	0
KNN k = 4	0,8934	1,0000	0,0003	2	0	0
KNN k = 5	0,7541	1,0000	0,0007	2	0	0
KNN k = 10	0,7131	1,0000	0,0008	2	0	0
KNN k = 15	0,5902	1,0000	0,0011	2	0	0
KNN k = 30	0,3770	1,0000	0,0017	1	0	1
SVM linear	0,6230	1,0000	0,0010	2	0	0
SVM RBF	1,0000	0,9997	0,0003	2	0	0
SVM poly 2	0,9918	0,9998	0,0002	2	0	0
SVM poly 3	0,6803	0,9925	0,0084	2	0	0
Random Forest B = 10	1,0000	0,9999	0,0001	2	0	0

In Table 5.14 the results from using only measurements extracted from the gyroscope placed on Patient 55 are used for classification. The performance of all methods are enhanced, except for QDA, compared to the test when only using the accelerometer data. QDA gets one false positive and both linear regression and KNN with 30 neighbors finds both seizures.

Table 5.14: Test results when training and testing on gyroscope data from Patient 55

	Sensitivity	Specificity	Train Error	TP	FP	FN
Linear regression	0,4508	1,0000	0,0011	2	0	0
Logistic regression	0,7213	1,0000	0,0006	2	0	0
QDA	0,9508	0,9992	0,0009	2	1	0
KNN k = 2	1,0000	0,9998	0,0002	2	0	0
KNN k = 3	0,9672	0,9999	0,0001	2	0	0
KNN k = 4	0,9672	0,9999	0,0001	2	0	0
KNN k = 5	0,8852	1,0000	0,0002	2	0	0
KNN k = 10	0,8525	1,0000	0,0003	2	0	0
KNN k = 15	0,6393	1,0000	0,0007	2	0	0
KNN k = 30	0,4426	1,0000	0,0011	2	0	0
SVM linear	0,9098	1,0000	0,0002	2	0	0
SVM RBF	1,0000	0,9998	0,0002	2	0	0
SVM poly 2	1,0000	0,9999	0,0001	2	0	0
SVM poly 3	0,5328	1,0000	0,0010	2	0	0
Random Forest B = 10	1,0000	0,9999	0,0001	2	0	0

In Table 5.15 the results from using measurements extracted from both the accelerometer and gyroscope placed on Patient 55 are used for classification. No change in performance is made for any method except Random Forest which lowers its sensitivity with less than one percentage.

Table 5.15: Test results when training and testing on accelerometer and gyroscope data from Patient 55

	Sensitivity	Specificity	Train Error	TP	FP	FN
Linear regression	0,4508	1,0000	0,0011	2	0	0
Logistic regression	0,7213	1,0000	0,0006	2	0	0
QDA	0,9508	0,9992	0,0009	2	1	0
KNN k = 2	1,0000	0,9998	0,0002	2	0	0
KNN k = 3	0,9672	0,9999	0,0001	2	0	0
KNN k = 4	0,9672	0,9999	0,0001	2	0	0
KNN k = 5	0,8852	1,0000	0,0002	2	0	0
KNN k = 10	0,8525	1,0000	0,0003	2	0	0
KNN k = 15	0,6393	1,0000	0,0007	2	0	0
KNN k = 30	0,4426	1,0000	0,0011	2	0	0
SVM linear	0,8443	1,0000	0,0003	2	0	0
SVM RBF	1,0000	0,9998	0,0002	2	0	0
SVM poly 2	1,0000	0,9998	0,0002	2	0	0
SVM poly 3	0,5328	1,0000	0,0010	2	0	0
Random Forest B = 10	0,9918	0,9999	0,0001	2	0	0

6

Discussion

6.1 Feature performance

The features affect the performance of the classification differently depending on which method they are combined with, which can be seen in the test carried out in Section 4.4.1. In the test there were some features that did not enhance the performance in the majority of the methods. RMS and accumulated sum are two of those features, suggesting that there is a possibility that since they are derived from the mean value that the actual mean value may perform better for methods that gain performance on that type of information content. The correlation feature only enhances the performance of one method, SVM with RBF kernel, this might be due to that the sensors are also correlated in many movements not related to seizures. The correlation measure does not differ between different correlated movements, and therefore the classifier will have difficulty separating seizures from non-seizures. The frequency peak feature is in theory a promising feature since it points out in which frequency band the maximum amount of energy in a time instance is placed and we know from literature that the seizure energy should lie in bands over 2 Hz. However, it does not point out the amount of energy content and therefore the classifier can not distinguish peaks located in the higher bands containing low energy compared to similar peaks having high energy, like in a seizure. Also, a seizure can contain peaks in both high and low frequency bands since a GTC seizure has different characteristics in the frequency domain from start to end. This feature could possibly be refined by using an energy threshold that would serve to remove peaks that do not correspond to the energy levels of a seizure.

The lowest and highest frequency bands ($0.75-2.25$ Hz and $9-11.25$ Hz) seem to add information to the classifier since normal activity usually contains energy in the lower bands and no energy in the higher ones, while a seizure generates energy over many bands, especially the higher ones that normal activity usually never reaches. The difference between the lowest and highest bands are therefore clearer from seizure to normal activity than the bands distributed in between. The energy distribution over a seizure are visualised in the figures in Section 2.3. An improvement could be to lower the limit of the highest band to ensure that these frequencies actually contain energy from seizures. The lower limit of the highest band is 9 Hz, which may to

be close to the frequency limit for seizure movements in some cases. Entropy is a feature containing information that provides good classification performance for the majority of the methods that are evaluated. The reason for entropy being a good feature to use in classification of GTC seizures is most likely that the information content in data from seizures is much higher than the information content in non-seizure activities. Movements displayed by patients suffering a GTC seizure are distinctly more violent and rapid than any movements that are voluntarily generated, providing a clear difference in information content in acceleration readings.

6.1.1 Evaluation of classification performance and dimensionality reduction techniques

Cross-validation of each method together with their respective feature space provides measures on how well they are suited for classifying GTC seizures from a patient specific point of view. Their mean prediction error, sensitivity and specificity gives understanding of their overall classification performance. In this regard, all methods except linear regression and SVM with RBF kernel score over 80%, which make most methods to seem reasonably good at classifying seizures on a time instance level. When looking at the standard deviation of prediction error, sensitivity and specificity there are differences between models to be noted. KNN with $k = 2$ has a standard deviation of 0,06%, 9,08% and 0,62% for prediction error, sensitivity and specificity respectively. Compared to QDA, KNN seems to be more stable over multiple data sets used in cross-validation since QDA has higher standard deviation measures of 0,28%, 11,45% and 2.95% for prediction error, sensitivity and specificity respectively. However, it is dangerous to draw any definitive conclusion about the stability of certain methods performance over different data sets, as this evaluation is limited to only one patient. With that in mind, the results from Test 2 still indicate that methods such as KNN and SVM, among others, perform well both in terms of high means and low standard deviations of the extracted performance measures and are suitable for classification of GTC seizures.

The evaluation of whether any enhancement in performance could be gained by reducing the dimensionality of the feature space used for each method, resulted in no significant gains in sensitivity. There is, however, cases where a reduction with either PCA or PLS can achieve lower prediction error or higher specificity. Since the evaluation of PCA and PLS is done with already manually reduced feature spaces, it is possible that the manual selection of features already has reduced the feature spaces to a level near the optimal one. As the majority of methods provide a high enough sensitivity to correctly classify all seizures in the data available, manual feature selection seems to be preferable. If the methods are to be used in a more general way for a large number of patients, the manual feature selection could become too time consuming and a more automated approach where the entire feature space with all available features could be reduced automatically by variance analysis with PCA or PLS. In the choice between PCA and PLS there is little basis to make a decision from the results obtained in this thesis since no significant gain

in performance have been observed for either of the two methods.. It should be noted that despite no significant performance gains were observed for the different reductions, there are examples such as linear regression, where the sensitivity does not decrease remarkably with reduced feature spaces. This indicates that although performance may not be increased, it can be maintained when reducing the feature space, which could be helpful if it is desired to reduce the computational effort in the future.

6.2 Decision layer parameters

The median filter works not only as a filter to improve the specificity by removing misclassified non-seizure time instances but also to improve the sensitivity. This is due to that the characteristics throughout a GTC seizure changes and the accelerometer data for the beginning and the end of it can look similar to normal activity, causing methods to have difficulty classifying these seconds as seizures. By having a longer filter length than the threshold the correctly classified seizures can be smeared and the sensitivity for time instance classification is then improved. Also, the median filter serves to remove misclassified seconds in the middle of a seizure. Since the change in characteristics over a seizure can lower the sensitivity, the decision threshold cannot be as large as a typical seizure is. The decision layer parameters selected through the empirical study are therefore reasonable.

6.3 Patient specific classification

Two tests were made to evaluate the capability of methods detecting seizures in measurements made on the same patient used for training. The first test consisted of four fold cross-validation of measurement B from Patient 7. The second test consisted of training on measurement A from Patient 7 and testing on measurement B from Patient 7. Even though the seizures in each measurement differ in appearance nearly all methods can find all seizures in the tests. However, the sensitivity is far worse for the latter test which shows that the seizures do differ in characteristics between the measurement occasions. The specificity affects the amount of false positives and does not change drastically between the tests. The effect of specificity is evident since no FP are added, except in QDA that in general seems to give comparably lower specificity to other methods. These results should be treated with caution since only one patient is tested and it should only be used as an insight for further testings when more patient measurements are available. Also, the feature spaces generated for these tests can contain information that is too patient specific.

6.4 Robustness against high-frequency normal activity

The robustness in methods against high-frequency normal activity works well for many of the methods and for some it only generates a single false positive in total. The most robust methods are linear regression, KNN with 4,5,10 and 30 neighbors, SVM with second order polynomial kernel and Random Forest. The robustness in these methods is arguably because they use simple functions to separate data, especially linear regression, making them insensitive to disturbances. KNN being robust for higher numbers of neighbors supports this assumption since the decision boundary produced for a high number of neighbors is smoother and less complex than a decision boundary produced with a low neighbor count. QDA performs worse than SVM with second order polynomial kernel despite both models using a rather simple quadratic boundary. The reason for this could be that the covariance matrix estimation used in QDA is inaccurate due to the limited amount of seizure data available.

The badminton measurement may contain many instances of powerful motions but they are generally short and not rhythmic which could be the reason to why most of the methods are robust against these types of movements. It was stated in the paper published by S. Beniczky [10] that toothbrushing, which is a rhythmic activity, introduces false positivies in classification with a threshold. Our results show that toothbrushing does in fact not deteriorate classification performance which indicates that the approach of manually selecting features that highlight seizures in measurements for each machine learning method is successful. The measurements from dishwashing causes false positives in some instances. The movements from dishwashing are also rhythmic but contain higher amplitudes than toothbrushing since movements are typically larger and more forceful. Possibly the higher amplitudes that are periodic and repeated for longer time sequences found in dishwashing measurements closely resemble seizure data in feature space, which causes some classifiers to misclassify the data as seizures.

To get a more general evaluation of the robustness against high frequency motions more activities need to be tested. These type of activities should be rapid and/or rhythmic movements, e.g. running, fitness class, basketball, running on cross-trainer and rapid weight lifting. Also, documentation on which type of movements that the patients are exposing themselves to, especially the rhythmic and/or powerful ones, would help when investigating the robustness of methods and the occurrence of false positives.

6.5 Generalisation qualities in methods

The thesis made by A. Hildeman stated the possibility of finding methods that can classify seizures on multiple patients [12]. We evaluated the possibility with measurements from the three different patients that are available and concluded that the ability for generalisation of models may be possible. All methods succeed in detecting all seizures for the tests with results shown in Tables 5.8 and 5.9 but when training on Patient 7 with measurement B and testing on patients 48 and 55 some methods produce false positives. The measurement from Patient 55 contains powerful activity stemming from physical exercise and we believe that these activities are the ones causing some of the false positives. Since these activities are not specified in time or what type of exercises that are performed we can not conclude that this is the case. Another possible reason could be the different placement of the sensors causing the measurements to differ from the other patients since it may not have the same information content.

The test for training on Patient 48 and testing on Patient 7 with measurement B gave promising results since the training data only contains two seizures and still the majority of all methods succeeded to classify all seizures in measurement B. Also, the sensitivity and specificity increased for this test compared to training and testing on Patient 7 with measurement A and B, respectively. This could possibly be caused by models overfitting when training on measurement A. Since the seizures differ between the two measurement occasions it is difficult for the models to adapt to the second measurement. It could also be the other way around, that Patient 48 overfits the model due to that it only contains two seizures and that the seizures in Patient 7 with measurement B coincide with the characteristics in those two.

6.6 Evaluation of gyroscope measurement in addition to acceleration data

The results from evaluating the performance of using gyroscope measurement for classification instead of accelerometer data shows that the gyroscope could carry information that easier separates the seizures from normal activity. All methods get decreased training error, except Random Forest where it is unchanged. When both accelerometer and gyroscope measurements are used together no gain is obtained compared to when only gyroscope is used. The gyroscope has one disadvantage since it is more energy consuming than the accelerometer and an increase in performance is therefore not decisive for which sensors to use. If the measurement from an accelerometer makes some ML methods detect all seizures, even if the prediction error is higher than for a gyroscope, then that sensor is presumably more worth having in a user perspective since the batteries will not need to be recharged as often.

7

Conclusion

The ML methods performing the best in this project are:

- KNN with $k > 3$
- Random Forest

KNN with a lower number of neighbors generates a small amount of false positives when the class separation is impeded since the specificity decreases for an enhancement in sensitivity. Random Forest generally has a high sensitivity compared to the other methods but the specificity can decrease for measurements that are non-separable and in turn generate false positives. Both methods are robust against rhythmic and powerful activities. To establish which of the two methods that is more robust the cross-validation test results in Section 5.2.7 can be evaluated. It shows that Random Forest has a higher standard deviation in terms of specificity and may therefore not be as robust as KNN even though it has a lower training error.

The simpler methods, QDA, linear- and logistic regression, do not have the same robustness and/or generalisation qualities as KNN and Random Forest. Also, SVM can be a bit unpredictable where all kernels tested gave different results. The kernels that performed the worst were RBF and third order polynomial which both did not perform well in the high frequency motion test and the generalisation test. By sweeping a higher resolution of box constraints the kernels may perform better in seizure classification for all tests. However, that is a time consuming task.

To be able to reliably conclude which methods that will work for classifying GTC seizures, both with patient specific models and the more general case when training and testing on different patients, more acceleration data needs to be collected from patients having GTC seizures. To make a model patient specific more than two seizures are needed from Patient 48 and 55 to get a statistical base for choosing methods. For generalisation of a model more GTC seizures that differs beyond the general GTC characteristics are needed to be able to ensure that the best methods from this evaluation still give sufficient performance.

7.1 Future work

This thesis is a part of a project that aims at classifying epileptic seizures from measurements extracted by wearable sensors sewn into clothes worn by patients in their normal environment undergoing analysis at Sahlgrenska University Hospital. This study is therefore a beginning of finding a ML method that could detect different kinds of epileptic seizures in the future, either on specific patients or generally for a selection of patients. To conclude which methods that work well with GTC seizures more data needs to be collected for further investigation. It is also desired to collect out-of-clinic movements since the environment at the hospital restricts the amount of different activities the patients can conduct.

The credibility of the results would probably be increased if all measurements were done with a prototype of the garment to ensure that the same type of sensors are used and that they are placed similarly on the patients. The measurements will then behave similar when using the final product and the performance of the chosen classification method would not be decreased due to different measurement settings.

This study has only involved GTC seizures and the features, methods and decision layer may not work for other seizure types. The decision layer is tailored to suit seizures with a duration of at least one minute and therefore a change is needed to suit shorter seizures or a different type of decision layer needs to be designed. Instead of designing a decision layer that can detect all wanted types of seizures, different decision layers tailored for each seizure type could be applied to the predicted classification vector for a complete detection of every type of seizure.

Even though the accelerometer measurements are enough for the models to detect all GTC seizures in this project, the gyroscope should be investigated more thoroughly since it showed encouraging results. It may provide information missed by the accelerometers that will help the methods in separating other types of seizures from normal activities.

Bibliography

[1] Schulc E, Unterberger I, Saboor S, Hilbe J, Ertl M, Ammenwerth E, et al. Measurement and quantification of generalized tonic-clonic seizures in epilepsy patients by means of accelerometry-An explorative study. Epilepsy Research. 2011 March;2011(95):173–183.

[2] Poh MZ, Loddenkemper T, Reinsberger C, Swenson NC, Goyal S, Sabtala MC, et al. Convulsive seizure detection using a wrist-worn electrodermal activity and accelerometry biosensor. Epilepsia. 2012 May;53(5):93–97.

[3] Nijsen T, Arends J, Aarts R, Cluitmans P. Automated detection of tonic seizures using 3-D accelerometry. IFMBE Proceedings. 2008;22(1):188–191.

[4] Sirven J, Shafer P. What is epilepsy? [homepage on the Internet]; [updated 2014 January; cited 2015 March 25]. Available from: `http://www.epilepsy.com/learn/epilepsy-101/what-epilepsy`.

[5] Fisher RS, Acevedo C, Arzimanoglou A, Bogoacz A, H CJ, Elger CE, et al. ILAE Official Report: A practical clinical definition of epilepsy. Epilepsia. 2014 April;55(4):475–482.

[6] Nijsen T, Arends J, Griep P, Cluitmans P. The potential value of three-dimensional accelerometry for detection of motor seizures in severe epilepsy. Epilepsy & Behavior. 2005 Aug;7(1):74–84.

[7] Nijsen T, Arends J, Griep P, Cluitmans P. Detection of Subtle Nocturnal Motor Activity From 3-D Accelerometry Recordings in Epilepsy Patients. IEEE Transactions on Biomedical Engineering. 2007 Nov;54(11):2073–2081.

[8] Nijsen T, Aarts R, Griep P, Cluitmans P. Time-Frequency Analysis of Accelerometry Data for Detection of Myoclonic Seizures. IEEE Transactions on Information Technology in Biomedicine. 2010 Sept;14(5):1197–1203.

[9] Lockman J, Fisher RS, Olson DM. Detection of seizure-like movements using a wrist accelerometer. Epilepsy and Behavior. 2011 March;2011(20):638–641.

[10] Beniczky S, Polster T, Kjaer TW, Hjalgrim H. Detection of generalized tonic–clonic seizures by a wireless wrist accelerometer: A prospective, multicenter study. Epilepsia. 2013;54(4):58–61.

[11] Wipenmyr J. Utveckla metoder/algoritmer for att diskriminera epileptiska anfall fran normalt beteende. Luleå: Luleå University of Technology;. Retrieved from Acreo.

[12] Hildeman A. Classification of epileptic seizures using accelerometers. Gothenburg: Department of Mathematical Sciences, Mathematical Statistics, Chalmers University of Technology; 2011.

[13] Allen JB. Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform. IEEE Transactions on acoustics, speech, and signal processing. 1977 June;25(3):235–238.

[14] Shannon CE. A Mathematical Theory of Communication. The Bell System Technical Journal. 1948 July;27(3):379–423.

[15] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. 2nd ed. Stanford, California: Springer-Verlag; 2008. Available from: `http://statweb.stanford.edu/~tibs/ElemStatLearn/`.

[16] Smith IL. A tutorial on Principal Components Analysis. New Zealand: Department of Computer Science, University of Otago; 2002.

[17] Flach P. Machine Learning: The Art and Science of Algorithms that Make Sense of Data. 4th ed. Cambridge, United Kingdom: Cambridge University Press; 2012.

[18] Li J, Yu Y. STAT557: Data Mining [homepage on the Internet]; [updated 2014 December; cited 2015 June 26]. Available from: `https://onlinecourses.science.psu.edu/stat557/`.

[19] Lin HT, Lin CH. A study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods. Taiwan: Department of Computer Science and information engineering, National Taiwan University; 2003.

A

Lists of figures and tables

List of Figures

List of Tables

B

Appendix

B.1 Results from testing feature spaces on different methods

This appendix contains all test results from evaluating features combined with classification method described in 4.4.1.

B.1.1 Results with linear regression

In Table B.1 the results from using linear regression as classification method is shown.

Table B.1: Measures from tests on different feature spaces with linear regression

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,7238	0,7017	0,7017	0,7238	0,7072	0,6961	0,7017	0,7017
Specificity:	0,9982	0,9987	0,9989	0,9987	0,9985	0,9987	0,9987	0,9985
Error:	0,0107	0,0109	0,0107	0,0102	0,0109	0,0111	0,0109	0,0111

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,2044	0,5249	0,6961	0,6796	0,7238	0,6906	0,7072
Specificity:	0,9950	0,9991	0,9987	0,9991	0,9993	0,9993	0,9985
Error:	0,0305	0,0163	0,0111	0,0113	0,0096	0,0107	0,0109

B.1.2 Results with logistic regression

In Table B.2 the results from using logistic regression as classification method is shown.

Table B.2: Measures from tests on different feature spaces with logistic regression

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,7403	0,7127	0,7017	0,7514	0,7127	0,6906	0,7017	0,7072
Specificity:	0,9906	0,9941	0,9939	0,9946	0,9935	0,9939	0,9941	0,9926
Error:	0,0175	0,0150	0,0155	0,0132	0,0155	0,0159	0,0154	0,0166

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,3923	0,6851	0,7790	0,7735	0,8122	0,7790	0,7127
Specificity:	0,9906	0,9939	0,9969	0,9935	0,9961	0,9935	0,9937
Error:	0,0288	0,0161	0,0102	0,0136	0,0098	0,0134	0,0154

B.1.3 Results with quadratic discriminant analysis

In Table B.3 the results from using quadratic discriminant analysis as classification method is shown.

Table B.3: Measures from tests on different feature spaces with QDA

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9724	0,9669	0,9669	0,9669	0,9724	0,9669	0,9669	0,9724
Specificity:	0,9029	0,9046	0,9044	0,9068	0,9053	0,9039	0,9042	0,9044
Error:	0,0948	0,0934	0,0936	0,0913	0,0925	0,0941	0,0938	0,0934

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,8453	0,9669	0,9613	0,9724	0,9669	0,9669	0,9669
Specificity:	0,8529	0,9059	0,8904	0,9144	0,8996	0,9146	0,9068
Error:	0,1473	0,0921	0,1073	0,0838	0,0982	0,0838	0,0913

B.1.4 Results with k-nearest neighbors

In Tables B.4-B.10 the results from using k-nearest neighbors as classification method is shown.

Table B.4: Measures from tests on different feature spaces with KNN having two neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9171	0,9006	0,8895	0,8950	0,9061	0,8895	0,9061	0,9006
Specificity:	0,9808	0,9791	0,9812	0,9810	0,9780	0,9801	0,9825	0,9782
Error:	0,0213	0,0234	0,0218	0,0218	0,0243	0,0229	0,0200	0,0243

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,7514	0,9337	0,8785	0,8619	0,9006	0,8674	0,8729
Specificity:	0,9646	0,9779	0,9801	0,9839	0,9786	0,9841	0,9801
Error:	0,0423	0,0236	0,0232	0,0200	0,0239	0,0196	0,0234

Table B.5: Measures from tests on different feature spaces with KNN having three neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8785	0,8785	0,8785	0,8729	0,8729	0,8729	0,8785	0,8729
Specificity:	0,9932	0,9915	0,9922	0,9917	0,9911	0,9919	0,9924	0,9911
Error:	0,0105	0,0121	0,0114	0,0121	0,0127	0,0120	0,0113	0,0127

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,6685	0,8950	0,8619	0,8453	0,8840	0,8398	0,8232
Specificity:	0,9806	0,9919	0,9926	0,9952	0,9897	0,9952	0,9921
Error:	0,0295	0,0113	0,0116	0,0096	0,0138	0,0098	0,0134

Table B.6: Measures from tests on different feature spaces with KNN having four neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8840	0,8785	0,8840	0,8729	0,8729	0,8729	0,8785	0,8729
Specificity:	0,9915	0,9887	0,9904	0,9904	0,9889	0,9893	0,9906	0,9884
Error:	0,0120	0,0148	0,0130	0,0134	0,0148	0,0145	0,0130	0,0154

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,7238	0,9061	0,8729	0,8619	0,8950	0,8508	0,8453
Specificity:	0,9766	0,9897	0,9904	0,9937	0,9878	0,9932	0,9900
Error:	0,0316	0,0130	0,0134	0,0105	0,0152	0,0114	0,0146

Table B.7: Measures from tests on different feature spaces with KNN having five neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8674	0,8674	0,8674	0,8674	0,8674	0,8564	0,8619	0,8674
Specificity:	0,9950	0,9937	0,9941	0,9939	0,9935	0,9941	0,9939	0,9934
Error:	0,0091	0,0104	0,0100	0,0102	0,0105	0,0104	0,0104	0,0107

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,6630	0,9006	0,8508	0,8398	0,8840	0,8343	0,8177
Specificity:	0,9832	0,9941	0,9937	0,9956	0,9924	0,9956	0,9939
Error:	0,0271	0,0089	0,0109	0,0095	0,0111	0,0096	0,0118

Table B.8: Measures from tests on different feature spaces with KNN having ten neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8619	0,8619	0,8619	0,8564	0,8619	0,8564	0,8564	0,8619
Specificity:	0,9959	0,9943	0,9950	0,9946	0,9948	0,9946	0,9943	0,9946
Error:	0,0084	0,0100	0,0093	0,0098	0,0095	0,0098	0,0102	0,0096

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,6740	0,8950	0,8508	0,8398	0,8729	0,8398	0,8177
Specificity:	0,9815	0,9950	0,9937	0,9958	0,9932	0,9959	0,9946
Error:	0,0284	0,0082	0,0109	0,0093	0,0107	0,0091	0,0111

Table B.9: Measures from tests on different feature spaces with KNN having 15 neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8508	0,8453	0,8453	0,8343	0,8508	0,8453	0,8398	0,8508
Specificity:	0,9969	0,9967	0,9965	0,9965	0,9967	0,9967	0,9965	0,9967
Error:	0,0079	0,0082	0,0084	0,0088	0,0080	0,0082	0,0086	0,0080

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,6133	0,8895	0,8453	0,8343	0,8619	0,8232	0,8122
Specificity:	0,9862	0,9970	0,9948	0,9963	0,9946	0,9970	0,9961
Error:	0,0259	0,0064	0,0100	0,0089	0,0096	0,0086	0,0098

Table B.10: Measures from tests on different feature spaces with KNN having 30 neighbors

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8177	0,8066	0,8066	0,8066	0,8066	0,8066	0,8232	0,8066
Specificity:	0,9970	0,9970	0,9976	0,9970	0,9970	0,9970	0,9967	0,9970
Error:	0,0088	0,0091	0,0086	0,0091	0,0091	0,0091	0,0089	0,0091

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5525	0,8398	0,8177	0,8177	0,8232	0,7845	0,7901
Specificity:	0,9862	0,9976	0,9958	0,9972	0,9958	0,9970	0,9976
Error:	0,0279	0,0075	0,0100	0,0086	0,0098	0,0098	0,0091

B.1.5 Results with linear Support Vector Machine

In Tables B.11-B.15 the results from using linear Support Vector Machine as classification method is shown.

Table B.11: Measures from tests on different feature spaces with linear SVM and box constraint of 0.1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8729	0,8287	0,8287	0,8453	0,8343	0,8232	0,8287	0,8453
Specificity:	0,9902	0,9913	0,9908	0,9900	0,9900	0,9917	0,9910	0,9897
Error:	0,0136	0,0139	0,0145	0,0146	0,0150	0,0138	0,0143	0,0150

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4144	0,8343	0,8232	0,8177	0,8343	0,8011	0,8287
Specificity:	0,9893	0,9893	0,9937	0,9937	0,9932	0,9928	0,9915
Error:	0,0293	0,0157	0,0118	0,0120	0,0120	0,0134	0,0138

Table B.12: Measures from tests on different feature spaces with linear SVM and box constraint of 1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8398	0,8398	0,8398	0,8453	0,8398	0,8343	0,8398	0,8508
Specificity:	0,9862	0,9887	0,9889	0,9875	0,9882	0,9891	0,9891	0,9863
Error:	0,0186	0,0161	0,0159	0,0171	0,0166	0,0159	0,0157	0,0180

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4254	0,8343	0,8232	0,8122	0,8398	0,8066	0,8398
Specificity:	0,9889	0,9871	0,9937	0,9924	0,9915	0,9919	0,9893
Error:	0,0293	0,0179	0,0118	0,0134	0,0134	0,0141	0,0155

Table B.13: Measures from tests on different feature spaces with linear SVM and box constraint of 10

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8508	0,8343	0,8398	0,8453	0,8398	0,8453	0,8398	0,8453
Specificity:	0,9830	0,9862	0,9858	0,9839	0,9867	0,9867	0,9854	0,9843
Error:	0,0213	0,0188	0,0189	0,0205	0,0180	0,0179	0,0193	0,0202

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4254	0,8398	0,8232	0,8122	0,8398	0,8066	0,8398
Specificity:	0,9889	0,9843	0,9937	0,9924	0,9915	0,9917	0,9863
Error:	0,0293	0,0204	0,0118	0,0134	0,0134	0,0143	0,0184

Table B.14: Measures from tests on different feature spaces with linear SVM and box constraint of 100

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8619	0,8398	0,8398	0,8453	0,8453	0,8453	0,8398	0,8453
Specificity:	0,9804	0,9860	0,9854	0,9834	0,9847	0,9847	0,9851	0,9847
Error:	0,0234	0,0188	0,0193	0,0211	0,0198	0,0198	0,0196	0,0198

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4254	0,8453	0,8232	0,8122	0,8398	0,8066	0,8453
Specificity:	0,9889	0,9832	0,9937	0,9941	0,9908	0,9924	0,9849
Error:	0,0293	0,0213	0,0118	0,0118	0,0141	0,0136	0,0196

Table B.15: Measures from tests on different feature spaces with linear SVM and box constraint of 1000

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8729	0,8785	0,8453	0,8232	0,7569	0,8066	0,7956	0,8232
Specificity:	0,9779	0,9633	0,9771	0,9753	0,9766	0,9823	0,9875	0,9714
Error:	0,0255	0,0395	0,0271	0,0296	0,0305	0,0234	0,0188	0,0334

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,0939	0,8177	0,8343	0,7956	0,8840	0,7403	0,8232
Specificity:	0,9980	0,9863	0,9871	0,9775	0,9812	0,9930	0,9795
Error:	0,0313	0,0191	0,0179	0,0284	0,0220	0,0152	0,0255

B.1.6 Results with Support Vector Machine with second polynomial

In Tables B.16-B.20 the results from using Support Vector Machine with second polynomial kernel as classification method is shown.

Table B.16: Measures from tests on different feature spaces with SVM having second polynomial as kernel and box constraint of 0.1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8840	0,8729	0,8508	0,8729	0,8895	0,8398	0,8564	0,8674
Specificity:	0,9791	0,9803	0,9797	0,9812	0,9817	0,9828	0,9797	0,9797
Error:	0,0239	0,0232	0,0245	0,0223	0,0213	0,0218	0,0243	0,0239

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4530	0,8840	0,7901	0,8729	0,9171	0,8785	0,8453
Specificity:	0,9889	0,9801	0,9932	0,9858	0,9793	0,9871	0,9841
Error:	0,0284	0,0230	0,0134	0,0179	0,0227	0,0164	0,0204

Table B.17: Measures from tests on different feature spaces with SVM having second polynomial as kernel and box constraint of 1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9116	0,8398	0,8785	0,8674	0,8729	0,8564	0,8674	0,8674
Specificity:	0,9743	0,9668	0,9638	0,9707	0,9725	0,9710	0,9660	0,9731
Error:	0,0277	0,0373	0,0389	0,0327	0,0307	0,0327	0,0371	0,0304

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4862	0,8564	0,7845	0,8564	0,9227	0,8840	0,8729
Specificity:	0,9897	0,9660	0,9928	0,9749	0,9721	0,9843	0,9769
Error:	0,0266	0,0375	0,0139	0,0289	0,0295	0,0189	0,0264

Table B.18: Measures from tests on different feature spaces with SVM having second polynomial as kernel and box constraint of 10

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8895	0,8619	0,8343	0,8453	0,8619	0,8674	0,8508	0,8619
Specificity:	0,8717	0,9662	0,9631	0,9720	0,8970	0,9577	0,9664	0,6582
Error:	0,1277	0,0371	0,0411	0,0321	0,1041	0,0452	0,0373	0,3352

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5028	0,8674	0,7956	0,8287	0,8950	0,8564	0,8895
Specificity:	0,9887	0,9653	0,9922	0,9651	0,9670	0,9793	0,9605
Error:	0,0270	0,0379	0,0141	0,0393	0,0354	0,0246	0,0418

Table B.19: Measures from tests on different feature spaces with SVM having second polynomial as kernel and box constraint of 100

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8785	0,8840	0,8785	0,8287	0,8840	0,8508	0,8453	0,8619
Specificity:	0,9631	0,8775	0,6612	0,9616	0,9260	0,3641	0,8952	0,9138
Error:	0,0396	0,1223	0,3318	0,0427	0,0754	0,6202	0,1064	0,0879

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5470	0,9006	0,8066	0,7901	0,9116	0,8232	0,8564
Specificity:	0,9924	0,9539	0,9828	0,9385	0,9552	0,9391	0,9566
Error:	0,0220	0,0479	0,0229	0,0663	0,0463	0,0646	0,0466

Table B.20: Measures from tests on different feature spaces with SVM having second polynomial as kernel and box constraint of 1000

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8840	0,9116	0,8729	0,8674	0,8950	0,8840	0,8508	0,8453
Specificity:	0,9673	0,8968	0,8389	0,9590	0,9269	0,3423	0,8431	0,9666
Error:	0,0354	0,1027	0,1600	0,0439	0,0741	0,6402	0,1566	0,0373

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,8840	0,8840	0,0000	0,8564	0,9006	0,8122	0,8508
Specificity:	0,9378	0,8913	0,9430	0,9088	0,9317	0,8479	0,9550
Error:	0,0639	0,1089	0,0875	0,0929	0,0693	0,1532	0,0484

B.1.7 Results with Support Vector Machine with third polynomial

In Tables B.21-B.25 the results from using Support Vector Machine with third polynomial kernel as classification method is shown.

Table B.21: Measures from tests on different feature spaces with SVM having third polynomial as kernel and box constraint of 0.1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9282	0,9282	0,9116	0,9171	0,9282	0,9061	0,9282	0,9116
Specificity:	0,9771	0,9749	0,9710	0,9753	0,9743	0,9784	0,9707	0,9740
Error:	0,0245	0,0266	0,0309	0,0266	0,0271	0,0239	0,0307	0,0280

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5414	0,9171	0,8785	0,9006	0,9337	0,9392	0,8619
Specificity:	0,9841	0,9760	0,9871	0,9812	0,9738	0,9755	0,9801
Error:	0,0302	0,0259	0,0164	0,0214	0,0275	0,0257	0,0238

Table B.22: Measures from tests on different feature spaces with SVM having third polynomial as kernel and box constraint of 1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9337	0,9116	0,9171	0,9116	0,9116	0,9116	0,8564	0,9116
Specificity:	0,9692	0,9729	0,9686	0,9732	0,9720	0,9736	0,9701	0,9690
Error:	0,0320	0,0291	0,0330	0,0288	0,0300	0,0284	0,0336	0,0329

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,3867	0,9006	0,8840	0,9227	0,9171	0,9558	0,8564
Specificity:	0,9830	0,9745	0,9734	0,9762	0,9766	0,9684	0,9644
Error:	0,0363	0,0279	0,0295	0,0255	0,0254	0,0320	0,0391

Table B.23: Measures from tests on different feature spaces with SVM having third polynomial as kernel and box constraint of 10

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9282	0,9282	0,9116	0,9171	0,9282	0,9061	0,9282	0,9116
Specificity:	0,9771	0,9749	0,9710	0,9753	0,9743	0,9784	0,9707	0,9740
Error:	0,0245	0,0266	0,0309	0,0266	0,0271	0,0239	0,0307	0,0280

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5414	0,9171	0,8785	0,9006	0,9337	0,9392	0,8619
Specificity:	0,9841	0,9760	0,9871	0,9812	0,9738	0,9755	0,9801
Error:	0,0302	0,0259	0,0164	0,0214	0,0275	0,0257	0,0238

Table B.24: Measures from tests on different feature spaces with SVM having third polynomial as kernel and box constraint of 100

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8840	0,9116	0,8729	0,8674	0,8950	0,8840	0,8508	0,8453
Specificity:	0,9673	0,8968	0,8389	0,9590	0,9269	0,3423	0,8431	0,9666
Error:	0,0354	0,1027	0,1600	0,0439	0,0741	0,6402	0,1566	0,0373

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,8840	0,8840	0,0000	0,8564	0,9006	0,8122	0,8508
Specificity:	0,9378	0,8913	0,9430	0,9088	0,9317	0,8479	0,9550
Error:	0,0639	0,1089	0,0875	0,0929	0,0693	0,1532	0,0484

Table B.25: Measures from tests on different feature spaces with SVM having third polynomial as kernel and box constraint of 1000

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9061	0,9061	0,9227	0,9006	0,9116	0,8950	0,9006	0,9116
Specificity:	0,9688	0,9638	0,9611	0,9679	0,9638	0,9594	0,9636	0,9668
Error:	0,0332	0,0380	0,0402	0,0343	0,0379	0,0427	0,0384	0,0350

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4033	0,9171	0,8674	0,9006	0,9171	0,9171	0,8729
Specificity:	0,1757	0,9600	0,3253	0,9673	0,9745	0,9642	0,9325
Error:	0,8170	0,0414	0,6571	0,0348	0,0273	0,0373	0,0695

B.1.8 Results with Support Vector Machine with radial basis function kernel

In Tables B.26-B.30 the results from using Support Vector Machine with radial basis function kernel as classification method is shown.

Table B.26: Measures from tests on different feature spaces with SVM having radial basis function as kernel and box constraint of 0.1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0	0	0	0	0	0	0	0
Specificity:	1	1	1	1	1	1	1	1
Error:	0,0323	0,0323	0,0323	0,0323	0,0323	0,0323	0,0323	0,0323

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,1215	0	0	0	0	0	0
Specificity:	0,9987	1	1	1	1	1	1
Error:	0,0296	0,0323	0,0323	0,0323	0,0323	0,0323	0,0323

Table B.27: Measures from tests on different feature spaces with SVM having radial basis function as kernel and box constraint of 1

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,0221	0,0276	0,0276	0,0276	0,0221	0,0276	0,0276	0,0221
Specificity:	1,0000	0,9996	0,9998	0,9996	0,9996	0,9996	0,9996	0,9996
Error:	0,0316	0,0318	0,0316	0,0318	0,0320	0,0318	0,0318	0,0320

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,4862	0,0497	0,3149	0,0276	0,0331	0,0331	0,0166
Specificity:	0,9915	0,9996	0,9991	0,9996	0,9998	0,9993	0,9994
Error:	0,0248	0,0311	0,0230	0,0318	0,0314	0,0320	0,0323

Table B.28: Measures from tests on different feature spaces with SVM having radial basis function as kernel and box constraint of 10

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,0221	0,0276	0,0276	0,0276	0,0276	0,0276	0,0276	0,0276
Specificity:	0,9996	0,9985	0,9982	0,9987	0,9991	0,9982	0,9976	0,9994
Error:	0,0320	0,0329	0,0332	0,0327	0,0323	0,0332	0,0338	0,0320

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5359	0,0497	0,3315	0,0276	0,0608	0,0387	0,0166
Specificity:	0,9854	0,9985	0,9941	0,9982	0,9976	0,9982	0,9963
Error:	0,0291	0,0321	0,0273	0,0332	0,0327	0,0329	0,0354

Table B.29: Measures from tests on different feature spaces with SVM having radial basis function as kernel and box constraint of 100

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,0276	0,0276	0,0276	0,0276	0,0276	0,0221	0,0276	0,0276
Specificity:	0,9993	0,9972	0,9976	0,9978	0,9974	0,9965	0,9961	0,9958
Error:	0,0321	0,0341	0,0338	0,0336	0,0339	0,0350	0,0352	0,0355

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5304	0,0663	0,3039	0,0276	0,0608	0,0387	0,0221
Specificity:	0,9769	0,9963	0,9911	0,9967	0,9969	0,9967	0,9934
Error:	0,0375	0,0338	0,0311	0,0346	0,0334	0,0343	0,0380

Table B.30: Measures from tests on different feature spaces with SVM having radial basis function as kernel and box constraint of 1000

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,0276	0,0276	0,0221	0,0276	0,0276	0,0276	0,0276	0,0276
Specificity:	0,9993	0,9961	0,9967	0,9974	0,9969	0,9934	0,9950	0,9950
Error:	0,0321	0,0352	0,0348	0,0339	0,0345	0,0379	0,0363	0,0363

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,5414	0,0663	0,2707	0,0276	0,0552	0,0331	0,0221
Specificity:	0,9779	0,9948	0,9902	0,9952	0,9952	0,9948	0,9891
Error:	0,0363	0,0352	0,0330	0,0361	0,0352	0,0363	0,0421

B.1.9 Results with Random Forest

In Tables B.31-B.36 the results from using Random Forest as classification method is shown.

Table B.31: Measures from tests on different feature spaces with Random Forest with 5 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8950	0,8729	0,8674	0,8840	0,8895	0,8895	0,9006	0,9337
Specificity:	0,9897	0,9902	0,9851	0,9863	0,9889	0,9893	0,9882	0,9780
Error:	0,0134	0,0136	0,0188	0,0170	0,0143	0,0139	0,0146	0,0234

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,3149	0,9006	0,7182	0,8729	0,8950	0,8232	0,8564
Specificity:	0,9871	0,9801	0,9790	0,9871	0,9926	0,9887	0,9851
Error:	0,0346	0,0225	0,0295	0,0166	0,0105	0,0166	0,0191

Table B.32: Measures from tests on different feature spaces with Random Forest with 10 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8674	0,8840	0,8785	0,8453	0,8950	0,9061	0,8840	0,8508
Specificity:	0,9950	0,9913	0,9910	0,9910	0,9895	0,9887	0,9911	0,9887
Error:	0,0091	0,0121	0,0127	0,0138	0,0136	0,0139	0,0123	0,0157

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,2431	0,8729	0,8343	0,8895	0,9006	0,8619	0,8729
Specificity:	0,9889	0,9919	0,9919	0,9910	0,9913	0,9895	0,9867
Error:	0,0352	0,0120	0,0132	0,0123	0,0116	0,0146	0,0170

Table B.33: Measures from tests on different feature spaces with Random Forest with 15 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9006	0,9337	0,9116	0,8785	0,8785	0,8840	0,9061	0,8895
Specificity:	0,9900	0,9887	0,9891	0,9902	0,9911	0,9911	0,9902	0,9900
Error:	0,0129	0,0130	0,0134	0,0134	0,0125	0,0123	0,0125	0,0132

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,2983	0,8840	0,7790	0,9006	0,8895	0,8895	0,8950
Specificity:	0,9882	0,9906	0,9926	0,9876	0,9930	0,9921	0,9882
Error:	0,0341	0,0129	0,0143	0,0152	0,0104	0,0113	0,0148

Table B.34: Measures from tests on different feature spaces with Random Forest with 20 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9006	0,8840	0,8674	0,8785	0,8785	0,8564	0,9006	0,9116
Specificity:	0,9917	0,9917	0,9904	0,9930	0,9924	0,9908	0,9906	0,9887
Error:	0,0113	0,0118	0,0136	0,0107	0,0113	0,0136	0,0123	0,0138

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,2597	0,9006	0,7901	0,8895	0,9061	0,8508	0,8619
Specificity:	0,9899	0,9926	0,9910	0,9899	0,9926	0,9891	0,9902
Error:	0,0338	0,0104	0,0155	0,0134	0,0102	0,0154	0,0139

Table B.35: Measures from tests on different feature spaces with Random Forest with 25 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,8785	0,8785	0,8674	0,8895	0,9061	0,8729	0,9061	0,8729
Specificity:	0,9908	0,9904	0,9893	0,9917	0,9913	0,9917	0,9900	0,9895
Error:	0,0129	0,0132	0,0146	0,0116	0,0114	0,0121	0,0127	0,0143

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,2376	0,9006	0,8343	0,8895	0,8895	0,8729	0,8508
Specificity:	0,9928	0,9895	0,9930	0,9900	0,9930	0,9921	0,9878
Error:	0,0316	0,0134	0,0121	0,0132	0,0104	0,0118	0,0166

Table B.36: Measures from tests on different feature spaces with Random Forest with 30 trees

Feature Space:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Sensitivity:	0,9006	0,8950	0,9061	0,8619	0,8895	0,8729	0,8895	0,8950
Specificity:	0,9926	0,9910	0,9878	0,9911	0,9908	0,9910	0,9928	0,9917
Error:	0,0104	0,0121	0,0148	0,0130	0,0125	0,0129	0,0105	0,0114

Feature Space:	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Sensitivity:	0,9006	0,7901	0,8895	0,8785	0,8895	0,8895	0,8950
Specificity:	0,9910	0,9922	0,9910	0,9926	0,9922	0,9886	0,9917
Error:	0,0120	0,0143	0,0123	0,0111	0,0111	0,0146	0,0114

C

Results from evaluating ML methods and dimensionality reduction techniques with 10 fold cross-validation

C.1 Linear regression

Table C.1: Measures from cross-validation for linear regression.

Reduction:	**none, 18 features**	**PCA, 5 comp.**	**PCA, 12 comp.**	**PCA, 18 comp.**
Pred. err., μ:	0,0107	0,0153	0,0126	0,0123
Pred. err., σ:	0,0009	0,0014	0,0011	0,0011
Sensitivity, μ:	0,8157	0,6747	0,7313	0,7443
Specificity, μ:	0,9982	1	0,9996	0,9996
Sensitivity, σ:	0,1351	0,248805	0,2350	0,2059
Specificity, σ:	0,0018	0	0,0007	0,0007

Reduction:	**none, 18 features**	**PLS, 2 comp.**	**PLS, 4 comp.**	**PLS, 18 comp.**
Pred. err., μ:	0,0107	0,0144	0,0123	0,0122
Pred. err., σ:	0,0009	0,0013	0,0011	0,0011
Sensitivity, μ:	0,8157	0,6915	0,7414	0,7443
Specificity, μ:	0,9982	1	0,9997	0,9996
Sensitivity, σ:	0,1351	0,2414	0,2033	0,2059
Specificity, σ:	0,0018	0	0,0007	0,0007

C.2 Logistic regression

Table C.2: Measures from cross-validation for logistic regression.

Reduction:	**none, 18 features**	**PCA, 5 comp.**	**PCA, 12 comp.**	**PCA, 18 comp.**
Pred. err., μ:	0,0107	0,0111	0,0103	0,0107
Pred. err., σ:	0,0009	0,0011	0,0009	0,0009
Sensitivity, μ:	0,8157	0,7863	0,8162	0,8157
Specificity, μ:	0,9982	0,9993	0,9987	0,9982
Sensitivity, σ:	0,1351	0,1727	0,1358	0,1351
Specificity, σ:	0,0018	0,0008	0,0011	0,0018

Reduction:	**none, 18 features**	**PLS, 2 comp.**	**PLS, 4 comp.**	**PLS, 18 comp.**
Pred. err., μ:	0,0107	0,0122	0,0112	0,0107
Pred. err., σ:	0,0009	0,0011	0,0009	0,0009
Sensitivity, μ:	0,8157	0,7698	0,8122	0,8157
Specificity, μ:	0,9982	0,9986	0,9977	0,9982
Sensitivity, σ:	0,1351	0,2002	0,1448	0,1351
Specificity, σ:	0,0018	0,0039	0,0027	0,0018

C.3 Quadratic discriminant analysis

Table C.3: Measures from cross-validation for quadratic discriminant analysis.

Reduction:	**none, 18 features**	**PCA, 4 comp.**	**PCA, 10 comp.**	**PCA, 18 comp.**
Pred. err., μ:	0,0435	0,0532	0,0419	0,0435
Pred. err., σ:	0,0028	0,0033	0,0022	0,0028
Sensitivity, μ:	0,8718	0,8425	0,8587	0,8718
Specificity, μ:	0,9607	0,9519	0,9630	0,9607
Sensitivity, σ:	0,1145	0,1394	0,1243	0,1145
Specificity, σ:	0,0295	0,0339	0,0220	0,0294

Reduction:	**none, 18 features**	**PLS, 2 comp.**	**PLS, 4 comp.**	**PLS, 18 comp.**
Pred. err., μ:	0,0435	0,0640	0,0506	0,0435
Pred. err., σ:	0,0028	0,0029	0,0027	0,0028
Sensitivity, μ:	0,8718	0,8653	0,8691	0,8718
Specificity, μ:	0,9607	0,9394	0,9535	0,9607
Sensitivity, σ:	0,1145	0,1298	0,1253	0,1145
Specificity, σ:	0,0295	0,02992	0,02806	0,0295

C.4 KNN

Table C.4: Measures from cross-validation for KNN, no reduction performed on feature spaces.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0150	0,0100	0,0112	0,0096	0,0086	0,0090	0,0096
Pred. err., σ:	0,0006	0,0007	0,0007	0,0007	0,0008	0,0008	0,0009
Sensitivity, μ:	0,8867	0,8689	0,8562	0,8542	0,8529	0,8367	0,8175
Specificity, μ:	0,9904	0,9964	0,9958	0,9972	0,9985	0,9989	0,9992
Sensitivity, σ:	0,0908	0,1077	0,1104	0,1123	0,1206	0,1334	0,1390
Specificity, σ:	0,0062	0,0038	0,0041	0,0039	0,0026	0,0022	0,0017

Table C.5: Measures from cross-validation for KNN, 6 PCA components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0178	0,0113	0,0127	0,0108	0,0097	0,0097	0,0096
Pred. err., σ:	0,0008	0,0008	0,0009	0,0009	0,0009	0,0009	0,0008
Sensitivity, μ:	0,8617	0,8549	0,8369	0,8453	0,8592	0,8487	0,8334
Specificity, μ:	0,9887	0,9958	0,9951	0,9967	0,9971	0,9977	0,9983
Sensitivity, σ:	0,1055	0,1047	0,1243	0,1114	0,1064	0,1135	0,1306
Specificity, σ:	0,0091	0,0055	0,0066	0,0054	0,0053	0,0041	0,0034

Table C.6: Measures from cross-validation for KNN, 10 PCA components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0153	0,0104	0,0111	0,0096	0,0094	0,0095	0,0096
Pred. err., σ:	0,0006	0,0007	0,0007	0,0007	0,0008	0,0008	0,0008
Sensitivity, μ:	0,8780	0,8627	0,8468	0,8576	0,8566	0,8431	0,8252
Specificity, μ:	0,9905	0,9962	0,9963	0,9973	0,9976	0,9980	0,9986
Sensitivity, σ:	0,0943	0,1026	0,1155	0,1090	0,1088	0,1227	0,1341
Specificity, σ:	0,0067	0,0040	0,0044	0,0037	0,0043	0,0039	0,0028

Table C.7: Measures from cross-validation for KNN, all PCA components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0150	0,0100	0,0112	0,0096	0,0086	0,0090	0,0096
Pred. err., σ:	0,0006	0,0007	0,0007	0,0007	0,0008	0,0008	0,0009
Sensitivity, μ:	0,8867	0,8689	0,8562	0,8542	0,8529	0,8367	0,8175
Specificity, μ:	0,9904	0,9964	0,9958	0,9972	0,9985	0,9989	0,9992
Sensitivity, σ:	0,0908	0,1077	0,1104	0,1123	0,1206	0,1334	0,1390
Specificity, σ:	0,0062	0,0038	0,0041	0,0039	0,0026	0,0022	0,0017

Table C.8: Measures from cross-validation for KNN, 2 PLS components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0318	0,0186	0,0204	0,0165	0,0147	0,0136	0,0138
Pred. err., σ:	0,0017	0,0012	0,0013	0,0013	0,0011	0,0011	0,0011
Sensitivity, μ:	0,7695	0,7467	0,7565	0,7542	0,7751	0,7724	0,7625
Specificity, μ:	0,9790	0,9934	0,9915	0,9954	0,9958	0,9971	0,9975
Sensitivity, σ:	0,1648	0,1831	0,1720	0,1806	0,1726	0,1736	0,1792
Specificity, σ:	0,0203	0,0080	0,0114	0,0069	0,0076	0,0056	0,0050

Table C.9: Measures from cross-validation for KNN, 6 PLS components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0164	0,0117	0,0118	0,0112	0,0097	0,0102	0,0099
Pred. err., σ:	0,0006	0,0008	0,0007	0,0008	0,0008	0,0009	0,0009
Sensitivity, μ:	0,8517	0,8291	0,8224	0,8157	0,8361	0,8210	0,8289
Specificity, μ:	0,9905	0,9966	0,9965	0,9975	0,9982	0,9984	0,9985
Sensitivity, σ:	0,1189	0,1299	0,1357	0,1362	0,1243	0,1338	0,1332
Specificity, σ:	0,0069	0,0036	0,0036	0,0034	0,0033	0,0029	0,0024

Table C.10: Measures from cross-validation for KNN, all PLS components.

Neighbors:	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 10$	$k = 15$	$k = 30$
Pred. err., μ:	0,0170	0,0125	0,0141	0,0127	0,0107	0,0182	0,0234
Pred. err., σ:	0,0008	0,0010	0,0011	0,0011	0,0009	0,0018	0,0022
Sensitivity, μ:	0,8255	0,8093	0,7911	0,7870	0,8114	0,6327	0,5194
Specificity, μ:	0,9916	0,9971	0,9963	0,9979	0,9985	0,9999	0,9999
Sensitivity, σ:	0,1526	0,1509	0,1726	0,1643	0,1487	0,2689	0,3364
Specificity, σ:	0,0062	0,0025	0,0033	0,0021	0,0016	0,0003	0,0002

C.5 SVM

C.5.1 Linear kernel

Table C.11: Measures from cross-validation on SVM with trivial kernel, no reduction performed on feature space.

Box constraint:	$b = 0{,}1$	$b = 1$	$b = 10$	$b = 100$	$b = 1000$
Pred. err., μ:	0,0107	0,0107	0,0107	0,0110	0,0163
Pred. err., σ:	0,0009	0,0009	0,0009	0,0010	0,0011
Sensitivity, μ:	0,8172	0,8188	0,8188	0,8134	0,8066
Specificity, μ:	0,9982	0,9981	0,9981	0,9981	0,9927
Sensitivity, σ:	0,1374	0,1353	0,1353	0,1388	0,1429
Specificity, σ:	0,0017	0,0019	0,0019	0,0019	0,0081

Table C.12: Measures from cross-validation on SVM with trivial kernel, 4 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0115	0,0114	0,0114	0,0110	0,0308
Pred. err., σ:	0,0011	0,0010	0,0010	0,0010	0,0040
Sensitivity, μ:	0,7783	0,7793	0,7793	0,7926	0,8060
Specificity, μ:	0,9993	0,9993	0,9993	0,9992	0,9782
Sensitivity, σ:	0,1718	0,1711	0,1711	0,1556	0,1701
Specificity, σ:	0,0009	0,0009	0,0009	0,0009	0,0433

Table C.13: Measures from cross-validation on SVM with trivial kernel, 10 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0103	0,0101	0,0100	0,0099	0,0183
Pred. err., σ:	0,0009	0,0009	0,0009	0,0009	0,0014
Sensitivity, μ:	0,8160	0,8196	0,8204	0,8218	0,8148
Specificity, μ:	0,9987	0,9987	0,9987	0,9986	0,9906
Sensitivity, σ:	0,1397	0,1359	0,1358	0,1390	0,1387
Specificity, σ:	0,0010	0,0010	0,0010	0,0014	0,0142

Table C.14: Measures from cross-validation on SVM with trivial kernel, 3 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0346	0,0340	0,0122	0,0104	0,0103
Pred. err., σ:	0,0033	0,0033	0,0011	0,0009	0,0009
Sensitivity, μ:	0,3000	0,3165	0,7457	0,8043	0,8081
Specificity, μ:	1,0000	1,0000	0,9997	0,9989	0,9987
Sensitivity, σ:	0,4830	0,4725	0,1990	0,1454	0,1441
Specificity, σ:	0,0000	0,0000	0,0005	0,0014	0,0015

Table C.15: Measures from cross-validation on SVM with trivial kernel, 6 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0346	0,0339	0,0117	0,0101	0,0105
Pred. err., σ:	0,0033	0,0033	0,0011	0,0009	0,0008
Sensitivity, μ:	0,3000	0,3195	0,7585	0,8220	0,8248
Specificity, μ:	1,0000	1,0000	0,9995	0,9984	0,9979
Sensitivity, σ:	0,4830	0,4708	0,1950	0,1388	0,1387
Specificity, σ:	0,0000	0,0000	0,0009	0,0013	0,0019

C.5.2 RBF kernel

Table C.16: Measures from cross-validation on SVM with RBF kernel, no reduction performed on feature space.

Box constraint:	$b = 0{,}1$	$b = 1$	$b = 10$	$b = 100$	$b = 1000$
Pred. err., μ:	0,0288	0,0189	0,0203	0,0232	0,0238
Pred. err., σ:	0,0027	0,0016	0,0016	0,0017	0,0017
Sensitivity, μ:	0,4222	0,7033	0,7372	0,7400	0,7328
Specificity, μ:	0,9995	0,9949	0,9919	0,9890	0,9888
Sensitivity, σ:	0,4073	0,2359	0,2034	0,2013	0,2067
Specificity, σ:	0,0016	0,0122	0,0140	0,0147	0,0139

Table C.17: Measures from cross-validation on SVM with RBF kernel, 4 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0223	0,0211	0,0240	0,0260	0,0297
Pred. err., σ:	0,0020	0,0020	0,0021	0,0021	0,0021
Sensitivity, μ:	0,6305	0,6635	0,6525	0,6427	0,6481
Specificity, μ:	0,9954	0,9945	0,9920	0,9906	0,9870
Sensitivity, σ:	0,2667	0,2451	0,2528	0,2543	0,2510
Specificity, σ:	0,0131	0,0156	0,0174	0,0173	0,0168

Table C.18: Measures from cross-validation on SVM with RBF kernel, 10 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0270	0,0206	0,0252	0,0281	0,0308
Pred. err., σ:	0,0024	0,0018	0,0019	0,0017	0,0018
Sensitivity, μ:	0,4928	0,6849	0,6795	0,6739	0,6570
Specificity, μ:	0,9977	0,9941	0,9896	0,9873	0,9855
Sensitivity, σ:	0,3601	0,2340	0,2375	0,2376	0,2513
Specificity, σ:	0,0065	0,0146	0,0171	0,0154	0,0158

Table C.19: Measures from cross-validation on SVM with RBF kernel, 3 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0346	0,0346	0,0342	0,0272	0,0277
Pred. err., σ:	0,0033	0,0033	0,0033	0,0025	0,0025
Sensitivity, μ:	0,3000	0,3000	0,3913	0,5484	0,5367
Specificity, μ:	1,0000	1,0000	0,9971	0,9949	0,9950
Sensitivity, σ:	0,4830	0,4830	0,4367	0,3422	0,3508
Specificity, σ:	0,0000	0,0000	0,0087	0,0140	0,0142

Table C.20: Measures from cross-validation on SVM with RBF kernel, 6 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0346	0,0346	0,0342	0,0272	0,0277
Pred. err., σ:	0,0033	0,0033	0,0033	0,0025	0,0025
Sensitivity, μ:	0,3000	0,3000	0,3913	0,5484	0,5367
Specificity, μ:	1,0000	1,0000	0,9971	0,9949	0,9950
Sensitivity, σ:	0,4830	0,4830	0,4367	0,3422	0,3508
Specificity, σ:	0,0000	0,0000	0,0087	0,0140	0,0142

C.5.3 Polynomial kernel, order 2

Table C.21: Measures from cross-validation on SVM with second order polynomial kernel, no reduction performed on feature space.

Box constraint:	$b = 0{,}1$	$b = 1$	$b = 10$	$b = 100$	$b = 1000$
Pred. err., μ:	0.0114	0.0107	0.0132	0.0185	0.1283
Pred. err., σ:	0.0010	0.0009	0.0010	0.0010	0.0179
Sensitivity, μ:	0.8376	0.8659	0.8556	0.8292	0.8251
Specificity, μ:	0.9961	0.9955	0.9938	0.9895	0.8761
Sensitivity, σ:	0.1311	0.1029	0.1142	0.1295	0.1308
Specificity, σ:	0.0054	0.0053	0.0056	0.0069	0.1836

Table C.22: Measures from cross-validation on SVM with second order polynomial kernel, 4 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0,0115	0,0114	0,0145	0,1627	0,2820
Pred. err., σ:	0,0011	0,0011	0,0011	0,0282	0,0351
Sensitivity, μ:	0,8014	0,8004	0,7837	0,5391	0,5698
Specificity, μ:	0,9984	0,9984	0,9961	0,8561	0,7327
Sensitivity, σ:	0,1463	0,1467	0,1668	0,4007	0,3823
Specificity, σ:	0,0033	0,0034	0,0050	0,2889	0,3593

Table C.23: Measures from cross-validation on SVM with second order polynomial kernel, 10 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0115	0.0115	0.0122	0.0621	0.1707
Pred. err., σ:	0.0010	0.0011	0.0011	0.0078	0.0219
Sensitivity, μ:	0.8184	0.8203	0.8111	0.6023	0.6293
Specificity, μ:	0.8184	0.8203	0.8111	0.6023	0.6293
Sensitivity, σ:	0.1415	0.1486	0.1465	0.3237	0.3505
Specificity, σ:	0.0051	0.0054	0.0058	0.0804	0.2243

Table C.24: Measures from cross-validation on SVM with second order polynomial kernel, 2 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0346	0.0239	0.0127	0.0114	0.0112
Pred. err., σ:	0.0033	0.0024	0.0012	0.0011	0.0010
Sensitivity, μ:	0.3000	0.5145	0.7344	0.7775	0.7839
Specificity, μ:	1.0000	1.0000	0.9997	0.9993	0.9992
Sensitivity, σ:	0.4830	0.3499	0.2180	0.1715	0.1687
Specificity, σ:	0.0000	0.0000	0.0007	0.0016	0.0019

Table C.25: Measures from cross-validation on SVM with second order polynomial kernel, 4 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0346	0.0218	0.0108	0.0105	0.0107
Pred. err., σ:	0.0033	0.0023	0.0010	0.0009	0.0009
Sensitivity, μ:	0.3000	0.5664	0.7858	0.8083	0.8108
Specificity, μ:	1.0000	1.0000	0.9992	0.9985	0.9983
Sensitivity, σ:	0.4830	0.3231	0.1619	0.1425	0.1429
Specificity, σ:	0.0000	0.0000	0.0012	0.0015	0.0019

C.5.4 Polynomial kernel, order 3

Table C.26: Measures from cross-validation on SVM with third order polynomial kernel, no reduction performed on feature space.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0120	0.0131	0.0171	0.0175	0.0182
Pred. err., σ:	0.0008	0.0009	0.0010	0.0012	0.0011
Sensitivity, μ:	0.8671	0.8770	0.8623	0.8667	0.8654
Specificity, μ:	0.9939	0.9924	0.9888	0.9880	0.9873
Sensitivity, σ:	0.1069	0.0979	0.1144	0.1103	0.1109
Specificity, σ:	0.0041	0.0062	0.0078	0.0107	0.0097

Table C.27: Measures from cross-validation on SVM with third order polynomial kernel, 4 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0108	0.0315	0.0476	0.0791	0.5246
Pred. err., σ:	0.0009	0.0013	0.0039	0.0046	0.0385
Sensitivity, μ:	0.8292	0.6382	0.6459	0.7819	0.7522
Specificity, μ:	0.9971	0.9821	0.9686	0.9278	0.4672
Sensitivity, σ:	0.1389	0.3355	0.3041	0.2073	0.2295
Specificity, σ:	0.0051	0.0109	0.0244	0.0440	0.3984

Table C.28: Measures from cross-validation on SVM with third order polynomial kernel, 10 PCA components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0143	0.0166	0.0567	0.1250	0.3809
Pred. err., σ:	0.0010	0.0011	0.0078	0.0148	0.0261
Sensitivity, μ:	0.8179	0.8594	0.8600	0.8660	0.8767
Specificity, μ:	0.9940	0.9899	0.9478	0.8792	0.6115
Sensitivity, σ:	0.1620	0.1173	0.1104	0.1051	0.1144
Specificity, σ:	0.0059	0.0092	0.0782	0.1546	0.2683

Table C.29: Measures from cross-validation on SVM with third order polynomial kernel, 3 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0346	0.0158	0.0107	0.0106	0.0106
Pred. err., σ:	0.0033	0.0015	0.0010	0.0009	0.0009
Sensitivity, μ:	0.3000	0.6671	0.7893	0.8080	0.8150
Specificity, μ:	1.0000	1.0000	0.9994	0.9986	0.9983
Sensitivity, σ:	0.4830	0.2609	0.1575	0.1411	0.1394
Specificity, σ:	0.0000	0.0000	0.0008	0.0016	0.0017

Table C.30: Measures from cross-validation on SVM with third order polynomial kernel, 6 PLS components.

Box constraint:	0,1	1	10	100	1000
Pred. err., μ:	0.0346	0.0155	0.0106	0.0107	0.0105
Pred. err., σ:	0.0033	0.0015	0.0010	0.0009	0.0009
Sensitivity, μ:	0.3000	0.6754	0.7920	0.8053	0.8092
Specificity, μ:	1.0000	1.0000	0.9992	0.9985	0.9986
Sensitivity, σ:	0.4830	0.2512	0.1535	0.1457	0.1450
Specificity, σ:	0.0000	0.0000	0.0012	0.0015	0.0016

C.5.5 Random forest

Table C.31: Measures from cross-validation on Random Forest, no reduction performed on feature space.

B:	**5**	**10**	**15**	**20**	**25**	**30**
Pred. err., μ:	0,0103	0,0103	0,0102	0,0101	0,0096	0,0096
Pred. err., σ:	0,0008	0,0008	0,0008	0,0009	0,0008	0,0008
Sensitivity, μ:	0,8424	0,8392	0,8514	0,8465	0,8495	0,8590
Specificity, μ:	0,9971	0,9971	0,9964	0,9971	0,9973	0,9969
Sensitivity, σ:	0,1256	0,1330	0,1241	0,1203	0,1193	0,1129
Specificity, σ:	0,0059	0,0058	0,0071	0,0064	0,0056	0,0068

Table C.32: Measures from cross-validation on Random Forest, 4 PCA components.

B:	**5**	**10**	**15**	**20**	**25**	**30**
Pred. err., μ:	0,0125	0,0117	0,0116	0,0116	0,0109	0,0114
Pred. err., σ:	0,0009	0,0010	0,0009	0,0010	0,0009	0,0009
Sensitivity, μ:	0,8180	0,8264	0,8331	0,8352	0,8389	0,8303
Specificity, μ:	0,9959	0,9964	0,9963	0,9962	0,9966	0,9965
Sensitivity, σ:	0,1311	0,1228	0,1195	0,1201	0,1159	0,1223
Specificity, σ:	0,0063	0,0066	0,0062	0,0069	0,0065	0,0065

Table C.33: Measures from cross-validation on Random Forest, 10 PCA components.

B:	**5**	**10**	**15**	**20**	**25**	**30**
Pred. err., μ:	0,0107	0,0107	0,0106	0,0104	0,0096	0,0103
Pred. err., σ:	0,0008	0,0009	0,0009	0,0008	0,0008	0,0008
Sensitivity, μ:	0,8457	0,8330	0,8439	0,8414	0,8517	0,8467
Specificity, μ:	0,9966	0,9972	0,9967	0,9971	0,9972	0,9968
Sensitivity, σ:	0,1136	0,1221	0,1156	0,1172	0,1137	0,1159
Specificity, σ:	0,0049	0,0049	0,0064	0,0056	0,0053	0,0061

Table C.34: Measures from cross-validation on Random Forest, 2 PLS components.

B:	**5**	**10**	**15**	**20**	**25**	**30**
Pred. err., μ:	0,0206	0,0195	0,0195	0,0194	0,0186	0,0183
Pred. err., σ:	0,0013	0,0013	0,0013	0,0013	0,0013	0,0012
Sensitivity, μ:	0,7496	0,7385	0,7377	0,7370	0,7392	0,7424
Specificity, μ:	0,9912	0,9928	0,9928	0,9931	0,9937	0,9939
Sensitivity, σ:	0,1871	0,1918	0,1966	0,1953	0,1909	0,1895
Specificity, σ:	0,0110	0,0095	0,0092	0,0100	0,0092	0,0088

Table C.35: Measures from cross-validation on Random Forest, 4 PLS components.

B:	**5**	**10**	**15**	**20**	**25**	**30**
Pred. err., μ:	0,0129	0,0107	0,0116	0,0106	0,0105	0,0106
Pred. err., σ:	0,0009	0,0007	0,0008	0,0007	0,0008	0,0007
Sensitivity, μ:	0,8181	0,8314	0,8277	0,8368	0,8325	0,8346
Specificity, μ:	0,9954	0,9971	0,9965	0,9970	0,9974	0,9971
Sensitivity, σ:	0,1435	0,1322	0,1315	0,1270	0,1260	0,1273
Specificity, σ:	0,0062	0,0042	0,0044	0,0042	0,0037	0,0040

D

Empirical studies for decision layer

The different decision layer parameters tested and their results for SVM with radial basis function and QDA are shown in Tables D.1 and D.2, respectively.

Table D.1: Results from adding a decision layer to the cross-validation results when using SVM with radial basis function kernel. The window lengths are presented in the order: median filter window length, median filter threshold and minimum decision threshold. For each tested parameter set the results contain the mean sensitivity, specificity and prediction error after the median filter in the first sublayer and the true positives, false positives and false negatives after both sublayers.

Test:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Thresholds [s]:	1 1 1	1 1 5	5 4 5	5 4 10	10 8 10	20 10 10	20 10 10	1 1 10	20 10 15
Sensitivity:	0,5672	0,5672	0,4555	0,4555	0,6500	0,3486	0,6102	0,5672	0,6102
Specificity:	0,9944	0,9944	0,9992	0,9992	0,9982	0,9999	0,9995	0,9944	0,9995
Error:	0,0165	0,0165	0,0163	0,0163	0,0114	0,0193	0,0126	0,0165	0,0126
TP:	11	11	10	10	11	10	11	10	10
FP:	36	2	1	0	1	0	1	0	0
FN:	0	0	1	1	0	1	0	1	1

10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
10 5 15	25 10 15	5 4 15	10 4 15	10 6 15	25 10 20	25 15 20	10 5 20	15 5 20
0,6500	0,7448	0,4555	0,7171	0,5613	0,7448	0,4593	0,6500	0,7865
0,9982	0,9989	0,9992	0,9961	0,9991	0,9989	0,9999	0,9982	0,9959
0,0114	0,0054	0,0163	0,0100	0,0122	0,0054	0,0182	0,0114	0,0068
11	11	10	11	10	11	9	9	11
0	1	0	0	0	0	0	0	1
0	0	1	0	1	0	2	2	0

19	**20**	**21**
10 5 25	15 5 25	20 5 25
0,6500	0,7865	0,8422
0,9982	0,9959	0,9934
0,0114	0,0068	0,0068
8	11	11
0	0	2
3	0	0

Table D.2: Results from adding a decision layer to the cross-validation results when using QDA. The window lengths are presented in the order: median filter window length, median filter threshold and minimum decision threshold. For each tested parameter set the results contain the mean sensitivity, specificity and prediction error after the median filter in the first sublayer and the true positives, false positives and false negatives after both sublayers.

Test:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Thresholds [s]:	1 1 1	1 1 5	5 4 5	5 4 10	10 8 10	20 10 10	20 10 10	1 1 10	20 10 15
Sensitivity:	0,8215	0,8215	0,7915	0,7915	0,8446	0,7495	0,8446	0,8215	0,8446
Specificity:	0,9588	0,9588	0,9768	0,9768	0,9656	0,9892	0,9801	0,9588	0,9801
Error:	0,0217	0,0217	0,0100	0,0100	0,0084	0,0088	0,0064	0,0217	0,0064
TP:	11	11	11	11	11	11	11	11	11
FP:	93	40	21	8	19	7	17	9	11
FN:	0	0	0	0	0	0	0	0	0

10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
10 5 15	25 10 15	5 4 15	10 4 15	10 6 15	25 10 20	25 15 20	10 5 20	15 5 20
0,8446	0,9017	0,7915	0,8710	0,8176	0,9017	0,7796	0,8446	0,9017
0,9656	0,9727	0,9768	0,9506	0,9767	0,9727	0,9885	0,9656	0,9473
0,0084	0,0044	0,0100	0,0147	0,0082	0,0044	0,0084	0,0084	0,0088
11	11	11	11	11	11	11	11	11
12	17	5	14	8	12	3	5	13
0	0	0	0	0	0	0	0	0

19	**20**	**21**
10 5 25	15 5 25	20 5 25
0,8446	0,9017	0,9488
0,9656	0,9473	0,9302
0,0084	0,0088	0,0098
11	11	11
2	7	13
0	0	0