Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Marie Pauline Strømme Kristiansen

Classification and Interpretation of Cerebral Palsy-related movements by means of Multivariate Analysis

Master's thesis in Cybernetics and Robotics
Supervisor: Frank Westad
June 2019

NTNU
Norwegian University of
Science and Technology

Marie Pauline Strømme Kristiansen

Classification and Interpretation of Cerebral Palsy-related movements by means of Multivariate Analysis

Master's thesis in Cybernetics and Robotics
Supervisor: Frank Westad
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Preface

This thesis is the finishing work for my Master's degree in Engineering Cybernetics at NTNU. The work has been done during the spring of 2019 and is based on research and initial work done in the specialization project in TTK4550 during the autumn of 2018. In the specialization project I researched methods for data mining, classification, feature extraction and feature selection, and the work done this spring is based on what I learned during this project. The scope of the thesis has been narrowed since working with the specialization project, but the data set and overall problem is the same. The data analysis in this thesis has mainly been done using the software The Unscrambler®by Camo Analytics. The licence to the software was provided by NTNU. I have also used MATLAB and a licence to MATLAB, Simulink and 65 of its toolboxes is available for all students at NTNU.

The work in this thesis is based on a data set recorded as part of a large research project lead by Lars Adde and his research group at St. Olav's University Hospital in Trondheim. Lars Adde, Espen Ihlen and PhD Research Fellow Daniel Groos have been available for help throughout the project. This thesis would not be the same without their questions, suggestions and guidelines, so thank you all for letting me work on this project and giving me an educational year! Hopefully some of the results found in this thesis can be used further in the project, if not only for knowing what not to test again.

My supervisor, Frank Westad, has been a big contributor for this thesis. Nearly all methods were tested based on suggestions from him, and he has always had an answer to any problem that has occurred during the year. He also proposed the initial modeling procedure on the dataset, which eventually led to the models in this thesis. Thank you for always being available for questions no matter if you're in Trondheim, Abidjan, Berlin or anywhere else in the world!

Sammendrag

Cerebral parese er en samlebetegnelse på tilstander med endret motorisk funksjon forårsaket av en permanent skade på hjernen. Påvirkningen denne skaden har på motorisk funksjon er veldig individuelt og er derfor vanskelig å diagnostisere tidlig. Studier har funnet en sammenheng mellom bevegelsesmønster hos spedbarn $9-13$ uker etter termindato og CP [1]. Forskningsgruppen InMotion på St. Olavs Hospital i Trondheim jobber med å utvikle digitale verktøy for å benytte denne sammenhenhengen i klinisk bruk. De har samlet inn 378 videoer av spedbarn i høy risiko for CP når de er $9-13$ uker og utviklet en algoritme for å spore bevegelsene til spedbarna i videoene. Bevegelsene gis som tidsserier og disse tidsseriene brukes i denne masteroppgaven for å utvikle klassifiseringsfunksjoner som skal skille mellom CP og ikke-CP.

For å regne ut egenskaper, eller features, for tidsseriene brukes et rammeverk kalt *hctsa*. Det regner ut og sammenlikner tusenvis av egenskaper og har innebygde funksjoner for seleksjon. Med dette rammeverket velges det ut 7606 egenskaper. Videre utvikles en PLS-DA modell som brukes for videre seleksjon. Egenskapene med høyest regresjonskoeffisienter i PLS-DA modellen velges ut og gir et nytt sett med 416 egenskaper. Denne prosessen gjentas og med en ny PLS-Da modell står 105 egenskaper igjen. Det er disse som brukes som input til klassifiseringsfunksjonene.

Det er utviklet modeller av typen PLS-DA, PCA, SVM og Random Forest. Det er brukt både test sett validering og kryssvalidering, og alle modellene viser at noen av subjektene med CP er enkle å skille fra de andre og noen er vanskeligere. Det er mange underliggende faktorer som ikke er tatt hensyn til i denne oppgaven, som andre nevrologiske abnormaliteter, kjønn, alder og størrelse, som kan være årsaken til denne forskjellen i subjektene. Det kan også være tilfeldig, men det bør ses videre på. SVM og Random Forest modellene er testet med både PCA skårer og det oprinnelige egenskapssettet som input. De viser at dersom man bruker egenskapssettet direkte så vil modellene være sårbare for overfitting. Å bruke PCA skårer som input gjør at modellene blir mer generelle og sannsynligvis vil være bedre for klassifisering av nye subjekter.

Ingen av klassifiseringsmodellene utviklet i denne masteroppgaven er gode nok for klinisk bruk, men de understreker flere utfordringer med denne typen klassifiseringsproblemer og kan brukes som base for videre utvikling. PLS-DA modellen med kryssvalidering har en spesifisitet på 100% og sensitivitet på 45.24%, noe som gjør det litt bedre enn de andre modellene men den lave sensitiviteten gjør at den fortsatt ikke god nok for klinisk bruk.

Summary

Cerebral Palsy is a syndrome of motor impairment that results from a lesion occuring in the developing brain. The affect this lesion has on motoric function is individual, and early diagnosis is therefore a challenge. Studies have found a connection between movement qualitites at infants $9 - 13$ weeks post term and CP [1]. The research group InMotion at St. Olavs Hospital in Trondheim is working on developing digital tools for using this connection in diagnosis. They have a video collection of 378 infants at high risk of CP, and have developed an algorithm that tracks the movements in the videos. The movements are given as time series and these time series are used in this master thesis to develop classification functions to separate between CP and not CP.

To compute features for the time series, a framework called *hctsa* is used. It computes thousands of features and compares them for feature selection. With this framework, 7606 features are extracted. Further a PLS-DA model is used for selection. The features with the highest absolute value for the weighted regression coefficients are selected and a new feature set is made of the 416 selected features. This process is repeated, and with selection from a new PLS-DA model 105 features remain in the final feature set. This is the feature set that is used as input to the classification functions.

Classification models are developed of the type PLS-DA, PCA, SVM and Random Forest. Both test set validation and cross validation are used, and all the models show that some of the subjects with CP are easier to separate from the healthy than the others. There are many possible underlying sources of variation that is not accounted for in this thesis, i.e. other neurological abnormalities, gender, age and size. These may or may not influence the model, and further research should be done on this. The SVM and Random Forest models are tested on both PCA scores and the feature set as input. They show that using the feature set directly makes the models prone to overfitting. Using PCA scores as input makes the models more general and likely better for classification of new subjects.

None of the classification models developed in this thesis are good enough for clinical use, but they point out several challenges with this type of classification problems and may be used as a basis for further research. The PLS-DA model with cross validation have a specificity of 100% and sensitivity of 45.24%, which makes it better than the other models in this thesis but the sensitivity must be higher for clinical use.

Table of Contents

List of Tables

List of Figures

Abbreviations

CP	=	Cerebral Palsy
GMA	=	General Movement Assessment
FM	=	Fidgety Movements
hctsa	=	Highly Comparative Time-Series Analysis
EDA	=	Exploratory Data Analysis
PCA	=	Principal Component Analysis
SVD	=	Singular Value Decomposition
PCs	=	Principal Components
PLSR	=	Partial Least Squares Regression
PLS-DA	=	Partial Least Squares Discriminant Analysis
SVM	=	Support Vector Machine
CV	=	Cross Validation
TN	=	True Negatives
FP	=	False Positives
TP	=	True Positives
FN	=	False Negatives

Chapter 1

Introduction

1.1 Background and motivation

1.1.1 Cerebral Palsy

Cerebral Palsy (CP) is a syndrome of motor impairment that results from a lesion occurring in the developing brain. The syndrome affects everyone differently and several classification systems exists to assess the type and form of CP that an individual has. These classifications are defined according to the anatomical site of the brain lesion, clinical symptoms and signs, topographical involvement of extremities and classification of degree of muscle tone [2].

With advances in neonatal intensive care, the survival of very preterm (born $\leq$ 32 weeks of gestation) and very low-birth-weight (VLBW) (weighing $\leq$ 1500 g) children has improved considerably [3]. Cerebral Palsy has a prevalence of 2.0-3.5 per 1000 live-births [4], but multiple studies demonstrate an increasing prevalence of CP with decreasing birthweight and gestational age. In a report from Sweden, the prevalence of cerebral palsy was 6.7 per 1000 live births for children born at 32 to 36 weeks of gestation, 40.4 per 1000 live births for children born at 28 to 31 weeks of gestation, and as high as 76.6 per 1000 live births for children born before 28 weeks of gestation [5]. There was a similar increase when looking at birth weight.

Because of the different classifications of Cerebral Palsy, prediction at an early age is a challenge. About 85% of children with CP show an abnormal MRI scan, which can provide an estimate of the timing of the lesion and whether it causes a motoric impairment [2]. However, MRI scans are not optimal for children. It requires them to lie still for 30 to 60 minutes, and the hollow tube is easily considered scary.

In a clinical report from 2013 [6], the American Academy of Pediatrics write about the importance of early diagnosis of Cerebral Palsy. In the report the Academy stresses the importance of early diagnosis as a way to receive interventions that will help the child master everyday tasks, increase mobility and improve their quality of life. Early diagnosis can also address the ongoing anxiety parents have about their child's health condition [7].

1.1.2 General Movement Assessment and Fidgety Movements

After birth, infants have a spontaneous movement pattern with a writhing character. At the age of 6 to 9 weeks post term the form and character of the general movements change from the writhing type into a fidgety pattern. These fidgety movements are defined as an ongoing stream of small, circular and elegant movements of neck, trunk and limbs. Fidgety movements in a healthy infant is a transient phenomenon; they emerge gradually at 6 weeks, come to full expression between 9 and 13 weeks post term and taper off again between the age of 14 to 20 weeks post term [8].

In 1997 Prechtl et al. presented a tool to predict motoric dysfunction in infants based on their movement pattern [1]. The tool, known as General Movement Assessment (GMA), uses fidgety movements as a marker for a normal neurological outcome. In [1], the 60 infants with abnormal and absent fidgety movements included 57 infants with an abnormal outcome. 49 of the ones with abnormal FM had cerebral palsy and eight had developmental retardation or minor neurological signs. Only three were diagnosed as normal at age 2 years [1].

GMA provides a method to predict CP at an earlier age. It is non-invasive and cost-efficient compared to i.e. MRI scans [9]. A disadvantage is the subjectivity of the physicians. Prechtl et al. found this method to have a higher specificity and sensitivity than ultrasound, where their method had a specificity of 96% and sensitivity 95% and ultrasound only 83% and 80%, but this requires trained experts in the field. Training physicians in GMA is expensive and time consuming, and hence the method is not available for everyone. A computer-based tool for prediction has the advantage that it can easier be made available for clinics all over the world and make early prediction available for everyone.

1.1.3 Related work

The study of fidgety movements in relation with CP is done is several studies [9] [10] [1] [8] [11]. In some, physicians analyze the movements visually [9] [10] and in others they use sensors like pressure sensitive mats, Kinect cameras and motion sensors [12]. In 2010, Adde et. al. [11] did a feasibility study on computer-based video analysis of general movements and found it to be an objective and feasible tool for early prediction of CP in high-risk infants.

The study of movements through data analysis is an active field of research. In [13], Principal Components Analysis (PCA) and Probabilistic PCA are used for segmenting motion capture data, only unordered sets of poses are analyzed and no information about temporal dynamics is taken into account. Other examples of using PCA as a feature extraction method or preprocessing step to increase the performance of a classifier is shown in [14] [15] [16]. In [17] Hidden Markov Model's are applied to discover groupings of similar objects motions observed in a video collection. [18] uses SVMs to classify motion from a set of filtered images, [19] uses SVM for nonlinear prediction of chaotic time series, [20] uses multi-class SVM for recognition of abnormal human activity and [21] uses a SVM-based computer-aided diagnosis system for early detection of Alzheimer's disease.

1.2 Goal and hypothesis

The main goal for this project is to develop a model that can evaluate and separate movements from babies with CP from those without. The model will be trained on motion data from a video database recorded as part of a large research project driven by Lars Adde and his research group at St. Olav's Hospital in Trondheim, Norway.

The hypothesis is that fidgety movements is recognizable as repeating movements with small amplitude in the healthy babies. In this thesis several features from the motion data will be investigated, and the goal is to obtain a high accuracy as well as interpretable results that can be explained to physicians. The field of computer learning algorithms has gained large success over the previous years, and the hypothesis is that a computer based model can obtain an accuracy that is just as good, or better, as gestalt perception used by the physicians.

1.3 Outline of this work

This master thesis will analyze motion data from babies and use multivariate methods to classify whether the movements indicate CP or not. The input will be raw coordinate time series and suitable transforms thereof, and several different classification methods are tested. Dimension reduction as part of the classification process will also be investigated. The performance of the classifiers will be discussed, as well as some of the different features.

The thesis is structured as follows: Chapter 1 presents the background and motivation for the project, Chapter 2 looks into relevant theory and methods, Chapter 3 describes the dataset that is used and some challenges with this kind of dataset, Chapter 4 presents the results gained, Chapter 5 discusses the results and Section 6 presents a conclusion. Section 7 presents some ideas for future work.

This master thesis is based on initial research and testing done by me in the course 'TTK4550 - Engineering Cybernetics, Specialization Project' the fall of 2018. The course resulted in a project report which overlaps with Chapters 1, 2 and 3 of this master thesis.

Chapter 2

Theory and methods

2.1 Pre-processing of dataset

2.1.1 Scaling and normalizing

Scaling of data sets can be done in various ways. In this thesis the min-max-normalization (or min-max scaling) which scales all features to be within a given range will be used. This scales every time series within the range [0,1] where maximum movement for one sensor corresponds to 1. Min-max-normalization is computed in the following way:

$$\boldsymbol{X}_{sc} = \frac{\boldsymbol{X} - \min(\boldsymbol{X})}{\max(\boldsymbol{X}) - \min(\boldsymbol{X})} \tag{2.1}$$

2.1.2 Centering

Given a vector, X, the centered vector X_c is obtained by calculating the mean of the series and subtracting this from every entry in the original vector.

$$\boldsymbol{X}_c = \boldsymbol{X} - \text{mean}(\boldsymbol{X}) \tag{2.2}$$

2.1.3 Interpolation

Interpolation is a method of constructing new data points within a field of points. Through sampling a limited number of data points is obtained. These points represents the values of a function for a number of values for an independent variable. Interpolating is to estimate the value for the function for an intermediate value of the independent variable [22].

Linear interpolation is the simplest method of interpolation [22]. Linear interpolation takes two data points $(x_a, f(x_a))$ and $(x_b, f(x_b))$ and the interpolant $f(x)$ is given by

$$f(x) = f(x_a) + (f(x_b) - f(x_a))\,\frac{x - x_a}{x_b - x_a} \text{ at the point } (x, f(x)) \tag{2.3}$$

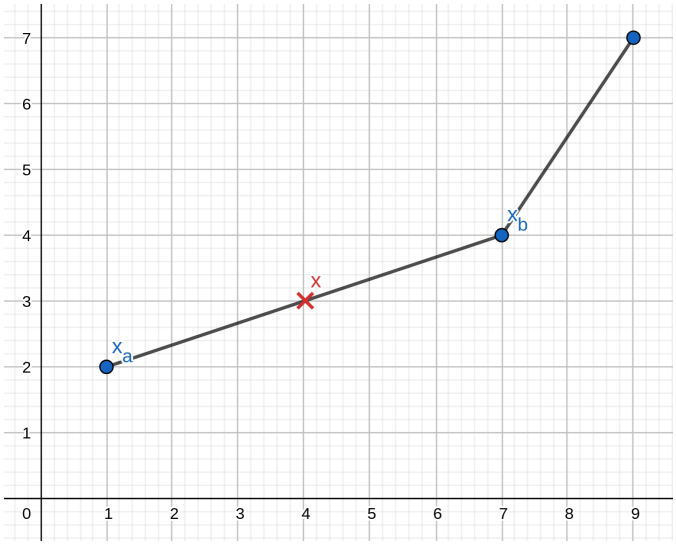


Figure 2.1: Linear interpolation

Interpolation can also be done with other interpolation functions, such as polynomial functions, a nearest neighbor approach or with a spline function.

A visual example of linear interpolation is shown in Figure 2.1. Here $x_a = 1$, $f(x_a) = 2$, $x_b = 7$ and $f(x_b) = 4$. Hence, for $x = 4$,

$$f(x) = 2 + (4 - 2)\frac{4 - 1}{7 - 1} = 2 + 2\frac{1}{2} = 3 \tag{2.4}$$

2.1.4 Hampel filter

The Hampel Filter is in the decision-based filter class, and is closely related to the median filter as it uses the local median and median absolute deviation (MAD) to detect outliers [23].

Given a sequence of data points, $x_1, x_2, \ldots, x_n$, and a sliding-window of length k, the Hampel filter calculates the local median m_i and standard deviation σ_i for each window.

$$m_i = \text{median}(x_{i-k}, x_{i-k+1}, \ldots, x_{i+k-1}, x_{i+k}) \tag{2.5}$$

$$\sigma_i = \kappa\text{median}(|x_{i-k} - m_i|, |x_{i-k+1} - m_i|, \ldots, |x_{i+k-1} - m_i|, |x_{i+k} - m_i|) \tag{2.6}$$

where $\kappa = 1.4826$. The scaling factor κ makes σ_i an unbiased estimate of the standard deviation for Gaussian data [23].

A sample x_i is declared an outlier if it is such that

$$|x_i - m_i| > n_\sigma \sigma_i \tag{2.7}$$

for a given threshold n_σ. If the point is marked as an outlier it is replaced with m_i.

The Hampel filter has the advantage that it doesn't introduce distortion into the signal. A standard median filter replaces every point with the median of the window which can lead to loss of information.

2.2 Feature Extraction and Feature Selection

In order to build a good predictor, good features are needed. Features are measurable properties or characteristics of the phenomenon being observed, and usually the term "features" is used for variables constructed from the input variables while the raw input variables are called "variables" [24]. There are thousands and thousands of features that can be calculated from a time series data set [25], and in order to build a computationally effective model one should select the best subset of them to use in developing a model. There are many benefits of feature selection: simplification of the model to make it easier to understand and interpret, reducing the measurement and storage requirements, reducing training and utilization times, defying the curse of dimensionality to improve prediction performance and enhanced generalization as it reduces overfitting [26] [24].

In this thesis feature extraction and selection is done through the framework *hctsa* and using PLSR models.

2.2.1 *hctsa*: highly comparative time series analysis

In 2017, Ben D. Fulcher and Nick S. Jones presented *hctsa*: a software tool for computing over $7'700$ time series features and a suite of analysis and visualization algorithms to automatically select useful and interpretable time series features for a given application. It includes classification algorithms and compares the computed features to find out which are most important in this classification. The comparative feature-based approach to time series classification was first introduced in [27] and then the computational framework was presented in [25].

hctsa is MATLAB-based and the framework is easily run through MATLAB. In this thesis *hctsa* is used for feature extraction and feature selection.

2.2.2 Feature extraction

The full feature set of over 7700 features in *hctsa* is produced by running 165 master operations. All of these are run with different sets of input parameters, and produces a set of outputs for each input parameter set. These master operations are divided into 11 broad categories. The categories are described in the *hctsa* documentation [28] and are as follows:

Distribution Code summarizing properties of the distribution of values in a time series (disregarding their sequence through time).

Correlation Code summarizing basic properties of how values of a time series are correlated through time.

Entropy and information theory Entropy and complexity measures for time series

Time series model fitting and forecasting Fitting time series models, and doing simple forecasting on time series.

Stationarity and step detection Quantifying how properties of a time series change over time.

Nonlinear time series analysis and fractal scaling Nonlinear time series analysis methods, including embedding dimensions and fluctuation analysis.

Fourier and wavelet transforms, periodicity measures Properties of the time series power spectrum, wavelet spectrum, and other periodicity measures.

Symbolic transformations Properties of a discrete symbolization of a time series.

Statistics from biomedical signal processing Simple time series properties derived mostly from the heart rate variability (HRV) literature.

Basic statistics Basic statistics of a time series, including measures of trend.

Others Other properties, like extreme values, visibility graphs, physics-based simulations, and dependence on pre-processing applied to a time series.

2.2.3 Feature selection

After having run the calculations for the time series, *hctsa* includes a range of processing, analysis and plotting functions to understand and interpret the results. One of these is the function "TopFeatures" which determine the features that individually best distinguish between the two groups of time series (CP and non-CP). The function compares each feature individually in terms of its ability to separate the classes. The output of this function is the mean linear classification accuracy across all operations and a list of the operations with top performance.

2.3 Exploratory Data Analysis

From an early age people are told that the easiest way to investigate a problem is to do it piece by piece. In mathematics this is done by changing one variable at a time to see how the system reacts, only this is too simple for most complex real life systems. With a large number of variables with unknown, complex relationships, Multivariate Analysis is needed. Multivariate Analysis is a way of investigating a large number of variables simultaneously to understand the relationship that may exist between them [29]. For small data sets with few variables it may be enough to present the data as disjointed graphs, but

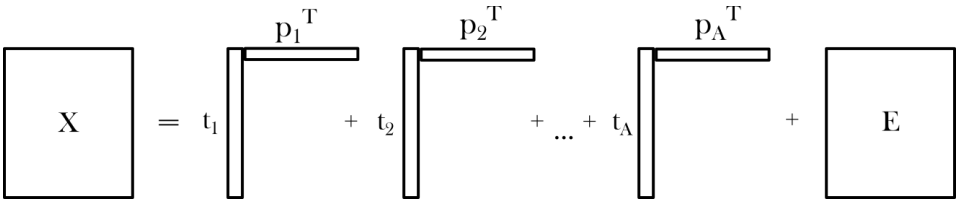


Figure 2.2: PCA decomposition

for big data sets this will be too complex and it will be very hard to find the dependencies manually.

Exploratory Data Analysis (EDA), or *data mining*, attempts to find the hidden structure in large, complex data sets. EDA finds the structure that results from the influence of all variables acting simultaneously, not just the influence of one variable. The two main methods used in EDA are cluster analysis and Principal Component Analysis.

2.3.1 Principal Component Analysis

Principal Component Analysis (PCA) is a method that analyzes the variability in a data set. It's a mathematical procedure that transforms a number of variables into a smaller number of orthogonal variables called *principal components* (PCs). PCA transforms the data using an orthogonal linear transformation onto a new coordinate system, where the coordinates are the principal components. The PCs are extracted so that the first PC explains a larger part of the total variance than the next PC, and so on. This can be visualized in the terms of the eigenvalues, often in a cumulative way.

Given a zero-mean data matrix $\boldsymbol{X}$, with n rows containing data from a new repetition of the experiment and p columns that each gives a particular feature, the PCA splits $\boldsymbol{X}$ into a structure part $\boldsymbol{M}$ and an error part $\boldsymbol{E}$.

$$\boldsymbol{X} = Structure + Noise = \boldsymbol{M} + \boldsymbol{E} \tag{2.8}$$

The structure matrix $\boldsymbol{M}$ may be regarded as a sum of contributions from different functions of the rows and columns

$$\boldsymbol{M} = f(rows) \cdot g(columns) \tag{2.9}$$

where each function can be approximated by a linear model, which together forms

$$\boldsymbol{M} = \boldsymbol{T}\boldsymbol{P}^T \tag{2.10}$$

The matrix $\boldsymbol{T}$ contains the *scores* and the matrix $\boldsymbol{P}^T$ contains the *loadings*. The decomposition is shown in Figure 2.2. The scores and loadings can be estimated in different ways, e.g. through Singular Value Decomposition (SVD) or the NIPALS algorithm [30].

A non-trivial task when using PCA is choosing the number of dimensionality, aka the number of principal components A_{opt}. A model with a high percentage of explained variance is wanted, but one does not want to include the noise in the scores and loadings. The validation methods of Section 2.5 can be used to find A_{opt}.

The NIPALS algorithm

Given the matrix $\boldsymbol{X}$, the NIPALS algorithm can be used to find the principal components with the decomposition

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T \tag{2.11}$$

where the matrix $\boldsymbol{T}$ contains the *scores* and the matrix $\boldsymbol{P}^T$ contains the *loadings*. The algorithms initializes with $a = 1$ and $\boldsymbol{X_a} = \boldsymbol{X}$, and proceeds through the following steps [31]:

1. Choose $\boldsymbol{t_a}$ as any column of $\boldsymbol{X_a}$.
2. Compute loadings $\boldsymbol{p_a} = \boldsymbol{X_a}'\boldsymbol{t_a}'/\boldsymbol{t_a}'\boldsymbol{t_a}$
3. Normalize $\boldsymbol{p_a}$ to length 1.
4. Compute scores $\boldsymbol{t_a} = \boldsymbol{X_a}\boldsymbol{p_a}/\boldsymbol{p_a}'\boldsymbol{p_a}$

Then repeat point 3 and 4 until convergence for the a^{th} principal component. Let $\boldsymbol{X_{a+1}} = \boldsymbol{X_a} - \boldsymbol{t_a}\boldsymbol{p_a}'$. Let $\lambda_a = \boldsymbol{t_a}'\boldsymbol{t}$. Increment $a = a + 1$ and repeat for the next principal component[31].

The resulting scores and loadings matrices are obtained by assembling the columns of $\boldsymbol{T}$ from the $\boldsymbol{t_a}$ and the columns of $\boldsymbol{P}$ from the vectors $\boldsymbol{p_a}$.

The implementation used in this thesis uses the NIPALS algorithm with convergence stopping criteria $||t_{old} - t|| < 1e\text{-}12$ [32].

2.4 Classification

The problem of classification is to identify which category or *class* a new observation belongs to [33]. In classification we build a function $f(\boldsymbol{X})$ that predicts the class membership Y based on the input attributes or features $\boldsymbol{X}$.

There are numerous different classification algorithms. Some examples are: Linear Discriminant Analysis, Partial Least Squares Discriminant Analysis, Support Vector Machines, Random Forests, Neural Networks and Cluster analysis (e.g. K-means).

Some multivariate methods, like PLS-DA, can handle data with numerous features where many of them describe the same underlying latent variables, while others are more prone to overfitting when this is the case. Using score vectors from PCA as input to a classifier may reduce the danger of overfitting as the dimensionality in the input becomes smaller.

2.4.1 Partial Least Squares Discriminant Analysis

Partial Least Squares Regression is a dimension reduction method, but unlike PCA described in Section 2.3.1 it is supervised. This means that the identification of the principal components is *supervised* in such a way that the new features are related to the response Y [34]. PLSR is a regression method, but can be used as a binary classification method by defining the classes as i.e. 0 and 1. When using the model to classify new samples each

sample is assigned to the class which the output value is closest. When used for classification the method is called PLS-DA. The method is popular in the field of chemometrics because of the high number for variables per sample [34], and it is useful as a method to understand which variables carry the class separating information [35].

2.4.2 Support Vector Machines

A Support Vector Machine (SVM) classifier separates a set of binary-labeled training data by computing an optimal separating hyperplane: $(w \times x) + b = 0$, to obtain the maximum margin between the two classes [36]. The function that maps the variables onto the new space is called a kernel function, and there is no theoretical tool to find the best one. There are several options, such as a linear function, polynomial function or the Radial Basis function [37]. In addition to being dependent on the choice of kernel function, SVMs are in risk of overfitting.

The implementation used in this thesis uses a Radial Basis function as kernel type, and the C value and Γ are chosen through a grid search procedure using segmented cross validation implemented in the software [38]. C and Γ are slack variables introduced to the SVM optimization problem to be able to find a feasible solution when the data is not linearly separable.

Grid search

Grid search is a method of hyper parameter tuning used for finding the optimal values for a given model. It is a systematic procedure which creates a model for every set of hyper parameters within a given range and chooses the parameter values which gives the best model.

2.4.3 Random Forests

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute. An example of a decision tree is shown in Figure 2.3. Random Forests (RF) are "forests" built from individual decision trees, where a random subset of variables is selected for each tree. It is claimed that Random Forest is unexcelled in accuracy among current algorithms, and runs efficiently on large data bases [37]. An advantage for the classification problem for this method is that it gives an estimate and visualization of which variables that are important.

Random Forests has become a popular technique for both classification, prediction, studying variable importance, variable selection and outlier detection. Examples of studies where RF have been applied and compared are explored as a survey in [39].

2.5 Validation

In order to validate that a classification model is useful for new data and evaluate how well it performs one needs validation methods. Conceptually it is distinguished between *external* and *internal* validation; external validation concerns whether it is used correct

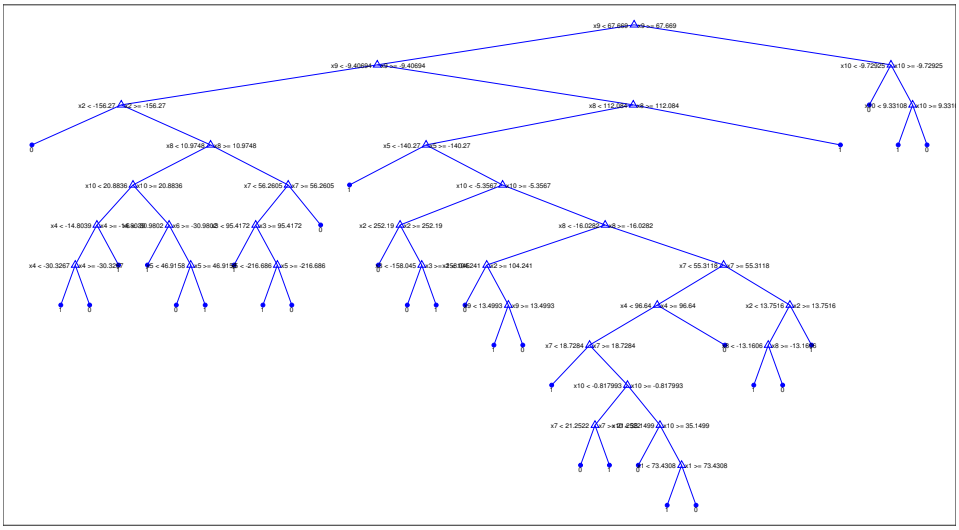


Figure 2.3: Example of a decision tree.

information in making of the model and validating that different models give the same results, while internal validation is based on numerical validation. Some internal validation methods are cross validation, test set validation and cross model validation.

2.5.1 Test set validation

For test set validation the full data set is split into two groups: test set and training set. The model is built with the training samples and then the prediction error is computed by predicting the outcome for the test samples.

A challenge with test set validation is how to split the data set. Splitting at random is not sufficient, as it will in most real applications be subgroups in the data due to underlying sources of variation. In this context this could e.g. be ethnicity of the infants or age group. One algorithm developed to split the data is the Kennard-Stone sampling algorithm (KS). It selects a subset of samples that has an uniform distribution over the predictor space, and hence tries to avoid the problem of subgroups in the two sample sets. Another algorithm is the One-Sided Selection [40].

Using test set validation with PCA scores

When using test set validation it is important to separate the samples before using any pretreatment of original variables or methods that are dependent on the data. Even though PCA is an unsupervised method, the model still depends on the data and must only be created using the training set samples. If the variance between classes is large compared to the variance within the classes, variance between the classes will influence the PCA projection and the test set samples create a bias in the model. If the variance is small it

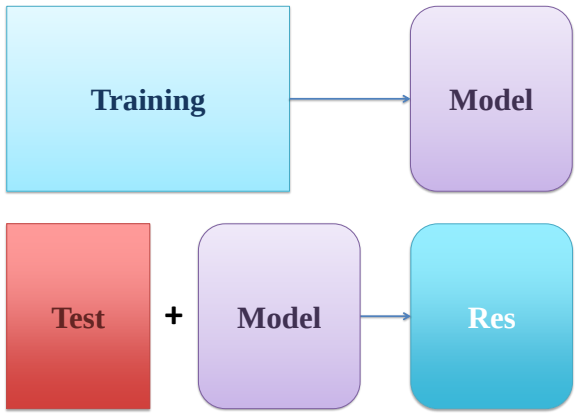


Figure 2.4: Test set validation.

wont affect the model as much, but it wont contribute to separate the classes in the PCA model either.

When using PCA scores as input to a classifier, i.e. SVM og Random Forest, the way to use test set validation in each step without inserting a bias in the PCA model is as follows:

1. Create a PCA model based only on training set samples.
2. Using the resulting PCA scores from the training samples, create a classifier.
3. For validation project the test set samples onto to created PCA model.
4. Classify the projected PCA scores from test set samples with the created classifier.

It is important to use projection onto the existing PCA model and not create a new PCA model for the test set, for the same reasons as not to create a PCA model for the full sample set.

2.5.2 Cross validation

In cross validation it is iteratively chosen a new subset of samples to be training and test set. The cross validation method used in this thesis is usually called k-fold cross validation, but will only be referred to as cross validation in this thesis for simplicity.

The procedure is as follow: pick out k samples from the calibration set and build a model with the remaining samples. Predict the outcome on the k left-out samples and calculate the residual. Put these samples back into the calibration set, and repeat the procedure until all samples have been left out at least once. Combine the prediction residuals for all iterations.

In cross validation, all samples are used in both training and testing. This avoids the problem of subgroups in the sample set It is also applicable to smaller data sets that doesn't have a sufficient number of samples to take out as an independent test set.

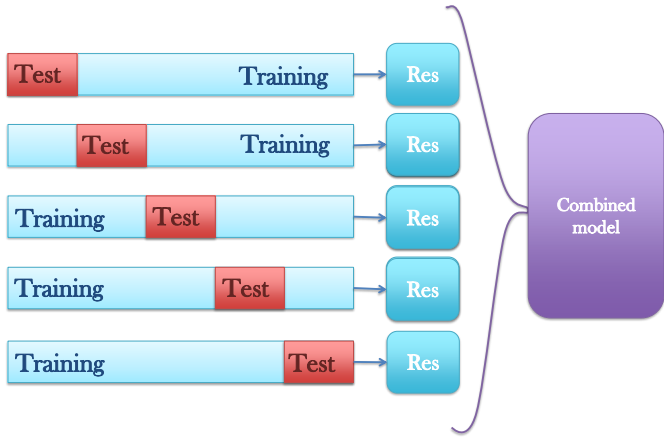


Figure 2.5: Cross-validation.

2.6 Performance metrics

A confusion matrix is a way to present the prediction result from a classifier. Each row of the matrix represents the instances in the predicted class while each column represents the instances in the actual class.

There are several useful metrics to be calculated from a confusion matrix. For simplicity the instances that are correctly classified as healthy are abbreviated TN (true negatives), instances that are healthy but are classified with CP are abbreviated FP (false positives), instances that are correctly classified with CP are abbreviated TP (true positives) and the instances with CP that are classified as healthy are abbreviated FN (false negatives).

2.6.1 Sensitivity

Sensitivity, also called recall, hit rate, or true positive rate, is defined as

$$\frac{TP}{TP + FN} \tag{2.12}$$

2.6.2 Specificity

Specificity, also called selectivity or true negative rate, is defined as

$$\frac{TN}{TN + FP} \tag{2.13}$$

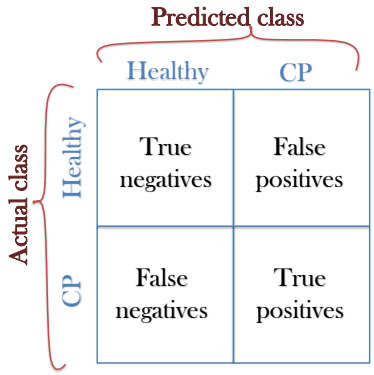


Figure 2.6: Confusion matrix.

2.6.3 Positive Predictive Value

Positive Predictive Value (PPV), also called precision, is defined as

$$\frac{TP}{TP + FP} \tag{2.14}$$

2.6.4 Negative Predictive Value

Negative Predictive Value (NPV) is defined as

$$\frac{TN}{TN + FN} \tag{2.15}$$

2.6.5 Accuracy

Accuracy is used in many ways, i.e. to describe the closeness of a measurement to the true value [41]. In the case of binary classification it is defined as

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{2.16}$$

Chapter 3

Dataset and challenges

3.1 The dataset

The data in this thesis comes from a database of 378 standardized video recordings at St. Olav's University Hospital of infants at risk of neurological dysfunctions from Norway, USA and India. There are certain prerequisites that must be fulfilled for the videos to be usable, for example that all infants must not be crying, not be disturbed by external influence and not be hungry.

The videos are analyzed using a tracker algorithm developed in a master thesis in 2018 done for the InMotion project [42]. The output of this tracker algorithm is time series for the x- and y-coordinates for 7 points on the body of the infants in the videos. The tracking is done on the 378 videos which gives 378×14 time series. These coordinate time series are the input for the model developed through this project. An overview of this data collecting system is shown in Figure 3.1. For each subject the CP status is known. The CP status is given as 1 and 0, where 1 means that the subject has CP while 0 means it does not.

As mentioned in Section 1, there are several classification systems used to assess the type and form of CP that a subject has. One of these classifications is the subtype, specified by the SCPE Collaborative Group [43]. This system divides CP into Spastic bilateral, Spastic unilateral, Dyskinetic and Ataxic. The subtype of each subject is known.

The coordinates are normalized so that the coordinate system is the same for every frame. This is shown in Figure 3.2. Here it is shown that origo is set it the top left corner, and max value for both x and y is 1.

A graphic representation of the data structure is shown in Figure 3.3, where the $I = 378$, $K = 14$ and the time J might be varied as the optimal size of a time segment is not known. Plots of the data from two of the subjects are shown in Figure 3.4 and 3.5. Every subject has 14 time series of the same length but the videos are of different length and hence the time series varies in length from subject to subject. In order to have equal length on the time series, only the first 60 seconds are used for each subject. The framerate is also standardized. The original data's framerate varies between $24 - 32$ frames per second. To

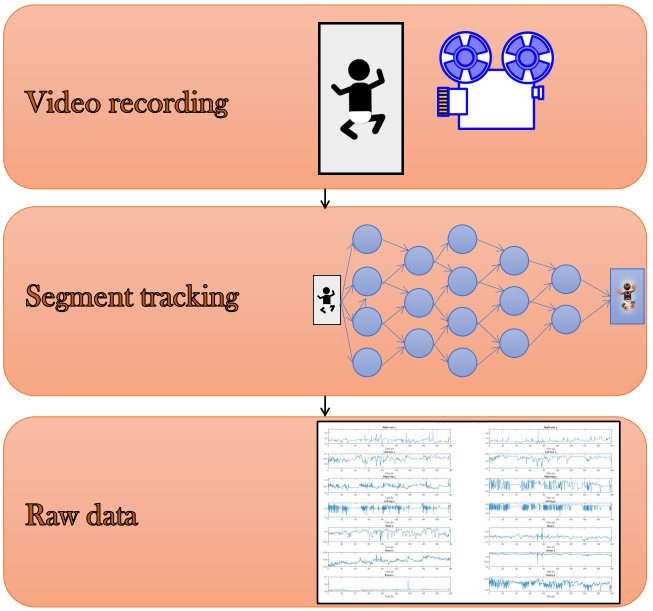


Figure 3.1: Overview of collection of raw data.

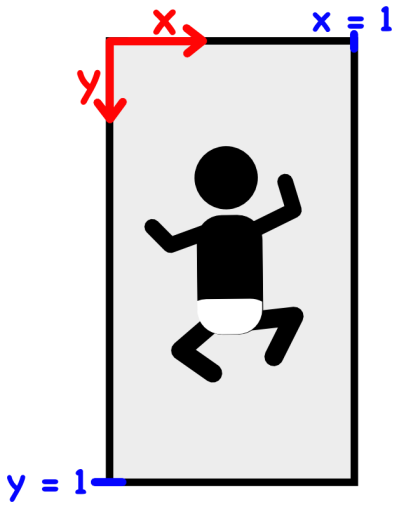


Figure 3.2: Coordinate system on video frames.

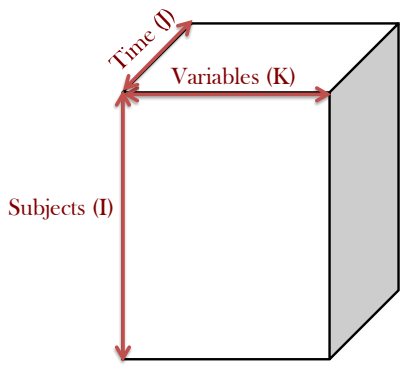


Figure 3.3: Data structure.

have the same amount of datapoints for every subject, 24 frames per second is chosen and all the time series are interpolated as to match this framerate.

3.1.1 Sensitive Personal data

Sensitive data is data that must be protected against unwanted disclosure [44]. The General Data Protection Regulation (GDPR) defines personal data as: "'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;" [45]. When handling and dealing with sensitive data, special attention should be given to collecting, processing, handling and storing data throughout the research process [44]. All steps in the research process should focus on keeping the sensitive data secure and out of reach from the public.

Video data is sensitive personal data. Combined with data such as gender, date of birth, size and location of filming the data in this research project must be considered sensitive and be secured at every step.

There are several ways to ensure the security of the data in a research project, and the way it has been done in this project to ensure that the data still can be used in this master thesis is through pseudonymization. Pseudonymization substitutes the identity of the data subject in such a way that additional information is required to re-identify the data subject [44]. This differs from anonymization where all person-related data that could allow backtracking has been purged, but since the personal identifiers are stored somewhere else it is still considered a secure approach [44].

To do this master thesis only access to the time series, an ID number and the CP subtype for each subject has been given. This ensures that no sensitive personal data is

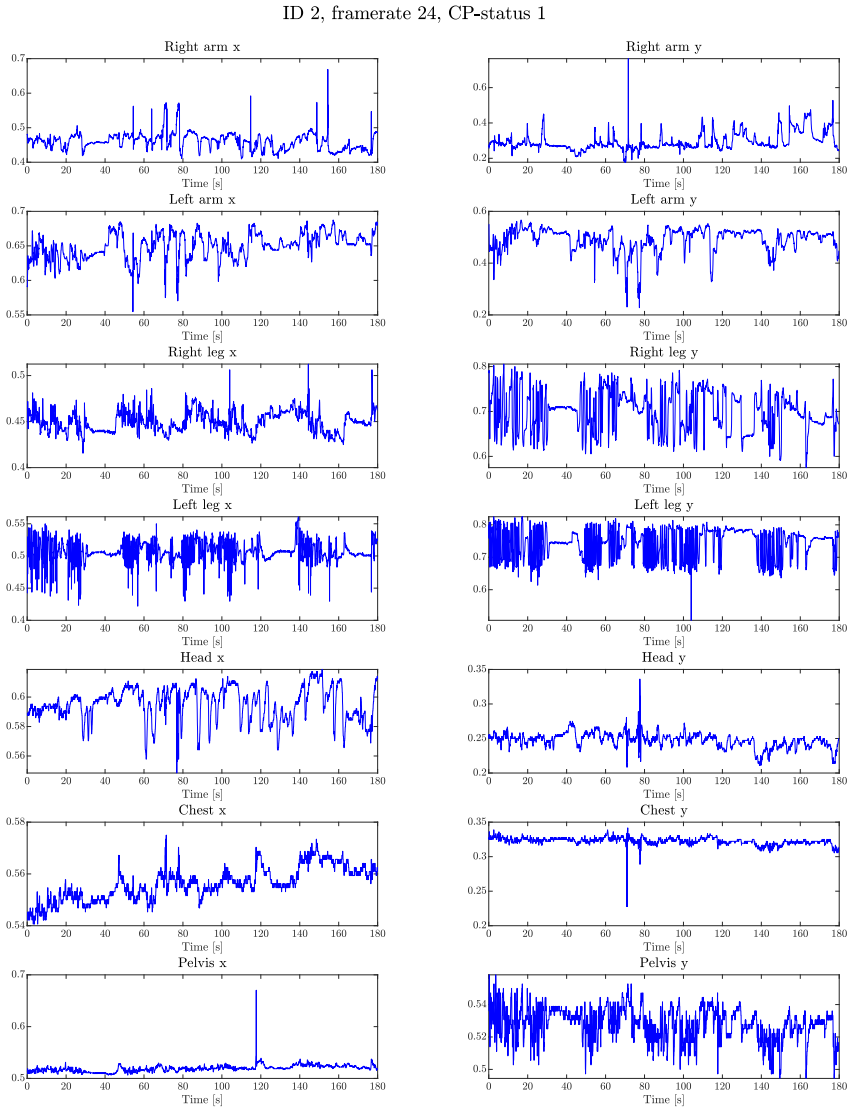


Figure 3.4: Raw data from a subject with CP.

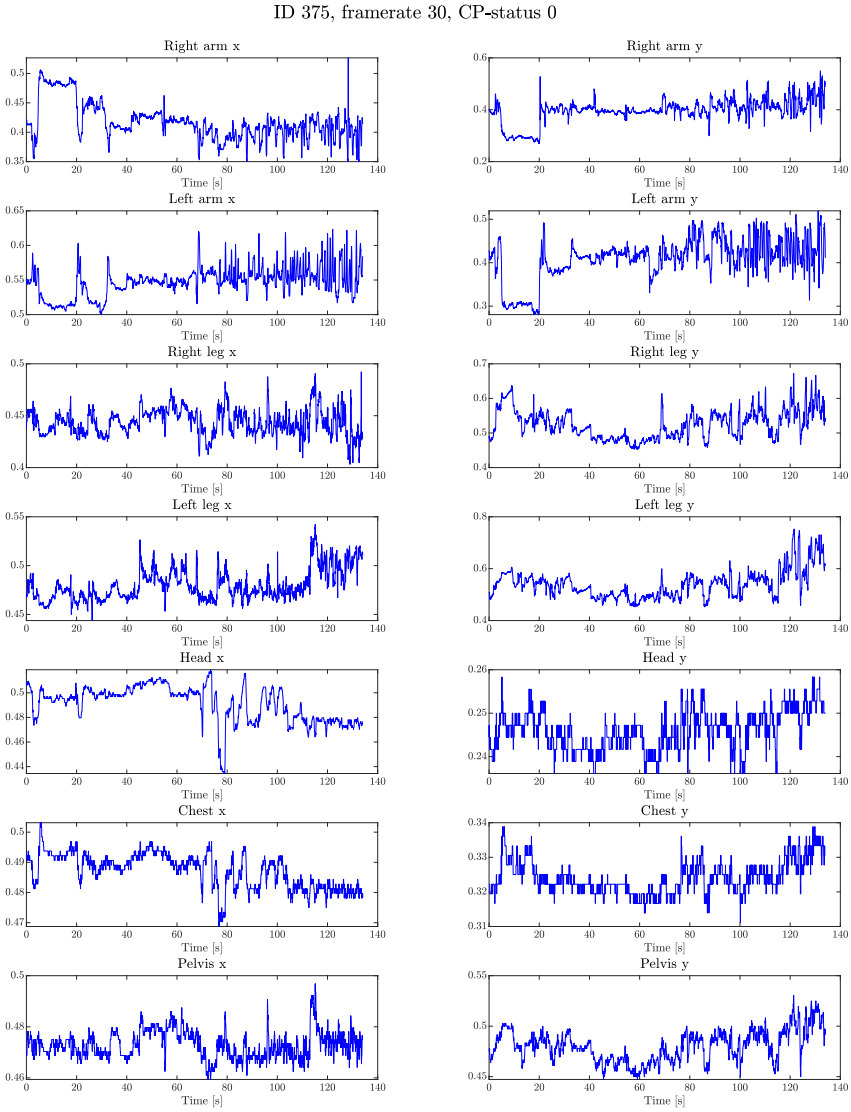
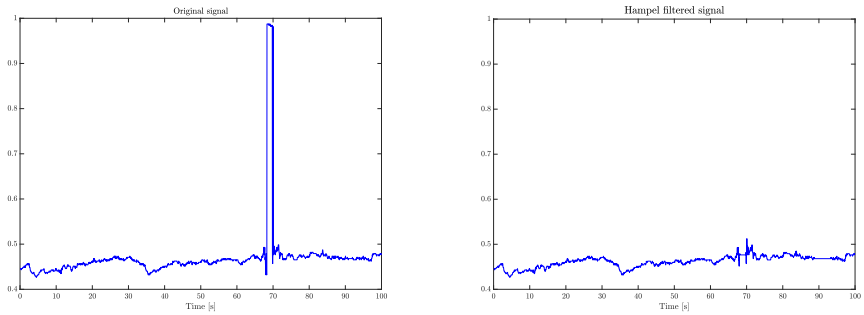


Figure 3.5: Raw data from a healthy subject.



(a) Coordinate data with typical error spike.

(b) Filtered data.

Figure 3.6: Coordinate data with error in tracking. Original signal and filtered.

available. However, it also limits the interpretation for some of the methods. Since no information about the size of each subject is given, a normalization can't be done based on size. The location of filming and nationality of the subjects can't be accounted for either. When only the time series are available and not the original videos, verifying the results with visual inspection must be done on the coordinate series directly.

3.1.2 Errors from tracker

The time series are made through analyzing the videos frame by frame. There is no comparison of the tracking between frames, so if a segment is labeled wrong the model does not correct it. The score for each labeling is not part of the dataset, but some of the errors are still quite visible. One example is shown in Figure 3.6a. Here it is shown that the tracked point jumps from ~ 0.45 to ~ 0.98 from one frame to the next. One frame corresponds to ~ 90cm in real life, which means that a change in amplitude of 0.53 means that the baby would have to move ~ 47 cm in $1/24$ second.

The pinpointing and removal of these points can be done in several ways. One way is to use a Hampel filter which will remove spikes by comparing with the points around it. The effect of a Hampel filter is shown in Figure 3.6b. The use of this filter is done to remove the effects rapid changes have on the calculated features.

3.1.3 Class imbalance

The main issue with this dataset is that even though data from 378 subjects are available, only 42 of these have Cerebral Palsy. This makes up only $\frac{1}{9}$ of the subjects, meaning that if every subject is classified as healthy the accuracy will be as high as 88.9%. This problem is called the accuracy paradox [46], as accuracy is not a good metric for the predictive model. One way to account for this problem is to use other measurements such as sensitivity, specificity or precision. Another approach is under- or oversampling. Undersampling is to even the balance by deleting instances from the over-represented class and over-sampling is to copy instances from the under-represented class.

In this thesis the performance metrics sensitivity and specificity will be emphasized.

The small number of subjects with CP also makes dividing the set into training and test set a challenge. In [40] this problem is addressed and it is stated that noisy or otherwise unreliable examples from the majority class can overwhelm the minority class. In this thesis the majority class is the healthy subjects. The hypothesis is that these infants have fidgety movements while the ones with CP does not. When the task is to locate specific movements the training samples of the majority class will be unreliable when they do not appear to be showing fidgety movements. If this is the case for some of the healthy subjects they will influence the model greater than the few training samples from the minority class because of their outnumbering. Selecting samples into training and test sets should account for the variations inside each class and making the two sets fair, meaning that all factors should be represented in both sets. This is however not an easy task in research where there are so many factors to account for and so few samples.

As training and test sets in the InMotion project has not yet been divided accounting for all these factors, they are divided at random in this thesis.

3.2 Visualization of dataset

As mentioned in Section 3.1.1 the videos have not been available for this thesis. Hence the time series must be inspected directly, and three different visualizations have been used to get an initial understanding of the dataset.

3.2.1 Time series

The simplest way to investigate the time series is to plot each time series in a 2D-plot. One example is in figure 3.7. Here the x- and y-coordinates are plotted against time.

With such plots, visual inspection of similarities and variations is hard. Even differentiating between rapid movement and errors in the tracking algorithm is not trivial, for example in Figure 3.7 in the plot for the y-movement. After approximately 71 seconds there is a jump in the time series which may look like a tracker error, but just by visual inspection it may also be a rapid movement of the arm.

Another approach to plotting the time series directly is to plot the different series on top of each other to easier compare two series. One example is given in Figure 3.8. Here the x-data for both legs are plotted. With such plots it is easier to investigate where the two legs move together and how the movements correspond with each other. In this plot it is shown that after approximately 47 seconds the two legs cross and then cross back again. It is also shown from this plot that the legs are nearly motionless at the same interval (0 s - 40 s).

3.2.2 Video of moving subject

A problem with plotting the original time series in 2D-plots is that the human brain is not trained at adding two coordinate series together as motion. Because of this, a tool has been developed to help on this task. Using MATLAB's animatedline-function all the time series

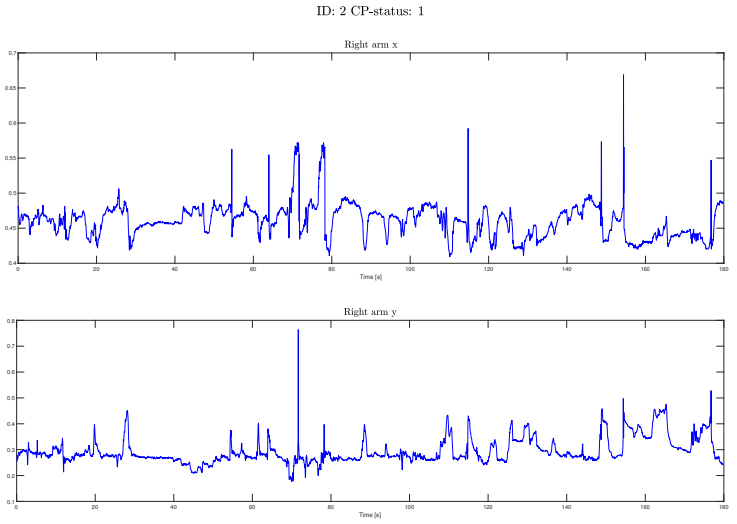


Figure 3.7: Plot of time series from right arm from subject 2

from every sensor are added together into one plot. The plot is then updated for each time step and made into a video using MATLAB's VideoWriter.

Some frames from one such video is shown in Figure 3.9. Here the lines are not deleted after each frame, seen by all the accumulated lines in the last frame, but this is optional when making videos and are only added here to better show the development throughout the video.

Showing the movements of an infant through video instead of as 14 different time series is much more intuitive for the human brain. This way it is more intuitively to see how the body parts move together or differs from each other. Spotting the error points is also easier as it will be a bigger difference between the error points and the rapid movements. Rapid movements in one arm will affect the rest of the body in some way, while an error from the tracker wont influence the other sensors. An example of a tracker error in the time series plot and how it is shown in a video is shown in Figure 3.10. Here a rapid change for the chest point in the y-direction is shown in a plot of the original time series. In the frames from the video (Figure 3.10b-3.10e) it can be seen that the chest point is originally in Figure 3.10b, but then jumps considerably upwards in Figure 3.10c, then down again in 3.10d and again to an unnaturally high point in 3.10e. Note that in the time series plot in 3.10a the coordinates for y is plotted inverse of what is up and down in 3.10b-3.10e, as shown is Figure 3.2.

3.2.3 Heatmaps of movement

A bivariate histogram is a histogram which divides data into bins in 2 dimensions. It is used as a visualization tool to create surface plots of data. It can also show data as heatmaps, where the values in each bin are shown as colors. Given a X- and a Y-vector

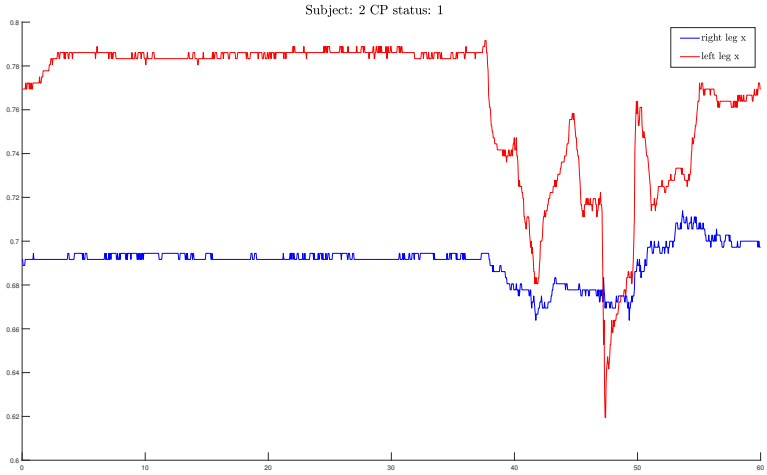
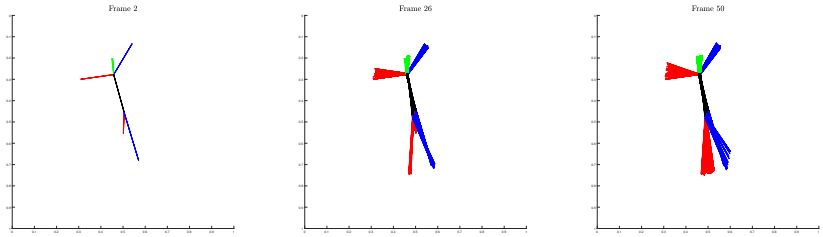


Figure 3.8: x-coordinate time series for both legs for subject 2.

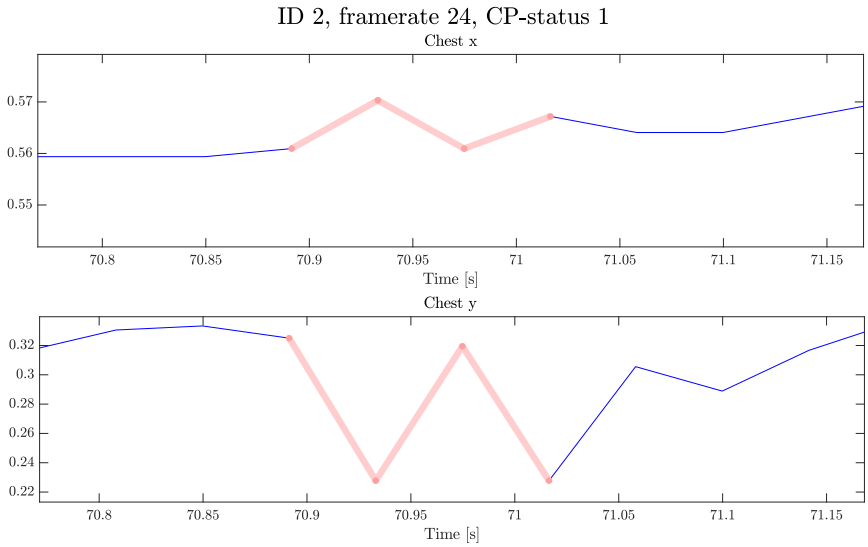


(a) time $= 0$ s **(b)** time $= 1$s **(c)** time $= 2$ s

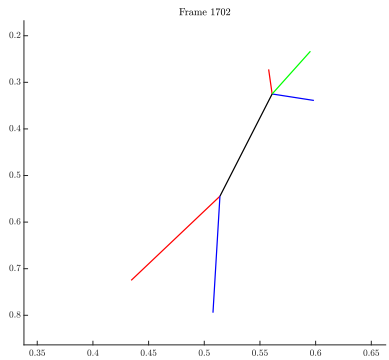
Figure 3.9: Frames from one video at different time steps.

the 2D-space is divided into N bins of size $n \times n$. Before making a heatmap the coordinates are normalized with min-max normalization. All points in the time dimension (x, y) are added to the histogram, and the resulting matrix contains the cumulative number of observations in all of the bins. Hence the heatmap shows movement over each rectangular bin. Heatmaps from 6 different subjects is shown in Figure 3.11.

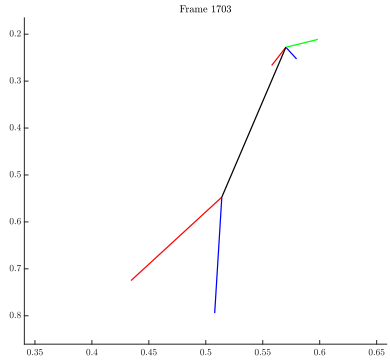
Heatmaps are a good tool to visualize movements in space when the time dimension is of lower importance. In this project it is not important whether a specific movement occurs at $t = 3s$ or $t = 45s$. In Figure 3.11 the movement of the different subjects is quite different. In Figure 3.11a it is shown that Subject 1 does not move much and the different sensors are easy to tell apart. In Figure 3.11b however, Subject 2 is moving a lot all over the space and the sensors can not be distinguished as easy. Visual inspection does not show any obvious differences in the two classes of subjects, but this has not been analyzed further.



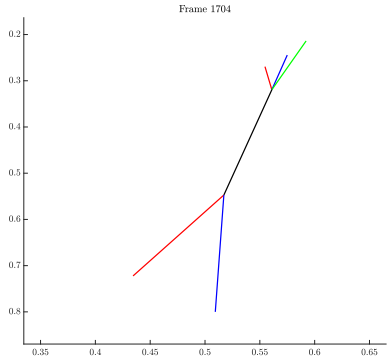
(a) Original time series for chest sensor. Marked points are the frames below.



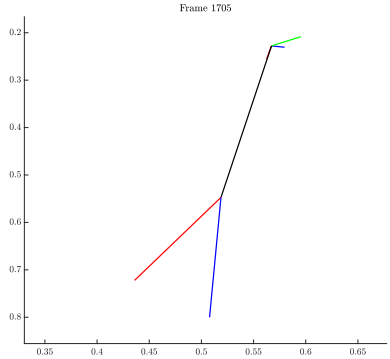
(b) Frame 1



(c) Frame 2

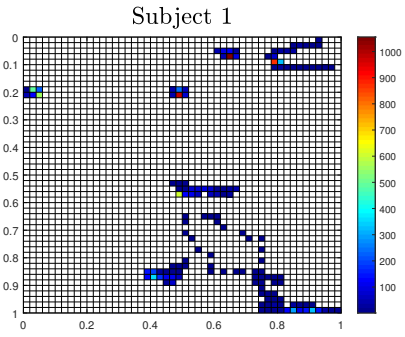


(d) Frame 3

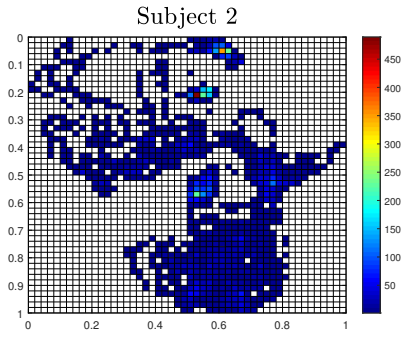


(e) Frame 4

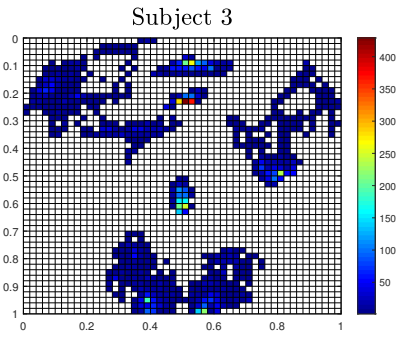
Figure 3.10: Tracker error in chest coordinate. Visualized frame by frame.



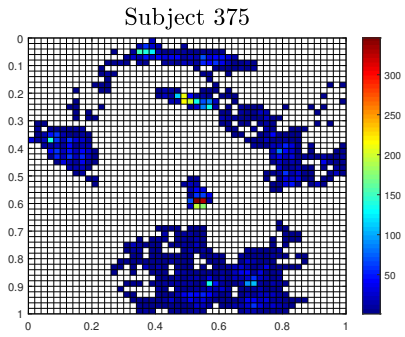
(a) Subject with CP



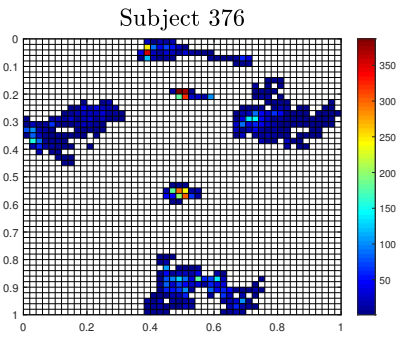
(b) Subject with CP



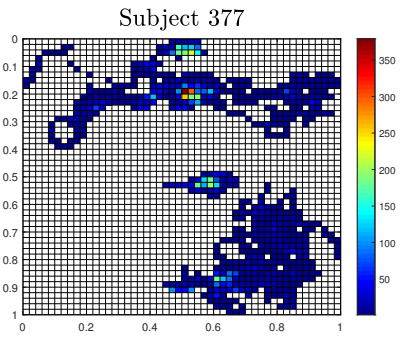
(c) Subject with CP



(d) Subject without CP



(e) Subject without CP



(f) Subject without CP

Figure 3.11: Heatmaps from different subjects. Created with 50 bins in each direction.

Chapter 4

Results

In this chapter the results from the feature selection done through three steps is presented. Then the results from classification of the selected feature set are given for several different classification methods. The input and validation method for each classification model is given.

4.1 Feature selection

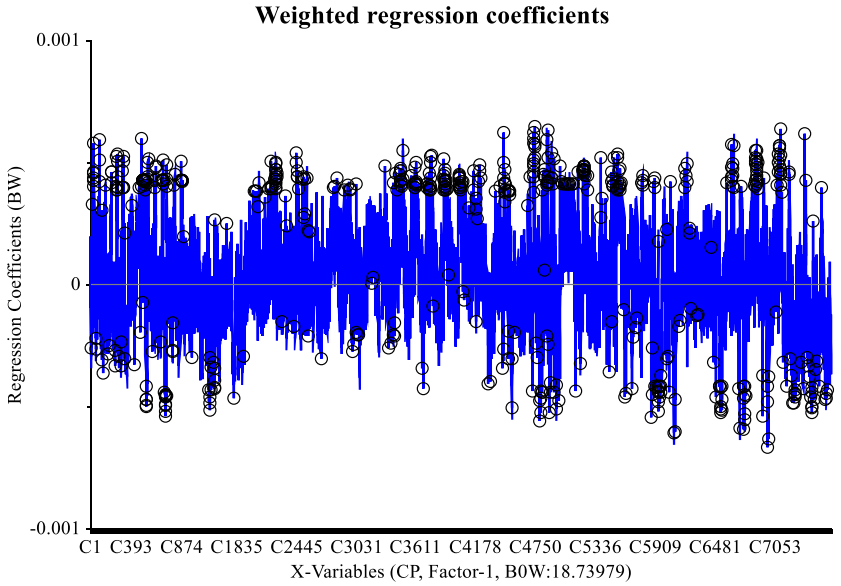
4.1.1 hctsa

Feature extraction and selection is done using the framework *hctsa* as described in Section 2.2.2 and 2.2.3. All 7700 features are calculated, and the top 20 features are chosen by "TopFeatures" as described in Section 2.2.3. The mean linear classifier performance across these 543 features is 54.66%.

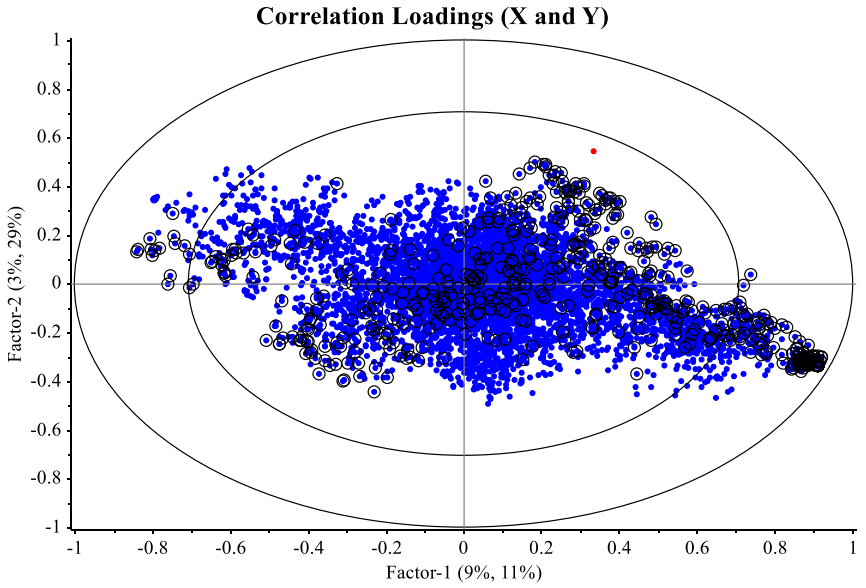
Using the list of the 20 top features, a new subset of feature calculations are made based on the master operations from the top list and all variations of these. The new feature set consists of 543 features, which are all variations of 12 master operations.

4.1.2 Manual feature selection

A PLS-DA model is made based on the feature set of 543 features per sensor. With 14 sensors this gives a total of $543 \times 14 = 7606$ feature columns. The regression coefficients are used to select the features which influences the model the most The variables are selected by going through the regression coefficient plots and the loadings weights for the first six factors and selecting the ones with the highest absolute value. A plot of the weighted regression coefficients showing the selected features is shown in in Figure 4.1a. A total of 416 features were chosen. In Figure 4.1a all selected features are marked. The selected features are also shown in the Loadings plot in Figure 4.1b. There will be many combinations of $400 - 500$ features with the same classification ability. Nevertheless, any subset of $400+$ variables are assumed to represent the various types of features.



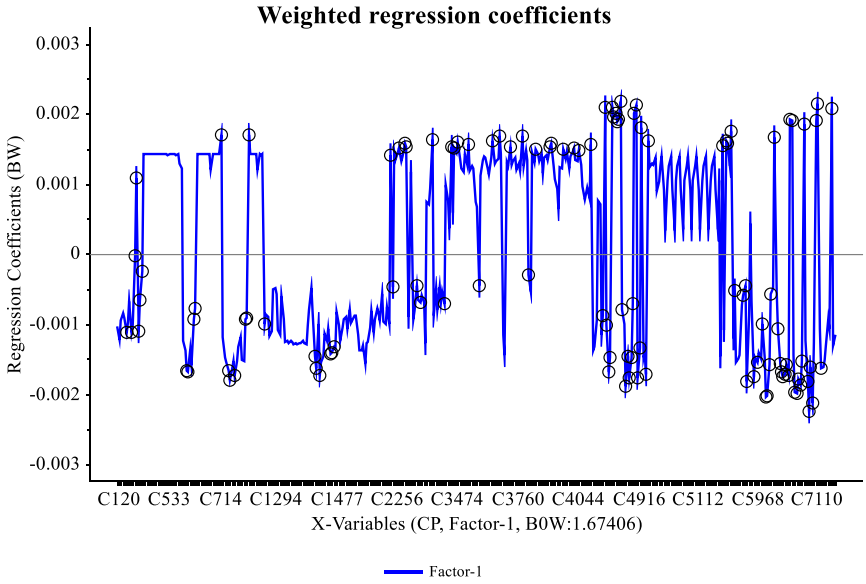
(a) Weighted regression coefficients with selected features marked.



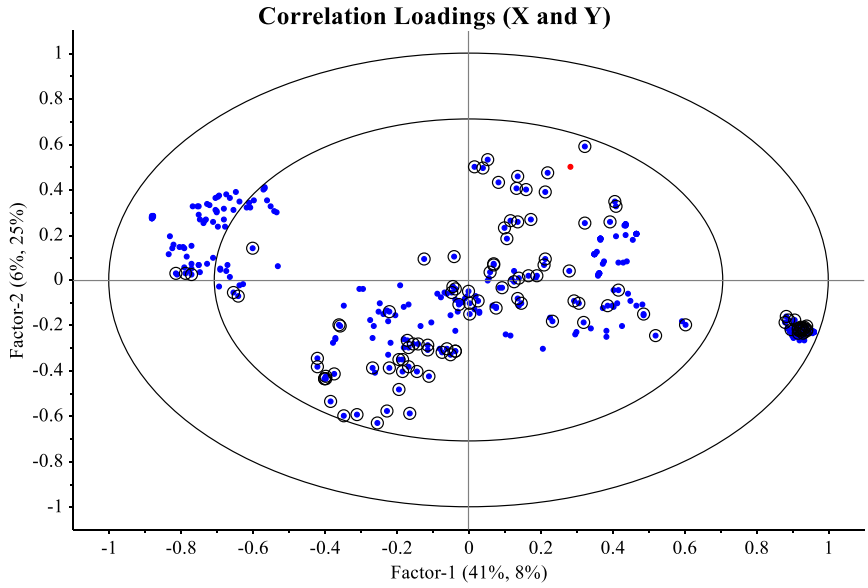
(b) Loadings plot for Factor 1 and 2 with selected features marked.

Figure 4.1: PLS-DA model on the feature subset from *hctsa*.

A new PLS-DA model is made on these 416 features. The procedure of choosing the features with highest absolute value in the weighted regression coefficient plot is repeated, and this time 105 features were chosen. The selected subset of features is shown in Figure 4.2. This is the feature subset that will be used further in the thesis. An overview of the 105 selected features by sensor and feature keyword is shown in Figure 4.3 and 4.4. Figure 4.3 divides the features into groups based on what sensor time series the feature is computed from, and Figure 4.4 groups the features by the master operation categories presented in Section 2.2.2.



(a) Weighted regression coefficients with selected features marked.



(b) Loadings plot with selected features marked.

Figure 4.2: PLS-DA model on the 416 selected features.

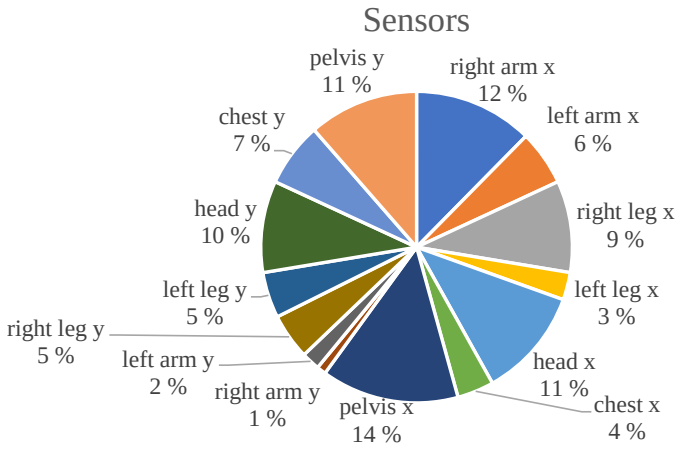


Figure 4.3: Percentage of features from sensor in feature subset.

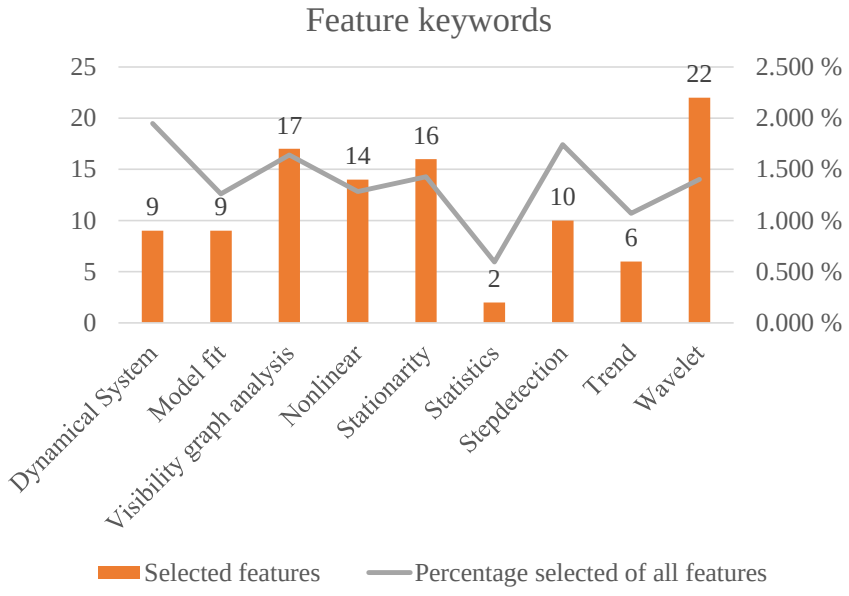


Figure 4.4: Feature keywords of features in feature subset.

4.2 PLS-DA

For classification with PLS-DA, two different validation approaches are tested:

1. Test set validation
2. Cross-validation

4.2.1 Validation with test set

For the model with test set validation, $\frac{1}{3}$ of the samples are randomly chosen as test set. Of these 126 samples, 14 have CP. The PLS-DA model is trained on the remaining $\frac{2}{3}$, and the best results are obtained when using 15 factors. The resulting score plot for the model is shown in Figure 4.5. The cut-off of 0.5 was applied to assign the samples. The confusion matrix and performance metrics for the model is shown in Table 4.1.

Table 4.1: Confusion matrix and performance metrics in training step of PLS-DA model with test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	224	0
	CP	1	27

Accuracy	99.6 %
NPV	99.56 %
PPV	100 %
Sensitivity	96.43 %
Specificity	100 %

Using the model on the test set to validate the model gives the predictions shown in Figure 4.6b. The confusion matrix and performance metrics for the test set is given in Table 4.2.

Table 4.2: Confusion matrix and performance metrics for PLS-DA model on test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	107	5
	CP	13	1

Accuracy	85.71 %
NPV	89.170 %
PPV	16.67 %
Sensitivity	7.14 %
Specificity	95.54 %

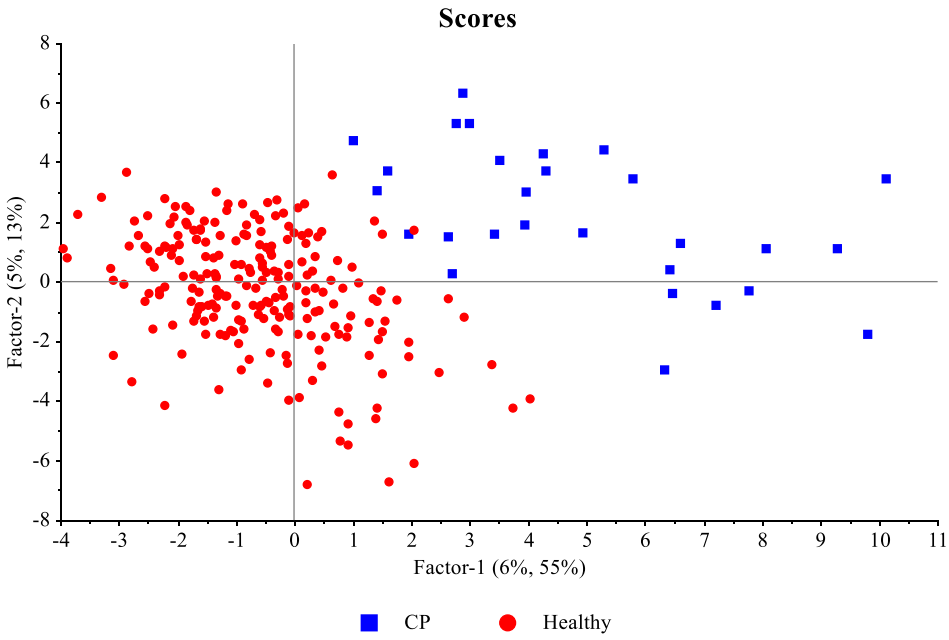
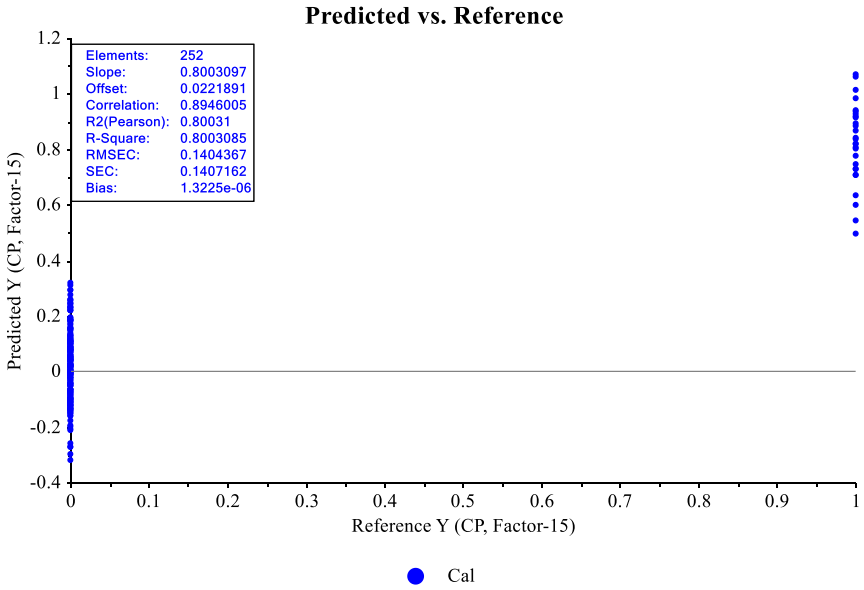
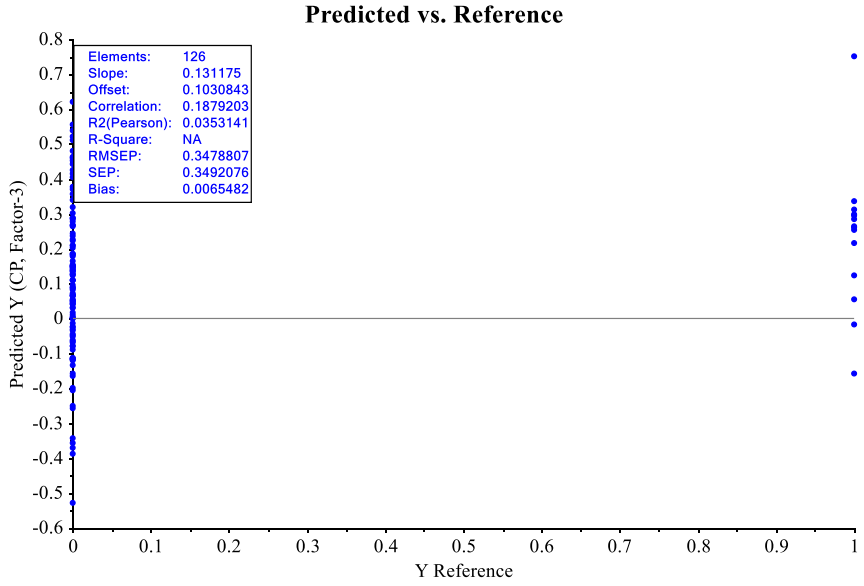


Figure 4.5: Score plot from PLS-DA model with test set validation.



(a) Prediction of value during training.



(b) Prediction of value of test set.

Figure 4.6: Prediction with PLS-DA model with test set validation. Class 0 equals healthy and 1 is CP.

4.2.2 Validation with Cross validation

For the second PLS-DA model, all samples are used in cross validation. The best results are obtained when using a model with 10 factors. A 20-fold cross validation is used, and the resulting score plot for the model is shown in Figure 4.7. The prediction done by this model in calibration is shown in Figure 4.8a and in validation in Figure 4.8b. The same cut-off of 0.5 used in the test set validated PLS-DA model is also used here. The confusion matrix for calibration is shown in Table 4.3 and validation in Table 4.4.

Table 4.3: Confusion matrix and performance metrics for PLS-DA model with cross validation in training.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	336	0
	CP	20	22

Accuracy	94.71 %
NPV	94.38 %
PPV	100 %
Sensitivity	52.38 %
Specificity	100 %

Table 4.4: Confusion matrix and performance metrics for PLS-DA model with cross validation in validation.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	336	0
	CP	23	19

Accuracy	93.92 %
NPV	93.59 %
PPV	100 %
Sensitivity	45.24 %
Specificity	100 %

CP subtype

The score plot from the PLS-DA model with CP subtype marked for each sample is shown in Figure 4.9. The samples that are classified wrong are marked with a circle. The subtypes are numbered like:

-1. Healthy

0. Unknown

1. Spastic Bilateral

2. Spastic Unilateral

3. Dyskinetic

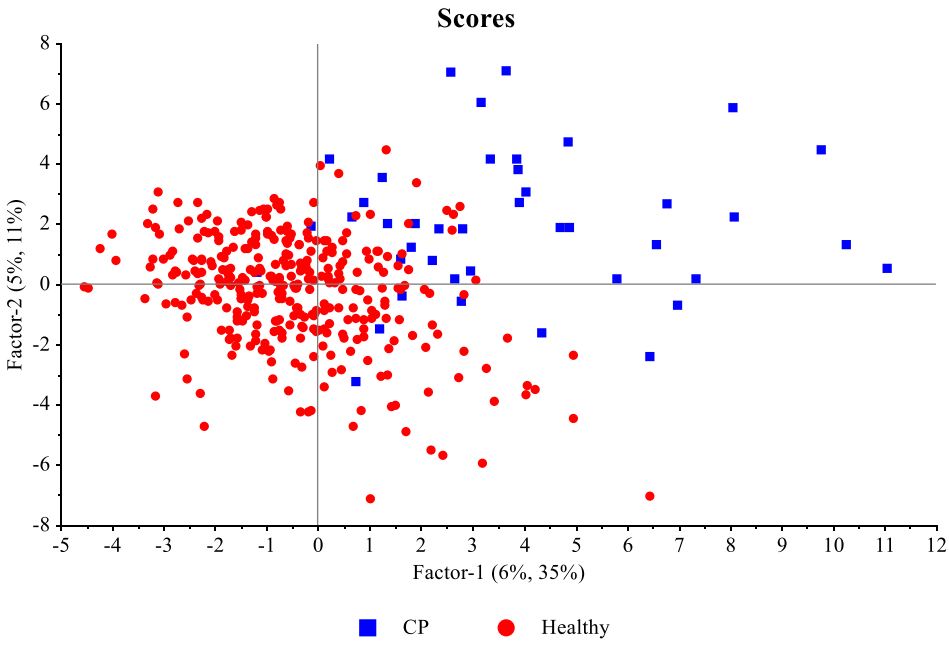
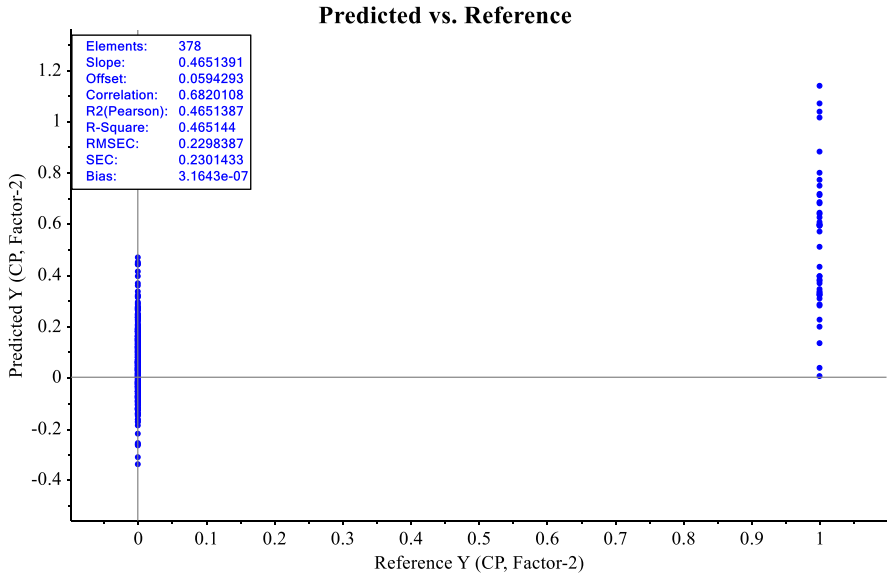
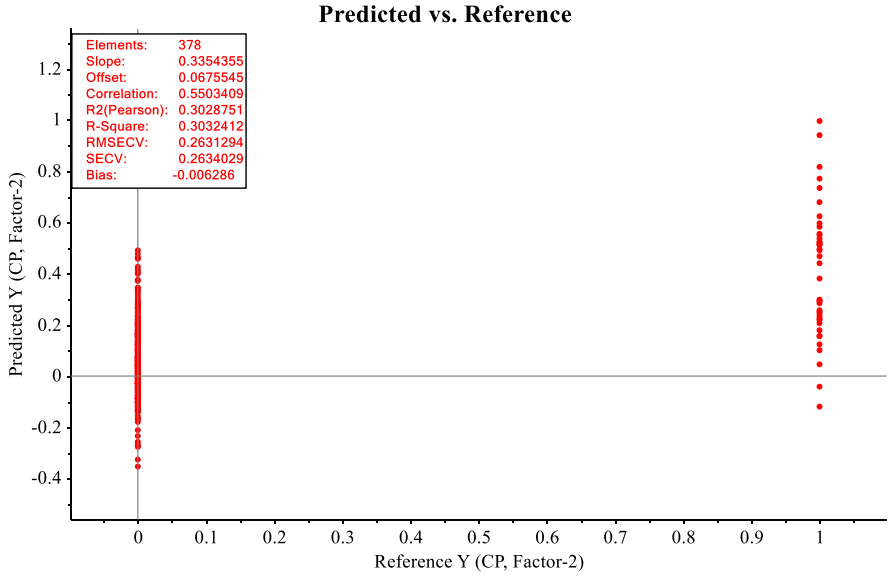


Figure 4.7: Score plot from PLS-DA model with cross validation.



(a) Prediction of value during training.



(b) Prediction of value in validation.

Figure 4.8: Prediction with PLS-DA model with cross validation. Class 0 equals healthy and 1 is CP.

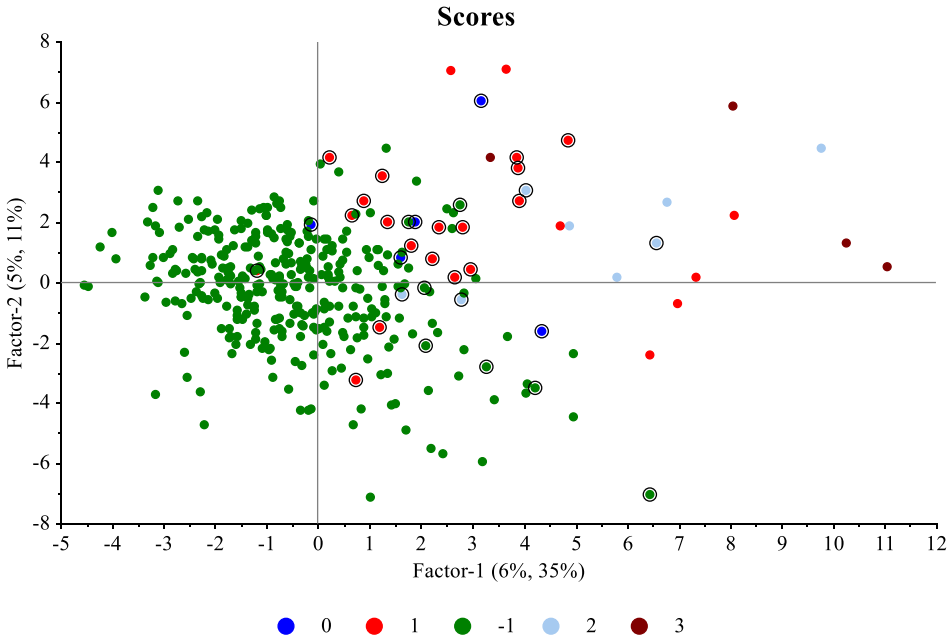


Figure 4.9: Score plot from PLS-DA model with cross validation. Each sample is colored according to CP subtype.

4.3 PCA

For classification with SVM and Random Forest the same two validation approaches are tested as with the PLS-DA modelling; test set validation and cross validation.

Test set validation

For the test set validation procedure, PCA is done first on the training set, which is the same set as the training set in Section 4.2. Then the test set is projected onto this PCA model to be able to use the scores of the test set for testing the SVM and Random Forest model. This is the same procedure as described in Section 2.5.1. This PCA model will further be referred to as "PCA with test set validation" even though it is not the PCA model that is validated with the test set. The explained variance for the PCA model on the training set is shown in Figure 4.10.

Cross validation

For the cross validation procedure, PCA is done on the full sample set. The explained variance of the model is shown in Figure 4.11.

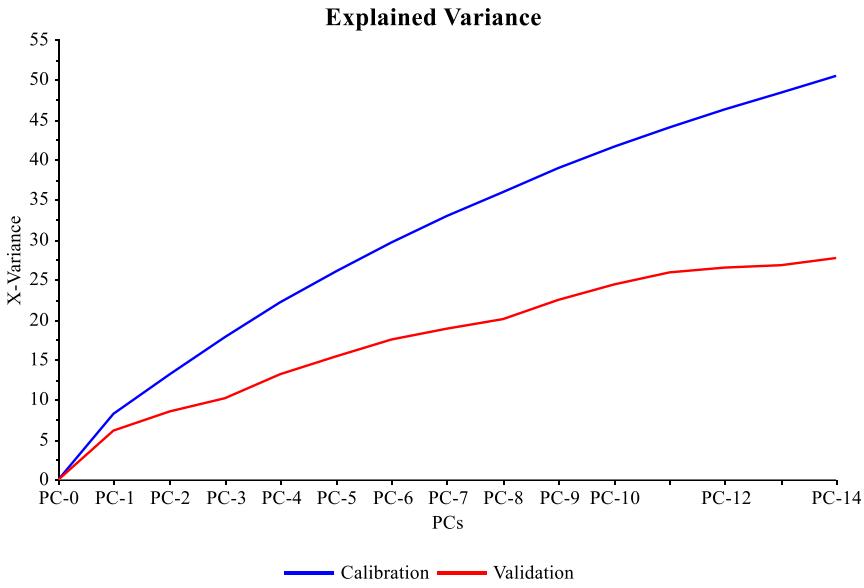


Figure 4.10: Explained Variance of PCA on training set.

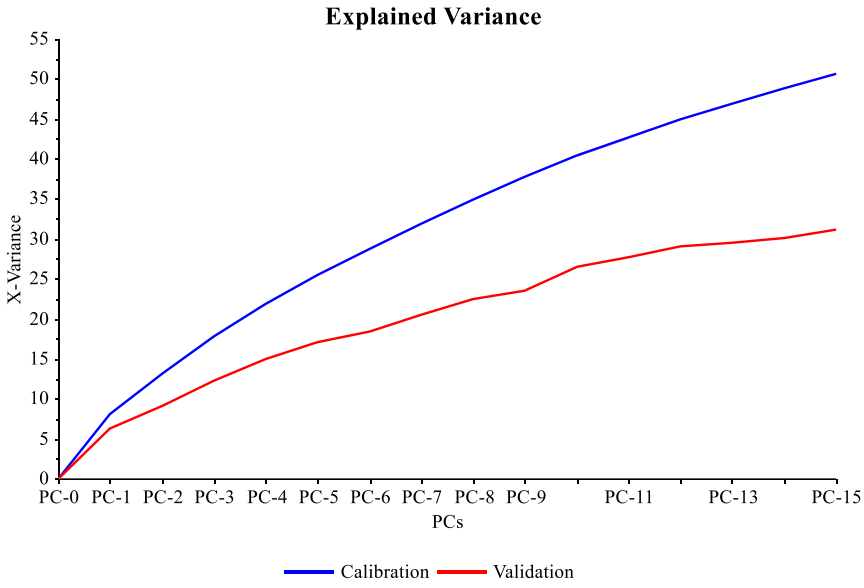


Figure 4.11: Explained Variance of PCA of full set.

4.4 SVM

Four different SVM classifiers are made:

1. On the PCA scores for the training set. Tested on the projected scores of the test set.
2. On the PCA scores for the full sample set. Validated with cross validation.
3. On the feature subset directly. Trained on the training set and tested on the test set.
4. On the feature subset directly. Trained on full sample set and validated with cross validation.

4.4.1 On PCA scores

Two SVM classifiers are trained and tested on the PCA scores from Section 4.3.

Test set validation

When training a SVM on the PCA scores of the training set the best results are obtained when using the first 10 principal components. This gives a training accuracy of 97.22% and the model use 47 support vectors. Using cross validation in the training step gives a validation accuracy during training of 94.44%.

Classifying the projected test samples onto the PCA model gives the confusion matrix and performance metrics given in Table 4.5.

Table 4.5: Confusion matrix and performance metrics for SVM on PCA scores for test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	110	2
	CP	13	1

Accuracy	88.10 %
NPV	89.43 %
PPV	33.33 %
Sensitivity	7.14 %
Specificity	98.21 %

Full subject set.

When training a SVM on the PCA scores of the full subject set and using 20-fold cross validation the best results are obtained using the first 15 principal components. This gives a training accuracy of 94.71% and the model use 90 support vectors. The validation accuracy for this model is 92.86%.

4.4.2 On feature subset

Two SVM are trained and tested on the feature subset chosen in Section 4.1.

Test set validation

Training a SVM on the training set gives a training accuracy of 100%. This model uses 128 support vectors. The cross validation internal in the training gives a validation accuracy of 92.857%.

Classifying the test set with this model gives the confusion matrix and performance metrics given in Table 4.6.

Table 4.6: Confusion matrix and performance metrics for SVM on feature subset for test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	110	2
	CP	13	1

Accuracy	88.10 %
NPV	89.43 %
PPV	33.33 %
Sensitivity	7.14 %
Specificity	98.21 %

Full subject set.

Training a SVM on the full subject set and validating with 20-fold cross validation gives a model with training accuracy 100%. This model uses 208 support vectors. The validation accuracy is 88.89%.

Summary SVM

Table 4.7: Performance metrics for all SVM models

	With PCA scores		**Without PCA scores**	
	Test set	**Cross validation**	**Test set**	**Cross validation**
Support vectors	47	90	128	208
Training accuracy	97.22 %	94.71 %	100 %	100 %
Validation accuracy	88.10 %	92.86 %	92.86 %	88.89 %

Table 4.8: Misclassifications of subjects with CP by ID number. X indicates misclassification.

ID	PLS	PCA-SVM	SVM	PLS-DA (CV)	PCA-SVM (CV)
'UoC_019_1_1'	X	X	X	X	
'LCH_181-1-1'	NA	NA	NA		
'LCH_003-1-1'	NA	NA	NA		
'LCH_078-1-1'	X	X	X	X	X
'LCH_001-1-1'	NA	NA	NA	X	X
'016-1-1'	NA	NA	NA		
'064-1-1'	X	X	X	X	X
'146-1-1'	NA	NA	NA		
'127-2-1'	NA	NA	NA		
'031-2-2'	X	X	X	X	X
'032-1-1'	NA	NA	NA		
'044-1-1'	NA	NA	NA		
'022-1-1'					
'UoC_040_1_3'	NA	NA	NA	X	
'UoC_109_1_1'	NA	NA	NA		
'UoC_006_1_1'	X	X	X	X	X
'LCH_182-1-1'	NA	NA	NA	X	X
'LCH_167-1-2'	NA	NA	NA	X	
'LCH_171-1-1'	X	X	X	X	X
'LCH_087-2-1'	NA	NA	NA	X	
'LCH_017-1-1'	NA	NA	NA		
'LCH_030-1-1'	X	X	X	X	X
'LCH_152-1-1'	NA	NA	NA	X	X
'LCH_073-1-1'	NA	NA	NA	X	
'LCH_080-1-1'	X	X	X	X	X
'108-2-1'	NA	NA	NA		
'021-2-1'	NA	NA	NA		
'067-1-1'	X	X	X	X	X
'114-1-1'	NA	NA	NA		
'035-1-1'	NA	NA	NA	X	X
'130-1-1'	X	X	X	X	X
'134-1-1'	NA	NA	NA		
'111-1-1'	NA	NA	NA	X	X
'028-1-1'	X	X	X	X	
'019-1-1'	NA	NA	NA		
'099-1-1'	NA	NA	NA	X	X
'075-2-1'	X	X	X	X	X
'092-1-1'	NA	NA	NA	X	X
'047-1-1'	NA	NA	NA		
'014-1-1'	X	X	X	X	X
'029-2-1'	NA	NA	NA	X	X
'024-2-1'	NA	NA	NA		

4.5 Random Forest

Four different Random Forest classifiers are made:

1. On the PCA scores for the training set. Tested on the projected scores of the test set.
2. On the PCA scores for the full sample set. Validated with cross validation.
3. On the feature subset directly. Trained on the training set and tested on the test set.
4. On the feature subset directly. Trained on full sample set and validated with cross validation.

4.5.1 On PCA scores

Test set validation

A Random forest classifier is trained on the PCA scores of the training set. The Out-Of-Bag classification error for this model is shown in Figure 4.12.

Using the Random forest classifier gives the confusion matrix and performance metrics given in Table 4.9.

Table 4.9: Confusion matrix and performance metrics for Random forest on PCA scores for test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	111	1
	CP	13	1

Accuracy	88.89 %
NPV	89.52 %
PPV	50 %
Sensitivity	7.14 %
Specificity	99.11 %

Full subject set.

A Random forest classifier is trained on the PCA scores of the full sample set. The Out-Of-Bag classification error for this model is shown in Figure 4.13.

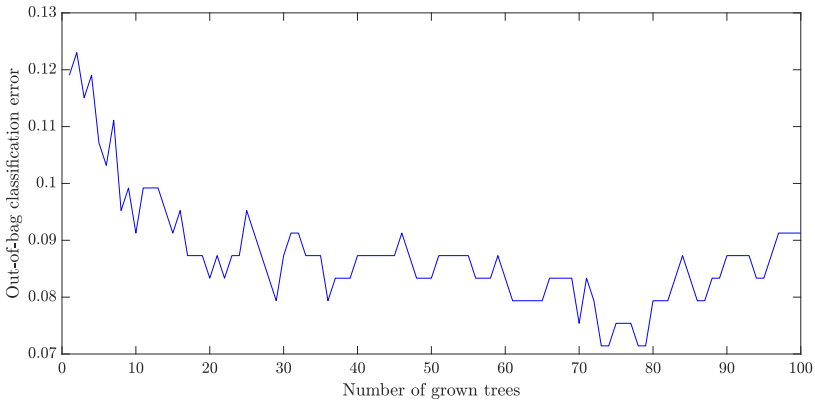


Figure 4.12: Out-of-bag classification error for model with PCA scores. Trained on training set.

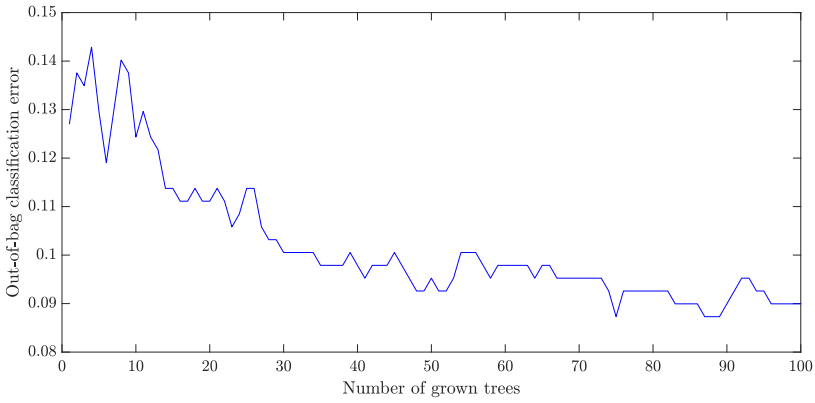


Figure 4.13: Out-of-bag classification error for model with PCA scores. Trained on full sample set.

4.5.2 On feature subset

Two Random Forest classifiers are trained and tested on the feature subset chosen in Section 4.1.

Test set validation

Training a Random Forest classifier on the training set gives the Out-Of-Bag classification error shown in Figure 4.14.

Using the Random forest to classify the test set gives the confusion matrix and performance metrics given in Table 4.10.

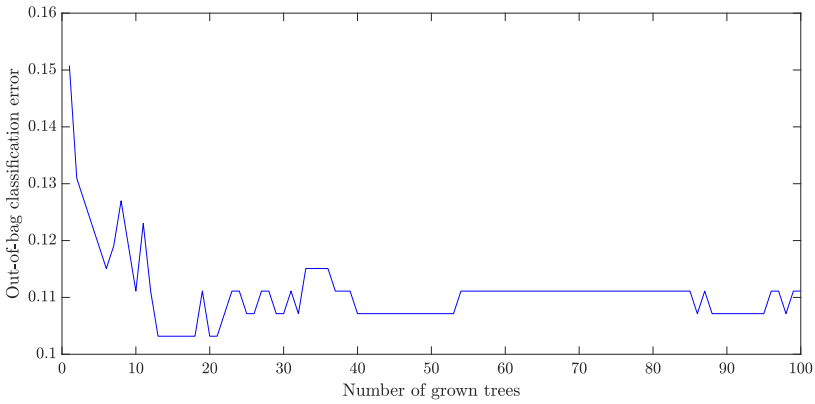


Figure 4.14: Out-of-bag classification error for model on feature set. Trained on training set.

Table 4.10: Confusion matrix and performance metrics for Random forest on feature subset for test set.

		Predicted class	
		Healthy	**CP**
True class	**Healthy**	110	2
	CP	14	0

Accuracy	87.3 %
NPV	88.71 %
PPV	0 %
Sensitivity	0 %
Specificity	98.21 %

Full subject set

Training a Random forest classifier on the full sample set gives the Out-Of-Bag classification error shown in Figure 4.15.

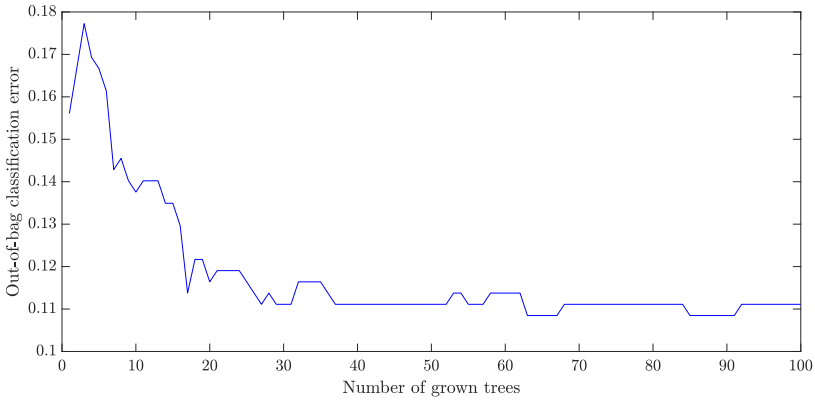


Figure 4.15: Out-of-bag classification error for model on feature set. Trained on full sample set.

Chapter 5

Discussion of results

Using test set validation is a great way to validate a model without bias and to compare different models. However, when the sample set is small, test set validation does not capture the full variation of all the samples. Having a total of only 42 samples with CP, the variation between all babies with CP is most likely only partially captured and making this sample set even smaller by dividing into two sets makes the model even less general. The big difference in the outcome of CP for every subject explained in Section 1 makes this problem of few samples even bigger as the variation between samples is large.

In both the PLS-DA model and the SVM classifier with cross validation there are ~ 20 subjects with CP that are classified as healthy. This can be seen in Table 4.8. These are the same subjects in both models, which may indicate that these subjects have a movement pattern that is more similar to the healthy babies' than the rest. The remaining 22 subjects are always rightly classified with CP. This significant difference in how much harder some subjects are to classify than the rest also affects the classifiers with test set validation. Even though the training and test set were chosen at random, the 14 subjects with CP in the test set are all but one part of the set of subjects that the cross validated models does not manage to classify rightly in validation. The one that is rightly classified by the CV classifiers are also always rightly classified by the test set validated classifiers. This unfair selection of test set contributes to the high accuracy in training and the equally bad accuracy in testing. Inspecting the PLS-DA models score plots visually, Figure 4.5 and 4.7, it may seem that they are very similar, only with some added samples in the CV model. This similar shape and distribution of samples may indicate that the test set samples that are added to the training step in the CV model does not contribute to the model.

It is worth noting that all the subjects with CP subtype Dyskinetic are classified rightly with CP by all the models. There are only 3 of these subjects so it may be a coincidence, but it is worth investigating further. The subtypes and classification is shown in Figure 4.9. Nearly all subjects that are wrongly classified have the CP subtype Bilateral Spastic, but as the dataset is so small this may not be valid for subjects outside this dataset.

The SVM classifier using PCA scores and the SVM classifier using the feature subset directly show that the overfitting problem can be reduced by dimension reduction. A significant gap between the training accuracy and validation accuracy may indicate overfitting and this gap should raise a red flag. The SVM classifier trained on the feature set and validated with cross validation has a training accuracy of 100% and validation accuracy of 88.89%. This means that all the samples lie on the correct side of the separating hyperplane during training, but a validation accuracy of 88.89% is equal to the accuracy obtained when all samples are classified as healthy due to the class imbalance. It is therefore safe to assume that the classifier is prone to overfitting and probably won't give as valid results for new samples. The SVM classifier trained on the PCA scores of the feature subset has a training accuracy of 94.71% and validation accuracy of 92.76%. This is not as good as the training accuracy of 100% of the feature subset model, but the high validation accuracy gives a hint that this model is more generalized and may be better for classifying new samples. Signs of overfitting may also be seen in the number of support vectors. In the PCA-SVM classifier there are 90 support vectors while in the classifier on the feature set directly there are 208 support vectors. When there are more support vectors the classifier is most likely less generalized than a classifier with fewer. This may also result in a higher misclassification rate, but in a lot of cases a higher misclassification rate is tolerable to be able to have a classifier that is more generalized. The Random Forest classifier on the feature subset also show signs of overfitting. The training accuracy is 100%, but the Out-Of-Bag misclassification error is $\sim 11\%$. This error means the same as the SVM's accuracy at 88.89%: all subjects are classified as healthy in the validation step. The Random Forest classifier with the PCA scores only have an Out-Of-Bag misclassification error of $\sim 9\%$. This is still not as good as the SVM classifier, but may show that the PCA scores reduce the effects of the overfitting, making the classifier more general.

It is interesting to note that the results of the PCA-SVM is good even though the PCA model only has an explained variance of $\sim 50\%$ when using all 15 PCs. This is shown in Figure 4.11. This means that about half of the variation in the dataset is not accounted for in the PCA model, but using these scores still gives a high accuracy in the SVM classifier. This may indicate that the variance captured is sufficient and manages to capture the variance that is needed to separate the subjects. Introducing more PCs would likely introduce more noise to the model and not give any better results.

It is also seen that when training a SVM classifier on only the training set and then validating with the test set, the best results are obtained using only the first 10 PCs. They only explain $\sim 40\%$ of the variance, but gives a higher accuracy than using all PCs. This may indicate that there is some information in the last 5 PCs that does not contribute to the model. However, when training a SVM on the full sample set and using cross validation instead of test set validation, 15 PCs is needed for the best results. This means that the last 5 PCs tells something about the variation that is needed to get a better classifier when training for the full sample set. The last PCs may contain some information that is needed to classify the test samples. This would not be discovered when training only on the training samples, but may be utilized by the cross validated SVM.

For the PLS-DA models the number of factors that gives the best accuracy has the inverse correlation with number of training samples than the PCA-SVM models. The best test set validated model uses 15 factors while the cross validated model uses 10 factors.

This may be explained by the fact that using more factors in the model will always make the model better, but after a certain number of factors the new factors will only introduce more noise into the model. The noise may increase the training accuracy but the validation accuracy will drop, which is a sign of overfitting. In this thesis the training accuracy for the PLS-DA model with test set validation is best when 15 factors are used, but the test validation accuracy is the same when using 10 or 15 factors, so the model with 15 factors may be prone to overfitting even though it is hard to tell with such a small test set.

The number of trees in the Random Forests affect the Out-of-bag classification error. All forests gets down to a OOB classification error aof $\sim 11\%$ at around 10 ± 5 trees, except the model trained on the full feature set. This requires ~ 27 trees, shown in Figure 4.15. This means that with ~ 10 trees a RF model has the same accuracy as obtained when classifying all subjects as healthy. The OOB classification error of the RF models trained on PCA scores decreases when more trees are added and gets their best result at $\sim 70 - 80$ trees. The trees added to the models trained on the feature set does not decrease the classification error, and in Figure 4.14 it is easy to see that the extra trees actually increases the OOB classification error.

Figures 4.3 and 4.4 shows some statistics about the 105 features that were chosen to be in the feature subset. Figure 4.3 shows which sensors' time series the different features were computed from. It is shown that the pelvis' time series in both x- and y-coordinate make up a big part of the feature set, in total this point has $28/105$ features. It may seem that because of this the pelvis movement is significant, but since this is a point of the body which does not move a lot, it is more likely that the reason there are many pelvis features is that it takes more features to capture the same variation as from some of the other sensors. It is also shown that the x-coordinates dominate in the feature subset: 63 of the features come from x-coordinated while only 42 are from y-coordinates. Physically it is likely that there is more movement in the x-direction than the y-direction, and that this movement is more important in the separation between the subjects. The number of features from the right arm and leg outnumber the features from the left side. The right extremities have 29 features while the left have 16. About 90% of the population is right-handed [47], which may explain this skewness. It sounds likely that the movement of the dominant hand may contain more information than the other.

In Figure 4.4 the features are grouped by the keywords of the computation done. Both the number of features from each group and the percentage that this selection make up of the full operation set in *hctsa* is shown. Few statistics features are chosen, as well as quite few in both trend and model fit. The Dynamical System operations have a higher percentage of chosen features, but still does not have a significant amount of features in the set. The dominating group of features is Wavelet based operations. Wavelet transforms exhibits good energy localization in the time-scale plane [48] and is more suitable for non-stationary signals than i.e. the Fourier Transform. Since the time series in this thesis are highly nonstationary it is reasonable that wavelet decomposition can give valuable information about the time series. Other nonlinear methods are also a big part of the feature set. Visibility graph analysis uses the tool of visibility graphs [49] to calculate various statistics on the properties of the resulting network. [49] states that the visibility graph characterizes

nontrivial time series and, in that sense, the method may be relevant in specific problems of different garments, such as human behavior time series. The Stationarity-group consists of operations that measures how properties of a time series change over time. This is highly relevant to this dataset as it is human movement that is analyzed. A hypothesis is that the fidgety movements of the healthy subjects are more regular and repeating than the movements of the subjects with CP. This makes analyzing the change over time interesting, as the hypothesis states that the change over time will be more unpredictable and less repeating for the subjects with CP. It is important to notice that the selected features were selected by hand, and no hard threshold was used for the weighted regression coefficients. The 105 selected features were selected through 3 steps where a manual selection was involved. This means that there are probably several feature subsets that will give the same results, and this is not necessarily the optimal one.

In [1], which is the primary reference for the hypothesis in this thesis, they assess the quality of fidgety movement and divide them into three categories: normal, abnormal and absent. They used these movements to predict the neurological outcome, and showed that normal fidgety movements was a sign of normal neurological outcome with an accuracy of 96%. One problem with the classification done in this thesis is that it does not separate between whether FM are normal, abnormal or absent but rather just the outcome. Also it does not account for other neurological abnormalities as they do in the study [1]. The models developed in this thesis shows that without having information about movement qualities the classifiers manages to extract some of this information either way and use in classification. The models show that there is most likely something in common for the healthy subjects while there is a bigger differentiation between the subjects with CP. This is consistent with normal FM as a sign of a normal neurological outcome. The difference between normal, abnormal and absent FM may be an explanation for why some of the samples are easier to separate than the other. Some of the subjects that are marked as healthy in this dataset may have other neurological abnormalities than CP that may cause abnormal or absent FM.

Chapter 6

Conclusion

The work done in this thesis highlights several challenges in data analysis. The class imbalance of the dataset makes it harder to compare different classification methods, and the small number of subjects with CP makes it best to avoid using test set validation. The SVM classifiers and Random Forests are prone to overfitting when using the feature set as input. This is shown by the fact that even though the training accuracy for these models is 100% the validation accuracy is 88.89%. This can be reduced by using PCA scores as input. The models also show that there is probably some underlying sources of variation that contributes to making ~ 20 of the samples harder to separate than the others.

The models in this thesis support the finding in [11] that a computer-based tool for early prediction of CP is feasible. All the classifiers find some similarities in the movements of the healthy samples, and are able to classify most subjects correctly as well as give a certainty of that classification. However, the sensitivity of the models are worse than all existing tools for diagnosis. The PLS-DA model with CV has a specificity of 100% which is as good as it can get and is better than for both GMA and ultrasound, but the sensitivity of 45.24% can't compare to GMA's 95% or ultrasound's 80%. To be able to use the methods in clinics both these performance metrics should be high and at the same level. Because of this none of the classifiers developed in this thesis are good enough for clinical use, but they show that the methods are promising.

The hypothesis that this thesis was built on was that fidgety movements can be recognized and used to classify infants without CP. The classifiers in this thesis manages to find some similarities between the healthy subjects, but as they also wrongly classify about half of the subjects with CP it is not only FM that they separate on. The models in this thesis does not manage to find the FMs and separate based only on these movements. The subjects may also have other neurological abnormalities than CP which may influence their movement qualities. This is not accounted for in this thesis and should be a subject for further research.

Chapter 7

Future Work

As mentioned in Section 5, the feature subset of 105 features used in this thesis are chosen manually using appropriate models. This gives a good feature set, but does not mean that it is neither the only one or the optimal one. It would be interesting to use a wrapper method for feature selection on the same big feature set from *hctsa* to find the best possible feature set.

The high frequency in the feature set of both wavelet operations and stationarity measures support the hypothesis that it is interesting to look at the frequency and development of the movements as fidgety movements are believed to be repeating, circular movements. More work should be done in investigating how these features influence the model, and it would also be interesting to map these features back onto the original time series. This could maybe give an even deeper understanding of the fidgety movements and how to locate them.

The heatmaps developed as a visualization tool in Section 3.2.3 could be used as a feature and analyzed with multivariate methods. If these heatmaps used as a feature would give any results it would be easy to visualize the results.

More work should be done in the selection of training and test set. This thesis emphasizes the challenge of choosing a representative training set and to be able to use this validation method a common training and test set should be selected for validating all methods in the project.

Other neurological abnormalities than CP have not been known in this thesis. It would be interesting to see if other neurological outcomes affect the movements and can lead to better results in classification. A classification based on known movement qualities instead of known outcome should also be tested.

Bibliography

[1] Giovanni Cioni Arend F Bos Fabrizio Ferrari Dieter Sontheimer Heinz F R Prechtl, Christa Einspieler. An early marker for neurological deficits after perinatal brainlesions. *Lancet*, 1997.

[2] Peter O D Pharoah Allan Colver, Charles Fairhurst. Cerebral palsy. *Lancet*, 2014.

[3] Cornelieke S. Aarnoudse-Moens Jorrit F. de Kieviet, Jan P. Piek. Motor development in very preterm and very low-birth-weight children from birth to adolescence. *JAMA*, 2009.

[4] Doernberg NS Benedict RE Kirby RS Durkin MS Yeargin-Allsopp M, Van Naarden Braun K. Prevalence of cerebral palsy in 8-year-old children in three areas of the united states in 2002: a multisite collaboration. *Pediatrics*, 2008.

[5] Beckung E Hagberg B Uvebrant P. Himmelmann K, Hagberg G. The changing panorama of cerebral palsy in sweden. ix. prevalence and origin in the birth-year period 1995-1998. *Acta Paediatr.*, 2005.

[6] Nancy A. Murphy MD Garey H. Noritz, MD. Motor delays: Early identification and evaluation. *The American Academy of Pediatrics*, 2013.

[7] CerebralPalsy.org. Aap urges for early diagnosis. `https://www.cerebralpalsy.org/about-cerebral-palsy/diagnosis/aap`, 2013. Accessed: 2018-10-23.

[8] Prechtl HFR Hadders-Algra M. Developmental course of general movements in early infancy. i: descriptive analysis of change in form. *Ealry Hum Dev*, 1992.

[9] Peter B. Marschik Christa Einspieler, Robert Peharz. Fidgety movements - tiny in appearance, but huge in impact. *Journal de Pediatria*, dec 2015.

[10] Gunn Kristin Øberg Nils Thomas Songstad Cathrine Labori Inger Elisabeth Silberg Marianne Loennecken Unn Inger Møinichen Randi Vågen Ragnhild Støen Lars Adde Toril Fjørtoft, Kari Anne I. Evensen. High prevalence of abnormal motor repertoire at

3months corrected age in extremely preterm infants. *Official Journal of the European Paediatric Neurology Society*, 2015.

[11] Jensenius A. R. Taraldsen G. Grunewaldt K. H. Støen R. Adde L., Helbostad J. Early prediction of cerebral palsy by computer-based video analysis of general movements: a feasibility study. *Developmental Medicine & Child Neurology*, 2010.

[12] B. Marschik P. A novel way to measure and predict development: A heuristic approach to facilitate the early detection of neurodevelopmental disorders. *Pediatric Neurology*, 2017.

[13] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. *Segmenting Motion Capture Data into Distinct Behaviors*. Canadian Human-Computer Communications Society, 2004.

[14] L.J. Cao, K.S. Chua, W.K. Chong, H.P. Lee, and Q.M. Gu. A comparison of pca, kpca and ica for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1):321 – 336, 2003. Support Vector Machines.

[15] Ergun Gumus, Niyazi Kilic, Ahmet Sertbas, and Osman N. Ucan. Evaluation of face recognition techniques using pca, wavelets and svm. *Expert Systems with Applications*, 37(9):6404 – 6408, 2010.

[16] Kevin B. Englehart Levi J. Hargrove, Guanglin Li and Bernard S. Hudgins. Principal components analysis preprocessing for improved classification accuracies in pattern-recognition-based myoelectric control. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 56(5):1407 – 1414, 2009.

[17] George Kollios Jonathan Alon, Stan Sclaroff. Discovering clusters in motion time-series data. *Proc. IEEE CVPR*, June 2003.

[18] Boley D. Papanikolopoulos N. Cao D., Masoud O. T. Online motion classification using support vector machines. *International Conference on Robotics & Automation*, April 2004.

[19] Sayan Mukherjee, Edgar Osuna, and Federico Girosi. Nonlinear prediction of chaotic time series using support vector machines. *Neural Networks for Signal Processing - Proceedings of the IEEE Workshop*, 07 1999.

[20] A. Palaniappan, R. Bhargavi, and V. Vaidehi. Abnormal human activity recognition using svm based approach. In *2012 International Conference on Recent Trends in Information Technology*, pages 97–102, April 2012.

[21] M.M. López, J. Ramírez, J.M. Górriz, I. Álvarez, D. Salas-Gonzalez, F. Segovia, and R. Chaves. Svm-based cad system for early detection of the alzheimer's disease using kernel pca and lda. *Neuroscience Letters*, 464(3):233 – 238, 2009.

[22] Interpolation. Interpolation — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Interpolation`, 2019. Accessed: 2019-03-15.

[23] Ronald K. Pearson, Yrjö Neuvo, Jaakko Astola, and Moncef Gabbouj. Generalized hampel filters. *EURASIP Journal on Advances in Signal Processing*, 2016(1):87, Aug 2016.

[24] André Elisseeff Isabelle Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, March 2003.

[25] Ben D. Fulcher and Nick S. Jones. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. *Cell Systems*, 5, November 2017.

[26] Feature selection. Feature selection — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Feature_selection`, 2018. Accessed: 2018-12-04.

[27] Ben D. Fulcher and Nick S. Jones. Highly comparative feature-based time-series classification. *IEEE Transactions On Knowledge and Data Engineering*, 26, December 2014.

[28] Ben D. Fulcher and Nick S. Jones. hctsa-manual. `https://hctsa-users.gitbook.io/hctsa-manual/`, 2017.

[29] Brad Swarbrick. *Multivariate Analysis for Dummies*. John Wiley Sons, Ltd, 07 2012.

[30] Harald Martens. Simple algorithms for pca and plsr, 2016. Accessed: 2018-12-0.

[31] Kevin Wright. The nipals algorithm. `https://cran.r-project.org/web/packages/nipals/vignettes/nipals_algorithm.pdf`, October 2017.

[32] CAMO Software AS. The unscrambler® appendices: Method references. Accessed: 2019-05-07.

[33] Statistical classification. Statistical classification — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Statistical_classification`, 2018. Accessed: 2018-12-05.

[34] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[35] Camo. Pls-da. `https://www.camo.com/resources/pls-da.html`. Accessed: 2018-12-05.

[36] Henry Han and Xiaogian Jiang. Overcome support vector machine diagnosis overfitting. *Cancer informatics*, 13:145–158, Dec 2014.

[37] Frank Westad. Cluster analysis, classification and discrimination. 2018.

[38] CAMO Software AS. Unscrambler. https://www.camo.com/unscrambler/.

[39] A. Verikas, A. Gelzinis, and M. Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern Recognition*, 44(2):330 – 349, 2011.

[40] Miroslav Kubat and Stan Matwin. Addressing the course of imbalanced training sets: One-sided selection. 1997.

[41] Accuracy and precision. Accuracy and precision — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Accuracy_and_precision, 2019. Accessed: 2019-05-07.

[42] Kristian Aurlien and Daniel Groos. Master thesis, 2018.

[43] SCPE Collaborative Group. Surveillance of cerebral palsy in europe: a collaboration of cerebral palsy surveys and registers. *Developmental Medicine and Child Neurology*, 56(42), 2000.

[44] OpenAIRE. Guides for researchers - how to deal with sensitive data. https://www.openaire.eu/sensitive-data-guide.

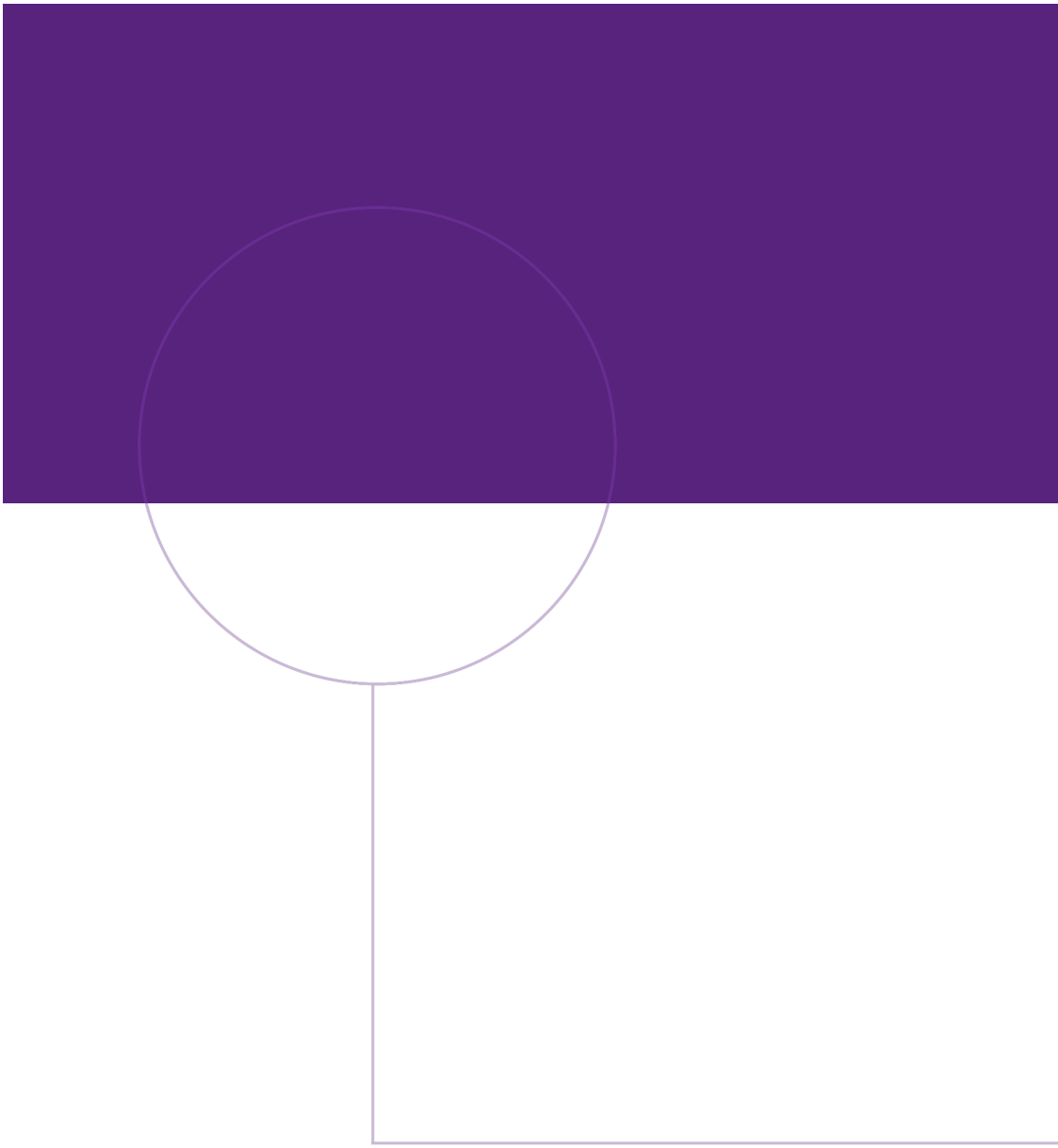
[45] Gdpr - article 4: Definitions. https://gdpr-info.eu/art-4-gdpr/.

[46] Accuracy Paradox. Accuracy paradox — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Accuracy_paradox, 2019. Accessed: 2019-03-05.

[47] Petrinovich LF Hardyck C. Left-handedness. *Psychol Bull*, 84:385–404, 1977.

[48] Chowdhury, Reaz, Ali, Bakar, Chellappan, and Chang. Surface electromyography signal processing and classification techniques. *Sensors*, sep 2013.

[49] Fernando Ballesteros Jordi Luque Lucas Lacasa, Bartolo Luque and Juan Carlos Nuño. From time series to complex networks: the visibility graph. *Proceedings of the National Academy of Sciences of the United States of America*, 105:4972–4975, 2008.



NTNU
Norwegian University of
Science and Technology