Brein-computer interfaces met machinaal leren: dataselectie voor overdracht van informatie in ingebeelde beweging

Bjorn Vuylsteker

Promotoren: prof. dr. ir. Joni Dambre, dr. ir. Pieter van Mierlo
Begeleider: ir. Thibault Verhoeven

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: elektronica-ICT

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. Rik Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2015-2016

UNIVERSITEIT GENT

Brein-computer interfaces met machinaal leren: dataselectie voor overdracht van informatie in ingebeelde beweging

Bjorn Vuylsteker

Promotoren: prof. dr. ir. Joni Dambre, dr. ir. Pieter van Mierlo
Begeleider: ir. Thibault Verhoeven

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: elektronica-ICT

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. Rik Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2015-2016

UNIVERSITEIT GENT

Voorwoord

Eerst en vooral wil ik promotor Professor Joni Dambre en Pieter van Mierlo bedanken voor het ter beschikking stellen van deze thesis en in het bijzonder Thibault Verhoeven voor de onverbiddelijke steun en het enorme vertrouwen tijdens het verloop van deze thesis.

Ook wens ik Thijs Windels te bedanken voor de goede samenwerking in het begin van deze thesis en Nina Proesmans voor de leerrijke discussies en de verschillende keren dat ze me bepaalde aspecten in een ander opzicht liet zien.

Mijn oprechte dank gaat ook uit naar mijn ouders, die mij altijd volop gesteund hebben doorheen mijn studies. Zonder de kansen die ze me gaven, was ik nooit geraakt waar ik nu ben.

Vuylsteker Bjorn

Toelating tot bruikleen

Vuylsteker Bjorn, mei 2016

Brein-computer interfaces met machinaal leren: dataselectie voor overdracht van informatie in ingebeelde beweging

door

Vuylsteker Bjorn

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de Industriële Wetenschappen: Elektronica-ICT

Academiejaar 2015–2016

Promotor: Prof. Dr. Ir. J. Dambre, Dr. Ir. P. van Mierlo
Begeleider: Ir. T. Verhoeven

Faculteit Ingenieurswetenschappen en Architectuur
Universiteit Gent

Vakgroep Elektronica en Informatiesystemen
Voorzitter: Prof. Dr. Ir. R. VAN DE WALLE

Samenvatting

Een brein-computer interface (BCI) is een input-output systeem dat ervoor zorgt dat de gebruiker een bepaald computersysteem kan aansturen via zijn hersenen. Dit systeem is echter heel persoonsgebonden, waardoor de gebruiker een lange kalibratietijd moet doorgaan van 20 tot 30 minuten. Daar dit veel concentratie en tijd van de gebruiker vergt, is dit niet gewenst.

Er bestaan echter technieken om die kalibratietijd te reduceren. Één hiervan is transfer learning, dat toelaat om via vooropgenomen data van eerdere gebruikers een BCI te bouwen met weinig kalibratiedata van de nieuwe gebruiker. Het is echter moeilijk om te achterhalen welke eerdere gebruikers overeenkomstig zijn met de nieuwe gebruiker.

In deze thesis wordt onderzoek gevoerd naar een transfer learning algoritme dat in staat is een accurate BCI op te stellen door gebruik van weinig kalibratiedata van een nieuwe gebruiker. Hierbij wordt ook onderzoek gedaan naar het vinden van een overeenkomstige gebruiker in een poele van eerdere gebruikers. De gebruikte algoritmen uit de literatuur, de bepaling van enkele parameters en de doorlopen stappen bij het opstellen van het uiteindelijk algoritme worden uitvoerig besproken in deze thesis.

Trefwoorden

Brain-Computer Interfaces, Machine Learning, Transfer Learning, Data Selection, RSAMS

Brain-Computer Interfaces with Machine Learning: Data Selection for Transfer Learning in Motor Imagery

Bjorn Vuylsteker

Supervisor(s): Thibault Verhoeven, Joni Dambre, Pieter van Mierlo

Abstract— **A major limitation of motor imagery-based brain-computer interfaces (BCI) is their long calibration time. Due to inter-subject variability, a new subject has to undergo a 20-30 minutes calibration session to train the BCI model based on the subject's brain patterns. Whereas this is exhausting and time consuming for the subject, a reduce in calibration time would be highly desirable. This paper proposes a new algorithm to reduce the calibration time, by using data from other subjects. More specifically, an algorithm that is able to automatically select a compatible subject from a large group of other subjects, whilst also correctly choosing the best method to construct the BCI with. An evaluation shows that the algorithm outperforms the standard BCI design and a similar subject-to-subject transfer learning algorithm. With a reduction in calibration time of 83.75%, the algorithm performed on average only 2% less accurate than the subject-dependent calibration results.**

Keywords— **Brain-Computer Interfaces, Machine Learning, Transfer Learning, Data Selection, RSAMS**

I. Introduction

Brain-computer interfaces (BCI) are input-output systems, that allow the control of a system via brain activity [1]. In most BCI-cases, brain activity is captured through Electro-Encefalography (EEG) signals [2]. Due to large inter-subject variabilities, a BCI has to be calibrated subject specific. This results in a long calibration session, lasting up to 20-30 minutes [3]. As this is a very time consuming and concentration intense process, a reduce in calibration time would be highly desirable.

As far as known, few studies have specifically addressed the comparability between multiple subjects in a subject-to-subject transfer learning environment. It is known that work presented in [4] has succeeded in making a selection of a subset of source subjects by using a variant of the Sequential Forward Floating Search algorithm. However, this method was only based on a single comparison metric and had few subjects to compare to. Also, the computational expensiveness of this proposed algorithm rises exponential with increasing source subjects, which does not make it suitable when having a large pool of source subjects. Another algorithm proposed in [5], determines the most comparable source subject by two metrics when more than one source subject has the same value on the first metric. This again limits the possibility of combining information from multiple metrics. However, this algorithm is computational effective when used with a large pool of source subjects, which is why it is used as a comparison measure.

This paper proposes a new way of selecting EEG-data from source subjects, to effectively use in the calibration process of the target subject. Hereby only a small portion of calibration data from the target subject is needed to accurately calibrate the BCI, hence reducing calibration time. In the first step of the algorithm called RSAMS (Ranking-based Subject And Methodology Selection), different subject-to-subject transfer learning algorithms set up a BCI based on the most compatible source subject as selected by a sorting system. Secondly, a selection between different algorithms is made by using Leave-One-Out-Validation (LOOV). This way, a BCI is created specifically optimized towards the target subject.

This paper is organized as follows: Section 2 will describe the different steps and motivation of the proposed algorithm. Then, Section 3 will describe the configuration of the experiments on this algorithm. Following, Section 4 will cover the discussion of the results from those experiments. Finally, Section 5 will conclude the paper.

II. Method

The new proposed algorithm consists of a two step process. First of all, different transfer learning algorithms are used to set up different types of BCI. These transfer learning algorithms all select a comparable target subject by using a general sorting system. This sorting system uses four different metrics, which will be explained in the following subsection. In the second step of the algorithm, a consideration is made between the different algorithms by using LOOV.

A. Metrics

To enable the sorting system to accurately discriminate the comparability between different subjects, a combination of multiple metrics is used to determine different kinds of comparability between distributions. All used metrics are elaborately explained in the paragraphs below.

Validation accuracy The validation accuracy is defined as the test accuracy of the calibration data, send through a BCI entirely trained on one or multiple source subjects. Literature has proven this to be an excellent and commonly used metric in the field of transfer learning [4], [5].

Consistency index The consistency index as described in [6] is a measure that describes the amount of correlation between two distributions. Whereas correlation is hereby defined as an exact match of compared values, it has to be tweaked to describe the amount of comparability between two distributions. This means that in the formula described in [6], parameter r_k has to be changed is such a matter that for a big difference between features l_A and l_B, the parameter has to approach $r_k = 0$. Vice versa, $r_k = 1$ should be reached when both features are

identical. This way of thinking is reflected in following formula

$$r_k(l_A, l_B) = \sum_{i=0}^{k-1} \frac{1}{1 + |l_A(i) - l_B(i)|} \tag{1}$$

where for a certain trial l, a comparison is made between distributions A and B for a constant subset of features k. To set up the consistency index over a group of trials, following formula is used

$$I_c(A, B) = \frac{1}{n * L} \sum_{l=0}^{L} \sum_{k=1}^{n-1} \frac{r_k(l_A, l_B) * n - k^2}{k(n-k)} \tag{2}$$

where the average of the comparisons is taken over all features n and trials L present in the dataset.

Maximum Mean Discrepancy Another method found in literature is the Maximum Mean Discrepancy (MMD) [7]. It allows the measurement of distance between two distributions. Another widely used metric to measure this distance, is Kullback-Leibler [8]. The disadvantage with KL however, is the computational expensiveness of the distribution density calculation. As a non-parametric distance measure, the idea behind MMD is that, under a sufficiently rich reproducing kernel Hilbert space (RKHS), if feature means of the population are identical then it is guaranteed that the distributions are the same. Based on this theorem, distance between samples found in two distributions can be measured by the difference of the empirical means of the samples in a RKHS.

Least Squared Euclidean Distance The Least Squared Euclidean Distance (LSED) is a method that uses the euclidean distance metric to determine the distance between two distributions. It is defined as the sum of the squares over different classes, of the euclidean distance taken between features of different distributions. This is formally noted as

$$LSED(A, B) = \sum_{c=1}^{C} \|\mu_A(c) - \mu_B(c)\|^2 \tag{3}$$

where C denotes the amount of distinguished classes and $\mu_A(c)$ and $\mu_B(c)$ define the average feature vectors corresponding to distributions A and B for a certain class c. Because the difference between average feature vectors of distributions is calculated, it can be used as a metric to relay a comparison between two distributions.

B. Subject selection

To effectively select a comparable target subject, a multi step sorting system was invented. This sorting system covers different procedures to eliminate bad comparisons, in order to reduce the computational expensiveness and raise the selection accuracy.

First, a comparison is made between the source subject's train data and the target subject's calibration data using four metrics explained in the previous subsection. Hereafter, an elimination is executed, removing all designed BCI that have a validation accuracy 15% lower than the maximum reached validation accuracy. The remaining BCI are sorted metric-specific and are assigned an order number accordingly as shown in figure 1. Each metric is then given a certain weight, corresponding to the importance and accuracy of that metric. Although these weights can be determined by training on multiple examples, it was deliberately chosen to appoint these weights empirical. To achieve a final ranking of the remaining BCI, a summation is made over all metrics by multiplying the metric-specific order numbers with their corresponding weights and sorting them by ascending order. Finally, the BCI with the highest ranking order in this final ranking is chosen as the most comparable BCI.

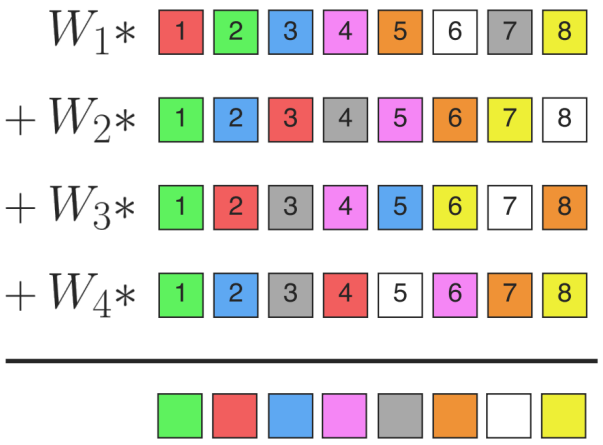


Fig. 1. A schematic display of the sorting system with 8 designed BCI. Each row represents the order of the designed BCI per metric, sorted by that particular metric. Next, each order number per BCI is multiplied by a weight assigned to that metric. To determine the final order, a summation is made per BCI over all metrics, sorted by ascending order. The BCI with the highest final order number is chosen as the one most suitable to use with the current target subject.

C. Methodology selection

Adjacent to subject selection, the proposed algorithm is also able to select the proper methodology that gives the highest accuracy according to the current target subject. This is done by letting each selected BCI undergo a Leave-One-Out-Validation. The BCI that scores the highest is then selected as the most comparable BCI and will be used in future processing of real-time data from the target subject.

In the proposed version of the RSAMS algorithm, three methodologies are used for comparison as shown in figure 2. The first algorithm is subject-dependent as it does not make use of transfer learning. The BCI is hereby only trained on the provided calibration data from the target subject. The second algorithm designs a BCI for each available source subject and trains them on their data accordingly. A comparable BCI is then selected using the sorting system explained in the previous subsection. The third algorithm makes use of a linear transformation as proposed in [5]. This transformation matrix is computed between the source subject's data and the target subject's calibration data in order to minimize the difference between both subjects. Hereafter, BCI are made, each trained on the transformed source subject's data. Finally, a comparable BCI is selected using the sorting system, much like in the second algorithm.

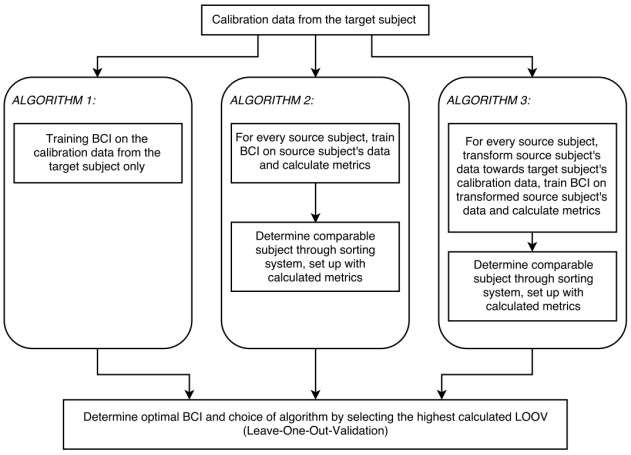


Fig. 2. A schematic display of the RSAMS algorithm

III. Experiments

As a large group of source subjects is needed, a publicly available dataset provided by Physionet [9] is used with evaluation, consisting of 109 subjects. As for the target subjects, EEG-data from BCI Competition IV, Dataset 1 [10] was used, consisting of 5 different subjects imagining left or right hand movement. From both datasets, corresponding channels were used for a total of 41 channels per subject. EEG-data from the Physionet database was subsampled to 100Hz to coincide with the EEG-data from the BCI Competition IV.

For each subject, signals from 0.5 to 4.5 seconds after the cue were applied. EEG-signals were bandpass filtered into a 8Hz to 35Hz frequency band using a Butterworth filter of the 4th order. Thereafter, signals were spatially filtered using Common Spatial Patterns (CSP), by using one single filterpair. Finally, log of variances of those signals are used as inputs for the LDA classifier. Note that this classifier does not use any form of regularization.

IV. Results and Discussion

In this study, only 30 calibration trials from the target subject were used for evaluation of the proposed transfer learning algorithm. Subsequently, all classification results presented in this section were obtained using the remainder of the target subject's data, unless otherwise specified. Prediction accuracy will be used as the main evaluation method.

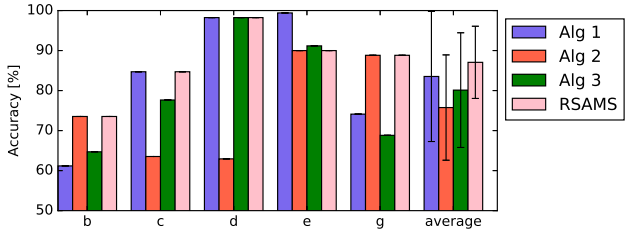


Fig. 3. A comparison between the RSAMS algorithm and the three algorithms used in RSAMS individually

In figure 3, a comparison is shown between the proposed algorithm and the individual methodologies it combines. These individual methodologies correspond to the ones explained in section II-C. The weights of the selection algorithm applied in algorithm 2 and 3, are 2, 1, 1 and 1, corresponding to the order they were addressed in in this paper.

On average, it is clear that RSAMS achieves the highest accuracy (87.06%) with the lowest standard deviation (9.02%) of the four compared algorithms. When looking at the subjects individually, RSAMS is always able to select the methodology with the highest achieving accuracy, except with subject e. This may be due to the overall high accuracy for this subject and the small difference in accuracy between the used methodologies.

For online-applications, it is very desirable to make the calibration session as short as possible, in comparison to the loss of accuracy. In figure 4, an evaluation is presented that shows the impact of changing the amount of calibration data on the test accuracy. Note that in this evalutation, the test accuracy was achieved on the last 40 trials of the target subject. This way, a comparison can be made to a subject-dependant BCI that was trained on a full calibration session (160 trials) and tested on those same 40 trials of the target subject. With this subject-dependant BCI, an average test accuracy of 90% was achieved. In the comparison of RSAMS and EEG-DSA can be seen that

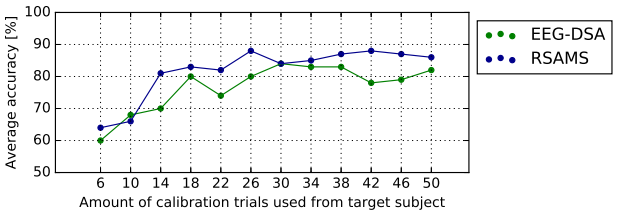


Fig. 4. A comparison between EEG-DSA and the RSAMS algorithm, with a variation in the amount of calibration trials used to set up the BCI

the proposed algorithm achieves a higher overall accuracy. It also more quickly achieves a higher and more stable accuracy when rising the amount of calibration data. As previously mentioned, the RSAMS algorithm achieves an average test accuracy of 88% with only 26 calibration trials used. This results in a reduce of 83.75% in calibration time, with only a small 2% loss in accuracy.

V. Conclusions

This paper proposed a new algorithm to reduce the calibration time for motor-imagery based BCI. Using the RSAMS algorithm, a consideration is made between multiple source subjects and methodologies. This way, a BCI is designed optimized for the target subject itself. Unlike several past studies, multiple metrics are used to determined compatibility between source subjects and a target subject. Furthermore, a great degree of freedom is applied to this algorithm, as more methodologies can be swapped or added and weights of the sorting system can be changed as pleased. The experimental results showed that with a small 2% loss in accuracy, the algorithm can reduce the calibration time by 83.75%, which is equivalent to a short 3-4 minute calibration session.

REFERENCES

[1] P. Brunner, S. Joshi, S. Briskin, J.R. Wolpaw, H. Bischof, and G. Schalk, "Does the "p300" speller depend on eye gaze," *Journal of Neural Engineering*, vol. 7, no. 5, pp. 056013, 2010.

[2] Biao Zhang, Jianjun Wang, and Thomas Fuhlbrigge, "A review of the commercial bci technology from perspective of industrial robotics," in *IEEE International Conference on Automation and Logistics*, 2010, pp. 379–384.

[3] Sami Dalhoumi, Grard Dray, and Jacky Montmain, "Knowledge transfer for reducing calibration time in brain-computer interfacing," in *IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014, pp. 634–639.

[4] F. Lotte and C. Guan, "Learning from other subjects helps reducing brain-computer interface calibration time," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 614–617.

[5] Mahnaz Arvaneh, Ian Robertson, and Tomas E. Ward, "Subject-to-subject adaptation to reduce calibration time in motor imagery-based brain-computer interface," in *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 6501–6504.

[6] Ludmila I. Kuncheva, "A stability index for feature selection," in *25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, 2007, pp. 390–395.

[7] Selen Uguroglu and Jaime Carbonell, "Feature selection for transfer learning," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011, pp. 430–442.

[8] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[9] G. Schalk, D.J. McFarland, T. Hinterberger, N. Birbaumer, and J.R. Wolpaw, "Bci2000: A general-purpose brain-computer interface (bci) system," *IEEE Transactions on Bio-Medical Engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[10] Benjamin Blankertz, Guido Dornhege, Matthias Krauledat, Klaus-Robert Mller, and Gabriel Curio, "The non-invasive berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, 2007.

Inhoudsopgave

Lijst met afkortingen

ALS	Amyotrofische Laterale Sclerose
BCI	Brein-Computer Interface
CSP	Common Spatial Patterns
EEG	Elektro-Encefalografie
ERD	Stimulus gerelateerde Desynchronisatie
ERP	Stimulus gerelateerde Potentialen
ERS	Stimulus gerelateerde Synchronisatie
LDA	Linear Discriminant Analysis
LOOV	Leave-One-Out-Validation
SL	Surface Laplacian
SNR	Signaal-ruisratio
TL	Transfer learning

Hoofdstuk 1

Introductie

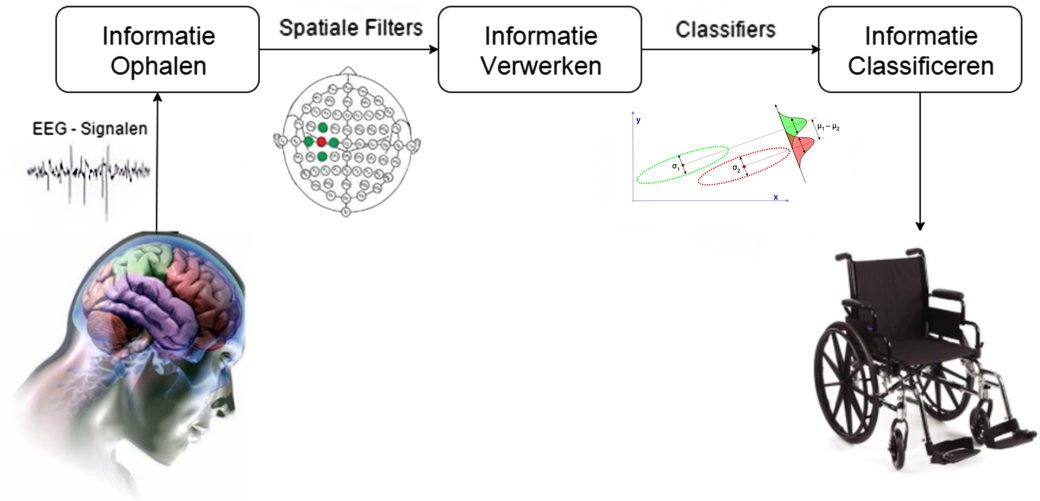
Informatie en communicatie technologie, het bracht een enorme revolutie met zich mee. Meer en meer toestellen en mechanismen krijgen een upgrade, die hen meer functionaliteit toekent. Denk maar aan zelf-rijdende wagens [1] of slimme thermostaten [2]. Het laat ons ook toe om meer geavanceerde systemen te ontwikkelen om het voor mensen met een beperking gemakkelijker te maken. Enkele voorbeelden hiervan zijn navigatiesystemen voor blinden [3], hoorimplantaten [4], robotische ledematen [5] en brein-computer interface systemen (BCI) [6]. Dit laatste is een systeem dat geen aansturing door middel van spierkracht vereist, waardoor het veel toegepast wordt bij mensen met een motorische beperking. Hieronder vallen ook mensen met ALS (Amyotrofische Laterale Sclerose). Dit is een aandoening van de motorische zenuwcellen in zowel het ruggenmerg als de hersenstam [7]. Wanneer motorische neuronen afsterven door de gevolgen van deze aandoening, verliezen de hersenen dus de mogelijkheid om de hieraan gelinkte spieren aan te sturen. In het meest gevorderde stadium van deze aandoening, zal de persoon zich bevinden in een pseudocoma, wat ook het Locked-in-syndroom genoemd wordt [8]. Hiermee wordt bedoeld dat de persoon zich in een minimaal responsieve status bevindt, hoewel het bewustzijn nog volledig actief is. Zij zitten als het ware gevangen in hun eigen lichaam. Hierdoor wordt het enorm moeilijk voor die persoon om te interageren met de omgeving. Daar de hersenen nog steeds actief zijn, kan hier gebruik gemaakt worden van een BCI om de interactie met de omgeving te bevorderen [9].

1.1 Brein-computer interfaces

Een brein-computer interface (BCI) is een input-output systeem dat hersenactiviteit gebruikt als aansturing van een computersysteem. Hierdoor kan er gecommuniceerd worden met de buitenwereld, zonder dat er lichamelijke activiteit plaatsvindt [10]. Een voorbeeld van zo een systeem is het aansturen van een rolstoel door middel van het inbeelden van een bepaalde beweging [11].

Een BCI wordt omschreven als een algemeen concept waarbij er veel verschillende soorten uitvoeringen zijn. Zo heb je een onderscheid tussen invasieve en niet-invasieve BCI. Het verschil tussen beide is dat bij invasieve BCI de hersengegevens vastgelegd worden door meetapparatuur die in het lichaam ingebracht is. De meestgebruikte manier is echter de niet-invasieve BCI, waarbij de hersensignalen gemeten worden op de schedel. Hierbij zijn er verschillende meetmethoden mogelijk die verder toegelicht worden in 1.1.2. Er worden ook verschillende neurologische signalen opgewekt in de hersenen, die vervolgens als controlesignaal gebruikt kunnen worden om een computersysteem aan te sturen.

In figuur 1.1 wordt schematisch uitgelegd wat de algemene werking van een BCI is. Vooraleerst wordt informatie opgehaald uit de hersenen. Deze worden vervolgens verwerkt door middel van verschillende algoritmes, om een bepaalde actie uit te voeren aan de hand van die informatie. Die actie wordt bepaald door een vooraf getrainde classifier.



Figuur 1.1: Een schematische voorstelling van de algemene werking van een BCI

Één van de wellicht meest publiekelijk gekende BCI gebruikers is de wereldberoemde fysicus Stephen Hawking. Hij werd reeds op jonge leeftijd gediagnosticeerd met ALS en kan dankzij een computersysteem ontwikkeld door Intel terug sociale activiteiten uitvoeren, zoals boeken schrijven, praten via een spraakcomputer en zichzelf voortbewegen. Zo schreef hij bijvoorbeeld de bestseller *A Brief History of Time* [12] met behulp van deze middelen.

1.1.1 De hersenen

Één van de belangrijkste organen in het lichaam zijn de hersenen. Deze maken deel uit van het centrale zenuwstelsel dat zich zowel in het hoofd als in het ruggenmerg bevindt. Dit zenuwstelsel bestaat uit vele miljarden neuronen, ook wel zenuwcellen genoemd. Deze neuronen zijn de informatie- en signaalverwerkers van het lichaam. Alle stimuli in- en uitwendig aan het lichaam worden hier verwerkt, met de nodige reacties hieraan gekoppeld. Wanneer een gevoel van pijn ervaren wordt, is dit bijvoorbeeld een reactie op een externe stimulus.

Er kunnen echter ook acties uitgevoerd worden zonder externe stimuli. Zo kan de stem gebruikt worden om iets duidelijk te maken, de handen om iets vast te nemen, etc. Natuurlijk wordt het volledige brein niet altijd benut bij het uitvoeren van bepaalde acties. Hiervoor zijn er verschillende regio's in de hersenen aanwezig, die allerlei taken simultaan kunnen uitvoeren, waardoor er verschillende categorieën van handelingen mogelijk zijn [13]. Neem nu aan dat er zich een stimulatie plaatsvindt op twee verschillende regio's in het deel van de hersenen die verantwoordelijk is voor beweging in het lichaam. Dit kan bijvoorbeeld als gevolg hebben dat de persoon tegelijk het hoofd draait en zijn hand opent.

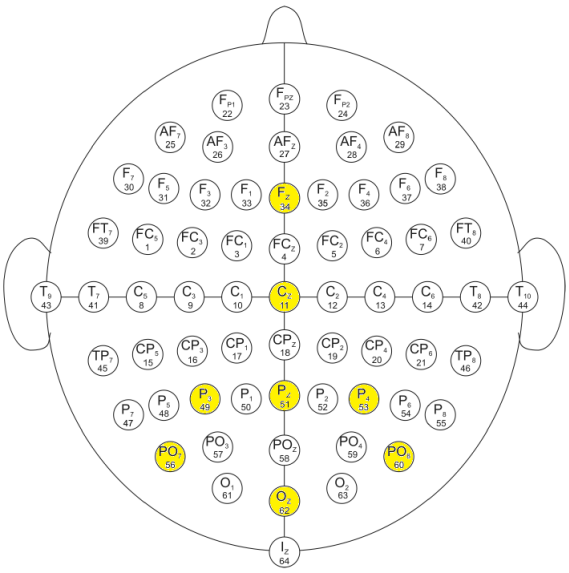
Doordat het brein het eigen lichaam kan aansturen, lijkt het ook logisch dat ze rechtstreeks externe toestellen kunnen aansturen [14]. Dit gebeurt via hersengolven, die opgevangen kunnen worden vanop de hersenschedel met de nodige meetapparatuur. Hoe deze nu net uitgemeten worden, wordt verduidelijkt in sectie 1.1.2. Na het uitmeten van deze hersengolven, kunnen ze door een BCI gestuurd worden om zo die externe toestellen aan te sturen.

1.1.2 Van hersensignaal tot controlesignaal

Het menselijk brein is een enorm ingewikkeld systeem dat zich in een tijd-ruimtelijke context beweeglijk gedraagt. Het meten van activiteit in het brein kan dan ook op verschillende manie-

ren. De meest gebruikte manier is zoals reeds aangehaald de niet-invasieve manier. Hierbij zijn er verschillende soorten methoden beschikbaar, zoals magneto-encefalografie (MEG) [15], near infrared spectroscopy (NIRS) [15], elektro-encefalografie (EEG) [15], etc. In deze thesis wordt er gebruik gemaakt van EEG-signalen om de analoge signalen van het brein op te vangen. EEG meet het elektrisch veld gecreëerd aan het oppervlak van de hoofdhuid, waardoor de elektrische potentiaalverschillen op de hoofdhuid uitgemeten kunnen worden. Zo kunnen de analoge signalen duidelijker gemeten worden [16].

Bij de meting wordt gebruik gemaakt van een conductieve gel, die geplaatst wordt tussen de hoofdhuid en de elektroden waarop gemeten zal worden (zie figuur 1.2 voor mogelijke elektroden locaties). Aangezien dit analoog signaal relatief zwak is (orde mV) doordat het door been en hoofdhuid gaat, moet het versterkt worden. Dit versterkte signaal zal vervolgens worden omgezet tot een digitaal signaal met behulp van een ADC (Analoog-Digitaal Omzetter) [17].



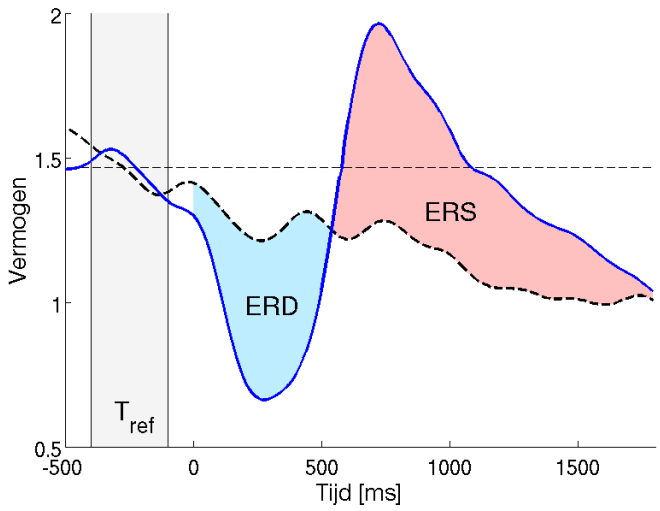
Figuur 1.2: Positie elektroden bij EEG-meting [18]

Tegenwoordig wordt er ook gebruik gemaakt van EEG-kappen waarbij gewerkt wordt met droge elektroden en er bijgevolg geen gel nodig is [19]. Hierdoor kan er snel begonnen worden met het uitmeten van hersenactiviteit. Bij EEG-kappen waarbij gebruik gemaakt wordt van conductieve gel, duurt het ongeveer 10 á 30 minuten tot soms 1 uur of langer om deze kap in gebruik te nemen. Deze tijd wordt bijna volledig besteed aan het aanbrengen van die gel. Algemeen wordt EEG het meest gebruikt vanwege de lage kost en verplaatsbaarheid [20].

Ondanks EEG een veel gebruikte methode is, heeft het ook een sterk nadeel. Het is namelijk zeer gevoelig aan ruis en artefacten, doordat het signaal zwak is. Het knipperen van de ogen of slikken kunnen de signalen enorm verstoren en bijgevolg de data moeilijk bruikbaar maken [21].

1.2 Ingebeelde beweging en toepassing

Bij het concept van ingebeelde beweging wordt het BCI systeem aangestuurd door middel van de gedachte dat er een beweging uitgevoerd wordt. Deze zogenaamde 'ingebeelde beweging' zorgt voor een opwekking van hersenactiviteit in de motorische cortex, die verantwoordelijk is voor de uitvoering van bewegingen. De mate van deze opwekking is als het ware vergelijkbaar met de impact indien de beweging fysiek uitgevoerd zou worden [22].



Figuur 1.3: Het vermogen van een EEG-kanaal in functie van de tijd. Bij beweging is er een laag vermogen, wat aangeduid is met ERD. Wanneer de beweging stopt, wordt de neurale oscillatie hersteld en wordt er een groter vermogen opgemeten. Dit is aangeduid met ERS. [23]

Wanneer een persoon zich in een rusttoestand bevindt, zullen neuronen synchroon met elkaar lopen in een repetitief patroon. Dit patroon wordt neurale oscillatie genoemd en reflecteert zichzelf in de alfa band van de hersenactiviteit die wordt opgemeten [24]. Dit fenomeen wordt ook wel het μ-ritme genoemd. Indien er zich nu echter een bewegingsvoorbereiding, -uitvoering of -inbeelding voordoet, zal dit synchroon patroon doorbroken worden. Dit wordt ERD (stimulus gerelateerde desynchronisatie) genoemd. Wanneer deze actie beëindigd wordt, zal de synchronisatie van de neuronen hersteld worden en zal de oscillatie in het EEG toenemen. Dit wordt

dan ERS (stimulus gerelateerde synchronisatie) genoemd [25]. In figuur 1.3 worden zowel ERD als ERS voorgesteld bij een opgemeten EEG-kanaal.

Om een onderscheid te maken tussen de beweging van linker of rechter lichaamsdelen, moet er bij elke beweging vastgesteld worden in welk deel van het brein de verandering in hersenactiviteit plaatsvindt. Beweging van een linker lichaamsdeel zal overeenkomen met activiteit in de rechterhersenhelft en vice versa [26]. Hoe dit onderscheid gebruikt wordt om een BCI aan te sturen, wordt verder toegelicht in sectie 2.5.

1.3 P300 en toepassing

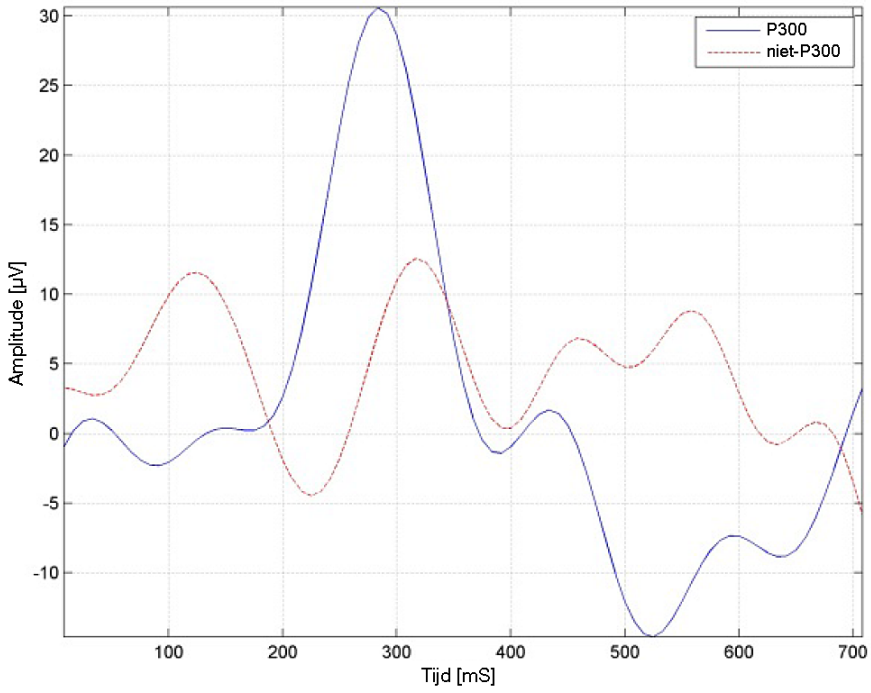
In tegenstelling tot BCI gebaseerd op ingebeelde beweging die zoeken naar onderbrekingen van ritme in EEG, zijn er ook BCI die zich richten op het zoeken naar patronen van gebeurtenissen in de hersenen. Deze neurosignalen worden stimulus gerelateerde potentialen (ERP) genoemd en worden omschreven als zintuiglijke prikkels die ontstaan door middel van elektrische stimulatie van de hoofdhuid [27].

Onder deze categorie van BCI valt de P300-speller. Dit is een BCI gebruikt voor het spellen van tekst. Hierbij wordt aan de gebruiker een scherm getoond met een raster van symbolen. Wanneer de gebruiker nu een bepaald symbool wil spellen, moet deze zich concentreren op dat ene symbool. Doordat symbolen of groepen van symbolen oplichten in willekeurige volgorde, zal de gebruiker een soort verrassingseffect krijgen wanneer het gewenste symbool oplicht. Dit effect wordt ook het oddball paradigma genoemd [28]. Hierdoor ontstaat er ongeveer 300 milliseconden na de stimulus een potentiaalpiek in de hersenen, die de P300-golfvorm genoemd wordt [29]. Door het opvangen van deze piek, kan men dus bepalen welk symbool gespeld wordt.

Één van de grote uitdagingen bij dit systeem is de afweging tussen snelheid en nauwkeurigheid. Dit reflecteert zich in de twee verschillende manieren van oplichting. Ofwel kunnen enkele symbolen apart na elkaar oplichten (single-character), ofwel kunnen rijen of kolommen van symbolen oplichten. Bij dit eerste zal duidelijker opgevangen kunnen worden welk symbool de gebruiker wil spellen, maar duurt het langer tot dit ene symbool opgelicht wordt. Bij het oplichten van rijen en kolommen zal het gewenste symbool sneller aan bod komen, maar zal het minder snel duidelijk worden over welk symbool het gaat [30].

Stel als wijze van voorbeeld dat een gebruiker een P300-speller gebruikt met een 6 x 6 raster van symbolen. Indien gewerkt wordt met single-character belichting zal er slechts 1 kans op 36 zijn dat het gewenste symbool opgelicht wordt. Wanneer gewerkt wordt met rij-kolom gebaseerde belichting wordt de kans verhoogt naar 2 op 12. Deze kans wordt bekomen doordat een symbool eens in een rij en eens in een kolom opgelicht dient te zijn voordat geweten wordt over welk symbool het gaat.

Om een systeem te bouwen rond deze P300-golfvorm, moet deze efficiënt gedetecteerd kunnen worden. Hiervoor wordt er gebruik gemaakt van een P300 en niet-P300 signaal. In figuur 1.4 wordt een voorbeeld van deze signalen weergeven.



Figuur 1.4: EEG-kanaal van een stimulus [18]

Wanneer deze golfvorm gedetecteerd wordt, kan er bepaald worden welk symbool er 300 milliseconden eerder op het bord opgelicht werd. Op deze manier kunnen woorden en uiteindelijk ook zinnen gespeld worden. Het is dus belangrijk om goede detectie te hanteren, zodat het aantal fouten minimaal is. Om een gewenste output te bekomen aan de hand van die data, wordt gebruik gemaakt van een getrainde classifier. Deze gebruikt een set van voorbeeldsignalen om P300-golfvormen te herkennen. Hoe met deze voorbeeldsignalen een classifier gebouwd

kan worden, wordt verder uitgelegd in sectie 2.4.

1.4 Probleemstelling

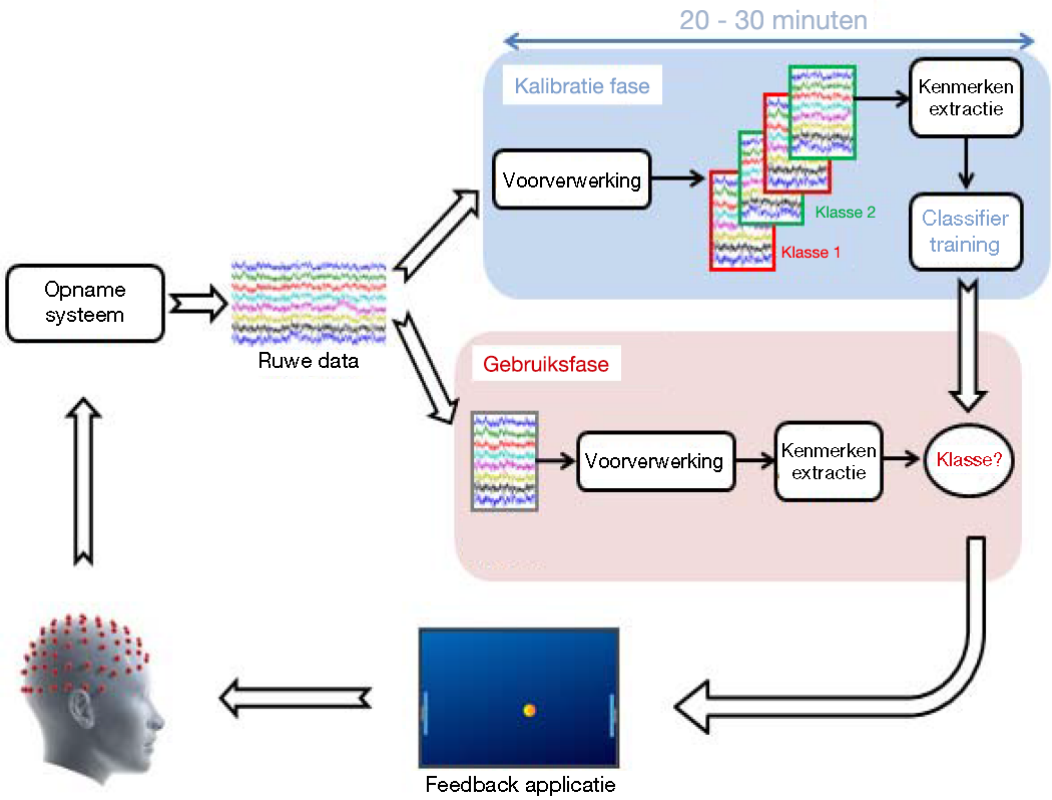
Brein-computer interfaces kunnen enorm krachtige systemen zijn, maar bevatten nog steeds een groot nadeel. Het systeem moet namelijk persoonsgebonden geconfigureerd worden, zelfs over verschillende sessies met éénzelfde persoon. Dit komt doordat er een grote inter-subject variabiliteit aanwezig is tussen verschillende gebruikers. De reden waarom een BCI ook over verschillende sessies gekalibreerd moet worden, kan omschreven worden door één of een combinatie van de volgende [31]: 1) Intra-subject variabiliteit, door veranderingen in de gemoedstoestand van de gebruiker per sessie; 2) Fysiologische artefacten, door het knipperen van de ogen, bewegen van spieren of ademhaling; 3) Instrumentale artefacten, door veranderingen in de positie van elektroden of hun impedantie tijdens het opnemen van signalen.

Daar het kalibreren van een BCI veel tijd en concentratie van de gebruiker vergt, moet deze zo laag mogelijk gehouden worden. In deze sectie wordt verder toegelicht hoe de kalibratietijd verminderd kan worden en wordt een mogelijkheid tot verbetering uitgelegd.

1.4.1 Classificeren van hersensignalen

Wanneer de data uitgemeten en verwerkt is (zie sectie 2.3), moet deze door een classifier gestuurd worden om een bepaalde actie uit te voeren. Zo zal een classifier voor ingebeelde beweging bijvoorbeeld beslissen of het om een linker of rechter beweging gaat. Deze classifier is echter persoonsgebonden en moet dus gekalibreerd worden. Dit wordt gedaan aan de hand van een kalibratieset. De gebruiker wordt hierbij gevraagd om bepaalde handelingen uit te voeren (zoals het inbeelden van een beweging), terwijl EEG-data verzameld wordt. Voor een systeem die gebruik maakt van ingebeelde beweging, bestaat deze kalibratiesessie bijvoorbeeld uit het tonen van een links of rechts wijzende pijl aan de gebruiker, waarbij de gebruiker zich vervolgens de overeenkomstige beweging inbeeld. Dit wordt gemiddeld 20 á 30 minuten gedaan, vooraleer de BCI bruikbaar is in ware tijd [32]. In Figuur 1.5 wordt het verschil tussen kalibratie en gebruik in ware tijd verduidelijkt.

Het feit dat het systeem een bepaalde kalibratie nodig heeft, vergt veel geduld en inspanning van de gebruiker. Deze wil namelijk het systeem zo snel mogelijk in ware tijd kunnen gebrui-



Figuur 1.5: Patroon classificatie in brein-computer interfacing [33]

ken. Omdat deze procedure juist zo tijdrovend en intensief is, zijn onderzoekers op zoek naar alternatieve methoden om de kalibratietijd te verminderen, met als uiteindelijk doel de systemen onmiddellijk in ware tijd te kunnen gebruiken. Één van de mogelijkheden om deze tijd te reduceren is transfer learning.

1.4.2 Transfer learning

Transfer learning is een veel onderzochte methode om kalibratietijd te verminderen. Het is een methode die zich focust op het gebruiken van kennis opgedaan bij eerdere gebruikers, om deze vervolgens op één of andere manier toe te passen op een nieuwe gebruiker. Kort gezegd laat transfer learning het gebruik van vooropgenomen datasets toe, om de kalibratie van de huidige BCI te versterken. Hierdoor kan de grootte van de kalibratieset van de nieuwe gebruiker verkleint worden, met als gevolg dat de kalibratietijd in kwestie gaat dalen.

Transfer learning kan zowel toegepast worden tussenin sessies van éénzelfde persoon als tus-

sen verschillende personen. Dit eerste wordt session-to-session transfer learning genoemd [34]. Deze methode van werken gebruikt vooropgenomen datasets van dezelfde gebruiker, maar uit voorgaande sessies. Dit lost een veel voorkomend BCI paradigma op, waar er veel sessies zijn per gebruiker, maar weinig data beschikbaar is per sessie.

Een tweede vorm van transfer learning is subject-to-subject transfer learning. Deze zorgt ervoor dat de informatie genomen uit data van eerdere gebruikers herbruikt kan worden [34]. In deze thesis wordt de focus vooral gelegd op deze vorm van transfer learning. Enkele algoritmen hieromtrent worden verder besproken in sectie 2.6.1. Het is ook belangrijk om weten dat niet alle eerdere gebruikers even compatibel zijn met de nieuwe gebruiker. Het zoeken naar de eerdere gebruiker die het meest op de nieuwe gebruiker lijkt, is dan ook een groot deel van de onderzoeksvraag van deze thesis.

1.5 Doelstelling van de Thesis

Het vergt veel tijd van de gebruiker om een brein-computer interface te personaliseren en optimaliseren voordat het klaar is voor gebruik in ware tijd. Het is nu net deze kalibratietijd die door middel van subject-to-subject transfer learning geminimaliseerd probeert te worden. In deze thesis wordt bestudeerd hoe een brein-computer interface uitgebouwd kan worden voor gebruik met ingebeelde beweging door middel van vooropgenomen data van vele andere testpersonen, in combinatie met weinig kalibratie data van de gebruiker. Hierbij wordt de data van vele eerdere gebruikers genomen met voor elke eerdere gebruiker een beperkte datahoeveelheid. Ook wordt onderzocht welke statistische parameters het meest geschikt zijn om een eerdere gebruiker mee te selecteren die het meest overeenkomstig blijkt om een BCI mee uit te bouwen voor de nieuwe gebruiker.

In het volgende hoofdstuk zullen de gebruikte datasets en methodologieën aangehaald worden die gebruikt werden tijdens deze thesis. In hoofdstuk 3 zal vervolgens een zelf ontworpen subject-to-subject transfer learning algoritme besproken worden. Hoofdstuk 4 bevat de bekomen resultaten en bevindingen tijdens het verloop van de thesis. Tot slot wordt in hoofdstuk 5 een besluit gevormd over het verrichte werk.

Hoofdstuk 2

Materialen & methoden

2.1 Inleiding

In dit hoofdstuk worden enkele technieken en methoden besproken die gebruikt worden bij het opstellen van een brein-computer interface. Hieronder wordt nog eens de algemene werking (zie figuur 1.1) overlopen hoe van opgewekte hersengolven overgegaan wordt naar de eigenlijke besturing van een computersysteem.

Informatie ophalen De basis van het systeem ligt bij de hersenen, dus moeten er hersengolven opgevangen worden om te kunnen verwerken. Zoals eerder vermeld in het eerste hoofdstuk worden deze in de opstellingen geregistreerd met EEG-signalen. Voor meer informatie hieromtrent wordt er verwezen naar sectie 1.1.2.

Informatie verwerken De EEG-signalen moeten vervolgens verwerkt worden. Een groot deel hiervan is besteed aan het het wegfilteren van de onnodige delen van het signaal. Er wordt enkel doorgegaan met de specifieke delen en tijden die van belang zijn voor het systeem. Een goede voorverwerking kan er namelijk voor zorgen dat de BCI een stuk efficiënter werkt. Specifieke stappen die genomen worden tijdens de voorverwerking worden verder toegelicht in sectie 2.3.

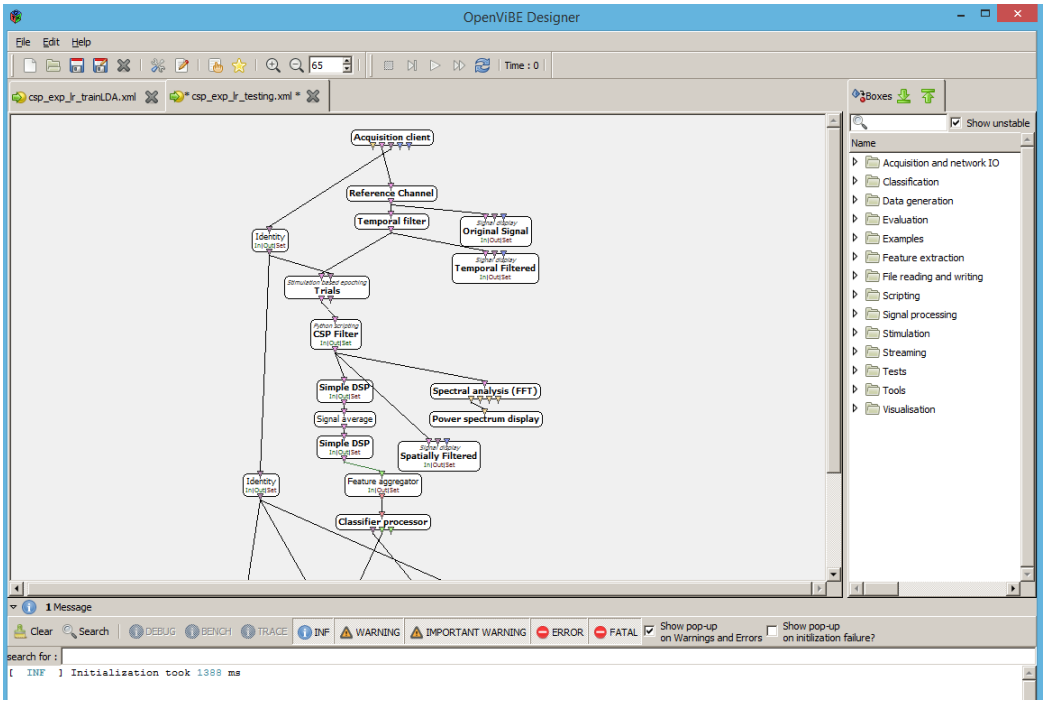
Informatie classificeren Ten laatste worden de verwerkte signalen door een classifier gestuurd. Deze is reeds op voorhand getrained en geoptimaliseerd om met de gebruiker samen te werken. De verschillende uitgangen van de classifier kunnen vervolgens gekoppeld worden aan bepaalde acties, om zo de gebruiker interacties te laten uitvoeren met de omgeving. Meer uitleg omtrent classifiers en machinaal leren in het algemeen wordt gegeven in sectie 2.4.

2.2 Software en data

Voor het opstellen en testen van verschillende BCI modellen wordt er in deze thesis gebruik gemaakt van verschillende soorten software en datasets. Dewelke nu juist gebruikt worden en binnenin welke context van deze thesis, wordt toegelicht in de volgende subsecties.

2.2.1 OpenVibe

Om verscheidene opstellingen van BCI systemen voor te stellen, wordt er gebruik gemaakt van de high-level programmeeromgeving OpenVibe [35]. Dit maakt het mogelijk om snel en efficiënt veranderingen aan te brengen in bestaande systemen. OpenVibe maakt gebruik van vooropgestelde codeblokken om codetaal schematisch voor te stellen (zie figuur 2.1). Parameters worden vervolgens geoptimaliseerd per blok. OpenVibe maakt het ook mogelijk om signalen en andere spectra grafisch te weergeven, waardoor feedback per tussenstap weergeven kan worden. Door gebruik te maken van de programmeeromgeving Python kunnen aangepaste codeblokken gecreëerd worden, waardoor het mogelijk wordt om eigen algoritmen te verwerken in het blokschema.



Figuur 2.1: Grafische gebruiksomgeving (GUI) van het OpenVibe softwarepakket

2.2.2 Python

Daar OpenVibe gebruikt wordt om BCI systemen in ware tijd voor te stellen, wordt in het kader van deze thesis vooral gebruik gemaakt van Python [36] om parameter optimalisatie door te voeren bij gesimuleerde BCI. Python is een script-gebaseerde programmeertaal, waarvoor een hele waaier aan bibliotheken beschikbaar zijn. Twee van de meest gebruikte bij dit onderzoek zijn NumPy [37] (voor numerieke analyse op matrices) en scikit-learn [38] (reeds geïmplementeerde vormen van algoritmen omtrent machinaal leren). Door in Python het werkende BCI systeem gemaakt in OpenVibe na te bouwen, kan er bijvoorbeeld een script geschreven worden om de gemiddelde train en test accuraatheid te berekenen per parameter. Het wordt zo veel eenvoudiger om simulaties uit te voeren op grote schaal, waardoor vergelijkingen gemaakt kunnen worden tussen verschillende systemen en over verschillende parameters.

2.2.3 Gebruikte datasets

De onderzoekers achter OpenVibe bieden ook open source datasets aan, gebaseerd op ingebeelde beweging. Deze datasets zijn afkomstig van éénzelfde testpersoon en werden opgenomen binnen een tijdspanne van twee weken. Per dataset zijn er 20 linker en 20 rechter trials aanwezig. De EEG-data werd gesampled op 512 Hz en slechts 10 elektroden werden opgenomen (5 voor de linkerhersenhelft en 5 voor de rechterhersenhelft) [39].

Een alternatieve groep datasets die gebruikt werd, zijn afkomstig van de eerste groep van BCI competitie IV [40]. Deze biedt 7 uitgebreide datasets aan van verschillende personen die aan ingebeelde beweging doen. Hier wordt gebruik gemaakt van 100 linker en 100 rechter trials. De EEG data werd gesampled op 100 Hz en een uitgebreid aantal van 59 elektroden werd opgenomen. Zowel deze zeven datasets als degene van OpenVibe werden gebruikt bij de initiële opstelling van een BCI voor het verwerken van ingebeelde beweging.

Om een groter aantal aan proefpersonen te verkrijgen voor gebruik bij transfer learning, werd er gebruik gemaakt van 109 datasets die verkrijgbaar zijn bij PhysioNet [41] [42]. Elke dataset bevat drie opnames van een persoon, die gemiddeld twee minuten lang aan ingebeelde beweging doet. Deze data werd opgenomen over 64 verschillende elektroden en werd gesampled op 160 Hz.

Hoewel het uiteindelijk doel is om mensen te helpen die geparalyseerd zijn, werden voorgaande

datasets opgenomen bij gezonde mensen. Dit heeft echter een invloed op onze resultaten. Het is namelijk bewezen dat geparalyseerde gebruikers significant slechter scoren dan gezonde gebruikers [43]. Desondanks wordt toch gebruik gemaakt van datasets opgenomen op gezonde mensen, doordat deze in een grotere hoeveelheid beschikbaar zijn.

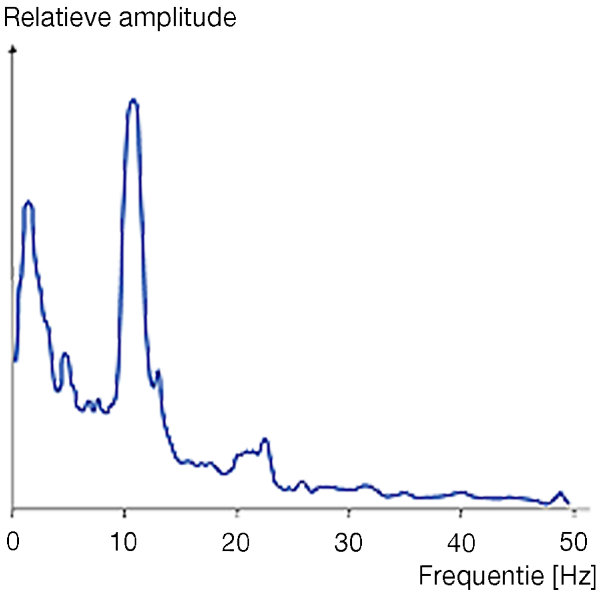
2.3 Verwerken van EEG-signalen

Het voorverwerken van EEG-signalen is een enorm belangrijk aspect van het BCI gebeuren. Zonder de juiste informatie uit de signalen te halen, is er een overvloed aan data en zou de classifier nooit in staat zijn om nieuwe signalen goed te beoordelen. Voorverwerking wordt hier gedaan op twee verschillende niveaus. Eerst en vooral wordt er gefilterd in het frequentiedomein, zodat irrelevante informatie niet verder meegenomen wordt. Vervolgens wordt de data spatiaal gefilterd om een beter onderscheid te kunnen maken tussen de verschillende klassen. Hieronder wordt in detail uitgelegd welke methoden er gebruikt worden bij de voorverwerking.

2.3.1 Frequentie filtering

Wanneer EEG-signalen voor de eerste maal bekeken worden, wordt er opgemerkt dat ze meestal opgenomen werden in een relatief breed frequentie spectrum. Er is echter niet evenveel nuttige informatie aanwezig in alle frequentiebanden. Uit onderzoek blijkt, dat de meeste informatie gerelateerd tot ingebeelde beweging zich bevindt in de alfa (8 - 12Hz), beta (14 - 18Hz) en gamma (36 - 40Hz) banden van de motorische cortex [44]. De belangrijkste van deze drie is de alfa band, omdat deze overlapt met de regio waarin het μ-ritme gevonden wordt (zie figuur 2.2).

Het patroon onderzoeken van dit ritme is van fundamenteel belang bij het bouwen van een systeem voor detectie en classificatie van ingebeelde beweging. Bij rusttoestand gaan de neuronen in de hersenen met elkaar synchroon proberen lopen. Dit doen ze door pulsen af te geven en deze op elkaar af te stemmen (ERS). Het ritme wordt echter verstoord (ERD) wanneer een persoon zich een bepaalde beweging inbeeld of uitvoert. Door de juiste locatie van deze stimulus te vinden, kan achterhaald worden in welke hersenhelft de activiteit plaats vindt. Hierdoor kan er gezegd worden dat de mate van aanwezigheid van het μ-ritme omgekeerd evenredig is met de motorische activiteit in dat deel van de hersenen [25]. Doordat er geweten is dat activiteit in de linker hersenhelft overeenkomt met een beweging uitgevoerd door het rechter deel van het lichaam en vice versa, kan er besloten worden of het om een linker of rechter beweging gaat [24].



Figuur 2.2: Momentopname van een EEG-signaal bij ERS in het frequentiedomein [45]. Rond de 10Hz zien we een duidelijke piek in activiteit, wat de aanwezigheid van het μ-ritme aantoont.

Om de datahoeveelheid van het EEG-signaal te verminderen, wordt het signaal door een banddoorlaatfilter gestuurd. Dit zal ervoor zorgen dat enkel de gewenste frequentieband doorgelaten wordt en dat alle overbodige frequenties weggefilterd zullen worden. Als type banddoorlaatfilter wordt in deze thesis een butterworth filter gebruikt van orde 4. Deze zorgt ervoor dat er geen afsnijding verkregen wordt aan de randen van het filterbereik, maar dat deze geleidelijk zullen afzwakken. Dit voorkomt het Gibbs fenomeen [46], wat betekent dat een verticale afsnijding van het signaal zou zorgen voor een lichte sinusoïdale uitwijking.

2.3.2 Spatiale filtering

Daar EEG over meerdere kanalen een enorme hoeveelheid aan data opneemt, is het heel moeilijk om er een bepaald patroon in te vinden. Het is dus van uiterst belang dat er een bepaald algoritme gebruikt wordt die ervoor kan zorgen dat orde gezien wordt in de chaos. Dit wordt gedaan aan de hand van spatiale filters, die meer gewicht geven aan bepaalde regio's in de hersenen waar er meer informatie aanwezig is. Door middel van deze spatiale filters wordt het mogelijk om verschillende handelingen te onderscheiden in de data [47]. Enkele verschillende algoritmen die gebruikt worden in BCI zijn Common Spatial Patterns (CSP) [48], Principle Component Analysis (PCA) [48], Common Average Referencing (CAR) [48], Surface Laplacian

(SL) [48], adaptive filtering [48] en Independent Component Analysis (ICA) [48]. In deze thesis wordt er gebruik gemaakt van SL en CSP, die hieronder in detail verduidelijkt worden.

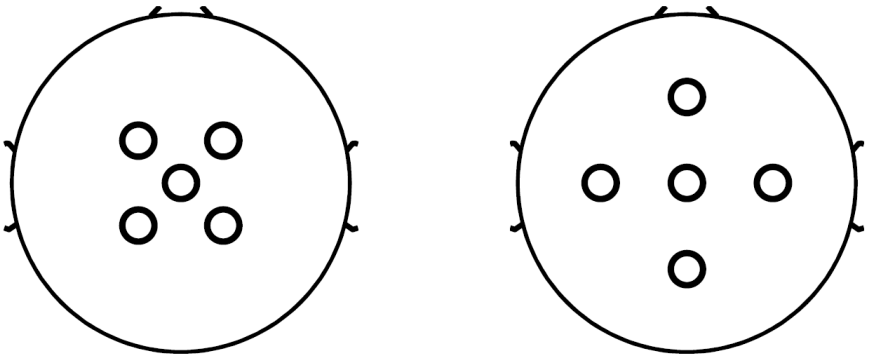
Surface Laplacian

De Surface Laplacian [48] is een relatief eenvoudige spatiale filter, die goed samenwerkt met een BCI. Hierbij wordt de gemiddelde waarde van de omgevende kanalen rond een centraal kanaal van de waarde van dat centrale kanaal afgetrokken. Deze omgevende kanalen bevinden zich hierbij op éénzelfde afstand van het centrale kanaal. Indien deze afstand klein is, spreken we van de Small Laplacian, wat de meest gebruikte Laplacian-filter is. Hierbij worden de dichtst nabije buren gebruikt als omgevende kanalen. Als centraal kanaal wordt voor de linker hersenhelft C3 gebruikt (zie formule 2.1) en voor de rechter hersenhelft C4 (zie formule 2.2).

$$C3^{LAP} = C3 - \frac{1}{4}(C1 + C5 + FC3 + CP3) \tag{2.1}$$

$$C4^{LAP} = C4 - \frac{1}{4}(C2 + C6 + FC4 + CP4) \tag{2.2}$$

Wanneer de omgevende elektroden echter wat verder genomen worden, wordt er gesproken van de Large Laplacian. Waar nu net het verschil ligt tussen beide filters wordt getoond in figuur 2.3 en wordt grondig onderzocht in [49]. Het grote voordeel van de Surface Laplacian is dat het geen nood heeft aan een referentie elektrode, doordat het de signaal-ruisratio (SNR) al goed verbetert [48]. Het gebruikt ook een beperkt aantal elektroden, waardoor het ook robuust is tegen artefacten die elders zouden kunnen opduiken.

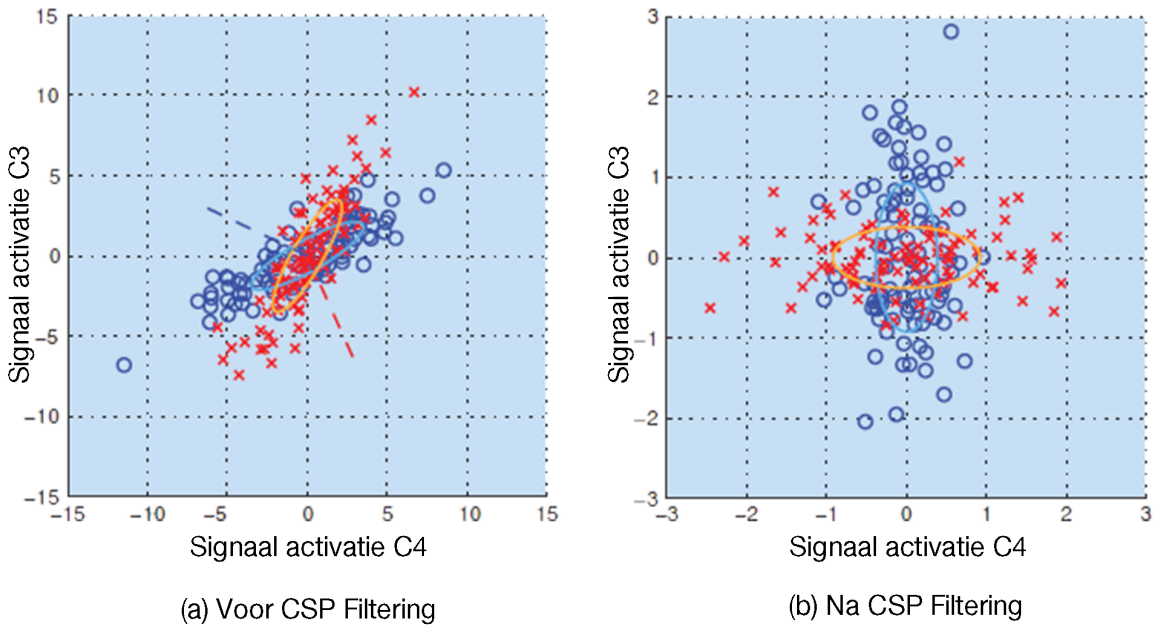


(a) Small Laplacian. (b) Large Laplacian.

Figuur 2.3: De verschillen tussen de twee Laplacian-filters. Bij de Large Laplacian liggen de kanalen die men aftrekt van het centrale kanaal verder weg [16].

Common Spatial Patterns

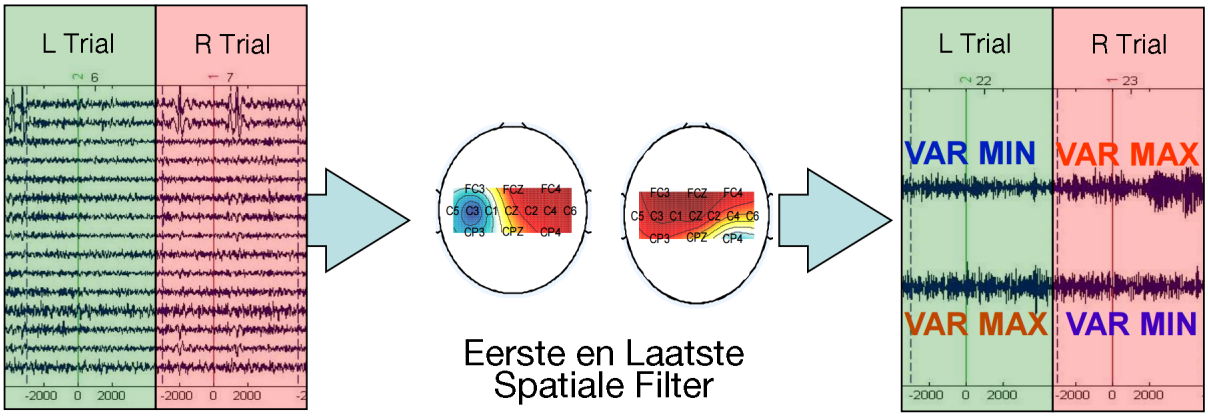
Het Common Spatial Pattern [48] algoritme, ook wel CSP genoemd, is een vaak gebruikte spatiale filter bij BCI [50]. Het wordt gezien als een lineaire transformatie, dat de variantie van de ene klasse maximaliseert, terwijl het de variantie van de andere minimaliseert. Dit kan gezien worden in figuur 2.4. Voor de CSP filtering kan opgemerkt worden dat er een licht verschil in variantie is tussen beide klassen. Er kan echter geen duidelijk onderscheid gemaakt worden tussen deze klassen doordat ze gecorreleerd zijn. Om dit op te lossen wordt CSP gebruikt, met als doel de variantie van de klassen duidelijk te kunnen meten. De sterkte correlatie tussen de klassen wordt hierdoor verminderd, waardoor een goed onderscheid gemaakt wordt tussen beide klassen.



Figuur 2.4: Een voorbeeld van CSP filtering, voorgesteld in 2D, door een scatterplot over C3 en C4 van een links ingebeelde beweging. In de linker afbeelding (a) duiden de blauwe cirkels en rode kruisjes de verschillende gaussiaans verdeelde distributies aan van samples binnenin één enkele trial. De twee ellipsen geven een schatting van de covariantie van die klassen en de streepjeslijnen duiden de richting van de CSP projecties aan. In de rechter afbeelding (b) is er een duidelijke herverdeling te merken van de data. Het onderscheid tussen beide klassen is dan ook duidelijker merkbaar [33].

Dit komt uiterst goed van pas bij het bouwen van een BCI voor ingebeelde beweging, aangezien hierbij gewerkt wordt met ERD en ERS. Wanneer er een (ingebeelde) beweging ingezet wordt, zal het μ-ritme in een bepaald deel van de hersenen onderbroken worden, waardoor er ERD plaatsvindt. Hierdoor zal het vermogen in die regio veel lager liggen. Bijgevolg kan door CSP filtering een veel duidelijker onderscheid gemaakt worden of die regio zich in de linker of rechter hersenhelft bevindt, waardoor de aard van de beweging achterhaald kan worden.

Door deze manier van scheiden, kan de gewichtenmatrix van het Common Spatial Pattern bepaald worden. Deze bevat de nodige informatie om de data te transformeren naar de nieuwe vorm. Elke kolom van deze gewichtenmatrix bevat de filtercoëfficiënten van een unieke spatiale filter. Hierbij is er één spatiale filter beschikbaar per gegeven ingangskanaal. Bij de gewichtenmatrix bevatten de uiterste kolommen het meeste informatie, aangezien deze de data zodanig transformeren dat ze een maximale scheiding bekomen in zake van variantie (zie figuur 2.5).



Figuur 2.5: Voorstelling van linker en rechter trial EEG-data voor en na het CSP filteren van de data door middel van twee spatiale filters, opgesteld met de uiterste filtercoëfficiënten van de gewichtenmatrix [51].

Het kiezen van een beperkt aantal filterparen, zodat de dimensionaliteit gereduceerd wordt en tegelijk de overdracht van informatie maximaal blijft, is vaak voorkomend in het BCI gebeuren. 1, 2 of 3 filterparen zijn gebruikelijke aantallen. Om het belang van deze dimensionaliteitsreductie te benadrukken, wordt het volgende voorbeeld gegeven. Stel een opgenomen EEG-signaal van 64 kanalen, gesampled aan 100 Hz met een duur van 4 seconden per trial. Dit betekent dat er 25600 kenmerken aanwezig zijn bij één enkele trial voor de spatiale filtering plaats vindt.

Door het gebruik van CSP kunnen deze echter gereduceerd worden tot bijvoorbeeld 2, 4 of 6 kenmerken, wat een enorm verschil is.

Daar CSP een veelgebruikte techniek is, heeft het toch enkele beperkingen. Een eerste beperking is het feit dat het CSP algoritme helemaal geen weet heeft van de neurofysiologische achtergrond van de hersenen. Hierdoor weet het algoritme niet of het de juiste varianties aan het maximaliseren is. Het gevolg hiervan is dat CSP heel erg gaat overfitten indien het te maken heeft met veel ruis of kleine datasets. Meer informatie omtrent overfitten wordt gegeven in sectie 2.4.3. Bij kleinere datasets zal de ruis namelijk niet goed verdeeld zijn over beide klassen en zal het algoritme het contrast te specifiek gaan zoeken tussen die twee klassen, waardoor de CSP filters slecht scoren bij gebruik met nieuwe data. Bij grotere datasets is dit minder een probleem, doordat de ruis en onnuttige varianties verdeeld zullen zijn over beide klassen. Indien de data van de kleine dataset echter heel zuiver is, kan het ook gebruikt worden in combinatie met CSP. Spijtig genoeg is dit bijna nooit het geval wanneer er gewerkt wordt met EEG.

Een tweede beperking van dit algoritme is dat het slechts twee klassen kan onderscheiden van elkaar of in de context van ingebeelde beweging slechts twee bewegingen. Er bestaat echter een variatie op dit algoritme die multi-class CSP genoemd wordt [52]. Hier werd echter geen verder onderzoek naar gedaan, doordat er hier bij ingebeelde beweging enkel het onderscheid gemaakt wordt tussen linker en rechter beweging.

2.4 Machinaal leren

2.4.1 Algemeen

Machinaal leren is een veel besproken en onderzocht topic. Het maakt nieuwe functionaliteiten mogelijk voor hedendaagse technologie, door deze artificiële intelligentie aan te bieden [53]. Het dient als een input-output model, waar het model zichzelf dingen aanleert aan de hand van de gegeven input. Zonder dat gebruikers het beseffen, gebruiken ze dagelijks tientallen systemen die algoritmen omtrent machinaal leren gebruiken. Wanneer bijvoorbeeld een Google zoekopdracht uitgevoerd wordt, zullen de beste en meest gewenste resultaten bovenaan weergeven worden. Dit komt doordat Google een algoritme geïmplementeerd heeft, dat kan inschatten wat het belang is van de verschillende zoekresultaten en hoe het die moet sorteren voor weergave aan de

gebruiker. Een ander voorbeeld van machinaal leren in het dagelijkse leven is de classificatie van spam mails. Hierbij gaat een achterliggend systeem verschillende kenmerken van spam aanleren en vervolgens de ontvangen mails onderzoeken op basis van die kenmerken. Indien het de mail aanziet als spam, wordt het rechtstreeks naar de spam folder verplaatst.

In de volgende subsecties worden de verschillende soorten machinaal leren besproken, enkele problemen beschreven omtrent het gebruik ervan en wordt een zeer krachtige methode aangehaald die gebruikt wordt doorheen het volledige verloop van de thesis.

2.4.2 Soorten machinaal leren

Er worden twee grote types machinaal leren onderscheid, genaamd supervised learning en unsupervised learning. Beide worden in deze sectie verder toegelicht.

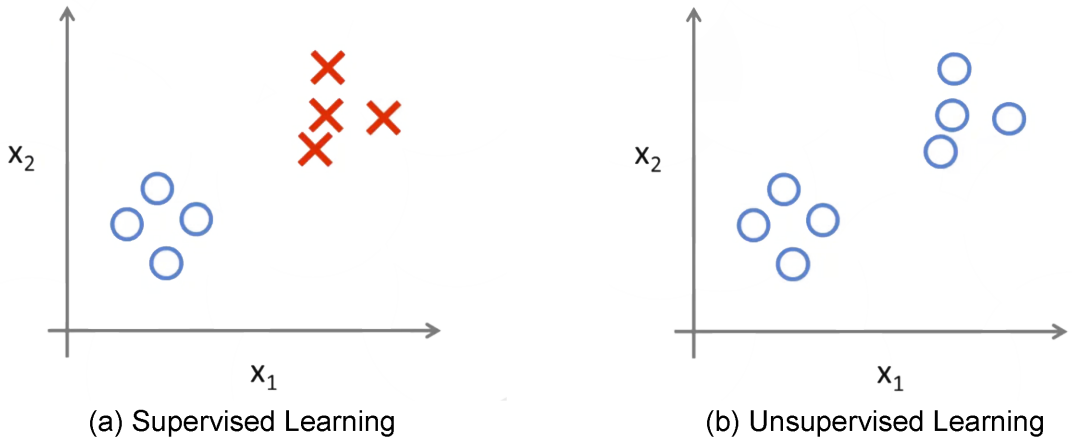
Supervised learning

Supervised learning is wellicht de meest gebruikte vorm van machinaal leren. Hierbij wordt het input-output model getrained aan de hand van gegeven input-output paren. Het systeem verkrijgt hierdoor enig inzicht in het verband tussen die input-output, waardoor het nieuwe data kan onderzoeken met behulp van de opgedane kennis [53]. Er is dus nood aan een reeds bestaande dataset met relevante data, waarbij zowel de nodige kenmerken (inputs) als de uitkomst ervan (outputs) gegeven wordt. Wanneer de uitkomst van de data meegegeven wordt, spreekt men van gelabelde data.

Een voorbeeld waar supervised learning gebruikt wordt, is bij het opzetten van een spam filter. Eerst en vooral zal een verzameling gemaakt worden van mails die geclassificeerd worden als spam en niet-spam. Vervolgens zal het systeem alle mails onderzoeken en zichzelf aanleren waar nu net het onderscheid gemaakt wordt tussen een spam mail en een niet-spam mail. Wanneer hierna dit systeem gebruikt wordt op nieuwe toegekomen mails, zou het in staat moeten zijn om deze mails aan de hand van de opgedane kennis te classificeren als spam of niet-spam.

Op figuur 2.6 wordt visueel aangehaald wat het verschil is tussen supervised en unsupervised learning. Bij beide werd éénzelfde dataset gebruikt. Bij supervised learning werd de data echter gelabeld, terwijl dit niet gedaan werd bij unsupervised learning. Doch zou via clustering

algoritmen eenzelfde onderscheid gemaakt moeten worden bij de data in unsupervised learning als bij supervised learning.



Figuur 2.6: Het verschil tussen supervised en unsupervised learning. Bij supervised learning (a) worden twee klassen (blauwe cirkels en rode kruisjes) onderscheid. Dit duidt aan dat de data gelabeld is en de uitkomsten dus meegegeven worden met de data. Bij unsupervised learning (b) wordt niet-gelabelde data meegegeven, waardoor er initieel geen onderscheid gemaakt kan worden in de data. Hierdoor wordt de data enkel weergeven met blauwe cirkels [53]

Unsupervised learning

De tweede vorm machinaal leren is unsupervised leren. Hierbij leert de computer zichzelf aan hoe hij iets moet doen, om vervolgens patronen en structuren in de data te gaan ontdekken [53]. Het input-output model leert zichzelf hier dus geen verbanden aan door het gebruik van voorbeelddata. Bij unsupervised learning wordt er gewerkt met cluster algoritmen. Deze zorgen er voor dat het systeem verschillende datagroepen van elkaar kan onderscheiden om deze vervolgens te kunnen indelen in groepen of clusters.

Een voorbeeld waar dit in gebruik genomen wordt is Google News. Het algoritme hierachter gaat op het web zoeken naar allerlei nieuws gerelateerde artikelen. Vervolgens wordt hierop een unsupervised learning algoritme los gelaten, zodat deze de nieuws artikelen kan groeperen per onderwerp. Op deze manier kan de gebruiker meer informatie verkrijgen over éénzelfde onderwerp of kan deze het verhaal op een alternatieve manier bekijken.

Een veel gebruikt algoritme van dit type machinaal leren is k-Means clustering [54]. Aangezien het onderzoekstopic van deze thesis ligt bij transfer learning dat gebruik maakt van supervised learning, werd geen verder onderzoek gedaan naar unsupervised learning algoritmen.

2.4.3 Vaak voorkomende problemen

De keuze om classificatie te gebruiken brengt ook enkele problemen met zich mee. Hieronder worden twee vaak voorkomende problemen besproken waar zeker rekening mee gehouden moet worden bij het opstellen van een classifier.

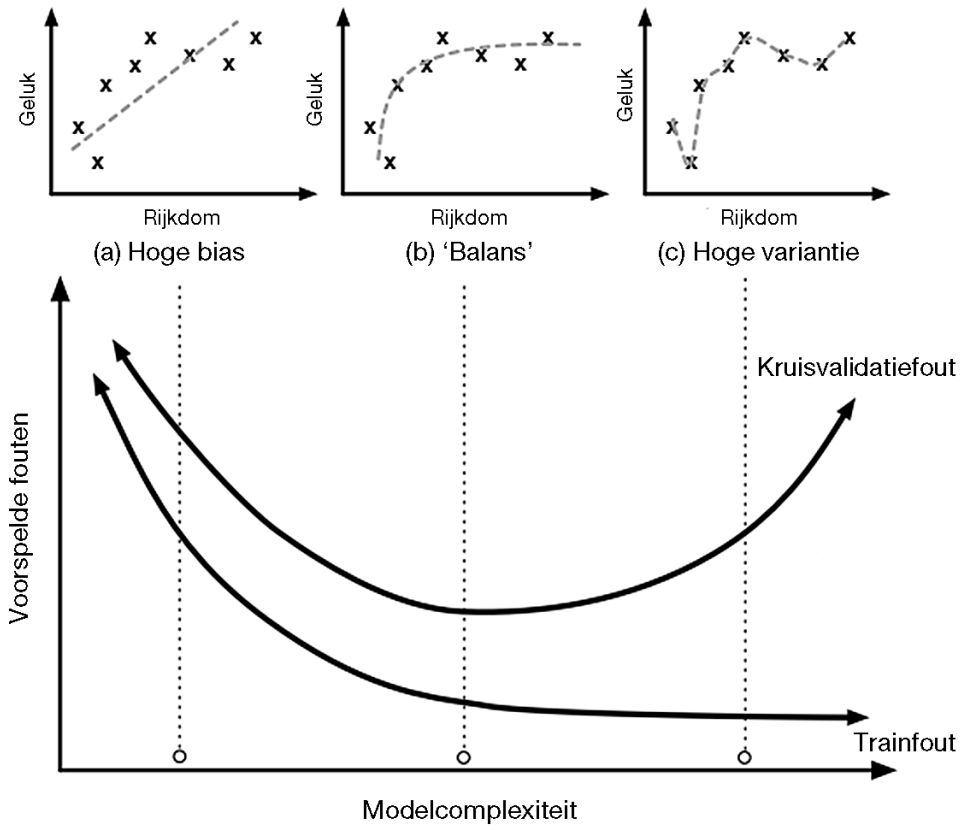
Curse of dimensionality

Het eerste probleem is de curse of dimensionality [55]. Hierbij wordt aangehaald dat de meeste algoritmen omtrent machinaal leren niet goed presteren wanneer de dimensionaliteit hoog is ten opzichte van het aantal voorbeelden. Wanneer dit het geval is, kunnen de algoritmen heel moeilijk gaan zoeken naar de relevante informatie in die data, waardoor het moeilijk is om het onderscheid te kunnen maken tussen verschillende klassen. Algemeen wordt aangenomen dat voor eenvoudige classifiers als Linear Discriminant Analysis [55], het aantal training samples op zijn minst tien keer zo groot moet zijn als het aantal gebruikte kenmerken bij de samples [56]. Overdimensionaliteit kan bijgevolg minimaal aangepakt worden door een groter aantal trials te voorzien. Hierdoor ontstaat echter een ander probleem. Een grotere hoeveelheid data verwerken betekent een lagere communicatie snelheid, waardoor de classifier misschien niet snel genoeg zal werken in ware tijd. Een goede voorverwerking van de data zodat een beperkt aantal kenmerken gebruikt wordt, is dus van cruciaal belang om de algoritmen omtrent machinaal leren eenvoudiger toe te laten de relevante informatie te vinden in de data.

Bias-variantie afweging en kruisvalidatie

Een tweede vaak voorkomend probleem is de bias-variantie afweging [55]. Variantie op data staat maat voor de gevoeligheid van het systeem aan uitschieters. De bias op data duidt aan hoe gevoelig de fout op patroonherkenning mag zijn. Een hoge bias betekent dus een slechte patroonherkenning, wat underfitting genoemd wordt. Het probleem is echter, dat je om een lage bias te verkrijgen, een hoge variantie nodig hebt. Hierdoor wordt het systeem echter weer gevoeliger voor uitschieters, wat overfitting genoemd wordt. Een goeie afweging tussen de twee is dus van cruciaal belang bij het opzetten van een BCI.

Om een goeie keuze te maken in model parameters, zodat een hoge bias of varantie voorkomen kan worden, wordt er gebruik gemaakt van kruisvalidatie. Hierbij wordt de beschikbare train-data van een gebruiker in k gelijke delen opgedeeld (ook wel folds genoemd). Deze test wordt hierdoor ook de *k-fold* test genoemd. Vervolgens wordt de classifier getrained op $k-1$ folds en getest op de éne overige fold. Dit wordt k-keer herhaald en vervolgens uitgemiddeld om een beeld van de prestatie van die classifier te verkrijgen. Door modellen op deze manier te testen met verschillende parameters, kan gezocht worden naar het model met de hoogste prestatie bij kruisvalidatie of de kleinste kruisvalidatiefout. De bias-variantie afweging met kruisvalidatie- en trainfout wordt weergeven in figuur 2.7.

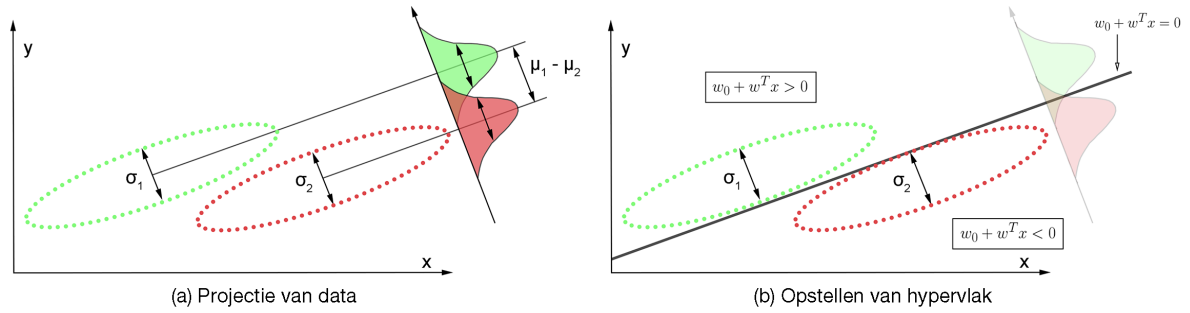


Figuur 2.7: Kruisvalidatie- en trainfouten voor modelcomplexiteit in functie van de voorspelde fouten. De gebruikte modellen worden gefit op een voorbeelddatabase betreffende geluk in functie van rijkdom. Bij een hoge bias (a) is er zowel een hoge kruisvalidatie- als trainfout. Het model is niet goed gefit op de traindata en werkt globaal ook niet. Dit wordt underfitting genoemd. Bij 'Balans' (b) is de trainfout relatief laag en is de kuisvalidatiefout minimaal. Op dit punt is de bias-variantie afweging optimaal. Bij een hoge variantie (c) is de trainfout heel klein, wat aanduidt dat het model goed gefit is op de trainset. De kruisvalidatiefout is echter hoog, waardoor dit model niet aanvaard kan worden als globaal model. Hier wordt gesproken van overfitting. [57]

2.4.4 Linear Discriminant Analysis

Wanneer de EEG-signalen verwerkt zijn, moeten er patronen in de data herkend worden. Om patronen te herkennen die ervoor zullen zorgen dat bepaalde acties uitgevoerd worden, wordt er gebruik gemaakt van het classificatie algoritme genaamd Linear Discriminant Analysis (LDA of ook gekend als Fisher's LDA) [55]. Dit algoritme werkt in twee stappen:

- In de eerste stap, geïllustreerd in (a) op figuur 2.8, zal LDA de data projecteren op een nieuwe as die zodanig gekozen is dat de afstand tussen de gemiddelden van de distributies minimaal wordt. Op deze manier zullen beide geprojecteerde distributies een maximale afscheiding van elkaar bekomen.
- In de tweede stap zal LDA een hypervlak maken die loodrecht staat op de projectie-as en loopt tussen beide geprojecteerde distributies. Dit wordt aangetoond in (b) op figuur 2.8. Vervolgens zal dit hypervlak gebruikt worden om de verschillende distributies of klassen van elkaar te onderscheiden. Indien gewerkt wordt met een twee-klasse probleem zoals bij ingebeelde beweging, wordt een klasse bepaald door de ligging van de data ten opzichte van het hypervlak.



Figuur 2.8: (a) Projectie van twee guassiaans verdeelde distributies op een nieuwe as, zodanig dat de afstand tussen de gemiddelden van deze klassen maximaal is met minimale variantie binnenin die geprojecteerde klassen [58]. (b) Opstellen van een hypervlak $w_0 + w^T x = 0$ loodrecht op de projectie-as om een onderscheid te maken tussen beide klassen

Een veelgebruikt alternatief voor LDA in het onderzoek naar BCI is Support Vector Machines (SVM) [55]. In deze thesis werd LDA echter als classifier gekozen, daar het voldoet aan de eisen van lage complexiteit en hoge nauwkeurigheid en het reeds een veelgebruikte classifier is bij BCI gebaseerd op ingebeelde beweging [55]. De computationele vereiste van LDA ligt ook heel laag, waardoor het perfect is om te gebruiken bij ware-tijdsapplicaties.

2.4.5 Regularisatie

Bij veel classifiers is het mogelijk om het model gefit op de traindata wat te regulariseren. Dit wordt gedaan door toevoeging van een extra parameter λ bij het model. Deze parameter wordt zo bepaald, dat overfitting van het model tegenwerkt zal worden.
Specifiek bij LDA wordt regularisatie toegepast door middel van een shrinkage parameter. Deze wordt vooral gebruikt om de schatting van LDA te verbeteren, wanneer het aantal train samples klein is ten opzichte van het aantal kenmerken per sample. Shrinkage zal er namelijk voor zorgen dat LDA beter in staat zal zijn om een gewenste inschatting te maken inzake tot welke klasse toebehoort aan een bepaalde sample. In de meeste gevallen wordt de shrinkage parameter automatisch bepaald door middel van het Ledoit en Wolf algoritme [59].

2.5 Classificatie van ingebeelde beweging

In deze sectie worden de voorgaande secties samengevat tot het opzetten van een werkende niet-invasieve BCI voor gebruik met ingebeelde beweging. In de eerste fase wordt deze BCI geconfigureerd om samen te werken met de gebruiker. Hiervoor moet de gebruiker een kalibratie van 20 á 30 minuten uitvoeren op het systeem, waarna het systeem vervolgens getrained kan worden op de verzamelde data. Een kalibratie verloopt als volgt:

1. De gebruiker wordt klaargemaakt om een goede opname te hebben van de hersengolven. Hiervoor wordt er conductieve gel aangebracht op de hoofdhuid van de gebruiker. Vervolgens wordt ook de EEG-kap aangebracht (zie sectie 1.1.2).

2. Op een computersysteem wordt de opname van EEG-signalen gestart. De gebruiker krijgt vervolgens pijlen te zien die ofwel naar links of naar rechts wijzen. Wanneer een pijl tevoorschijn komt, moet de gebruiker zich een beweging inbeelden in diezelfde richting als aangegeven door de pijl (zie sectie 1.4.1).

3. De verzamelde EEG-data wordt hierna voorwerwerkt door middel van filtering in het frequentiedomein en vervolgens door het gebruik van een spatiale filter. Een veelgebruikte en efficiënte filter is CSP (zie sectie 2.3).

4. Door de data voor te verwerken, is het aantal kenmerken aanzienlijk gedaald en kan de data gebruikt worden om een classifier te trainen. Hierbij wordt een supervised algoritme gebruikt, genaamd LDA (zie sectie 2.4.4).

Nu het systeem volledig gekalibreerd is, kan het door diezelfde gebruiker in ware tijd gebruikt worden. Het systeem is nu in staat om te achterhalen of de gebruiker in kwestie aan het denken is aan een linker of rechter beweging. Het zou echter wenselijk zijn indien de kalibratiesessie minder tijd in beslag zou nemen. Deze vergt namelijk veel tijd en concentratie van de gebruiker. Het doel is dus om een BCI te bouwen die een hoge accuraatheid behaald, maar weinig kalibratiedata van de gebruiker vereist. Hoe de accuraatheid van een BCI bepaald wordt, zal in de volgende sectie benaderd worden.

2.5.1 Evaluatiemethode

De evaluatie van een BCI gebeurt aan de hand van de accuraatheid. Deze is een maat voor de verhouding tussen het aantal juist geclassificeerde trials en het totaal aantal geclassificeerde trials:

$$Accuraatheid = \frac{\text{Correct geclassificeerd}}{\text{Totaal geclassificeerd}} \tag{2.3}$$

De accuraatheid vormt een goede maat voor evaluatie indien er gewerkt wordt met een gebalanceerde dataset (evenveel linker als rechter trials), wat het geval is bij de gebruikte datasets in deze thesis. Een nadeel van de accuraatheid als evaluatiemethode is dat het niet in staat is om de verschillende sterkten en zwakheden van een systeem naar voor te schuiven [60]. Deze worden echter niet behandeld binnenin deze thesis.

Naast evaluatie op basis van accuraatheid wordt naar aanleiding van de probleemstelling ook gekeken hoe snel bepaalde BCI goed kunnen beginnen presteren. Hierbij wordt verwezen naar transfer learning, wat een alternatieve methode is voor het kalibreren van een BCI. Wat dit nu juist is en hoe het te werk gaat, wordt in de volgende sectie verduidelijkt.

2.6 Transfer learning

Transfer learning is een methode die zich focust op het gebruiken van opgenomen data van eerdere gebruikers, om deze vervolgens op één of andere manier toe te passen op een nieuwe gebruiker. Zoals reeds vermeld in sectie 1.4.2 wordt in deze thesis gebruik gemaakt van subject-to-subject transfer learning. In de volgende subsectie worden enkele transfer learning algoritmen vanuit de literatuur verduidelijkt.

2.6.1 Algoritmen

Naïve transfer learning

Een eerste heel eenvoudig algoritme is naïve transfer learning. Hierbij wordt per eerdere gebruiker een BCI opgebouwd om vervolgens de data van de nieuwe gebruiker door deze BCI te sturen. Uiteindelijk worden de resultaten van al die BCI uitgemiddeld om een beslissing te vormen over de uitkomst van de data.

LFOS

Learning from other subjects (LFOS) is een subject-to-subject transfer learning algoritme dat desondanks de grote inter-subject variabiliteiten het mogelijk acht om gemeenschappelijke informatie te vinden in de EEG-signalen tussen eerdere gebruikers en de nieuwe gebruiker [61]. Zowel CSP als LDA hebben veel data nodig om goed te kunnen functioneren, aangezien beide algoritmen gebaseerd zijn op het schatten van de klasse covariantiematrices. Om nu net beiden te trainen aan de hand van weinig kalibratiedata, wat het doel is van transfer learning, wordt voorgesteld om informatie van eerdere gebruikers te integreren in die covariantiematrices. Bij CSP wordt dit gedaan door middel van een regularisatie parameter λ:

$$\bar{C}_t = (1-\lambda)C_t + \lambda \frac{1}{|S_t(\Omega)|} \sum_{i \in S_t(\Omega)} C_i \tag{2.4}$$

Hierbij stelt C_t de covariantiematrix voor, bepaald vanuit de kalibratiedata, $S_t(\Omega)$ de deelverzameling van eerdere gebruikers, $|S_t(\Omega)|$ het aantal eerdere gebruikers aanwezig in die deelverzameling en C_i de covariantiematrix van de overeenkomstige eerdere gebruiker i. De gemiddelde kenmerken vectoren μ van LDA worden vervolgens op een gelijkaardige manier geregulariseerd:

$$\bar{\mu_t} = (1-\lambda)\mu_t + \lambda \frac{1}{|S_t(\Omega)|} \sum_{i \in S_t(\Omega)} \mu_i \tag{2.5}$$

Op deze manier wordt een betere en robuustere schatting gemaakt van de covariantiematrices van beide methoden, wat moet leiden tot een betere classificatie performantie. Dit zou vooral moeten uitblijken wanneer heel weinig kalibratiedata van de nieuwe gebruiker beschikbaar is.

Een eerste vraag die hierbij opduikt, is hoe nu een goeie deelverzameling van eerdere gebruikers geselecteerd kan worden. Dit wordt gedaan aan de hand van een variant op het Sequential Forward Floating Search algoritme [62], wat gebruikt wordt om een relevante deelverzameling

van kenmerken te selecteren. Bij deze variant voegt het algoritme sequentieel eerdere gebruikers toe of verwijdert deze een eerdere gebruiker uit de deelverzameling. Dit wordt zodanig gedaan, dat de validatie accuraatheid (zie sectie 2.6.2) bekomen door de kalibratiedata van de nieuwe gebruiker te laten testen op een BCI getrained op die deelverzameling zo hoog mogelijk is. Bij deze BCI wordt zowel CSP als LDA getrained op een combinatie van de data over alle eerdere gebruikers in de bekomen deelverzameling. Door deze variant te gebruiken, wordt convergentie van het algoritme verzekerd en wordt een goede deelverzameling van eerdere gebruikers bekomen.

Een tweede vraag die hier geadresseerd moet worden, is hoe de regularisatie parameter efficiënt gekozen kan worden. Deze zou gekozen kunnen worden op basis van kruisvalidatie, maar is niet aangeraden doordat dit een heel tijdrovende methode is. De dataset waarop getrained wordt is ook relatief klein, waardoor het zonde zou zijn om een deel ervan te verliezen aan kruisvalidatie. Er wordt voorgesteld om te werken met een computationeel efficiënte manier om deze regularisatie parameter te zoeken:

1. Bereken de classificatie accuraatheid *targetAcc* door een BCI te trainen op de kalibratiedata door middel van Leave-One-Out-Validation (LOOV). Hoe LOOV juist in zijn werk gaat, wordt verder uitgelegd in sectie 2.6.2.

2. Bereken de test accuraatheid *selectedAcc* door een BCI te trainen op de deelverzameling van eerdere gebruikers en deze te testen op de kalibratiedata van de nieuwe gebruiker.

3. Bereken de accuraatheid *randAcc* als dat de classifier een performantie zou hebben van een gokker (vb. 50% voor een twee-klasse probleem).

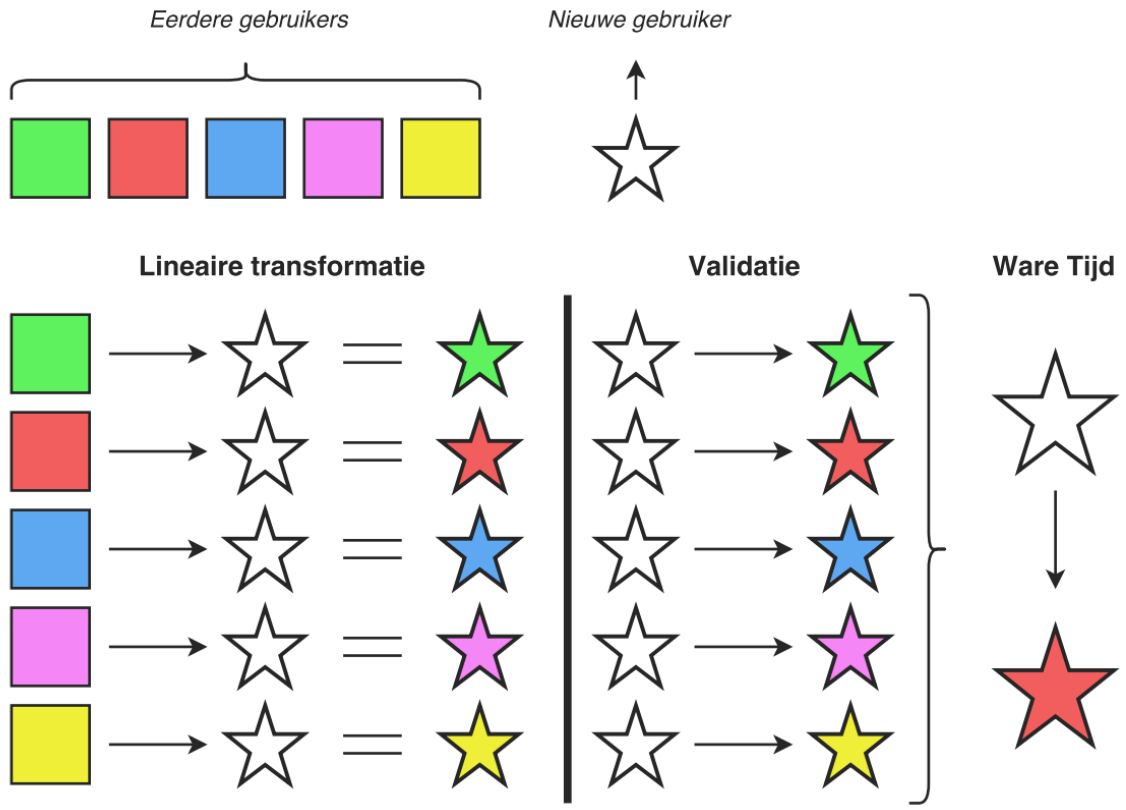
4. Bepaal de waarde van de regularisatie parameter als volgt:

$$\lambda = \begin{cases} 1 & targetAcc \leq randAcc \\ 0 & targetAcc \geq selectedAcc \\ \frac{selectedAcc - targetAcc}{100 - randAcc} & anders \end{cases}$$

Door deze manier van werken te hanteren, vindt er een afweging plaats tussen de invloed van de kalibratiedata van de nieuwe gebruiker ten opzichte van de data van eerdere gebruikers.

EEG-DSA

Een ander subject-to-subject transfer learning algoritme uit de literatuur is EEG data space adaption (EEG-DSA). Hierbij wordt de data van een nieuwe gebruiker getransformeerd in de richting van de data van eerdere gebruikers, waardoor het verschil tussen beide distributies geminimaliseerd wordt.



Figuur 2.9: Figuurlijke voorstelling van het EEG-DSA algoritme

Het voorgestelde algoritme doorloopt twee stappen (zie figuur 2.9). In de eerste stap wordt gekeken om de data van de nieuwe gebruiker te adapteren naar elke eerdere gebruiker individueel, door middel van een lineaire transformatie met transformatie matrix V_k. In deze thesis wordt de data van eerdere gebruikers echter aangepast naar de data van de nieuwe gebruiker. Hierbij blijft V_k dezelfde.

De transformatie matrix wordt berekend aan de hand van de klasse covariantiematrices van beide distributies:

$$V_k = \sqrt{2}((\bar{\Sigma}_{k,1}^{-1}\Sigma_1 + \bar{\Sigma}_{k,2}^{-1}\Sigma_2)^{\dagger})^{0.5} \tag{2.6}$$

Hierbij is † de pseudo inverse van de matrix, $\bar{\Sigma}_{k,i}$ de i-de klasse covariantiematrix van eerdere gebruiker k, Σ_i de i-de klasse covariantiematrix bepaald door de kalibratiedata van de nieuwe gebruiker en V_k de optimale lineaire transformatie matrix dat de EEG-data van de nieuwe gebruiker transformeert, zodanig dat het verschil tussen die nieuwe data en de data van de k-de eerdere gebruiker minimaal is.

Wanneer de transformatie matrices van alle eerdere gebruikers bepaald zijn, wordt het beste kalibratiemodel bepaald. Deze wordt beslist door de hoogste accuraatheid behaald op één van de opgestelde BCI. Deze BCI worden getrained op één van de k eerdere gebruikers en getest op de getransformeerde kalibratiedata h_k van de nieuwe gebruiker (zie formule 2.7).

$$h_k = V_k^T X \tag{2.7}$$

Door nu net die met de hoogste accuraatheid te selecteren, wordt een overeenkomstige gebruiker gevonden waar er meer data van beschikbaar is. Hierdoor kan er een sterke BCI gebouwd worden ondanks de kleine hoeveelheid aan kalibratiedata van de nieuwe gebruiker.

2.6.2 Discrimineerbaarheid van kenmerken

Inzake transfer learning is het heel belangrijk op te merken dat de ene gebruiker niet altijd overeenkomstig is met de andere gebruiker door inter-subject variabiliteiten. Het zou echter mogelijk moeten zijn om gebruikers te vinden die in een bepaalde mate overeenkomstige kenmerken hebben. Door gebruik van deze kenmerken, kunnen vervolgens compatibele personen aangeduid worden voor gebruik in subject-to-subject transfer learning. Hieronder worden enkele methodologieën uit de literatuur aangehaald die de gelijkenis tussen verschillende distributies aantonen.

Validatie accuraatheid

De validatie accuraatheid wordt omschreven als de test accuraatheid van de kalibratie data, gestuurd door een BCI die getrained is op één of meerdere eerdere proefpersonen. Dit is een enorm goede maatstaf om uit te maken welke van die eerdere gebruikers overeenkomstig zijn met de nieuwe gebruiker. Dit blijkt ook uit het gebruik ervan in zowel LFOS als EEG-DSA.

Leave-One-Out-Validation

Leave-One-Out-Validation (LOOV) is een speciale vorm van kruisvalidatie. Hierbij wordt de data ingedeeld in k-folds, met k gelijk aan het aantal trials. Zo wordt er k-keer een BCI opgesteld, die vervolgens getrained wordt op $k-1$ trials en getest wordt op de ene overige trial. Vervolgens wordt de LOOV bekomen door de resultaten van alle BCI uit te middelen. Hierdoor wordt een idee verkregen van hoe goed de BCI werkt met de huidige data, net zoals bij kruisvalidatie. LOOV wordt echter meer aangeraden dan kruisvalidatie bij gebruik met kleinere datasets, aangezien een grotere k ervoor zou zorgen dat er te weinig data overblijft om een BCI op te trainen, waardoor de resultaten onbetrouwbaar zouden zijn.

Consistentie index

Een alternatief voorstel om een verband te zoeken tussen twee datasets is het opstellen van een consistentie index [63]. Deze index geeft numeriek weer hoe goed twee datasets met elkaar verwant zijn. Hierbij wordt rekening gehouden met volgende eigenschappen:

1. **Monotoniciteit.** Voor een vaste grootte k van een deelverzameling en een vast aantal kenmerken n wordt de consistentie index groter naarmate de kruising tussen twee deelverzamelingen groter wordt. Deze kruising r wordt gedefinieerd als $r = |A \cap B|$.

2. **Gelimiteerd.** De index moet onafhankelijk van n of k gelimiteerd worden door constanten. De maximale waarde moet behaald worden wanneer twee deelverzamelingen identiek zijn ($r = k$).

3. **Correctie voor kans.** De index moet een constante waarde behouden voor onafhankelijk getekende deelverzamelingen van kenmerken met éénzelfde grootte k. Een algemene vorm van zo een index wordt weergeven als volgt:

$$\frac{\text{Geobserveerde } r - \text{Verwachte } r}{\text{Maximale } r - \text{Verwachte } r} \tag{2.8}$$

Om een formule op te bouwen om de consistentie te weergeven tussen twee datasets, wordt gestart vanuit formule 2.8. Hierbij wordt de geobserveerde r vervangen door $r = |A \cap B|$, de verwachte r door $\frac{k^2}{n}$ en de maximale r door k. Zo wordt de volgende formule voor de consistentie index I_C bekomen:

$$I_C(A,B) = \sum_{k=1}^{n-1} \frac{r - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \sum_{k=1}^{n-1} \frac{rn - k^2}{k(n-k))} \tag{2.9}$$

Deze formule voldoet aan alle drie bovenstaande voorwaarden. Zo wordt $I_C(A, B)$ groter, wanneer k groter wordt, wordt de maximale waarde van de index, $I_C(A, B) = 1$, bereikt wanneer $r = k$ en zal de index richting nul gaan voor onafhankelijke deelverzamelingen A en B, doordat de verwachte waarde van r geschat wordt op $\frac{k^2}{n}$.
Om nu over een volledige dataset de consistentie weer te geven, worden alle consistentie indexen over het aantal vergeleken trials uitgemiddeld. Deze zal de consistentie I_C als één enkele waarde tussen 1 en -1 weergeven.

In de context van transfer learning wordt de consistentie index in deze thesis gebruikt om te kijken of een eerdere gebruiker enige overeenkomstigheid vertoont met een nieuwe gebruiker. Hierbij wordt voor elke set van voorgefilterde kenmerken, per trial een variatie op de consistentie index $I_C(A, B)$ bepaald (zie 3.2.1), met A en B trials van respectievelijke gebruikers. Vervolgens wordt via uitmiddeling over alle trials een volledige schatting gemaakt van de overeenkomstigheid tussen beide gebruikers. Hoe hoger deze waarde, hoe meer overeenkomstig beide gebruikers zouden moeten zijn.

Maximum Mean Discrepancy

Een laatste methode dat gebruikt wordt vanuit de literatuur is de Maximum Mean Discrepancy (MMD) [64]. Dit is een methode dat de afstand berekent tussen twee distributies. Een andere veelgebruikte methode om de afstand tussen twee distributies te berekenen is Kullback-Leibler [65]. Het nadeel hiermee is echter dat de berekening voor de dichtheid van de distributies computationeel belastend is. MMD scoort op dit vlak beter. Het gaat uit van het principe dat, indien de gemiddelde kenmerken van de populaties identiek zijn in een reproduceerbare kernel Hilbert ruimte (RKHS), de distributies gelijk zijn aan elkaar. Hierop gebaseerd zal MMD de afstand berekenen tussen twee distributies, door het verschil tussen de empirische gemiddelden van de samples in een RKHS te nemen.

Hoofdstuk 3

Ranking-Based Subject And Methodology Selection (RSAMS)

3.1 Inleiding

In dit hoofdstuk wordt gezocht naar een subject-to-subject transfer learning algoritme dat kan omgaan met een grote hoeveelheid eerdere gebruikers. Enkele voorwaarden die worden opgelegd in het zoeken naar een algoritme zijn de volgende:

1. Het algoritme moet in staat zijn om uit een poele van eerdere gebruikers één of meerdere gebruikers te vinden die overeenkomstig zijn met een nieuwe gebruiker.

2. Er moet rekening gehouden worden met een afweging tussen verschillende methoden van werken. Stel dat er bijvoorbeeld genoeg kalibratiedata aanwezig is om een accurate BCI te bouwen zonder gebruik van transfer learning. Indien vastgesteld wordt dat deze methode beter scoort, kan deze manier van werken gevolgd worden.

3. Ondanks de grote daling in kalibratiedata mag er slechts een minimale daling zijn in accuraatheid van de BCI.

Met deze voorwaarden in het achterhoofd wordt een sequentieel proces doorlopen om tot slot het RSAMS algoritme op te stellen. De verschillende algoritmen bekomen tijdens deze ontwikkeling worden in de volgende secties uitvoerig besproken, met oog op de eerste twee voorwaarden. Resultaten bekomen door deze algoritmen komen in het volgende hoofdstuk aan bod, alsook de validatie van de derde voorwaarde.

3.2 Keuze van overeenkomstige gebruikers

Vooraleer een volwaardig subject-to-subject transfer learning algoritme ontwikkeld kan worden, moet er eerst gedacht worden aan een manier om een gebruiker te vinden die overeenkomstig is met een nieuwe gebruiker. Dit moet ook goed werken wanneer de poele van eerdere gebruikers groot is. De keuze van parameters alsook het sorteer systeem zelf worden uitvoerig besproken in de volgende secties.

3.2.1 Gekozen parameters

Als eerste werden vier parameters uitgekozen die een bepaalde gelijkenis aangeven tussen twee verschillende distributies. Twee van deze parameters zijn de MMD en de validatie accuraatheid. Deze worden gehaald uit de literatuur en werden reeds uitgelegd in sectie 2.6.2. De andere twee gebruikte parameters, consistentie index en Least Squared Euclidean Distance (LSED), zijn varianten op de literatuur en worden hieronder besproken.

Variant op consistentie index

De consistentie index zoals beschreven in de literatuur (zie sectie 2.6.2) is een maat voor het aantal overeenkomstigheden tussen twee distributies. Hierbij moeten de vergeleken waarden echter exact dezelfde zijn. Aangezien kenmerken bepaald door CSP geen gehele getallen zijn, moet factor r_k zodanig aangepast worden dat voor een groot verschil tussen kenmerken l_A en l_B deze parameter toegaat naar $r_k = 0$. Omgekeerd moet $r_k = 1$ bekomen worden indien beide kenmerken exact dezelfde blijken te zijn. Dit wordt verwezenlijkt aan de hand van de volgende formule:

$$r_k(l_A, l_B) = \sum_{i=0}^{k-1} \frac{1}{1 + |l_A(i) - l_B(i)|} \tag{3.1}$$

Hierbij wordt voor een bepaalde trial l een vergelijking gemaakt tussen distributies A en B bij een constant aantal kenmerken k. Om nu de consistentie index op te stellen over de volledige datasets, wordt de formule gebruikt uit de literatuur en wordt het gemiddelde genomen over alle kenmerken n en trials L heen:

$$I_c(A, B) = \frac{1}{n * L} \sum_{l=0}^{L} \sum_{k=1}^{n-1} \frac{r_k(l_A, l_B) * n - k^2}{k(n-k)} \tag{3.2}$$

LSED

Least Squared Euclidean Distance (LSED) is een parameter die gebruik maakt van de euclidische afstand tussen twee punten. Deze punten worden bepaald als de som over de verschillende klassen van de kwadraten, genomen van de euclidische afstand tussen kenmerken van verschillende distributies:

$$LSED(A,B) = \sum_{c=1}^{C} \|\mu_A(c) - \mu_B(c)\|^2 \tag{3.3}$$

Hierbij is C het aantal klassen die onderscheid worden (ingebeelde beweging is een twee-klasse probleem) en $\mu_A(c)$ en $\mu_B(c)$ de gemiddelde kenmerkvectoren van distributies A en B voor een klasse c. Door het gebruik van deze parameter kan aangetoond worden hoe groot het verschil is tussen de gemiddelde kenmerken van beide distributies, waardoor het als maat dient om een gelijkenis tussen beide aan te tonen.

3.2.2 Sorteer systeem

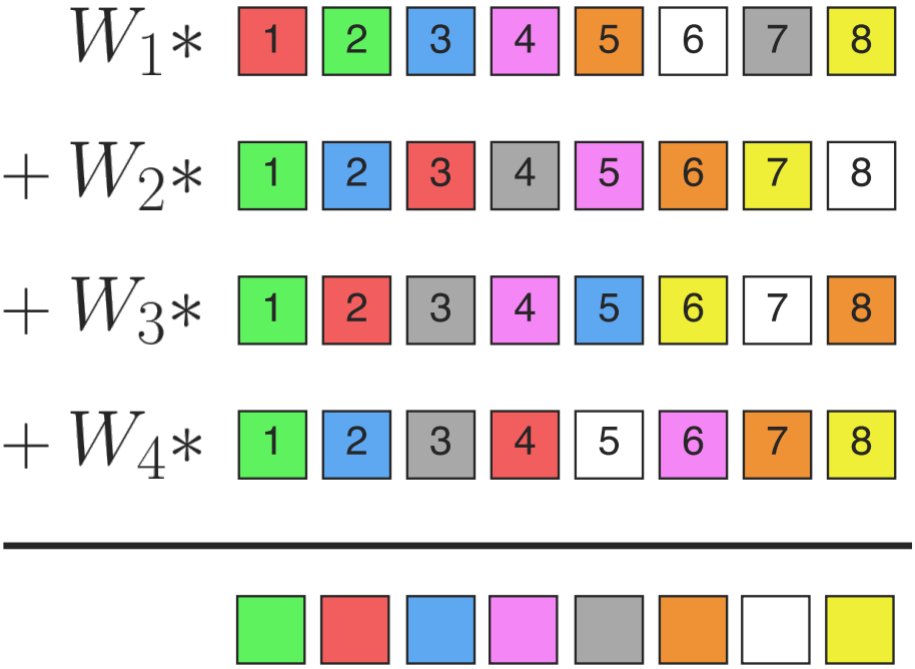
Één van de belangrijke vooropgestelde voorwaarden is dat het algoritme in staat moet zijn om een overeenkomstige gebruiker te zoeken in een poele van eerdere gebruikers. Dit wordt gedaan aan de hand van het sorteer systeem. Hierbij zullen de kalibratiedata van de nieuwe gebruiker en de traindata van de eerdere gebruikers vergeleken worden op vier verschillende manieren:

- **Validatie accuraatheid**, bepaald door de kalibratiedata door een BCI te sturen die getrained is op een eerdere gebruiker.
- **Consistentie index**, die een maat biedt voor de gelijkenis tussen kenmerken.
- **Mean Maximum Discrepancy**, om een afstand te kunnen opmeten tussen kenmerken.
- **Least Squared Euclidean Distance**, die aanduidt hoe groot de afstand is tussen de gemiddelden van distributies.

Na het bepalen van deze parameters vindt er eerst een voorfiltering plaats op basis van de validatie accuraatheid. Hierbij wordt gezocht naar de maximaal behaalde validatie accuraatheid over alle BCI, om vervolgens diegene met een validatie accuraatheid minder dan 15% onder dat maximum uit het sorteer systeem te verwijderen.

Met de overgebleven BCI wordt vervolgens een ranking opgesteld per parameter. Hierbij krijgt

de BCI per parameter een rangwaarde toegewezen. Dit wordt voorgesteld in figuur 3.1. Vervolgens worden gewichten toegekend per parameter. Op deze manier kan de ene parameter meer doorslag bieden dan de ander. In het kader van deze thesis worden de gewichten empirisch bepaald. Dat wil zeggen dat resultaten bekomen met verschillende gewichten bestudeerd worden, om zo een idee te krijgen van de gewenste gewichten. Een alternatief hierop zou zijn om een systeem op te stellen dat optimale gewichten kan bepalen aan de hand van voorbeelddata. Deze methode wordt echter niet toegepast binnenin deze thesis, maar kan gezien worden als verdere uitbreiding. De gewichten worden hierna per parameter vermenigvuldigd met de rangorden toegekend aan de BCI. Tot slot wordt een sommatie genomen van die rangorden over alle parameters om zo een finale rangorde op te stellen.



Figuur 3.1: Het sorteer systeem schematisch voorgesteld met 8 opgestelde BCI. Elke rij stelt de volgorde van de BCI voor, gesorteerd op een bepaalde parameter. Vervolgens wordt de som genomen van de volgorden van elke rij, vermenigvuldigd met de respectievelijke gewichten. Dit geeft een finale score weer, waarbij de BCI met de laagste score het meest overeenkomstig blijkt.

Uit deze finale volgorde wordt besloten dat de BCI met de laagste rangorde, het meest overeenkomstig blijkt te zijn met de nieuwe gebruiker. Doordat dit sorteer systeem toelaat om zelfs in een grote poele van eerdere gebruikers een overeenkomstige gebruiker te vinden en het rekening houdt met verschillende parameters, wordt het gebruikt bij het opstellen van het RSAMS

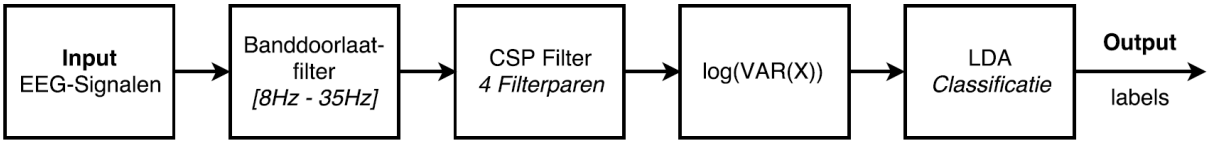
algoritme. Hoe dit algoritme tot stand gekomen is, wordt verduidelijkt in de volgende sectie.

3.3 Verschillende methodologieën

Doorheen het verloop van de thesis werden meerdere algoritmen uitgeprobeerd. Deze werden opgesteld in de vorm van een sequentieel proces, waarbij het opeenvolgende algoritme verder bouwt op het voorgaande. Stapsgewijs wordt het algoritme dus steeds verder geoptimaliseerd, om uiteindelijk te kunnen voldoen aan alle vooropgestelde voorwaarden.

3.3.1 Zonder transfer learning

Om een BCI te kunnen bouwen gebaseerd op transfer learning, moet eerst een BCI opgesteld worden die een goede accuraatheid bekomt door gebruik van een grote hoeveelheid kalibratiedata. Op deze manier wordt een vaste basis opgebouwd om verdere BCI op te baseren. De setup van deze werking wordt schematisch voorgesteld in figuur 3.2.



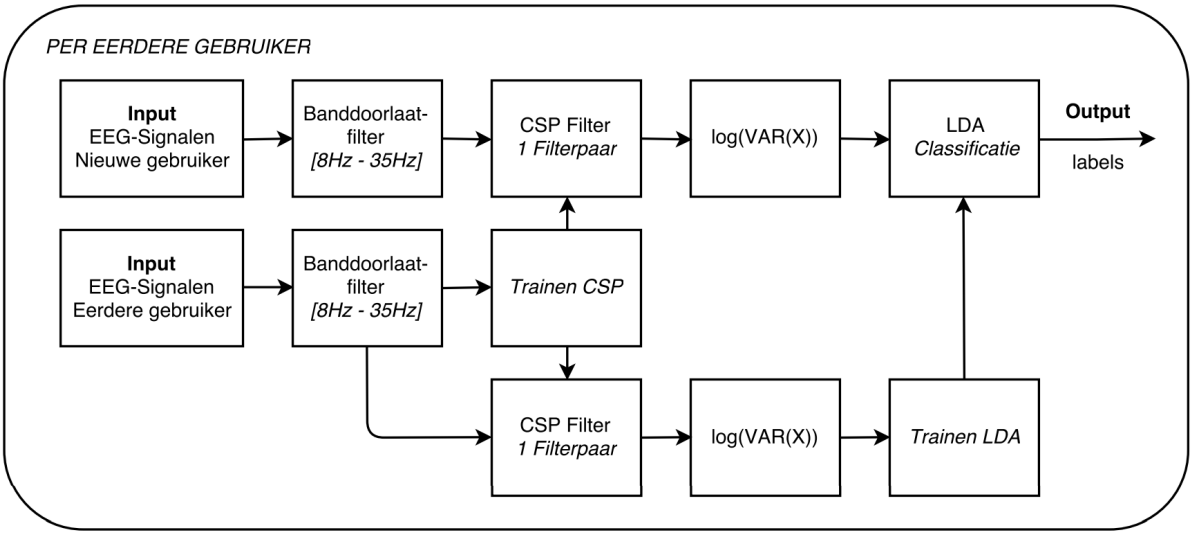
Figuur 3.2: Schematische voorstelling van een BCI gebaseerd op ingebeelde beweging

Bij de werking in ware tijd vindt er eerst een filtering plaats van de EEG-signalen in het frequentiedomein. Dit wordt bekomen door gebruik van een banddoorlaatfilter, die alle frequenties tussen de 8Hz en 35Hz zal doorlaten. Vervolgens zal de data spatiaal gefilterd worden door de 4 uiterste filterparen van een CSP filter. Deze is hierbij reeds getrained op een grote hoeveelheid kalibratiedata. Hierna wordt de variantie van de data berekend en wordt het logaritme ervan genomen om ervoor te zorgen dat de data meer normaal verdeeld is. Tot slot zal een LDA classifier bepalen of het om een linker of rechter trial gaat. Deze werd ook op voorhand getrained aan de hand van de kalibratiedata. Deze voorgaande stappen en meer info omtrent kalibratie werd reeds gegeven in sectie 2.5.

3.3.2 Eigen algoritme 1

In deze eerste eenvoudige vorm van het subject-to-subject transfer learning algoritme, worden er verschillende BCI gebouwd aan de hand van eerdere data. Deze BCI bouwen verder op

degene uitgelegd in de vorige sectie. Bij elke BCI zal zowel CSP als de LDA getrained worden op de volledige dataset van een eerdere gebruiker (zie figuur 3.3). Hierdoor wordt per eerdere gebruiker een stabiele BCI opgebouwd. Vervolgens wordt de validatie accuraatheid bepaald per opgemaakte BCI door de kalibratiedata van de nieuwe gebruiker door alle BCI individueel te sturen. Na het berekenen van de parameters die de correlatie tussen twee distributies weergeven (zie sectie 3.2.1), wordt via het sorteer systeem de meest overeenkomstige eerdere gebruiker geselecteerd. Als laatste zal de BCI opgezet met de meest overeenkomstige eerdere gebruiker in gebruik genomen worden om verdere EEG-data van de nieuwe gebruiker te verwerken en classificeren.



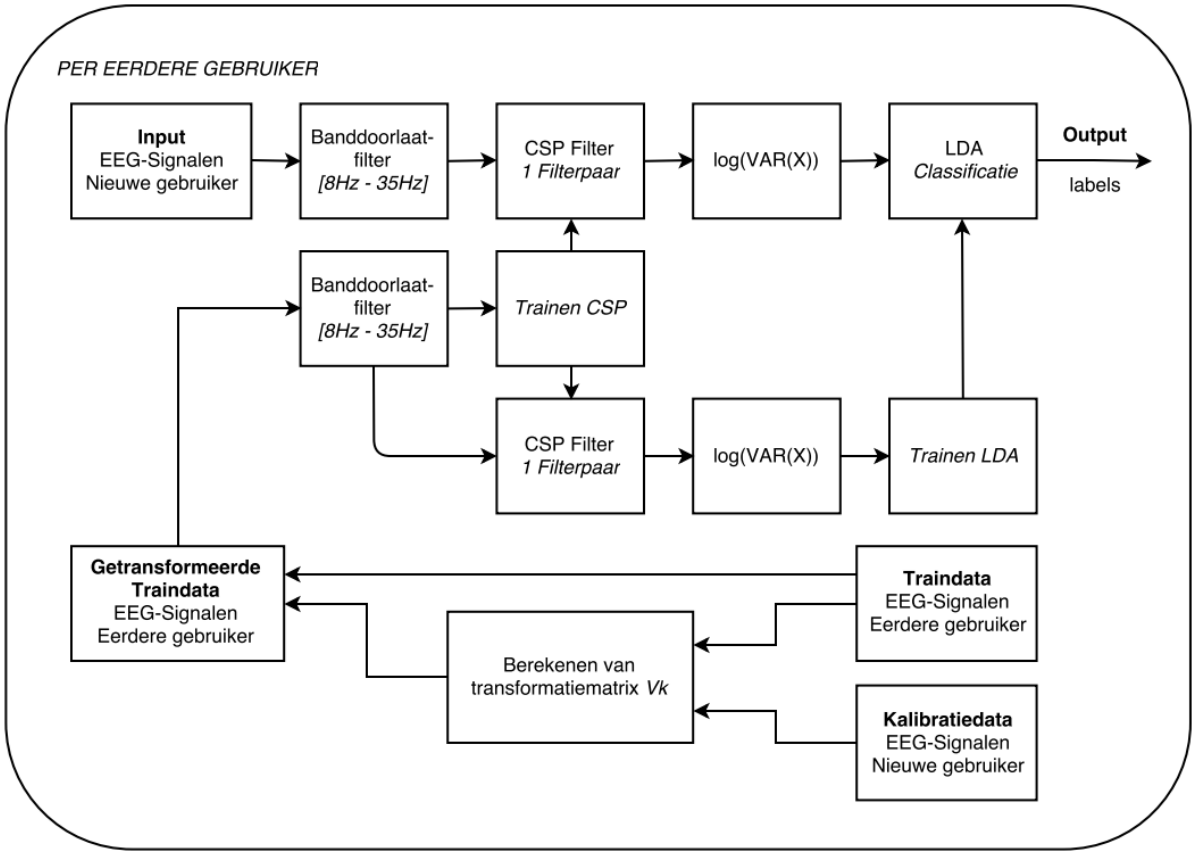
Figuur 3.3: Schematische voorstelling van de eerste uitvoering van het subject-to-subject transfer learning algoritme

Wanneer er terug gekeken wordt naar de voorwaarden, wordt opgemerkt dat door het gebruik van het sorteer systeem dit algoritme reeds voldoet aan de eerste voorwaarde. Het is dan ook logisch dat het sorteer systeem meegenomen wordt in de volgende uitvoering van het algoritme.

3.3.3 Eigen algoritme 2

Bij de tweede uitvoering van het algoritme wordt ervoor gezorgd dat de data van eerdere gebruikers meer overeenkomstig gemaakt wordt met de data van de nieuwe gebruiker. Dit wordt gedaan door een combinatie te maken van de voorgaande uitvoering en EEG-DSA (zie sectie 2.6.1). Hierbij zal de data van eerdere gebruikers eerst getransformeerd worden naar de data van de nieuwe gebruiker toe door middel van een lineaire transformatie. Vervolgens zal per eer-

dere gebruiker een BCI opgesteld worden, getrained op de getransformeerde data van de eerdere gebruiker (zie figuur 3.4). Per BCI zullen dan opnieuw enkele parameters bepaald worden, in zake gelijkenis tussen de getransformeerde data van de eerdere gebruiker en de kalibratiedata van de nieuwe gebruiker. Door opnieuw het sorteer systeem toe te passen, wordt het mogelijk om de meest overeenkomstige eerdere gebruiker te selecteren uit de poele van eerdere gebruikers. Tot slot wordt de BCI gelinkt aan die eerdere gebruiker verder gebruikt om nieuwe data te classificeren van de nieuwe gebruiker.



Figuur 3.4: Schematische voorstelling van de tweede uitvoering van het subject-to-subject transfer learning algoritme

Daar dit algoritme een alternatieve aanpak heeft op de voorgaande uitvoering, blijkt het qua voorwaarden ook enkel te voldoen aan de eerste. Het algoritme is net zoals de voorgaande uitvoering in staat om de meest overeenkomstige eerdere gebruiker naar voor te schuiven door gebruik van het sorteer systeem. Om aan de tweede voorwaarde te voldoen, moet er echter een afweging in methodologieën gemaakt worden. Deze afweging wordt geïmplementeerd in de volgende uitvoering.

3.3.4 Eigen algoritme (Beste van 2)

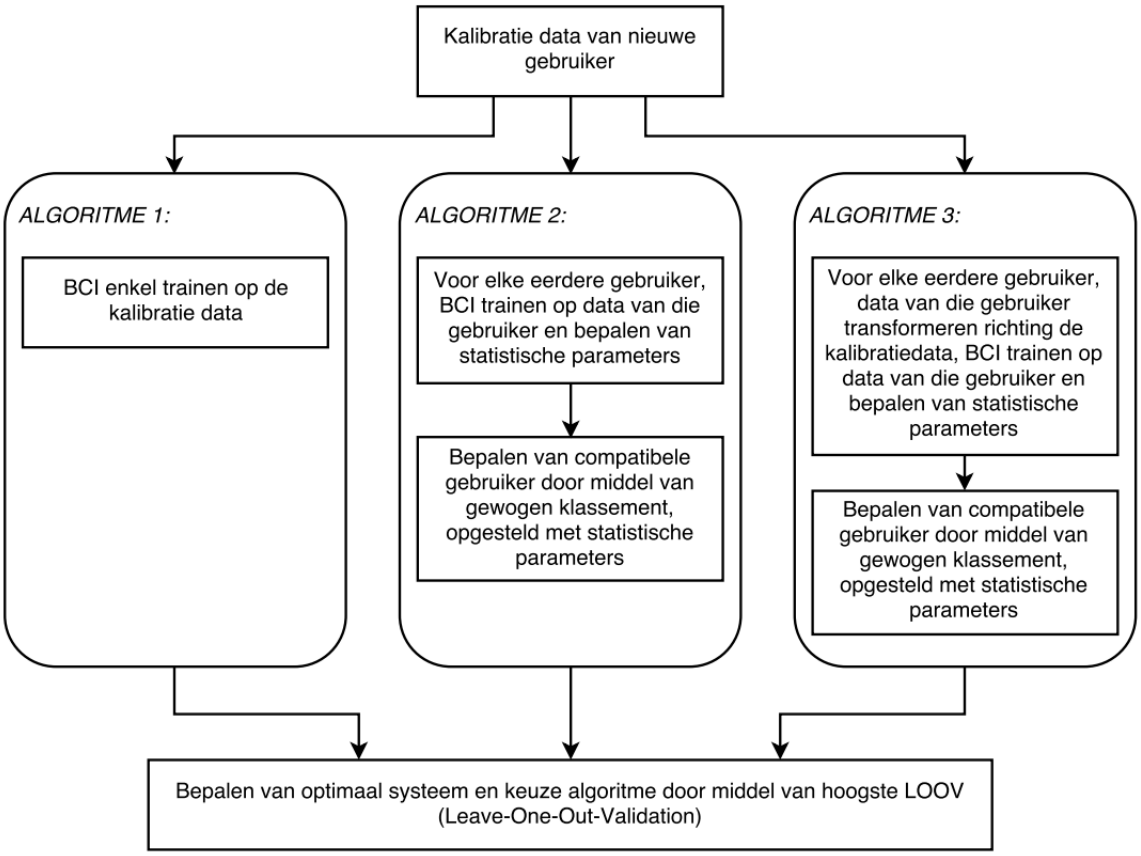
Beide voorgaande uitvoeringen zijn in staat om een overeenkomstige gebruiker te zoeken aan de hand van kalibratiedata van een nieuwe gebruiker. Ze verschillen echter in hoe de BCI opgesteld worden. Door dit verschil, wordt het bijvoorbeeld in eigen algoritme 2 mogelijk om een andere gebruiker als overeenkomstig te achten dan in eigen algoritme 1 door een lineaire transformatie er op toe te passen. Bijgevolg kan het ook gebeuren dat verschillende resultaten naar voren geschoven worden. Bij dit algoritme wordt nu net gezocht welk van die twee methoden het beste zou moeten werken bij een bepaalde nieuwe gebruiker.

Doordat beide algoritmen dezelfde parameters gebruiken in combinatie met het sorteer systeem, kunnen beide als het ware samen gesorteerd worden. Het idee hierbij is om de BCI van zowel de eerste als de tweede uitvoering samen in een sorteer systeem te stoppen. Vervolgens kan het systeem hieruit de optimale BCI kiezen voor een nieuwe gebruiker, met hieraan de gebruikte methode gelinkt om die BCI op te stellen. Hierdoor voldoet deze uitvoering aan zowel de eerste als de tweede voorwaarde.

3.4 Overzicht van RSAMS algoritme

Ranking-Based Subject And Methodology Selection of RSAMS in het kort, combineert het beste van voorgaande uitvoeringen. Het is namelijk in staat om een overeenkomstige gebruiker te selecteren (voorwaarde 1) door middel van het sorteer systeem én het is in staat om een afweging te maken tussen verschillende algoritmen naar keuze (voorwaarde 2).

Voor dit algoritme werd een afweging gemaakt in methodologieën tussen het algoritme zonder transfer learning, de eerste uitvoering van het algoritme (Eigen algoritme 1) en de tweede uitvoering (Eigen algoritme 2). Om te weten welk van deze methoden het best geschikt lijkt bij een bepaald individu, wordt de Leave-One-Out-Validation (zie sectie 2.6.2) berekend per methode. De methode met de hoogste LOOV wordt dan naar voor geschoven als gebruikte methode. Het volledige algoritme wordt schematisch beschreven in figuur 3.5.



Figuur 3.5: Een schematische benadering van het RSAMS algoritme

Hoofdstuk 4

Evaluatie

4.1 Inleiding

In dit hoofdstuk worden de besproken methoden vanuit voorgaande hoofdstukken toegepast om een realistische BCI gebaseerd op ingebeelde beweging op te bouwen. Hierbij wordt gebruik gemaakt van verschillende datasets om de BCI zo uitgebreid mogelijk te testen. Data uit die datasets wordt ingedeeld in kalibratiedata en ware-tijdsdata. Hierdoor wordt een realistisch scenario nagebootst, waarbij een gebruiker een kalibratie fase doorloopt vooraleer het systeem in ware tijd gebruikt kan worden. Doordat er echter gezocht wordt naar een optimale werking van de BCI, worden de resultaten bekomen door de ware-tijdsdata gebruikt om bepaalde keuzes te maken inzake parameters van de BCI. Hierdoor worden deze experimenten ook offline-experimenten genoemd.

De datasets waarmee de methoden geëvalueerd worden, werden reeds verduidelijkt in sectie 2.2.3. Hierbij worden de datasets van BCI Competitie IV gebruikt voor het uitbouwen van een optimale gekalibreerde BCI en voor het evalueren van de transfer learning algoritmen. De datasets aangeboden door Physionet dienen als poele van eerdere gebruikers in de te testen transfer learning algoritmen.

Het verloop van dit hoofdstuk kan ingedeeld worden in twee grote delen. Eerst zal er een BCI opgebouwd worden die goed scoort over verschillende datasets. Hierbij zullen enkele parameters zoals frequentiebanden, filterparen en tijdsindelingen geoptimaliseerd worden, zodat een optimaal systeem bekomen wordt. In het tweede deel zal deze BCI als basis gebruikt worden

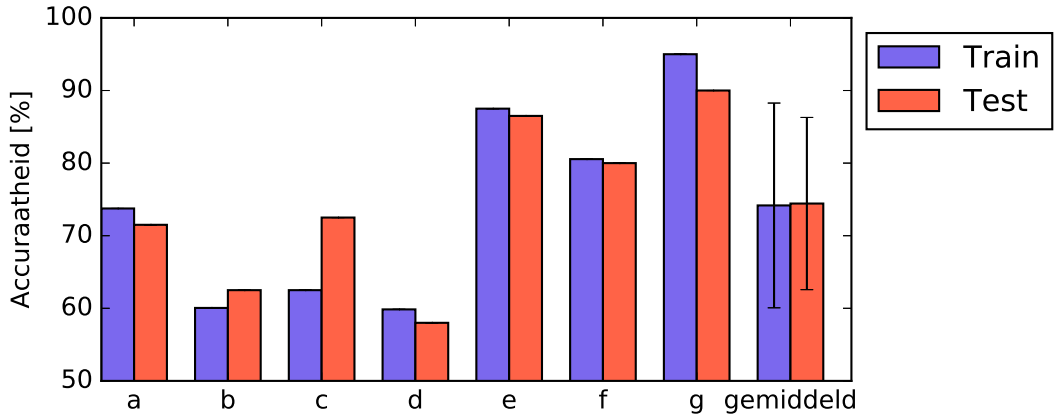
om verschillende transfer learning algoritmen vanuit de literatuur en zelf opgestelde algoritmen te evalueren.

4.2 Bouwen van BCI

In deze sectie wordt een BCI opgebouwd volgens de manier besproken in sectie 2.5. Hierbij wordt gebruik gemaakt van data van 7 verschillende proefpersonen die aanwezig is in de datasets van BCI Competition IV. Per gebruiker wordt 80% van de data gebruikt voor kalibratie en de overige 20% wordt gebruikt om de BCI op te testen. Bij dit testen wordt gebruik gemaakt van accuraatheid als evaluatiemethode (zie sectie 2.5.1).
Eerst en vooral wordt een vergelijking gemaakt tussen SL en CSP als spatiale filter. Vervolgens worden enkele parameters in verband met de gekozen spatiale filter en BCI in het algemeen geoptimaliseerd. Op deze manier wordt een optimaal functionerende BCI bekomen.

4.2.1 Naïeve manier (Small Laplacian)

Een eerst gebouwde vorm van de BCI maakt gebruik van de Small Laplacian (SL) (zie sectie 2.3.2) als spatiale filter. Hierbij wordt per persoon een geregulariseerde BCI opgesteld (zie sectie 2.4.5), waarbij de data eerst door een banddoorlaatfilter [3Hz - 40Hz] gestuurd wordt vooraleer het door de spatiale filter gaat. Het kalibratie gedeelte van de data (80%) wordt vervolgens gebruikt om de LDA classifier te trainen, terwijl de overige data (20%) gebruikt wordt om te testen. De bekomen accuraatheden worden beschreven in figuur 4.1.



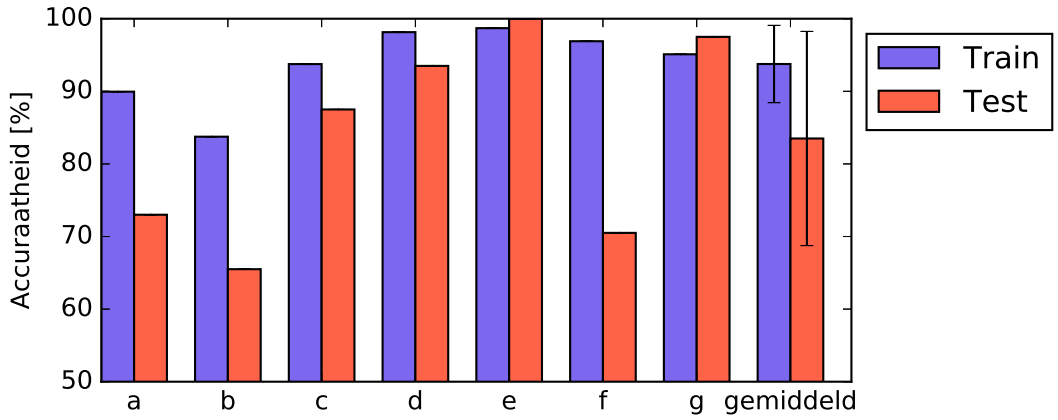
Figuur 4.1: BCI met Laplaciaan filtering (80-20)

Er wordt opgemerkt dat zowel de train als de test accuraatheden niet zo hoog liggen. Bij

sommige proefpersonen wordt slechts een test accuraatheid behaald rond de 60%, terwijl andere een test accuraatheid hoger dan 85% bekomen. Er kan dus gesteld worden dat de BCI met Small Laplacian als spatiale filter slechts goed werkt bij enkele proefpersonen.

4.2.2 CSP filtering

Als tweede vorm van BCI wordt de spatiale filter uit de voorgaande sectie vervangen door CSP (zie sectie 2.3.2). Deze moet echter training ondergaan voor het in ware tijd gebruikt kan worden. Dit wordt gedaan aan de hand van de kalibratiedata van de gebruiker, net zoals bij de training van de LDA classifier. Resultaten bekomen door deze BCI uitvoering worden in figuur 4.2 weergeven.

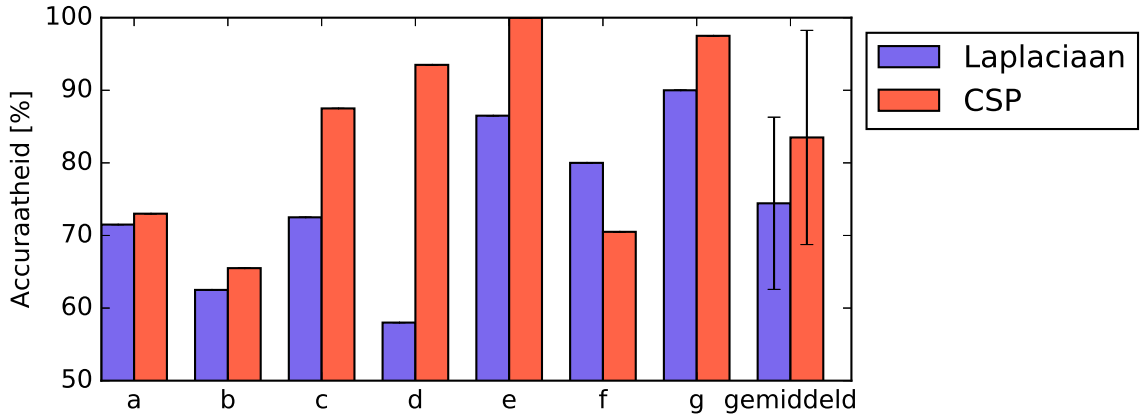


Figuur 4.2: BCI met CSP filtering (80-20)

Bij het bekijken van deze resultaten wordt opgemerkt dat de train accuraatheden algemeen gezien hoog liggen. Dit wil zeggen dat de BCI goed gefit is op de kalibratiedata van de gebruiker. Wanneer er vergeleken wordt met de resultaten bekomen bij gebruik van SL (zie figuur 4.1), wordt opgemerkt dat de train accuraatheden bij CSP gemiddeld gezien (93.77%) een heel stuk hoger liggen dan die bij SL (74.17%). Dit kan verklaard worden doordat het CSP algoritme training vereist vooraleer het in staat is om een goed onderscheid te maken tussen de twee klassen. Doordat training uitgevoerd wordt op diezelfde data, is het onderscheid in beide klassen beter zichtbaar, waardoor de train accuraatheid een stuk hoger ligt dan bij SL.

Wanneer er gekeken wordt naar de test accuraatheden wordt gezien dat deze gemiddeld ook goed scoren (83.50%). Individueel gezien zijn er toch weer enkele gebruikers die slechts een test accuraatheid rond de 65% bekomen.

Daar zowel een BCI met SL als CSP opgesteld werd, kunnen de test accuraatheden van beide zowel individueel als gemiddeld gezien vergeleken worden met elkaar (zie figuur 4.3).



Figuur 4.3: Vergelijking in test accuraatheid tussen BCI met Laplaciaan en CSP filtering met 3 filterparen (80-20)

Het eerste wat opvalt is dat op één gebruiker na, de BCI met CSP als spatiale filter altijd beter scoort dan die met SL. Ook gemiddeld gezien wordt opgemerkt dat deze BCI ongeveer 10% beter scoort qua test accuraatheid (74.43% ten opzichte van 83.50%), ondanks de standaarddeviatie bij beide uitwerkingen hoog is. Een reden hiervoor is dat SL in het totaal slechts gebruik maakt van 10 EEG-kanalen, terwijl CSP gebruik maakt van alle EEG-kanalen. Uit de resultaten vanop figuur 4.3 kan dus geconcludeerd worden dat bij deze proefpersonen CSP in het algemeen beter werkt als spatiale filter dan SL. Het is dan ook logisch dat deze als spatiale filter gebruikt wordt in verdere testen.

4.2.3 Optimalisatie van parameters

Een BCI heeft vele parameters die elk een eigen invloed hebben op de accuraatheid en snelheid ervan. Zo kan de frequentieband van de banddoorlaatfilter gekozen worden of kan deze onderverdeeld worden in meerdere banden. De lengte van de trials kunnen gekozen worden en kunnen eventueel verder opgesplitst worden. Er kan geopteerd worden om al dan niet regularisatie toe te passen bij de classifier. Door het gebruik van CSP kan het aantal filterparen ook een invloed hebben op de accuraatheid van de BCI.

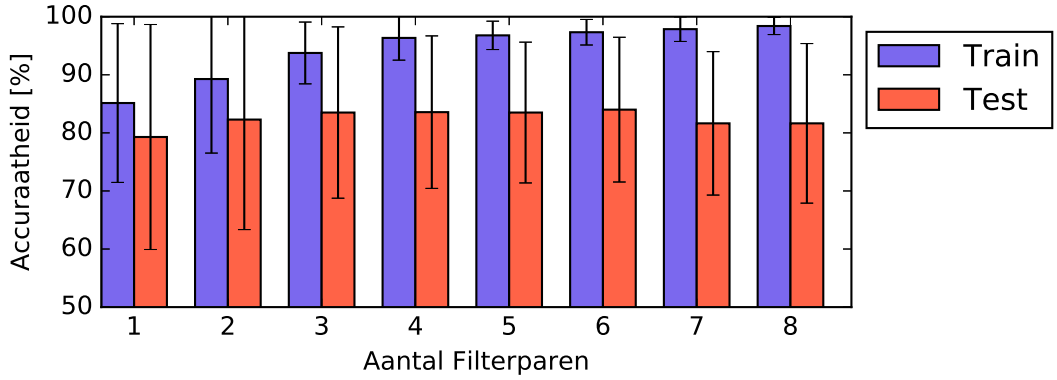
In de volgende subsecties worden enkele parameters besproken en geoptimaliseerd. Dit wordt gezien als een sequentieel proces, waardoor de optimale parameter van een bepaalde test gebruikt

wordt om een volgende parameter mee te optimaliseren. Eerst zal het aantal filterparen van de CSP filter geoptimaliseerd worden. Hierna zal het effect van regularisatie bekeken worden, de bandbreedte van de banddoorlaatfilter benaderd worden en de resultaten bekomen bij opdeling in frequentie- en tijdsdomein vergeleken worden.

Een alternatieve aanpak van optimalisatie zou zijn om alle parameters parallel te optimaliseren [66]. Om de computationele last hiervan te verlagen, zou kunnen gebruik gemaakt worden van een steekproef systeem waarbij een uiteindelijke schatting gemaakt kan worden van de optimale parameters. Dit type van optimaliseren wordt echter niet uitgevoerd binnenin deze thesis, maar kan gezien worden als een verdere uitbreiding.

Filterparen

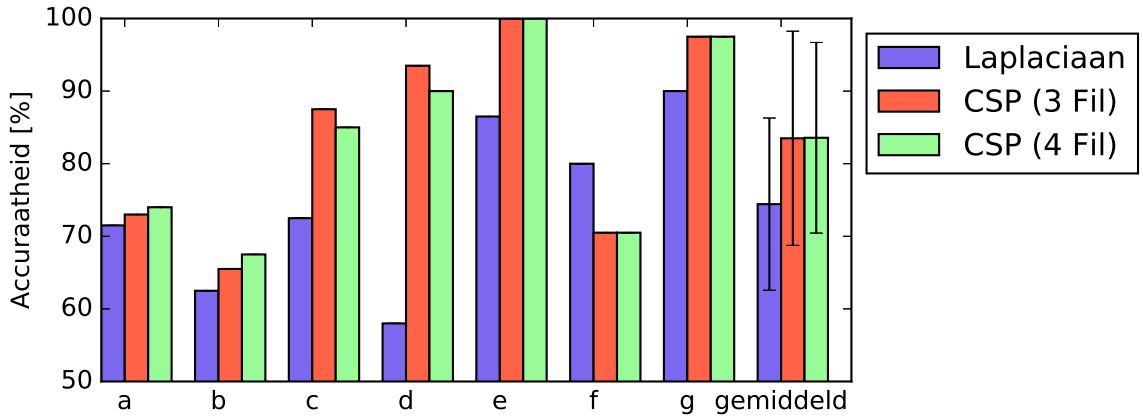
Een eerste optimalisatie die doorgevoerd wordt is bij het aantal filterparen. Initieel wordt gewerkt met 3 filterparen, doordat dit een veelgebruikt aantal is in literatuur (zie sectie 2.3.2). Bij deze test wordt de gemiddelde train en test accuraatheid bepaald over verschillende BCI, waarbij een variatie gemaakt wordt in het aantal filterparen voor CSP. Deze BCI maken gebruik van regularisatie. Bij deze test worden BCI opgebouwd met een variatie in filterparen van 1 tot en met 8, bij een constante frequentieband van 3Hz tot 40Hz (zie figuur 4.4).



Figuur 4.4: Invloed van verandering aantal filterparen op test accuraatheid bij BCI met CSP filtering (80-20)

Uit deze resultaten wordt opgemerkt dat er zich een klein verschil voordoet in test accuraatheid bij het veranderen van het aantal filterparen. Dit verschil is echter redelijk minimaal, waardoor deze parameter gemiddeld gezien geen grote invloed heeft op de accuraatheid van deze BCI. Door

een afweging tussen de hoeveelheid kenmerken dat CSP als uitgang heeft bij een bepaald aantal filterparen en de gemiddelde test accuraatheid bekomen hiermee, wordt gekozen om verder te werken met 4 filterparen. In figuur 4.5 worden de BCI met SL en CSP (3 filterparen) zowel persoonsspecifiek als gemiddeld gezien vergeleken met de BCI met 4 filterparen.

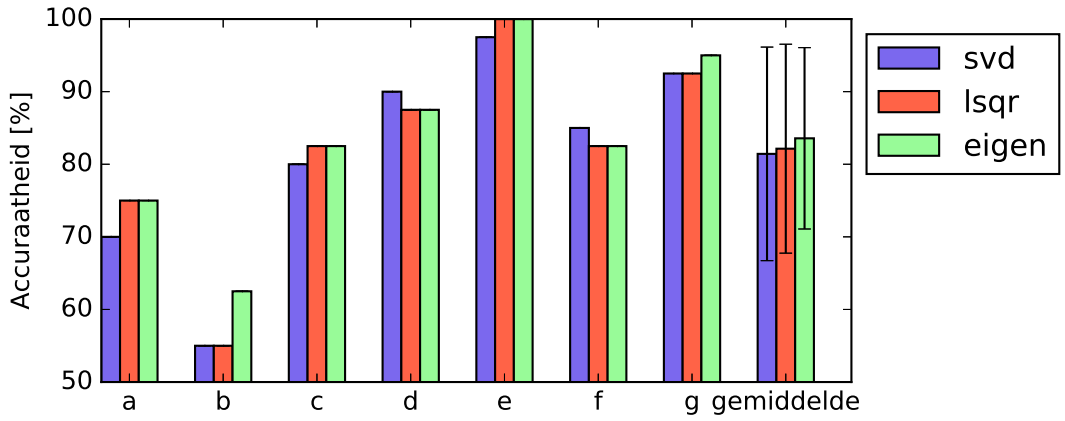


Figuur 4.5: Vergelijking in test accuraatheid tussen BCI met Laplaciaan en CSP filtering bij 3 en 4 filterparen (80-20)

Hierbij wordt gezien dat er weinig verschil is tussen BCI met CSP die 3 of 4 filterparen gebruiken. Bij sommige gebruikers scoren ze gelijk, bij andere scoort de ene iets beter dan de ander en vice versa. Gemiddeld gezien scoort de BCI met 4 filterparen (83.57%) net iets beter dan die met 3 filterparen (83.50%), waardoor de keuze voor 4 filterparen bevestigd wordt.

LDA solvers

Bij LDA is het mogelijk om via verschillende algoritmen (genaamd solvers) de nodige coëfficiënten te berekenen om zo een onderscheid te kunnen maken tussen verschillende klassen. Tot nu toe werd gewerkt op basis van de 'eigen' solver met regularisatie (zie sectie 2.4.5), daar dit aanbevolen werd in literatuur [38]. De scikit-learn library laat echter ook toe om andere solvers te gebruiken, die een alternatieve aanpak hebben bij het classificeren met LDA. In figuur 4.6 worden de resultaten bekomen met deze drie solvers - Singular value decomposition (svd), Least squares solution (lsqr) en Eigenvalue decomposition (eigen) - vergeleken met elkaar, waarbij shrinkage met een parameter op basis van het Ledoit and Wolf algoritme [59] gebruikt wordt bij de lsqr en eigen solver.

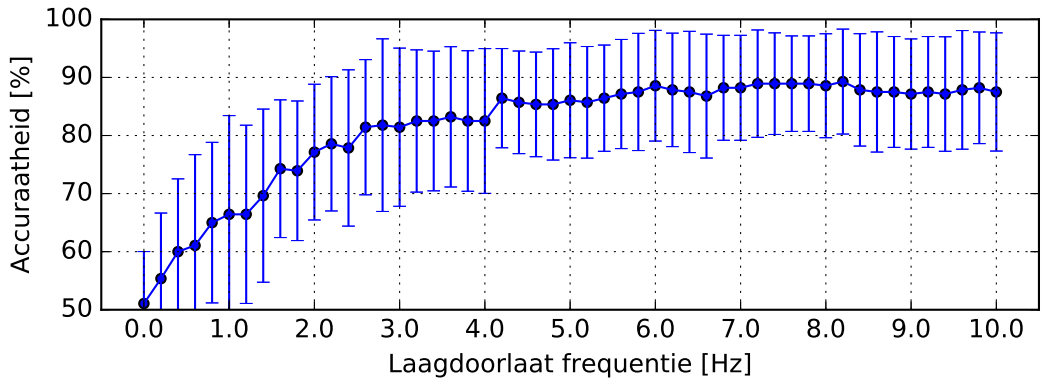


Figuur 4.6: Invloed van regularisatie op test accuraatheid bij een BCI met CSP filtering (80-20)

Bij het nader vergelijken van deze verschillende aanpakken, kan gemiddeld gezien worden dat de initiële keuze de beste was (83.57%), ondanks het verschil met andere solvers minimaal is (81.43% en 82.14%). Volgende testen gebruiken dus ook LDA met de 'eigen' solver, gecombineerd met regularisatie.

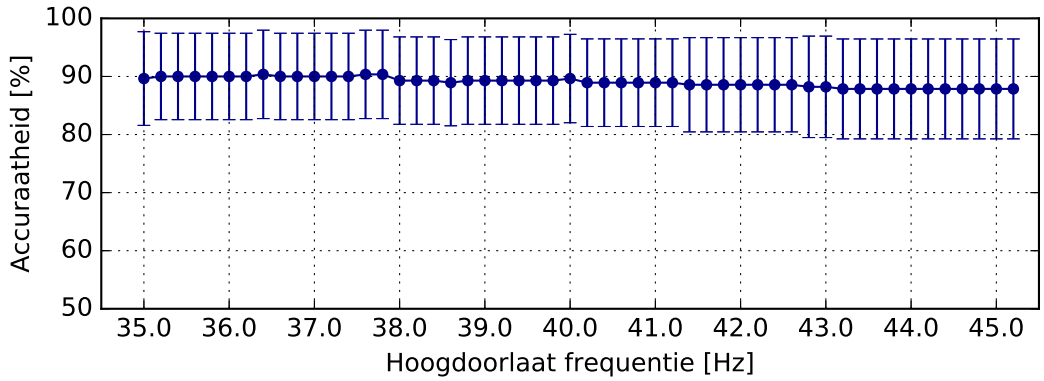
Frequentieband

Een volgende optimalisatie die doorgevoerd wordt, is die op de frequentieband waarin de band-doorlaatfilter omschreven in sectie 2.3.1 zal filteren. In voorgaande experimenten werd gewerkt met een bandbreedte van [3Hz - 40Hz]. In dit experiment wordt eerst de laagdoorlaat frequentie geoptimaliseerd, om vervolgens met die waarde de optimale hoogdoorlaat frequentie te bepalen.



Figuur 4.7: Invloed van verandering laagdoorlaat frequentie op gemiddelde test accuraatheid bij een BCI met CSP filtering (80-20). De hoogdoorlaat frequentie bij deze test is 40Hz.

Uit figuur 4.7 kan heel duidelijk afgeleid worden dat de laagdoorlaat frequentie een enorm grote invloed heeft op de accuraatheid van de BCI. Vooral in het bereik van 0Hz tot 4Hz wordt opgemerkt dat de accuraatheid stijgt van 51% tot 82.5%. Dit kan te wijten zijn aan eventuele ruissignalen die zich bevinden rond de 0Hz. Bij verdere analyse van deze gegevens, wordt opgemerkt dat de hoogst gemiddelde waarde (88.92%) bekomen wordt bij een laagdoorlaat frequentie van 7.6Hz. Dit komt overeen met de 8Hz die beschreven wordt in literatuur [67]. Met de waarde van 7.6Hz wordt vervolgens de hoogdoorlaat frequentie geoptimaliseerd.



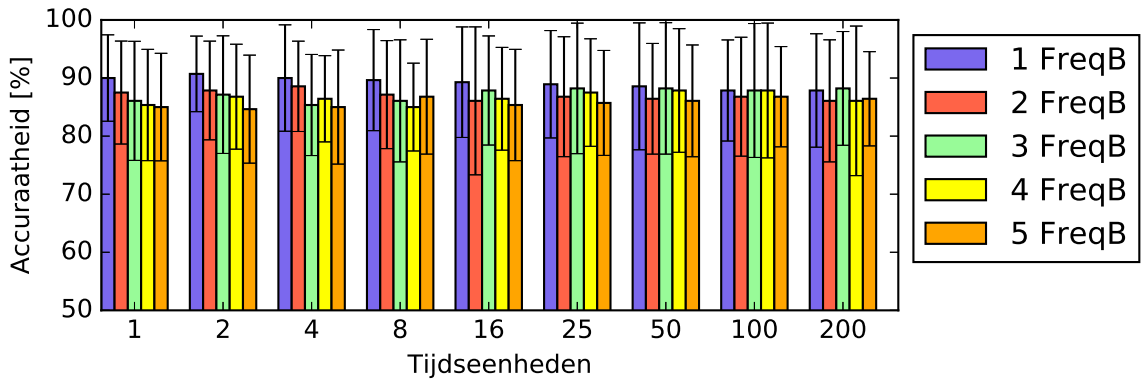
Figuur 4.8: Invloed van verandering hoogdoorlaat frequentie op gemiddelde test accuraatheid bij een BCI met CSP filtering (80-20). De laagdoorlaat frequentie bij deze test is 7.6Hz

Bij het bekijken van figuur 4.8 wordt er opgemerkt dat er amper een wijziging is in test accuraatheid wanneer de hoogdoorlaat frequentie varieert over een bereik van [35Hz - 45Hz]. Er kan dus gesteld worden dat deze heel weinig invloed heeft op de accuraatheid van de BCI binnenin een bereik van [35Hz - 45Hz]. Daar in de richting van 35Hz een hele lichte stijging opgemerkt wordt, is deze niet significant genoeg om een beduidend verschil te maken. Bij deze test wordt 36.4Hz gekozen als de hoogdoorlaatfrequentie waarbij een maximale test accuraatheid (90.36%) behaald werd.

De bekomen frequentieband voor de banddoorlaatfilter waardoor de EEG-data gefilterd moet worden, wordt bijgevolg geplaatst op [7.6Hz - 36.4Hz]. Wanneer dit vergeleken wordt met literatuur, wordt opgemerkt dat deze een frequentieband van [8Hz - 35Hz] voorstellen [67]. Daar de bekomen frequentieband na optimalisatie weinig verschilt van de voorgestelde waarden in literatuur, wordt besloten om met een frequentieband van [8Hz - 35Hz] verder te werken.

Opdeling in frequentie- en tijdsdomein

Een laatste parameter waarvoor optimalisatie bestudeerd wordt, is de opdeling in het frequentie- en tijdsdomein. Met opdeling in het tijdsdomein wordt bedoeld dat de trial opgedeeld wordt in gelijke delen, om deze vervolgens individueel toe te voegen als extra kenmerken. Met opdeling in het frequentiedomein wordt een gelijkaardige aanpak gebruikt, waarbij de originele frequentieband gelijk opgedeeld wordt in aparte banden, met als doel extra kenmerken te creëren.
In figuur 4.9 wordt op de x-as weergeven in hoeveel delen de trial opgesplitst wordt. De balken per tijdseenheid geven vervolgens de opsplitsing in het frequentiedomein weer in aantal frequentiebanden.



Figuur 4.9: Invloed van frequentie- en tijdsindeling op test accuraatheid bij een BCI met CSP filtering (80-20)

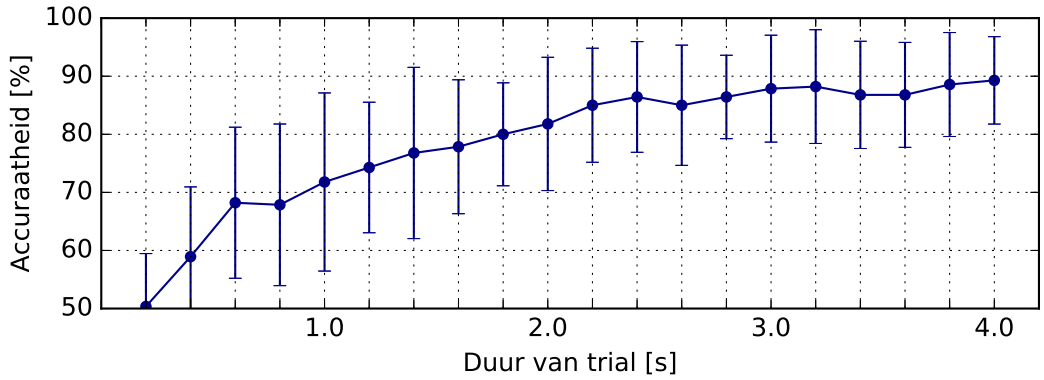
Over het algemeen kan besloten worden dat beide opdelingen weinig effect hebben op de accuraatheid van de BCI. Opdeling van de frequentieband in gelijke delen heeft in dit geval bovendien een negatief effect op de accuraatheid. Om de complexiteit en het aantal kenmerken van de BCI zo laag mogelijk te houden, wordt geopteerd om te blijven werken zonder opdeling in het frequentie- of tijdsdomein.

Een alternatieve aanpak bij de opdeling in het frequentie- en tijdsdomein zou zijn om enkel de relevante delen uit beide domeinen te halen en hierop apart enkele classifiers te trainen [68]. Hierna zou via een combinatie van de voorspelde waarden een uiteindelijk besluit gevormd kunnen worden over welke klasse nu aan de orde is. Deze manier van werken wordt echter niet geëvalueerd binnenin deze thesis, maar kan gezien worden als een verdere uitbreiding.

Duur van een trial

Een parameter die in deze thesis niet geoptimaliseerd wordt, is de lengte van een trial. Doordat er gebruik gemaakt wordt van datasets met reeds opgenomen data, is er tijdsdata beschikbaar over wanneer een pijl getoond werd aan de gebruiker. Er wordt van uit gegaan dat de gebruiker zich op dat moment bewust was van de richting van die pijl en dat deze geprobeerd zal hebben op zich een overeenkomstige beweging in te beelden. Doordat de gebruiker een bepaalde reactietijd nodig heeft om zich deze beweging effectief in te beelden, wordt het begin van de trial 0.5 seconden na het tonen van de pijl geplaatst. Vervolgens zal volgens informatie verschaft door de dataset de gebruiker zich de beweging 4 seconden lang inbeelden.

Daar de duur van een trial een impact heeft op de accuraatheid bij offline experimenten, heeft het ook een grote impact op de snelheid bij online experimenten. Het is namelijk niet gewenst om de gebruiker van het systeem zich in ware-tijd 4 seconden lang een bepaalde beweging te laten inbeelden, vooraleer er feedback van het systeem gegeven wordt. In figuur 4.10 wordt een evaluatie gedaan van de geoptimaliseerde BCI bij een variatie in de tijdsduur van een trial. Hierbij wordt gewerkt met een range van 0.2 tot en met 4 seconden.



Figuur 4.10: Invloed van trial duur op de gemiddelde test accuraatheid

Op figuur 4.10 wordt duidelijk weergeven dat wanneer er een langere duur genomen wordt voor een trial, de accuraatheid van het systeem verhoogt. Dit is ook logisch, aangezien de BCI bij een langere trialduur meer data heeft om een beslissing van klasse (links of rechts) mee te maken. Ook wordt opgemerkt dat bij een trialduur van slechts 1.8s, er reeds een accuraatheid van 80% bekomen wordt. Dit is slechts een daling van ongeveer 10% accuraatheid vergeleken met de resultaten bekomen bij een trialduur van 4s, terwijl de trialduur hierbij met 55% daalt.

4.3 Transfer learning

In deze sectie worden verschillende transfer learning algoritmen uit zowel de literatuur als eigen methoden getest. Bij de opbouw van deze BCI wordt teruggekeken naar de voorgaande sectie, waarbij een geoptimaliseerde BCI uitgebouwd werd. Het aantal filterparen wordt dus initiëel geplaatst op 4. Een soortgelijke optimalisatie als in de voorgaande sectie wordt ook doorgevoerd bij de datasets van Physionet (zie sectie 2.2.3). Hierbij wordt gekeken wat gemiddeld gezien de optimale frequentieband is en of een indeling in frequentie- of tijdsdomein een invloed heeft op de accuraatheid.

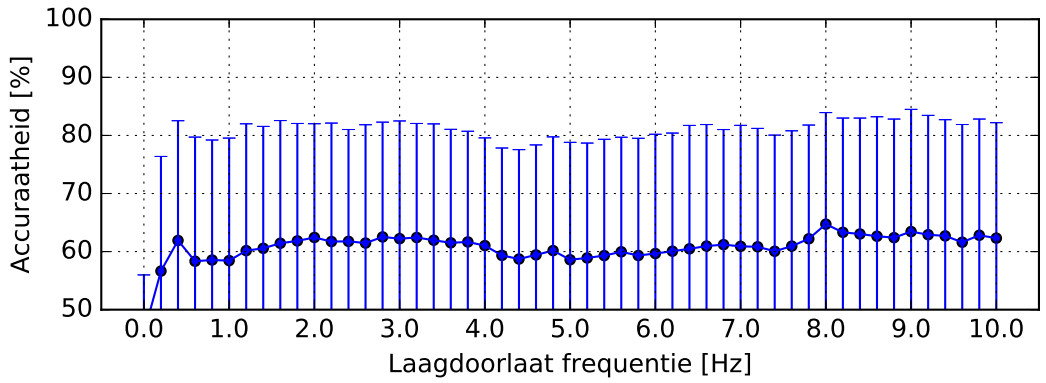
Bij het evalueren van transfer learning algoritmen wordt een onderverdeling gemaakt tussen nieuwe gebruikers en eerdere gebruikers. Bij de nieuwe gebruikers wordt de dataset van de BCI Competition IV gebruikt, zonder proefpersonen *a* en *f*. Bij deze twee proefpersonen wordt namelijk het onderscheid gemaakt tussen linker beweging en beweging van de voeten, in plaats van linker en rechter beweging. Als eerdere gebruikers worden de datasets van Physionet gebruikt. Hierbij worden 106 van de 109 proefpersonen gebruikt (zonder S88, S92 en S100) doordat sommige datadelen in de drie overige datasets corrupt blijken te zijn.

In de volgende subsecties zullen eerst enkele optimalisatiestappen uitgevoerd worden op BCI die gebruik maken van proefpersonen uit de Physionet datasets. Hierna zullen enkele methoden vanuit de literatuur getest en vergeleken worden. Vervolgens zullen enkele eigen opgestelde algoritmen geëvalueerd worden om ten slotte een vergelijking te maken tussen de literatuur en eigen methodologieën.

4.3.1 Optimalisatie van parameters voor eerdere gebruikers

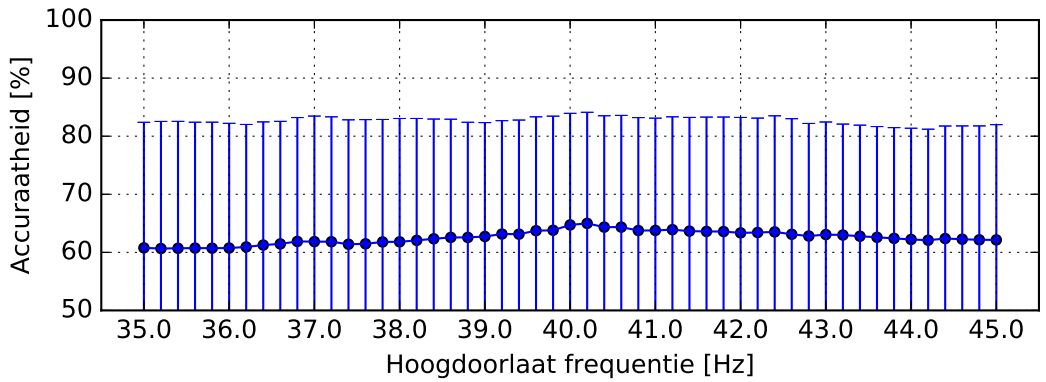
Frequentieband

Als eerste optimalisatie wordt gekeken wat gemiddeld gezien de meest optimale frequentieband is om een banddoorlaatfilter mee te maken voor de proefpersonen uit de Physionet datasets. Hierbij wordt eerst de laagdoorlaat frequentie benaderd, waarbij de hoogdoorlaat frequentie opnieuw op 40Hz geplaatst wordt. Voor de laagdoorlaat frequentie wordt een bereik doorlopen van 0Hz tot en met 10Hz.



Figuur 4.11: Invloed van verandering laagdoorlaat frequentie op gemiddelde test accuraatheid. De hoogdoorlaat frequentie bij deze test is 40Hz.

In figuur 4.11 wordt een stijging in accuraatheid opgemerkt wanneer de de laagdoorlaat frequentie wat hoger ligt dan 0Hz. Zoals eerder vermeld in sectie 4.2.3, ligt de oorzaak hiervan mogelijks bij eventueel aanwezige ruis rond deze frequentie. Er wordt echter geen groot verschil opgemerkt bij verdere verandering van de laagdoorlaat frequentie. De oorzaak hiervan kan liggen bij het feit dat er over vele proefpersonen uitgemiddeld wordt en dat de laagdoorlaat frequentie een meer persoonsspecifieke invloed heeft op de accuraatheid van de BCI. Er wordt echter wel een lichte stijging in accuraatheid gezien bij een laagdoorlaat frequentie van 8Hz. Daar deze overeenkomt met wat besproken werd in sectie 4.2.3 en met de literatuur [67] wordt deze waarde verder gebruikt bij het optimaliseren van de hoogdoorlaat frequentie. Hierbij wordt een bereik van 35Hz tot en met 45Hz doorlopen.

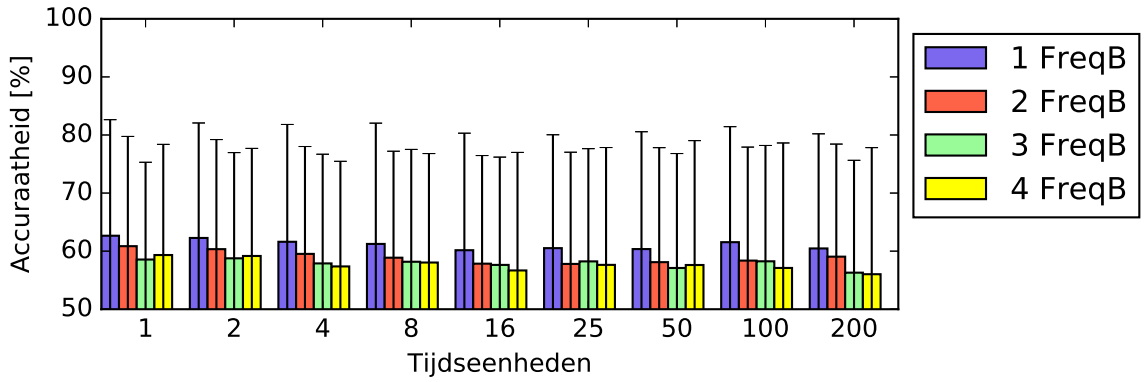


Figuur 4.12: Invloed van verandering hoogdoorlaat frequentie op gemiddelde test accuraatheid. De laagdoorlaat frequentie bij deze test is 8Hz.

Al snel wordt opgemerkt dat er opnieuw weinig verandering in test accuraatheid optreedt wanneer de hoogdoorlaat frequentie verandert. Deze parameter heeft dus gemiddeld gezien weinig invloed op de test accuraatheid van de BCI, zoals reeds besloten werd in sectie 4.2.3. Ondanks opgemerkt wordt dat er een hogere accuraatheid behaald wordt bij 40Hz, wordt toch besloten om de frequentieband te plaatsen op [8Hz - 35Hz]. Dit om uniform te blijven met de reeds bekomen resultaten en de literatuur [67].

Opdeling in frequentie- en tijdsdomein

Een tweede parameter die bestudeerd wordt voor optimalisatie is de opdeling in frequentie- en tijdsdomein. Zoals reeds uitgelegd in sectie 4.2.3 wordt met opdeling in het tijdsdomein bedoeld dat de trial opgedeeld wordt in gelijke delen, om deze vervolgens individueel toe te voegen als extra kenmerken. Met opdeling in het frequentiedomein wordt weer een gelijkaardige aanpak gebruikt, waarbij de originele frequentieband gelijk opgedeeld wordt in aparte banden, met als doel extra kenmerken te creëren.



Figuur 4.13: Invloed van frequentie- en tijdsindeling op test accuraatheid

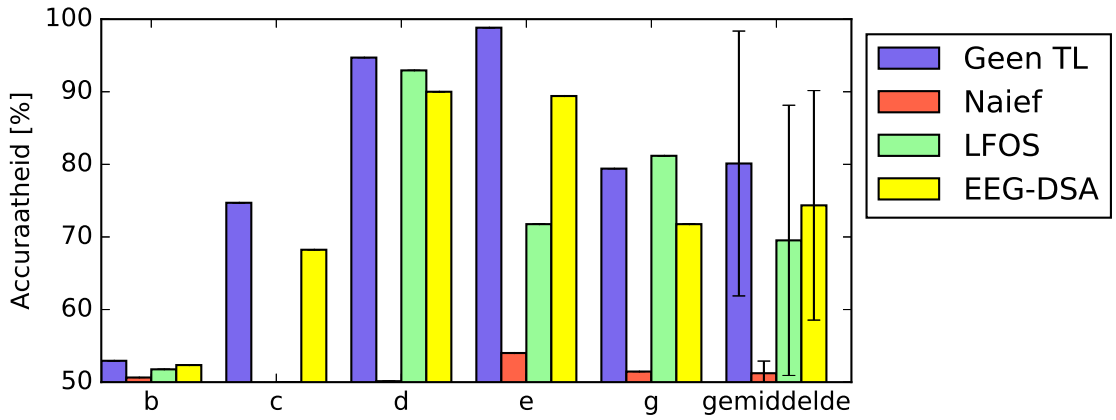
Op figuur 4.13 wordt opgemerkt dat deze parameter met de voorgeschreven manier van werken heel weinig invloed heeft op de accuraatheid van de BCI. Om gelijk te lopen met wat reeds besloten werd in sectie 4.2.3, wordt er verder gewerkt zonder opdeling in het frequentie- of tijdsdomein.

In de volgende sectie zullen enkele methoden uit de literatuur geëvalueerd worden aan de hand van nieuwe en eerdere gebruikers, zoals beschreven in de inleiding van sectie 4.3. Bij deze eerdere gebruikers zal de volledige dataset gebruikt worden bij het trainen van de BCI. Vervolgens zal bij de nieuwe gebruiker een opdeling gemaakt worden van 15-85, waarmee bedoeld wordt dat

15% van de data (30 trials) van de nieuwe gebruiker gebruikt wordt als kalibratiedata, terwijl de overige 85% van de data (170 trials) aanzien wordt als ware-tijdsdata. De test accuraatheden behaald in alle testen die volgen worden dan ook bepaald op die overige 85% van de data, tenzij anders vermeld.

4.3.2 Algoritmen uit de literatuur

In sectie 2.6.1 werden enkele subject-to-subject transfer learning algoritmen besproken uit de literatuur. Deze worden nu geëvalueerd, zodat een goede richtlijn bekomen wordt om eigen algoritmen mee te vergelijken. Deze test wordt uitgevoerd op basis van de reeds geoptimaliseerde parameters, waarbij 4 filterparen, regularisatie met de 'eigen' solver, een frequentieband van [8Hz - 35Hz] voor de banddoorlaatfilter en geen opsplitsing in het frequentie- of tijdsdomein gebruikt worden.



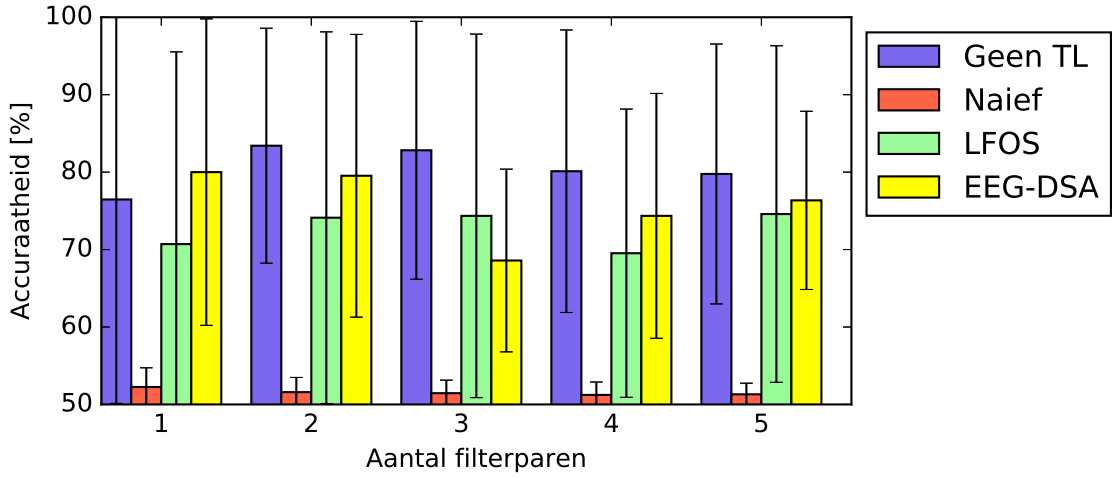
Figuur 4.14: Test accuraatheid voor verschillende transfer learning algoritmen uit de literatuur. Hierbij wordt gewerkt met 4 filterparen en wordt gebruik gemaakt van regularisatie.

De bekomen resultaten over de verschillende algoritmen worden gezien in figuur 4.14. Al snel wordt opgemerkt dat er een groot verschil ontstaat tussen de naïeve methode en de rest. Vooraleer de bekomen resultaten verder besproken worden, zal er eerst een optimalisatie van enkele parameters doorgevoerd worden.

Optimalisatie voor transfer learning

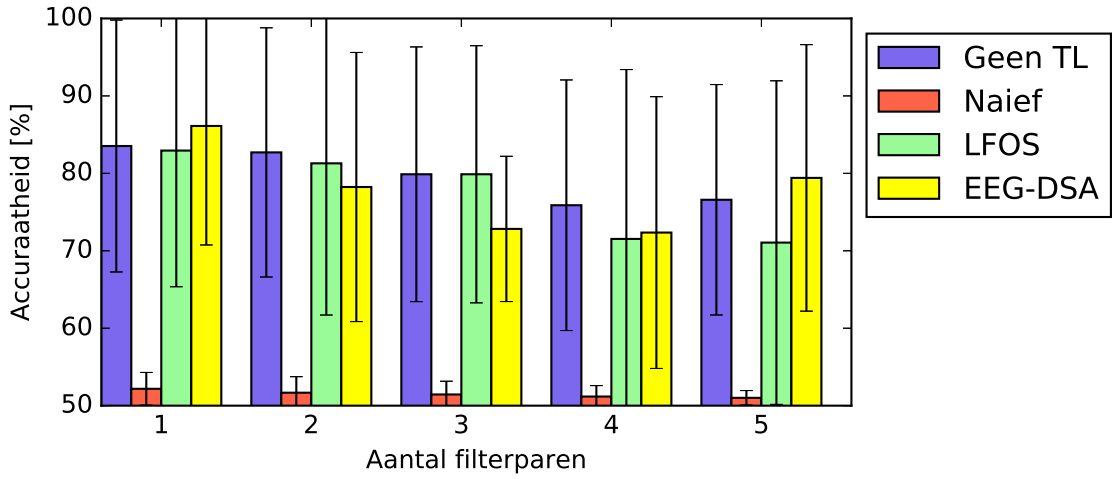
Ondanks dat er reeds een parameter optimalisatie uitgevoerd werd voor zowel de nieuwe als de eerdere gebruikers individueel, is het ook van belang om de invloed van bepaalde parameters te gaan bekijken in het belang van transfer learning. Het kan namelijk mogelijk zijn dat andere parameters ervoor zorgen dat er een betere overdracht van informatie plaatsvindt, ook al werkt dit individueel niet beter.

Een eerste parameter die in dit kader bekeken wordt, is de invloed van regularisatie op de test accuraatheid van de resultaten. De tweede parameter die hierbij gepaard geoptimaliseerd wordt, is het aantal filterparen. De resultaten bekomen door het aantal filterparen te laten variëren tussen 1 en 5 met en zonder regularisatie worden weergeven in figuren 4.15 en 4.16.



Figuur 4.15: Test accuraatheid voor verschillende transfer learning algoritmen uit de literatuur, bij het veranderen van het aantal filterparen. Bij de opbouw van de BCI werd **regularisatie** gebruikt.

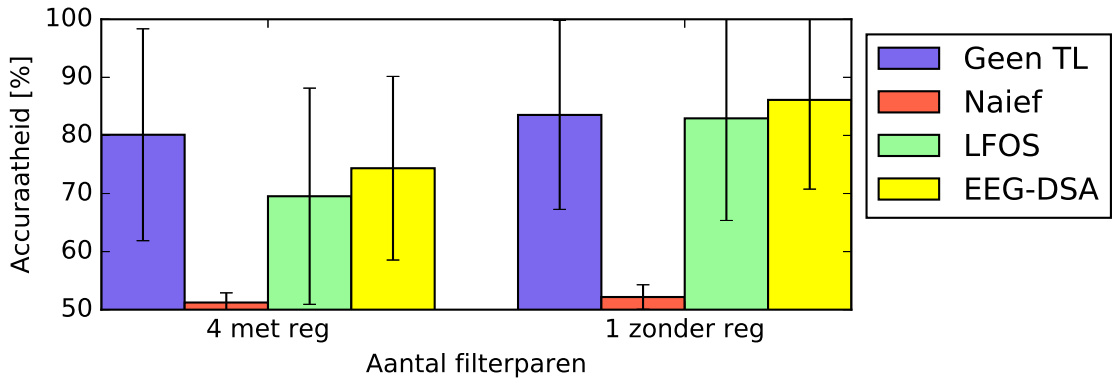
Bij een vergelijking van beide figuren kan duidelijk gezien worden dat er zich een verschil voordoet tussen de test accuraatheden bij 1 filterpaar zonder regularisatie, dan bij de andere. Ook wordt opgemerkt bij de methode zonder transfer learning, dat bij 1 en 2 filterparen zonder regularisatie de hoogste accuraatheden bekomen worden (83.53% en 82.71%) en bij 2 en 3 filterparen met regularisatie gelijkaardige resultaten bekomen worden (83.41% en 82.82%). Dit kan verklaard worden door de curse of dimensionality. In sectie 2.4.3 werd namelijk gesteld dat



Figuur 4.16: Test accuraatheid voor verschillende transfer learning algoritmen uit de literatuur, bij het veranderen van het aantal filterparen. Bij de opbouw van de BCI werd **geen regularisatie** gebruikt.

bij LDA het aantal training samples op zijn minst tien maal het aantal kenmerken moet zijn. Daar deze verschillende BCI getrained worden met 40 tot 45 training samples, mag het aantal kenmerken dus maximaal 4 zijn, wat overeenkomt met 1 of 2 filterparen. Uit de resultaten op figuur 4.15 en 4.16 wordt dan ook besloten dat er verder gewerkt wordt met 1 filterpaar voor CSP en dat er geen gebruik gemaakt wordt van regularisatie bij de classifier. Ter vergelijking worden de bekomen resultaten met deze parameters in figuur 4.17 nog eens vergeleken met degene bekomen in 4.14.

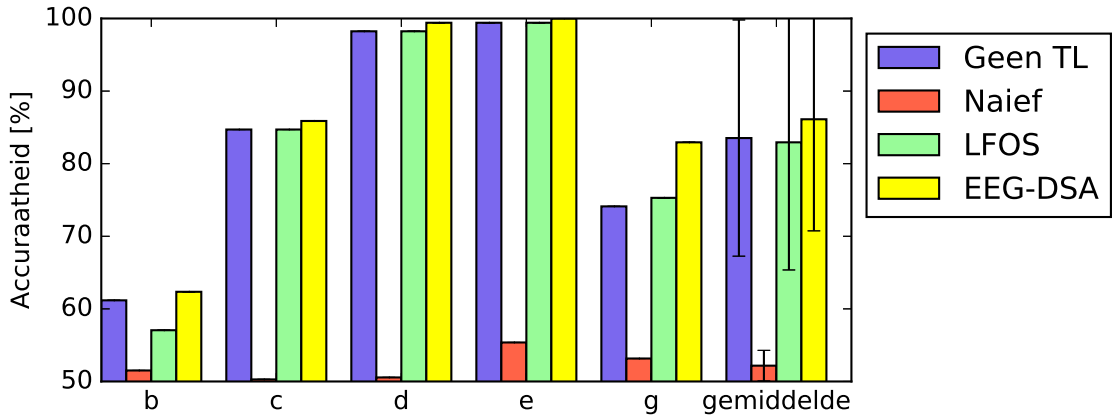
Uit de vergelijking in figuur 4.17 tussen de behaalde accuraatheden voor en na de optimalisatie, wordt besloten dat de keuze van deze parameters een enorm grote invloed kan hebben op de accuraatheid van de opgestelde BCI. De bekomen test accuraatheden bij LFOS stijgen van 69.53% tot 82.94% en bij EEG-DSA van 74.35% tot 86.12%. Zelfs de methode die geen gebruik maakt van transfer learning, stijgt van 80.12% tot 83.53%. Hierdoor wordt een BCI met 1 filterpaar zonder gebruik van regularisatie aanzien als de optimale manier om transfer learning bij uit te voeren.



Figuur 4.17: Test accuraatheid voor verschillende transfer learning algoritmen uit de literatuur, bij vergelijking van 4 filterparen & regularisatie en 1 filterpaar & geen regularisatie.

Bespreking resultaten

Nu een optimale setting in het kader van transfer learning gecreëerd werd, worden de bekomen resultaten nog eens persoonsspecifiek bekeken (zie figuur 4.18). Deze resultaten worden hieronder per algoritme individueel besproken.



Figuur 4.18: Test accuraatheid voor verschillende transfer learning algoritmen uit de literatuur. Hierbij wordt gewerkt met 1 filterpaar en wordt geen gebruik gemaakt van regularisatie.

Zonder transfer learning (15 - 85) De methode *zonder transfer learning* zoals beschreven in sectie 3.3.1 maakt gebruik van de beschikbare kalibratiedata (15%) om de BCI op te trainen. Vervolgens wordt de BCI getest op de overige data (85%) van de gebruiker. Hierdoor kan gezien worden of een BCI ook goed gekalibreerd kan worden met weinig kalibratiedata en hoe goed dit scoort ten opzichte van transfer learning algoritmen.

Uit de resultaten op figuur 4.18 wordt opgemerkt dat de test accuraatheden bekomen met deze manier van werken in dezelfde lijn liggen als die bekomen bij transfer learning algoritmen uit de literatuur. Dit komt doordat er reeds een voldoend aantal kalibratietrials aanwezig is (30 trials) om een robuuste BCI mee te bouwen.

Naïeve methode Het meest eenvoudige transfer learning algoritme dat geëvalueerd wordt, is de naïeve methode. Bij deze manier van werken wordt snel opgemerkt dat de resultaten heel slecht zijn ten opzichte van de andere. Er kan dus gesteld worden dat de naïeve methode niet efficiënt is wanneer er vele eerdere gebruikers zijn. Doordat veel BCI niet overeenkomstig zullen zijn met de nieuwe gebruiker, zullen veel BCI functioneren als gokkers. Door vervolgens uit te middelen over alle BCI worden nagenoeg dezelfde resultaten bekomen als bij een gokker.
Het algoritme heeft ook geen weet van de compatibiliteit van de eerdere gebruikers ten opzichte van de nieuwe gebruiker, waardoor het niet in staat is om de optimale BCI te selecteren uit alle opgestelde BCI.

LFOS Gemiddeld gezien liggen de resultaten voor LFOS in figuur 4.18 iets lager dan die bekomen door EEG-DSA. Individueel gezien wordt echter opgemerkt dat er weinig verschil is tussen de bekomen accuraatheden van LFOS en EEG-DSA, behalve bij gebruiker g.
Een belangrijk punt bij het uitvoeren van deze test is dat slechts de eerste 10 eerdere gebruikers van de Physionet datasets gebruikt werden om de resultaten voor LFOS te verkrijgen. Dit is te wijten aan de computationele complexiteit van het algoritme.

EEG-DSA EEG-DSA of Data Space Adaption scoort gemiddeld gezien het best bij de bekomen resultaten. Ook individueel wordt deze altijd als winnaar naar voor geschoven in vergelijking met de verschillende algoritmen. Hierdoor wordt dit algoritme aanzien als de richtlijn bij vergelijking tussen eigen algoritmen en de literatuur.

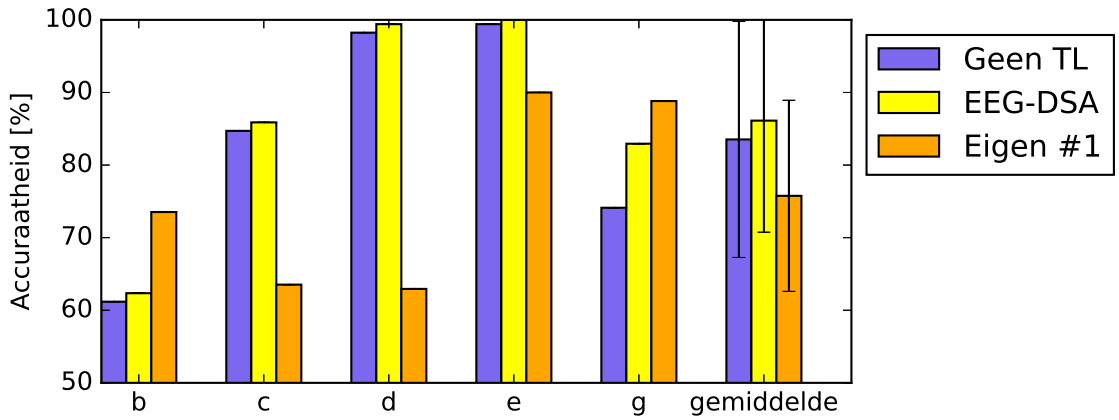
Nu een optimalisatie doorgevoerd werd van enkele parameters in het kader van transfer learning, kunnen deze gebruikt worden als basis om een eigen transfer learning algoritme op te stellen. De reeds bekomen resultaten met methoden uit de literatuur kunnen hierbij gebruikt worden als richtlijn. In de volgende sectie zullen de doorlopen stappen besproken worden bij het opstellen van het RSAMS algoritme en zullen enkele gemaakte keuzes tijdens dit proces verduidelijkt worden.

4.3.3 Eigen ontwerpen

In deze sectie zullen de verschillende algoritmen en methoden uitgelegd in hoofdstuk 3 geëvalueerd worden. Hierbij zal ook een beeld gevormd worden van de gemaakte keuzes tijdens de ontwikkeling van het RSAMS algoritme. Daar nieuwe uitvoeringen verder bouwen op voorgaande, zal elke uitvoering gedetailleerd besproken worden.

Eigen algoritme 1

Het eerste algoritme dat geëvalueerd wordt, is degene die beschreven wordt in sectie 3.3.2. Hierbij wordt de volledige dataset van een eerdere gebruiker gebruikt om de BCI mee te trainen. Vervolgens wordt de BCI getest met de overige 85% van de nieuwe gebruiker. Bij dit algoritme wordt de eerdere gebruiker geselecteerd op basis van het sorteer systeem. Hierbij wordt via verschillende parameters een overeenkomstige gebruiker gezocht, door gebruik te maken van de kalibratie data van de nieuwe gebruiker. De gewichten voor de parameters van het sorteer systeem - Validatie accuraatheid, Consistentie index, MMD en LSED - worden respectievelijk op 2, 1, 1 en 1 geplaatst. Dit doordat de validatie accuraatheid een zeer goede maatstaf is voor de compatibiliteit van een gebruiker.

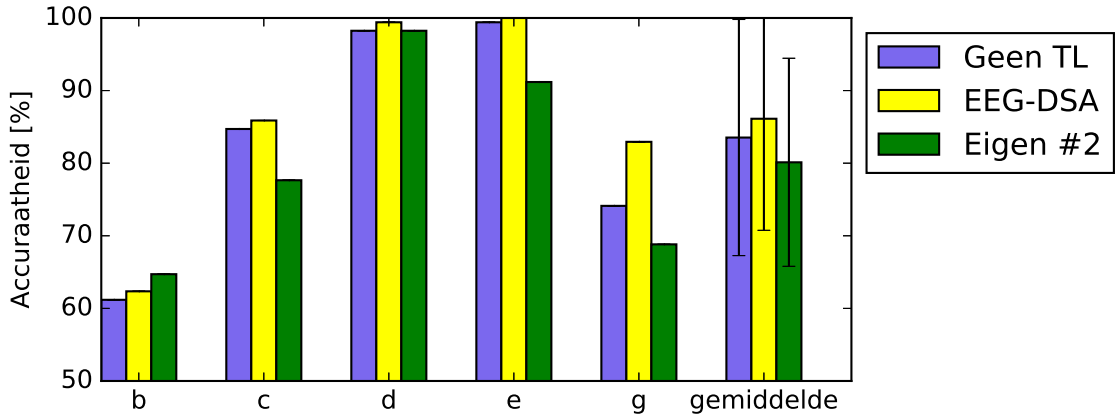


Figuur 4.19: Test accuraatheid bekomen met eigen algoritme 1, vergeleken met EEG-DSA en de methode die geen gebruik maakt van transfer learning

Aan de hand van figuur 4.19 kan gezien worden dat dit algoritme gemiddeld ongeveer 10% lager scoort dan de andere algoritmen waarmee het vergeleken wordt. Wanneer er echter individueel naar de gebruikers gekeken wordt, wordt er opgemerkt dat dit algoritme een stuk hoger scoort bij gebruiker *b* en *g*. Ondanks dat dit algoritme over het algemeen niet goed scoort, is het toch opmerkelijk dat het bij sommige individuën beter scoort dan de literatuur.

Eigen algoritme 2

Als alternatief op eigen algoritme 1 wordt een samensmelting gedaan tussen deze en EEG-DSA. Hierbij wordt de data van de eerdere gebruiker eerst getransformeerd naar de kalibratiedata van de nieuwe gebruiker toe, zoals beschreven in sectie 3.3.3. Hierna wordt met deze getransformeerde data een BCI opgesteld en worden dezelfde stappen als bij het voorgaande algoritme doorlopen. De gekozen gewichten voor het sorteer systeem zijn dan ook dezelfde als bij eigen algoritme 1.



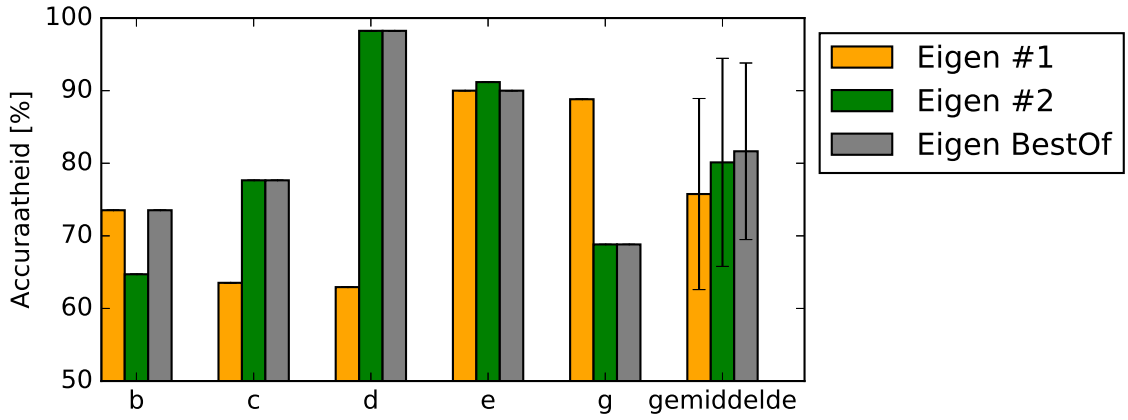
Figuur 4.20: Test accuraatheid bekomen met eigen algoritme 2, vergeleken met EEG-DSA en de methode die geen gebruik maakt van transfer learning

Op figuur 4.20 kan gezien worden dat dit algoritme gemiddeld gezien in de buurt komt van de test accuraatheid behaald door de andere methoden. Bij het bekijken van de individuele resultaten wordt echter gezien dat dit algoritme, behalve bij persoon b, altijd slechter scoort dan EEG-DSA. Niettemin is dit algoritme gemiddeld gezien een verbetering ten opzichte van eigen algoritme 1, wat aangetoond wordt in figuur 4.21.

Eigen algoritme (Beste van 2)

Uit de resultaten bekomen door eigen algoritme 1 en 2 wordt opgemerkt dat er bijna altijd een redelijk verschil aanwezig is tussen beide bij de test accuraatheden per persoon. Hieruit kan geconcludeerd worden dat beide elkaar complementeren. Op basis van dit idee, werd een soort *beste van 2* gemaakt zoals beschreven in sectie 3.3.4. De gebruikte gewichten voor deze parameters van het sorteer systeem - Validatie accuraatheid, Consistentie index, MMD en LSED - zijn respectievelijk 3, 4, 5 en 6. Deze gewichten werden manueel bepaald door analyse van de resultaten en parameterwaarden. Een verdere optimalisatie of automatisatie van de keuze

in gewichten maakt geen onderdeel van deze thesis uit, maar kan gezien worden als een verdere uitbreiding.



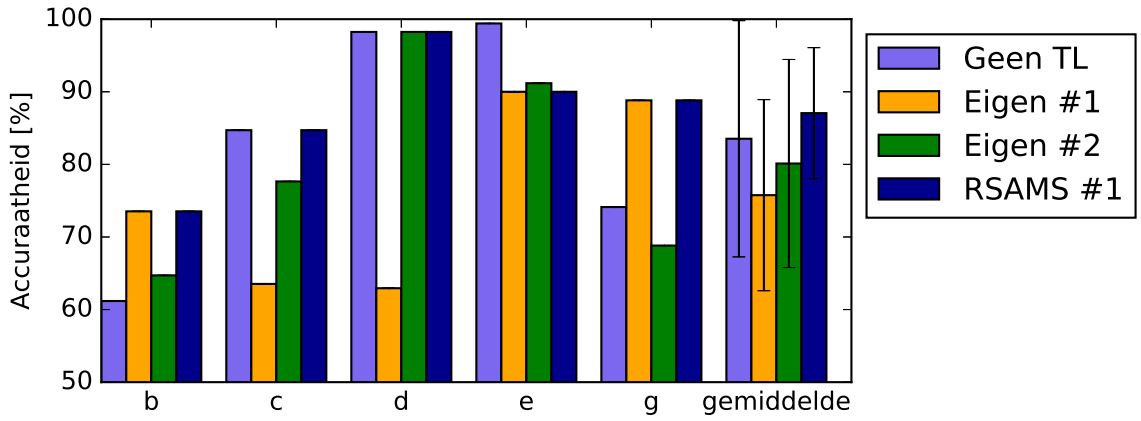
Figuur 4.21: Test accuraatheid bekomen met eigen algoritme (beste van 2), vergeleken met eigen algoritme 1 en 2 individueel

In figuur 4.21 kan een duidelijke vergelijking gezien worden tussen de drie reeds opgemaakte algoritmen. Gemiddeld gezien scoort eigen algoritme (beste van 2) het hoogst. Dit is logisch, aangezien het een afweging maakt tussen de twee andere algoritmen. Individueel gezien wordt bij eigen algoritme (beste van 2), behalve bij proefpersonen e en g, altijd het algoritme met de hoogste test accuraatheid genomen. Bij proefpersoon e scheelt de test accuraatheid amper met die van eigen methode 2. Bij proefpersoon g is het verschil tussen beste van 2 en eigen 1 echter redelijk groot. Dit kan een gevolg zijn van het niet genoeg optimaliseren van de gebruikte gewichten.

Ranking-Based Subject And Methodology Selection (RSAMS 1)

Het laatste algoritme dat geëvalueerd wordt, is het RSAMS algortime (zie sectie 3.4). Hierbij wordt niet alleen een afweging gemaakt tussen eigen algoritme 1 en 2, maar ook tussen het algoritme zonder transfer learning. Hierdoor kan de optimale manier van werken geselecteerd worden in de juiste situatie. Om verwarring met andere RSAMS uitwerkingen te vermijden, wordt deze uitvoering van het RSAMS algoritme RSAMS 1 genoemd.

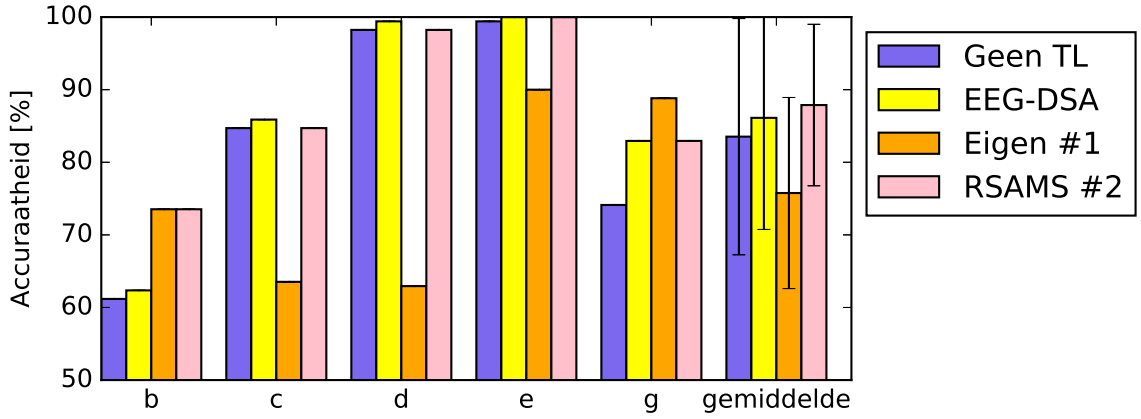
Op figuur 4.22 wordt aangetoond dat RSAMS 1 gemiddeld gezien het hoogste scoort. Individueel gezien is het ook zeer goed in staat om het beste algoritme naar voor te schuiven. Enkel bij proefpersoon e wordt de methode met de hoogst behaalde test accuraatheid niet geselecteerd.



Figuur 4.22: Test accuraatheid bekomen met RSAMS 1 ten opzichte van de drie algoritmen dat het algoritme met elkaar vergelijkt

RSAMS met EEG-DSA (RSAMS 2)

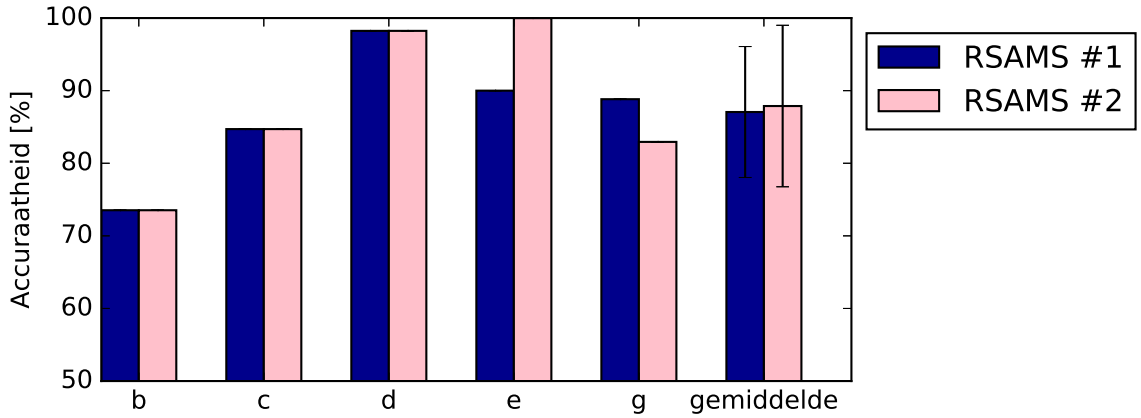
Daar RSAMS in staat is om een selectie te maken over verschillende methodologieën, wordt in deze uitvoering eigen algoritme 2 vervangen door EEG-DSA. Uit figuur 4.20 blijkt namelijk dat deze over het algemeen beter scoort dan eigen algoritme 2. Deze uitvoering van het algoritme wordt RSAMS 2 genoemd.



Figuur 4.23: Test accuraatheid bekomen met RSAMS 2 ten opzichte van de drie algoritmen dat het algoritme met elkaar vergelijkt

Ook bij deze uitvoering wordt in figuur 4.23 duidelijk gezien dat RSAMS goed in staat is om het optimale algoritme naar voor te schuiven. Bij proefpersoon d en g kiest RSAMS wel het

algoritme met de tweede hoogste test accuraatheid. Doordat het verschil tussen degene met de hoogste test accuraatheid klein is, vormt dit geen probleem.



Figuur 4.24: Vergelijking van RSAMS 1 en RSAMS 2

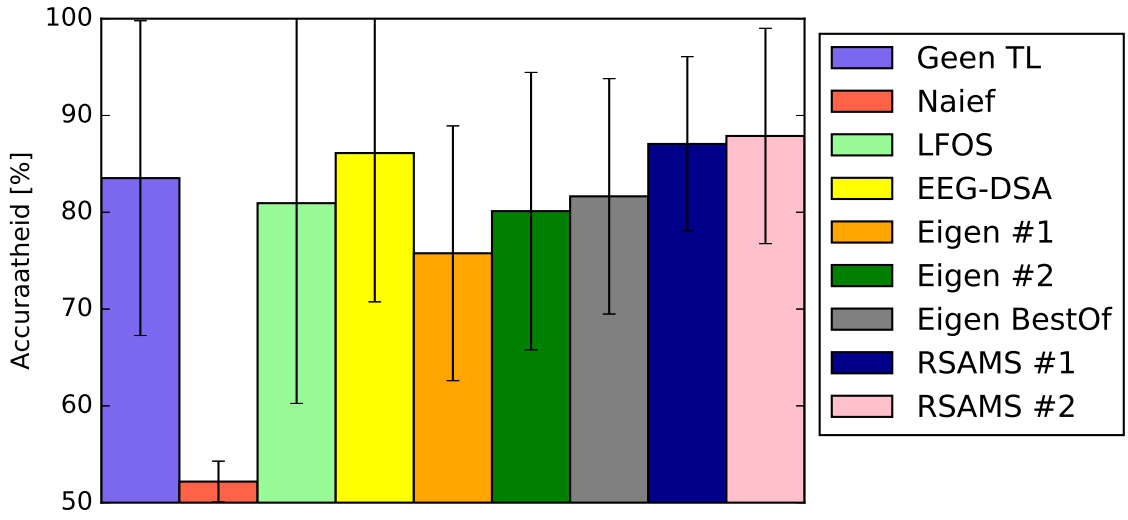
Wanneer beide uitvoeringen van het RSAMS algoritme vergeleken worden met elkaar in figuur 4.24, wordt gezien dat er weinig verschil is tussen de bekomen resultaten van beide. Enkel bij proefpersoon e en g worden verschillende resultaten bekomen. Er kan dus besloten worden dat zowel eigen algoritme 2 als EEG-DSA niet zo een grote invloed hebben op de resultaten van deze 5 testpersonen. Om na te gaan hoe groot de invloed van beide effectief is, zou getest moeten worden op een groter aantal testpersonen. Dit valt echter buiten de scoop van deze thesis.

4.3.4 Vergelijking

Nu alle algoritmen vanuit literatuur en eigen ontwikkeling geëvalueerd werden, worden ze in deze sectie vergeleken met elkaar. Hierbij wordt per proefpersoon terug 15% van de data gebruikt voor kalibratie en de overige 85% om op te testen. Zo kan besloten worden welke algoritmen gemiddeld gezien beter scoren dan de rest, in het geval van de 5 gebruikte testpersonen. De individuele resultaten van deze test worden weergeven in bijlage A.1.

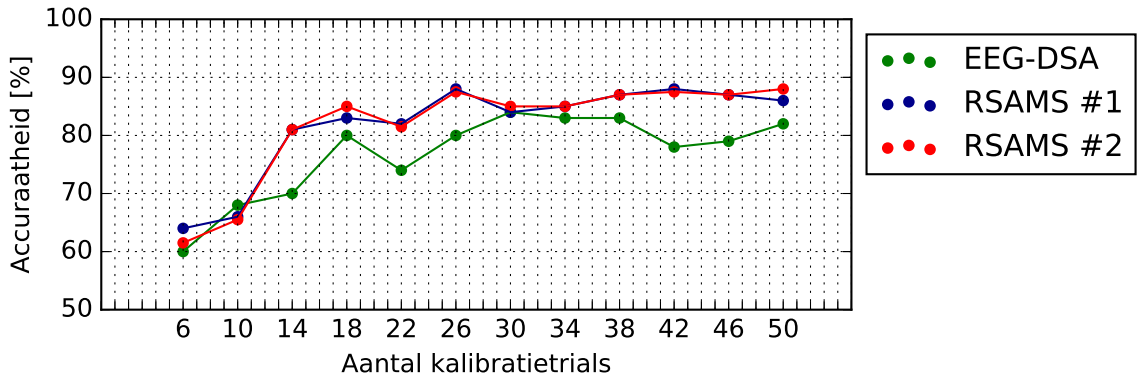
In de vergelijking over alle algoritmen, te zien op figuur 4.25, worden grote verschillen opgemerkt tussen de algoritmen. Zoals reeds besloten, scoort de naïëve manier het slechtst (52.18%). Hieropvolgend zijn eigen algoritme 1 en 2 de slechtst scorende algoritmen (75.76% en 80.12%). Dit is geen probleem, aangezien deze gecombineerd worden in het RSAMS algoritme. De resultaten bekomen zonder transfer learning (15 - 85) en LFOS liggen in hetzelfde bereik qua test accuraatheid (83.53% en 80.94%). Hierna komt EEG-DSA aan bod, die gemiddeld de hoog-

ste test accuraatheid behaald van alle transfer learning methoden uit de bestudeerde literatuur (86.12%). De hoogst scorende methoden zijn vervolgens beide RSAMS uitvoeringen (87.06% en 87.88%). Er kan dus besloten worden dat het RSAMS algoritme in deze test gemiddeld gezien beter scoort dan de literatuur, doordat de gemiddelde bekomen accuraatheid hoger ligt en de standaardafwijking een stuk lager ligt (15.37% voor EEG-DSA ten opzichte van 9.02% en 11.12% voor RSAMS 1 en 2 respectievelijk).



Figuur 4.25: Vergelijking over alle geëvalueerde methoden, uitgemiddeld voor alle proefpersonen

De probleemstelling van deze thesis (zie sectie 1.4) werd omschreven als de te lange duurtijd van kalibratie om een goed werkende BCI op te stellen. Om nu een idee te hebben van hoeveel kalibratiedata nodig is om een goede accuraatheid te behalen, worden EEG-DSA, RSAMS 1 en RSAMS 2 vergeleken met elkaar bij een variërende grootte van de kalibratiedata. In figuur 4.26 wordt deze grootte omschreven als kalibratietrials, waarbij altijd getest werd op de 20% laatste trials van de gebruiker (40 trials). Ter vergelijking wordt in gedachten gehouden dat bij de optimalisatie van deze nieuwe gebruikers een gemiddelde accuraatheid bekomen werd van 90% (zie sectie 4.2.3). Bij het behalen van die 90% werden de BCI gekalibreerd op 80% van de data, wat overeenkomt met 160 kalibratietrials. Deze gemiddelde accuraatheid is een belangrijke referentie, aangezien het de bedoeling is om met zo weinig mogelijk kalibratiedata een zo goed mogelijke BCI te bekomen. Zowel de individuele als de gemiddelde resultaten over de verschillende algoritmen zijn terug te vinden in bijlage A.2, A.3 en A.4.



Figuur 4.26: Vergelijking tussen EEG-DSA, RSAMS 1 en RSAMS 2 bij een variëeren in het aantal kalibratie trials. Hierbij werd bij alle methoden getest op de laatste 20% van de data.

In figuur 4.26 wordt gezien dat zowel RSAMS 1 als RSAMS 2 gemiddeld beter scoren dan EEG-DSA. Er wordt ook sneller een betere accuraatheid bekomen met een kleine hoeveelheid kalibratiedata. Ook wordt opgemerkt dat beide RSAMS algoritmen vanaf ongeveer 26 kalibratie-trials een bijna even hoge test accuraatheid bekomen (88% en 87.50%) als bij een volle kalibratie van 160 trials (90%). Er kan dus besloten worden dat het RSAMS algoritme goed in staat is om met weinig kalibratiedata van de nieuwe gebruiker een goed werkende BCI te bouwen.

Hoofdstuk 5

Besluit

In deze thesis werd er onderzoek gedaan naar een transfer learning algoritme dat in staat is om met weinig kalibratiedata en een grote poele van eerdere gebruikers een accurate BCI op te bouwen voor ingebeelde beweging. Hierbij moest het algoritme ook in staat zijn om voor de nieuwe gebruiker een overeenkomstige gebruiker te selecteren uit de poele van eerdere gebruikers.

Eerst werd gestart met het uitbouwen van een BCI voor ingebeelde beweging door gebruik van LDA. Voor de spatiale filter werd een afweging gemaakt tussen SL en CSP. Na de keuze voor CSP, werd de BCI verder geoptimaliseerd op het vlak van filterparen, LDA solver, frequentiebereik van de banddoorlaatfilter en opsplitsing in het tijds- en frequentiedomein. Deze optimalisaties hadden hun nut, daar de gemiddelde test accuraatheid steeg tot 90%. Op deze manier werd een goede basis uitgebouwd om hieruit verschillende transfer learning algoritmen op te stellen.

Daar de opgestelde transfer learning algoritmen in staat moeten zijn om een overeenkomstige gebruiker te selecteren uit de poele van eerdere gebruikers, werden enkele parameters gebruikt om de gelijkenis tussen verschillende distributies aan te tonen. Door vervolgens gewichten toe te kennen per parameter, werd zo een sorteersysteem opgesteld die in staat is om een overeenkomstige gebruiker naar voor te schuiven. Hierna werden enkele transfer learning algoritmen opgesteld die gebruik maken van dit systeem.

Het eerst opgestelde transfer learning algoritme scoorde gemiddeld gezien 10% slechter dan de literatuur (EEG-DSA), ondanks het individueel soms beter scoorde. Hierdoor werd voor het

tweede algoritme een combinatie gemaakt van het eerste en EEG-DSA. Uit de resultaten van dit algoritme werd opgemerkt dat het eerste en tweede algoritme elkaar complementeren, waardoor een combinatie van beide en een methode zonder transfer learning gemaakt werd. Hieruit ontstond het Ranking-Based Subject And Methodology Selection (RSAMS) algoritme. Met dit algoritme werd een gemiddelde test accuraatheid bekomen van 87.06% met een standaardafwijking van slechts 9.02% ten opzichte van 86.12% met een standaardafwijking van 15.37% bij EEG-DSA.

Ondanks de huidige versie van het RSAMS algoritme gewenste resultaten behaalt, kan het nog verder geoptimaliseerd worden. Er kan namelijk gekeken worden of het toevoegen van andere transfer learning algoritmen een positieve invloed zou hebben op het systeem. Ook kan de keuze in parameters bij het sorteersysteem bijgesteld worden of kunnen de gebruikte gewichten geoptimaliseerd worden door deze bijvoorbeeld automatisch te trainen aan de hand van eerdere gebruikers.

Naast het optimaliseren van het RSAMS algoritme zelf, kan de achterliggende werking van de BCI ook anders opgesteld worden. Zo kunnen de gebruikte parameters eventueel parallel geoptimaliseerd worden of kan er gekeken worden wat het effect is bij een alternatieve opsplitsing in het tijds- en/of frequentiedomein.

Bijlage A

Resultaten

In deze bijlage worden enkele resultaten vermeld die in de figuren bij hoofdstuk 4 weergeven worden.

	Geen TL	**Naief**	**LFOS**	**EEG-DSA**	**Eigen 1 Ψ**
b	61.18	51.51	47.65	62.35	73.53
c	84.71	50.28	78.82	85.88	63.53
d	98.24	50.55	97.65	99.41	62.94
e	99.41	55.37	98.82	100.00	90.00
g	74.12	53.16	81.76	82.94	88.82
gemiddelde	83.53	52.18	80.94	86.12	75.76

	Eigen 2	**Eigen BestOf**	**RSAMS 1**	**RSAMS 2 Ψ**
b	64.71	73.53	73.53	73.53
c	77.65	77.65	84.71	84.71
d	98.24	98.24	98.24	98.24
e	91.18	90.00	90.00	100.00
g	68.82	68.82	88.82	82.94
gemiddelde	80.12	81.65	87.06	87.88

Tabel A.1: Vergelijking tussen alle geëvalueerde methoden

Kalibratietrials	**b**	**c**	**d**	**e**	**g**	**gemiddelde Ψ**
6	37.50	50.00	62.50	100.00	47.50	59.50
10	50.00	70.00	67.50	100.00	52.50	68.00
14	50.00	77.50	70.00	80.00	72.50	70.00
18	50.00	77.50	100.00	100.00	72.50	80.00
22	50.00	70.00	97.50	100.00	52.50	74.00
26	55.00	80.00	80.00	100.00	85.00	80.00
30	52.50	87.50	100.00	100.00	80.00	84.00
34	50.00	87.50	97.50	100.00	80.00	83.00
38	50.00	87.50	100.00	95.00	80.00	82.50
42	57.50	92.50	95.00	100.00	45.00	78.00
46	57.50	82.50	80.00	100.00	72.50	78.50
50	62.50	82.50	100.00	100.00	65.00	82.00

Tabel A.2: Test accuraatheden bij een variëeren in het aantal kalibratie trials door gebruik van **EEG-DSA**

Kalibratietrials	**b**	**c**	**d**	**e**	**g**	**gemiddelde Ψ**
6	52.50	50.00	70.00	87.50	60.00	64.00
10	45.00	50.00	87.50	87.50	57.50	65.50
14	50.00	82.50	97.50	87.50	87.50	81.00
18	50.00	87.50	100.00	87.50	87.50	82.50
22	45.00	90.00	97.50	87.50	87.50	81.50
26	57.50	95.00	100.00	100.00	87.50	88.00
30	57.50	90.00	97.50	87.50	87.50	84.00
34	52.50	87.50	97.50	100.00	87.50	85.00
38	57.50	92.50	97.50	100.00	87.50	87.00
42	57.50	95.00	97.50	100.00	87.50	87.50
46	52.50	97.50	97.50	100.00	87.50	87.00
50	50.00	92.50	97.50	100.00	87.50	85.50

Tabel A.3: Test accuraatheden bij een variëeren in het aantal kalibratie trials door gebruik van **RSAMS 1**

Kalibratietrials	**b**	**c**	**d**	**e**	**g**	**gemiddelde Ψ**
6	52.50	50.00	70.00	87.50	47.50	61.50
10	45.00	50.00	87.50	87.50	57.50	65.50
14	50.00	82.50	97.50	87.50	87.50	81.00
18	50.00	87.50	100.00	100.00	87.50	85.00
22	45.00	90.00	97.50	87.50	87.50	81.50
26	57.50	95.00	100.00	100.00	85.00	87.50
30	57.50	90.00	97.50	100.00	80.00	85.00
34	52.50	87.50	97.50	100.00	87.50	85.00
38	57.50	92.50	97.50	100.00	87.50	87.00
42	57.50	95.00	97.50	100.00	87.50	87.50
46	52.50	97.50	97.50	100.00	87.50	87.00
50	62.50	92.50	97.50	100.00	87.50	88.00

Tabel A.4: Test accuraatheden bij een variëeren in het aantal kalibratie trials door gebruik van **RSAMS 2**

Bibliografie

[1] A. Lari, F. Douma, and I. Onyiah, "Self-driving vehicles: Current status of autonomous vehicle development and minnesota policy implications," University of Minnesota, Tech. Rep., 2014.

[2] A. Saha, M. Kuzlu, and M. Pipattanasomporn, "Demonstration of a home energy management system with smart thermostat control," in *Innovative Smart Grid Technologies (ISGT)*, 2013, pp. 1–8.

[3] C.-H. Yang, S.-L. Hwang, and J.-L. Wang, "The design and evaluation of an auditory navigation system for blind and visually impaired," in *Computer Supported Cooperative Work in Design (CSCWD)*, 2014, pp. 342–345.

[4] F.-G. Zeng, S. Rebscher, W. Harrison, X. Sun, and H. Feng, "Cochlear implants: System design, integration, and evaluation," *IEEE Reviews in Biomedical Engineering*, vol. 1, pp. 115–142, 2008.

[5] F. Tenore and R. Etienne-Cummings, "Biomorphic circuits and systems: Control of robotic and prosthetic limbs," in *IEEE Biomedical Circuits and Systems Conference*, 2008, pp. 241–244.

[6] B. Rebsamen, C. Guan, H. Zhang, C. Wang, C. Teo, M. H, and E. Burdet, "A brain controlled wheelchair to navigate in familiar environments," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, pp. 590–598, 2010.

[7] K.-Y. Lee and D. Jang, "Ethical and social issues behind brain-computer interface," in *Brain-Computer Interface (BCI)*, 2013, pp. 72–75.

[8] J. Whyte, "Complete locked-in syndrome: What's in a name?" *The Journal of Head Trauma Rehabilitation*, vol. 2, pp. 144–145, 2013.

[9] U. Chaudhary and N. Birbaumer, "Communication in locked-in state after brainstem stroke: a brain-computer-interface approach," s29.

[10] P. Brunner, S. Joshi, S. Briskin, J. Wolpaw, H. Bischof, and G. Schalk, "Does the "p300"speller depend on eye gaze," *Journal of Neural Engineering*, vol. 7, p. 056013, 2010.

[11] R. S. Naveen and J. Anitha, "Brain computing interface for wheel chair control," in *Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–5.

[12] S. Hawking, *A Brief History of Time*, 1988.

[13] R. Vandenberghe, *Inleiding tot de gedragsneurowetenschappen. Deel II*, 2010.

[14] S.-W. Lee, H. H. Bülthoff, and K.-R. Müller, *Recent Progress in Brain and Cognitive Engineering*, 2015.

[15] B. Zhang, J. Wang, and T. Fuhlbrigge, "A review of the commercial bci technology from perspective of industrial robotics," in *IEEE International Conference on Automation and Logistics*, 2010, pp. 379–384.

[16] S. Vercruysse, "Adaptieve common spatial patterns voor de classificatie van ingebeelde bewegingen," Master's thesis, University of Ghent, 2011.

[17] R. Kottaimalai, M. P. Rajeskaran, V. Selvam, and B. Kannapiran, "Eeg signal classification using principal component analysis with neural network in brain computer interface applications," in *Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN)*, 2013, pp. 227–231.

[18] S. Amiri, A. Rabbi, L. Azinfar, and R. Fazel-Rezai, *A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain-Computer Interface Systems*, 2013, ch. 10, p. 20.

[19] F. Popescu, S. Fazli, Y. Badower, B. Blankertz, and K. Müller, "Single trial classification of motor imagination using 6 dry eeg electrodes," *PLoS ONE*, vol. 2, p. e637, 2007.

[20] S. Amiri, A. Rabbi, L. Azinfar, and R. Fazel-Rezai, "A review of p300, ssvep, and hybrid p300/ssvep brain-computer interface systems," *Advances in Human-Computer Interaction*, vol. 2013, p. 8, 2013.

[21] H. Hallez, A. Vergult, R. Phlypo, P. V. Hese, W. D. Clercq, Y. D'Asseler, R. V. de Walle, B. Vanrumste, W. V. Paesschen, S. V. Huffel, and I. Lemahieu, "Muscle and eye movement artifact removal prior to eeg source localization," in *Conf Proc IEEE Engineering in Medicine and Biology Society*, 2006, pp. 1002–1005.

[22] L. M. Alonso-Valerdi, R. A. Salido-Ruiz, and R. A. Ramirez-Mendoza, "Motor imagery based brain-computer interfaces: An emerging technology to rehabilitate motor deficits," *Neurophychologia*, vol. 79, pp. 354–363, 2015.

[23] S. Lemm, K.-R. Müller, and G. Curio. (2009) A generalized framework for event-related desynchronization (erd). Accessed: 2016-05-20. [Online]. Available: http://www.bbci.de/supplementary/conditionalERD/conditionalERD.htm

[24] M. Kawanabe, C. Vidaurre, S. Scholler, and K.-R. Müller, "Robust common spatial filters with a maxmin approach," *Neural Computation*, vol. 26, pp. 1–28, 2014.

[25] G. Pfurtscheller, C. Brunner, A. Schlögl, and F. L. da Silva, "Mu rhythm (de)synchronization and eeg single-trial classification of different motor imagery tasks," *NeuroImage*, vol. 31, pp. 153–159, 2006.

[26] H.-G. Yeom and K.-B. Sim, "Ers and erd analysis during the imaginary movement of arms," in *Control, Automation and Systems (ICCAS)*, 2008, pp. 2476–2480.

[27] L. Citi, R. Poli, C. Cinel, and F. Sepulveda, "P300-based bci mouse with genetically-optimized analogue control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, pp. 51–61, 2008.

[28] C. Guger, S. Daban, E. Sellers, C. Holzener, G. Krauze, R. Carabalona, F. Gramatica, and G. Edlinger, "How many people are able to control a p300-based brain-computer interface(bci)?" *Neuroscience Letters*, vol. 462, pp. 94–98, 2009.

[29] V. Kolev, T. Demiralp, J. Yordanova, A. Ademoglu, and U. Isoglu-Alka, "Time-frequency analysis reveals multiple functional components during oddball p300," *Neuroreport*, vol. 8, pp. 2061–2065, 1997.

[30] T. Verhoeven, P. Buteneers, J. Wiersema, J. Dambre, and P. Kindermans, "Towards a symbiotic braincomputer interface: exploring the applicationdecoder interaction," *Journal of Neural Engineering*, vol. 12, p. 066027, 2015.

[31] A. Y. Kaplan, A. A. Fingelkurts, S. V. Borisov, and B. S. Darkhovsky, "Nonstationary nature of the brain activity as reavealed by eeg meg methodological, practical and conceptual challenges," *Signal Processing*, vol. 85, pp. 2190–2212, 2005.

[32] S. Dalhoumi, G. Dray, and J. Montmain, "Knowledge transfer for reducing calibration time in brain-computer interfacing," in *IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014, pp. 634–639.

[33] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust eeg single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, pp. 41–56, 2008.

[34] V. Jayaram, M. Alamgir, Y. Altun, B. Schölkoph, and M. Grosse-Wentrup, "Transfer learning in brain-computer interfaces," 2015.

[35] Y. Renard, F. Lotte, G. Gilbert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. Lécuyer, "Openvibe: An open-source software platform to design, test and use brain-computer interfaces in real and virtual enviroments," *MIT Press*, vol. 19, pp. 35–53, 2010.

[36] P. S. Foundation, *Python Language Reference, version 2.7*. [Online]. Available: http://www.python.org

[37] S. C. C. Stéfan van der Walt and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engineering*, vol. 13, pp. 22–30, 2011.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[39] N. Brodu, F. Lotte, and A. Lécuyer, "Exploring two novel features for eeg-based brain-computer interfaces: Multifractal cumulants and predictive complexity," *Neurocomputing*, vol. 79, pp. 87–94, 2012.

[40] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, "The non-invasive berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, pp. 539–550, 2007.

[41] A. Goldberger, L. Ameral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, C.-K. Peng, and H. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, 2000.

[42] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "Bci2000: A general-purpose brain-computer interface (bci) system," *IEEE Transactions on Bio-Medical Engineering*, vol. 51, pp. 1034–1043, 2004.

[43] F. Piccione, F. Giorgi, P. Tonin, K. Prifits, S. Giove, S. Silvoni, G. Palmas, and F. Beverina, "P300-based brain computer interface: Reliability and performance in healthy and paralysed participants," *Clinical Neurophysiology*, vol. 117, pp. 531–537, 2006.

[44] T. Mladenov, K. Kim, and S. Nooshabadi, "Accurate motor imagery based dry electrode brain-computer interface system for consumer applications," in *Consumer Electronics (ISCE)*, 2012, pp. 1–4.

[45] J. Malmivuo and R. Plonsey, *Bioelectromagnetism - Principles and Applications of Bioelectric and Biomagnetic Fields*, 1995.

[46] P. Cheh, "Gibbs phenomenon removal and digital filtering directly through the fast fourier transform," *IEEE Transactions on Signal Processing*, vol. 49, pp. 444–448, 2001.

[47] A. D. Tirkey and N. K. Verma, "Minimizing intra class variations in multi-class common spatial patterns for motor imagery eeg signals," in *9th International Conference on Industrial and Information Systems (ICIIS)*, 2014, pp. 1–5.

[48] M. Z. Ilyas, P. Saad, and M. I. Ahmad, "A survey of analysis and classification of eeg signals for brain-computer interfaces," in *Biomedical Engineering (ICoBE)*, 2015, pp. 1–6.

[49] D. McFarland, L. McCane, S. David, and J. Wolpaw, "Spatial filter selection for eeg-based communication," *Electroencephalography and clinical Neurophysiology*, vol. 103, pp. 386–394, 1997.

[50] C. Sannelli, C. Vidaurre, K.-R. Müller, and B. Blankertz, "Common spatial pattern patches - an optimized filter ensemble for adaptive brain-computer interfaces," in *Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010, pp. 4351–4354.

[51] M. Barsotti. (2014) Workshop: "brain computer interfaces and haptics". Accessed: 2016-05-20. [Online]. Available: http://www.slideshare.net/MicheleBarsotti/hs2014-bci-mi

[52] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller, "Boosting bit rates in non-invasive eeg single-trial classifications by feature combination and multi-class paradigms," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 993–1002, 2004.

[53] A. Ng, "Machine learning - introduction," University Lecture, 2015.

[54] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881–892, 2002.

[55] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, p. 24, 2007.

[56] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 252–264, 1991.

[57] S. Fortmann-Roe. (2012) Accurately measuring model prediction error. Accessed: 2016-05-20. [Online]. Available: http://scott.fortmann-roe.com/docs/MeasuringError.html

[58] V. Lavrenko, "Pca 11: Linear discriminant analysis," University Lecture, 2013.

[59] O. Ledoit and M. Wolf, "Honey, i shrunk the sample covariance matrix," *Journal of Portfolio Management*, vol. 30, pp. 110–119, 2004.

[60] N. Japkowicz, "Why question machine learning evaluation methods?" 2006.

[61] F. Lotte and C. Guan, "Learning from other subjects helps reducing brain-computer interface calibration time," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 614–617.

[62] P. Pudil, F. J. Ferri, and J. Kittler, "Floating search methods for feature selection with nonmonotonic criterion functions," *Pattern Recognition*, vol. 2, pp. 279–283, 1994.

[63] L. I. Kuncheva, "A stability index for feature selection," in *25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, 2007, pp. 390–395.

[64] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011, pp. 430–442.

[65] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[66] N. Proesmans, "Brain-computer interfaces using machine learning: Reducing calibration time in motor imagery," Master's thesis, University of Ghent, 2016, unpublished.

[67] M. Arvaneh, I. Robertson, and T. E. Ward, "Subject-to-subject adaptation to reduce calibration time in motor imagery-based brain-computer interface," in *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 6501–6504.

[68] X. Steenbrughe, "Multiclass detection of eeg signals with filter bank common spatial patterns using linear discriminant analysis and subject specific time frames," Master's thesis, University of Ghent, 2015.