



EXAMENSARBETE INOM TEKNIK,
GRUNDNIVÅ, 15 HP
STOCKHOLM, SVERIGE 2016

Genomförbarhetsstudie av att känna igen två tankemönster i följd med EEG

Feasibility study of recognizing two subsequent
thought patterns with EEG

OSKAR WILHELMSSON

VICTOR WIKÉN

Genomförbarhetsstudie av att känna igen två
tankemönster i följd med EEG

Feasibility study of recognizing two
subsequent thought patterns with EEG

DM129X Examensarbete inom Medieteknik, grundnivå
Vid CSC skolan, KTH
Vårterminen 2016

Wilhelmsson, Oskar
oskarwi@kth.se

Wikén, Victor
vwiken@kth.se

Handledare: Walldius, Åke
Examinator: Pargman, Daniel

Genomförbarhetsstudie av att känna igen två tankemönster i följd med EEG

Sammanfattning

Studien implementerade ett hjärna-dator-gränssnitt med hjälp av EEG-instrumentet MindWave Mobile Headset. Vi undersökte om det var möjligt att utföra fyra operationer genom att använda tankemönster. Fyra försökspersoner deltog i studien. Deras uppgift var att tänka i två tankemönster i följd som resulterade i en operation. EEG-signalen förbehandlas så att en mönsterigenkänningsmetod (k-NN) lättare kunde urskilja två tankemönster ur signalen. Denna undersökning har till vår vetskap inte tidigare utförts och är därmed kunskapsluckan vi ämnar fylla. Att fylla denna kunskapslucka är av intresse för bland annat användargrupperna: rörelsehindrade, spelintresserade och Virtual Reality-användare. Vi tog fram en modell som modellerade det bästa möjliga utfallet av metodiken i föreliggande studie. Undersökningens resultat kunde inte användas för att göra slutsatser angående frågeställningen då detta skulle vara att post hoc-teoretisera. I modellen visades dock tre av fyra operationer vara genomförbara, med en indikation om att även den fjärde var möjlig att utföra. Resultatet indikerar att det finns anledning att utföra en fortsatt studie. Den föreslagna fortsatta studien bör innefatta nya mätningar som testas av modellen för att fullt ut besvara problemformuleringen.

Nyckelbegrepp: EEG, MindWave, BCI, egenskapsvektor, förbehandling, k-NN, dimensionalitetsreducering, klassificeringsalgoritm.

Feasibility study of recognizing two subsequent thought patterns with EEG

Abstract

This study implements a Brain-Computer-Interface using the EEG-instrument MindWave Mobile Headset. We studied the feasibility of performing four operations using thought patterns. Four test subjects participated in the study. Their task was to think in two subsequent thought patterns that resulted in an operation. The EEG-signal was pre-processed in such a way that a pattern recognition algorithm (k-NN) more easily could recognize two thought patterns in the signal. This study has to our knowledge not been done before and thus aims to fill this lack of knowledge in the scientific community. User groups that have an interest in filling this gap are, amongst others; disabled people, gamers, and Virtual Reality users. We created a model that modeled the best possible outcome of the method used in this study. Conclusions drawn from the result can not be used to fully answer the problem statement, since it would be to post hoc-theorize. However, three out of four operations were possible to perform in the model, with an indication that the fourth also was possible to perform. These results indicate that there are grounds to continue this study. The proposed continued study should include new measurements that are tested by the model to determine if it is feasible to distinguish all four operations.

Keywords: EEG, MindWave, BCI, feature vector, pre-processing, k-NN, dimensionality reduction, classification algorithm.

Innehållsförteckning	4
1. Inledning	5
1.1 Bakgrund	5
1.2 Syfte	5
1.3 Problemformulering	6
1.3.1 Avgränsningar	6
2. Teori	7
2.1 EEG-instrument	7
2.2 k -Närmsta Grannar (k -NN)	7
2.3 Dimensionalitet	8
2.4 Förbehandling	8
3. Metod	10
3.1 Tankemönster och operationer	10
3.2 Experimentellt upplägg	11
3.3 Mätningssmetod	11
3.3.1 Kalibreringsmätningar	11
3.3.2 Testmätningar	11
3.4 Databehandling	12
3.4.1 Behandling av kalibreringsmätningar	12
3.4.2 Behandling av testmätningar	14
3.4.3 k -NN	15
3.4.4 Matlab	18
3.4.5 Konfidensintervall	18
4. Empiri	19
5. Diskussion	23
5.1 Metodkritik	23
5.1.1 Experimentellt upplägg	23
5.1.2 Försökspersoner	24
5.1.3 Optimeringskalibrering	24
5.2 Resultat	24
5.3 Möjliga användningsområden	25
6. Slutsats	27
6.1 Nästa steg	27
7. Referenslista	28
8. Bilagor	30
8.1 Resultat	30
8.2 Slumpträkning	31
8.3 Matlabkod	31
8.3.1 Kalibreringsmätningar	31
8.3.2 Testmätningar	32
8.3.3 Klassificering	32

1. Inledning

Genomförbarhetsstudie av att känna igen två tankemönster i följd med EEG.

1.1 Bakgrund

Ett hjärna-datorgränssnitt, så kallat Brain-Computer Interface (BCI), definieras som ett system som innefattar hårdvara och mjukvara för att skapa en kommunikation mellan hjärna och dator.

BCI är ett ytterst aktuellt ämne idag. Antalet artiklar som publicerats inom BCI det senaste decenniet har ökat exponentiellt. BCI används idag främst inom forskning för att styra proteser. Ytterligare användningsområden inom BCI inkluderar bokstaveringsprogram, styrning av rullstol, och spel. (Nicolas-Alonso & Gomez-Gil 2012)

Elektroencefalografi (EEG) är en BCI-metod för att få information om en hjärnas aktiviteter. Denna metod utförs med hjälp av ett EEG-instrument. Den *minsta konfigurationen* för ett EEG-instrument är en aktiv elektrod, en referenselektrod och en jordningselektrod (Teplan 2002). Joniska strömmar genereras i hjärnan när en människa tänker. EEG-instrument kan användas för att mäta denna hjärnaktivitet med dess aktiva elektrod. Genom att avläsa dessa mätningar har tidigare forskning visat att det är möjligt att urskilja olika mönster i hjärnaktivitet (Lotte et al. 2007). Ett *tankemönster* är ett mönster av hjärnaktivitet som uppstår vid en eller flera tankar (Lotte et al. 2007). Genom att tänka så att ett visst tankemönster genereras kan användaren utföra en operation på en dator. (Nicolas-Alonso & Gomez-Gil 2012)

I föreliggande studie används EEG-instrumentet MindWave. MindWave kostar cirka \$100 dollar, vilket är billigt i jämförelse med instrument som används inom EEG-forskning (Neurosky.com 2016; Crowley et al. 2010). MindWave har den minsta möjliga konfigurationen för EEG-instrument, vilket beskrivs mer utförligt i avsnitt 2.1.

Studier har visat potential i EEG, till exempel har en studie gjorts som visar att det är möjligt att utföra flera operationer för att styra ett objekt i tre dimensioner på en datorskärm (Wolpaw 2010). I studier där kontroll av flera operationer uppnåtts används ofta ett EEG-instrument med fler än en aktiv elektrod (McFarland et al. 1997; Wolpaw & McFarland 1994). Dessa elektroder kan vara otympliga att placera ut (Lin et al. 2011). I kontrast är EEG-instrument med en aktiv elektrod mer lätthanterliga, men inte lika bra på att urskilja tankemönster (Kayikcioglu & Aydemir 2010; Dunn et al. 1999). Därmed är MindWave ett förhållandevis praktiskt EEG-instrument men som inte är lika bra på att urskilja tankemönster jämfört med EEG-instrument som vanligtvis används inom forskning.

1.2 Syfte

Genom att använda ett billigare och mindre otympligt EEG-instrument som MindWave anser vi att det är möjligt att nå en användargrupp som söker ett mer praktiskt användande av EEG-instrument. Denna användargrupp kan inkludera bland annat rörelsehindrade, spelintresserade och Virtual Reality-användare.

Vi upplever att de EEG-implementationerna som finns för instrument med endast en aktiv elektrod, som MindWave, är begränsade i funktionalitet. Till vår vetskap kan dessa instrument endast utföra två operationer. Genom att öka antalet operationer kan EEG-instrument bli mer användbara. Vi ämnar undersöka om det är genomförbart att utföra fler än två operationer genom att använda två tankemönster i kombination med varandra för att utöka antalet möjliga operationer.

1.3 Problemformulering

Är det genomförbart att utföra fyra olika operationer med EEG-instrumentet MindWave Mobile Headset?

1.3.1 Avgränsningar

Vi begränsade antalet mätningar som utfördes på grund av tidsbegränsning, antalet mätningar beskrivs i avsnitt 3.3. Vi avgränsade oss till fyra försökspersoner och fyra operationer av samma anledning.

2. Teori

Här presenteras de teorier som ger förståelse för problemen och möjligheterna med EEG. Teorierna har använts för att utforma den metod vi använt i föreliggande studie.

2.1 EEG-instrument

I föreliggande studie används EEG-instrumentet MindWave som har den minsta möjliga konfigurationen för EEG-instrument, vilket nämns i avsnitt 1.1. MindWave har en aktiv elektrod, en referenselektrod och en jordningselektrod. Den aktiva elektroden placeras i centrum av pannan. Denna elektrod mäter antalet volt av joniska strömmar mellan hjärnans neuroner. Instrumentet har även en referenselektrod som placeras på den vänstra öronsribben för att reducera störningar. Störningar kan orsakas av till exempel muskelspänningar och reducerar mängden värdefull information som kan avläsas från mätningen. (Nicolas-Alonso & Gomez-Gil 2012)

Placeringen av EEG elektroderna görs vanligtvis i studier efter 10-20 systemet, som anses vara en internationell standard. I detta system placeras 19 elektroder ut jämnt över skalpen i ett förutbestämt mönster, där avstånden mellan elektroderna baseras på huvudets storlek. Dessutom placeras två elektroder i öronsribbarna. Liknande system finns för 256 elektroder (Teplan 2002). (Jasper 1958)

För att kunna utföra operationer på en dator med ett EEG-instrument behöver datorn urskilja tankemönster ur EEG-signalen. Detta görs vanligtvis inom BCI med en klassificeringsalgoritm. (Lotte et al. 2007)

2.2 k-Närmsta Grannar (k-NN)

En klassificeringsalgoritm jämför ny data med tidigare kategoriserad data. Genom denna jämförelse kan algoritmen avgöra vilken kategori den nya datan tillhör, detta kallas klassificering. Olika klassificeringsalgoritmer kan användas för att jämföra data. (Michie et al. 1994)

I föreliggande studie begränsar vi oss till algoritmer som är relativt enkla att implementera på grund av tidsbegränsning. En klassificeringsalgoritm som är enkel att implementera och har gett bra resultat i tidigare studier är *k-närmaste grannar* (k-NN) (Lotte et al. 2007; Kayikcioglu & Aydemir 2010). Algoritmen k-NN får kalibreringsmätningar vars kategori är känd. Dessa mätningar placeras ut i ett koordinatsystem där varje mätning motsvarar en punkt. Gruppen av de mätningar som tillhör en viss kategori kallas en klass. När klasser har byggts upp kan k-NN klassificera EEG-data som inte har någon klass. För att klassificera ny data som inte har någon klass placeras den nya datan ut i koordinatsystemet. Genom avståndsberäkningar tar k-NN reda på vilka punkter som ligger närmast den nya datan. De k-antal punkterna som ligger närmast kallas de k-närmaste grannarna. k-NN undersöker vilken klass de k-närmaste grannarna tillhör. Den mest förekommande klassen bland de k-närmaste grannarna blir den okända datans klass. Därmed blir den okända datan klassificerad. (Weinberger & Saul 2009; Lotte et al. 2007; Chaovalitwongse et al. 2007)

Inom algoritmen k-NN kan olika metoder användas för att beräkna avståndet till de k-närmsta grannarna. Val av metod för avståndsberäkning kan påverka resultatet. (Weinberger & Saul 2009) En annan faktor som påverkar resultatet av k-NN är antalet grannar som undersöks, det vill säga värdet på k för k-NN. Värdet för k kan

varieras för att optimera resultatet av k-NN. (Chaovalitwongse et al. 2007; Song et al. 2007)

En vanligt förekommande metod för att beräkna avståndet till de k-antal närmaste grannarna är Euclidean Distance (ED). En mer avancerad metod för att beräkna avståndet är Mahalanobis Distance (MD). ED beräknar det geometriska avståndet mellan två punkter. MD använder sig av samma principer som ED men MD tar hänsyn till korrelationen i datan. Om datapunkterna är helt okorrelerade så är MD och ED ekvivalenta metoder. (De Maesschalck et al. 2000; Weinberger & Saul 2009; Xiang et al. 2008)

2.3 Dimensionalitet

Som beskrivs i avsnitt 2.2 placeras mätningar ut i ett koordinatsystem där varje mätning motsvarar en punkt. Dimensionen för koordinatsystemet avgörs av antalet värden i en mätning. Alla mätningar kommer ha samma antal värden. Ett förhållande som är viktigt för de flesta klassificeringsalgoritmer inom BCI, och speciellt för k-NN, är låg *dimensionalitet* (Lotte et al. 2007). Dimensionalitet är antalet dimensioner som behövs för att representera koordinatsystemet. Dimensionaliteten är därmed antalet värden en mätning innehåller. (Schlögl et al. 2005; Weinberger & Saul 2009). Mätningen som erhålls från en EEG-elektrod är en vektor med sampelvärden. Inom BCI-forskning kallas dessa sampelvärden för egenskapsvärden. En vektor som består av egenskapsvärden kallas för egenskapsvektor. (Lotte et al. 2007).

Dimensionaliteten av egenskapsvektor är antalet egenskapsvärden i vektorn (Schlögl et al. 2005). Dimensionaliteten av en egenskapsvektor kan reduceras genom att tillämpa metoder för att välja ut viktig, eller ta bort onödig, information. Att reducera dimensionaliteten av en egenskapsvektor kallas för *dimensionalitätsreducering* (Nicolas-Alonso & Gomez-Gil 2012). En anledning att dimensionalitätsreducering ofta utförs inom BCI är att hög dimensionalitet av egenskapsvektorer ger upphov till ett välkänt problem inom BCI, curse-of-dimensionality. Curse-of-dimensionality innebär att mängden data som krävs för att kunna beskriva olika klasser ökar exponentiellt desto högre dimensionalitet egenskapsvektorn har. Sämre resultat kan därmed erhållas om datamängden är liten. En studie rekommenderar att ta fem till tio gånger fler mätningar än antalet egenskapsvärden i en egenskapsvektor. (Lotte et al. 2007)

Inom EEG hamnar olika tankemönster inom vissa frekvensband (Nicolas-Alonso & Gomez-Gil 2012). Denna egenskap kan användas för att reducera dimensionaliteten av egenskapsvektorer. Detta kan tillämpas genom att filtrera signalen. (Kayikcioglu & Aydemir 2010)

2.4 Förbehandling

Discrete Fourier Transform (DFT) är en metod att transformera en signal från tidsdomän till frekvensdomän. Inom EEG-forskning kan data förbehandlas i frekvensdomänen genom att ta medelvärden av olika delar av frekvensbanden. Denna förbehandlingsmetod kan användas för att reducera dimensionaliteten av en mätning. (Yazdani et al. 2009; Soininen et al. 1989; Akin 2002)

Olika frekvensband avläses olika bra beroende på positionering av elektroder. Framsidan av huvudet ger särskilt användbara utslag för frekvensbanden alfa och beta, som vi kallar alfabandet och betabandet. Eftersom den aktiva elektroden i instrumentet MindWave placeras i pannan är dessa frekvenser lämpliga, vilket beskrivs i avsnitt 2.1. Alfabandet uppträder inom 8-12 Hz. Alfabandet är associerat

med hög eller låg mental ansträngning och kan därmed vara användbart för att mäta graden av mental ansträngning. Betabandet uppträder inom 12-32 Hz där den lägre delen, 13-21 Hz, av betabandet främst är associerat till ett fokuserat och tänkande tillstånd. Det högre frekvensbandet, 20-32 Hz, är associerat med ett intensivt eller oroligt tillstånd. Inom alfa- och betabandet kan även föreställda rörelser ge utslag. En föreställd rörelse innebär att en person föreställer sig att utföra en rörelse, utan att faktiskt utföra rörelsen. Sammanfattningsvis är de olika frekvensbanden inom frekvenserna 8-32 Hz associerade med graden av mental ansträngning och föreställda rörelser. (Nicolas-Alonso & Gomez-Gil 2012; Huang & Charyton 2008; Dunn et al. 1999).

3. Metod

Målet med föreliggande undersökningen är att undersöka om det är genomförbart att klassificera fyra operationer med EEG-instrumentet MindWave. Vi beskriver först vilka tankemönster som används och hur vi definierar en operation. Därefter beskrivs hur studien utfördes i form av experimentellt upplägg, mätningar och databehandling.

3.1 Tankemönster och operationer

Olika tankemönster kan avläsas med olika noggrannhet av EEG-instrument, vilket medför att klassificeringsalgoritmens resultat varierar beroende på vilket tankemönster som används (Nicolas-Alonso & Gomez-Gil 2012).

Alfa- och betabandet har visats vara associerade med tankemönstren fokus, avkoppling och föreställda rörelser, vilket beskrivs i avsnitt 2.4. Det finns redan applikationer för MindWave som använder tankemönstren fokus och avkoppling vilket enligt oss indikerar att dessa två tankemönster går att känna igen med hjälp av MindWave (Neurosky.com 2016).

De flesta människor har svårigheter att utföra föreställda rörelser på ett sätt som ger tillräckligt utslag för ett BCI. Istället för att föreställa sig själva rörelsen så bygger de istället upp en visuell bild av rörelsen, vilket inte ger ett tillräckligt utslag för ett BCI. (Nicolas-Alonso & Gomez-Gil 2012) I denna studie valde vi att använda tankemönstren fokus och avkoppling. Grunden till detta val var indikationen att fokus och avkoppling går att använda med MindWave, samt att det finns svårigheter i att använda föreställda rörelser.

För att utföra tankemönstret för avkoppling ombads försökspersonen koppla av genom att fokusera på sin andning för att ha så få tankar som möjligt i hjärnan. För tankemönstret fokus ombads försökspersonen subtrahera ett tal kontinuerligt från talet 101 ner till 0. Vi varierade mellan talen 3 och 7 vid subtrahering så att försökspersonen inte vände sig vid en talföljd. Det vi hade för avsikt att mäta var därmed tankemönstret för uträkningsoperationer i hjärnan, inte tankemönstret för att komma ihåg en talföljd. Liknande mentala övningar har även använts i andra studier för att orsaka tankemönster (Dunn et al. 1999; Penny et al. 2000). Tankemönster för avkoppling kallar vi tankemönster 0 och tankemönstret för fokus kallar vi tankemönster 1.

I föreliggande studie används klassificeringsalgoritmen k-NN för att urskilja de två tankemönstren ur en EEG-signal. För varje tankemönster används en klass, därmed används två klasser. k-NN behöver klasserna för att kunna känna igen dessa tankemönster, vilket beskrivs i avsnitt 2.2. Om tankemönster 1 klassificeras korrekt så klassificeras det till klass 1. Om tankemönster 0 klassificeras korrekt så klassificeras det till klass 0.

Som nämns i avsnitt 1.1 kan en användare utföra en operation på en dator genom att tänka så att ett visst tankemönster genereras. Det som undersöks i föreliggande studie är genomförbarheten av att utföra fyra operationer med hjälp av två tankemönster, vilket beskrivs i 1.2. Tankemönstren kan kombineras enligt binär princip på följande vis: 00, 01, 10 och 11. Tankemönster 1 motsvarar 1 och tankemönster 0 motsvarar 0 i denna studie. Därav används följande beteckningar för operationerna i denna studie: operation 00, operation 01, operation 10 och operation 11.

3.2 Experimentellt upplägg

Fyra vuxna försökspersoner deltog i föreliggande undersökning, varav vi två författare är inkluderade. De resterande försökspersonerna valdes från författarnas umgängeskrets baserat på intresse då mätningarna uppskattades ta en timme, vilket kan anses vara omfattande. Undersökningen utfördes i ett rum där vi minimerat störningar i form av stimuli från ljud och ljus. En maskin fanns i närheten som försökspersoner hörde och reagerade på vid enstaka tillfällen. Om försökspersonen reagerade på detta så exkluderades mätningen från studien.

EEG-instrumentet placerades först på försökspersonens huvud. Därefter kontrollerades att instrumentet satt ordentligt och inte gled. Försökspersonen ombads att koppla av i en sittande position samt att undvika muskelspänningar i kroppen.

Försökspersonen ombads att informera studieansvarig om:

- Eventuella muskelryckningar förekommit.
- Försökspersonen tappade tankemönstret.
- Försökspersonen upplevt stress.

I dessa fall exkluderades mätningen från studien och proceduren upprepades.

Därefter startade studieansvarig den mobila applikationen för MindWave vid namn Recorder. I denna applikation presenteras eventuella störningar i signalen samt två indikatorer: en för fokus och en för meditation. Dessa indikatorer användes inte under studien. Anledningen till att de inte användes var att vi inte lyckades ta reda på hur värdena räknades ut. Efter mätningen påbörjats kontrollerade studieansvarig kontinuerligt om störningar förekom under mätningen, vilket uppstod till exempel om instrumentet gled.

Studien har två mätningfaser vars första fas, kalibreringsmätningar, används för att bygga upp två k-NN klasser. Den andra fasen samlar in testmätningar där försökspersonen ämnar utföra de olika operationerna. En testmätning motsvarar därmed en operation. Samtliga mätningar pågick i minst fem sekunder.

3.3 Mätningsmetod

3.3.1 Kalibreringsmätningar

Följande procedur användes för kalibreringsmätningarna:

- Försökspersonen informerades om vilket tankemönster som skulle utföras.
- Försökspersonen påbörjade utföra tankemönstret.
- Försökspersonen indikerade beredskap med en handrörelse. Därefter påbörjades mätningen efter minst tre sekunder beroende på eventuella störningsindikationer.
- Försökspersonen meddelades verbalt efter fem sekunder av mätningen förlöpt.

Respektive tankemönster mättes åtta gånger, vilket innebär totalt 16 stycken kalibreringsmätningar. För att bygga upp de två k-NN klasserna för varje försöksperson används därmed 16 mätningar.

3.3.2 Testmätningar

Vi simulerade en operation genom att göra en fem sekunder lång mätning där endast den första och sista sekunden av mätningen användes. Detta innebär att

försökspersonen hade tre sekunder på sig att byta tankemönster. Dessa två ensekundssegment klassificerades av k-NN till ett av de två tankemönstren. En korrekt utförd operation inträffar därmed om båda tankemönster stämmer överens med användarens avsikt.

Följande procedur följdes i mätning av de efterföljande tankemönstren:

- Försökspersonen informerades om vilka tankemönster, och i vilken ordning, som skulle utföras.
- Försökspersonen påbörjade utföra det första av de två tankemönstren.
- Försökspersonen indikerade beredskap med en handrörelse, varpå mätningen påbörjades efter minst tre sekunder beroende på eventuella störningsindikationer.
- Försökspersonen meddelades verbalt när en sekund av mätningen förlöpt.
- Försökspersonen meddelades verbalt efter fem sekunder av mätningen förlöpt.

När både den första och sista sekunden av mätningen motsvarade samma tankemönster så meddelades försökspersonen *inte* när övergångsperioden mellan tankemönstren påbörjades. De fyra operationerna mättes tre gånger var per försöksperson. Testmätningarna omfattade totalt 12 mätningar per försöksperson.

I applikationen gavs en indikation på hur mycket tid som förlöpt genom dess uppdateringsfrekvens. Denna indikation använde studieansvarig för att avgöra när en sekund förlöpt.

3.4 Databehandling

EEG-datan från instrumentet sparas ner som en textfil vilket är kopplat till en Cloud-tjänst. Filerna namnges av studieansvarig efter tankemönster, försöksperson, och proceduriteration för att därefter laddas in i Matlab. Nedan följer databehandlingsprocessen för en försöksperson.

3.4.1 Behandling av kalibreringsmätningar

Samtliga kalibreringsmätningar kapades först till fem sekunder så alla mätningar hade samma antal värden. Därefter delades kalibreringsmätningen upp till fem vektorer som motsvarar en sekunds mätning. Dessa fem vektorer transformerades med DFT från tidsdomän till frekvensdomän. I frekvensdomänen filterades vektorn så att endast frekvenserna 8-32 Hz var kvar. Värdena inom frekvenserna 8-32 Hz delades upp till tre frekvensband. Medelvärde för varje frekvensband beräknades vilket resulterade i tre egenskapsvärden. Denna process reducerade dimensionen av egenskapsvektorn. Behandlingen av en kalibreringsmätning resulterade i fem egenskapsvektorer. Eftersom det var totalt 8 kalibreringsmätningar per tankemönster så resulterade det i 40 egenskapsvektorer för respektive tankemönster. Dessa användes för att bygga upp klasserna. Behandling av kalibreringsmätningarna förklaras nedan, Figur 1 och Figur 2.

Kalibreringar

Från tidsdomän till frekvensdomän

1. En kalibreringsmätning



Sampelvärden för fem sekunder



2. Segmentering till fem ensekundersintervall



Ett segment innehåller sampelvärden för en sekund.



3. Omvandling till frekvensdomänen (DFT)



*Frekvensvärden 1-256 Hz
(Steg 3 upprepas för samtliga segment)*



4. Dimensionalitätsreducering (Figur 2)

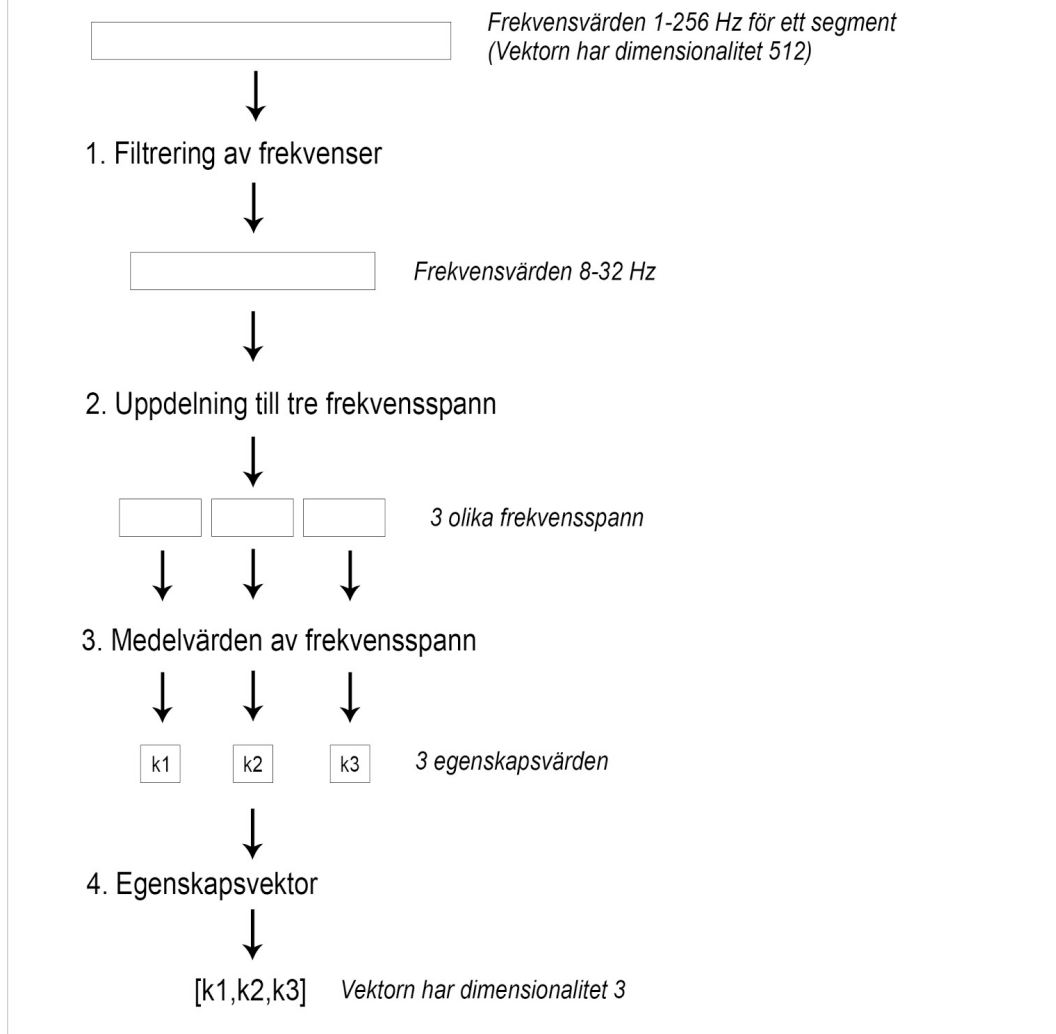


$$\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \\ \vdots & \vdots & \vdots \\ k_{5,1} & k_{5,2} & k_{5,3} \end{bmatrix}$$

Dimensionalitätsreducering upprepas för varje ensekunderintervall vilket leder till totalt 5 egenskapsvektorer. Dessa vektorer placeras sedan i en matris.

Figur 1 - Kalibrering

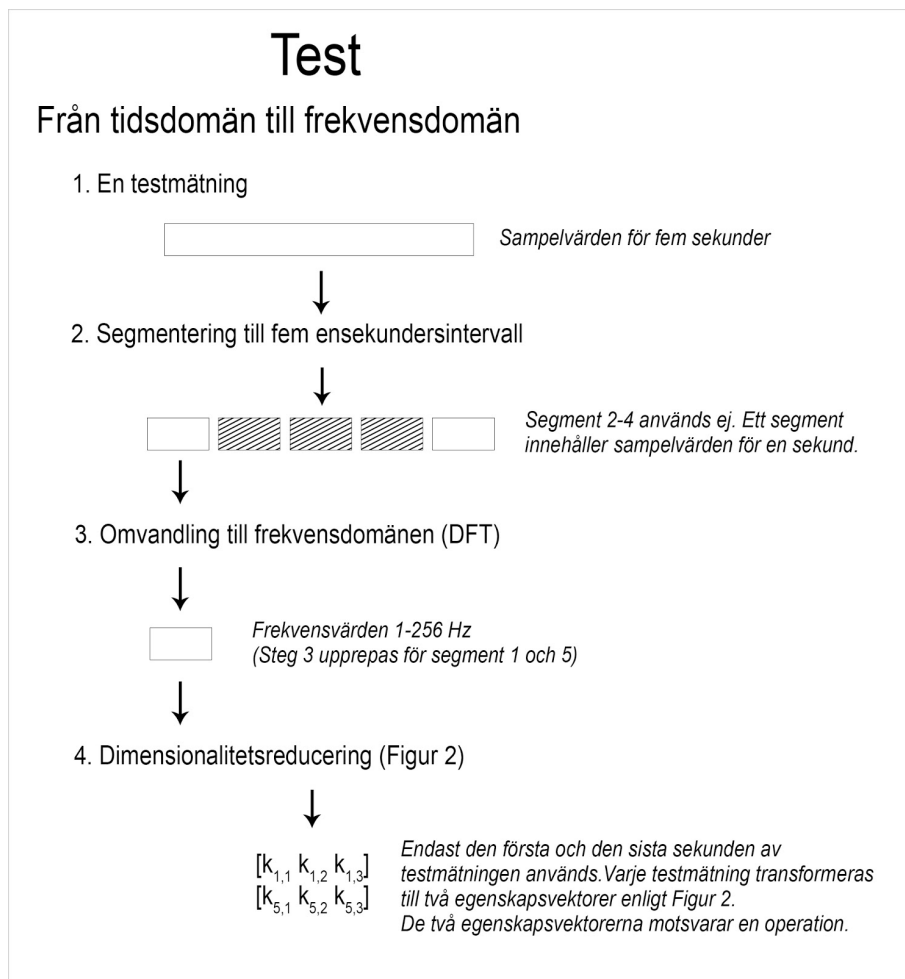
Dimensionalitätsreducering



Figur 2 - Dimensionalitätsreducering

3.4.2 Behandling av testmätningar

Testmätningarna behandlades enligt samma principer som kalibreringsmätningarna. Skillnaden i behandling var att endast den första och sista sekunden av femsekundersmätningen användes. Sekunderna 2-4 var en övergångsperiod för att försökspersonen skulle kunna byta tankemönster. Efter behandling fick vi ut två egenskapsvektorer som då var redo att klassificeras av k-NN. Processen förklaras ytterligare i Figur 3.



Figur 3 - Test

3.4.3 k-NN

Algoritmen k-NN fick som input 80 stycken egenskapsvektorer som byggde upp de två klasserna. Därefter behandlades varje testmätning och kontrollerade om dess två egenskapsvektorer klassificerades korrekt av k-NN. I föreliggande studie användes Mahalanobis distance (MD) för att avgöra avståndet till de k-närmaste grannarna, se avsnitt 2.3. Om båda egenskapsvektorerna klassificeras korrekt så lyckades försökspersonen utföra en operation. Processen från mätning till klassificering förklaras i Figur 4-6.

Från mätningar till klassificering

1. Mätningar

Klass 1	Klass 0
Mätning 1	Mätning 9
Mätning 2	Mätning 10
⋮	⋮
Mätning 8	Mätning 16

Åtta mätningar har gjorts per klass.

2. Klasser

Varje kalibreringsmätning transformeras till 5 egenskapsvektorer enligt Figur 1. Mätningarna från respektive klass placeras i var sin 40 x 3 matris.

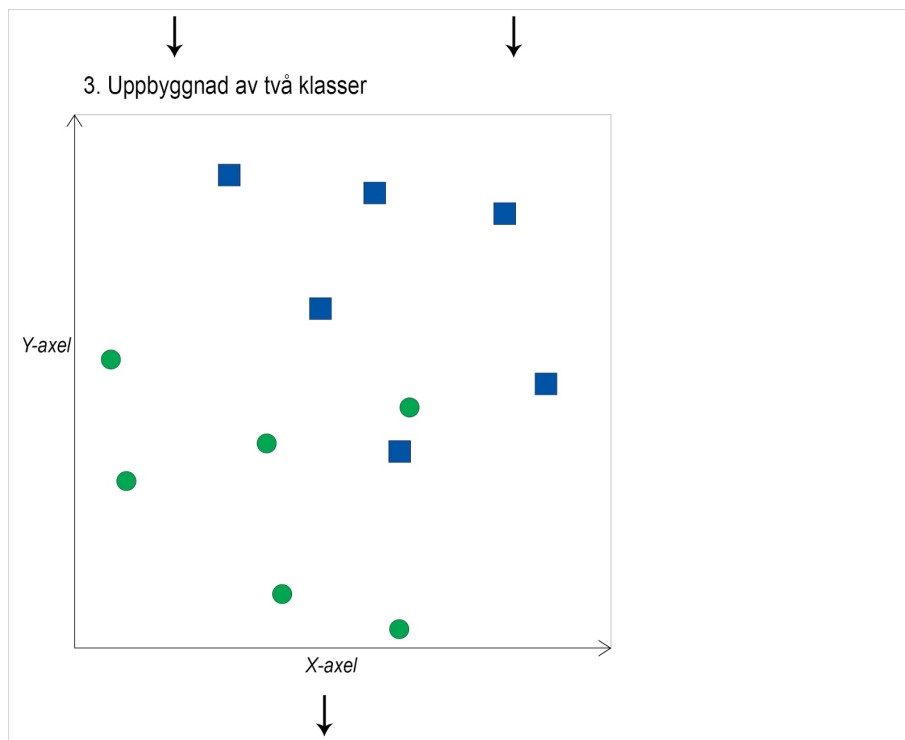
Klass 1	Klass 0
$[k_{1,1} \ k_{1,2} \ k_{1,3}]$	$[k_{1,1} \ k_{1,2} \ k_{1,3}]$
$[k_{2,1} \ k_{2,2} \ k_{2,3}]$	$[k_{2,1} \ k_{2,2} \ k_{2,3}]$
⋮	⋮
$[k_{40,1} \ k_{40,2} \ k_{40,3}]$	$[k_{40,1} \ k_{40,2} \ k_{40,3}]$

Varje rad motsvarar en egenskapsvektor.
Varje element motsvarar en egenskapsvärde.



Figur 4 - Från mätning till klassificering

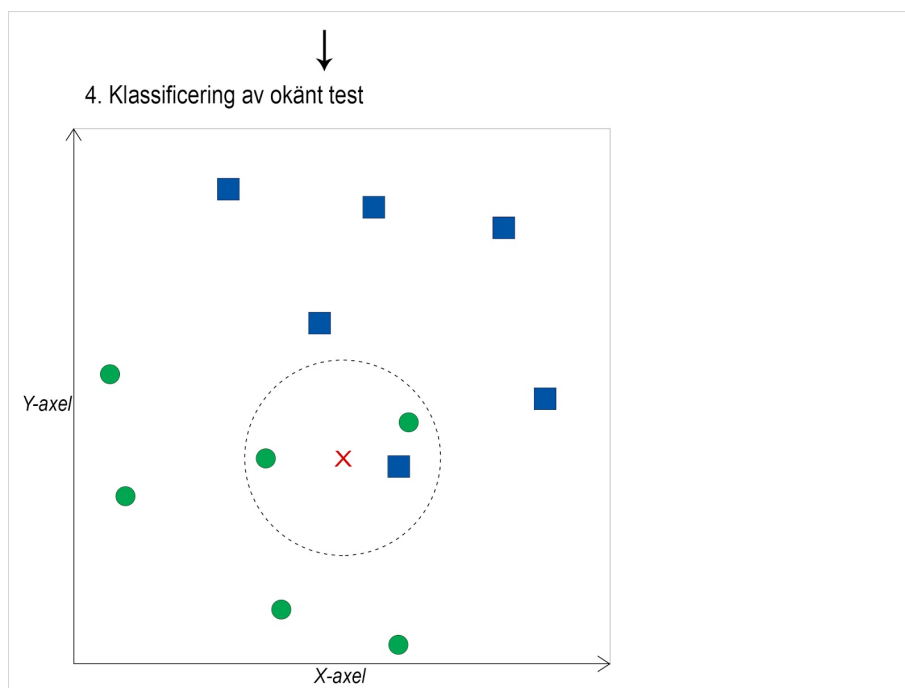
Antalet dimensioner avgörs av antalet egenskapsvärden i egenskapsvektorn, vilket beskrivs i avsnitt 2.3. Eftersom tre egenskapsvärden per egenskapsvektor användes i denna studie representerades datan i tre dimensioner. I Figur 5 visas en förenklad tvådimensionell representation av hur k-NN bygger upp två klasser.



Figur 5 - klasser i två dimensioner.

En grön cirkel representerar en kalibreringsmätning i en klass. En blå kvadrat representerar en kalibreringsmätning i en annan klass. Kalibreringsmätningens placering i X-led motsvarar värdet av det första egenskapsvärdet i en tvådimensionell egenskapsvektor. Placering i Y-led motsvarar det andra egenskapsvärdet i samma vektor.

Varje testmätning transformerades till två egenskapsvektorer enligt Figur 3. Därefter klassificerade k-NN en egenskapsvektor i taget. I denna studie motsvarar koordinatsystemets axlar medelvärdena av frekvensspannen som tagits fram i förbehandlingen. Klassificering av en egenskapsvektor illustreras i Figur 6.



Figur 6 - klassificering med k-NN i två dimensioner

Egenskapsvektorn som motsvarar ett test markeras i Figur 6 med ett rött kryss. Den streckade cirkeln har ringat in de tre närmaste grannarna till testet. Eftersom två av de tre närmaste grannarna är gröna cirkelar så klassificeras testet till den gröna klassen av k-NN.

3.4.4 Matlab

Vi använde Matlabs inbyggda funktion för att bygga upp k-NN klasser (`fitcknn`) samt klassificering (`predict`). Dessutom användes Matlabs funktion för absolutbelopp (`abs`) samt en funktion (`fft`) för Discrete Fourier Transform. Se bilaga 8.2 Matlabkod.

3.4.5 Konfidensintervall

I denna genomförbarhetsstudie ämnade vi endast att dra slutsatser om deltagarnas resultat. Vi hade inget intresse av att göra slutsatser om eventuell genomförbarhet för andra människor än försökspersonerna. Av denna anledning använde vi standardavvikelsen vid beräkning av konfidensintervall i denna studie.

3.5 Optimeringskalibrering

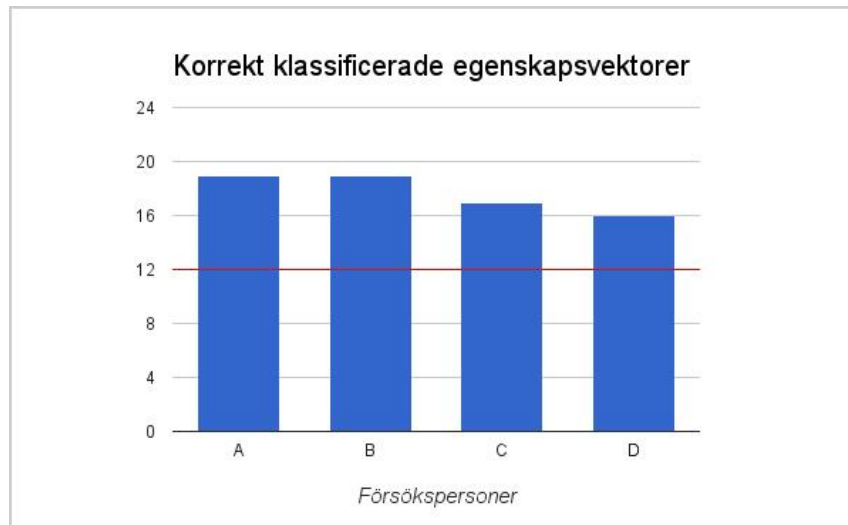
Alla människors hjärnvågor har unika mönster (Teplan 2002). Av denna anledning anpassade vi vilka frekvenskonfigurationer som användes för varje försöksperson. Vi testade 52 olika frekvenskonfigurationer per försöksperson. Programmet som testade konfigurationerna hittade den konfiguration med flest korrekt klassificerade operationer. Även antalet grannar kan *individuell*t anpassas för k-NN (Chaovalitwongse et al. 2007). Av denna anledning undersökte vi vilket antal grannar som gav bäst resultat. Vi varierade från en till 40 grannar, där 40 motsvarar antalet egenskapsvektorer i en klass. Processen att hitta den bästa konfigurationen av frekvensband och antalet grannar kallar vi i denna studie *optimeringskalibrering*.

Om fördelningen av antalet korrekta klassificeringar är ojämnt fördelad över operationerna så är inte nödvändigtvis alla operationer genomförbara. Det finns därmed ett intresse av att kalibrera k-NN för en jämn fördelning av operationer. Vid två tillfällen stötte vi på olika konfigurationer med samma antal korrekt klassificerade operationer. Vid dessa tillfällen valde vi den konfigurationen som gav jämnast fördelning mellan de korrekt klassificerade operationerna.

Optimeringskalibreringarna optimerade resultatet med hänsyn till antal korrekt klassificerade operationer. Den optimerade kalibreringen kan ses som en *modell* för att uppnå det bästa möjliga utfallet med metoden som används i föreliggande studie.

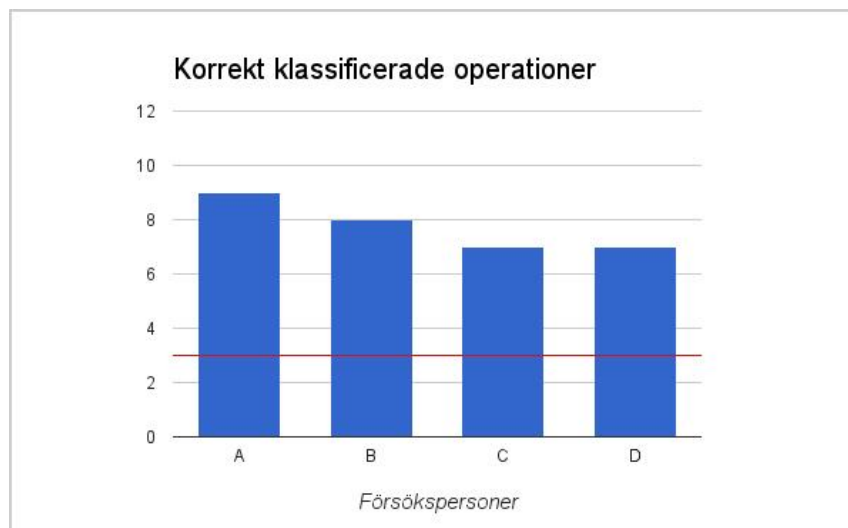
4. Empiri

I den första delen av empirin beskriver vi antalet korrekt klassificeringar för varje del av modellens resultat. Därefter presenteras konfidensintervallen för samma resultat. Se bilaga 8.2 för slumputräkningar.



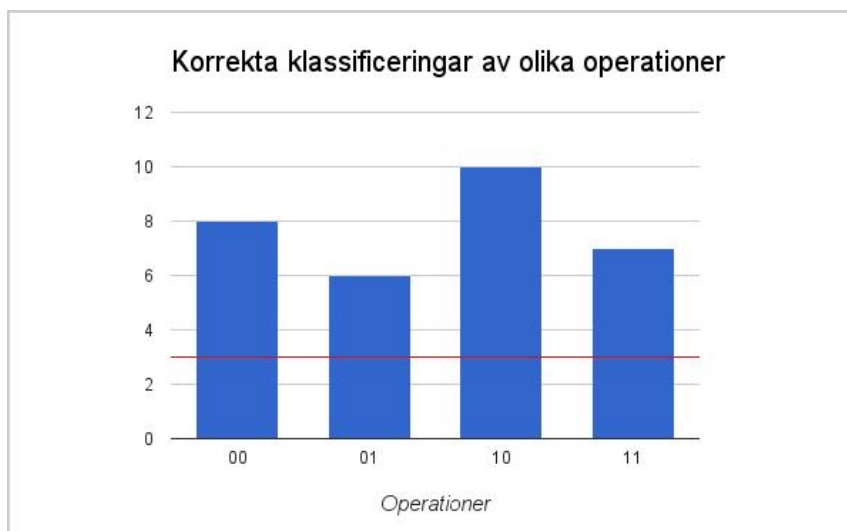
Figur 7 - Korrekt klassificerade tankemönster
Röd linje i figuren är utfallet för en slumpmässig klassificering.

Av Figur 7 utläses att författarna (A & B) fick sammanlagt bättre resultat för korrekt klassificerade egenskapsvektorer än de övriga försökspersonerna (C & D). Dessutom utläses att antalet klassificeringar av egenskapsvektorer för varje försöksperson låg ovanför en slumpmässig klassificering. Försöksperson A uppnådde bäst resultat med 79% av egenskapsvektorerna, och försöksperson D sämst med 67%, se bilaga 8.1.



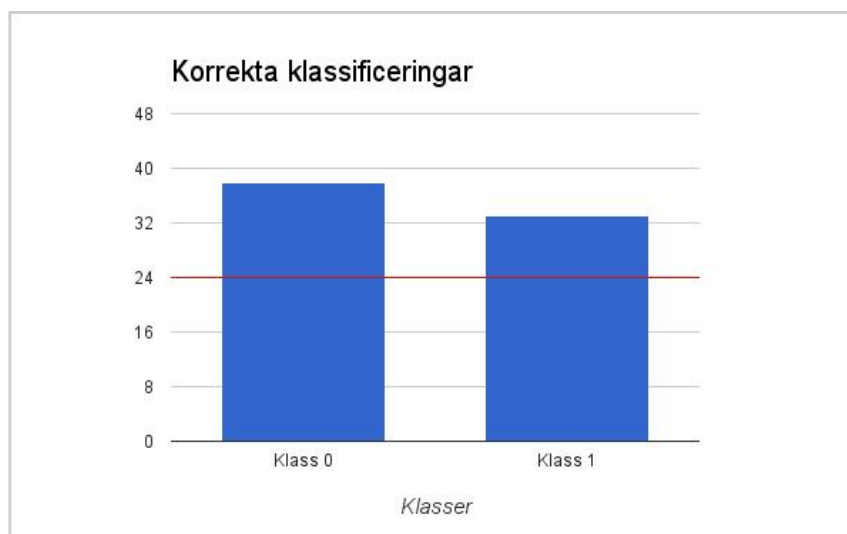
Figur 8 - Korrekt klassificerade operationer
Röd linje i figuren är utfallet för en slumpmässig klassificering.

Av Figur 8 utläses att författarna (A & B) fick sammanlagt bättre resultat för korrekt klassificerade operationer än de övriga försökspersonerna (C & D). Dessutom utläses att antalet korrekt klassificerade operationer för samtliga försökspersonerna låg ovanför en slumpmässig klassificering.



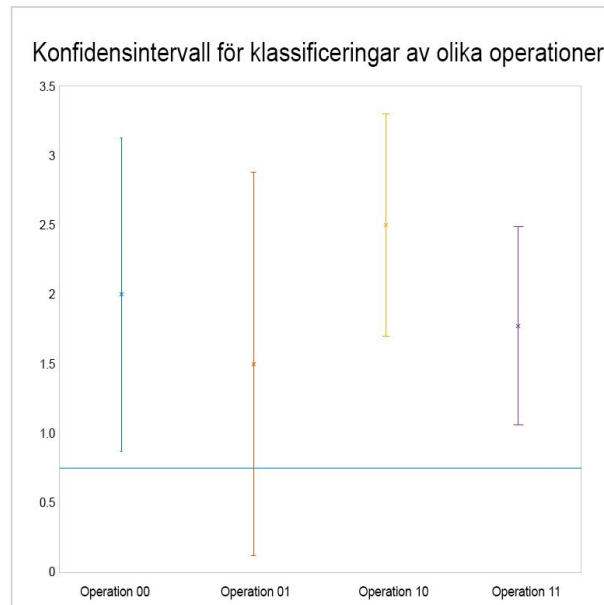
Figur 9 - Korrekta klassificeringar av olika operationer
Röd linje i figuren är utfallet för en slumpmässig klassificering.

Av Figur 9 utläses att fördelningen av antalet korrekta klassificeringar av operationer inte var jämn mellan de olika operationerna 00,01,10 & 11. Dessutom utläses att antalet korrekt klassificerade operationer för varje enskild operation låg ovanför en slumpmässig klassificering.



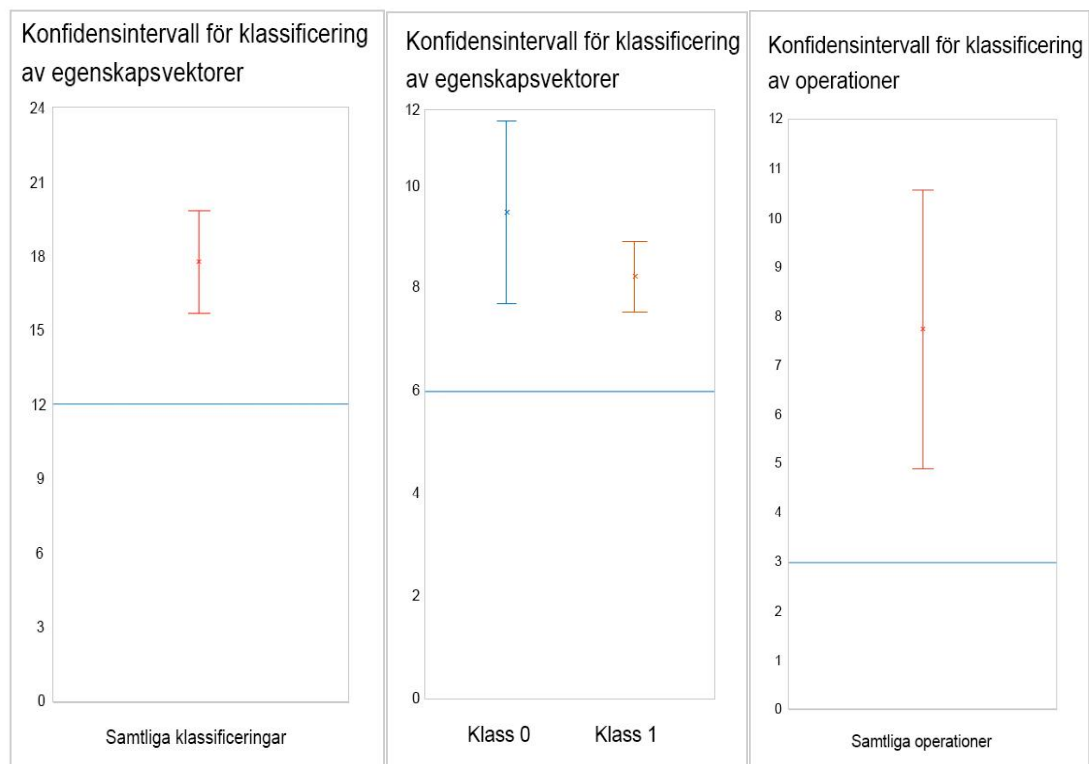
Figur 10 - Antal korrekta klassificeringar
Röd linje i figuren är utfallet för en slumpmässig klassificering.

Av Figur 10 utläses att fördelningen av korrekta klassificeringar mellan klasserna Klass 1 och Klass 0 inte var likafördelade. Dessutom utläses att antalet nollor och antalet ettor som klassificerats låg ovanför en slumpmässig klassificering.



Figur 11 - Konfidensintervall för klassificering av olika operationer
Horisontell blå linje i figuren är utfallet för en slumpmässig klassificering.

Av Figur 11 utläses att slumpen hamnade inom konfidensintervallet för Operation 01. Vilket innebär att klassificeringar av *Operation 01* kan bero på slumpen vid konfidensgraden 95%. Vidare utläses av Figur 11 att klassificering av Operation-00,10 & 11 inte beror på slumpen, med 95% konfidensgrad.



Figur 12 - konfidensintervall för klassificering av egenskapsvektorer
Figur 13 - konfidensintervall för klassificering av egenskapsvektorer
Figur 14 - konfidensintervall för klassificering av operationer
Horisontell blå linje i figurerna är utfallet för slumpmässig klassificering.

Av Figur 12 utläses att konfidensintervallet för klassificering av egenskapsvektorer till klass 0 och klass 1 inte innefattade slumpen, med 95% konfidensgrad. Dessutom utläses att konfidensintervallet för *klass 0* var större än konfidensintervallet för *klass 1*. Av Figur 13 avläses att konfidensintervallet för klassificering av egenskapsvektorer till båda klasserna inte innefattar slumpen, med 95% konfidensgrad. Av Figur 14 avläses att konfidensintervallet för klassificering av en operationsmätning till någon av de fyra olika operationerna inte innefattar slumpen.

5. Diskussion

Här diskuteras metodiken i föreliggande studie och därefter resultatet. Resultatet ses som det bästa möjliga utfallet med metodiken i föreliggande studie, vilket beskrivs i avsnitt 3.5. Faktorer som eventuellt kan förbättra resultatet beskrivs i detta avsnitt.

5.1 Metodkritik

Post hoc-teoretisera innebär att testa en hypotes på data som redan är anpassad för att svara på denna hypotes (Pedhazur et al. 2006). Eftersom vi utför optimeringskalibreringar och inte testar de optimerade kalibreringarna på ny data så medför det att göra faktiska slutsatser om genomförbarhet är att post hoc-teoretisera. På grund av tidsbegränsning gjorde vi inga ytterligare mätningar för att testa den optimerade kalibreringen på ny data.

5.1.1 Experimentellt upplägg

Försökspersonen meddelades när det var dags att byta från ett tankemönster till ett annat, vilket beskrivs i avsnitt 3.3.2. I och med att försökspersonen meddelades verbalt kan den externa stimuli som uppstått vid meddelandet ha påverkat försökspersonens koncentration, och därmed resultatet. Eftersom personen inte meddelades när samma tankemönster följde så kunde inte stimuli påverka dessa mätningar. Därmed hade det enligt oss varit fördelaktigt att vara konsekvent med meddelandet. Påverkan som meddelandet kan ha haft är till exempel att fler operationer registrerats när försökspersonen övergick från ett avkopplat till ett fokuserat tillstånd, förutsatt att stimuli orsakade ett fokuserat tankemönster.

För att utföra tankemönstren fokus och avkoppling använde vi två mentala övningar som beskrivs i avsnitt 3.1. Det finns fler mentala övningar, utöver dem som används i denna studie, som kan användas i ett BCI för att orsaka tankemönster (Lotte et al. 2007). Ett exempel är att utföra ett mantra. Ett mantra är en upprepning av ett ord eller ett ljud som utförs antingen tyst eller utan ljud (Dunn et al. 1999). Ett annat exempel är föreställd rotation av en geometrisk figur (Lotte et al. 2007). Grunden till vårt val av mental övning för tankemönstret fokus var att vi ansåg att mental matematik orsakar ett fokuserat mentalt tillstånd. Den mentala övningen vi valde för matematik har tidigare använts i en annan studie (Penny et al. 2000). Ett annat val av mentala övningar för att orsaka tankemönstren hade eventuellt medfört ett bättre resultat.

Vi ämnade att i undersökningen klassificera två tankemönster i följd. När vi mätte samma tankemönster i följd så mätte vi ett tankemönster i fem sekunder istället för två kortare tankemönster. Det hade enligt oss därmed varit mer representativt att även meddela försökspersonen när samma tankemönster följde. Dessutom kan det vara svårare att upprätthålla ett tankemönster i fem sekunder jämfört med en sekund. Anledningen att vi inte meddelade försökspersonen var att vi inte ville påverka försökspersonen med extern stimuli.

Eftersom mätningarna var omfattande så är även ansträngning och tidsåtgång en faktor för genomförbarheten. Beroende på koncentrationsnivå kan hjärnvågorna se annorlunda ut (Nicolas-Alonso & Gomez-Gil 2012). Eftersom en människas koncentrationsförmåga generellt sjunker efter förlängd koncentration kan därmed mätningarnas kvalitet avta med tiden.

Det var studieansvariges uppgift att avgöra när en sekund hade förlöpt. Eftersom en människas uppskattning av tid varierar kan längden av pausen påverkats. Pausens längd kan ha påverkat om försökspersonen hunnit ställa om till nästa tankemönster, eller slutat för tidigt med det första tankemönstret. Då tankemönstren kan ha

påverkats kan även resultatet påverkats. Mer konsekventa pauser hade uppnåtts om *försökspersonen* startade en timer som börjar räkna ner, startar mätningen, och meddelar försökspersonen efter en sekund.

Under studiens gång noterades att den tidigare nämnda mobila applikationen Recorder gav fler störningsindikationer vid låg batterinivå än vid hög batterinivå. Mindwaves indikator för lågt batteri upptäcktes lysa efter mätningarna för försöksperson B var slutförda. Därmed kan mätningarna för försöksperson B vara av lägre kvalitet.

5.1.2 Försökspersoner

För att undersöka vår problemformulering krävdes endast en försöksperson. Om vi använt en försöksperson hade vi eventuellt hunnit testa optimeringskalibreringen på ny data, vilket är nödvändigt för att fullt ut besvara problemformuleringen. Dessutom hade försökspersonen fått mer träning. Mer träning hade kunnat medföra ett bättre resultat eftersom träning kan påverka hur tydligt ett tankemönster uppfattas i en BCI implementation (Nicolas-Alonso & Gomez-Gil 2012). Anledningen att fyra försökspersoner användes var att öka säkerheten i det eventuella svaret på problemformuleringen. Vi ansåg att det var nödvändigt att ha med andra försökspersoner än oss författare för att öka trovärdigheten av studien.

5.1.3 Optimeringskalibrering

Optimeringskalibrering är i denna studie processen att hitta den bästa konfigurationen av frekvensband och antalet grannar. Programmet hittar de konfigurationer som leder till optimalt antal klassificerade operationer. Programmet tar dock inte hänsyn till fördelningen av operationer. Som beskrivs i avsnitt 3.5 är en jämn fördelning att föredra. Vi gjorde två val mellan konfigurationer för att optimera fördelningen men vi var inte konsekventa med denna metodik. Därmed kan fördelningen av antalet klassificeringar per operation vara missvisande. Genom att justera programmet så att det tar hänsyn till fördelningen så skulle en jämnare fördelning kunna uppnås. Programmet justerades inte på grund av tidsbrist.

5.2 Resultat

Optimeringskalibreringarna behöver testas på ny data för att fullt ut besvara problemformuleringen utan att vara post hoc-teoretisering, vilket beskrivs i avsnitt 5.1. Trots detta kan slutsatser baserade på modellen ge indikationer om fortsatt användande av modellen är att rekommendera utan att vara post hoc-teoretisering. Därmed analyserades resultatet av modellen för att avgöra om vi kan rekommendera att testa modellen på ny data.

Till skillnad från operation 11 och operation 00 innehåller operation 10 och operation 01 alternerande tankemönster. Tankemönster alterneras inte när endast ett tankemönster används i en operation. Alternerande tankemönster ger större insikt i genomförbarheten av att utföra två tankemönster i följd eftersom *två* tankemönster används i en operation. Därmed lägger vi diskussionens fokus på operation 10 och operation 01.

Figur 11 visar att operation 10 inte innefattade slumpen med 95% konfidensgrad, vilket indikerar att alternerandet av tankemönster för operation 10 är genomförbart i modellen med 95% konfidensgrad. Däremot kan genomförbarheten varken bekräftas eller förkastas gällande operation 01, eftersom slumpen innefattas i konfidensintervallet.

Föreliggande studie kan inte fastställa att operation 01 är genomförbar i modellen med 95% konfidensgrad i bästa fall. Studien indikerar dock att operation 01 går att genomföra i modellen eftersom försöksperson A utförde operation 01 tre av tre gånger. Försöksperson A utförde också operation 10 två av tre gånger vilket stärker indikationen, se bilaga 8.1. Den fjärde operationen skulle eventuellt kunna genomföras i modellen om en utjämning av operationerna utfördes enligt avsnitt 5.1.3.

Det finns så vitt vi vet inga studier som visat att EEG-instrument med en aktiv elektrod som använder två tankemönster för att utföra fler än två operationer, vilket nämns i avsnitt 1.2. Avsaknaden av liknande studier försvårar jämförelse för att bedöma hur rimligt modellens resultat är. Trots detta gör vi en bedömning genom att jämföra modellens resultat med andra studier som använt k-NN i kombination med EEG. På detta sätt bildar vi en uppfattning om modellen är rimlig eller inte. Modellen i denna studie uppnådde en korrekt klassificering av 74% för enskilda tankemönster, se bilaga 8.1.

En studie som utförts med k-NN och samma antal kalibreringsmätningar uppnådde 88% korrekt klassificering (Kayikcioglu & Aydemir 2010). En annan studie uppnådde 83% korrekt klassificering (Murugappan 2010). Däremot använde denna studie 64 elektroder, jämfört med en elektrod som används i föreliggande studie. En tredje studie uppnådde 81% korrekt klassificering (Güneş et al. 2010). En fjärde studie klassificerar k-NN med 41% säkerhet (Schlögl et al. 2005). I denna studie användes k-NN på egenskapsvektorer med hög dimensionalitet, vilket är ett förhållande då k-NN presterar dåligt, se avsnitt 2.3. Samtliga ovan nämnda studier skiljer sig från denna studie i aspekterna dimensionalitetsreducering och tankemönster. Trots varierande grader av korrekt klassificering anser vi, baserat på nämnda studier, att modellens resultat är rimligt. Eftersom vi anser att modellen är rimlig så ser vi studiens resultat som en indikation på hur resultatet skulle se ut i praktiken, vilket behöver konfirmeras i fortsatta studier.

5.3 Möjliga användningsområden

Till vår vetskap kan instrument med endast en aktiv elektrod, som MindWave, endast utföra två operationer, vilket nämns i avsnitt 1.2.

Förutsatt att det är möjligt att känna igen tankemönster i följd enligt metodiken i denna studie så skulle antalet möjliga operationer kunna utökas. Vidare kan potentiellt tankemönstren användas i längre följder för att uppnå en större mängd kombinationer av tankemönster, och därmed ännu fler operationer. Detta skulle kunna medföra nya, och förbättringar inom befintliga, användningsområden inom EEG. Däremot kan tiden det tar att utföra en enskild operation öka när antalet möjliga operationer utökas. Tiden kan öka eftersom fler tankemönster i följd behövs för att utföra en operation.

Rörelsehindrade är en användargrupp som kan dra nytta av en utökad mängd möjliga operationer. Livskvalitén för rörelsehindrade kan höjas genom att styra TV, ljussättning och temperatur i hemmet med hjälp av EEG. EEG kan därmed hjälpa till att minska mängden arbete för omhändertagare och öka livskvalitén för rörelsehindrade. (Nicolas-Alonso & Gomez-Gil 2012) Genom att utöka antalet möjliga operationer för EEG-instrumentet MindWave anser vi att möjligheterna utökas för styrning i hemmet med MindWave.

Spelintresserade kan också ha intresse i en utökad mängd möjliga operationer. Att MindWave endast kan använda två operationer begränsar användningsområden för MindWave inom spel. Genom att utöka antalet möjliga operationer kan MindWave användas inom spel i en större utsträckning än tidigare. De utökade operationerna ger spelutvecklare fler möjligheter vilket kan resultera i en roligare spelupplevelse.

En annan användargrupp som enligt oss har intresse av EEG är VR-användare. Som nämnt i avsnitt 1.1 kan EEG användas för att styra ett objekt i tre dimensioner. Eftersom EEG kan användas för att styra ett objekt i tre dimensioner anser vi att EEG framöver kan användas för att styra en avatar i en VR-värld. Detta innebär att användaren potentiellt kan sitta helt stilla och samtidigt utföra rörelser i den virtuella världen. Kontroll av en avatar i en virtuell värld följer inte från att känna igen två tankemönster i följd enligt metodiken i denna studie. Däremot anser vi att den potentiella utökningen av funktionalitet som undersöks i denna studie skulle kunna öka intresset för både EEG och en eventuell kombination av EEG och VR-teknologi.

6. Slutsats

Här presenteras slutsatsen av föreliggande studie.

Föreliggande studie ämnade besvara om det är genomförbart att utföra fyra operationer med EEG-instrumentet MindWave Mobile Headset?

Föreliggande studie visar att det är genomförbart att utföra tre av fyra operationer i modellen. Den fjärde operationen utfördes tre av tre gånger för en av försökspersonerna, vilket indikerar att den är genomförbar. Genomförbarheten av den fjärde operationen i modellen fastställdes dock inte eftersom slumpen innefattades i dess konfidensintervall.

I och med att modellen behöver testas på ny data är slutsatser om problemformuleringen att post hoc-teoretisera. Vi anser dock att modellen, i jämförelse med andra studier, är rimlig. Därmed rekommenderar vi en fortsatt studie.

6.1 Nästa steg

För att bedöma genomförbarheten av att utföra fyra operationer bör testerna upprepas med samma försökspersoner. Samma försökspersoner rekommenderas eftersom kalibreringarna i modellen anpassas individuellt till försökspersonerna.

7. Referenslista

- Akin, M., 2002. Comparison of wavelet transform and FFT methods in the analysis of EEG signals. *Journal of Medical Systems*, 26(3), pp.241–247.
- Chaovalitwongse, W.A., Fan, Y.J. & Sachdeo, R.C., 2007. On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(6), pp.1005–1016.
- Crowley, K. et al., 2010. Evaluating a brain-computer interface to categorise human emotional response. *Proceedings - 10th IEEE International Conference on Advanced Learning Technologies, ICALT 2010*, pp.276–278.
- De Maesschalck, R., Jouan-Rimbaud, D. & Massart, D.L., 2000. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1), pp.1–18. Available at: <http://www.sciencedirect.com/science/article/pii/S0169743999000477>.
- Dunn, B.R., Hartigan, J. a & Mikulas, W.L., 1999. Concentration and mindfulness meditations: unique forms of consciousness? *Applied psychophysiology and biofeedback*, 24(3), pp.147–165.
- Güneş, S., Polat, K. & Yosunkaya, Ş., 2010. Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Systems with Applications*, 37(12), pp.7922–7928. Available at: <http://www.sciencedirect.com/science/article/pii/S095741741000343X> [Accessed May 14, 2016].
- Huang, T.L. & Charyton, C., 2008. A comprehensive review of the psychological effects of. *Alternative Therapies*, 14(5), pp.38–49.
- Jasper, H.H., 1958. The ten-twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology*, 10(2), pp.371–375. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/10590970>.
- Jurcak, V., Tsuzuki, D. & Dan, I., 2007. 10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4), pp.1600–11. Available at: <http://www.sciencedirect.com/science/article/pii/S1053811906009724> [Accessed November 25, 2015].
- Kayikcioglu, T. & Aydemir, O., 2010. A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data. *Pattern Recognition Letters*, 31(11), pp.1207–1215. Available at: <http://www.sciencedirect.com/science/article/pii/S0167865510001212> [Accessed March 15, 2016].
- Lin, C.-T. et al., 2011. Novel dry polymer foam electrodes for long-term EEG measurement. *Biomedical Engineering, IEEE Transactions on*, 58(5), pp.1200–1207.
- Lotte, F. et al., 2007. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering*, 4(2), p.R1.
- McFarland, D.J. et al., 1997. Spatial filter selection for EEG-based communication. *Electroencephalography and Clinical Neurophysiology*, 103(3), pp.386–394.
- Michie, E.D., Spiegelhalter, D.J. & Taylor, C.C., 1994. Machine Learning , Neural and Statistical Classification. *Proceeding*.
- Murugappan, M., 2010. Classification of human emotion from EEG using discrete wavelet transform. *Journal of Biomedical Science and Engineering*, 03(04), pp.390–396.

- Neurosky.com, (2015). Body and Mind. Quantified. [online] Available at: <http://store.neurosky.com/> [Accessed 14 May. 2016].
- Nicolas-Alonso, L.F. & Gomez-Gil, J., 2012. Brain computer interfaces, a review. *Sensors (Basel, Switzerland)*, 12(2), pp.1211–79. Available at: <http://www.mdpi.com/1424-8220/12/2/1211/htm> [Accessed April 19, 2016].
- Pedhazur, E.J., Schmelkin, L.P. & Press., P., 2006. *Measurement, design, and analysis : an integrated approach*, New York; Hove, E. Sussex: Psychology Press, Taylor & Francis Group.
- Penny, W.D. et al., 2000. EEG-based communication: A pattern recognition approach. *IEEE Transactions on Rehabilitation Engineering*, 8(2), pp.214–215.
- Schlögl, A. et al., 2005. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of neural engineering*, 2(4), pp.L14–L22.
- Soininen, H. et al., 1989. Longitudinal EEG spectral analysis in early stage of Alzheimer's disease. *Electroencephalography and Clinical Neurophysiology*, 72(4), pp.290–297.
- Song, Y. et al., 2007. IKNN: Informative K-Nearest Neighbor Pattern Classification. *Proc. of the European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp.248–264.
- Teplan, M., 2002. Fundamentals of EEG measurement. *Measurement Science Review*, 2, pp.1–11.
- Weinberger, K.Q. & Saul, L.K., 2009. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10, pp.207–244. Available at: http://dl.acm.org/citation.cfm?id=1577069.1577078\files/261/JMLR-Weinberger-Saul-2009-Distance_Metric_Learning_for_Large_Margin_Nearest_Neighbor_Classification2.pdf.
- Wolpaw, D.J.M. and W.A.S. and J.R., 2010. Electroencephalographic (EEG) control of three-dimensional movement. *Journal of Neural Engineering*, 7(3), p.36007. Available at: <http://stacks.iop.org/1741-2552/7/i=3/a=036007>.
- Wolpaw, J.R. & McFarland, D.J., 1994. Multichannel EEG-based brain-computer communication. *Electroencephalography and Clinical Neurophysiology*, 90(6), pp.444–449. Available at: <http://www.sciencedirect.com/science/article/pii/S001346949490135X> [Accessed May 6, 2016].
- Xiang, S., Nie, F. & Zhang, C., 2008. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12), pp.3600–3612. Available at: <http://www.sciencedirect.com/science/article/pii/S0031320308002057> [Accessed February 13, 2016].
- Yazdani, A., Ebrahimi, T. & Hoffmann, U., 2009. Classification of EEG signals using Dempster Shafer theory and a k-nearest neighbor classifier. *Neural Engineering, 2009. NER '09. 4th International IEEE/EMBS Conference on*, pp.327–330.

8. Bilagor

8.1 Resultat

	<u>Sammanfattning av studie</u>		
	Antal	Procentuell	Slumpprocent
Antal test	48		
Antal egenskapsvektorer	96		
Antalet korrekt klassificerade egenskapsvektorer	71	74%	50%
Antalet korrekt klassificerade operationer	31	65%	25%
Antalet korrekt klassificerade klass 0	38	79%	50%
Antalet korrekt klassificerade klass 1	33	69%	50%
Operation 00	8	67%	25%
Operation 01	6	50%	25%
Operation 10	10	83%	25%
Operation 11	7	58%	25%

Figur 15 - Sammanlagda resultat för försökspersonerna

	<u>Försöksperson A</u>			<u>Försöksperson B</u>		
	Antal	Procentuell	Slumpprocent	Antal	Procentuell	Slumpprocent
Antal test	12			12		
Antal egenskapsvektorer	24			24		
Antalet korrekt klassificerade egenskapsvektorer	19	79%	50%	19	79%	50%
Antalet korrekt klassificerade operationer	9	75%	25%	8	67%	25%
Antalet korrekt klassificerade klass 0	11	92%	50%	10	83%	50%
Antalet korrekt klassificerade klass 1	8	67%	50%	9	75%	50%
	Deltest 1	Deltest 2	Deltest 3	Deltest 1	Deltest 2	Deltest 3
Operation 00	Rätt	Rätt	Rätt	Fel	Rätt	Rätt
Operation 01	Rätt	Rätt	Rätt	Rätt	Fel	Fel
Operation 10	Rätt	Rätt	Fel	Rätt	Rätt	Rätt
Operation 11	Fel	Rätt	Fel	Rätt	Fel	Rätt
	<u>Försöksperson C</u>			<u>Försöksperson D</u>		
	Antal	Procentuell	Slumpprocent	Antal	Procentuell	Slumpprocent
Antal test	12			12		
Antal egenskapsvektorer	24			24		
Antalet korrekt klassificerade egenskapsvektorer	17	71%	50%	16	67%	50%
Antalet korrekt klassificerade operationer	7	58%	25%	7	58%	25%
Antalet korrekt klassificerade klass 0	9	75%	50%	8	67%	50%
Antalet korrekt klassificerade klass 1	8	67%	50%	8	67%	50%
	Deltest 1	Deltest 2	Deltest 3	Deltest 1	Deltest 2	Deltest 3
Operation 00	Fel	Rätt	Rätt	Rätt	Fel	Fel
Operation 01	Fel	Rätt	Fel	Fel	Fel	Rätt
Operation 10	Fel	Rätt	Rätt	Rätt	Rätt	Rätt
Operation 11	Fel	Rätt	Rätt	Rätt	Rätt	Fel

Figur 16 - Detaljerade resultat per försöksperson

8.2 Slumputräkning

När en egenskapsvektor klassificeras kan k-NN välja mellan två klasser. I ett slumpmässigt utfall har en korrekt klassificering av en egenskapsvektor därmed 50% sannolikhet att inträffa. En egenskapsvektor representerar ett tankemönster. En operation består av två egenskapsvektorer i följd. Därmed har en operation 25% sannolikhet att inträffa.

Figur 7

Varje försöksperson utförde 24 tankemönster vilket resulterar i slumpen 12.

Figur 8

Varje försöksperson utförde 12 operationer vilket resulterar i slumpen 3.

Figur 9

Varje försöksperson utförde 12 test av varje operation vilket resulterar i slumpen 3.

Figur 10

Försökspersonerna utförde totalt varje tankemönster 48 gånger vilket resulterar i slumpen 24.

Figur 11

Varje försöksperson utförde varje operation 3 gånger vilket resulterar i slumpen 0.75.

Figur 12

Varje försöksperson utförde varje tankemönster 12 gånger vilket resulterar i slumpen 6.

Figur 13

Varje försöksperson utförde 24 tankemönster totalt vilket resulterar i slumpen 12.

Figur 14

Varje försöksperson utförde 12 operationer vilket resulterar i slumpen 3.

8.3 Matlabkod

8.3.1 Kalibreringsmätningar

- Alla kalibreringsmätningar hämtas för en försöksperson.
- Varje kalibreringsmätning kapas till fem sekunder.
- Kalibreringsmätningarna delas upp i mindre delar.
- Vi transformerar kalibreringsmätningarna till frekvensdomänen och tar absolutbeloppet av varje värde.
- Vi tar medelvärdet av de tre förbestämda frekvensspannen, vilket resulterar i en egenskapsvektor med tre egenskapsvärden. Varje kalibreringsmätning resulterar i fem egenskapsvektorer.
- Vi sparar alla egenskapsvektorer i en matris och skickar till k-NN.
- Vi skapar en vektor som beskriver motsvarande tankemönstren för varje kalibreringsmätning för k-NN. På detta sätt får k-NN reda på vilken kategori varje kalibreringsmätning tillhör.
- k-NN skapar de två klasserna och är redo för att klassificera testmätningar.

8.3.2 Testmätningar

- Alla testmätningar hämtas för en försöksperson.
- Varje testmätning kapas till fem sekunder.
- Testmätningarna delas upp i mindre delar.
- Vi transformerar testmätningarna till frekvensdomänen och tar absolutbeloppet av varje värde.
- Vi tar medelvärdet av de tre förbestämda frekvensspannen, vilket resulterar i en egenskapsvektor med tre egenskapsvärden. Varje testmätning resulterar i två egenskapsvektorer.
- Vi sparar alla egenskapsvektorer i en matris.

8.3.3 Klassificering

- Vi skickar en testmätning i taget till k-NN och får ut en klassificering, 1 eller 0, för tankemönster 1 respektive tankemönster 0.

Därefter utför matrisoperationer för att få ut konkreta svar för antal korrekt klassificerade egenskapsvektorer och operationer. Dessutom fås information om vilka operationer som klassificerades korrekt. Även information om antalet gånger klass 0 och klass 1 klassificerade för försökspersonen i fråga. Eftersom k-NN ser på antalet grannar i närheten behövs en majoritet. När det blev oavgjort så avgjordes detta av den mest närliggande grannen till testet.

```

clear all;

%Frekvensbandsintervallerna f?r de tre olika egenskapskoefficienterna
f1start = 8;
f1end = 13;
f2start = 14;
f2end = 18;
f3start = 19;
f3end = 24;

%Specificerar antalet grannar som k-NN algoritmen anv?nder
nrOfNeighbours=30;

nrOfSegments=5; %Antal segment (Figur1, punkt 2)
fs = 512; %EEG-instrumentet har samplingsfrekvens 512Hz

%Avg?r hur l?nga segmenten ?r beroende p? samplingsfrekvens
Second = fs;

%%Kalibreringsm?tningar
%H?mtar alla kalibreringsm?tningar
allTextFilesCalibrations= dir('*kal*');

%Inl?sning av data
for i = 1:length(allTextFilesCalibrations)
    dataAllCalibrations(:,i) =
importdata(allTextFilesCalibrations(i).name);
end

for i=1:length(allTextFilesCalibrations)
    dataAllCalibrations(i).name = allTextFilesCalibrations(i).name;
end
%Skapar beh?llare f?r datan
Structure.segments = struct([]);
Structure.segmentsFFT = struct([]);

for k = 1:length(allTextFilesCalibrations)
    %Best?mmer hur m?nga och hur l?nga segment i
    %kalibreringsm?tningarna.
    tempMatrix=zeros(nrOfSegments,512);
    N=length(dataAllCalibrations(k).data);
    tempData = dataAllCalibrations(k).data;

    S=1;
    E=S+Second-1;

    i=0;
    while i<nrOfSegments

        %Placerar segmenten i en matris

        %Duplicerar segmenten och applicerar absolutbelopp samt
        %transformerar segmenten till frekvensplanet med hj?lp av FFT.
        i=i+1;
        tempMatrix(i,:) = tempData(S:E)';
        tempMatrixFFT(i,:)= abs(fft(tempData(S:E)'));
        S=E+1;
    end
end

```

```

        E=S+Second-1;

    end

    %Placerar matriserna i en cellstruktur
    Structure(k).segments = [tempMatrix];
    Structure(k).segmentsFFT = [tempMatrixFFT];

    %Förbereder en vektor för egenskapskoefficienter
    featureValues = zeros(nrOfSegments,3);

    for f = 1:nrOfSegments

        %Placerar egenskapskoefficienter i vektorn featureValues
        featureValues(f,1) = mean(tempMatrixFFT(f,f1start:f1end));
        featureValues(f,2) = mean(tempMatrixFFT(f,f2start:f2end));
        featureValues(f,3) = mean(tempMatrixFFT(f,f3start:f3end));

    end

    %Placerar samtliga egenskapskoefficienter för en kalibrering i en
    %cellstruktur.
    Structure(k).featureVector = [featureValues];
    Structure(k).name = dataAllCalibrations(k).name; %justera str-
ngarna i namnen

    end

%%Testmetningar
%Samma procedur som för kalibreringsmetningarna enligt ovan med undantaget
att vi enbart tar ut första och
%ista segmentet av metningen.

%Hämtar alla testmetningar
allTextFilesTests= dir('*test*');

for i = 1:length(allTextFilesTests)
    dataAllTests(:,i) = importdata(allTextFilesTests(i).name);
end

for i=1:length(allTextFilesTests)
    dataAllTests(i).name = allTextFilesTests(i).name;
end

testStructure.segments = struct([]);
testStructure.segmentsFFT = struct([]);

for k = 1:length(allTextFilesTests)
    tempMatrix=zeros(nrOfSegments,512);
    N=length(dataAllTests(k).data);
    tempData = dataAllTests(k).data;

    S=1;
    E=S+Second-1;

    i=0;
    while i<nrOfSegments

```

```

        i=i+1;
        tempMatrix(i,:) = tempData(S:E)';
        tempMatrixFFT(i,:)= abs(fft(tempData(S:E)'));

        S=E+1;
        E=S+Second-1;
    end

    testStructure(k).segments = [tempMatrix];
    testStructure(k).segmentsFFT = [tempMatrixFFT];

    featureValues = zeros(nrOfSegments,2);

    for f = 1:nrOfSegments

        featureValues(f,1) = mean(tempMatrixFFT(f,f1start:f1end));
        featureValues(f,2) = mean(tempMatrixFFT(f,f2start:f2end));
        featureValues(f,3) = mean(tempMatrixFFT(f,f3start:f3end));

    end
    testStructure(k).featureVector = [featureValues];
    testStructure(k).name = dataAllTests(k).name;
end

%%Klassificering

%Bygger upp en matris av egenskapsvektorer
class = [];
for i = 1:length(dataAllCalibrations)
    for k = 1:nrOfSegments
        class = [class;Structure(i).featureVector(k,1:3)];
    end
end

%Ger en klass till egenskapsvektorerna f?r kalibreringarna
knnClassInput = ['0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';
'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';
'0';'0';'0';'0';'0';'0';'0';'0';'0';'0';'1';'1';'1';'1';'1';'1';'1';'1';
'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';
'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1';'1'];

%Anv?nder Matlabs inbyggda k-NN algoritm (fitcknn) f?r att bygga upp
tv? klasser
Mdl =
fitcknn(class,knnClassInput,'NumNeighbors',nrOfNeighbours,'Distance','mahalanobis');

%Anv?nder Matlabs inbyggda k-NN algoritm (predict) f?r att klassificera
%egenskapsvektorerna som tillh?r testm?tningarna

%Notera att b?de fitcknn och predict anv?nds tillsammans, d?r fitcknn
%bygger klasserna och predict klassificerar egenskapsvektorerna.

classifications=[];
for i = 1:length(testStructure)
    classifications =
[classifications;str2num(predict(Mdl,testStructure(i).featureVector(1,1:3))

```



```

)];
    classifications =
[classifications;str2num(predict(Mdl,testStructure(i).featureVector(5,1:3))
)];
    end

    %En korrekt klassificering av alla test skulle resultera i
    %följande svar.
    expectedOutput = [0,0,0,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1,0,1,1,1,1,1,1]';

    %%Resultat
    %Ber?knar antalet korrekt klassificerade test
    CorrectAnswers = zeros(24,1);
    for i = 1:24
        if classifications(i) == expectedOutput(i)
            CorrectAnswers(i) = 1;
        end
    end

    %Ber?knar antalet korrekta klassificeringar individuellt f?r de olika
    %klasserna
    zero = 0;
    one = 0;
    for i = 1:length(expectedOutput)
        if expectedOutput(i) == 0 && CorrectAnswers(i) == 1
            zero = zero+1;
        end
        if expectedOutput(i) == 1 && CorrectAnswers(i) == 1
            one = one+1;
        end
    end

    Storage = [];
    Storage(:,1) = CorrectAnswers(1:2:length(CorrectAnswers));
    Storage(:,2) = CorrectAnswers(2:2:length(CorrectAnswers));

    %Ber?knar antal korrekt klassificerade operationer
    CorrectAnswersSequencial = [];
    for i = 1:length(Storage)
        if Storage(i,1) ==1 && Storage(i,2) == 1
            CorrectAnswersSequencial(i) = 1;
        else
            CorrectAnswersSequencial(i) = 0;
        end
    end

    %Strukturerar resultat
    CorrectAnswersSequencial = CorrectAnswersSequencial';
    CorrectMatrix = [];
    CorrectMatrix(:,1) =
CorrectAnswersSequencial(1:3:length(CorrectAnswersSequencial));
    CorrectMatrix(:,2) =
CorrectAnswersSequencial(2:3:length(CorrectAnswersSequencial));
    CorrectMatrix(:,3) =
CorrectAnswersSequencial(3:3:length(CorrectAnswersSequencial));
    for i = 1:length(Storage)
        testStructure(i).answers = [Storage(i,1:2)];
    end

```

```
disp('-----')
disp('Antal test')
disp(length(classifications/2))
disp('Antal egenskapsvektorer')
disp(length(CorrectAnswers))
disp('Antalet korrekt klassificerade egenskapsvektorer')
disp(sum(CorrectAnswers))
disp('Antalet korrekt klassificerade operationer')
disp(sum(CorrectAnswersSequencial));
disp('Antalet korrekt klassificerade nollor')
disp(sum(zero))
disp('Antalet korrekt klassificerade ettor')
disp(sum(one))
```

