




Universitetet  
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

## MASTEROPPGAVE

|  |   |
|--|---|
| Studieprogram/spesialisering:<br>Automatisering og signalbehandling  | Vårsemesteret, 2018<br><br>Åpen / <del>Konfidensiell</del>  |
| Forfatter: Svein Erik Stokka   | <br>(signatur forfatter) |
| Fagansvarlig: Professor Trygve Eftestøl<br><br>Veileder: Professor Trygve Eftestøl   |   |
| Tittel på masteroppgaven: Sannsynlighetsmodell for monitorering av kvalitet av hjertekompresjoner basert på sammenheng mellom kvalitetsparametre for hjertelungeredning og fysiologisk respons<br><br>Engelsk tittel: Probabilistic model to monitor quality of chest compressions based on relationship between parameters reflecting quality of cardiopulmonary resuscitation and physiological response |   |
| Studiepoeng: 30  |   |
| Emneord:<br>Hjertestans utenfor sykehus<br>Hjertelungeredning<br>Maskinlæring<br>Signalbehandling  | Sidetall: 51<br><br>+ vedlegg/annet: 13<br><br>Stavanger, 15.06.2018<br>dato/år                             |



# Sammendrag

*Bakgrunn:* Plutselig hjertestans utenfor sykehus er en alvorlig medisinsk tilstand, hvor det i Norge er rundt 3200 tilfeller hvert år. For å overleve hjertestans er det viktig med tidlig og god hjertelungeredning(HLR). Ved å monitorere kvalitet av hjertekompresjoner vil det være mulig å gi tilbakemelding til personen som utfører kompresjonene. Denne tilbakemeldingen kan brukes til å forbedre kvaliteten av kompresjonene, og dermed øke sannsynligheten for overlevelse. Det gir også anledning til å studere data retrospektivt, slik at en kan identifisere faktorer som er betydningsfulle for overlevelse. *Material og metode:* Datasettet består av 394 episoder med hjertestans utenfor sykehus. Med utgangspunkt i datasettet er det utviklet metoder for å merke kompresjonsintervall som å være av gunstig eller ikke gunstig kvalitet, basert på sannsynlighet for endring til bedre tilstand,  $P_{des}$ . Ettersom  $P_{des}$  indikerer pasientens tilstand, vil dermed endring i  $P_{des}$  gi en indikasjon på fysiologisk respons. Datasettet består av kvalitetsparametre som beskriver egenskaper for hjertekompresjoner. For å automatisk assosiere disse kvalitetsparametrene med fysiologisk respons, er det eksperimentert med 3 forskjellige maskinlæringsmetoder for klassifisering, hvor det beregnes sannsynlighet for gunstige kompresjoner,  $P_{gc}$ . Dermed benyttes kombinasjon av flere kvalitetsparametre til å gi tilbakemelding om kvalitet av behandling, som én verdi. Maskinlæringsmetodene som ble benyttet er: sannsynlighetsmaksimeringsmetoden(ML-metoden), logistisk regresjon(LR) og nevralt nettverk(NN). For å evaluere metodene ble klassifiseringsnøyaktighet benyttet som indikasjon på ytelse. Endelig evaluering ble utført ved å studere fordeling av  $P_{gc}$  mot kvalitetsparametre, og sammenligne fordelingen opp mot retningslinjer for HLR. *Resultat:* ML-metoden, LR og NN klassifiserte med en treffprosent på henholdsvis 58,61%, 62,96% og 63,39%. Forhold mellom  $P_{gc}$  og kvalitetsparametre viste at LR fordeler  $P_{gc}$  best i henhold til retningslinjer for HLR. *Konklusjon:* Det er utviklet en sannsynlighetmodell som indikerer kvalitet av hjertekompresjoner, og som er i henhold til retningslinjene for HLR. Ved å indikere kvalitet av hjertekompresjoner som sannsynlighetsverdi, er det mulig å visualisere kvalitet på en kompakt og dynamisk måte.



# Forord

Denne oppgaven ble skrevet i forbindelse med fullførelse av mastergrad i Automatisering og Signalbehandling ved Universitet i Stavanger.

Jeg ønsker å takke min veileder, Professor Trygve Eftestøl, for engasjert og god veiledning. Han har gitt god oppfølging og bidratt med mange gode idéer. Jeg vil også takke Professor Jan Terje Kvaløy for bidrag med gode råd og idéer i veiledningsmøter.



# Innhold

|  |           |
|--|-----------|
| Sammendrag . . . . .   | i         |
| Forord . . . . .   | iii       |
| <b>1 Introduksjon</b>  | <b>1</b>  |
| 1.1 Kapitteloversikt . . . . .                                 | 2         |
| <b>2 Bakgrunn</b>  | <b>3</b>  |
| 2.1 Medisinsk . . . . .  | 3         |
| 2.1.1 Hjertelungeredning . . . . .                             | 5         |
| 2.2 Maskinl ring . . . . .                                     | 5         |
| 2.2.1 Sannsynlighetsmaksimeringsmetoden (ML-metoden) . . . . . | 6         |
| 2.2.2 Logistisk regresjon . . . . .                            | 7         |
| 2.2.3 Nevralt nettverk . . . . .                               | 10        |
| 2.2.4 Transformasjon av datasett . . . . .                     | 15        |
| <b>3 Material og metode</b>                                    | <b>17</b> |
| 3.1 Datasett . . . . .   | 17        |
| 3.2 Hente relevante datapunkt . . . . .                        | 19        |
| 3.2.1 Metode 1 . . . . .                                       | 20        |
| 3.2.2 Metode 2 . . . . .                                       | 20        |
| 3.2.3 Metode 3 . . . . .                                       | 20        |
| 3.2.4 Algoritme . . . . .                                      | 21        |
| 3.3 Databehandling . . . . .                                   | 24        |
| 3.3.1 Inndeling i trening- og testsett . . . . .               | 24        |
| 3.3.2 Databehandlingsteknikker . . . . .                       | 25        |
| 3.4 Eksperiment . . . . .                                      | 29        |
| 3.4.1 ML-metoden . . . . .                                     | 29        |
| 3.4.2 Logistisk regresjon . . . . .                            | 30        |
| 3.4.3 Nevralt nettverk . . . . .                               | 30        |
| 3.5 Evaluering av resultat . . . . .                           | 32        |
| 3.5.1 Klassifiseringsn yaktighet . . . . .                     | 32        |

---

|          |   |           |
|----------|---|-----------|
| 3.5.2    | Vurdering mot retningslinjer for HLR . . . . .            | 33        |
| <b>4</b> | <b>Resultat</b>   | <b>35</b> |
| 4.1      | Klassifiseringsegenskaper . . . . .                       | 35        |
| 4.2      | Fordeling av sannsynlighetsverdier . . . . .              | 38        |
| 4.3      | Undersøkelse av valgt sannsynlighetsmodell . . . . .      | 39        |
| 4.4      | Visualisering av hjertestansepisode . . . . .             | 41        |
| <b>5</b> | <b>Diskusjon</b>  | <b>45</b> |
| 5.1      | Feilkilder . . . . .                                      | 46        |
| 5.2      | Konklusjon . . . . .                                      | 47        |
| 5.3      | Fremtidig arbeid . . . . .                                | 47        |
|          | <b>Referanser</b>   | <b>48</b> |
|          | <b>Vedlegg A</b>  | <b>53</b> |
| A.1      | Log-transformasjon . . . . .                              | 53        |
| A.2      | Sammenligning med og uten siling av treningsett . . . . . | 54        |
| A.3      | Fordeling av sannsynlighetsverdier . . . . .              | 55        |
| A.3.1    | ML-metoden . . . . .                                      | 56        |
| A.3.2    | Logistisk regresjon . . . . .                             | 57        |
| A.3.3    | Nevralt nettverk . . . . .                                | 58        |
| A.4      | Utvidet testsett . . . . .                                | 59        |
| A.5      | Sammenligning av sannsynlighetsmodeller . . . . .         | 60        |
|          | <b>Vedlegg B</b>  | <b>63</b> |
| B.1      | Matlab-kode . . . . .                                     | 63        |
| B.2      | Python-kode . . . . .                                     | 65        |



# Kapittel 1

## Introduksjon

Plutselig hjertestans utenfor sykehus er en alvorlig medisinsk tilstand som betyr at hjertet har sluttet å pumpe blod rundt i kroppen. Forekomst varierer avhengig av kontinent, hvor Australia og Nord-Amerika har hyppigst forekomst med henholdsvis 111.9 og 98.1 per 100 000 person-år[1]. I Norge er forekomsten av plutselig hjertestans på 61 per 100 000 person-år[2]. Med et befolkningstall på litt under 5,3 millioner i Norge[3], tilsvarer det rundt 3200 tilfeller av hjertestans utenfor sykehus, hvert år.

For å overleve hjertestans er det viktig med tidlig og god hjertelungeredning(HLR)[4]. Ved å monitorere kvalitet av hjertekompresjoner vil det være mulig å gi tilbakemelding til personen som utfører kompresjonene. Denne tilbakemeldingen kan brukes til å forbedre kvaliteten av kompresjonene[5], og dermed øke sannsynligheten for overlevelse[6]. Det gir også anledning til å studere data retrospektivt, slik at en kan identifisere faktorer som er betydningsfulle for overlevelse.

I denne studien utvikles metoder som automatisk klassifiserer hjertekompresjoner som å være av gunstig kvalitet eller ikke. For å detektere kompresjoner, benyttes metoder fra tidligere studie[7]. I samme studie ble det også benyttet kvalitetsparametre for HLR, basert på kompresjonsdybde(KD) og impedansmåling(thorax impedans(TI)). For å assosiere hjertekompresjoner som å være av gunstig kvalitet eller ikke, benyttes sannsynlighet for endring til bedre tilstand( $P_{des}$ ), utviklet i en tidligere studie[8]. Ettersom  $P_{des}$  indikerer pasientens tilstand, vil dermed endring i  $P_{des}$  gi en indikasjon på fysiologisk respons.

Hensikten med denne studien er å utvikle en probabilistisk modell av forholdet mellom kvalitetsparametre for HLR og fysiologisk respons. Sannsynlighetsmodellen beregner sannsynligheten for gunstige kompresjoner,  $P_{gc}$ . Ettersom kvalitet av hjertekompresjoner indikerers i form av sannsynlighetsverdi, er det mulig å

visualisere kvalitet av hjertekompresjoner på en kompakt og dynamisk måte, og kan brukes som tilbakemelding i et hjertestarter-system. Sannsynlighetsmodellen implementeres i et eksisterende system[8], hvor det presenteres et eksempel for hvordan  $P_{gc}$  kan benyttes til å studer data retrospektivt.

## 1.1 Kapitteloversikt

### **Kapittel 2 - Bakgrunn:**

Medisinsk og teknisk bakgrunn beskrives, hvor forskjellige sannsynlighetsmodeller og teknikker forklares.

### **Kapittel 3 - Material og metode**

Beskriver datasettet som er benyttet, samt fremgangsmåte for å hente ut relevante datapunkt. Alle teknikker for databehandling forklares, hvor deretter oppsett og utførelse av eksperiment beskrives. Til slutt forklares det hvordan sannsynlighetsmodellene evalueres.

### **Kapittel 4 - Resultat**

Oppsummerer klassifiseringsresultat for alle datasett og metoder. Viser også forhold mellom sannsynlighetsverdi og kvalitetsparametre, og sammenligner forholdet opp mot retningslinjer for HLR. Til slutt er det vist oversikt over en hjertestanseepisode hvor sannsynlighetsmodell er implementert i eksisterende program.

### **Kapittel 5 - Diskusjon**

Diskuterer resultat og eventuelle feilkilder. Basert på resultat konkluderes det om ytelse til modell, og fremtidig arbeid diskuteres.

### **Referanser**

Liste over alle referanser benyttet.

### **Vedlegg A**

Diverse sammenligninger.

### **Vedlegg B**

Programkode og funksjoner forklares.

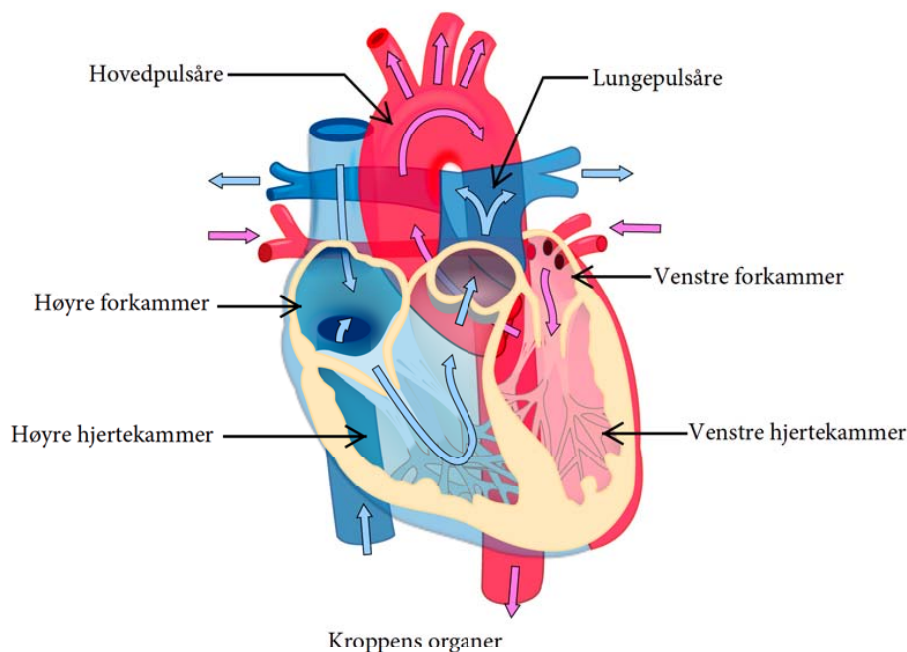
# Kapittel 2

## Bakgrunn

Dette kapitlet gir et overblikk over medisinsk og teknisk bakgrunn. I den medisinske delen beskrives hjertets funksjon, og behandling som utføres under hjertestans. Den tekniske delen beskriver teknologien benyttet, hvor forskjellige metoder for klassifisering forklares.

### 2.1 Medisinsk

Hjertet er kroppens viktigste muskel. Det ligger i brysthulen og har som oppgave å pumpe oksygenrikt blod rundt i kroppen. Hjertet består av totalt fire kammer, hvor to ligger i høyre del av hjertet og to i venstre del. Dette er vist i figur 2.1. Hver del består av et forkammer(atrium) og et hjertekammer(ventrikkel), som sammen pumper blodet ut i kroppen. Blodsirkulasjonen i hjertesystemet kan deles opp i 4 deler: 1. Blodet fra kroppens organer går gjennom høyre forkammer til høyre hjertekammer. 2. Fra hjertekammeret pumpes det videre til lungene via lungepulsårene. 3. Fra lungene strømmer så blodet til venstre forkammer som sender det videre til venstre hjertekammer. 4. Fra venstre hjertekammer pumpes så blodet ut til kroppens organer igjen, via hovedpulsåren[9, 10]. Blodet strømmer rundt i kroppen ved at forkamrene og hjertekamrene, i en koordinert rytme, trekker seg sammen og fungerer som en pumpe. Den koordinerte rytmen utføres av hjertets elektriske system, og kan overvåkes ved bruk av et elektrokardiogram(EKG). Figur 2.1 visualiserer hjertets oppbygning, hvor rød og blå piler indikerer blodsirkulasjonen.



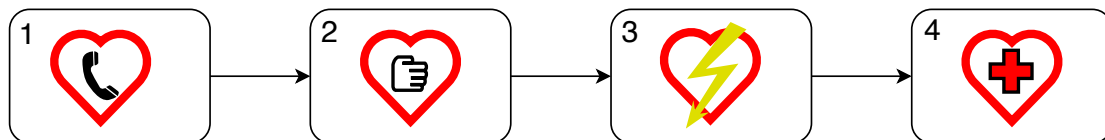
**Figur 2.1:** Figuren viser blodløpet i hjertet. Piler med rød farge representerer retning for oksygenrikt blod, og blå piler representerer retning for oksygenfattig blod. (Kilde: "Heart diagram blood flow en.svg" av ZooFari[11], med lisens Creative Commons Attribution-Share Alike 3.0 Unported(CC BY-SA 3.0)[12]. Bildet er redigert med norsk tekst og sorte piler.)

Ved hjerrestans slutter hjertet å pumpe blod rundt i kroppen. Det betyr at den elektriske impulsstyringen i hjertet har sviktet, og kan skyldes flere årsaker. Den vanligste årsaken er akutt eller kronisk sykdom i hjertet[13]. Det kan for eksempel være blokkeringer i hjertets pulsårer på grunn av blodpropp, som medfører hjerteinfarkt, og kan ende med hjerrestans. Hjerterytme som observeres i EKG, under hjerrestans, blir fordelt i 5 kategorier: ventrikkeltakykardi(VT), ventrikkelflimmer(VF), asystole(AS), pulsløs elektrisk aktivitet(PEA) og pulsgenererende rytme(PR). VT gjenkjennes som en hurtig hjerterytme, hvor det ikke pumpes tilstrekkelig blod rundt i kroppen. Ofte er VT en midlertidig hjerterytme, før den spontant endres til VF, gjenkjent som en uorganisert hjerterytme. Dersom det er tilnærmet ingen hjerterytme, og ingen mekanisk aktivitet i hjertet, karakteriseres hjerterytmen som AS. PEA gjenkjennes som en organisert hjerterytme, men uten mekanisk aktivitet i hjertet. PR er en organisert rytme, hvor hjertet har oppnådd spontan sirkulasjon[14].

I denne oppgaven rangeres hjerterytme fra klinisk mest gunstig til minst gunstig, basert på en tidligere utført studie[15]. Rangeringen benyttet her er: PR, PEA, VF/VT og AS, hvor PR er mest gunstig. Her betraktes VT og VF som samme rytme. Rangeringen begrunnes ut fra en tidligere utført studie, hvor sannsynlighet for endring av hjerterytme undersøkes ved bruk av Markov sannsynlighetsmodell[16]. Basert på høyest sannsynlighet for å endre hjerterytme til PR, ble rangering valgt.

### 2.1.1 Hjertelungeredning

Hjertelungeredning(HLR) er behandlingen som utføres dersom noen har hjertestans. Behandlingen deles opp i 4 trinn, hvor alle trinnene er oppsummert i figur 2.2. Første trinn er å se etter livstegn og tilkalle hjelp. Trinn 2 er å starte hjertekompresjoner og ventilasjoner, med forholdet 30:2. Her er det viktig med tidlig og god utførelse. Hensikten med hjertekompresjoner og ventilasjoner er å opprettholde oksygentilførsel til hjernen og andre viktige organer. Retningslinjene for HLR beskriver at optimal behandling består av kompresjonsdybde rundt 5 cm, men ikke over 6 cm, og kompresjonsrate mellom 100 og 120 kompresjoner per minutt[17]. Når hjertestarter er tilgjengelig, enten fra ambulanse eller hjertestarter plassert i nærheten, kobles denne til pasienten. Dersom den underliggende hjerterytmen er VF eller VT utføres det elektrisk sjokk, i håp om å endre hjerterytmen til PR, vist som trinn 3 i figur 2.2. Siste trinn er å utføre videre behandling på sykehus.



**Figur 2.2:** Figuren viser fremgangsmåte for behandling av hjertestans. 1) Se etter livstegn og tilkall hjelp. 2) Utfør hjertekompresjoner og ventilasjoner. 3) Gi elektrisk sjokk med defibrillator. 4) Videre behandling på sykehus.

## 2.2 Maskinlæring

Her beskrives teori for de forskjellige maskinlæringsteknikkene benyttet til klassifisering og modellering. Alle teknikkene tar utgangspunkt i Bayes' teorem, hvor målet er å modellere en funksjon som indikerer kvalitet av behandling i form av sannsynlighetsverdi. Ved bruk av Bayes' beslutningsteori velges den mest sannsynlige klassen, noe som minimerer sannsynligheten for feilklassifisering[18]. Sannsynligheten for klasse  $i$  er gitt av  $P(w_i|\underline{x})$ , og kalles à posteriori-sannsynlighet. Denne er vist i ligning (2.1), hvor  $c$  representerer antall klasser og  $w_i$  indikerer klasse nr.  $i$ .  $\underline{x}$  er egenskapvektor med dimensjon  $d$ .

$$P(w_i|\underline{x}) = \frac{P(w_i)p(\underline{x}|w_i)}{p(\underline{x})}, \quad i = 1, 2, \dots, c \quad (2.1)$$

$P(w_i)$  beskriver kunnskap om data før klassifisering, og kalles à priori-sannsynlighet.  $p(\underline{x}|w_i)$  er sannsynlighetstetthetsfunksjon(pdf) for  $\underline{x}$ , gitt klasse  $i$ . I nevneren er

det en skaleringsfaktor,  $p(\underline{x})$ . Denne sørger for at den totale à posteriori-sannsynligheten, for alle klassene, summeres opp til 1. Formelen for  $p(\underline{x})$  er gitt i ligning (2.2).

$$p(\underline{x}) = \sum_{j=1}^c P(w_j) p(\underline{x}|w_j) \quad (2.2)$$

Ved å beregne à posteriori-sannsynlighet for alle klassene, kan  $\underline{x}$  klassifiseres ved å velge den klassen som ga høyest à posteriori-verdi. Dette er vist i ligning (2.3).

$$w = \underset{w_i}{\operatorname{argmax}} P(w_i|\underline{x}) \quad (2.3)$$

I delkapittel 2.2.1-2.2.3 forklares 3 forskjellige metoder for å tilnærme à posteriori-sannsynligheten. I delkapittel 2.2.1 forklares sannsynlighetsmaksimeringsmetoden (ML-metoden). Her beregnes og estimeres alle funksjonene i à posteriori-sannsynligheten, vist til høyre i ligning (2.1). Logistisk regresjon og nevralt nettverk forklares i henholdsvis kapittel 2.2.2 og 2.2.3, og er metoder for å tilnærme en matematisk funksjon. Målet er å tilnærme  $P(w_i|\underline{x})$ . Det gjøres ved å bruke klassen datapunkt er merket med, til å definere en kostfunksjon. Kostfunksjonen benyttes til å oppdatere vektorer i den matematiske funksjonen, for å best mulig representere datapunktene.

## 2.2.1 Sannsynlighetsmaksimeringsmetoden (ML-metoden)

I ML-metoden beregnes og estimeres alle funksjonene i  $P(w_i|\underline{x}, \underline{\theta}_i)$ , vist til høyre i ligning (2.1), hvor  $\underline{\theta}_i$  er parametre for klasse  $i$ . Datapunktene fra hver klasse behandles uavhengig og deles inn i datasett, noteret med  $D_i$ . Fordi parametrene i  $\underline{\theta}_i$  ikke er kjent, må disse estimeres. Dette gjøres ved bruk av sannsynlighetsmaksimeringsestimator (MLE)<sup>1</sup>. Her estimeres de parametrene som beskriver de funksjonene som med høyest sannsynlighet representerer data for klasse  $i$  [18]. Estimering av parametrene medfører at à posteriori ikke lenger garanterer å minimere feilklassifiseringen. Dette på grunn av unøyaktigheter i estimeringen. Utgangspunkt for MLE er gitt i ligning (2.4), hvor alle pdf'er for datapunkt i  $D_i$  multipliseres. Denne funksjonen kalles for en rimelighetsfunksjon.

$$L(D_i|\underline{\theta}_i) = \prod_{k=1}^n p(\underline{x}_k|\underline{\theta}_i), \quad D_i = [\underline{x}_1, \dots, \underline{x}_n], \quad i = 1, \dots, c \quad (2.4)$$

<sup>1</sup>Engelsk: Maximum likelihood estimation

Ved å ta den naturlige logaritmen av ligning (2.4) forenkles produktene til en sum, uten å endre posisjonen til toppunktet. Når funksjonen så deriveres med hensyn på  $\underline{\theta}_i$  og settes lik 0, kommer en frem til uttrykk for  $\underline{\theta}_i$ . Dersom det antas at alle pdf'ene har normalfordeling, slik det er i denne oppgaven, resulterer MLE med de estimerte parametre gitt i ligning (2.5). Her er  $\hat{\underline{\mu}}_i$  og  $\widehat{\underline{\Sigma}}_i$  henholdsvis gjennomsnitt og kovarians for klasse  $i$ .

$$\hat{\underline{\theta}}_i = \begin{bmatrix} \hat{\theta}_{i,1} \\ \hat{\theta}_{i,2} \end{bmatrix} = \begin{bmatrix} \hat{\underline{\mu}}_i \\ \widehat{\underline{\Sigma}}_i \end{bmatrix} = \begin{bmatrix} \frac{1}{n_i} \sum_{k=1}^{n_i} \underline{x}_k \\ \frac{1}{n_i} \sum_{k=1}^{n_i} (\underline{x}_k - \hat{\underline{\mu}}_i)(\underline{x}_k - \hat{\underline{\mu}}_i)^t \end{bmatrix} \quad (2.5)$$

Pdf'en for et datapunkt  $\underline{x}$ , for klasse  $i$ , skrives nå som vist i ligning (2.6).

$$p(\underline{x} | \omega_i, \hat{\underline{\mu}}_i, \widehat{\underline{\Sigma}}_i) = \frac{1}{(2\pi)^{2/d} |\widehat{\underline{\Sigma}}_i|^{1/2}} \cdot e^{-\frac{1}{2}(\underline{x} - \hat{\underline{\mu}}_i)^t \widehat{\underline{\Sigma}}_i^{-1} (\underline{x} - \hat{\underline{\mu}}_i)} \quad (2.6)$$

$P(w_i)$  beregnes som forholdet mellom antall datapunkt i klasse  $i$ ,  $n_i$ , og totalt antall datapunkt,  $n$ . Dette er vist i ligning (2.7).

$$P(w_i) = \frac{n_i}{n} \quad (2.7)$$

Den estimerte à posteriori-sannsynligheten beregnes nå ved å bruke ligningene gitt i (2.6) og (2.7).

## 2.2.2 Logistisk regresjon

Logistisk regresjon er en populær modell for klassifisering av kategoriske variabler. Det er en generalisert modell av lineær regresjon, hvor det modelleres en matematisk funksjon som beskriver sannsynlighet for  $y$ , gitt egenskapvektor  $\underline{x}$ ,  $P(y|\underline{x})$ [19]. Dette viser sammenhengen med à posteriori-sannsynlighet, hvor  $w_i = y$ . Dette

kapitlet beskriver teori for binomisk regresjonsmodell, som betyr at det klassifiseres mellom to klasser, 0 og 1. Ved klassifisering mellom flere klasser, benyttes en multinomisk modell.

Den logistiske modellen settes opp som vist i ligning (2.8), hvor  $\hat{p}$  er den estimerte sannsynligheten.  $\hat{p}$  uttrykker sannsynligheten for at  $\underline{x}$  tilhører klasse 1,  $P(y = 1|\underline{x})$ . Ligningen viser sammenhengen mellom lineær og logistisk regresjon, ved at høyresiden av likehetstegnet er en lineær funksjon av egenskapene i  $\underline{x}$ .

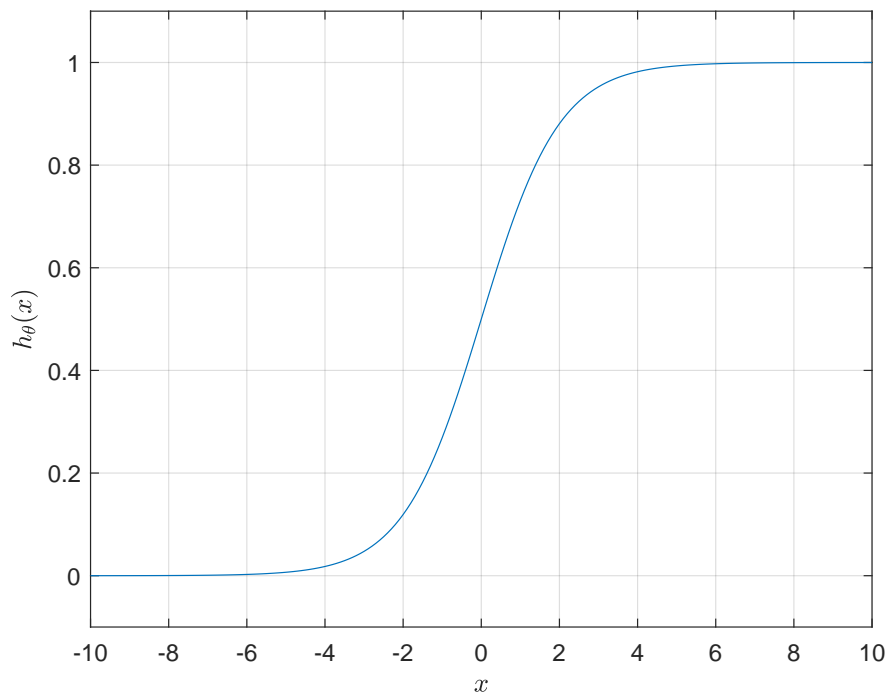
$$\ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = \underline{\theta}^t \underline{x} \quad \underline{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \underline{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{d+1} \end{bmatrix} \quad (2.8)$$

Det første elementet i  $\underline{x}$  er 1. Det påvirkes av parameteren  $\theta_1$  som beskriver biasleddet i modellen. Antall elementer i  $\underline{x}$  er avhengig av dimensjonen i datasettet. Dersom funksjonen som skal tilnærmes er ulineær, er det viktig å introdusere ulineære elementer i  $\underline{x}$ . Dette gjøres for eksempel ved å multiplisere eller kvadrere egenskaper, og legge disse til i egenskapsvektoren,  $\underline{x}$ . Å gjøre modellen ulineær medfører en mer fleksibel modell, noe som også øker risikoen for overtrening mot treningsdata[19]. Ved å løse ligning (2.8) med hensyn på  $\hat{p}$ , får en sigmoid-funksjonen, vist i ligning (2.9).

$$\hat{p} = h_{\underline{\theta}}(\underline{x}) = \frac{1}{1 + e^{-\underline{\theta}^t \underline{x}}} \quad (2.9)$$

Sigmoid-funksjonen er vist i figur 2.3. Figuren viser at funksjonen har horisontale asymptoter på  $h_{\underline{\theta}}(x)$  lik 0 og 1, noe som gjør det til en gunstig modell for sannsynlighet.





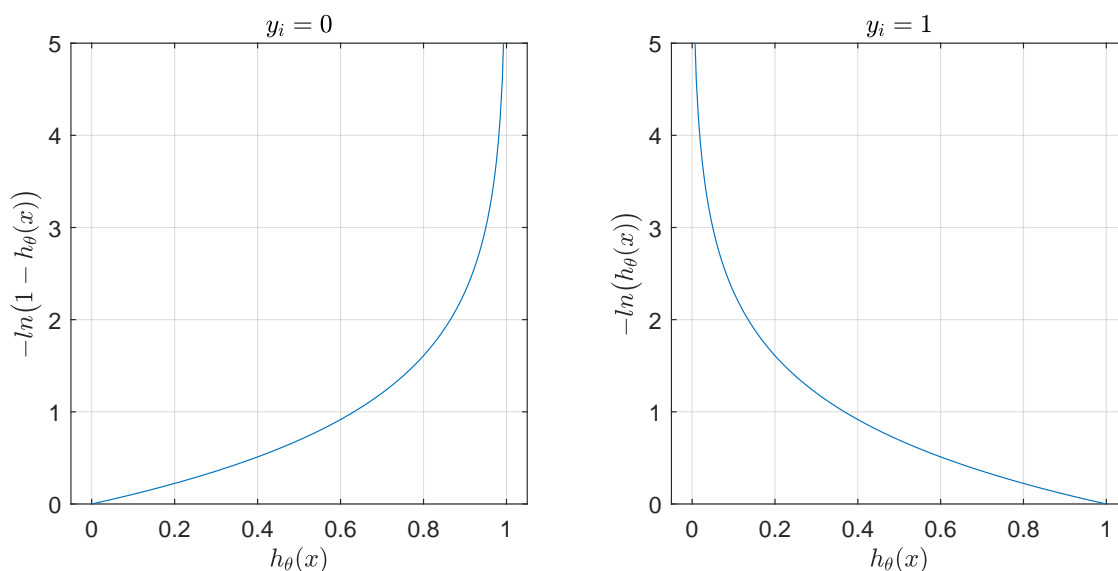
**Figur 2.3:** Sigmoid-funksjonen, gitt i ligning (2.9). Ser at funksjonen har vertikale asymptoter på  $h_{\underline{\theta}}(x)$  lik 0 og 1, noe som gjør det til en gunstig modell for sannsynlighet.

Målet med logistisk regresjon er å estimere de optimale parametrene  $\underline{\theta}$ . Dette gjøres ved bruk av MLE, som inneberer å finne parametrene som med høyest sannsynlighet representerer datasettet[20]. Utgangspunktet for MLE i logistisk regresjon er at datapunkt i datasettet er uavhengige og at  $y$ , som indikerer klassen til datapunktet, er en Bernoulli tilfeldig variabel. En Bernoulli tilfeldig variabel har to utfall, 0 eller 1. Koblingen mellom binomisk regresjonsmodell og Bernoulli variabelen er at summen av de uavhengige Bernoulli variablene er binomisk fordelt[21].

Ved bruk av MLE kommer en frem til funksjonen gitt i ligning (2.10). Denne funksjonen kalles for kryssentropi-tap[19]. Det finnes flere varianter av denne funksjonen, avhengig om optimaliseringen skal være et minimering- eller maksimeringsproblem. Denne funksjonen gir et minimeringsproblem, hvor optimal  $\underline{\theta}$  finnes ved å minimere  $J(\underline{\theta})$  med hensyn på  $\underline{\theta}$ .  $m$  er antall datapunkt i treningsdata.

$$J(\underline{\theta}) = -\frac{1}{m} \left( \sum_{i=1}^m y_i \ln(h_{\underline{\theta}}(x)) + (1 - y_i) \ln(1 - h_{\underline{\theta}}(x)) \right) \quad (2.10)$$

Funksjonen er plottet i figur 2.4 hvor den er delt opp i to deler, avhengig om datapunkt  $i$  tilhører klasse 0 ( $y_i = 0$ ) eller klasse 1 ( $y_i = 1$ ).



**Figur 2.4:** Funksjon for kryssentropi-tap delt opp i to deler, avhengig om datapunkt tilhører klasse 0 ( $y_i = 0$ ) eller klasse 1 ( $y_i = 1$ ). Observerer at dersom  $y_i$  tilhører klasse 0 og modellen,  $h_{\theta}(x)$ , gir en verdi nær 0, vil funksjonen gi en lav verdi.

Figur 2.4 viser at dersom  $y_i = 0$  og modellen,  $h_{\theta}(x)$ , gir en verdi nær 0, vil funksjonen gi en lav verdi. Dersom forskjellen mellom modellens predikerte sannsynlighet og faktisk klasse øker, observeres det at funksjonsverdien,  $J(\theta)$ , stiger eksponentielt. Dette er en gunstig egenskap ved trening av parametre, hvor dårlig prediksjon vil straffes med høy kost-verdi.

Fremgangsmåten for å estimere de optimale parametrene, med  $J(\theta)$  som utgangspunkt, er å benytte en optimaliseringsalgoritme. I denne oppgaven benyttes Quasi-Newton metoden[22, 23]. Den bruker en tilnærming av Hessian matrisen til å iterativt oppdatere parametrene[24].

### 2.2.3 Nevral nettverk

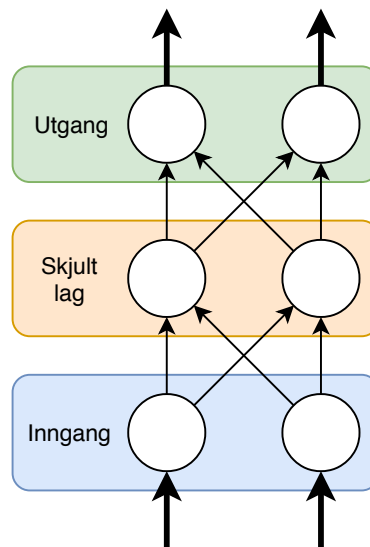
Et nevralt nettverk(NN) er, i likhet med logistisk regresjon, en metode for å tilnærme en matematisk funksjon. Den tilnærmede funksjonen beskriver et forhold mellom inngang og utgang, ved bruk av parametre,  $\underline{y} = f(\underline{x}; \underline{w})$ . Et nevralt nettverk lærer hvilke verdier for vektene  $\underline{w}$  som gir best funksjonstilnærming. Det finnes flere forskjellige nettverk, hvor de mest kjente er; forovermatet nevraltnett<sup>1</sup>(FNN), konvolusjonsnett<sup>2</sup>(CNN) og rekursivt nevraltnett<sup>3</sup>(RNN). Dette kapitlet dekker teori for FNN.

<sup>1</sup>Engelsk: Feedforward neural network

<sup>2</sup>Engelsk: Convolutional neural network

<sup>3</sup>Engelsk: Recurrent neural network

Strukturen for nevralt nettverk er inspirert av biologien i hjernen, hvor flere nevroner er koblet sammen. Figur 2.5 viser et FNN med 3 lag, hvor sirklene representerer nodene som er inspirert av nevronene. Koblingen mellom nevronene vises i form av piler, hvor hver pil også representerer et element i  $w$ .

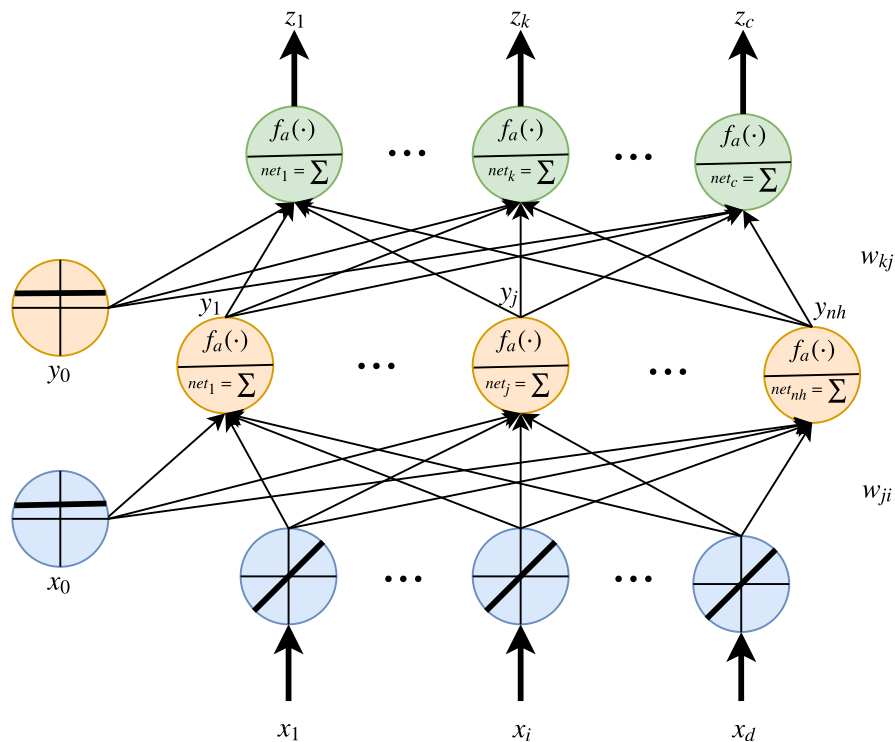


**Figur 2.5:** Et FNN med 3 lag, visualisert i form av et diagram. Sirklene representerer nodene som er inspirert av nevronene i hjernen. Figuren viser hvordan et nevralt nettverk deles opp i forskjellige lag, inngang, utgang og skjult lag. Et nettverk kan utvides ved å legge til flere skjulte lag.

Figur 2.5 viser hvordan et nevralt nettverk er delt opp i forskjellige lag, med inngang, utgang og skjult lag. Et nettverk kan utvides ved å legge til flere skjulte lag, som gir et dypt nevralt nettverk. Hver sirkel representerer en aktiveringsfunksjon,  $f_a(\cdot)$ , som transformerer inngangsverdien til funksjonen. Det finnes flere alternativer for valg av aktiveringsfunksjon, hvor rettet lineær enhet (ReLU) og sigmoid-funksjonen er blant de mest populære [25]. Ligning (2.11) viser sigmoid-funksjonen, som er den samme funksjonen benyttet i logistisk regresjon. Dersom det ikke benyttes skjulte lag, og aktiveringsfunksjonen på utgangen er en sigmoid-funksjon, vil det nevralt nettverket tilsvare en logistisk regresjonsmodell [18].

$$f_a(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad (2.11)$$

I et FNN går datapunkt fra inngang, gjennom vektene og skjulte lag, til utgang. Figur 2.6 visualiserer et generelt oppsett for FNN.



**Figur 2.6:** Et generelt oppsett for et FNN. Innganger markeres fra  $x_1$  til  $x_d$ , hvor  $d$  er antall egenskaper. Sirklene helt til venstre i figuren er konstantledd for nodene, representert av  $x_0$  og  $y_0$ . Disse har begge verdi lik 1. Verdiene ut fra inngangslaget, vist som blå sirkler, er lik verdien som blir sendt inn. Det illustreres med den lineære funksjonen  $y = x$ , i hver sirkel. Antall skjulte noder er gitt av  $nh$ , og  $c$  indikerer antall utganger(klasser). Hver pil representerer en vekt, hvor vekten mellom inngang  $i$  og node  $j$  er gitt av  $w_{ji}$ . Tilsvarende er vekten mellom node  $j$  og utgang  $k$  gitt av  $w_{kj}$ .

Innganger markeres fra  $x_1$  til  $x_d$ , hvor  $d$  er antall egenskaper i datapunkt. Sirklene helt til venstre i figuren er konstantledd for nodene, representert av  $x_0$  og  $y_0$ . Disse har begge konstant verdi lik 1. Verdiene ut fra inngangslaget, vist som blå sirkler, er lik verdien som blir sendt inn. Det illustreres med den lineære funksjonen  $y = x$ , i hver sirkel. I hver node summeres alle inngangsverdiene, før summen brukes i en aktiveringsfunksjon,  $f_a(\cdot)$ . Summen til inngangsverdiene for node nr.  $j$ , kalles for  $net_j$ . Den generelle formelen for  $net_j$  er gitt i ligning (2.12).

$$net_j = \sum_{i=1}^d w_{ji} x_i + w_{j0} = \sum_{i=0}^d w_{ji} x_i \quad \text{hvor } x_0 = 1 \quad (2.12)$$

Aktiveringsfunksjonen transformerer  $net$  og gir ut en skalar, vist i ligning (2.13).

$$y_j = f_a(net_j) \quad (2.13)$$

Ved å kombinere alle ligningene finner en et generelt uttrykk for utgang nummer  $k$ , vist i ligning (2.14).

$$z_k = f_a \left( \underbrace{\sum_{j=0}^{nh} w_{kj}}_{net_k} f_a \left( \underbrace{\sum_{i=0}^d w_{ji} x_i}_{net_j} \right) \right) \quad (2.14)$$

Fra inngangen,  $x$ , flyter datapunkt gjennom vektene og nodene, som resulterer i en verdi på utgangen,  $z$ . Dette kalles forover-forplantning. Resultatet fra  $z$  sammenlignes mot den sanne verdien til datapunktet,  $t$ . For å trene det nevralt nettverket er det nødvendig med et uttrykk som beskriver feilen mellom  $z$  og  $t$ . Til dette benyttes en kost-funksjon. Det finnes flere kost-funksjoner å velge mellom, hvor minste kvadraters metode<sup>1</sup>[26] og kryssentropi-tap er de mest populære. Sistnevnte er forklart i kapittel 2.2.2, og er kost-funksjonen benyttet for nevralt nettverk i denne oppgaven. Ligning (2.15) viser kryssentropi-tap for nettverket.

$$J(w) = -\frac{1}{m} \left( \sum_{i=1}^m t_i \ln(z_i) + (1 - t_i) \ln(1 - z_i) \right) \quad (2.15)$$

Med  $J(w)$  som utgangspunkt kan nettverket trenes ved å minimere  $J(w)$ , med hensyn på  $w$ . For å minimere  $J(w)$  brukes bakover-forplantning, som beregner gradienten til  $J(w)$ [25]. Gradienten beregnes ved bruk av kjerneregelen. Ligning (2.16) viser et eksempel hvor den deriverte av  $J(w)$  beregnes med hensyn på den siste vekten før utgang  $z_k$ ,  $w_{kj}$

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial z_k} \frac{\partial z_k}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} \quad (2.16)$$

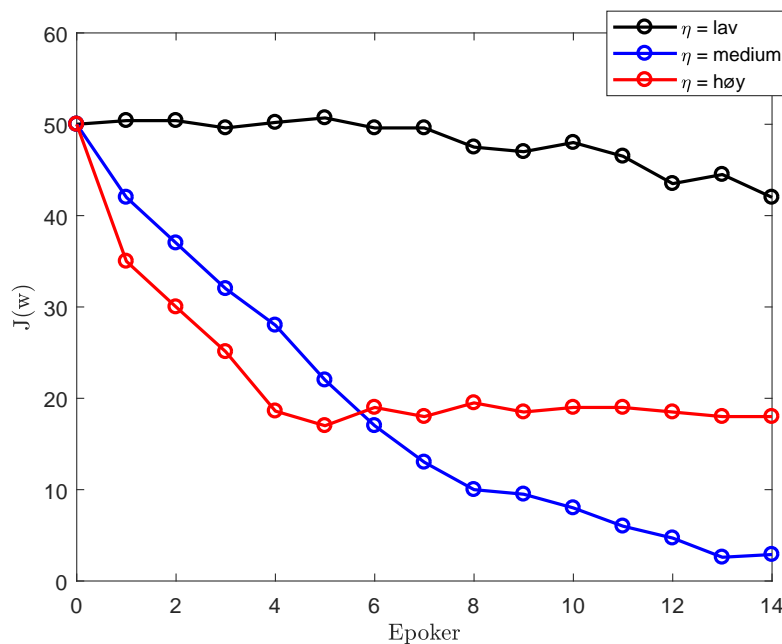
Når gradienten er beregnet benyttes en optimaliseringsalgoritme, for eksempel stokastisk gradient-nedstigning<sup>2</sup>(SGD), til å oppdatere vektene. Dette er vist i ligning (2.17), hvor vektene oppdateres i motsatt retning av den deriverte.  $\eta$  er læringsraten til nettverket og settes som en global parameter ved trening av modellen.

$$\underline{w}(i+1) = \underline{w}(i) - \eta \frac{\partial J}{\partial \underline{w}} \quad (2.17)$$

<sup>1</sup>Bruker gjennomsnitt av kvadrert feil. Engelsk: Minimum Mean Squared Error(MMSE).

<sup>2</sup>Engelsk: Stochastic Gradient Descent

Figur 2.7 visualiserer hvordan forskjellige verdi av  $\eta$  påvirker treningen. Y-aksen viser verdi for kost, mot epoke på x-akse, hvor en epoke betyr å trene nettverket med all treningsdata. For å unngå trening på all data samtidig, fordi en vanlig maskin ikke takler det, deles treningsdataen opp i mindre, tilfeldig sortert, parti. Antall epoker som skal benyttes velges basert på visualisering av trening, hvor kost plottes mot epoker, og velger antall epoker som gir lavest kost for testsett.



**Figur 2.7:** Figuren viser hvordan forskjellige læringsrater( $\eta$ ) påvirker ytelsen til nettverket. I y-akse vises kostverdien,  $J(w)$ . X-aksen viser epokenummer i treningen. Ser at lav  $\eta$  konvergerer tregt, i motsetning til høy  $\eta$  som konvergerer raskt, men ikke mot optimal verdi. Det er derfor viktig å velge riktig  $\eta$  til trening av nettverk.

I forbindelse med trening av et nevralt nettverk er det, i tillegg til  $\eta$ , flere globale parametre som må bestemmes. Disse er: partistørrelse<sup>1</sup>, antall skjulte lag<sup>2</sup>, antall noder i skjulte lag<sup>3</sup> og valg av aktiveringsfunksjon. Partistørrelse er en parameter for metoden som deler opp treningsdataen i mindre tilfeldige parti, hvor partistørrelse beskriver hvor stort hvert parti skal være. Denne velges basert på kapasiteten til maskinen som utfører treningen, men det er også mulig å eksperimentere med forskjellige partistørrelser. For å finne den optimale kombinasjonen av globale parametre må det utføres flere eksperimenter, hvor kost fra testsett overvåkes for forskjellige kombinasjoner[25].

<sup>1</sup>Engelsk: Batch

<sup>2</sup>Engelsk: Number of hidden layers

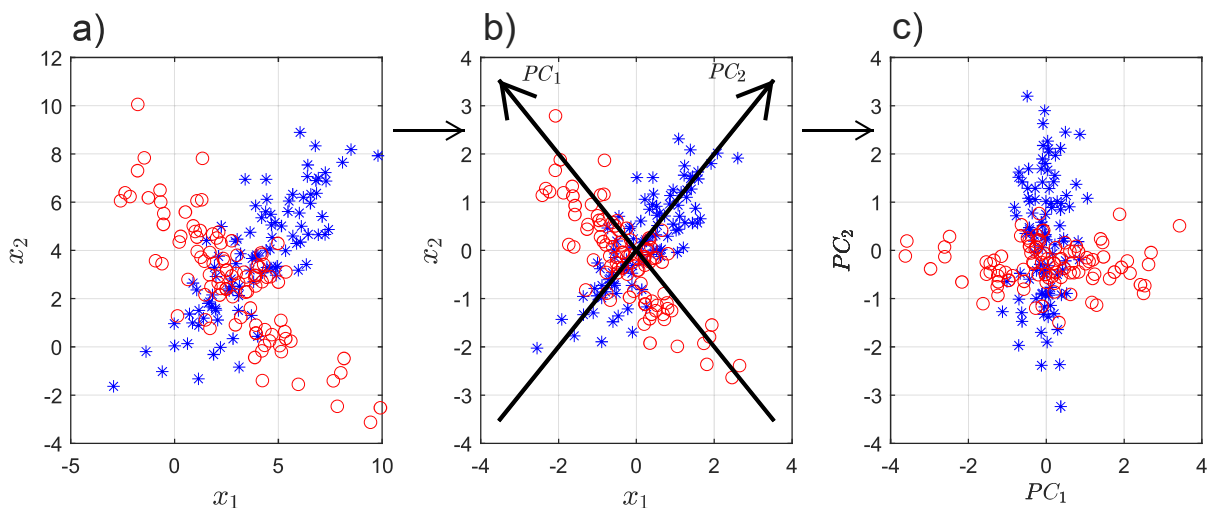
<sup>3</sup>Engelsk: Number of hidden nodes

## 2.2.4 Transformasjon av datasett

For å forenkle klassifiseringsproblemet benyttes en transformasjon på datasettet. Transformasjonen er basert på prinsipalkomponentanalyse (PCA), hvor det defineres nye egenskaper basert på treningsettet. Egenskapene blir definert i de retningene treningsettet har høyest variasjon. For å finne disse retningene og transformere datasett utføres 4 trinn, oppsummert i listen under, og visualisert i figur 2.8.

1. Normaliser data (**a** → **b** i figur 2.8)
2. Beregn kovariansmatrise til treningsett
3. Beregn egenvektorer og eigenverdier til kovariansmatrise. Egenvektorene er plottet i figur 2.8**b**), vist som sorte piler.
4. Velg hvor mange dimensjoner som skal benyttes og transformert datasett ved å høyremultiplisere egenvektorer med datasett (**b** → **c** i figur 2.8. Her er det transformert til like mange dimensjoner som i opprinnelig datasett.)

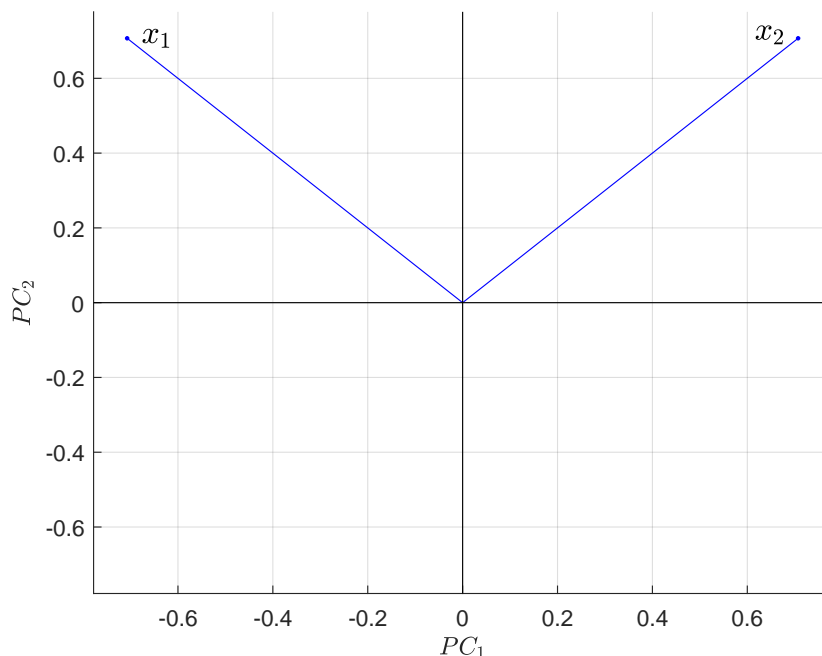
En viktig egenskap med denne transformasjonen er at det projiserte datasettet har ukorrelerte egenskaper[25]. For å visualisere en transformasjon er det generert 200 datapunkter med 2 dimensjoner. Datasettet er fordelt med like mange datapunkter i 2 klasser. Transformasjonen er visualisert i figur 2.8.



**Figur 2.8:** Figuren visualiserer fremgangsmåte ved PCA transformasjon. **a** til **b**: Normaliserer data. **b**) Egenvektorer beregnes for kovariansmatrise til treningsettet. Disse er plottet som sorte piler. **b** til **c**: Datasettet transformeres ved å høyremultiplisere egenvektorer med opprinnelig datasett.

Eigenvektorene som beregnes definerer de nye egenskapene, og tolkes ikke lengre som tidligere egenskapsnavn. For å analysere transformasjonen, benyttes et biplot.

Det viser sammenheng mellom nye ( $PC_1$  og  $PC_2$ ) og gamle ( $x_1$  og  $x_2$ ) egenskaper. Et biplot viser hvor mye en tidligere egenskap påvirker de nye egenskapene. De gamle egenskapene blir plottet som vektorer, hvor lengden på vektoren indikerer hvor mye den påvirker de nye egenskapene. Et biplot av transformasjonseksempelet ovenfor er vist i figur 2.9.



**Figur 2.9:** Biplot av transformasjon. Figuren viser sammenheng mellom de prinsipielle komponentene,  $PC_1$  og  $PC_2$  og tidligere egenskaper,  $x_1$  og  $x_2$ .

Figur 2.9 viser at de to tidligere egenskapene,  $x_1$  og  $x_2$ , påvirker den første prinsipielle komponenten,  $PC_1$ , i forskjellig retning. Det viser at positiv verdi for  $PC_1$  skyldes negativ  $x_1$  og positiv  $x_2$ . Denne informasjonen brukes til å koble transformert datasett til de opprinnelige egenskapene.



# Kapittel 3

## Material og metode

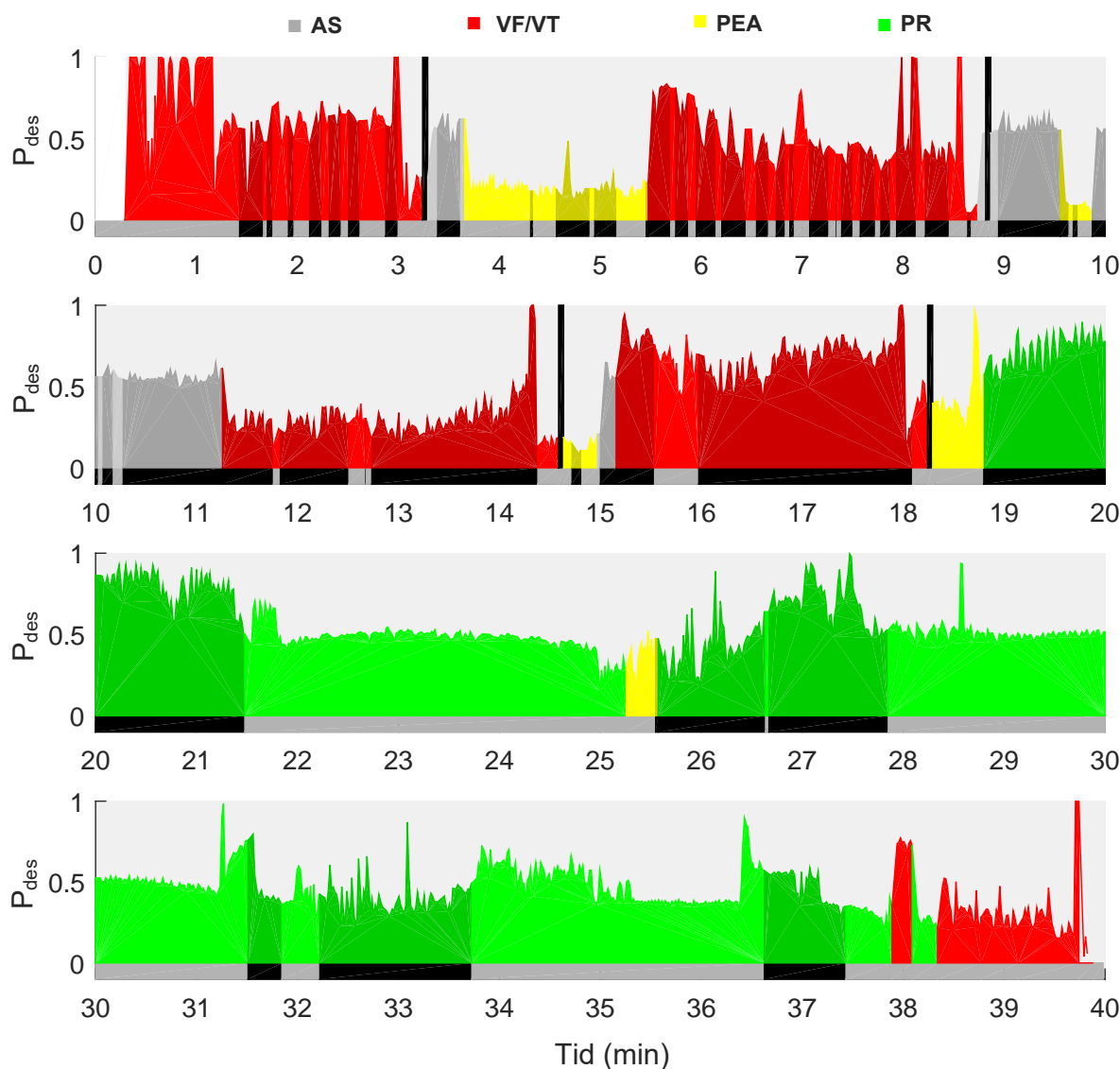
Dette kapitlet beskriver først datasettet. Deretter beskrives fremgangsmåte for å hente ut relevante datapunkt og databehandling av disse datapunktene. Til slutt forklares fremgangsmåte for utførte eksperimenter, og hvordan resultat fra eksperiment evalueres.

### 3.1 Datasett

Datasettet består av 394 episoder med hjertestans utenfor sykehus. Data ble samlet i forbindelse med en studie som undersøkte kvalitet av hjertelungeredning, utført av helsepersonell i ambulansse[6, 27]. Studien ble utført i tidsrommet mars 2002 til september 2004, i 3 byer: Akershus (Norge), Stockholm (Sverige) og London (England). I hver by ble det benyttet 6 modifiserte defibrillatorer av typen HeartStart 4000 (Phillips medical systems, Andover, MA, USA). Disse ble modifisert med ekstra brystpute og trykksensor, som gjør det mulig å måle relativ akselerasjon. Defibrillatorene ble plassert i ambulanser og logget data for EKG, thoraximpedans, akselerasjon og kraft. Det ble også estimert kompresjonsdybde, basert på akselerasjon. Alle episoder er gjennomgått av eksperter som har annotert hjerterytme og hjertekompresjoner. Hjerterytmene er fordelt i 5 klasser: VF, VT, AS, PEA og PR.

Det er utført mye arbeid på datasettet, hvor det blant annet er utviklet automatisk klassifisering av hjerterytme[14] og hjertekompresjoner[7]. Det er disse automatiske rytme- og kompresjonssekvensannoteringene som benyttes i denne oppgaven, og kan dermed medføre eventuelle feilkilder i form av feilklassifisering. Det er også utviklet metode for å analysere hele episoder, hvor det beregnes sannsynlighet for endring til bedre tilstand[8]. Denne sannsynligheten kalles  $P_{des}$  og er utgangspunktet for inndeling i klasser. Figur 3.1 viser en episode hvor  $P_{des}$  er gitt i y-aksen, mot

tid i minutt på x-aksen. Visualiseringen av hjertestanseepisode er utviklet i forbindelse med utviklingen av  $P_{des}$ . De klassifiserte hjerterytmene visualiseres i form av fargekode. Rød farge tilsvarer VF/VT, gul PEA, grå AS og grønn er PR. Intervaller med hjertekompresjon indikeres med sort horisontalt felt i bunn av hvert delplott. Sort loddrett strek betyr at det utføres elektrisk sjokk med defibrillator.

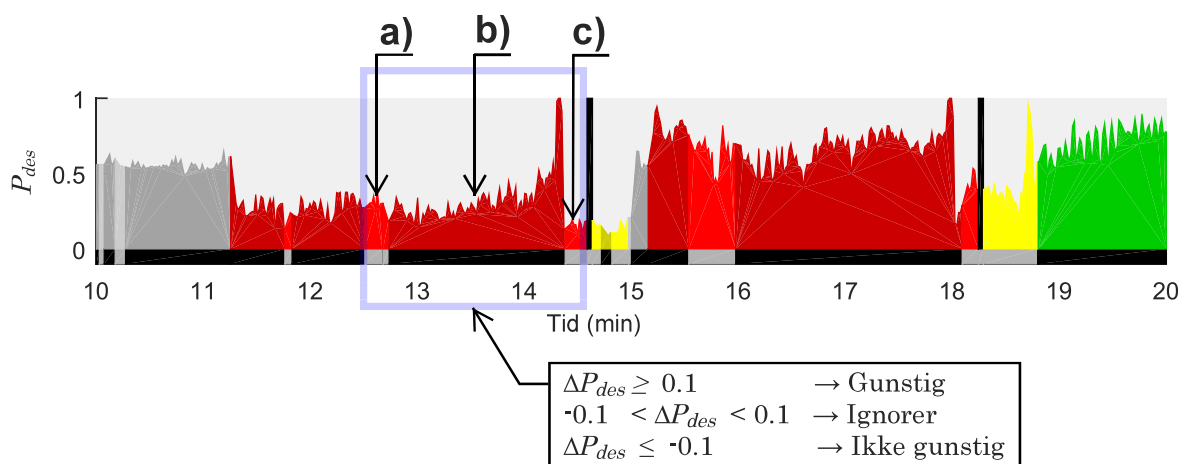


**Figur 3.1:** Oversiktsplott over en hjertestanseepisode, med tid(min) på x-akse og sannsynlighet for endring til bedre tilstand( $P_{des}$ ) på y-akse. Visualiseringen av hjertestanseepisode er utviklet i forbindelse med utviklingen av  $P_{des}$ [8], men er ikke forklart her. Hjerterytme indikeres med fargekode, vist over figur, sortert fra klinisk minst gunstig til mest gunstig. Grå: AS, Rød: VT/VE, Gul: PEA, Grønn: PR. Intervaller med hjertekompresjoner indikeres med sort felt i bunn av hvert delplott. Sort loddrett strek betyr at det er utført elektrisk sjokk med defibrillator.

Figur 3.1 viser dynamikk i episoden i form av rytmeendringer og endringer i  $P_{des}$ , og gjør det lettere å få oversikt over og å analysere hendelsen. Endringer i  $P_{des}$  benyttes for å hente ut relevante datapunkt, forklart i kapittel 3.2.

## 3.2 Hente relevante datapunkt

For å hente ut datapunkt som indikerer kvalitet av hjertekompresjoner, benyttes kriterier knyttet til endringer i  $P_{des}$ . Ettersom  $P_{des}$  gir en indikasjon på pasientens tilstand, vil dermed endringer i  $P_{des}$  gi en indikasjon på fysiologisk respons. Figur 3.2 visualiserer hvordan datapunkt fra et kompresjonsintervall hentes ut og sorteres i gruppene gunstig/ikke gunstig behandling, avhengig av endring i pasientens tilstand. Her deles episoden opp i sekvenser, hvor en endring i  $P_{des}$  er knyttet til to sekvenser. En ny sekvens oppstår dersom det forekommer en ny hjerterytme, start/slutt på hjertekompresjoner eller defibrillering. Kompakt representasjon av de forskjellige sekvensene er utviklet i tidligere studier[28, 29], og benyttes her til å finne kompresjonsintervall i datasett. I figur 3.2 viser **a)** en sekvens med hjerterytme VF/VT, uten hjertekompresjoner. For å beskrive endring i pasientens tilstand, defineres  $\Delta P_{des}$ . Den defineres som  $\Delta P_{des} = P_{des2} - P_{des1}$ , hvor  $P_{des1}$  og  $P_{des2}$  er gjennomsnittsverdier for to forskjellige sekvenser.  $\Delta P_{des}$  verdi lik 0.1 betraktes som økning i sannsynlighet, med 10%, for endring til bedre tilstand. Dersom det er tilfellet under et kompresjonsintervall, merkes det som gunstig behandling.



**Figur 3.2:** Utklipp fra figur 3.1. Viser hvordan  $\Delta P_{des}$  brukes for å hente ut og merke relevante datapunkt fra episode. Dette danner grunnlaget for datasettene i denne oppgaven. Figuren viser hvordan episoden deles opp i sekvenser, hvor det i den blå firkanten er 3 sekvenser, **a)**, **b)** og **c)**. **a)** og **c)**: Sekvens med hjerterytme VF/VT, uten hjertekompresjoner. **b)**: Sekvens med hjerterytme VF/VT, med hjertekompresjoner. For å definere  $\Delta P_{des}$  benyttes kombinasjoner av disse sekvensene.

For å bestemme hvilke sekvenser  $P_{des1}$  og  $P_{des2}$  skal bestemmes ut fra, finnes det flere alternativ. Valg av alternativ er avgjørende for om datapunkt merkes som gunstig eller ikke gunstig. Kompresjonsintervallet i figur 3.2, markert med blå ramme, kan enten settes som gunstig eller ikke gunstig, avhengig av hvilke sekvenser som benyttes. Intervallet blir satt som ikke gunstig dersom  $P_{des1}$  beregnes i sekvensen før

kompresjonsintervallet, merket med **a**), og  $P_{des2}$  beregnes i sekvensen etter, merket med **c**). Dersom  $P_{des}$ -verdi før intervall og  $P_{des}$ -verdi i kompresjonsintervall benyttes, henholdsvis **a**) og **b**) i figur 3.2, blir intervall markert som gunstig. For å finne hvilke valg av sekvenser som korrelerer mest med kvalitet av hjertekompresjoner, er det utviklet 3 metoder for å hente ut datapunkter. Hver metode benytter forskjellige sekvenser for  $P_{des1}$  og  $P_{des2}$ , og danner grunnlag for datasettene benyttet i denne oppgaven. Det er viktig at sekvensene benyttet har samme hjerterytme. Dette er fordi  $P_{des}$  beregnes forskjellig, avhengig av hjerterytme. I tillegg medfører hjertekompresjoner støy på EKG-signalet. Signalet blir derfor filtrert under hjertekompresjoner, og beregning av  $P_{des}$  kan derfor bli påvirket. Dette medfører en usikkerhet i  $P_{des}$ -verdi i overganger fra hjerterytme med og uten hjertekompresjoner. Et annet usikkerhetsmoment kan observeres i figur 3.1, som viser oversikt over en hjertestanseepisode, hvor  $P_{des}$  synker betraktelig før det utføres sjokk med defibrilator, merket med sort vertikal strek. Dette kan også være en potensiell feilkilde til merking av datapunkt. De forskjellige metodene for å hente ut datapunkt er forklart i delkapitlene 3.2.1-3.2.3.

### 3.2.1 Metode 1

Denne metoden bruker bare  $P_{des}$ -verdier i kompresjonsintervall, merket med **b**), hvor sekvensen deles opp i to deler.  $P_{des1}$  er gjennomsnittet av  $P_{des}$  til den første halvparten av kompresjonsintervallet.  $P_{des2}$  er gjennomsnittet av  $P_{des}$  for den andre halvparten av kompresjonsintervallet. I tilfellet vist i figur 3.2 blir datapunkt merket som gunstig kompresjonsintervall.

### 3.2.2 Metode 2

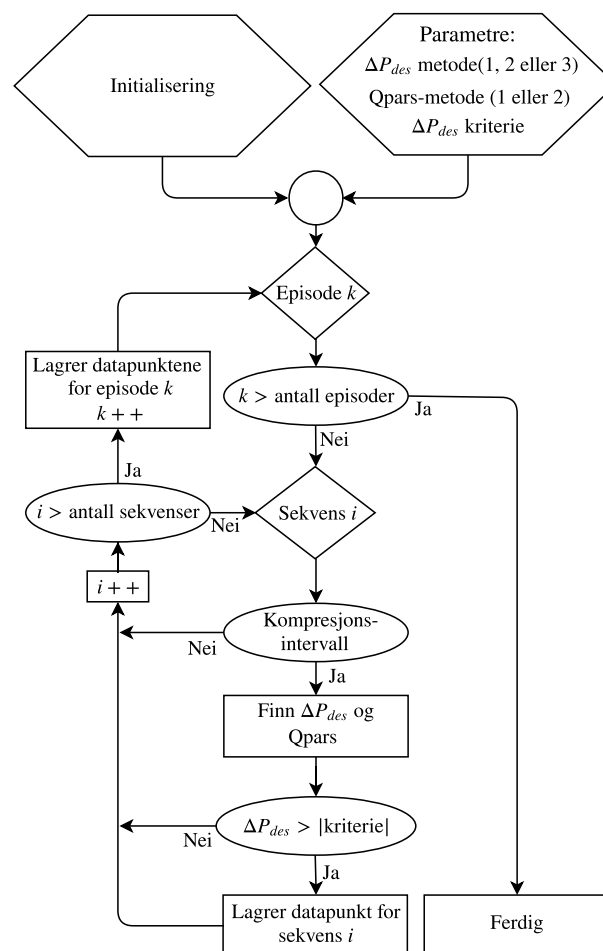
Denne metoden benytter sekvens før og etter hjertekompresjonene, henholdsvis **a**) og **c**) i figur 3.2.  $P_{des1}$  er gjennomsnittet av  $P_{des}$  i sekvensen før kompresjonsintervall, og  $P_{des2}$  er gjennomsnittet av  $P_{des}$  i sekvensen etter kompresjonsintervall. Denne metoden merker kompresjonsintervallet i figur 3.2 som ikke gunstig behandling. Dette er samme metode som benyttet i [8].

### 3.2.3 Metode 3

I denne metoden beregnes  $P_{des1}$  i sekvensen før kompresjonsintervallet, merket med **a**), og  $P_{des2}$  beregnes i kompresjonsintervallet, merket med **b**). Dersom sekvensen før kompresjonene er lengre enn 1 minutt, deles sekvensen i to, slik at bare den siste halvparten av sekvensen benyttes til å beregne  $P_{des1}$ . Formålet med det er å unngå at  $P_{des}$ -verdier lenge før kompresjonsintervallet skal påvirke  $\Delta P_{des}$ , som indikerer behandlingen. Denne metoden merker kompresjonsintervallet i figur 3.2 som gunstig.

### 3.2.4 Algoritme

Datapunkt fra kompresjonsintervall er hentet ut ved bruk av alle metoder. Alle datapunkt består av kvalitetsparametre som beskriver egenskapene for hjertekompresjoner, beregnet fra kompresjonsdybde og impedansmåling, benyttet i en tidligere studie[7]. Kvalitetsparametrene blir beregnet fra et vindu på 3 sekund, hvor vinduet flyttes mot høyre for hvert sekund. Det betyr at for et kompresjonsintervall som varer i 15 sekund, vil det beregnes 19 datapunkt med kvalitetsparametre. Fordi kvaliteten av behandling kan variere i et intervall, er det eksperimentert med to metoder for hvordan kvalitetsparametre skal hentes ut. Den første metoden, Qpars1, tar gjennomsnittet av alle kvalitetsparametre i kompresjonsintervallet, og indikerer dermed den gjennomsnittlige behandlingen i intervallet. Dette går på bekostning av antall datapunkt hentet. Den andre metoden, Qpars2, henter ut alle datapunktene i kompresjonsintervallet. Figur 3.3 viser et flytskjema for hvordan datapunkt hentes ut fra datasett.



**Figur 3.3:** Flytskjema for hvordan datapunkt hentes ut fra datasett. Rektangel representerer en funksjon, skrå firkant er for-loop og ellipse representerer sjekk av betingelse. Programmet er implementert i Matlab.

I tabell 3.1 oppsummeres antall datapunkt hentet ut for hver metode. Fordi det er 3 metoder for  $\Delta P_{des}$  og 2 metoder for kvalitetsparametre, resulterer det i totalt 6 datasett. Datasettene nummereres fra 1 til 6, hvor datasett 1 til 3 bruker Qpars1, med henholdsvis metode 1 til 3. Tilsvarende er datasett 4 til 6 hentet med bruk av Qpars2, med henholdsvis metode 1 til 3.

**Tabell 3.1:** Tabellen viser antall datapunkt samlet ved bruk av forskjellige metoder for  $\Delta P_{des}$  og kvalitetsparametre. Qpars1 tar gjennomsnitt av alle datapunkt i kompresjonsintervall, og viser dermed også hvor mange kompresjonsintervall det er hentet datapunkt fra. Qpars2 bruker alle datapunkt i kompresjonsintervall, som dermed resulterer i datasett med flere datapunkt. Nummerering av datasett er gitt i siste kolonne, sammen med antall datapunkt i datasett, markert som tall i parentes.

| Metode<br>$\Delta P_{des}$ | Qpars1  |              | Qpars2  |              | Totalt<br>(Datasett) |
|----------------------------|---------|--------------|---------|--------------|----------------------|
|                            | Gunstig | Ikke gunstig | Gunstig | Ikke Gunstig |                      |
| 1                          | 311     | 225          | 9329    | 6305         | 536(1)/15634(4)      |
| 2                          | 348     | 342          | 8014    | 6821         | 690(2)/14835(5)      |
| 3                          | 467     | 354          | 8702    | 5051         | 821(3)/13753(6)      |

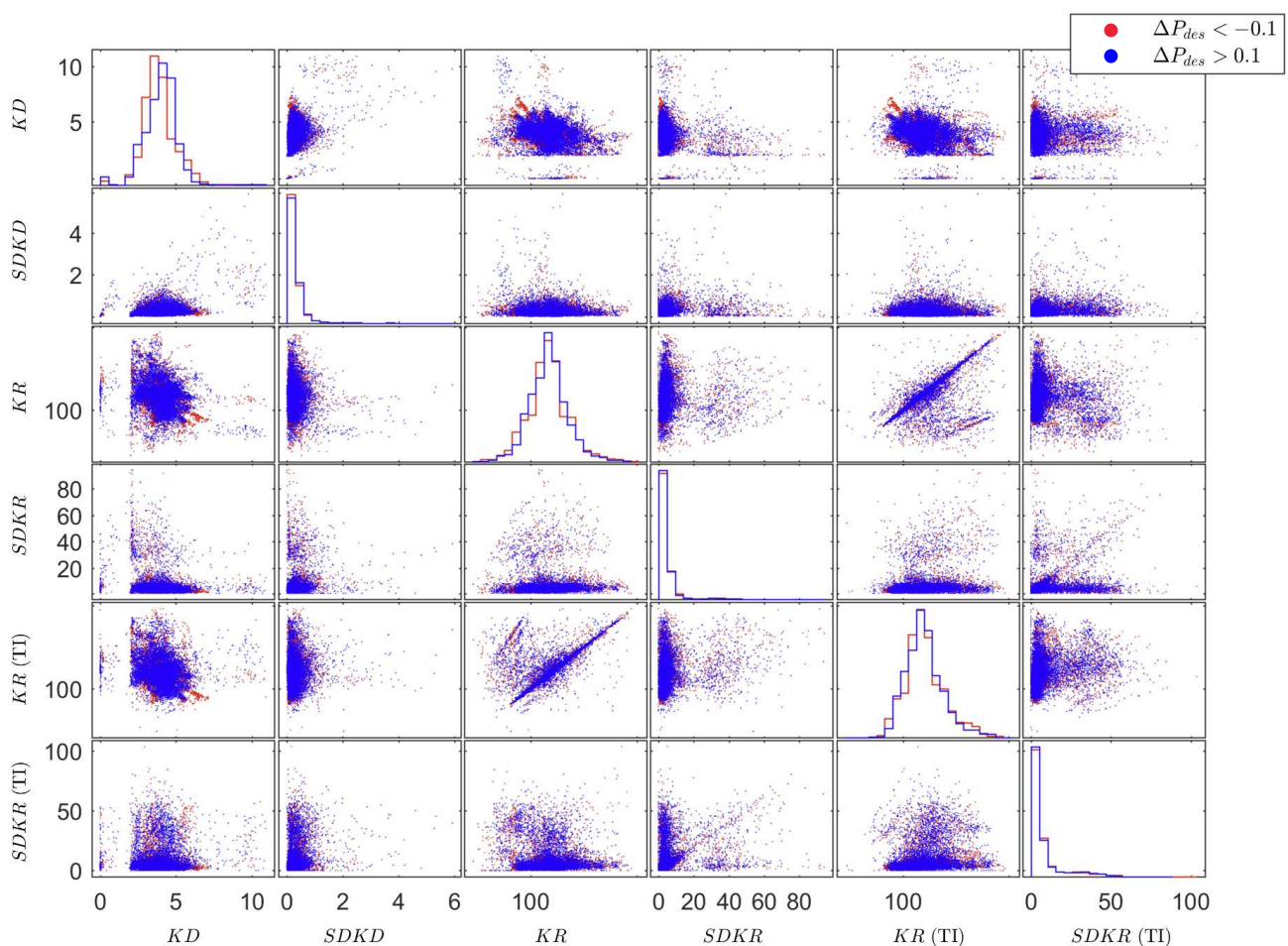
Kvalitetsparametrene består av totalt 15 egenskaper, hvor 6 av disse er benyttet. Dette fordi 9 av egenskapene ikke kan benyttes til sanntid klassifisering da de baserer seg på kunnskap om hele episodeforløpet. Alle kvalitetsparametre benyttet er oppsummert i tabell 3.2, hvor gjennomsnittsverdi er gitt sammen med 95% konfidensintervall i parentesene under.

**Tabell 3.2:** Tabellen viser gjennomsnitt for alle kvalitetsparametre for de forskjellige metodene med Qpars2(datasett 4, 5 og 6). Parentes under gjennomsnittsverdi viser et 95% konfidensintervall for estimatet. Datapunktene for metodene er delt opp i 2 klasser, gunstig og ikke gunstig, basert på  $\Delta P_{des}$ . KD - Kompresjonsdybde(cm), SD - Standardavvik, KR - Kompresjonsrate(kompresjoner per minutt), (TI) - Impedansmåling (thoraximpedans).

| Egenskap   | Metode 1                     |                              | Metode 2                     |                              | Metode 3                     |                              |
|------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
|            | Gunstig                      | Ikke gunstig                 | Gunstig                      | Ikke Gunstig                 | Gunstig                      | Ikke Gunstig                 |
| KD         | <b>3.92</b><br>(3.9-3.947)   | <b>3.73</b><br>(3.711-3.754) | <b>3.88</b><br>(3.858-3.9)   | <b>3.76</b><br>(3.73-3.783)  | <b>3.97</b><br>(3.945-3.991) | <b>3.91</b><br>(3.877-3.938) |
| SD KD      | <b>0.29</b><br>(0.283-0.296) | <b>0.28</b><br>(0.274-0.287) | <b>0.25</b><br>(0.243-0.254) | <b>0.24</b><br>(0.232-0.243) | <b>0.28</b><br>(0.276-0.289) | <b>0.26</b><br>(0.257-0.271) |
| KR         | <b>116</b><br>(115.5-116.3)  | <b>113</b><br>(113-114.1)    | <b>117</b><br>(117.1-117.9)  | <b>119</b><br>(119-119.9)    | <b>116</b><br>(115.9-116.7)  | <b>116</b><br>(115.6-116.7)  |
| SD KR      | <b>5.53</b><br>(5.36-5.701)  | <b>5.69</b><br>(5.477-5.904) | <b>5.34</b><br>(5.153-5.531) | <b>5.53</b><br>(5.31-5.748)  | <b>5.07</b><br>(4.904-5.232) | <b>5.57</b><br>(5.314-5.822) |
| KR (TI)    | <b>121</b><br>(120.3-121.1)  | <b>120</b><br>(120.1-121.1)  | <b>120</b><br>(119.7-120.5)  | <b>124</b><br>(123.5-125.4)  | <b>120</b><br>(119.7-120.4)  | <b>120</b><br>(120-121.1)    |
| SD KR (TI) | <b>9.08</b><br>(8.839-9.325) | <b>9.58</b><br>(9.286-9.882) | <b>7.68</b><br>(7.446-7.916) | <b>7.72</b><br>(7.465-7.979) | <b>8.18</b><br>(7.949-8.416) | <b>8.26</b><br>(7.949-8.575) |

Retningslinjene for HLR tilsier at optimal kompresjonsdybde er rundt 5 cm, men ikke over 6 cm, med kompresjonsrate mellom 100 til 120 kompresjoner per minutt[17]. Ved å sammenligne gjennomsnittsverdier for kompresjonsdybde(KD), i tabell 3.2, mot retningslinjer, kommer det frem at datasett 6 har et gjennomsnitt på gunstig behandling nærmest retningslinjene. Fra tabellen observere det også at alle datasettene har en gjennomsnittlig kompresjonsrate(KR), for gunstig behandling, mellom 100 og 120 kompresjoner per minutt. Tabell 3.2 viser at det er liten forskjell mellom verdiene for gunstig og ikke gunstig behandling, noe som gir en indikasjon om at det er vanskelig å skille mellom klassene.

For å undersøke fordelingen av datapunkt for hver klasse, benyttes et spredningsplott. Dette er vist i figur 3.4, hvor alle kvalitetsparametre fra datasett 6 er plottet mot hverandre. Sammenlignet med tabell 3.2, er gjennomsnittsverdier for datasett 6 vist i de to siste kolonnene. I figur 3.4 er gunstig og ikke gunstig behandling merket med henholdsvis blå og røde punkt.



**Figur 3.4:** Spredningsplott av datapunkt fra datasett 6. Klassene gunstig og ikke gunstig behandling er merket med henholdsvis blå og røde punkt. Diagonalen viser fordelingen av datapunkt for kvalitetsparametrene. Observerer stor overlapp mellom klassene.

### 3.3 Databehandling

For å evaluere modell, deles datasett inn i trening- og testsett. Dette er forklart i delkapittel 3.3.1. Figur 3.4 viser at det er stor overlapp mellom klassene gunstig og ikke gunstig behandling. På grunn av stor overlapp utføres flere databehandlingsteknikker, med formål om å skille klassene bedre. Disse teknikkene forklares i delkapittel 3.3.2.

#### 3.3.1 Inndeling i trening- og testsett

Ved inndeling i trening- og testsett er det flere ting å ta hensyn til. Først og fremst må krysskontaminering unngås. Krysskontaminering er når datapunkter fra minst én episode er fordelt til både trening- og testsettet. Det medfører at datapunkt i treningsettet, indirekte har informasjon om datapunkt i testsettet, som igjen medfører overvurdering av ytelse[30]. Trening- og testsett må også ha samme forhold mellom klassene. For eksempel dersom 60% av treningsettet består av klasse 1, så må også 60% av testsettet bestå av klasse 1. Siste betingelse ved inndeling i trening- og testsett er at fordelingen skal være tilfeldig. Dette er utført i den grad det er mulig å få det tilfeldig, basert på hensyn som tas.

Det ble generert en vektor med alle tall fra 1 til 394, tilfeldig sortert, som representerer alle episoder. Basert på denne vektoren ble 70% av episodene satt til trening og 30% til test. Fordi episodene inneholder forskjellige mengde datapunkt, ble det sjekket at antall datapunkt til trening var mellom 65% og 75% av totalt antall datapunkt. Det ble også sjekket om forhold mellom klassene i trening- og testsett var like. Dersom forholdene ikke var like, eller antall datapunkt i treningsett var utenfor intervallet, ble samme fremgangsmåte repetert med ny vektor. Dette ble utført for alle de 6 datasettene.

Det ble også hentet datapunkt til et utvidet testsett, hvor alle datapunkt fra alle kompresjonsintervall i episoder, satt av til testsett, ble hentet. Det ble utført for å undersøke hvordan metodene fordeler sannsynlighetsverdier mot kvalitetsparametre, uavhengig av  $\Delta P_{des}$ . Bruken av dette testsettet forklares nærmere i delkapittel 4.3, hvor fordeling av sannsynlighetsverdier mot verdier for kvalitetsparametre er vist. Denne fordelingen sammenlignes opp mot retningslinjene for HLR. Alle datapunkt i trening- og testsett, for alle datasett, er oppsummert i tabell 3.3. Siste kolonne viser antall datapunkt i utvidet testsett.



**Tabell 3.3:** Tabellen viser antall datapunkt fordelt til trening- og testsett, for alle datasett. I parentes under antall datapunkt er det vist mengde datapunkt for de forskjellige klassene, gunstig og ikke gunstig. Siste kolonne viser antall datapunkt i utvidet testsett, for alle datasett.

| Datasett | Treningssett                | Testsett                   | Utvidet testsett |
|----------|-----------------------------|----------------------------|------------------|
|          | (Gunstig/Ikke gunstig)      | (Gunstig/Ikke gunstig)     |                  |
| 1        | <b>356</b><br>(205/151)     | <b>180</b><br>(106/74)     | <b>43577</b>     |
| 2        | <b>471</b><br>(239/232)     | <b>219</b><br>(109/110)    | <b>43698</b>     |
| 3        | <b>592</b><br>(338/254)     | <b>229</b><br>(129/100)    | <b>43698</b>     |
| 4        | <b>10850</b><br>(6455/4395) | <b>4784</b><br>(2874/1910) | <b>43698</b>     |
| 5        | <b>10377</b><br>(5609/4768) | <b>4458</b><br>(2405/2053) | <b>36828</b>     |
| 6        | <b>9279</b><br>(5856/3423)  | <b>4474</b><br>(2846/1628) | <b>44123</b>     |

### 3.3.2 Databehandlingsteknikker

Teknikkene for databehandling er utført i samme rekkefølge som de presenteres under. For å unngå krysskontaminering mellom trening- og testsett er det tatt hensyn til at alle transformasjons- og normaliserings-parametre bestemmes ut fra treningsettet, og at disse parametrene benyttes til transformasjon av både trening- og testsett.

#### Sile treningsdata

Ved bruk av à priori kunnskap, baser på retningslinjene for HLR, siles treningsettet. Silingen tar utgangspunkt i datapunkt merket som gunstig behandling. Dette fordi en behandling ikke kan regnes som gunstig dersom det avviker betraktelig fra retningslinjene. Et betraktelig avvik er her definert som  $\pm 2$  cm for kompresjonsdybde, og  $\pm 20$  kompresjoner per minutt for kompresjonsrate. Det betyr at datapunkt i treningsettet, merket gunstig, med kompresjonsdybde utenfor intervallet 3-7 cm, eller kompresjonsrate utenfor intervallet 80-140 kompresjoner per minutt, fjernes fra treningsettet. Dermed blir også ekstremverdier for KD og KR, merket gunstig behandling og som kan skyldes feil på grunn av støy, fjernet fra treningsettet.

## Log-transformasjon

Fra figur 3.4 observeres det i diagonalen, som viser pdf for hver egenskap, at egenskapene SD KD, SD KR og SD KR (TI), ikke kan antas normalfordelt. Dette fordi de har en skjev fordeling. Det betyr at fordelingen av datapunktene ikke er symmetrisk over midten, hvor toppunktet er. Ved bruk av log-transformasjon, vist i ligning (3.1), kan denne skjeve fordelingen gjøres mindre skjev[31].

$$Y = \log(X + 1) \quad (3.1)$$

For å undersøke om transformert datasett kan antas å være normalfordelt, visualiseres pdf for egenskap<sup>1</sup>. Dette er vist i vedlegg A.1, hvor det observeres at datapunktene er mer normalfordelt, etter transformasjon.

## Normalisering

Normalisering kan forbedre ytelse til modell[32], og er derfor en viktig del av data-behandlingen. For å normalisere datasettet benyttes en transformasjon til standard normalfordeling. Dette er vist i ligning (3.2). Dersom  $X$  er en normalfordelt variabel,  $X \sim N(\mu, \sigma)$ , vil den transformerte variabelen  $Z$ , tilsvare normalfordeling med 0 i forventningsverdi, og 1 i standardavvik,  $Z \sim N(0, 1)$ .

$$Z = \frac{X - E(X)}{\sqrt{Var(X)}} = \frac{X - \mu}{\sigma} \quad (3.2)$$

For eksempel betyr en  $Z$ -verdi lik 2, at datapunkt er 2 standardavvik fra gjennomsnittet. Testdata normaliseres med  $\hat{\mu}$  og  $\hat{\sigma}$  estimert fra treningsdata, hvor  $\hat{\mu}$  og  $\hat{\sigma}$  er henholdsvis gjennomsnitt og standardavvik.

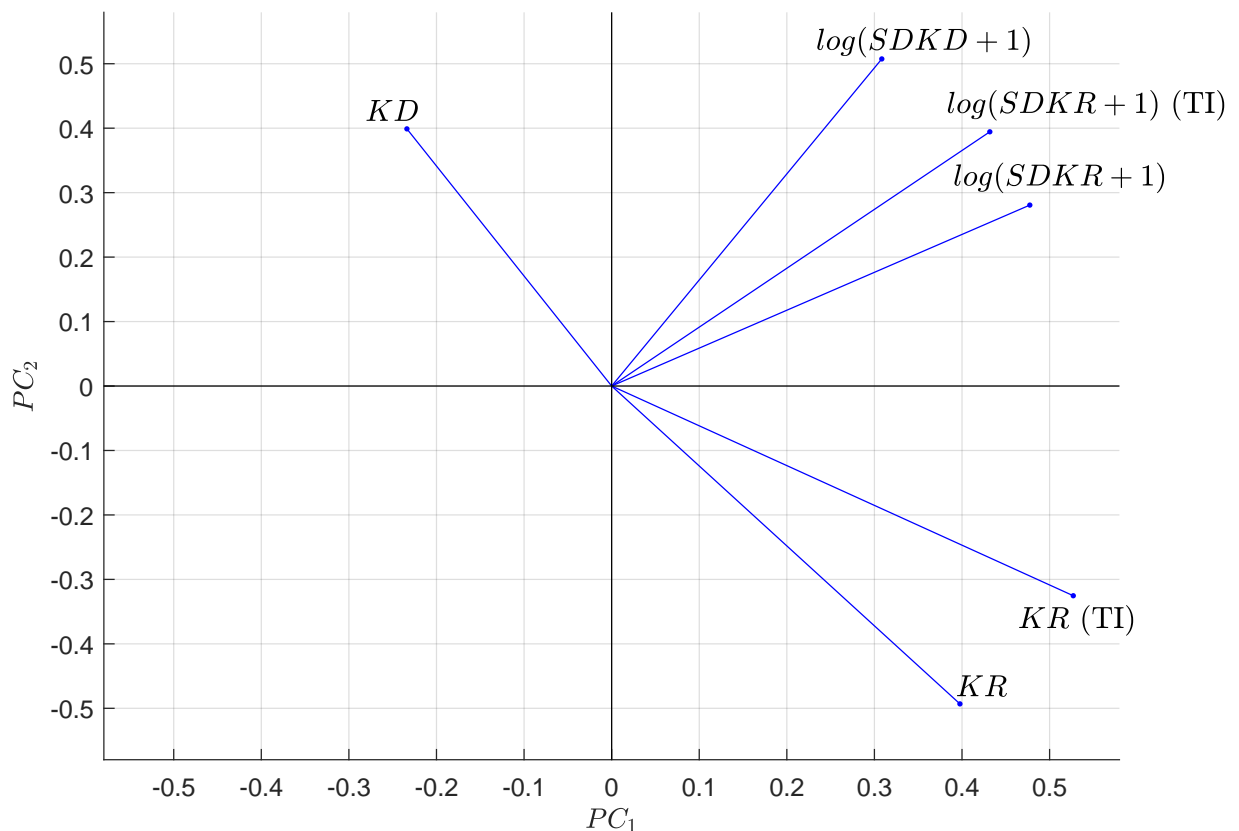
## Prinsipalkomponentanalyse (PCA)

Datasett er transformert med bruk av metoden beskrevet i kapittel 2.2.4. Eigenvektorer til kovariansmatrisen for treningsettet er beregnet, og brukes til transformasjon av trening- og testsett. Transformasjonen er vist i ligning (3.3), hvor  $v$  er matrise med eigenvektorene, og  $X$  er datasett som transformeres. Det er transformert til like mange dimensjoner som i opprinnelig datasett.  $Y$  er datasettet projisert på prinsipalaksene,  $PC_1$ - $PC_6$ .

$$Y = Xv \quad (3.3)$$

<sup>1</sup>En normalfordelingskurve er lett gjenkjennelig, og visualisering av pdf er derfor en enkel metode for å undersøke om datapunkt er normalfordelt.

Figur 3.5 viser de to nye egenskapene som representerer høyest varians i treningsettet,  $PC_1$  og  $PC_2$ . Sammenheng med tidligere egenskaper visualiseres som vektorer.

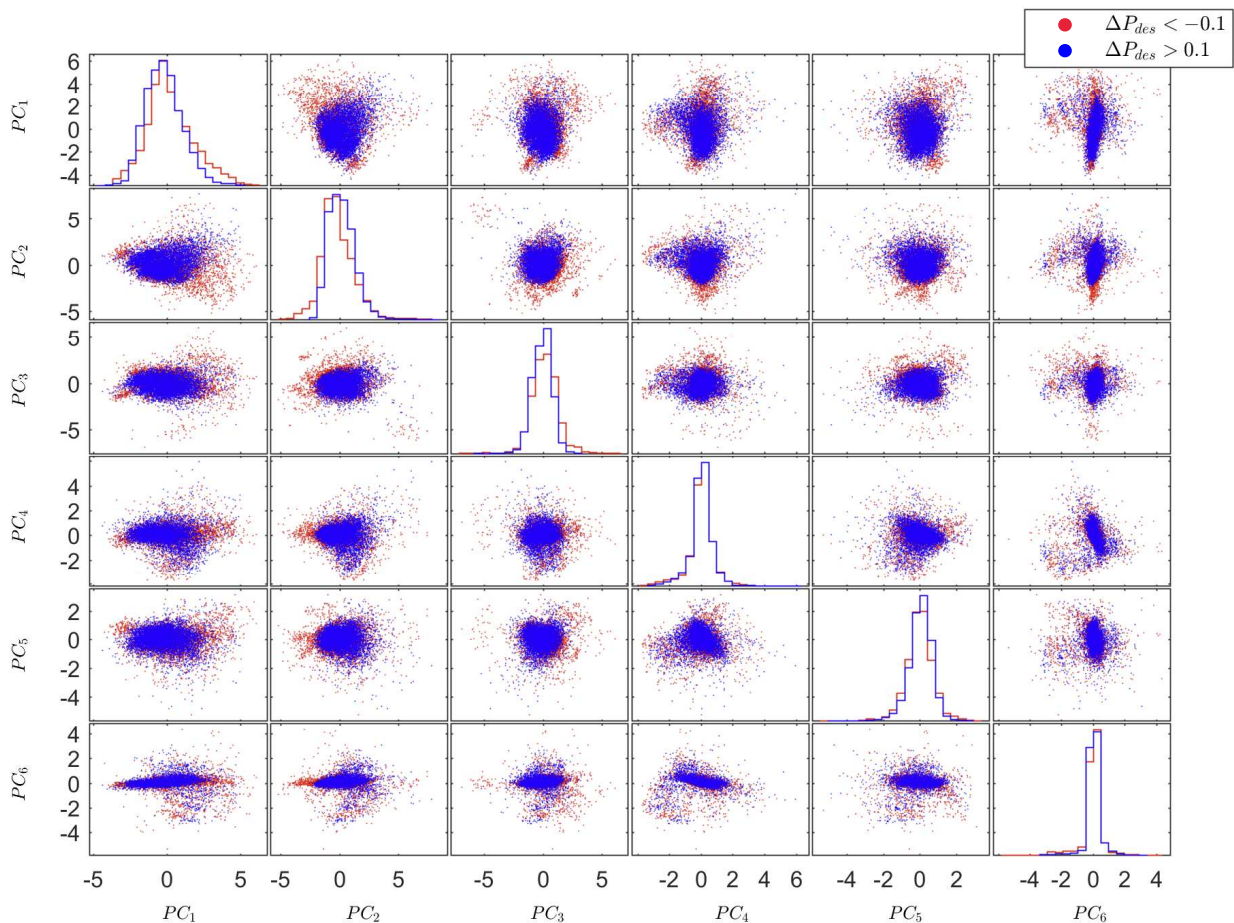


**Figur 3.5:** Analyse av transformasjon. Fordi datasett er normalisert, vil en  $PC$ -verdi lik 0 tilsvare gjennomsnittet for tidligere egenskap. Observerer at kompresjonsdybde( $KD$ ) og kompresjonsrate( $KR$ ) påvirker  $PC_1$  i motsatt retning. Det betyr at  $KR$  høyere enn gjennomsnitt, kombinert med  $KD$  lavere enn gjennomsnitt, vil gi en positiv  $PC_1$ -verdi, dersom  $SD$ -egenskaper har gjennomsnittsverdi, eller høyere. Dette vil også medføre en negativ  $PC_2$ -verdi.

Figur 3.5 viser at transformasjon mot  $PC_1$  og  $PC_2$  samler tidligere egenskaper som har store fellestrekk. For eksempel peker begge egenskapene for kompresjonsrate( $KR$ ) i samme retning. Fordi datasett er normalisert, vil en  $PC$ -verdi lik 0 tilsvare gjennomsnittet for tidligere egenskap. Fra figur 3.5 observeres det at kompresjonsdybde( $KD$ ) og kompresjonsrate( $KR$ ) påvirker  $PC_1$  i motsatt retning. Det betyr at  $KR$  høyere enn gjennomsnitt, kombinert med  $KD$  lavere enn gjennomsnitt, vil gi en positiv  $PC_1$ -verdi, dersom  $SD$ -egenskaper har gjennomsnittsverdi. Dette vil også medføre en negativ  $PC_2$ -verdi.

Transformasjonen av datasettet har medført at alle de nye egenskapene er ukorrelerte, hvilket bidrar til å gjøre egenskapene uavhengige. En ulempe med transformasjonen er at det ikke tas hensyn til hvilken klasse datapunkt er merket med,

som betyr at en slik transformasjon nødvendigvis ikke forenkler klassifiseringsproblemet. For å undersøke dette, brukes et spredningsplott av det transformerte treningsdatasettet, vist i figur 3.6. Figuren viser treningsett fra datasett 6, etter databehandling.



**Figur 3.6:** Spredningsplott av datapunkt fra datasett 6, etter databehandling. Eksempler på gunstig og ikke gunstig behandling er merket med henholdsvis blå og røde punkt. Diagonalen viser pdf for hver egenskap. Ved å sammenligne denne figuren med spredningsplott før databehandling, vist i figur 3.4, observeres det tydeligere skille mellom klassene, etter databehandling. Figuren viser også at det fortsatt er stor overlapp mellom klassene.

Etter transformasjon sorteres egenskaper etter hvor mye variasjon de representerer fra opprinnelig datasett. Dette gjøres ved bruk av eigenverdiene, beregnet fra kovariansmatrisen til treningsettet, hvor høyeste eigenverdi tilsvarer høyest variasjon. Ved å sammenligne spredningsplott før databehandling, vist i figur 3.4, med spredningsplott etter databehandling, observeres det tydeligere skille mellom klassene. Figur 3.6 viser også at det fortsatt er stor overlapp mellom klassene.

I figur 3.5 ble det observert at kompresjonsdybde( $KD$ ) og kompresjonsrate( $KR$ ) påvirker  $PC_1$  i motsatt retning. Figur 3.5 sammenlignes med datapunktene vist i

figur 3.6 rad 2, kolonne 1. Her observeres det en gruppering av ikke gunstig behandling nede til høyre, vist som røde punkt. Basert på analyse av transformasjon med bruk av biplott, vist i figur 3.5, kan denne grupperingen skyldes en kombinasjon av lav kompresjonsdybde og høy kompresjonsrate, dersom egenskapene for standardavvik er gjennomsnittlige for disse datapunktene.

## 3.4 Eksperiment

Det er utviklet 3 forskjellige sannsynlighetsmodeller i denne oppgaven, basert på forskjellige maskinlæringsmetoder. Alle modellene gir sannsynlighet for gunstige kompresjoner,  $P_{gc}$ , hvor datapunkt klassifiseres som gunstig behandling dersom  $P_{gc}$  er over 0,5. Det er også benyttet flere metoder for å hente ut datapunkt, som betyr at det er flere datasett å trene og teste modellene mot. Målet for eksperimentene er å finne optimal kombinasjon mellom sannsynlighetsmodell og datasett. Optimal i den forstand at sannsynlighetsmodell klarer å skille mellom gunstig og ikke gunstig behandling, samtidig som fordeling av sannsynlighetsverdier, mot kvalitetsparametre, er i trå med retningslinjer for HLR.

### 3.4.1 ML-metoden

Eksperiment for ML-metoden er utført i Matlab, hvor ligningene beskrevet i kapittel 2.2 er implementert. Det er laget sannsynlighetsmodell for alle datasett, basert på treningsettet. Trening av modell består av å estimere parametrene  $\hat{\mu}_i$  og  $\hat{\Sigma}_i$ , henholdsvis gjennomsnitt og kovarians for klasse  $i$ . Etter trening ble trening- og testdatasett klassifisert og sammenlignet mot sann klasse. For å verifisere at alle databehandlingsteknikkene bidro til forbedring av sannsynlighetsmodell, ble alle kombinasjoner av databehandling testet. Det gir totalt 16 eksperiment per datasett. Klassifiseringsresultat mot testsett, fra eksperimentet som ga best resultat, er gitt i delkapittel 4.1.

Ved utføring av eksperiment, med forskjellige kombinasjoner av databehandlingsteknikker, ble det oppdaget at siling av treningsett var avgjørende for å oppnå modell som fordelte sannsynlighetsverdier, mot kvalitetsparametre, med rimelig sammenheng med retningslinjene for HLR. Et eksempel hvor sannsynlighetsmodell er trent med og uten siling av treningsett, er vist i vedlegg A.2. Her kan det observeres at sannsynlighetsmodell, trent med datasett hvor det ikke er utført siling av treningsett, fordeler sannsynlighetsverdier rundt 0,6, uavhengig av verdi for kvalitetsparametre.

### 3.4.2 Logistisk regresjon

Eksperiment for logistisk regresjon ble utført i Matlab. I likhet med eksperiment for ML-metoden, ble det laget sannsynlighetsmodell for alle datasett, samt kombinasjoner av databehandlingsteknikker.

For flere kvalitetsparametre, som for eksempel  $KD$ , er det ønskelig å ha en modell som gir høy sannsynlighetsverdi for gunstige kompresjoner rundt retningslinjene for HLR. Dersom hjertekompresjoner utføres med avvik fra retningslinjer, er det ønskelig med lav sannsynlighetsverdi. Dette gir en ulineær funksjon, og det er derfor viktig å legge til kvadrerte egenskaper. De nye egenskapene som ble inkludert var kvadrerte egenskaper av  $KD$ ,  $KR$  og  $KR(TI)$ . De kvadrerte egenskapene ble inkludert i datasettet før utførelse av normalisering, etter log-transformasjon. For kvalitetsparametrene basert på standardavvik( $SD$ ) er det ikke ønskelig med ekstremalpunkt i den tilnærmede funksjonen, hvor det er rimelig å forvente at lav  $SD$  gir høy sannsynlighet, og høy  $SD$  gir lav sannsynlighet. Disse egenskapene ble derfor ikke kvadrert.

Kostfunksjonen, beskrevet i delkapittel 2.2.2, ble implementert i Matlab og benyttet for å estimere parametrene i  $\theta$ . Estimeringsprosessen tilsvarer trening av modell, hvor matlabfunksjonen *fminunc* ble benyttet til å minimere kostfunksjonen. Parametrene for denne funksjonen er optimaliseringsalgoritme og maks iterasjoner, hvor *quasi-newton*[22, 23] algoritmen ble benyttet som optimaliseringsalgoritme. Maks iterasjoner ble satt lik 1000.

### 3.4.3 Nevral nettverk

Eksperiment for nevralt nettverk ble utført med bruk av Matlab og Python. Fordi et nevralt nettverk krever en stor mengde datapunkt til trening, ble bare datasett nummer 4, 5 og 6 benyttet. Databehandling ble utført i Matlab, hvor alle databehandlingsteknikker ble benyttet. Det var fordi det ga best resultat for de andre metodene, og trening av nevralt nettverk er en tidkrevende prosess.

I tillegg til databehandlingsteknikkene forklart i kapittel 3.3.2, var det også nødvendig å endre på oppsettet av vektoren som markerer sann klasse for datapunktene. Det er fordi klassifiseringsproblemet er et toklasseproblem, og for å oppnå sannsynlighetsverdi på hver utgang må det derfor benyttes like mange utganger som klasser. Det betyr at vektoren som indikerer sann klasse må ha samme dimensjon som antall utganger. Derfor utvides vektoren fra 1 dimensjon til 2. Dette er vist i

ligning (3.4), hvor vektoren til venstre indikerer hvilken klasse datapunkt er merket med. Rader i matrisen til høyre tolkes som sannsynligheter for at datapunkt tilhører klasse 0 og 1.

$$\underline{t} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \rightarrow \underline{t} = \begin{array}{c} \text{Klasse} \\ \begin{matrix} 0 & 1 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{matrix} \end{array} \quad (3.4)$$

Trening av nevrale nettverk ble utført i Python, med bruk av tilleggspakken TensorFlow [33], utviklet av Google. Tilleggspakken forenkler prosessen med å implementere og trene nevrale nettverk. Ved trening av et nevralt nettverk er det flere globale parametre som må bestemmes. Dette er parametre som læringsrate, partistørrelse, antall skjulte lag og antall noder i skjulte lag. For å finne parametre ble det utført flere eksperiment over et spekter av globale parametre, hvor nettverk ble trent med bruk av alle kombinasjoner. Ved å studere klassifiseringsresultat for alle nevrale nettverk, velges den kombinasjonen av parametre som ga best resultat. Listen under viser hvilke globale parametre som ble benyttet, og gir totalt 72 forskjellige kombinasjoner.

- Læringsrate = [0.00008, 0.00004, 0.00001]
- Noder = [8, 12, 16]
- Partistørrelse = [64, 128]
- Antall skjulte lag = [1, 2, 3, 4]

Læringsrater ble bestemt ved å finne grenseverdier. Det vil si verdier som nettverket trener sakte med, og verdier som overtrener nettverket. Læringsratene benyttet ligger mellom disse grenseverdiene. For å visualisere trening i form av treffprosent for hver epoke, ble TensorBoard[33] benyttet. Under trening genereres en log-fil, hvor parametre, treffprosent, kost og verdier for alle vektorer blir lagret. TensorBoard bruker denne filen til å visualisere ytelsen til nettverket. Fordi parametrene benyttet til trening bare består av noen utvalgte parametre, ble det utført små justeringer av de globale parametrene til det nettverket som presterte best.

For å studere fordeling av sannsynlighetsverdier fra modell basert på nevralt nettverk, mot kvalitetsparametre, ble Matlab benyttet. Sannsynlighetsmodell for eksperiment som ga best resultat ble lagret, og nettverkstruktur ble implementert i Python-filen *KlassifiserNN.py*. Matlab bruker denne Python-filen, hvor et datapunkt med kvalitetsparametre sendes inn, og sannsynlighet for at dette datapunktet er gunstige kompresjoner, returneres.

Ved sammenligning av sannsynlighetsverdier og kvalitetsparametre, ble det observert at valg av aktiveringsfunksjon var avgjørende for å oppnå en modell som fordele sannsynlighetsverdier i henhold til retningslinjer. Det ble først benyttet ReLU som aktiveringsfunksjon, noe som ga sannsynlighetsverdier som fulgte sigmoid-funksjonen. Det betyr at desto høyere verdi for kompresjonsdybde, desto høyere var sannsynligheten for gunstige kompresjoner. Ved å benytte sigmoid-funksjonen som aktiveringsfunksjon, ble forholdet mellom  $P_{gc}$  og kompresjonsdybde som en normalfordelingskurve med toppunkt rundt retningslinjene for HLR.

## 3.5 Evaluering av resultat

For å evaluere resultat fra eksperiment benyttes to vurderinger. Først undersøkes treffprosent, som gir en indikasjon på hvordan sannsynlighetsmodell presterer ved klassifisering. Treffprosent beskriver antall datapunkt riktig klassifisert, uavhengig av klasse. Dersom en modell klassifiserer med en treffprosent rundt 50%, vil den ikke kunne brukes, ettersom 50% for et toklasseproblem tilsvarer vilkårlig gjetting. Til slutt undersøkes fordeling av sannsynlighetsverdier mot kvalitetsparametre, og sammenlignes opp mot retningslinjer for HLR.

### 3.5.1 Klassifiseringsnøyaktighet

For å beregne treffprosent benyttes en forvirringsmatrise. Den viser fordeling mellom sann og predikert klasse. Tabell 3.4 viser et generelt oppsett av en forvirringsmatrise for et toklasseproblem, hvor beregning av treffprosent er gitt under tabell.  $N_{i,j}$  er antall datapunkt klassifisert som  $j$ , hvor  $i$  er sann klasse.



**Tabell 3.4:** Generelt oppsett av en forvirringsmatrise for et toklasseproblem.  $N_{i,j}$  er antall datapunkt klassifisert som  $j$ , hvor  $i$  er sann klasse. Total antall datapunkt er gitt av  $N_{tot}$ .  $T_0$  og  $T_1$  er antall datapunkt for henholdsvis sann klasse 0 og 1. Tilsvarende er  $P_0$  og  $P_1$  antall datapunkt predikert som henholdsvis klasse 0 og 1.

|          |   | Predikert |           | $\Sigma$  |
|----------|---|-----------|-----------|-----------|
|          |   | 0         | 1         |           |
| Sann     | 0 | $N_{0,0}$ | $N_{0,1}$ | $T_0$     |
|          | 1 | $N_{1,0}$ | $N_{1,1}$ | $T_1$     |
| $\Sigma$ |   | $P_0$     | $P_1$     | $N_{tot}$ |

$$Treffprosent = \frac{N_{0,0} + N_{1,1}}{N_{tot}} \cdot 100\%$$

I tillegg til treffprosent er det ønskelig å undersøke klassespesifikk ytelse til sannsynlighetsmodellene. Til dette benyttes fire verdier: sensitivitet, spesifisitet, positiv prediksjon og negativ prediksjon[34], og forkortes henholdsvis som Sens, Spes, PPV og NPV. Det gir en kompakt oppsummering av informasjon fra forvirringsmatrisen. Beregning av disse verdiene er vist i ligning (3.5).

$$Sens = \frac{N_{0,0}}{T_0}, \quad Spes = \frac{N_{1,1}}{T_1}, \quad PPV = \frac{N_{0,0}}{P_0}, \quad NPV = \frac{N_{1,1}}{P_1} \quad (3.5)$$

Sensitivitet er definert som hvor stor andel av datapunkt av sann klasse 0 som klassifiseres korrekt. Spesifisitet er tilsvarende, men for klasse 1. Positiv og negativ prediksjon er forhold mellom antall datapunkt korrekt klassifisert, for henholdsvis klasse 0 og 1, mot antall datapunkt predikert i tilsvarende klasse.

### 3.5.2 Vurdering mot retningslinjer for HLR

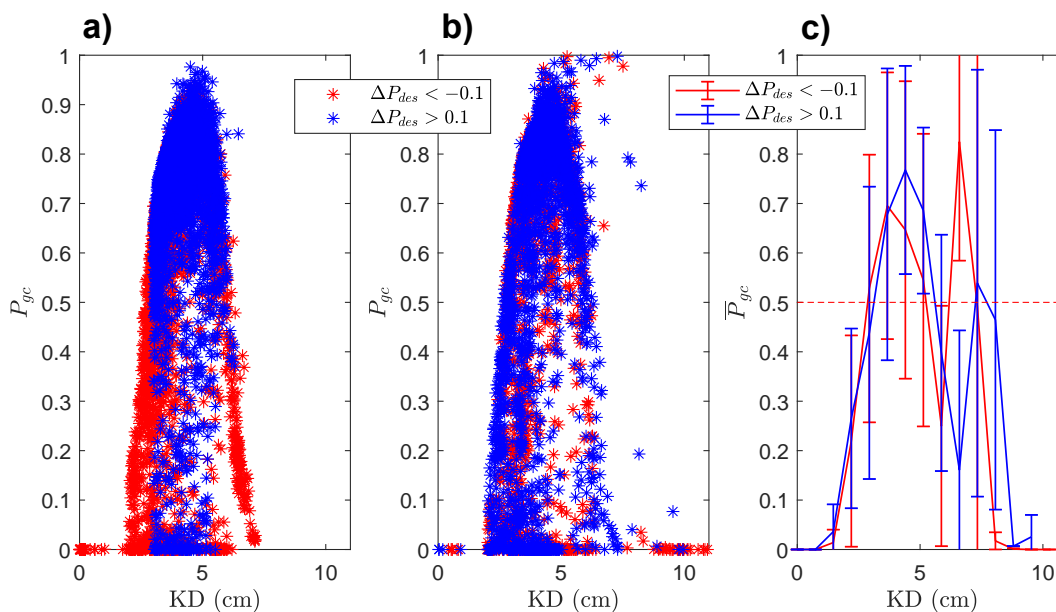
Endelig evaluering utføres ved å studere fordeling av sannsynlighetsverdier for gunstige kompresjoner,  $P_{gc}$ , mot kvalitetsparametre som kompresjonsdybde og -rate. Denne fordelingen sammenlignes opp mot retningslinjene for HLR. En slik sammenligning viser om det er en rimelig sammenheng mellom  $P_{gc}$  og retningslinjer. Rimelig i den forstand at det observeres høy sannsynlighetsverdi når behandling utføres etter retningslinjer, og lav sannsynlighetsverdi dersom behandling utføres med avvik fra retningslinjer.

Tabell 3.5 viser hvordan kompresjonsdybde og -rate blir fordelt over 4 intervaller, og sammenlignet opp mot gjennomsnittlig  $P_{gc}$ . I tabell 3.5 er intervallene som inneholder verdiene fra retningslinjene markert med \*.

**Tabell 3.5:** Tabellen viser gjennomsnittlig  $P_{gc}$  i 4 forskjellige intervall for kompresjonsdybde og -rate, fra testsett.  $SD$  - Standardavvik. \* - Markerer intervallene som inneholder verdiene fra retningslinjene for HLR.

| Testsett  | Kompresjonsdybde (cm)                                     |   |   |     |
|---|---|---|---|-----|
|   | 0-2   | 2-4   | 4-6*  | 6-8 |
| $\bar{P}_{gc,0-2}$<br>( $\bar{P}_{gc,0-2} \pm SD$ )     | $\bar{P}_{gc,2-4}$<br>( $\bar{P}_{gc,2-4} \pm SD$ )       | $\bar{P}_{gc,4-6}$<br>( $\bar{P}_{gc,4-6} \pm SD$ )         | $\bar{P}_{gc,6-8}$<br>( $\bar{P}_{gc,6-8} \pm SD$ )         |     |
| Kompresjonsrate (komp/min)                              |   |   |   |     |
| 40-70   | 70-100  | 100-130*  | 130-160   |     |
| $\bar{P}_{gc,40-70}$<br>( $\bar{P}_{gc,40-70} \pm SD$ ) | $\bar{P}_{gc,70-100}$<br>( $\bar{P}_{gc,70-100} \pm SD$ ) | $\bar{P}_{gc,100-130}$<br>( $\bar{P}_{gc,100-130} \pm SD$ ) | $\bar{P}_{gc,130-160}$<br>( $\bar{P}_{gc,130-160} \pm SD$ ) |     |

Tabell 3.5 gir en kompakt presentasjon av fordelingen av  $P_{gc}$ , mot kompresjonsdybde og -rate. For å undersøke fordelingen videre, benyttes en visuell presentasjon av fordelingen. Figur 3.7 illustrerer hvordan fordeling mellom  $P_{gc}$  og kompresjonsdybde benyttes til å sammenligne opp mot retningslinjer for HLR. Tabell 3.5 gjenspeiles i figur 3.7 c), hvor  $\bar{P}_{gc}$  og standardavvik fra testsett er vist. Forskjellen mellom tabell 3.5 og figur 3.7 c) er at figuren viser klassespesifikk fordeling.



**Figur 3.7:** Forhold mellom sannsynlighet for gunstige kompresjoner,  $P_{gc}$ , og kompresjonsdybde,  $KD$ . Gunstig og ikke gunstig behandling er merket med henholdsvis blå og røde punkt. **a)** Alle datapunktene fra treningsett. Her observeres det hvordan siling av treningsett fjerner datapunkt for gunstig behandling utenfor intervallet 3-7 cm. **b)** Alle datapunktene fra testsett. **c)** Gjennomsnitt og standardavvik for  $P_{gc}$ , fra testsett. Rød stiplet linje markerer grense for hvor datapunkt blir klassifisert som gunstige kompresjoner.  $P_{gc} > 0.5$  klassifiseres som gunstige kompresjoner.

# Kapittel 4

## Resultat

Dette kapitlet presenterer klassifiseringsresultat av testsett og fordeling av  $P_{gc}$  mot kvalitetsparametre for hjertekompresjoner. Det er utført eksperiment for flere datasett og sannsynlighetsmodeller. Klassifiseringsnøyaktighet for alle datasett og sannsynlighetsmodeller er vist i delkapittel 4.1, hvor datasettet som ga best klassifiseringsresultat brukes videre til sammenligning mot retningslinjer for HLR, vist i delkapittel 4.2. I delkapittel 4.3 presenteres maskinlæringsmetoden som ga best kombinasjon av klassifiseringsresultat og sammenheng med retningslinjer for HLR. Sannsynlighetsmodellen fra denne metoden undersøkes videre med bruk av utvidet testsett. Til slutt er det vist hvordan sannsynlighetsmodellen kan brukes til å gi en oversikt over kvalitet av hjertekompresjoner i en hjertestanseepisode. Dette er vist i delkapittel 4.4.

### 4.1 Klassifiseringsegenskaper

Det er utført flere eksperiment for forskjellige maskinlæringsmetoder og datasett. Her presenteres de eksperimentene som ga høyest klassifiseringsnøyaktighet. For ML-metoden og logistisk regresjon ble det utført eksperiment med forskjellige kombinasjoner av databehandlingsteknikker. Resultat fra eksperiment viste at bruk av alle databehandlingsteknikker ga høyest klassifiseringsnøyaktighet, og alle databehandlingsteknikker ble derfor også benyttet til eksperiment for nevralt nettverk. De globale parametrene som ga best klassifiseringsresultat for nevralt nettverk er vist i listen under.

- Læringsrate = 0.0001
- Noder = 15

- Partistørrelse = 128
- Antall skjulte lag = 1

Læringsraten som ga høyes klassifiseringsresultat er noe høyere enn de som ble inkludert i området for læringsrater, vist i kapittel 3.4.3. Denne læringsraten ble funnet ved små justeringer av de parametrene som ga høyest klassifiseringsnøyaktighet, og kan tyde på at området for læringsrate burde vært noe høyere. Tabell 4.1 viser klassifiseringsresultat av testsett, for alle datasett og maskinlæringsmetoder. Fordi nevralt nettverk krever en stor mengde datapunkt til trening, er nevralt nettverk bare benyttet for datasett 4, 5 og 6.

**Tabell 4.1:** Tabell viser klassifiseringsresultat av testsett, for alle maskinlæringsmetoder og datasett. ML - ML-metoden. LR - Logistisk regresjon. NN - nevralt nettverk. Fordi nevralt nettverk krever en stor mengde datapunkt til trening, er nevralt nettverk bare benyttet for datasett 4, 5 og 6. Resultatene som viser i tabellen er fra eksperiment med bruk av alle databehandlingsteknikker.

| Datasett | Modell | Treffprosent | Sens  | Spes  | PPV   | NPV   |
|----------|--------|--------------|-------|-------|-------|-------|
| 1        | ML     | 52,8%        | 37,8% | 63,2% | 41,8% | 59,3% |
|          | LR     | 46,7%        | 50%   | 44,3% | 38,5% | 55,9% |
| 2        | ML     | 50,7%        | 50,9% | 50,5% | 50,9% | 50,5% |
|          | LR     | 53%          | 63,6% | 42,2% | 52,6% | 53,5% |
| 3        | ML     | 50,2%        | 47%   | 52,7% | 43,5% | 56,2% |
|          | LR     | 45,9%        | 52%   | 41,1% | 40,6% | 52,5% |
| 4        | ML     | 52,2%        | 38,6% | 61,2% | 39,8% | 60%   |
|          | LR     | 41,7%        | 55,1% | 32,9% | 35,3% | 52,4% |
|          | NN     | 57,9%        | 32,8% | 74,5% | 46,1% | 62,5% |
| 5        | ML     | 48,8%        | 34,4% | 61,1% | 43%   | 52,2% |
|          | LR     | 52,4%        | 57,1% | 48,4% | 48,6% | 56,9% |
|          | NN     | 58,7%        | 58,6% | 58,7% | 54,8% | 62,4% |
| 6        | ML     | <b>58,6%</b> | 35,8% | 71,6% | 41,9% | 66,1% |
|          | LR     | <b>62,9%</b> | 50,0% | 70,4% | 49,1% | 71,1% |
|          | NN     | <b>63,4%</b> | 53,1% | 69,3% | 49,7% | 72,1% |

Tabell 4.1 viser at det bare er metoder med datasett 6, med unntak av nevralt nettverk med datasett 4 og 5, som oppnår treffprosent rundt 60%. Fra tabell 4.1 kan det også observeres at ML-metoden, med datasett 6, er dårlig til å klassifisere klasse 0, ettersom Sens og PPV har lav prosent. 36% sens betyr at 36% av totalt antall datapunkt, merket som sann klasse 0, blir korrekt klassifisert.

Basert på klassifiseringsresultatene, vist i tabell 4.1, velges datasett 6 til bruk ved

sammenligning mellom sannsynlighetsmodell og retningslinjer for HLR. For å undersøke klassifiseringsnøyaktighet av datasett 6 nærere, benyttes forvirringsmatrisen fra disse eksperimentene. Forvirringsmatrisene er vist i tabell 4.2.

**Tabell 4.2:** Forvirringsmatriser fra eksperiment som ga best klassifiseringsresultat mot testsett, for datasett 6 og alle metoder. 0 - Ikke gunstig behandling ( $\Delta P_{des} < -0.1$ ). 1 - Gunstig behandling ( $\Delta P_{des} > 0.1$ ).

| ML-metoden |   | Predikert |      |          |
|------------|---|-----------|------|----------|
| Testsett   |   | 0         | 1    | $\Sigma$ |
| Sann       | 0 | 583       | 1045 | 1628     |
|            | 1 | 807       | 2039 | 2846     |
| $\Sigma$   |   | 1390      | 3084 | 4474     |

| Logistisk regresjon |   | Predikert |      |          |
|---------------------|---|-----------|------|----------|
| Testsett            |   | 0         | 1    | $\Sigma$ |
| Sann                | 0 | 814       | 814  | 1628     |
|                     | 1 | 843       | 2003 | 2846     |
| $\Sigma$            |   | 1657      | 2817 | 4474     |

| Nevralt nettverk |   | Predikert |      |          |
|------------------|---|-----------|------|----------|
| Testsett         |   | 0         | 1    | $\Sigma$ |
| Sann             | 0 | 864       | 764  | 1628     |
|                  | 1 | 874       | 1972 | 2846     |
| $\Sigma$         |   | 1738      | 2736 | 4474     |

I tabell 4.2 kommer det frem at ML-metoden har problemer med å klassifisere sann klasse 0, ettersom kun 583 av totalt 1628 merket sann klasse 0, blir korrekt klassifisert. Dette ble også observert i tabell 4.1, hvor klassifiseringsresultat fra alle datasett ble vist. Alle sannsynlighetsmodellene favoriserer klasse 1, hvor rundt 2900 datapunkt, av totalt 4474 datapunkt, blir predikert som klasse 1. Favorisering av klasse 1 er å foretrekke fremfor favorisering av klasse 0, ettersom datapunkt merket som sann klasse 1, har gitt en positiv utvikling i fysiologisk respons, basert på  $\Delta P_{des}$ .

Tabell 4.2 viser store likhetstrekk mellom logistisk regresjon og nevralt nettverk, hvor begge metodene har tilnærmet like forvirringsmatriser. Forvirringsmatrisene viser at nevralt nettverk klassifisere sann klasse 0 bedre enn logistisk regresjon. Sammenlignet med logistisk regresjon, går det på bekostning av korrekt klassifisering av klasse 1. Tabell 4.2 viser at både logistisk regresjon og nevralt nettverk er bedre enn ML-metoden til å korrekt klassifisere klasse 0. For logistisk regresjon blir 814

datapunkt, av totalt 1628 datapunkt merket som sann klasse 0, korrekt klassifisert. Sammenlignet med ML-metoden er dette 231 flere datapunkter korrekt klassifisert som klasse 0.

## 4.2 Fordeling av sannsynlighetsverdier

For å undersøke hvordan sannsynlighetsmodellene, basert på de forskjellige maskinlæringsmetodene, fordeler sannsynlighetsverdier over verdier for kompresjonsdybde og -rate, benyttes gjennomsnitt av  $P_{gc}$ . Fordeling av  $\bar{P}_{gc}$  er vist tabell 4.3, hvor kompresjonsdybde og -rate fra testsettet er delt inn i fire intervaller. Her sammenlignes fordelingen av  $\bar{P}_{gc}$  mot retningslinjene for HLR. Ettersom avvik fra retningslinjene er benyttet som en del av databehandlingen, med bruk av siling av treningsett, er det rimelig å forvente at sannsynlighetsmodellene gir høyest  $\bar{P}_{gc}$  dersom behandling er utført etter retningslinjer. Som tidligere nevnt tilsier retningslinjene for HLR at optimal behandling består av kompresjonsdybde rundt 5 cm, men ikke over 6 cm, kombinert med en kompresjonsrate mellom 100 og 120 kompresjoner per minutt. I tabell 4.3 er intervallene som inneholder verdiene fra retningslinjene markert med \*.

**Tabell 4.3:** Tabellen viser gjennomsnittlig  $P_{gc}$  i 4 forskjellige intervall for kompresjonsdybde og -rate fra testsett i datasett 6, for alle metoder. I parentes under  $\bar{P}_{gc}$  vises  $\bar{P}_{gc} \pm$  standardavvik. ML - ML-metoden. LR - Logistisk regresjon. NN - nevralt nettverk. \* - Markerer intervallene som inneholder verdiene fra retningslinjene for HLR.

| Metode | Testsett                   |                              |                              |                              |
|--------|----------------------------|------------------------------|------------------------------|------------------------------|
|        | Kompresjonsdybde (cm)      |                              |                              |                              |
|        | 0-2                        | 2-4                          | 4-6*                         | 6-8                          |
| ML     | <b>0</b><br>(0-0)          | <b>0.47</b><br>(0.154-0.782) | <b>0.71</b><br>(0.451-0.968) | <b>0.3</b><br>(0.016-0.583)  |
| LR     | <b>0</b><br>(0-0.002)      | <b>0.37</b><br>(0.122-0.61)  | <b>0.68</b><br>(0.544-0.815) | <b>0.18</b><br>(0.04-0.319)  |
| NN     | <b>0.03</b><br>(0-0.088)   | <b>0.36</b><br>(0.124-0.6)   | <b>0.69</b><br>(0.523-0.849) | <b>0.35</b><br>(0.166-0.534) |
|        | Kompresjonsrate (komp/min) |                              |                              |                              |
|        | 40-70                      | 70-100                       | 100-130*                     | 130-160                      |
| ML     | <b>0</b><br>(0-0.016)      | <b>0.44</b><br>(0.005-0.821) | <b>0.69</b><br>(0.455-0.93)  | <b>0.39</b><br>(0.126-0.651) |
| LR     | <b>0.04</b><br>(0-0.073)   | <b>0.51</b><br>(0.295-0.726) | <b>0.59</b><br>(0.371-0.814) | <b>0.29</b><br>(0.07-0.51)   |
| NN     | <b>0.1</b><br>(0-0.252)    | <b>0.57</b><br>(0.273-0.859) | <b>0.58</b><br>(0.365-0.795) | <b>0.32</b><br>(0.1-0.538)   |

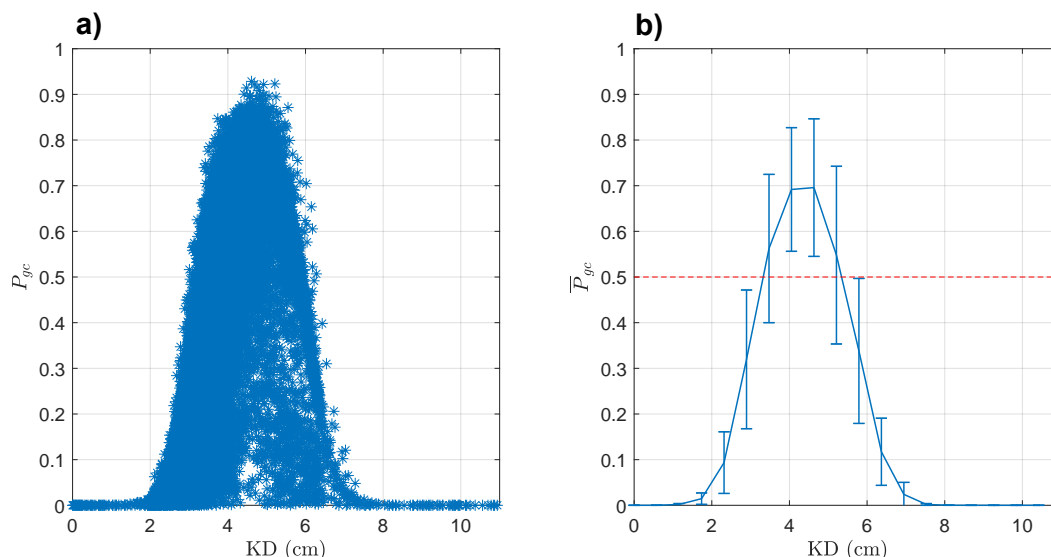
Fra tabell 4.3 observeres det at alle sannsynlighetsmodeller gir høyest  $\bar{P}_{gc}$  i intervallet som inneholder verdiene fra retningslinjene for HLR, merket med \*. For eksempel er  $\bar{P}_{gc}$  for LR lik 0.68 i intervallet 4–6 cm for kompresjonsdybde, samtidig som de andre intervallene gir en  $\bar{P}_{gc}$  under 0.4. Siste kolonne for kompresjonsdybde viser at det er en LR-basert  $P_{gc}$  som gir lavest gjennomsnittlig  $P_{gc}$  for kompresjonsdybde mellom 6 og 8 cm. Sammenhengen mellom  $P_{gc}$  og retningslinjer for HLR gir en indikasjon om at alle sannsynlighetsmodellene kan brukes til å indikere kvalitet av hjertekompersjoner.

Tabell 4.3 viser at sannsynlighetsmodell basert på ML-metoden gir høyest  $\bar{P}_{gc}$  i intervallet med retningslinjene, markert med \*. Sammenlignet med en LR-basert sannsynlighetsmodell går det på bekostning av høyere  $\bar{P}_{gc}$  for de andre intervallene for kompresjonsdybde. Basert på  $\bar{P}_{gc}$ -verdier i tabell 4.3 velges LR-basert  $P_{gc}$  som best sannsynlighetsmodell, ettersom den gir lavest  $\bar{P}_{gc}$  for kompresjonsdybde over 6 cm, kombinert med rimelig sammenheng med retningslinjene for HLR. En LR-basert  $P_{gc}$  gir også lavest  $\bar{P}_{gc}$  for kompresjonsrate mellom 130 og 160 kompersjoner per minutt.

En visuell fremstilling av fordelingen av  $P_{gc}$ , for alle maskinlæringsmetoder, er vist i vedlegg A.3, hvor også fordelingen av  $P_{gc}$ -verdier fra treningsettet er vist. Sammenlignet med tabell 4.3 viser figurene i vedlegg A.3 klassespesifikk fordeling. Her er det også benyttet flere intervaller, som gir en høyere oppløsning av fordeling. Ved å studere fordelingene av  $P_{gc}$ , vist i vedlegg A.3, observeres samme resultat som ble vist i tabell 4.3, hvor en LR-basert  $P_{gc}$  gir best sammenheng med retningslinjene for HLR.

### 4.3 Undersøkelse av valgt sannsynlighetsmodell

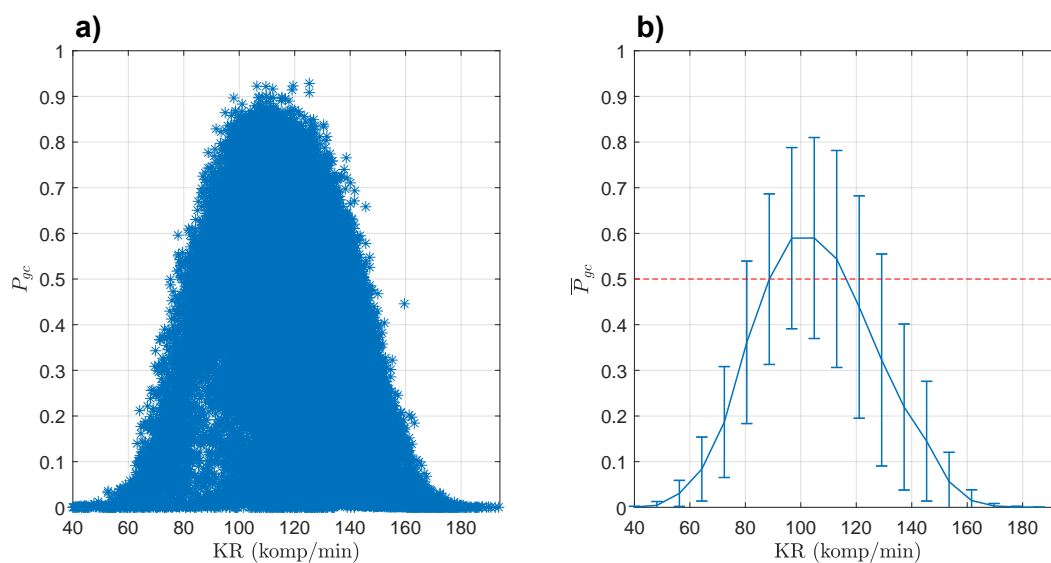
Basert på klassifiseringsnøyaktighet og rimelig sammenheng med retningslinjer for HLR, er heretter  $P_{gc}$  definert som LR-basert  $P_{gc}$ , ettersom logistisk regresjon ga best resultat. For å undersøke hvordan sannsynlighetsmodellen fordeler  $P_{gc}$ -verdier mot kvalitetsparametre som ikke er merket, benyttes utvidet testsett. Dermed blir datapunkt med  $\Delta P_{des}$  mellom -0.1 og 0.1 også undersøkt. Figur 4.1 viser forhold mellom  $P_{gc}$  og kompresjonsdybde.



**Figur 4.1:** Forhold mellom  $P_{gc}$  og kompresjonsdybde for alle datapunkt fra episoder i testsett. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik. Rød stiplet linje markerer grense for hvor data klassifiseres som gunstig behandling.

Figur 4.1 **b)** viser at toppunkt for gjennomsnittlig  $P_{gc}$  ligger rundt 4,5 cm, med en sannsynlighetsverdi på rundt 0,7. Figuren viser at KD lik 6 cm, gir en gjennomsnittlig  $P_{gc}$  rundt 0,25, samtidig som KD lik 5 cm, gir  $\bar{P}_{gc}$  rundt 0,6. Det betyr at dersom hjertekompresjoner utføres med en KD over 6 cm, vil sannsynlighetsmodell indikere dette som ugunstig behandling.

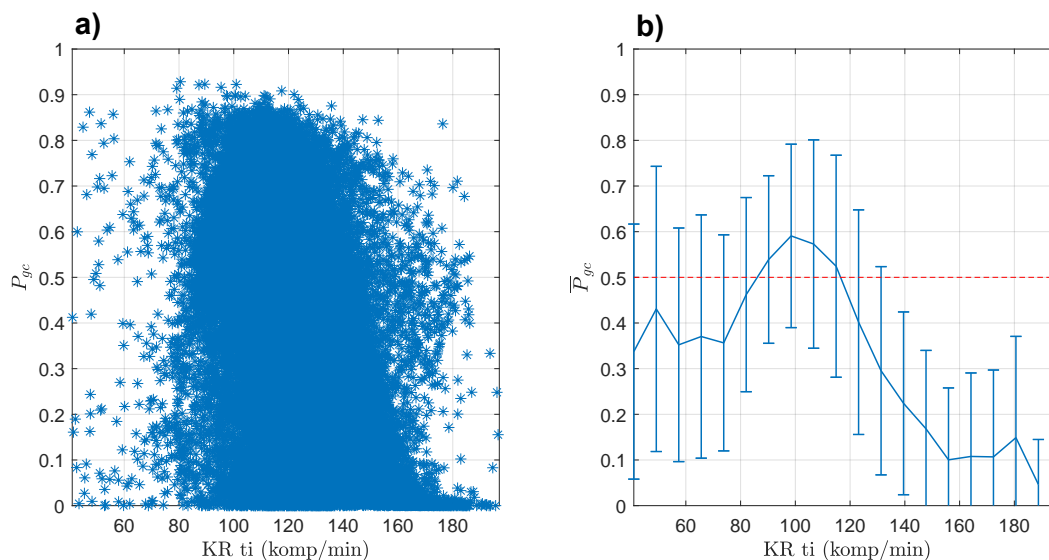
Tilsvarende forhold mellom  $P_{gc}$  og kompresjonsrate er vist i figur 4.2. Her er toppunkt for  $\bar{P}_{gc}$  rundt 105 kompresjoner per minutt.



**Figur 4.2:** Forhold mellom  $P_{gc}$  og kompresjonsrate for alle datapunkt fra episoder i testsett. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik. Rød stiplet linje markerer grense for hvor data klassifiseres som gunstig behandling.



Fra figur 4.2 observeres det at sannsynlighetsmodell, for KR lik 120, gir  $\overline{P}_{gc}$  rundt 0,45. Dette er en litt lavere verdi enn hva som hadde vært gunstig, med tanke på sammenligning mot retningslinjene for HLR. Figur 4.3 viser fordeling av  $P_{gc}$  mot kompresjonsrate basert på impedansmåling.



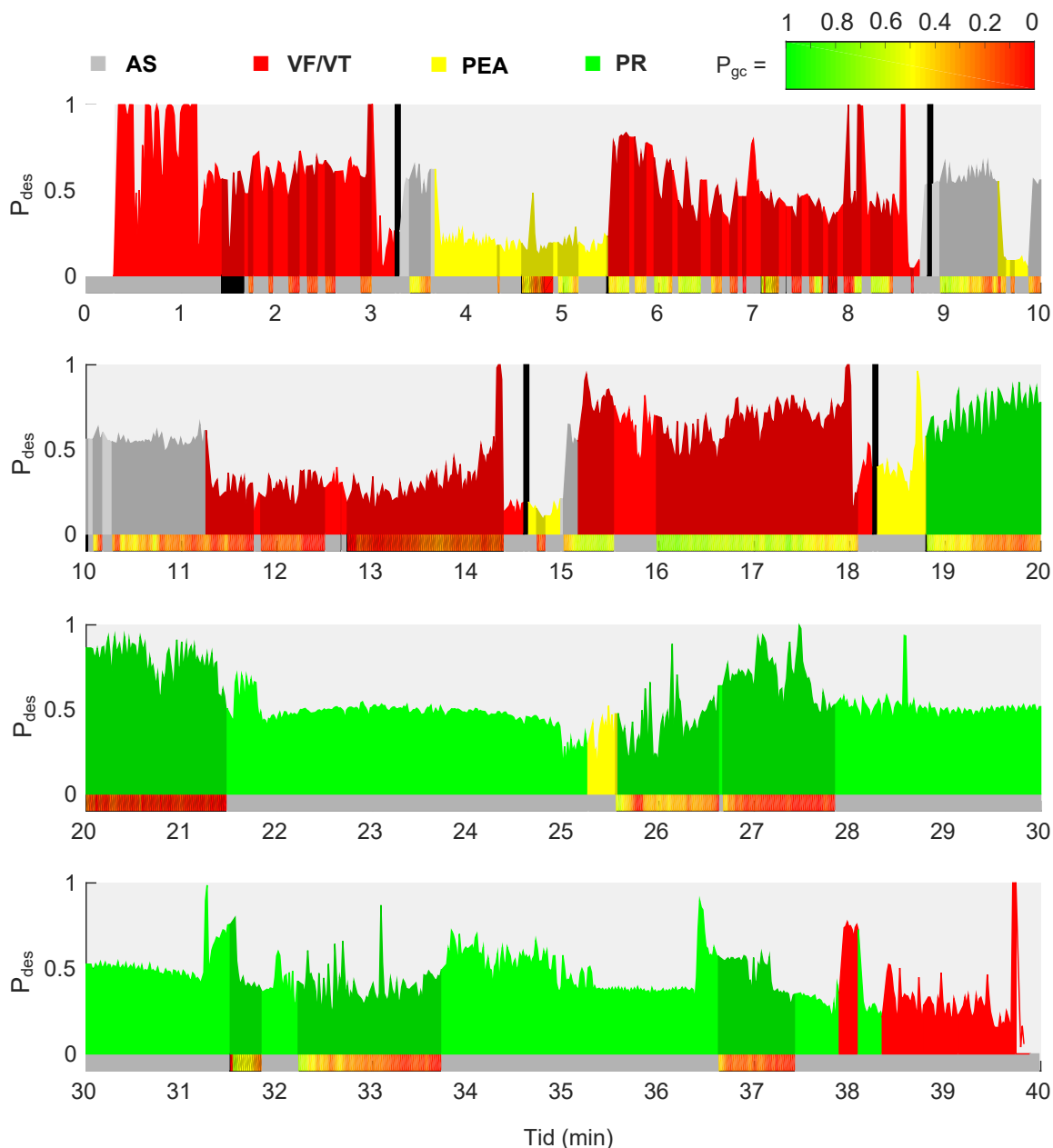
**Figur 4.3:** Forhold mellom  $P_{gc}$  og kompresjonsrate fra impedansmåling. Datapunkt fra datasett 6, hvor alle datapunkt fra episoder i testsett er plottet. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik. Rød stiplet linje markerer grense for hvor datapunkt klassifiseres som gunstig behandling.

Det undersøkes også hvordan sannsynlighetsmodell gir forhold mellom  $P_{gc}$  og de resterende kvalitetsparametrene. Det er vist i vedlegg A.4. Her observeres det at  $\overline{P}_{gc}$ , for kvalitetsparametrene  $SDKD$  og  $SDKR$ , har en synkende trend når verdi for kvalitetsparametrene øker. Det betyr at et større standardavvik for kompresjonsdybde og -rate, vil i gjennomsnitt medføre en lavere sannsynlighet for at det utføres gunstige kompresjoner. Dette er et gunstig resultat i forbindelse med behandling, ettersom hjertekompresjoner med lav variasjon er å foretrekke.

## 4.4 Visualisering av hjertestansepisode

For å koble klassifisering av kvalitet av hjertekompresjoner tilbake til figur som viser oversikt over hjertestansepisode, ble sannsynlighetsmodell implementert i eksisterende program. Figur 4.4 er et eksempel hvor kvalitet av hjertekompresjoner indikeres med bruk av  $P_{gc}$ . Denne figuren viser samme hjertestansepisode som figur 3.1. Alle intervall med hjertekompresjoner, tidligere markert med sort horisontal strek i bunn av hvert delplott, markeres nå med fargekode. Fargekoden representere sann-

synlighet for gunstige kompresjoner,  $P_{gc}$ . Grønn farge tilsvarer sannsynlighet lik 1, og rød farge tilsvarer sannsynlighet lik 0. Dermed visualiseres dynamikk i kvalitet av hjertekompresjoner som endring av farge over tid. Dersom komprejonsintervall er merket med sort farge, mangler intervallet kvalitetsparametre.



**Figur 4.4:** Oversiktsplott over en hjertestanseepisode, med tid(min) på x-akse og sannsynlighet for endring til bedre tilstand ( $P_{des}$ ) på y-akse. Hjerterytme indikeres med fargekode. Rød: VT/VE, Gul: PEA, Grå: AS, Grønn: PR. Sammenlignet med figur 3.1, hvor hjertekompresjoner ble indikert med sort horisontal strek, er nå hjertekompresjoner visualisert med sannsynlighet for gunstige kompresjoner ( $P_{gc}$ ). Sannsynligheten representeres i form av interpolert fargekode, fra grønn til rød, hvor grønn fargekode tilsvarer sannsynlighet for gunstige kompresjoner lik 1.

Figur 4.4 viser at flere hjertekompresjoner klassifiseres som ugunstig behandling, ettersom kompresjonsintervall markeres med rød farge. Ved å studere datagrunnlaget for denne episoden, kommer det frem at kompresjonsintervall, merket med rød farge, ble utført med kompresjonsdybde under 3.5 cm. Datagrunnlaget viser også at kompresjonsraten for disse intervallene var over 120 kompresjoner per minutt. Et unntak er intervallet som startet etter 16 minutter. Her viser datagrunnlaget at kompresjonsdybde lå rett under 4 cm, med en kompresjonsrate rundt 116 kompresjoner per minutt. Basert på resultat av fordeling av  $P_{gc}$  mot kvalitetsparametre for hjertekompresjoner, vist i delkapittel 4.3, forklarer det hvorfor dette intervallet er merket med en gul/grønn farge.

I vedlegg A.5 sammenlignes implementering av alle sannsynlighetsmodeller, basert på alle maskinlæringsmetoder, hvor 9 episoder fra testsettet er benyttet. Her undersøkes det om endring i pasientens hjerterytme kan forklares av gunstig eller ugunstig utførelse av hjertekompresjoner. Ved sammenligning observeres det store likhetstrekk mellom sannsynlighetsmodellene, ettersom  $P_{gc}$  gir tilsvarende lik fargeindikasjon for kompresjonsintervallene.



# Kapittel 5

## Diskusjon

Det er i denne oppgaven presentert flere sannsynlighetsmodeller som automatisk indikerer kvalitet av hjertekompresjoner, i form av  $P_{gc}$ . For å trene sannsynlighetsmodellene er det utført flere eksperimenter, med bruk av flere datasett, ettersom det finnes flere alternativ for å indikere endring i pasientens tilstand. Alle datasettene som er benyttet er fra faktiske hendelser med hjertestans, logget med defibrillator fra ambulanse. Datasettene består av egenskaper som kompresjonsdybde og kompresjonsrate, og standardavvik av disse. Klassifiseringsresultat for alle datasettene ble sammenlignet, hvor sammenligningen viste at klassifisering med datasett 6 ga høyest klassifiseringsnøyaktighet. Basert på klassifiseringsnøyaktighet, ble datasett 6 benyttet til å trene den endelige sannsynlighetsmodellen. Grunnen til at datasett 6, i gjennomsnitt, ga 10% høyere treffprosent enn de andre datasettene er noe uklart. Men det kan tenkes at oppdeling av sekvenser som var lengre enn 1 minutt var viktig, ettersom det ble benyttet for datasett 6. Det reduserer påvirkning av  $P_{des}$ -verdier lenge før kompresjonsintervall, og burde kanskje også vært implementert i metode 2 for  $\Delta P_{des}$ .

For å evaluere hvilken sannsynlighetsmodell som presterte best, ble treffprosent for klassifisering benyttet som indikasjon for ytelse til modell. Endelig evaluering ble gjort ved å studere fordeling av  $P_{gc}$  mot kvalitetsparametre, og sammenligne fordelingen opp mot retningslinjer for HLR. Det var her rimelig å forvente en sannsynlighetsmodell som ga høy  $P_{gc}$  dersom behandling ble utført etter retningslinjer. I tillegg var det rimelig å forvente at modellen indikerte ugunstig behandling, basert på avvik fra retningslinjer, med lav sannsynlighetsverdi. Sammenligning viste at alle sannsynlighetsmodellene hadde store likheter, men basert på kombinasjon av klassifiseringsresultat og rimelig sammenheng med retningslinjer, ble logistisk regresjon valgt som endelig modell.

I delkapitlene under diskuteres eventuelle feilkilder, konklusjon i forhold til valg av beste modell og fremtidig arbeid.

## 5.1 Feilkilder

I figur 3.4, hvor spredningsplott av datapunkt er vist, ble det observert at klassene i datasettet, merket gunstig og ikke gunstig behandling, hadde stor overlapp. Det forklarer hvorfor sannsynlighetsmodellene hadde store problemer med å oppnå høy treffprosent. Med utgangspunkt i denne figuren, er det logisk å anta at datapunkt merket som ikke gunstig behandling, nødvendigvis ikke er dårlig behandling, ettersom flere av disse datapunktene er i tråd med retningslinjene for HLR. Grunnen til at de er merket som ikke gunstig behandling kan skyldes andre faktorer som påvirker pasientens tilstand, som har medført en  $\Delta P_{des}$  under  $-0.1$ . Det ble vist at høyere treffprosent for nevralt nettverk, var på grunn av høyere grad av korrekt klassifisering av ikke gunstig behandling. Noe som gikk på bekostning av korrekt klassifisering av gunstig behandling. Dette er også et argument for at logistisk regresjon ble valgt som beste sannsynlighetsmodell, selv om nevralt nettverk hadde en høyere treffprosent.

Et kompresjonsintervall varer gjerne i opp mot flere minutter, hvor utførelsen av behandling kan variere. Det betyr at et intervall nødvendigvis ikke bare består av gunstig eller ugunstig behandling. I datasett 1-3 er det tatt hensyn til dette ved å bruke gjennomsnitt av kvalitetsparametre i kompresjonsintervall, som dermed reflekterer den gjennomsnittlige behandlingen for kompresjonsintervallet. Men fordi disse datasettene besto av få datapunkt, og ga lav treffprosent, ble ingen av disse datasettene valgt til trening av endelig modell. Datasett 4-6 tar ikke hensyn til varierende kvalitet i kompresjonsintervall, hvor hele kompresjonsintervall blir merket som gunstig eller ikke gunstig behandling. Dette kan være en kilde til lav treffprosent ved klassifisering, for eksempel dersom et intervall, merket som ikke gunstig, består av både gunstig og ikke gunstig behandling.

Merking av data er basert på  $P_{des}$ . Fordi beregningen av  $P_{des}$  er automatisk, kan det medføre eventuelle feil, som igjen kan påvirke merking av datapunkt. For eksempel kan det, i figur 3.1, observeres at  $P_{des}$  synker betraktelig før det utføres elektrisk sjekk med defibrillator. Det kan være en feilkilde for datasett 2 og 5, ettersom disse datasettene er hentet ved bruk av sekvenser etter hjertekompresjoner. I tillegg er  $P_{des}$  beregnet forskjellig, avhengig av hjerterytme. Dette medfører en usikkerhet for verdiene av  $P_{des}$  i overganger for hjerterytme.

## 5.2 Konklusjon

Målet med denne oppgaven var å utvikle en sannsynlighetsmodell av sammenhengen mellom kvalitetsparametre for hjertekompressjoner og fysiologisk respons. I kapittel 4, hvor resultat ble presentert, ble logistisk regresjon valgt som beste sannsynlighetsmodell. Forhold mellom  $P_{gc}$  og kvalitetsparametre, sammenlignet opp mot retningslinjer for HLR, vist i delkapittel 4.3, indikerer at sannsynlighetsmodell klarer å skille mellom gunstig og ikke gunstig behandling. Sammenligningen, kombinert med gjennomgang av datagrunnlag fra hjertestanseepisode hvor sannsynlighetsmodell er implementert, gir grunnlag for å tro at  $P_{gc}$  kan brukes til å indikerer kvalitet av hjertekompressjoner. I delkapittel 4.4 ble det vist hvordan  $P_{gc}$  gjør det mulig å visualiserer kvalitet av hjertekompressjoner, over tid, på en kompakt og dynamisk måte.

## 5.3 Fremtidig arbeid

Til fremtidig arbeid kan  $P_{gc}$  benyttes til å undersøke sammenhenger mellom kvalitet i hjertekompressjoner og hjerterytmer. For eksempel kan det være interessant å undersøke hvordan kvalitet av hjertekompressjoner fordeler seg over tid. For episoder med dårlig utgangspunkt, i form av ugunstig hjerterytme, og som ender med spontan sirkulasjon, vil det da være mulig å identifisere om det er faktorer ved behandlingen som skiller gunstige fra ugunstige utfall. En sammenligning mellom hjerterytmer og  $P_{gc}$  er vist i vedlegg A.5, hvor det observeres hjerterytmer som endres til PR, etter det er utført hjertekompressjoner med gunstig kvalitet. Det gir en indikasjon om at det finnes sammenhenger som bør undersøkes.





# Bibliografi

- [1] Meiso Hayashi, Wataru Shimizu, and Christine M Albert. The spectrum of epidemiology underlying sudden cardiac death. *Circulation research*, 116(12):1887–1906, 2015.
- [2] Vibeke Lund Pettersen Eva Stensland. Norsk hjertestansregister. Hentet 26.03.2018 fra <https://www.kvalitetsregistre.no/registers/norsk-hjertestansregister>.
- [3] Ssb nøkkeltall for befolkning. Hentet 26.03.2018 fra <https://www.ssb.no/befolkning/nokkeltall/befolkning>.
- [4] Dana P Edelson, Benjamin S Abella, Jo Kramer-Johansen, Lars Wik, Helge Myklebust, Anne M Barry, Raina M Merchant, Terry L Vanden Hoek, Petter A Steen, and Lance B Becker. Effects of compression depth and pre-shock pauses predict defibrillation failure during cardiac arrest. *Resuscitation*, 71(2):137–145, 2006.
- [5] Charles N Pozner, Adam Almozlino, Jonathan Elmer, Stephen Poole, De’Ann McNamara, and David Barash. Cardiopulmonary resuscitation feedback improves the quality of chest compression provided by hospital health care professionals. *The American journal of emergency medicine*, 29(6):618–625, 2011.
- [6] Jo Kramer-Johansen, Helge Myklebust, Lars Wik, Bob Fellows, Leif Svensson, Hallstein Sørebo, and Petter Andreas Steen. Quality of out-of-hospital cardiopulmonary resuscitation with real time automated feedback: a prospective interventional study, 2006.
- [7] U Ayala, T Eftestøl, E Alonso, U Irusta, E Aramendi, S Wali, and J Kramer-Johansen. Automatic detection of chest compressions for the assessment of cpr-quality parameters. *Resuscitation*, 85(7):957–963, 2014.
- [8] Erik Alonso, Trygve Eftestøl, Elisabete Aramendi, Jo Kramer-Johansen, Eirik Skogvoll, and Trond Nordseth. Beyond ventricular fibrillation analysis: Comprehensive waveform analysis for all cardiac rhythms occurring during resuscitation. *Resuscitation*, 85(11):1541–1548, 2014.
- [9] Merethe Kvam Terje Johannessen. Hjertets elektriske system, 27.10.2015. <https://nhi.no/sykdommer/hjertekar/hjertearytmi/er/hjertets-elektriske-system/>.
- [10] Harald Arnesen. Hjerte. i store medisinske leksikon, 18.10.2017. Hentet 26. mars 2018 fra <https://sml.snl.no/hjerte>.
- [11] ZooFari. Heart diagram blood flow en.svg, Mars 2010. [https://commons.wikimedia.org/wiki/File:Heart\\_diagram\\_blood\\_flow\\_en.svg](https://commons.wikimedia.org/wiki/File:Heart_diagram_blood_flow_en.svg).
- [12] Attribution-sharealike 3.0 unported, (cc by-sa 3.0)). <https://creativecommons.org/licenses/by-sa/3.0/deed.en>.
- [13] Otto A. og Arnesen Harald Nordseth, Trond og Smiseth. Hjertestans. i store medisinske leksikon, 14.02.2018. Hentet 26. mars 2018 fra <https://sml.snl.no/hjertestans>.
- [14] Ali Bahrami Rad, Trygve Eftestøl, Unai Irusta, Jan Terje Kvaløy, Lars Wik, Jo Kramer-Johansen, Aggelos K Katsaggelos, and Kjersti Engan. An automatic system for the

- comprehensive retrospective analysis of cardiac rhythms in resuscitation episodes. *Resuscitation*, 122:6–12, 2018.
- [15] Jan Terje Kvaløy, Eirik Skogvoll, Trygve Eftestøl, Kenneth Gundersen, Jo Kramer-Johansen, Theresa Mariero Olasveengen, and Petter Andreas Steen. Which factors influence spontaneous state transitions during resuscitation? *Resuscitation*, 80(8): 863–869, 2009.
- [16] Eirik Skogvoll, Trygve Eftestøl, Kenneth Gundersen, Jan Terje Kvaløy, Jo Kramer-Johansen, Theresa Mariero Olasveengen, and Petter Andreas Steen. Dynamics and state transitions during resuscitation in out-of-hospital cardiac arrest. *Resuscitation*, 78(1):30–37, 2008.
- [17] Gavin D Perkins, Anthony J Handley, Rudolph W Koster, Maaret Castrén, Michael A Smyth, Theresa Olasveengen, Koenraad G Monsieurs, Violetta Raffay, Jan-Thorsten Gräsner, Volker Wenzel, et al. European resuscitation council guidelines for resuscitation 2015: Section 2. adult basic life support and automated external defibrillation. *Resuscitation*, 95:81–99, 2015.
- [18] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [19] Stephan Dreiseitl and Lucila Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, 35(5-6):352–359, 2002.
- [20] Scott A Czepiel. Maximum likelihood estimation of logistic regression models: theory and implementation. Available at [czep.net/stat/mlelr.pdf](http://czep.net/stat/mlelr.pdf), 2002.
- [21] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*, volume 5. Macmillan New York, 1993.
- [22] John E Dennis, Jr and Jorge J Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.
- [23] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.
- [24] C Nag. e04±minimizing or maximizing a function. *argument*, 4:8, 2006.
- [25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [26] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.
- [27] Lars Wik, Jo Kramer-Johansen, Helge Myklebust, Hallstein Sørebo, Leif Svensson, Bob Fellows, and Petter Andreas Steen. Quality of cardiopulmonary resuscitation during out-of-hospital cardiac arrest. *Jama*, 293(3):299–304, 2005.
- [28] Trygve Eftestøl, Kari Anne Haaland Thorsen, Erlend Tøssebro, Chunming Rong, and Petter Andreas Steen. Representing resuscitation data—considerations on efficient analysis of quality of cardiopulmonary resuscitation. *Resuscitation*, 80(3):311–317, 2009.
- [29] Trygve Eftestøl and Lawrence D Sherman. Towards the automated analysis and database development of defibrillator data from cardiac arrest. *BioMed research international*, 2014, 2014.
- [30] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):15, 2012.
- [31] David Lane. Online statistics education: A multimedia course of study. In *EdMedia: World Conference on Educational Media and Technology*, pages 1317–1320. Association for the Advancement of Computing in Education (AACE), 2003.

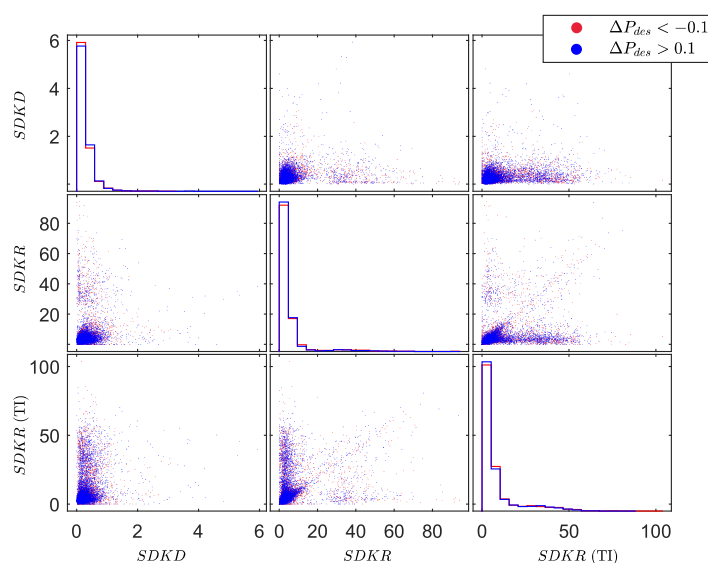
- 
- [32] Heman Mohabeer, KM Sunjiv Soyjaudah, and Narainsamy Pavaday. Enhancing the performance of neural network classifiers using selected biometric features. In *SENSORCOMM 2011, The fifth International Conference on Sensors Technologies and Applications*, pages 140–144, 2011.
- [33] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [34] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006.



# Vedlegg A

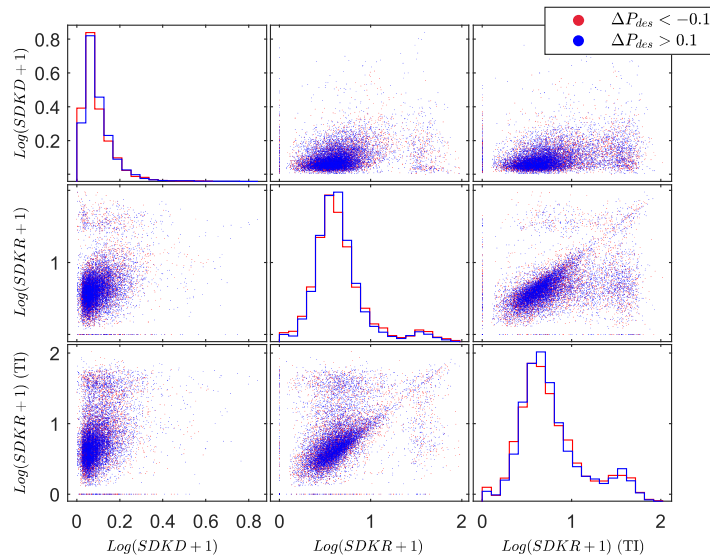
## A.1 Log-transformasjon

Visualiserer hvordan bruk av log-transformasjon endrer fordeling av datapunkt. Figur A.1 viser et spredningsplott av datapunkt før transformasjon.



**Figur A.1:** Spredningsplott av datapunkt fra datasett 6. Klassene gunstig og ikke gunstig behandling er merket med henholdsvis blå og røde punkt. Diagonalen viser hvordan datapunkt er fordelt for hver egenskap.

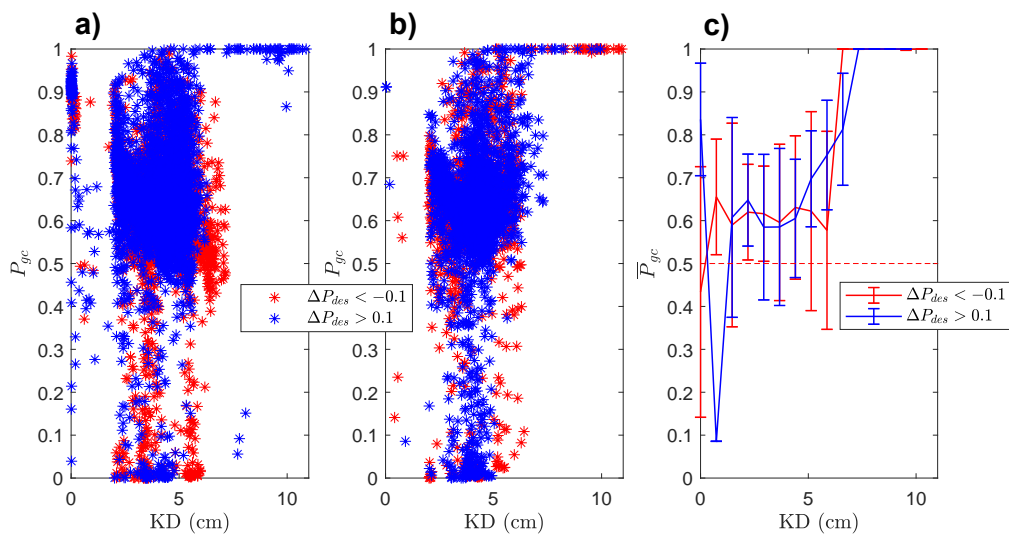
Resultat av log-transformasjon vises i form av et nytt spredningsplott. Dette er vist i figur A.2, hvor datapunkt nå har en mer symmetrisk fordeling.



**Figur A.2:** Spredningsplott av datapunkt fra datasett 6. Klassene gunstig og ikke gunstig behandling er merket med henholdsvis blå og rød punkt. Diagonalen viser hvordan datapunkt er fordelt for hver egenskap.

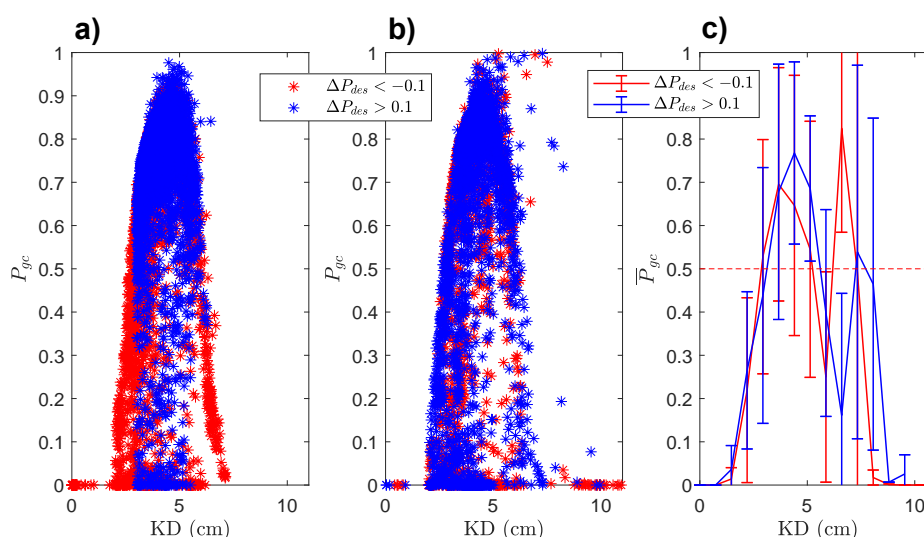
## A.2 Sammenligning med og uten siling av treningsett

I dette kapitlet sammenlignes ML-metoden med og uten bruk av siling av treningsett. Ved å plote sannsynlighet for gunstige kompresjoner,  $P_{gc}$ , mot kompresjonsdybde,  $KD$ , visualiseres fordeling av  $P_{gc}$ . Dette er vist i figur A.3, hvor det ikke er benyttet siling av treningsett.



**Figur A.3:** Klassifisering ved bruk av ML-metoden.  $P_{gc}$  gir sannsynlighet for at det utføres gode kompresjoner. Det er vist i y-akse, mot kompresjonsdybde ( $KD$ ) på x-akse. **a)** Alle datapunkt fra treningsett, uten siling. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

I figur A.3 observeres det at sannsynlighetsmodell fordeler  $P_{gc}$  rundt 0,6, uavhengig av kompresjonsdybde. Ved å implementere siling av treningsett som en del av databehandlingen, gir sannsynlighetsmodell fordelingen vist i figur A.4. Eneste forskjell av databehandling er siling av treningsett. Silingen kan observeres i figuren, hvor treningsett, merket med **a)**, ikke har datapunkt med gunstig behandling utenfor intervallet 3-7 cm. Figur A.4**b)** viser at datapunkt fra testsett, merket som gunstig behandling og som er utenfor intervallet, får lav sannsynlighetsverdi.



**Figur A.4:** Klassifisering ved bruk av ML-metoden.  $P_{gc}$  gir sannsynlighet for at det utføres gode kompresjoner. Det er vist i y-akse, mot kompresjonsdybde ( $KD$ ) på x-akse. **a)** Alle datapunkt fra treningsett. Her observeres det hvordan siling av treningsett fjerner datapunkt for gunstig behandling, utenfor intervallet 3-7 cm. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

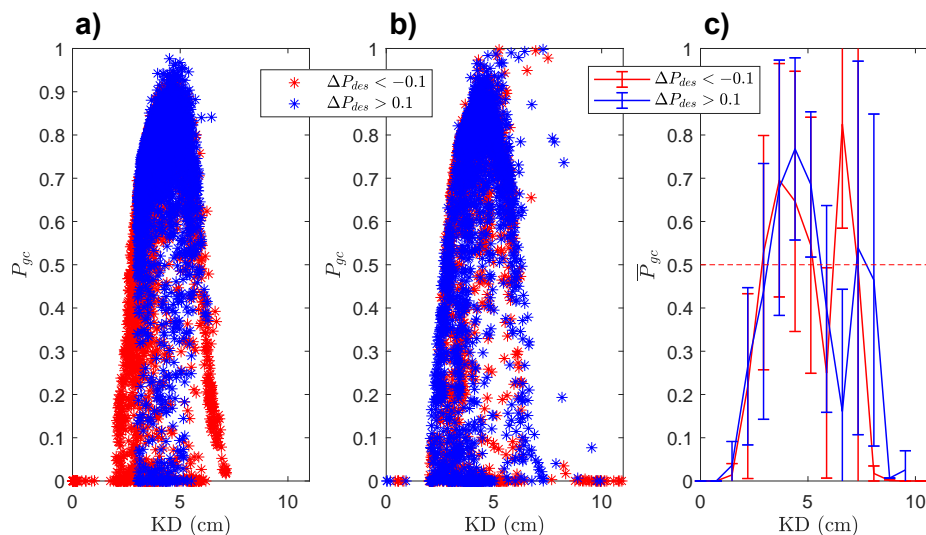
Fra figur A.4 observeres det en gunstig fordeling rundt retningslinjene for kompresjonsdybde. Gunstig ettersom  $P_{gc}$  fordeles med et toppunkt rundt 5 cm, samtidig som verdier for kompresjonsdybde som avviker fra retningslinjene gis lavere sannsynlighetsverdier.

### A.3 Fordeling av sannsynlighetsverdier

Her vises fordeling av sannsynlighetsverdier for gunstige kompresjoner,  $P_{gc}$ , mot kompresjonsdybde og -rate, som sammenlignes opp mot retningslinjer for HLR. Retningslinjene tilsier at optimal behandling består av kompresjonsdybde rundt 5 cm, men ikke over 6 cm, og kompresjonsrate mellom 100 og 120 kompresjoner per minutt.

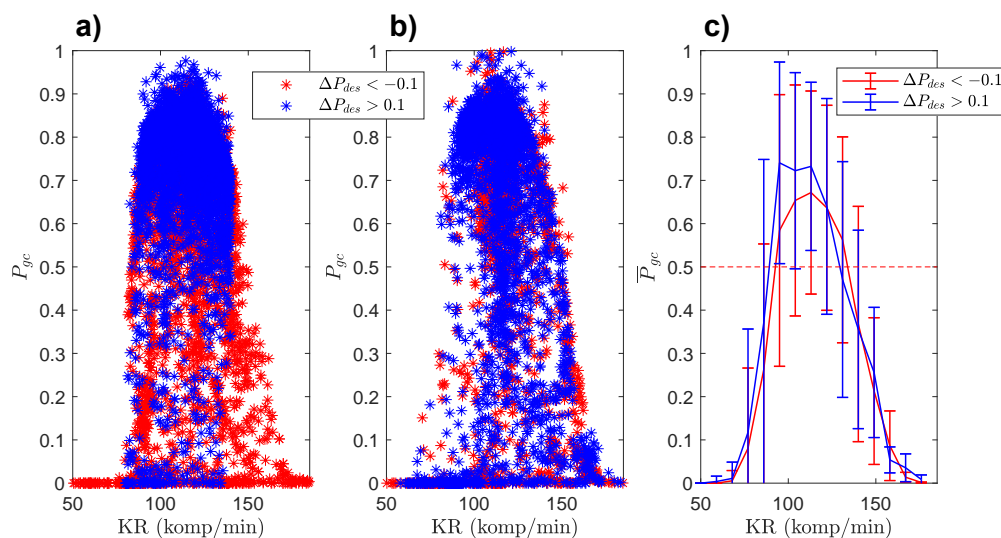
### A.3.1 ML-metoden

Figur A.5 viser fordeling av  $P_{gc}$  mot kompresjonsdybde.



**Figur A.5:** Klassifisering ved bruk av ML-metoden.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsdybde ( $KD$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

Tilsvarende fordeling mot kompresjonsrate er vist i figur A.6.

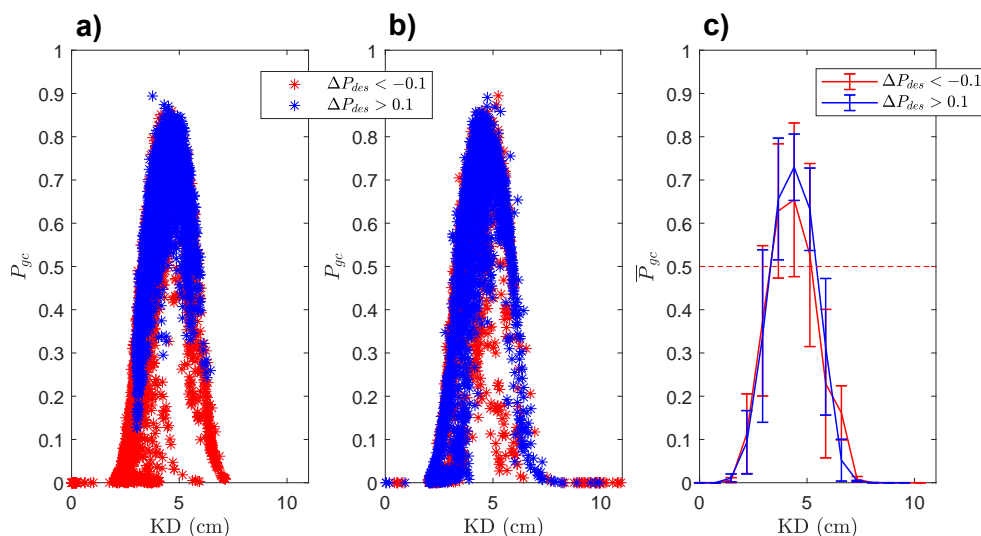


**Figur A.6:** Klassifisering ved bruk av ML-metoden.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsrate ( $KR$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.



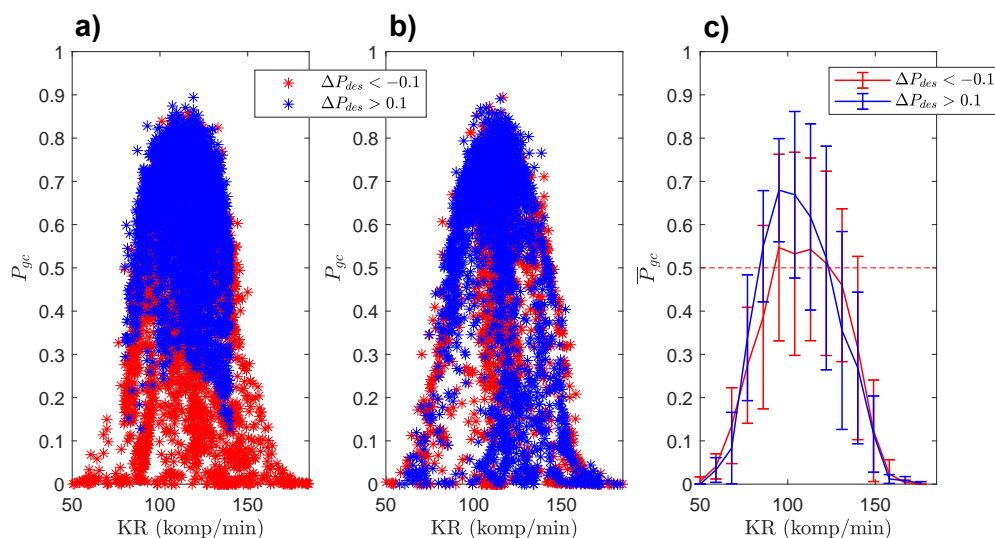
### A.3.2 Logistisk regresjon

Figur A.7 viser fordeling av  $P_{gc}$  mot kompresjonsdybde.



**Figur A.7:** Klassifisering ved bruk av logistisk regresjon.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsdybde ( $KD$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

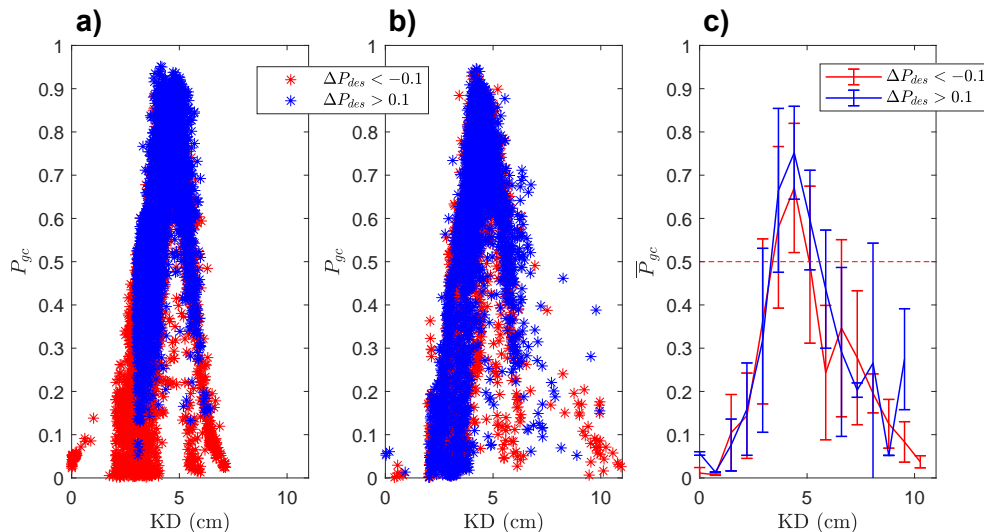
Tilsvarende fordeling mot kompresjonsrate er vist i figur A.8.



**Figur A.8:** Klassifisering ved bruk av logistisk regresjon.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsrate ( $KR$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

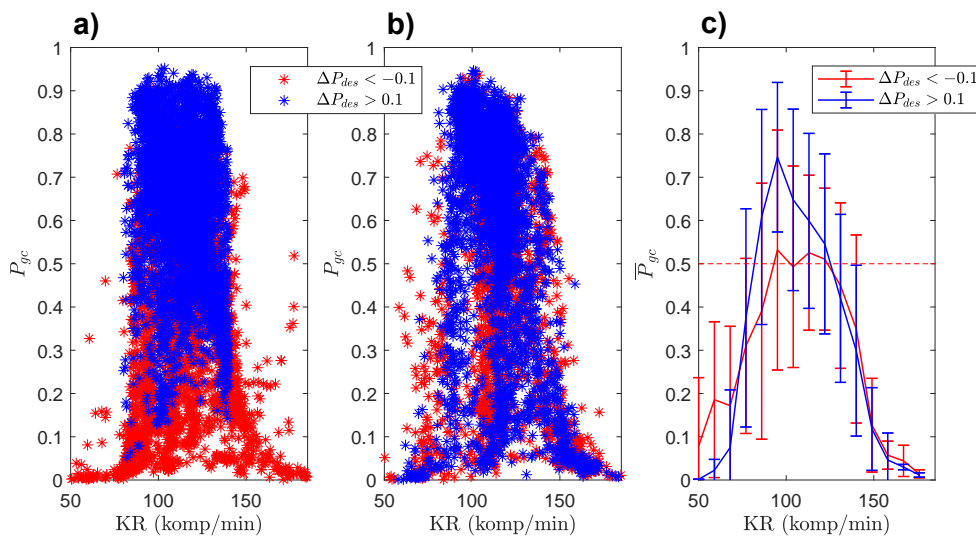
### A.3.3 Nevralt nettverk

Figur A.9 viser fordeling av  $P_{gc}$  mot kompresjonsdybde.



**Figur A.9:** Klassifisering ved bruk av nevralt nettverk.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsdybde ( $KD$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

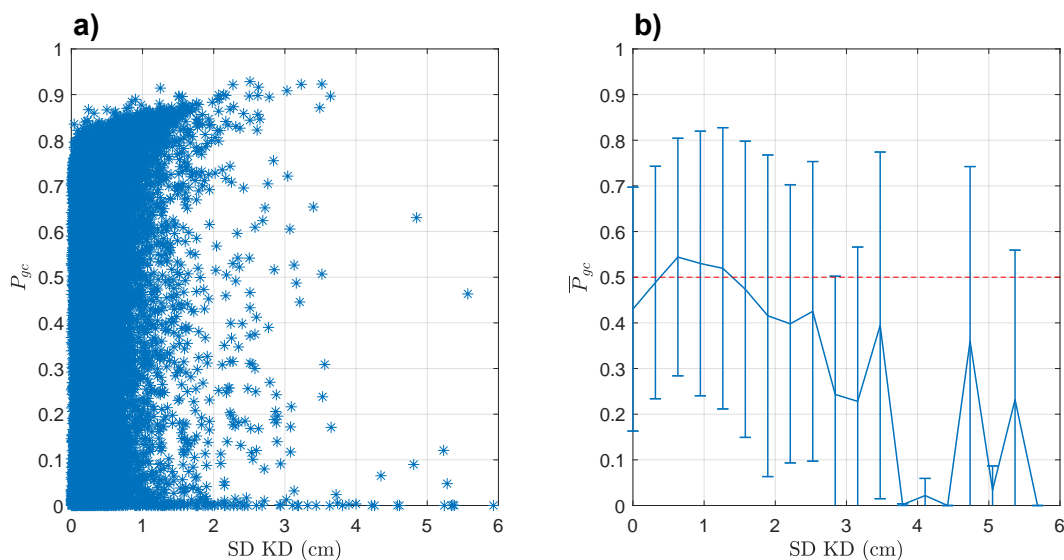
Tilsvarende fordeling mot kompresjonsrate er vist i figur A.10.



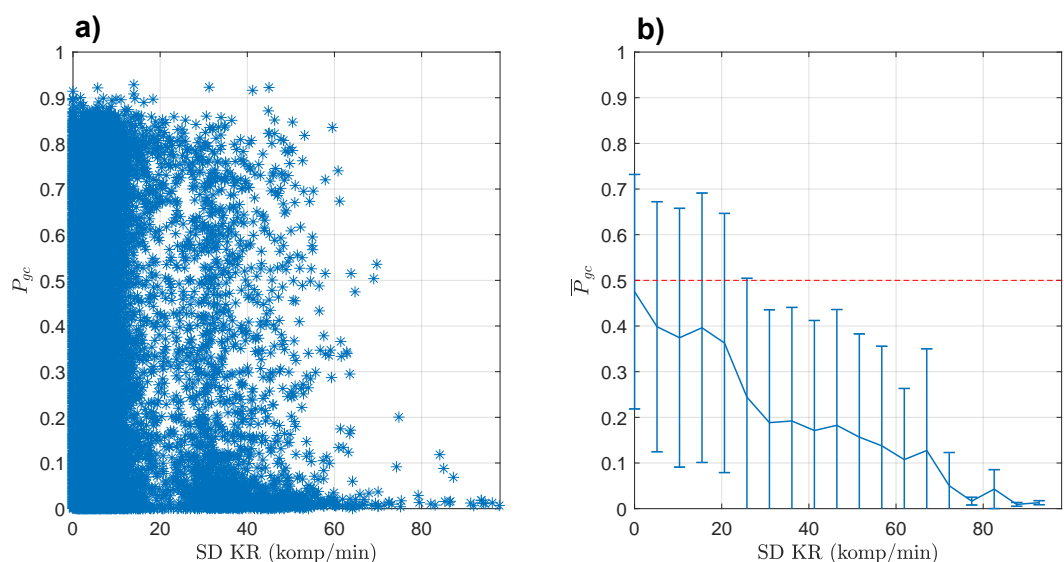
**Figur A.10:** Klassifisering ved bruk av nevralt nettverk.  $P_{gc}$  gir sannsynlighet for at det utføres gunstige kompresjoner. Det er vist i y-akse, mot kompresjonsrate ( $KR$ ) på x-akse. **a)** Alle datapunkt fra treningsett. **b)** Alle datapunkt fra testsett. **c)** Gjennomsnitt og standardavvik av  $P_{gc}$  fra testsett.

## A.4 Utvidet testsett

Her vises fordeling av  $P_{gc}$  mot kvalitetsparametre, med bruk av utvidet testsett. Fordeling av  $P_{gc}$  mot  $SDKD$ ,  $SDKR$  og  $SDKR(TI)$  er vist i henholdsvis figur A.11, A.12 og A.13.

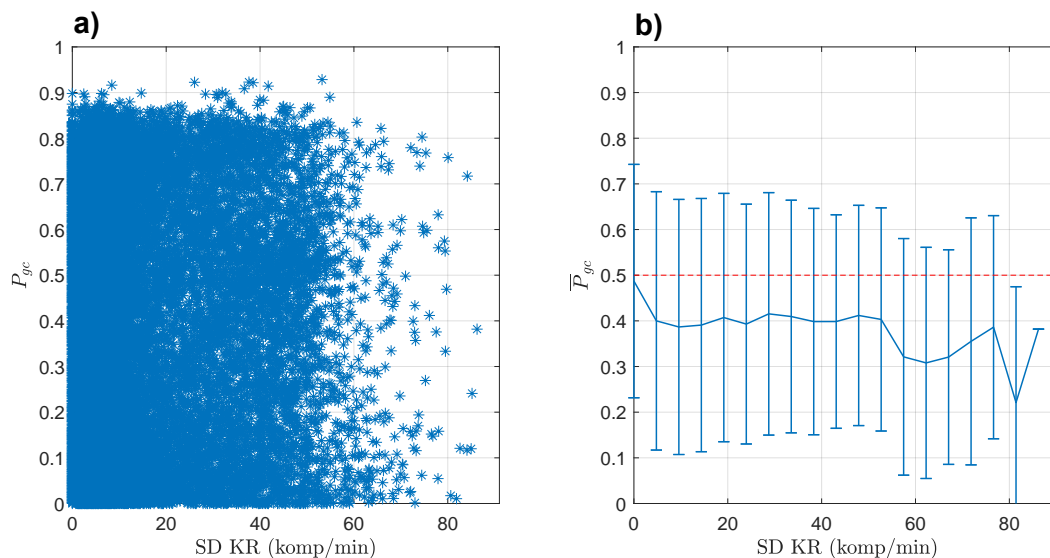


**Figur A.11:** Forhold mellom  $P_{gc}$  og standardavvik for kompresjonsdybde. Datapunkt fra datasett 6, hvor alle datapunkt fra episoder i testsett er plottet. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik av  $P_{gc}$ . Rød stiplet linje markerer grense for hvor datapunkt klassifiseres som gunstige kompresjoner.



**Figur A.12:** Forhold mellom  $P_{gc}$  og standardavvik for kompresjonsrate. Datapunkt fra datasett 6, hvor alle datapunkt fra episoder i testsett er plottet. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik av  $P_{gc}$ . Rød stiplet linje markerer grense for hvor datapunkt klassifiseres som gunstige kompresjoner.

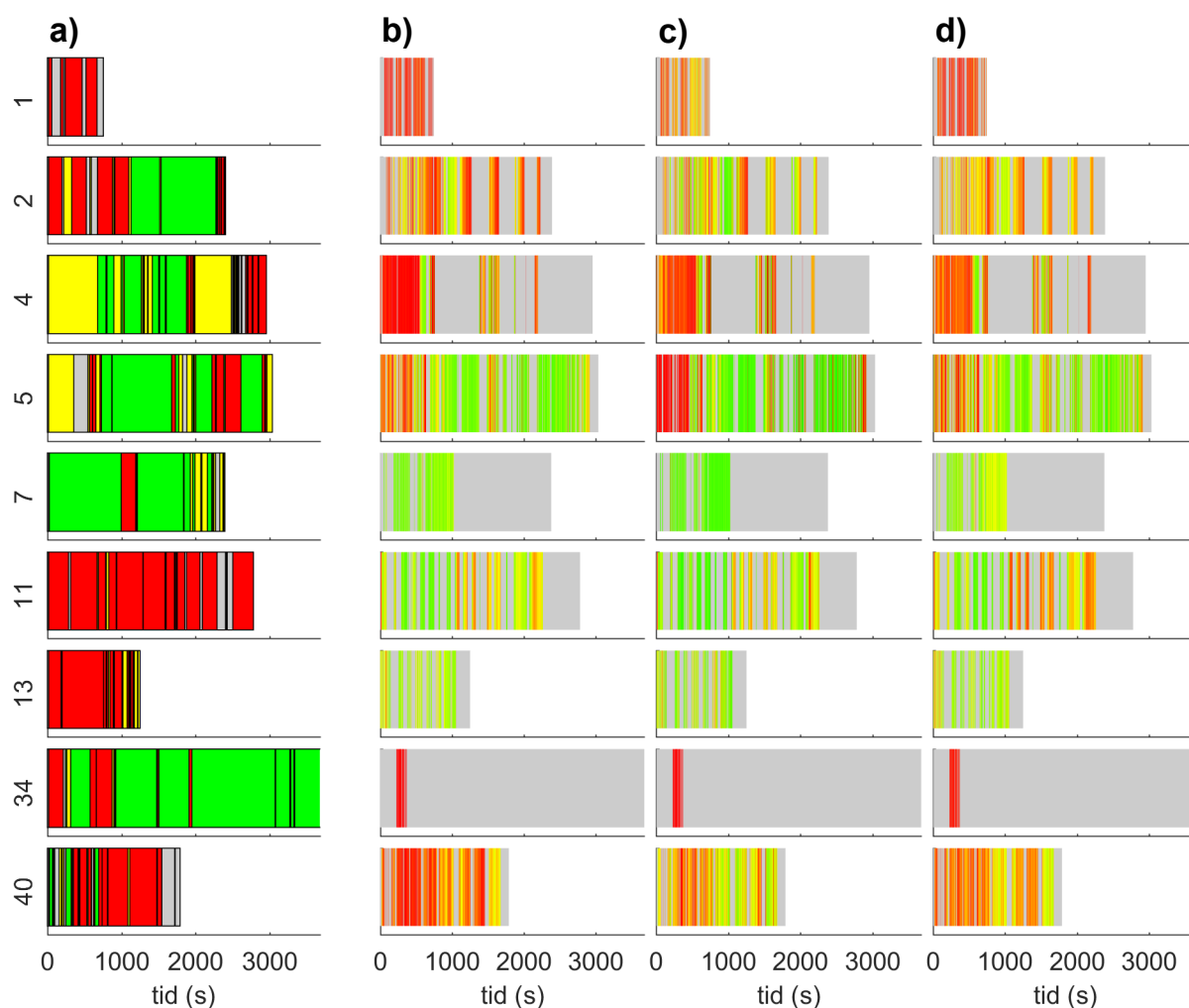
I figur A.11 og A.12 observeres det en synkende trend for  $\bar{P}_{gc}$ , når verdi for kvalitetsparametre øker.



**Figur A.13:** Forhold mellom  $P_{gc}$  og standardavvik for kompresjonsrate basert på impedansmåling. Datapunkt fra utvidet testsett i datasett 6, hvor alle datapunkt fra episoder i testsett er plottet. **a)** Alle datapunkt. **b)** Gjennomsnitt og standardavvik. Rød stiplet linje markerer grense for hvor datapunkt klassifiseres som gunstige kompresjoner.

## A.5 Sammenligning av sannsynlighetsmodeller

Her sammenlignes implementering av alle sannsynlighetsmodeller, basert på alle maskinlæringsmetoder. Det undersøkes hvordan de forskjellige sannsynlighetsmodellene indikerer kvalitet av hjertekompresjoner, for de samme kompresjonsintervallene. Sammenligningen kan også benyttes til å undersøke om endring i pasientens hjerterytme kan forklares av gunstig eller ikke gunstig utførelse av hjertekompresjoner. Visualisering av hjerterytme og  $P_{gc}$ , er vist i figur A.14. Pasientens hjerterytme er vist i **a)**, mot  $P_{gc}$  for logistisk regresjon, ML-metoden og nevralt nettverk, merket med henholdsvis **b)**, **c)** og **d)**. Figuren viser 9 hjerrestanseepisoder fra testsettet, fordelt på rader.



**Figur A.14:** Sammenligning av alle sannsynlighetsmodeller, mot endring av hjerterytme i episoder. Rader er forskjellige hjerrestanseepisoder, hentet fra testsettet. **a)** Hjerterytme over tid; rød - VF/VT, gul - PEA, grå - AS og grønn - PR. **b), c)** og **d)** Viser kompresjonsintervall over tid, hvor kvalitet av behandling er indikert med bruk av  $P_{gc}$  for de forskjellige sannsynlighetsmodellene. **b)** Logistisk regresjon. **c)** ML-metoden. **d)** Nevralt nettverk. Fargekoden for  $P_{gc}$  er vist i figur 4.4.

I figur A.14 observeres det store likhetstrekk mellom sannsynlighetsmodellene, ettersom  $P_{gc}$  gir tilsvarende lik fargeindikasjon for kompresjonsintervallene. I figuren kan det også observeres tilfeller hvor hjerterytmen endres til PR, merket som grønn farge i **a)**, etter det er utført gunstige kompresjoner, merket med grønn farge i **b), c)** og **d)**. Dette kan for eksempel observeres for episode 2, hvor det etter rundt 1000 sekunder utføres hjerstekompresjoner av gunstig kvalitet, vist i **b), c)** og **d)**. I **a)** observeres det da at hjerterytmen endres til PR litt etter 1000 sekunder.



# Vedlegg B

## B.1 Matlab-kode

- **Hente\_datapunkt.m:** Hente ut datapunkt fra kompresjonsintervall, samt beregne  $\Delta P_{des}$  for intervallet. Datapunkt sorteres i grupper, basert på hjerterytme.
- **Sorter\_data\_i\_matriser.m:** Genererer data.mat som inneholder alle datapunkt i en matrise, fordelt til trening- og testsett. data.mat lagres manuelt, og er datasett 1-6, avhengig av hvilket valg som ble benyttet for Qpars og  $\Delta P_{des}$ -metode. Programmet bruker funksjonen data2array.m for å hente ut og sortere datapunkt. Programmet bruker datapunkt som ble lagret ved bruk av Hente\_datapunkt.m.
- **EksperimentML.m:** Utfører eksperiment med bruk av en ML-basert sannsynlighetsmodell. Alle eksperiment blir lagret som .mat-format.
- **EksperimentLR.m:** Utfører eksperiment med bruk av en LR-basert sannsynlighetsmodell. Alle eksperiment blir lagret som .mat-format.
- **EksperimentNN.m:** Utfører databehandling og klargjøring av datasett, før datasett benyttes i Python.
- **Sammenlign\_modeller:** Eksperiment som ble utført lastes inn og brukes til å undersøke klassifiseringsresultat, samt forhold mellom  $P_{gc}$  og kvalitetsparametre. Dette programmet bruker funksjonen Test\_modeller.m
- **LR\_ML\_NN\_sammenligning:** Programmet gir en visuell sammenligning av hjerterytme mot kvalitet av hjertekompresjoner, for alle maskinlæringsmetoder. Programmet bruker datasett lagret fra funksjonen Test\_modeller.m.
- **test\_treningSplitt\_data.m:** Genererer en vektor som representerer alle episode. 70% blir satt til trening, og 30% til test. Fordelingen til trening og test blir lagret i splitt.mat.

Funksjoner:

- **beregn\_kost.m**
- **csv2mat\_pdes\_tab.m**
- **csv2mat\_qpar\_cd\_tab.m**
- **csv2mat\_qpar\_ti\_tab.m**
- **csv2mat\_resp\_tab.m**
- **data2matrise.m**
- **KlassifiserLR.m**
- **KlassifiserML.m**
- **Normaliser\_data.m**
- **normND.m**
- **Panel\_plott\_alle\_modeller.m** (Krever tilleggspakken: Panel)
- **Patch\_matriser.m**
- **PCA\_transformer.m**
- **Plott\_forhold.m**
- **Plott\_forhold\_treningsdata.m**
- **resultat\_klassifisering.m**
- **Sammenheng\_Pgc\_retningslinjer.m**
- **sigmoid.m**
- **Test\_modeller.m**

Fremgangsmåte for å rekonstruere eksperiment:

1. Kjør programmet Hente\_datapunkt.m. Alternativ for Qpars og  $\Delta P_{des}$  må velges (Qpars=1/2,  $\Delta P_{des}$ =1/2/3). Mapper for hvor datapunkt lagres opprettes manuelt. Stien benyttet er: Sortert\_data/Qpars(1/2)/Pdes\_alternativ(1/2/3).
2. Kjør programmet Sorter\_data\_i\_matriser.m. Her må tilsvarende alternativ som i punkt 1 velges. Lagre data.mat. Dette er datasettet.
3. Utfør eksperiment med bruk av EksperimentML.m, EksperimentLR.m eller EksperimentNN.m. Her må det også opprettes mapper for hvor eksperiment skal lagres. Sti benyttet er: Eksperiment(ML/LR)/qpars\_alt(1/2)/Pdes\_alt(1/2/3). For eksperiment med nevralt nettverk benyttes EksperimentNN.m til data-behandling, for så å utføre trening og testing av modell i Python. Modell fra Python lagres og benyttes i Matlab
4. Kjør programmet Sammenlign\_modeller.m. Her oppsummeres resultat fra valgt eksperiment og maskinlæringsmetode.



## B.2 Python-kode

- **EksperimentNN.py:** Utfører eksperiment. Datasett fra Matlab-programmet EksperimentNN.m lagres i samme mappe som dette programmet. Modell fra eksperiment lagres i mappen ModellNN.
- **KlassifiserNN.py:** Nettverkstruktur for nettverk som ga best resultat er implementert. Matlab bruker dette programmet. Kvalitetsparametre sendes inn fra Matlab, og program returnerer sannsynlighet for at kvalitetsparametre er gunstige kompresjoner.
- **KlassifiserNNdatasettFire.py:** Nettverkstruktur for nettverk trent med datasett 4.
- **KlassifiserNNdatasettFem.py:** Nettverkstruktur for nettverk trent med datasett 5.
- **KlassifiserNNdatasettSeks.py:** Nettverkstruktur for nettverk trent med datasett 6.

For å kjøre program i Python må følgende bibliotek være installert:

- Python v3.6
- tensorflow
- numpy
- scipy
- sklearn
- matplotlib
- pandas
- seaborn
- os