

Sammenligning av kardiologisk og algoritmebasert EKG-tolkning på idrettsutøvere: Kan kunstig intelligens forbedre algoritmene?

Bjørn-Jostein Singstad



Oppgave for graden
Master i Elektronikk, informatikk og teknologi
60 studiepoeng

Institutt for Fysikk
Det matematisk-naturvitenskapelige fakultet

UNIVERSITETET I OSLO

Våren 2021

**Sammenligning av
kardiologisk og
algoritmebasert
EKG-tolkning på
idrettsutøvere:
Kan kunstig intelligens
forbedre algoritmene?**

Bjørn-Jostein Singstad

Figurene i dette arbeidet er lisensiert under CC BY-SA 4.0-lisens (<https://creativecommons.org/licenses/by-sa/4.0/>), så sant noe annet ikke er spesifisert. CC BY-SA 4.0-lisensen er kompatibel med definisjonen for "Open Access" og alle figurer som faller inn under denne lisensen er tilgjengelige på denne nettsiden: <https://zenodo.org/record/4445257#.YBez10hKhaQ>

Deler av dette arbeidet ble utført på TSDs servere (Tjeneste for sensitive data), eid av Universitetet i Oslo, drevet og utviklet av TSDs servicegruppe ved Universitetet i Oslo, IT-avdeling (USIT). (tsd-drift@usit.uio.no)

DOI: 10.5281/zenodo.4445257

© 2021 Bjørn-Jostein Singstad

Sammenligning av kardiologisk og algoritmebasert EKG-tolkning på idrettsutøvere:

Kan kunstig intelligens forbedre algoritmene?

<http://www.duo.uio.no/>

Trykk: Reprosentralen, Universitetet i Oslo

Sammendrag

Hjertestans er den vanligste årsaken til plutselig død blant idrettsutøvere. Dette skyldes ofte arvelige hjertesykdommer som blant annet kan oppdages på et elektrokardiogram (EKG). I denne studien ble to problemstillinger undersøkt: (1) Om tolkningsalgoritmene i en klinisk anvendt elektrokardiograf er bedre tilpasset EKG fra den generelle befolkning enn idrettsutøvere? (2) Om moderne metoder, som kunstig intelligens (KI), kunne brukes til å forbedre dagens kliniske anvendte tolkningsalgoritmer slik at de kan bli bedre kalibrert for idrettsutøvere?

For å undersøke (1) ble det registrert EKG fra 28 idrettsutøvere ved bruk av en moderne elektrokardiograf. EKGene ble tolket av tolkningsalgoritmen i elektrokardiografen, og sammenlignet med en kardiologs tolkning. Tolkningsalgoritmen klassifiserte 46% av EKGene som normale, 25% i grenseområdet og 29% som unormale. Kardiologen fulgte retningslinjene fra *International Criteria*, og klassifiserte 93% som normale og 7% i grenseområdet. Resultatene viser at tolkningsalgoritmen feilvurderer noen av områdene i EKGene. Dette ga i noen tilfeller alvorlige feilklassifiseringer, som akutt hjerteinfarkt (STEMI).

For å undersøke (2) ble et stort åpent datasett av EKGene fra den generelle befolkning tatt i bruk. Ni ulike modeller ble trent, og validert på datasettet, deretter ble fire prediksjoner fra de to beste modellene forklart ved bruk av forklarbar KI. Den beste modellen fikk en PhysioNet/CinC Challenge skår $= 0,512 \pm 0,006$ og en F_1 -skår $= 0,405 \pm 0,008$ ved 10-ganger kryssvalidering. Modellarkitekturene lar seg forklare ved bruk av forklarbar KI. Forklarbarheten er viktig for modellens kliniske relevans. For videre arbeid med disse problemstillingene trengs tilgang til flere EKG fra idrettsutøvere.

Forord

Denne oppgaven er utført ved Fysisk institutt, Universitetet i Oslo og i samarbeid med Oslo Universitetssykehus. Først vil jeg få takke mine veiledere; Ørjan Grøttem Martinsen, Øyvind Haugen Lie, Christian Tronstad og Knut Gjesdal, som har hjulpet meg og svart på spørsmål både sent og tidlig. En ekstra takk går til Christian, som hjalp meg med min aller første vitenskapelige publikasjon i forbindelse med denne oppgaven. I tillegg må jeg takke Knut Gjesdal nok en gang for hans tolkning av de innsamlede EKGene, som er en stor del av grunnlaget for denne oppgaven.

Jeg må også få takke Dan Oddvar Strand og General Electric for et flott samarbeid og for at jeg fikk låne en splitter ny elektrokardiograf og tilgang til all nødvendig programvare.

Takk til kjæresten min, Stine Eriksen Hammer, som jeg har delt hjemmekontor med det siste året. Takk for at du har lest korrektur, heiet på meg og dratt meg med ut når jeg har holdt på å gro fast i kontorstolen. Når du nå har holdt ut med meg hele dette året bør resten av livet gå som en lek!

Til slutt vil jeg rette den største takken til de 28 idrettsutøverne som sa ja til å delta i studien. Uten dere ville ikke denne studien vært mulig å gjennomføre - Tusen takk!

Februar 2021

Bjørn-Jostein Singstad

Innhold

Forord	iii
Figurer	vi
Tabeller	viii
Forkortelser	ix
1 Innledning	1
1.1 Plutselig hjertestans blant idrettsutøvere	1
1.2 Elektrokardiografisk tolkning	1
1.3 Idrettskardiologi	2
1.4 Kunstig intelligens og EKG-tolkning	2
1.5 Hensikten med oppgaven	2
1.6 Oppgavens oppbygning	3
2 Teori	5
2.1 Hjertet	6
2.1.1 Hjertets anatomi	6
2.1.2 Hjertets elektriske system	7
2.2 Elektrokardiografen og elektrokardiogrammet	7
2.2.1 12-avlednings hvile-EKG	8
2.2.2 Karakteristiske bølger, takker og segmenter i et EKG	11
2.2.3 Støy i EKG-signaler	12
2.2.4 Elektrokardiografiske tolkningsalgoritmer	12
2.3 Idrettshjerte	13
2.3.1 Patologi som ligner på idrettshjertes	14
2.3.2 Normale treningsforandringer	17
2.3.3 Treningsrelaterte forandringer i grenseområdet	21
2.3.4 Atletisk unormal forandring av hjertet	22
2.3.5 Screening av idrettsutøvere	24
2.4 Sannsynlighet og diagnostiske tester	25
2.5 Kunstig intelligens	27
2.5.1 Veiledet læring	27
2.5.2 Dype nevralt nettverk	32
2.5.3 Konvolusjonelle nevralt nettverk	34
2.5.4 Ensemble modeller	37
2.5.5 Maskinlæring og datamengder	39
2.5.6 Forklarbar kunstig intelligens	40

3	Metode	43
3.1	EKG-registreringer på idrettsutøvere	43
3.1.1	Utstyr	44
3.1.2	Planlegging, rekruttering og utføring av registreringene	46
3.1.3	Analyse av innhentede EKG-data	48
3.1.4	Skåringsfunksjoner	49
3.2	Maskinlæring på EKG fra den generelle befolkning	50
3.2.1	Data	50
3.2.2	Preprosessering av data	52
3.2.3	Skåringsfunksjoner	53
3.2.4	Trening og valideringsdata	54
3.2.5	Konvolusjonelle nevrale nettverk	55
3.2.6	Ensemble-modell	60
3.2.7	Forklaringsmodeller	61
4	Resultater	63
4.1	Tolkning av EKG fra idrettsutøvere	63
4.1.1	Deltakerne i studien	64
4.1.2	GE Marquette SL12s tolkning	65
4.1.3	Kardiologens tolkning	66
4.1.4	Sammenligning mellom Marquette SL12 og kardiologens tolkning	67
4.2	Resultater fra maskinlæring på EKG fra den generelle befolkning	69
4.2.1	Forklarbar KI for <i>Encoder-modellen</i>	70
4.2.2	Forklarbar KI for <i>Ensemble-modellen</i>	72
4.3	Tilgjengelig kode	73
5	Diskusjon	75
5.1	Tolkning av EKG fra idrettsutøvere	75
5.1.1	Sammenligning av tolkning fra Marquette SL12 og kardiologen	75
5.1.2	Beregning av intervaller og segmenter	76
5.1.3	Feilplassering av elektroder	77
5.2	Maskinlæring på EKG fra den generelle befolkning	78
5.2.1	Sammenligning av maskinlæringsmodellene	78
5.2.2	Begrensninger og mulige feilkilder	79
5.3	Forklarbar KI på EKG	79
5.4	Maskinlæring på EKG fra idrettsutøvere	80
6	Konklusjon	83
	Bibliografi	84
	Appendiks	100

Figurer

2.1	Illustrasjon av blodets flyt gjennom hjertet	6
2.2	Hjertets elektriske ledningssystem	7
2.3	Trådgalvanometeret	8
2.4	Plassering av EKG-avledningene: <i>I</i> , <i>II</i> og <i>III</i>	9
2.5	Plassering av EKG-avledningene: aVL, aVF og aVR	9
2.6	Plassering av de prekordiale EKG-avledningene	10
2.7	De karakteristiske EKG-bølgene; P, Q, R, S, T og U	11
2.8	Forskjellige former for støy i et EKG-signal	12
2.9	Graden av statisk og dynamisk arbeidskrav for ulike idretter	13
2.10	Idrettshjerte vs patologi	14
2.11	En historisk oversikt over internasjonale EKG-tolkningskriterier	15
2.12	De siste internasjonale, idrettsspesifikke EKG-tolkningskriteriene	16
2.13	Sokolow-Lyon kriteriene for venstre ventrikkelhypertrofi	17
2.14	Sokolow-Lyon kriteriene for høyre ventrikkelhypertrofi	18
2.15	Eksempel på tidlig repolarisering med og uten J-bølge	19
2.16	Eksempel på første grads AV-blokk	20
2.17	Hexaksialt referansesystem for beregning av hjerteaksens dreining	21
2.18	AUC-kurver for to ulike modeller	26
2.19	Undertilpasning, overtilpasning og optimal tilpasning	28
2.20	Bias-varians kompromisset	29
2.21	En illustrasjon av et kunstig nevron	30
2.22	Lineær, Sigmoid, ReLU og tanh aktivering	31
2.23	Nevralt nettverk med et skjult lag	31
2.24	En illustrasjon av backpropagation	33
2.25	En 1-dimensjonal konvolusjonell operasjon	34
2.26	En maks pooling-operasjon	35
2.27	En global gjennomsnittlig pooling-operasjon	35
2.28	En illustrasjon av arkitekturen til en tilfeldig beslutningsskog	38
2.29	Data vs ytelse for maskinlæringsmodeller og dyplæringsmodeller	39
2.30	Prinsippskisse for Transfer Learning	40
2.31	Eksempel på forklarbar kunstig intelligens	41
2.32	En intuitiv forståelse av en lokal forklaringsmodell	42
3.1	GE MAC VU360 TM elektrokardiograf	44
3.2	Ambu [©] BlueSensor TM Q	45
3.3	Flytskjema for søknadsprosess	47
3.4	Diagnoser i PhysioNet/CinC Challenge 2020-utviklingssettet	51
3.5	Vektallmatrix for PhysioNet/CinC Challenge-skår	53

3.6	Parametere og klasser i den <i>regelbaserte algoritmen</i>	56
3.7	Overordnet arkitektur for de CNN-baserte modellene	57
3.8	Detaljert arkitektur for FCN	58
3.9	Detaljert arkitektur for Encoder	59
3.10	Oppdeling av klynger til <i>Ensemble-modellen</i>	61
4.1	Grafisk sammendrag av EKG-målingene gjort på idrettsutøverne	63
4.2	Boks-diagram over antall treningstimer for idrettsutøverne	64
4.3	Klassifisering av Marquette TM SL12	65
4.4	Diagnostisering av kardiologen.	66
4.5	Forvirringsmatrise 1 - Marquette TM SL12 vs Kardiolog	67
4.6	Forvirringsmatrise 2 - Marquette TM SL12 vs Kardiolog	68
4.7	10 ganger kryssvaliderte resultater for 9 ulike modeller.	69
4.8	Forklaring av sinustakykardi klassifisert av <i>Encoder-modellen</i>	70
4.9	Forklaring av T-bølgeinversjon klassifisert av <i>Encoder-modellen</i>	71
4.10	Forklaring av sinustakykardi klassifisert av <i>Ensemble-modellen</i>	72
4.11	Forklaring av T-bølgeinversjon klassifisert av <i>Ensemble-modellen</i>	73
5.1	Støy i et av EKGene fra PhyioNet/CinC-datasettet	77
5.2	Støy i et av EKGene fra PhyioNet/CinC-datasettet	79
5.3	En foreslått modellarkitektur for fremtidig arbeid	81

Tabeller

2.1	Eksempel på bruk av en dropout-operasjon i et nevralt nettverk	36
3.1	Oversikt kildene til utviklingsdataene brukt til å trene maskin- læringsmodellene	50
3.2	Åtte CNN-modeller utviklet i en tidligere studie	55
3.3	Eksempel på try-except-struktur i den <i>regelbaserte algoritmen</i> . .	57

Forkortelser

ARVC	Arrhythmogenic Right Ventricular Cardiomyopathy
AUC	Area under the curve
AV	Atrioventrikulær
CinC	Computing in Cardiology
CNN	Convolutional Neural Networks
DCM	Dilated Cardiomyopathy
EKG	Elektrokardiogram
FN	Falske negative
FP	Falske positive
GDPR	General Data Protection Regulation
GPU	Graphics processing unit
HCM	Hypertrophic Cardiomyopathy
HR	Hjerterate / hjertefrekvens
HVH	Høyre ventrikkelhypertrofi
iRBBB	Incomplete Right Bundle Branch Block
KI	Kunstig intelligens
LBBB	Left Bundle Branch Block
LIME	Local interpretable model-agnostic explanation
LVNC	Left ventricular noncompaction
MLP	Multilayer Perceptron
mV	Millivolt
NSD	Norsk senter for forskningsdata
PC	Personal Computer
RBBB	Right Bundle Branch Block

REK	Regional etisk komite
ReLU	Rectified Linear Unit
ROC	Receiver operating characteristic
SN	Sanne negative
SP	Sanne positive
STEMI	ST-elevet myokard infarkt
TSD	Tjenester for sensitive data
UiO	Universitet i Oslo
USB	Universal Serial Bus
VES	Ventrikulære ekstrasystoler
XML	Extensible Markup Language

1. Innledning

1.1 Plutselig hjertestans blant idrettsutøvere

Plutselig hjertestans er den hyppigste årsaken til plutselig død blant unge idrettsutøvere [1], [2], og skjer oftere enn hos jevnaldrende ikke-atleter [3], [4]. Antallet idrettsutøvere på verdensbasis som dør av plutselig hjertestans hvert år er usikkert, men estimerer varierer mellom $0,3 / 100\,000$ og $4,4 / 100\,000$ per år [4]–[6]. Noen idretter gir høyere risiko enn andre. For eksempel er risikoen for plutselig hjertestans blant basketballspillere rapportert til å være $32,3 / 100\,000$ per år [1]. Selv om forekomsten generelt er lav er dette veldig tragiske hendelser, som rammer ellers friske unge individer.

Det er flere sykdommer som kan føre til plutselig hjertestans [7]. Noen av disse sykdommene har likhetstrekk med normale forandringer av hjertet som følge av trening [8]. Dette gjør det ekstra vanskelig å skille hjertet til en toppidrettsutøver fra et sykt hjerte. Dessuten kan hard trening være med på å trigge plutselig hjertestans dersom personen har en underliggende hjertesykdom [9].

1.2 Elektrokardiografisk tolkning

En elektrokardiograf registrerer den elektriske aktiviteten i hjertet fra elektroder plassert på huden. Resultatet av en slik registrering er et elektrokardiogram (EKG). EKGGet er en grafisk fremstilling av den målte elektriske aktiviteten i hjertet med hensyn på tid. En av fordelene med denne metoden er at den ikke er invasiv, og ikke har noen kjente bivirkninger. En av utfordringene derimot er at EKGGet kan være vanskelig å tolke riktig, derfor kan tolkningen være tidkrevende, og kreve høy grad av ekspertise [10]. I tillegg vil tolkning av et EKG fra en idrettsutøver kreve spesialisering med dedikert trening i idrettskardiologi [11].

Mange av de moderne og klinisk anvendte elektrokardiografene er i dag utstyrt med et innebygd tolkningsprogram. Tolkningsprogrammet gjør en analyse av EKGGet, og skriver ut tolkningstekster som antyder forskjellige sykdommer. Studier viser flere begrensninger ved de automatiske tolkningsalgoritmene [12], [13]. Blant annet feiltolkes ofte pasienter med arytmier, ledningsforstyrrelser og pacemakerrytme [12]. Feiltolkningene fra de automatiske tolkningsalgoritmene gjør at legene eller kardiologene må lese over EKGene for å kvalitetssikre de. Likevel viser en studie av Morisbak, B.& Gjesdal, K. (1999) at feil fra tolkningsalgoritmen, sammenlignet med tilfeller uten støtte fra tolkningsalgoritmen, ikke ga noen signifikant forskjell i legenes konklusjon [14]. Samme studie viser også at EKGene, som er merket med korrekt tolkning av tolkningsalgoritmene gir

det en økning i korrekte tolkninger fra legene som leser over, sammenlignet med tilfeller uten støtte fra tolkningsalgoritmen.

1.3 Idrettskardiologi

Det eksisterer i dag spesifikke kriterier for tolkning av EKG fra idrettsutøvere [15]. Tolkningskriteriene tar sikte på å fange opp syke idrettsutøvere med risiko for plutselig hjertestans, men også utpeke færre falske positive. Dette er viktig fordi mange falske positive vil representere en tung belastning for helsevesenet, og en mental belastning for den enkelte idrettsutøver [6].

Tolkningskriteriene for idrettsutøvere er i stadig utvikling, og Berge et al. 2015 [16] viste hvordan Seattle kriteriene fra 2013 [17] senket antallet unormale EKG-funn fra 29,3% (spesifiserte European society of Cardiology's anbefalinger [18]) til 11,2%. Videre har Refined Criteria (2014) [19] vist seg å senke antallet falske positive ytterligere, og samtidig oppdage like mange av de syke [11], [15].

En studie har sammenlignet tolkningsalgoritmer med visuelle målinger på EKG fra idrettsutøvere, og identifiserte begrensninger ved algoritme-baserte EKG tolkninger på idrettsutøvere [20]. En av hypotesene i denne masteroppgaven er at disse begrensningene også er til stede for flere tolkningsalgoritmer. Det er grunn til å tro at en tolkningsalgoritme, bedre kalibrert for idrettsutøvere, vil kunne gjøre flere riktige tolkninger. Videre vil dette hjelpe legen til å ta flere riktige avgjørelser slik Morisbak, B. & Gjesdal, K. (1999) viste ved EKG-tolkning av den generelle befolkning [14].

1.4 Kunstig intelligens og EKG-tolkning

Tidligere studier har vist at kunstig intelligens (KI) kan brukes til å stille diagnoser ut i fra et 1-avlednings EKG [21]–[23], i en av studiene kunne KI måle seg med kardiologers klassifisering [23]. De siste årene har KI også vist seg å kunne fungere på 12-avlednings EKG [24]–[27], men ennå er det ingen som har forsøkt å bruke KI på EKG fra idrettsutøvere.

1.5 Hensikten med oppgaven

Hensikten med denne oppgaven er å undersøke to problemstillinger:

1. Er tolkningsalgoritmene i en klinisk anvendt elektrokardiograf bedre tilpasset EKG fra den generelle befolkning enn idrettsutøvere?
2. Kan moderne metoder som KI, brukes til å forbedre dagens kliniske anvendte tolkningsalgoritmer slik at de blir bedre kalibrert for idrettsutøvere?

I denne studien er EKGer fra toppidrettsutøvere samlet inn for å vurdere ytelsen til eksisterende tolkningsalgoritmer. Ytelsen til elektrokardiografens tolkningsalgoritme blir vurdert ved å sammenligne tolkningen med en kardiologs tolkning. Kardiologen, som har erfaring innen idrettskardiologi, har tolket EKGene både fra et idrettskardiologisk perspektiv og fra et ikke-idrettskardiologisk perspektiv. På denne måten blir det mulig å identifisere hvorvidt tolkningsalgoritmen er god eller dårlig til å tolke EKG fra idrettsutøvere.

Tilgangen til EKG'er fra idrettsutøvere er en utfordring, da et høyt antall EKG'er behøves for å trene en modell basert på KI. EKGene samlet inn i denne studien er for få til å trene eller validere en KI-modell. I stedet for å trene og validere modellene på EKG'er fra idrettsutøvere er modellene i denne studien basert på 12-avlednings EKG fra den generelle befolkning. EKGene kommer fra PhysioNet/Computing in Cardiology (CinC) Challenge 2020. Datasettet består av 43 101 EKG fra flere kilder, og har en stor variasjon i ulike diagnoser [28]. Målet er at modellarkitekturene og erfaringen vil kunne brukes videre på EKG'er fra idrettsutøvere i senere studier med et tilstrekkelig antall EKG'er.

For at en modell basert på KI skal kunne brukes til å stille diagnoser eller som et beslutningsverktøy for klinisk personell, er det viktig at modellen gir en forklaring av prediksjonen og beslutningsgrunnlaget. Det er lovfestet i General Data Protection Regulation (GDPR), artikkel 22, at dersom algoritmen tar beslutninger på vegne av mennesker, har personen rett til ikke å være underlagt en avgjørelse utelukkende basert på automatisert behandling [29].

Avanserte KI-modeller har blitt sett på som svarte bokser fordi det tidligere ikke har vært mulig å få en forklaring på prediksjonene [30]–[32]. Dette har ført til en ny disiplin, som heter forklarbar KI. Forklarbar KI kan forklare prediksjonen og beslutningsgrunnlaget til en ellers uforklarlig modell. I denne masteroppgaven vil noen av dagens mest avanserte former for KI brukes til å klassifisere EKGene. En forutsetning for at modellene skal ha klinisk relevans, uavhengig om de skal brukes på idrettsutøvere eller den generelle befolkning, er at modellene er forklarbare.

Det overordnede målet med oppgaven er å belyse problematikken rundt plutselig hjertestans hos idrettsutøvere, og vise hvordan nye metoder kan forbedre diagnostikken av denne gruppen. Forskningsfeltet er viktig fordi bedre sensitivitet kan forbygge tragiske hendelser, og bedre spesifisitet kan forhindre unødvendige og kostbare videre utredninger. Studier viser at evnen til å fange opp utøvere som senere har fått hjertestans, samt utelukke de som ikke har risiko for hjertestans er så dårlig at screening ikke anbefales i Norge [33]. En forbedret EKG-tolkningsalgoritme, spesielt kalibrert for idrettsutøvere vil kunne heve nivået, og hjelpe den lite erfarne EKG-tolker til å finne de som skal henvises videre, og utelukke de friske.

1.6 Oppgavens oppbygning

Det er to deler som belyses gjennom hele oppgaven: EKG fra idrettsutøvere, og KI på EKG fra den generelle befolkning.

I teoridelen blir det gitt grunnleggende bakgrunnsinformasjon. Først gis det en generell introduksjon om hjertet, elektrokardiografen og EKG'en. Deretter gis en innføring i idrettshjertet, og tolkningen av EKG fra idrettsutøvere. Til slutt introduseres KI, og noen av de sentrale temaene som ligger under dette begrepet.

I metoddelen blir det redegjort for hva slags utstyr og framgangsmåter som ble brukt i denne oppgaven. Dette omfatter planleggingen og innsamlingen av EKG fra idrettsutøvere og utviklingen samt treningen av ni ulike modeller.

I resultatdelen presenteres aggregerte data fra idrettsutøverne som deltok i studien. I tillegg sammenlignes EKG-tolkningene fra elektrokardiografens tolkningsalgoritmer med kardiologens tolkninger. I den andre delen av resultatene

sammenlignes de ni modellene, som er trent på data fra den generelle befolkningen, basert på ulike skåringsalgoritmer. Til slutt presenteres fire resultater fra forklaringsmodellene, som er brukt til å forklare prediksjonen fra to av modellene utviklet i denne masteroppgaven.

I diskusjonsdelen blir resultatene oppsummert, diskutert og sammenlignet med andre studier. I tillegg drøftes begrensninger ved metodene i studien.

Til slutt, i konklusjonsdelen, blir det gitt en oppsummering av oppgavens betydning samt forslag til implementering og videre arbeid. I tillegg består oppgaven av 7 appendiks fra A til G. Appendiks A består av konfigurasjonen, som ble brukt på GE MAC VU360TM og MarquetteTM SL12 under datainnsamlingen. Appendiks B består av et eksempel på en metadata-fil fra en av pasientene i PhysioNet/CinC Challenge-datasettet. Appendiks C viser 10-ganger kryssvalidert skår på treningsdataene. Appendiks D viser en forvirringsmatrise som sammenligner diagnostiseringen fra MarquetteTM SL12 og kardiologen. Appendiks E består av Python-implementeringen av alle de omtalte modellarkitekturene i denne oppgaven. Appendiks F viser utregningene for makroskårene, basert på sammenligningen mellom MarquetteTM SL12 og kardiologens tolkning. Appendiks G inneholder artikkelen som ble skrevet, og publisert i *Computation in Cardiology 2020*. Appendiks H viser posteren som ble presentert under *Computation in Cardiology-konferansen 13-16. september 2020* (online).

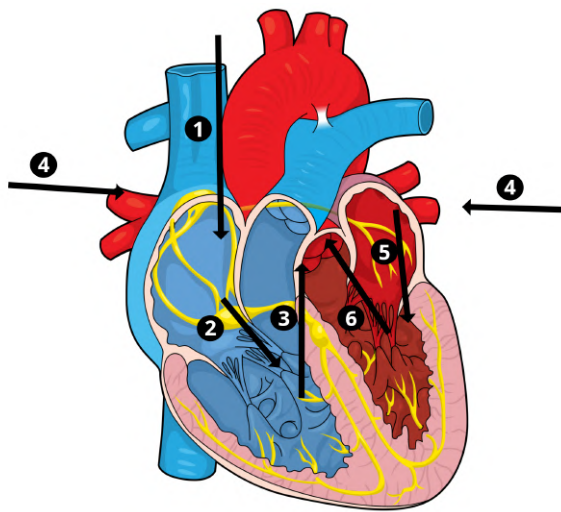
2. Teori

I dette kapitlet gis det en innføring i de mest sentrale temaene og metodene, som senere vil bli brukt i denne oppgaven. Først gis det en kort introduksjon til hjertets oppbygning og hjertets elektriske ledningssystem. Deretter blir det redegjort for hva et EKG er, og hvordan det måles og tolkes. Videre blir idretts-hjertet introdusert, samt idrettskardiologiske tolkningskriterier. Til slutt blir det redegjort for noen sentrale begreper innen medisinsk diagnostikk og statistikk før kunstig intelligens og forklarbar kunstig intelligens blir forklart.

2.1 Hjertet

2.1.1 Hjertets anatomi

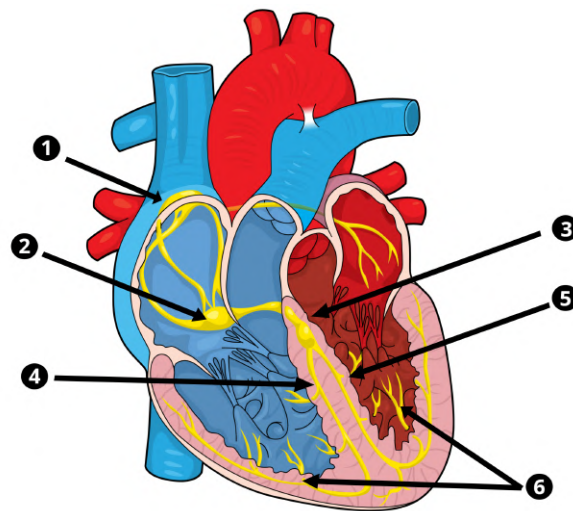
Hjertet er en muskel med fire kammere. Det er to forkammere, som også kalles for atriene, og to hovedkammere, som blir kalt ventriklene. Blodet som strømmer til hjertet fra kroppen eller lungene kommer alltid først til atriene, og renner derfra videre til ventriklene. Ventriklene pumper blodet videre til lungekretsløpet og kroppskretsløpet [34]. Figur 2.1 viser blodets flyt gjennom hjertet. Atriens funksjon er delvis å pumpe blod til ventriklene, men også å være et reservoar for blodet som strømmer tilbake etter sammentrekningen av ventriklene.



Figur 2.1: Illustrasjon av blodets flyt gjennom hjertet. (1) Det venøse blodet fra kroppen kommer inn til hjertet i høyre atrie. (2) Fra høyre atrie renner blodet videre til høyre ventrikkel. (3) Høyre ventrikkel pumper ut blod til lungekretsløpet hvor blodet blir tilført oksygen. (4) Det oksygenrike blodet strømmer til fra lungekretsløpet og videre til venstre atrie. (5) Blodet renner videre fra venstre atrie til venstre ventrikkel. (6) Til slutt blir blodet pumpet ut i arteriene, og ut i kroppen før det igjen kommer tilbake til høyre atrie, og syklusen starter på nytt. Figuren er laget i Mind the Graph (<https://mindthegraph.com/>) og arbeidet er lisensiert under CC BY-SA 4.0 [35].

2.1.2 Hjertets elektriske system

Atriene og ventriklens finkoordinerte sammentrekninger styres av en elektrisk impuls, som ledes gjennom hjertet. Den elektriske impulsen stimulerer muskelfibrene, som igjen fører til sammentrekning av musklene rundt og inni hjertekamrene. Den elektriske impulsen initieres i sinusknuten og ledes fra toppen av atriene, gjennom atriene og ned til den atrioventrikulære (AV) knuten. Dette fører til sammentrekningen av atriene. I AV-knuten oppstår det vanligvis en forsinkelse av den elektriske impulsen. Dette gjør at atriene får tid til å fylle ventriklene med blod [36]. Videre blir den elektriske impulsen ledet fra AV knuten via His-bunten [37], hvor impulsen forgrener seg ut i høyre og venstre gren, og deretter ut til ventriklene via Purkinje-fibrene [36]. Dette setter i gang sammentrekningen av ventriklene som igjen forsyner kropp- og lungekretsløpet med blod. Figur 2.2 illustrerer de viktigste delene av det elektriske ledningssystemet i hjertet.



Figur 2.2: Hjertets elektriske ledningssystem. 1: Sinusknuten, 2: Den atrioventrikulære knuten, 3: His-bunten, 4: Høyre gren, 5: Venstre gren, 6: Purkinjefibrene. Figuren er laget i Mind the Graph (<https://mindthegraph.com/>), og arbeidet er lisensiert under CC BY-SA 4.0 [35].

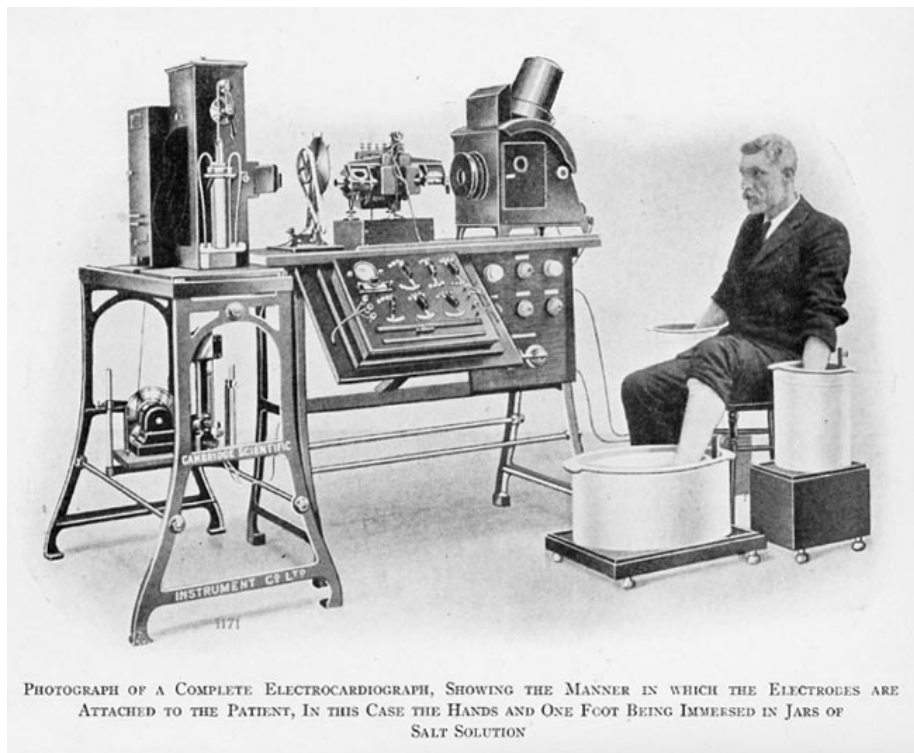
2.2 Elektrokardiografen og elektrokardiogrammet

En elektrokardiograf er et instrument som måler hjertets elektriske aktivitet. Elektrokardiografen benytter seg vanligvis av mellom to til ti elektroder som festes på huden. Elektrodene klarer å fange opp de små periodiske elektriske signalene som brer seg fra sinusknuten og utover i hjertet, men også i kroppen og ut til huden. Målingen av de elektriske signalene over tid, danner en karakteristisk kurve som kalles et EKG.

Det første EKGet fra et intakt menneskelig hjerte, ble målt av Augustus D. Waller i 1887 [38], [39]. Likevel er det Willem Einthoven som regnes som elektrokardiografiens far. I perioden 1901 til 1904 oppfant han trådgalvanometeret.

Noen år senere kom trådgalvanometeret på markedet og ble gjort tilgjengelig for kliniske applikasjoner. Et av de første kommersielle trådgalvanometrene er vist i figur 2.3. Trådgalvanometrene var en tidlig utgave av den moderne elektrokardiografen slik man kjenner den i dag [39], [40]. Senere beskrev Willem Einthoven også et system for å plassere elektrodene på kroppen, som vi i dag kjenner som Einthovens triangel [41]. I tillegg navnga han de karakteristiske bølgene og takene i EKG-et (P,Q,R,S,T og U) [42]. Til sammen ga dette Willem Einthoven nobelprisen i medisin i 1924 for “oppdagelsen av mekanismen til elektrokardiogrammet” [39].

I dag er elektrokardiografen mye brukt blant annet på sykehus, ambulanse og legevakt. Tall fra National Ambulatory Medical Care viser at det ble utført over 40 millioner EKG-undersøkelser i USA i 2015 [43]. To av de åpenbare fordelene, som gjør elektrokardiografen til et mye brukt instrument, er at den er ikke-invasiv og enkel å bruke sammenlignet med andre diagnostiske metoder, som ekkokardiografi og MRI av hjertet.



Figur 2.3: Trådgalvanometeret som Willem Einthoven utviklet. Apparatet var forløperen til elektrokardiografen slik vi kjenner den i dag. Figuren er hentet fra [44], og er publisert under lisensen CC0 [45]

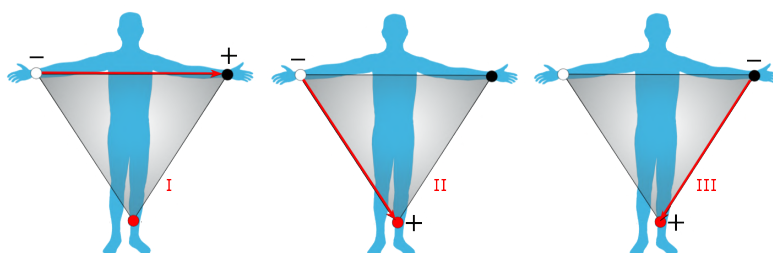
2.2.1 12-avlednings hvile-EKG

Ved sykehus og i andre kliniske sammenhenger er 12-avlednings hvile-EKG, med opptakslengde på ti sekunder, ofte standard protokoll. Et hvile-EKG er et EKG som utføres mens personen det måles på er i hvile. Standard prosedyre

er at personen skal ligge, mens registreringen varer i ti sekunder. 12-avlednings EKGet kobles opp ved å feste ti elektroder på kroppen i en bestemt formasjon.

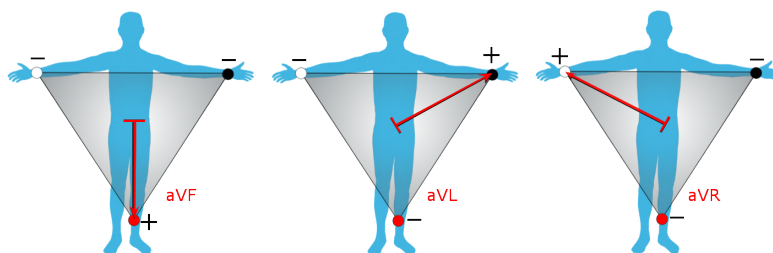
Plassering av elektroder i et 12-avlednings EKG

Fire av de ti elektrodene i et 12-avlednings hvile-EKG plasseres etter retninglinjene ved håndleddene og på ankene. Tre av de seks elektrodene måler spenningen parvis mellom seg slik figur 2.4 viser. Disse tre avledningen kalles vanligvis *I*, *II* og *III*.



Figur 2.4: Til venstre i figuren vises en illustrasjon av avledning *I*, som måles fra høyre arm til venstre arm. I midten av figuren vises en illustrasjon av avledning *II*, som måles fra høyre arm til venstre ben. Til høyre i figuren vises en illustrasjon av avledning *III*, som måles fra venstre arm til venstre ben. Figuren er en tilpasset utgave av illustrasjonen hentet fra [46], av Npatchett, 2015. CC BY-SA 4.0 [35].

De tre neste avledningene beregnes fra de samme tre elektrodene, som blir brukt til å måle avledning *I*, *II* og *III*. Disse avledningene kalles *aVF*, *aVL* og *aVR*. En illustrasjon av disse tre avledningene er vist i figur 2.5, og utregningene er gitt ved formel 2.1, 2.2 og 2.3 hvor LL = venstre bein, LA = venstre arm og RA = høyre arm.



Figur 2.5: Til venstre i figuren vises en illustrasjon av hvordan avledning *aVF* beregnes ut i fra måling fra venstre arm, høyre arm og venstre ben. I midten av figuren vises det hvordan avledning *aVL* beregnes ut i fra måling fra høyre arm, venstre ben og venstre arm. Til høyre i figuren vises en illustrasjon av hvordan avledning *aVR* beregnes ut i fra måling fra venstre arm, venstre ben og høyre arm. Figuren er en tilpasset utgave av illustrasjonen hentet fra [46], av Npatchett, 2015. CC BY-SA 4.0 [35].

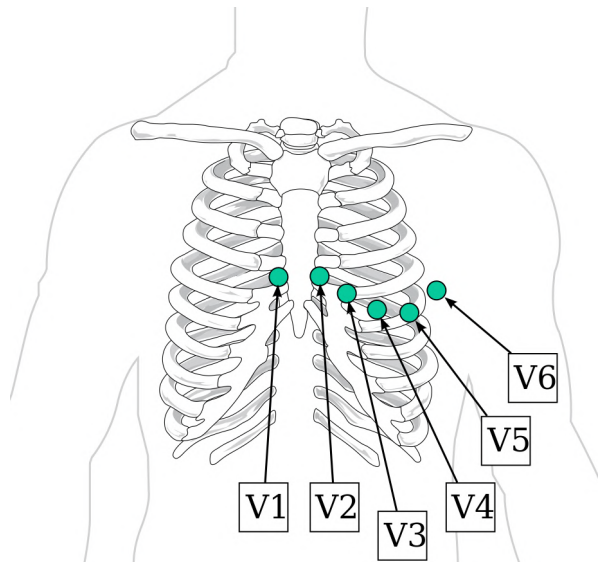
Elektroden festet til høyre ben fungerer som jord eller nullpunkt i den elektriske målekretsen og er med på å redusere forstyrrelsene sett fra EKG-forsterkeren [47].

$$aVF = LL - \frac{1}{2}(RA + LA) \quad (2.1)$$

$$aVL = LA - \frac{1}{2}(RA + LL) \quad (2.2)$$

$$aVR = RA - \frac{1}{2}(LA + LL) \quad (2.3)$$

De siste seks avledningene kalles de prekordiale avledningene. Disse har hver sin elektrode, og måler spenningsforskjellen mellom den undersøkende elektrodene og elektrodene på høyre ben (jord) [48]. De seks vanlige prekordiale avledningene kalles V1, V2, V3, V4, V5 og V6. Plasseringen av disse kan ses i figur 2.6.



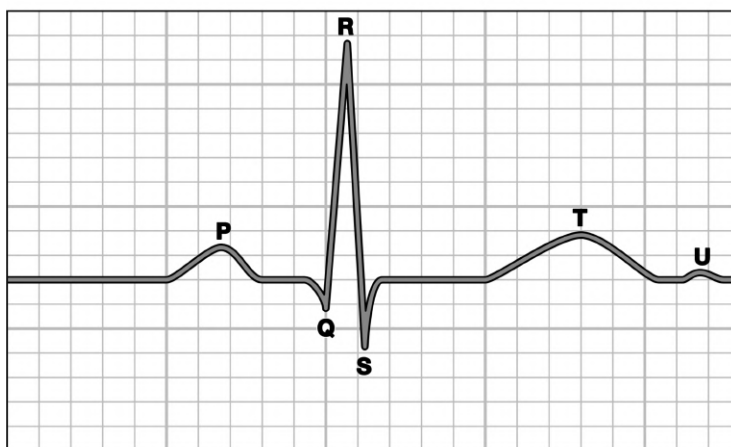
Figur 2.6: Plassering av de prekordiale EKG-avledningene; V1, V2, V3, V4, V5 og V6. Figuren er en tilpasset utgave av illustrasjonen hentet fra [49], av Twisp, 2008. CC 0 [45]

2.2.2 Karakteristiske bølger, takker og segmenter i et EKG

Et hjerteslag, sett på et EKG (figur 2.7), består av en P-bølge, en T-bølge og et QRS-kompleks [50]. P-bølgen representerer depolariseringen av atriene som fører til sammentrekning av atriene. Det flate partiet mellom P-bølgen og QRS-komplekset oppstår som følge av forsinkelsen i AV-knuten, og dette partiet er kalt PR-segmentet (eller alternativt PQ-segmentet). QRS-komplekset oppstår når den elektriske impulsen brer seg fra His-bunten, og utover i ventriklene via grenene, og til sist ut i purkinjefibrene og muskelcellene i ventriklene. Dette kalles også depolarisering av ventriklene. Når muskelcellene i ventriklene repolariseres, og kommer tilbake til elektrisk hvilenivå, oppstår T-bølgen. Segmentet mellom slutten av S-takken (også kalt J-punktet) og starten av T-bølgen kalles ST-segmentet, og representerer pausen mellom depolarisering og repolarisering av ventriklene.

Av og til er en U-bølge synlig i et EKG, slik som det er illustrert i figur 2.7. Dette er en liten bølge, som opptrer etter T-bølgen, og representerer repolariseringen av Purkinjefibrene.

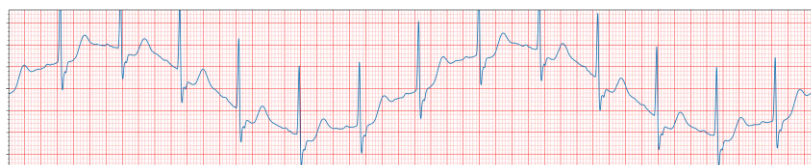
I tillegg til de ulike bølgene og segmentene er det også flere intervaller som det ofte refereres til i analyser av EKG. Noen av de mest omtalte intervallene er PQ/PR-intervallet, RR-intervallet, ST-intervallet og QT-intervallet. PQ/PR-intervallet er tiden fra starten av en P-bølge til starten av QRS-komplekset. RR-intervallet er tiden mellom to R-topper i EKGGet. ST-intervallet er tiden fra J-punktet/slutten av S-bølgen til slutten av T-bølgen. QT-intervallet er tiden fra starten på Q-bølgen til slutten av T-bølgen. I tillegg brukes ofte også varigheten på P-bølgen og QRS-komplekset i EKG-analyser.



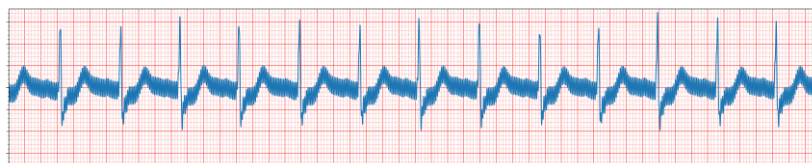
Figur 2.7: Et EKG med de tre karakteristiske bølgene; P, T og U, og de tre takkene; Q, R, S, markert på EKGGet. Figuren er laget i Mind the Graph (<https://mindthegraph.com/>). CC BY-SA 4.0 [35].

2.2.3 Støy i EKG-signaler

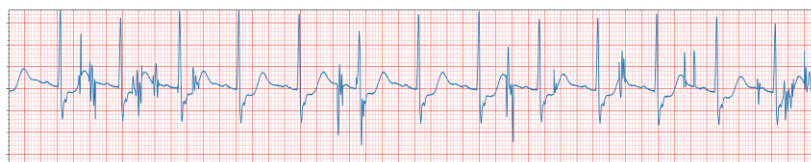
En av utfordringene med EKG-målinger ligger i å kun registrere de små spenningsignalene fra hjertet, og unngå at støy kontaminerer signalet. Det er hovedsakelig tre typer støy som kontaminerer EKG-opptakene. Disse tre typene er oscillerende grunnlinje, 50Hz støy og elektromyografisk støy [51]. Oscillerende grunnlinje er støy som kan komme som følge av bevegelse eller respirasjon [52], men også endring i elektrodens impedans [53]. Dette fremstår som store bølger med lav frekvens slik som i figur 2.8a. 50Hz støy kommer som følge av elektrisk interferens fra nettspenningen i bygget der elektrokardiografen er plassert. Dette kontaminerer EKG'et med høyfrekvent støy, som eksempelet i figur 2.8b viser. Elektromyografisk støy er støy som oppstår som følge av aktivering av andre muskler enn hjertet. Dette kan ses som korte perioder med støy, slik som i figur 2.8c.



(a) Støy fra respirasjon



(b) 50Hz støy



(c) Artefakter fra muskelspenninger

Figur 2.8: De tre vanligste formene for støy i EKG-signaler. (a) Viser støy som kommer som følge av respirasjon fra personen det måles på. (b) Viser 50Hz støy som ofte kommer fra nettspenningen i bygget der elektrokardiografen er plassert. (c) Viser artefakter fra muskelspenninger fra personen det måles på. Figuren er laget ved bruk av ecg plot [54].

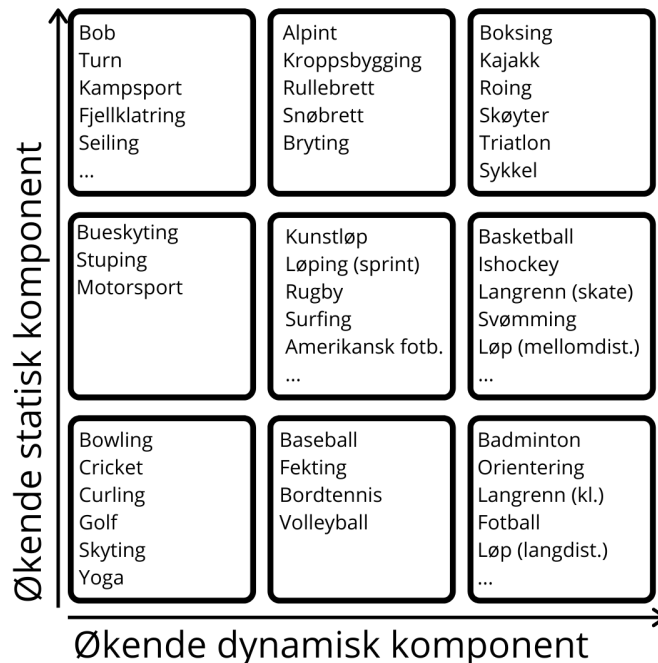
2.2.4 Elektrokardiografiske tolkningsalgoritmer

På 1950-tallet ble det mulig å konvertere analoge EKG-signaler til digital form, dette ledet videre til digitale tolkningsalgoritmer på 1960-tallet [13]. I dag er tolkningsalgoritmene ofte implementert i elektrokardiografen. Algoritmene som brukes for å tolke EKG'er er såkalte ekspertsystemer, som er bygget opp av regelbaserte algoritmer [55]. Disse algoritmene gir ofte alarmer og feiltolker

harmnløse unormaliteter i EKGGet. På grunn av feiltolkningene anbefales leger å lese over EKGene, og gjøre sin egen tolkning uavhengig av resultatet fra den automatiske EKG-tolkningen [56]. Dette er tidkrevende for legene, og krever av og til en høy grad av ekspertise [10].

2.3 Idrettshjerte

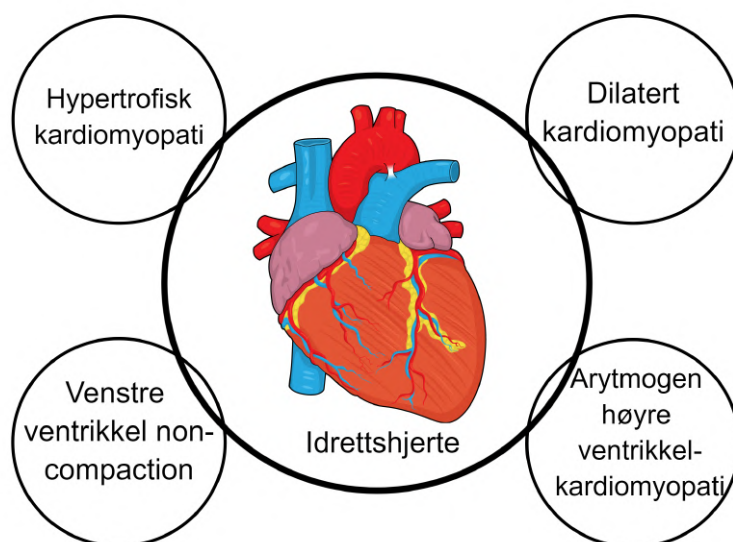
Mange toppidrettsutøvere har strukturelle endringer på hjertet som følge av trening. Dette er hva man kaller atletisk remodellering av hjertet, også omtalt som idrettshjerte. Dette ble tidlig beskrevet av Joel Morganroth (Morganroth hypotesen) i 1975. Hans hypotese sier at all trening øker muskelmassen til hjertet. Styrketrening gir i hovedsak tykkere hjertevegger, og utholdenhetstrening gir dilaterte (utvidede) hjertekamre [57]. Nyere studier viser i tillegg at idretter som stiller høye krav til både O₂-opptak (dynamisk komponent) og styrke (statisk komponent) har høyere sannsynlighet for å utvikle et idrettshjerte [58]. Utviklingen av idrettshjertet kommer som en følge av tilpasningen til blant annet det økte blodtrykket som oppstår ved trening. I figur 2.9 er en rekke idretter rangert etter deres andel av statisk og dynamisk komponent. Idretter som for eksempel roing, padling og sykkel har spesielt høy sannsynlighet for å utvikle et idrettshjerte.



Figur 2.9: En oversikt over statiske og dynamiske arbeidskrav for ulike idretter. Den horisontale akse viser graden av dynamisk komponent, mens den vertikale akse viser graden av statisk komponent.

Figuren er en tilpasset utgave av figur 2 i Mitchell W.H. et al. 2005 [59], gjengitt med tillatelse fra Journal of the American College of Cardiology[©].

De strukturelle forskjellene i et idrettshjerte er vanskelige å skille fra abnormaliteter som disponerer til plutselig hjertestans [8]. Fire av de vanligste patologiene som forveksles med idrettshjertes er: hypertrofisk kardiomyopati, venstre ventrikkel non-compaction, arytrogen høyre ventrikkel kardiomyopati og dilatert kardiomyopati [8]. Dette er illustrert i figur 2.10.



Figur 2.10: Et idrettshjerte kan forveksles med patologier som hypertrofisk kardiomyopati, venstre ventrikkel non-compaction kardiomyopati, arytrogen høyre ventrikkel kardiomyopati og dilatert kardiomyopati. Disse fire patologiene kan føre til plutselig hjertestans. Figuren er en tilpasset utgave av figur 5 i Sharma S. et al. 2015 [8], gjengitt med tillatelse fra Oxford University Press[©]. Figuren er modifisert i Mind the Graph (<https://mindthegraph.com/>)

2.3.1 Patologi som ligner på idrettshjertes

Hypertrofisk kardiomyopati

Hypertrofisk kardiomyopati (HCM) er en arvelig sykdom, som gir et hjerte med unormalt tykke hjertevegger. En studie har vist en så høy prevalens som 1 : 500 blant den generelle befolkning [60]. For mange er det mulig å leve et uproblematisk liv med hypertrofisk kardiomyopati, men likevel er sykdommen årsak til flest plutselige hjertestanser blant unge idrettsutøvere [61].

Venstre ventrikkel non-compaction kardiomyopati

Venstre ventrikkel non-compaction kardiomyopati (LVNC) ble først beskrevet i 1990 av Chin et al. [62]. Sykdommen regnes som arvelig, og en person som har denne sykdommen kan over tid utvikle hjertesvikt. Hjertesvikten utvikles som følge av at hjerteveggen blir svak, fordi den ikke har normal muskelkonsistens.

Arytmogen høyre ventrikkel kardiomyopati

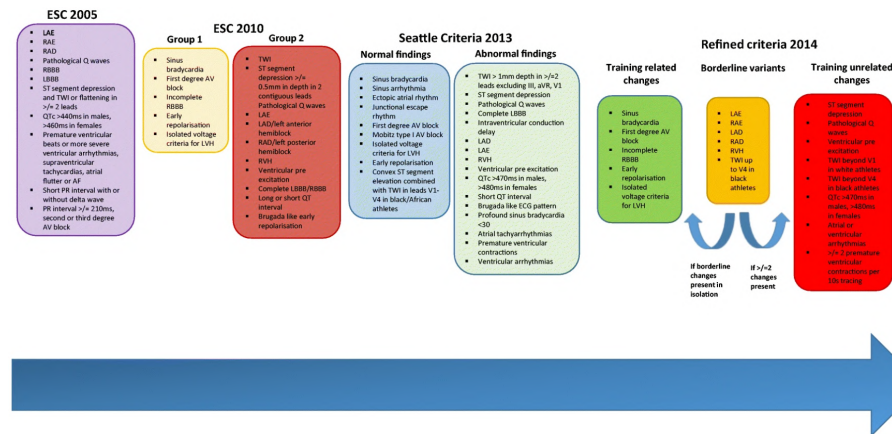
Arytmogen høyre ventrikkel kardiomyopati (ARVC) er en arvelig tilstand som kan medføre hjertesvikt. Ofte oppstår forandringer først i høyre hjertekammer[63], men kan affisere begge hjertekamre [64].

Hard trening over lengre tid kan gi utvidelse av høyre ventrikkel som ligner på ARVC, og hos de arvelig disponerte for ARVC, kan trening fremskynde sykdomsutviklingen, og dessuten utløse plutselig død [8], [65]. Studier indikerer at sykdommen kan ha en prevalens mellom 1:2500 og 1:5000 [66].

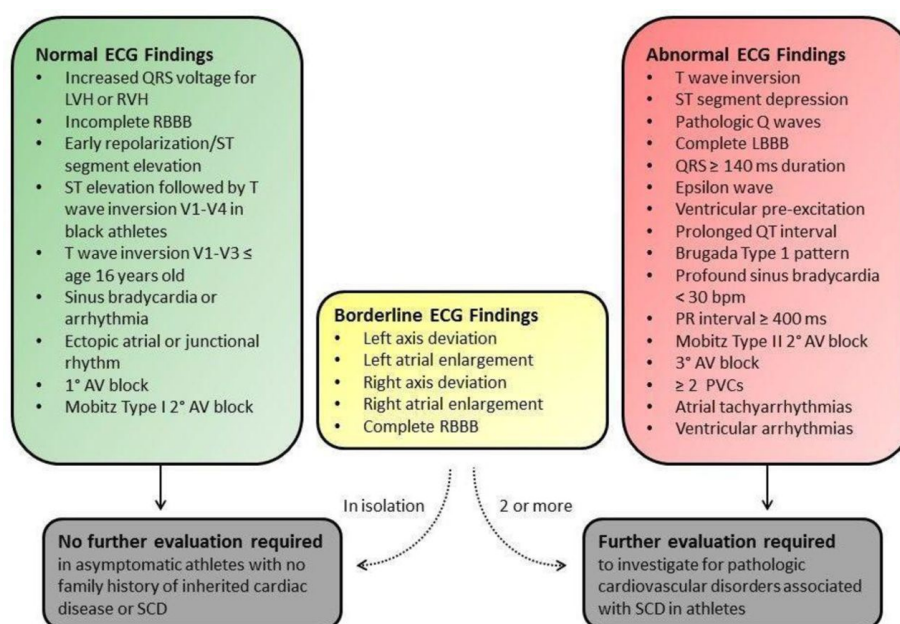
Dilatert kardiomyopati

Dilatert kardiomyopati (DCM) er i hovedsak en arvelig sykdom, som påvirker hjertet ved at hjertemusklene svekkes og hjertekammeret utvides. Dette affiserer vanligvis venstre hjertekammer først, men kan også påvirke høyre hjertekammer sekundært. Sykdommen kan føre til hjertesvikt og plutselig hjertestans. Mange pasienter er asymptomatiske [67]. Studier viser at sykdommen har en prevalens på omkring 1 : 2500 blant voksne [67].

For å kunne skille patologi som HCM, LVNC, ARVC og DCM fra et friskt idrettshjerte har kardiologer verden over utarbeidet retningslinjer, for elektrokardiografisk tolkning av idrettsutøveres EKG. Retningslinjene er under stadig utvikling og figur 2.11 viser utviklingen fra 2005 til 2014. Figur 2.12 viser de siste og gjeldende retningslinjene som ble utgitt i 2017/2018 og kalles *International Criteria*. Tolkningskriteriene deler en rekke funn inn i kategoriene normale treningsforandringer, forandringer i grenseområdet og atletisk unormal forandring av hjertet.



Figur 2.11: En historisk oversikt over tolkningskriteriene for idrettsutøveres EKG fra 2005 til 2014. Illustrasjonen er hentet fra Basu, J et al 2018 [68]. CC BY-SA 4.0 [35]



Figur 2.12: International criteria; de siste tolkningskriteriene for tolkning av EKG fra idrettsutøvere. Forskjellige EKG-funn deles inn i tre kategorier. I den grønne boksen er normale funn listet opp. Disse krever ikke videre utredning dersom de blir observert i et EKG fra en idrettsutøver. Den gule boksen viser funn som er i grenseområdet. I grenseområdet betyr at to eller flere funn i denne kategorien bør føre til videre utredning. Den røde boksen inneholder funn som er unormale hos idrettsutøvere, og krever utredning dersom ett eller flere funn blir observert i et EKG fra en idrettsutøver. Figuren er hentet fra Drezner J.A. et al 2020 [11] og gjengitt med tillatelse fra British Journal of Sports Medicine[©].

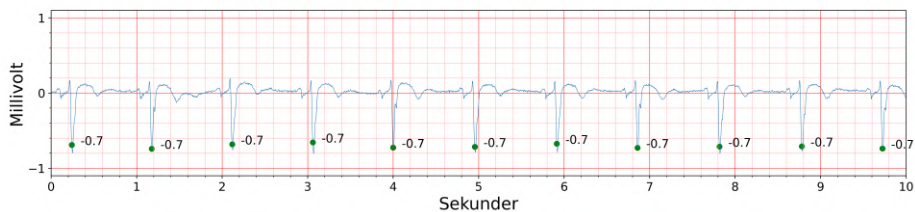
2.3.2 Normale treningsforandringer

Funnene som er beskrevet i denne kategorien, regnes som normale hos idrettsutøvere og bør ikke føre til videre utredning

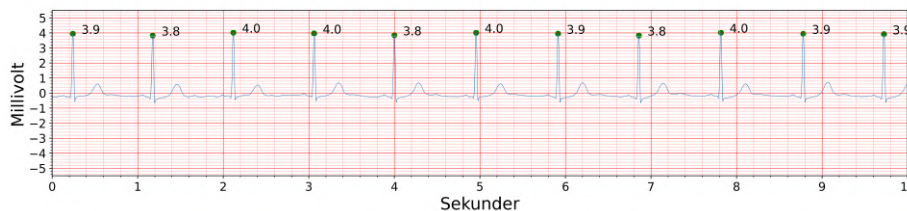
Økt QRS-spenning som følge av venstre eller høyre ventrikkelhypertrofi

Spenningskriterier for venstre ventrikkelhypertrofi (VVH) innfris ofte på idrettsutøveres EKG. Fysiologisk VVH er en økning av muskelmassen i hjertet som følge av treningen. Det finnes mange spenningskriterier og tolkningsalgoritmer for å definere VVH [69], men Sokolow-Lyon kriteriene [70] er mest brukt, og er definert ved formel 2.4. To studier har rapportert at Sokolow-Lyon kriteriene for VVH oppfylles hos 45% av idrettsutøvere [71], [72]. Figur 2.13 viser et EKG fra en person med VVH.

$$|V_{1_S}| + V_{X_R} > 3,5mV, V_{X_R} = \max(V_{5_R}, V_{6_R}) \quad (2.4)$$



(a) Avledning V1

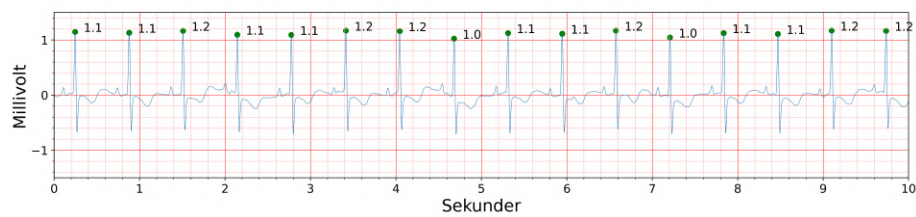


(b) Avledning V5

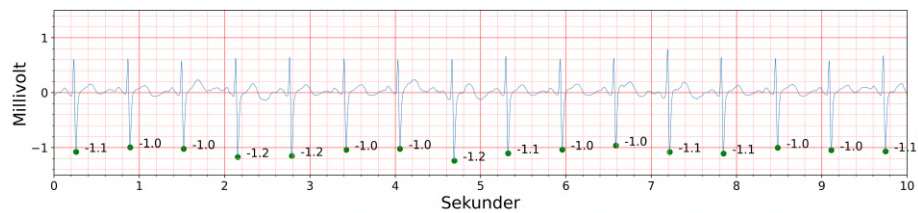
Figur 2.13: EKG fra en person som innfris Sokolow-Lyon kriteriene for VVH. (a) Avledning V1 med markerte S-takker. (b) Avledning V5 med markerte R-takker. Figurene er laget ved bruk av ecg plot [54].

Spenningskriterier for høyre ventrikkelhypertrofi (HVH) regnes også som et vanlig funn i International Criteria. En studie har rapportert en prevalens på 12% for HVH blant idrettsutøvere som oppfyller Sokolow-Lyon kriteriene [73]. Sokolow-Lyon kriteriene for HVH er vist i formel 2.5. Figur 2.14 viser EKG fra en pasient med HVH.

$$|V_{1_R}| + V_{X_S} > 1,05mV, V_{X_S} = \max(V_{5_S}, V_{6_S}) \quad (2.5)$$



(a) Avledning V1



(b) Avledning V5

Figur 2.14: EKG fra en person som innfrir Sokolow-Lyon kriteriene for HVH. (a) Avledning V1 med markerte R-takker. (b) Avledning V5 med markerte S-takker. Figurene er laget ved bruk av `ecg plot` [54].

Inkomplett høyresidig grenblokk (iRBBB)

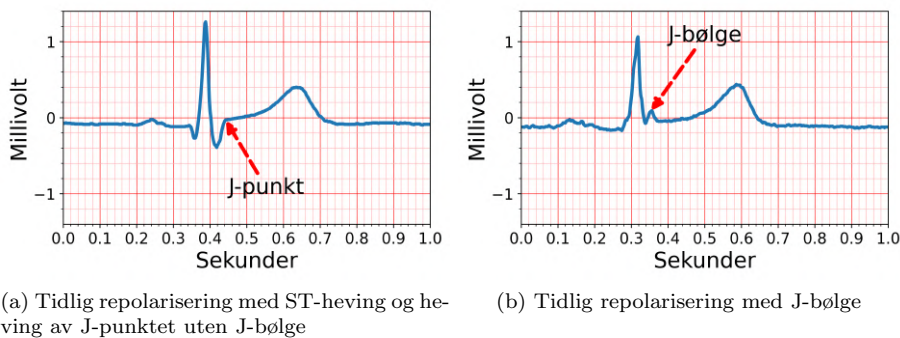
Grenblokk oppstår når de elektriske impulsene, som skal ledes fra His-bunten og gjennom ledningsfibrene, er forsinket eller avbrutte. Forsinkelsen eller blokaden kan være venstre (venstre grenblokk (LBBB)) eller høyresidig (høyre grenblokk (RBBB)). Dette gir en forsinket sammentrekning av det affiserte hjertekammeret [74].

Inkomplett høyresidig grenblokk (iRBBB) kan identifiseres på et EKG ved at QRS-varigheten er i intervallet $> 100ms$ til $< 120ms$ og at personen har et RBBB-mønster [15]. Et RBBB-mønster kjennetegnes ved at det er en ekstra R-takk, også kalt et rSR-mønster i V1-avledningen og en bred S-takk i avledning I og V6 [15]. Dette regnes for å være en normalvariant.

En studie viser til at godt trente idrettsutøvere kan forsinkelser i høyre gren/ventrikkelledning være et tegn på fysiologisk tilpasning til trening [75]. Forsinkelsen i høyre gren oppstår som følge av økt størrelse på dette hjertekammeret, og som dermed gir økt ledningstid [15].

Tidlig repolarisering/ST-segment elevasjon

QRS-komplekset i et EKG avsluttes med et knekkpunkt (J-punktet), som vanligvis ligger på grunnlinjen. I blant er J-punktet og starten på ST-segmentet løftet, og andre ganger fremstår det som en bølge (J-bølge). Tidlig repolarisering er definert som ST-elevasjon og heving av J-punktet med ≥ 0.1 mV eller tilstedeværelse av en J-bølge [76]. Studier har rapportert en prevalens mellom 10% og 90% hos atleter, dette er høyere enn i den generelle populasjonen [76].



Figur 2.15: To forskjellige former for tidlig repolarisering sett på et EKG. (a) Viser tidlig repolarisering uten J-bølge. (b) Viser tidlig repolarisering med en tydelig J-bølge. Figurene er laget ved bruk av ecg plot [54].

Juvenilt EKG-mønster

Juvenilt EKG-mønster er definert som T-bølgeinversjon i to naboavledninger fra V1 til V3 [77]. Dette er et vanlig funn hos utøvere yngre enn 16 år. Dette skyldes elektrisk dominans fra høyre ventrikel i barndommen, men dette normaliserer seg gradvis i puberteten for de fleste. Hos kvinner er negative T-bølger i V1-V3 et funn som ikke skal føre til videre utredning. Hos eldre personer kan T-bølgeinversjon imidlertid også være et kjennetegn på ARVC [77].

Sinus bradykardi

Sinus bradykardi defineres som hjertefrekvens under 50 eller 60 slag i minuttet (grensen kan variere litt i forskjellige land). Dette er svært vanlig hos godt trente idrettsutøvere. Ved fravær av symptomer som generell utmattelse, svimmelhet eller synkope (besvimelse), regnes hjertefrekvens > 30 slag i minuttet som normalt hos godt trente [15]

Sinusarytmi

Fysiologisk sinusarytmi beskriver variasjoner i hjertefrekvensen som følger respirasjonen. Hjerteslagene blir raskere ved innånding og langsommere ved utånding [78]. Dette er et vanlig fenomen hos idrettsutøvere og unge mennesker, og er assosiert med god helse. Sinusarytmi kan også være patologisk med hensyn

på rytme og frekvens, og kab forveksles med svekket sinusknutefunksjon («sick sinus syndrome»).

Junctional rytme eller ektopisk atrie rytme

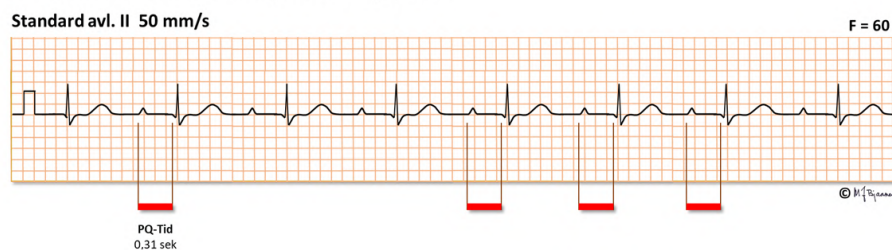
En junctional rytme oppstår når QRS-kompleksene er smale ($< 120ms$), som er et tegn på at de kommer fra forkammer eller His' bunten. I tillegg er rytmen til QRS-kompleksene raskere enn sinusrytmen / P-bølgen [15]. På et EKG kan P-bølgen mangle, være tett på eller gjemme seg i QRS-komplekset eller være uavhengig av QRS, fordi sinusfrekvensen er lavere enn QRS-frekvensen. Dette kan komme av økt aktivitet fra vagusnerven (økt vagustonus), som er et typisk trekk hos en idrettsutøver [15].

Ektopisk atrierytme kan også følge av økt vagustonus. P-bølger er da til stede, men med annen morfologi enn vanlige P-bølger. Ektopiske P-bølger kommer vanligvis fra nedre del i høyre atrie, og da er P-bølgen negativ i avledningene II, III og aVF [15]. For både junctional rytme og ektopisk atrierytme bør sinusrytmen gjenoppstå ved fysisk aktivitet [15].

Førstegrads AV-blokk

Første grads AV-blokk har en prevalens på mellom 10% og 33% hos idrettsutøvere [79]. Ved AV-blokk er PR/PQ-intervallet forlenget ($> 200ms$). Dette forårsakes av forsinket ledning i AV-knuten som følge av enten økt vagustonus, medfødte eller ervervede AV-knute-enderinger. Figur 2.16 viser et utdrag av et EKG fra en person med første grads AV-blokk.

Atrio-ventrikulært blokk (AV-blokk) grad 1



Figur 2.16: PQ-tiden $> 200ms$ som betyr at EKG innfrir kriteriene for 1 grads AV-blokk. Hentet fra UiO e-læringskurs om EKG [80], av M.J.Bjaanes ©, gjen-gitt med tillatelse av Gjesdal et al.

Annengrads AV-blokk, Mobitz type I (Wenckebachs)

Annengrads AV-blokk deles inn i to typer: Mobitz type I og Mobitz type II. Annengrads AV-blokk - Mobitz type I, også kalt Wenckebachs fenomen, er når PR/PQ-intervallet øker progressivt fra slag til slag helt til det er en P-bølge som ikke overledes, og et QRS-kompleks faller bort. Første PR-intervall etter det bortfalte slaget er kortere enn det siste overlede PR-intervallet før det bortfalte slaget. Mobitz type I andreggrads AV-blokk er et normalfunn hos asymptotiske, veltrente atleter. Ved fysisk aktivitet burde normal 1:1 overledning komme tilbake [15].

2.3.3 Treningsrelaterte forandringer i grenseområdet

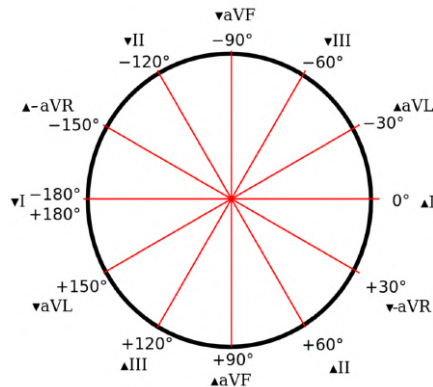
Ved to eller flere funn i denne kategorien skal personen ifølge retningslinjene henvises til videre utredning. Ved færre enn to funn i grenseområdet og ingen familiær sykdomshistorie relatert til hjertesykdom, skal det ikke henvises til videre utredning.

Høyre og venstre akseavvik

Høyre akseavvik er definert ved at QRS-komplekset har sin største vektor i feltet fra $+90^\circ$ til $+180^\circ$. Venstre akseavvik er definert ved at QRS-komplekset har sin største vektor mellom -30° til -90° . Normal akse ligger mellom -30° og $+90^\circ$. Ekstremt akseavvik ligger mellom 180° og -90° .

Aksene kan beregnes ved hjelp av avledningene i Einthovens triangel. Diagrammet i figur 2.17 har hver 30° markert med en av de seks avledningene. I tillegg er hver 30° også markert med en pil, som enten peker opp eller ned. Dette indikerer om det er en positiv (pil opp) eller negativ takk (pil ned) som må til for å innfri kriteriet.

Dersom QRS-komplekset har sin høyeste positive amplitude i avledning aVF, er hjertets frontalplanakse omtrent $+90^\circ$, og ligger på grensen mellom normal og høyre akseavvik. Har derimot QRS-komplekset sin største amplitude i III-avledningen, og den er negativ, er hjertets frontalplanakse på -60° , altså et venstre akseavvik.



Figur 2.17: Hexaxialt referansesystem. Diagrammet brukes til å lese av dreiningen til hjertets akse. Ved å finne høyeste positive amplitude eller største negative amplitude kan hjertets dreining avgjøres. Figuren er en tilpasset utgave av figuren hentet fra [81], av Mysid (2007) og modifisert i Mind the Graph (<https://mindthegraph.com/>). CC 0 [45]

Høyre atrieforstørrelse

EKG-kriteriene for høyre atrieforstørrelse er at amplituden til P-bølgen i avledning *II* og *III* eller *V1* $> 0.25\text{mV}$ [82]. Tidligere, i European Society of Cardiology (ESC) sine retningslinjer fra 2005, var høyre atrieforstørrelse et kriterie som skulle føre til videre utredning. I de siste kriteriene er høyre atrieforstørrelse flyttet til funn i grenseområdet.

Venstre atrieforstørrelse

Venstre atrieforstørrelse karakteriseres ved at varigheten på P-bølgen er lengere enn 110ms i avledning *II* eller at nedsenkningen av siste del av P-bølgen i V1 har en varighet på mer enn 40 ms og er minst 0,1mV dyp [83].

Komplett høyre grenblokk (RBBB)

RBBB skiller seg fra iRBBB ved at QRS-komplekset > 120 ms. Som for iRBBB må også typiske tegn som en ekstra R bølge (rSR-mønster) i V1-avledningen og en bred S-bølge i avledning I og V6 være tilstede for å bekrefte diagnosen RBBB [15].

2.3.4 Atletisk unormal forandring av hjertet

Funnene som er beskrevet i denne kategorien regnes ikke som normale funn hos en idrettsutøver. Dersom et av disse fenomenene oppdages i EKGGet bør idrettsutøveren henvises til videre utredning.

T-bølgeinversjon

T-bølgeinversjon i aVR, III og V1 er vanlig å se uavhengig av om EKGGet er fra en idrettsutøver eller ikke. T-bølgeinversjon i V1-V3 er vanlig hos unge utøvere og T-bølgeinversjon i V1-V4 er et vanlig funn hos afro-amerikanske idrettsutøvere. Derimot er T-bølgeinversjon i V5-V6, I og aVL å regne som et unormalt funn og bør føre til videre utredning [11].

ST-segment senkning

ST-segmentsenkning på mer enn 0,05 mV i forhold til PR-segmentet, i to eller flere avledninger, bør betraktes som et unormalt funn som krever videre utredning for underliggende, strukturell hjertesykdom [15].

Patologiske Q-bølger

En Q-bølge ses på som patologisk dersom den, i to eller flere hosliggende avledninger er mer enn 40ms bred, mer enn 0.2mV dyp og dypere enn 25% av høyden på etterfølgende R-takk [11].

Komplett Venstre grenblokk (LBBB)

Komplett venstresidig grenblokk er et uvanlig funn hos idrettsutøvere, og identifiseres ved at:

- $QRS \geq 120$ ms
- Dominant S-takk i V1
- Fravær av Q-takker i I, V5 og V6
- Forlenget R-varighet (> 60 ms) i V5 og V6

LBBB kan knyttes til blant annet dilatert kardiomyopati, og bør derfor utredes ved funn [15].

QRS $\geq 140\text{ms}$

Et QRS-kompleks med varighet lik eller større enn 140ms uten grenblokkmønster, viser en kraftig intraventrikulær ledningsforstyrrelse og bør føre til nærmere utredning [11].

Epsilonbølge

Epsilonbølger er definert som distinkte signaler med lav amplitude lokalisert mellom slutten av QRS-komplekset og begynnelsen av T-bølgen i avledningene V1-V3 [15]. Epsilonbølger er knyttet til økt risiko for ARVC.

Ventrikulær preeksitasjon

Ventrikulær preeksitasjon gir en forkortelse av PR/PQ-intervallet og breddeutvidelse av QRS-komplekset. Når preeksitasjon sees hos en person med arytmi, foreligger som regel et Wolf-Parkinson-White syndrom, som en sjelden gang kan gi ventrikkelflimmer [11]. Typiske tegn ved ventrikulær preeksitasjon er

- PR/PQ-intervall ≤ 120 ms
- Tilstedeværelsen av en delta-bølge (“slurring” av det opprinnelige QRS)
- QRS-varighet ≥ 120 ms

Forlenget QT-intervall

Et QT-intervall justert for hjertefrekvensen (QTc) på 470 ms eller over hos menn og 480 ms hos kvinner regnes som et patologisk forlenget QT-intervall som krever videre utredning [11]. Forlenget QT-intervall er assosiert med besvimelser og plutselig uventet hjertedød.

Brugada Type I mønster

Brugadas syndrom er en sykdom i hjertets elektriske system. Det kjenne-tegnes av buet ST-segment og forhøyelse $\geq 2\text{mm}$ og inversjon av T-bølgen i avledningene V1, V2 og V3 [11]. Personer med Brugadas syndrom har økt risiko for ventrikulære hjerterytmeforstyrrelser som kan føre til hjertestans [15], [84].

Uttalt sinus bradykardi ≤ 30 slag/min

Hjertefrekvens ≤ 30 slag i minuttet eller en sinuspause ≥ 3 sekunder kan være normalt hos en godt trent idrettsutøver, men bør likevel føre til ytterligere evaluering [15]. En av diagnosene man ønsker å utelukke, er skade på sinusknuten.

Tydelig første grads AV-blokk/hjerteblokk

AV-blokk der PR/PQ-intervallet er større enn 400ms, bør videre utredes [15]. Sykdommen skyldes forsinket ledningsevne i AV-knuten. Dette er en av de vanligste ledningsforstyrrelsene med en prevalens på 1,1% i den totale befolkningen [85].

Mobitz type II 2 grads AV-blokk

Mobitz type II skyldes en svikt i hjertets ledningsevne. Personer med Mobitz type II har ofte strukturelle skader på ledningssystemet i tillegg [86]. I motsetning til Mobitz I, som på EKG-et viser seg som en progressivt økende forsinkelse mellom P-bølgene og QRS-komplekset, og som ender i bortfall av ett QRS-kompleks, har Mobitz II plutselige bortfall av QRS-komplekser. Mobitz type II kan utvikle seg til en komplett hjerteblokk, og bør utredes [86].

3 grads AV-blokk

For personer med 3 grads AV-blokk er det ingen sammenheng mellom ventrikkel- og atrierytme [87]. På et EKG vil dette vise seg ved at QRS-komplekset og P-bølgen opptrer uavhengig av hverandre.

≥ 2 ventrikulære ekstrasystoler i et 10 sekunders hvile-EKG

Ventrikulære ekstrasystoler (VES) er unormale kontraksjoner som initieres i ventriklene i stedet for i atriene [88]. Dette viser seg på EKG-et ved at QRS-komplekset kommer uventet tidlig i forhold til forventet sinusrytme. To eller flere VES i et 10 sekunders hvile-EKG bør lede til videre utredning [15].

Supraventrikulære takykardier

Supraventrikulære takykardier er en samlebetegnelse for arytmier som oppstår i atriene med en frekvens på > 100 slag i minuttet. Dette inkluderer sinustakykardi, arietakykardi, AV-nodal takykardi, atrioventrikulær reentrytakykardi, atrieflimmer og atrieflutter [89], [90]. Atrielle takyarytmier er sjeldent livsfarlige, men kan være et tegn på hjertesykdommer som kan føre til plutselig hjertestans [15].

Ventrikulær arytmi

I likhet med VES starter ventrikkeltakykardier i ventriklene istedenfor i atriene. Ventrikulære arytmier er en hyppig årsak til plutselig død ved nesten alle hjertesykdommer, og bør derfor alltid utredes [15], [91].

2.3.5 Screening av idrettsutøvere

Et tiltak for å avdekke patologi hos idrettsutøvere, som kan føre til plutselig hjertestans, er å utføre hjertescreening. En studie utført i Italia mellom 1979 og 2004 rapporterte at screening med EKG reduserte forekomsten av plutselig hjertestans blant idrettsutøvere [4]. Likevel er temaet omdiskutert, og i Norge er det knyttet stor skepsis til å starte opp et screeningprogram. Et estimat gjort av lederen for medisinsk avdeling ved Olympiatoppen i Norge, var at tre norske idrettsutøvere dør årlig som følge av plutselig hjertestans. Estimater var at 325 000 idrettsutøvere må screenes for å redde ett liv årlig [33]. Det kan være vanskelig å rettfærdiggjøre en så omfattende screening for å redde relativt få liv. Screening av alle norske idrettsutøvere vil medføre store kostnader, og vil sannsynligvis også utpeke mange friske som syke, også kalt falske-positive. Fordi hjertesykdommene, som fører til plutselig hjertestans, har lav prevalens, vil selv

en metode med høy nøyaktighet likevel gi mange falske positive sammenlignet med sanne positive. Dette er kalt falsk positiv paradokset, og vil bli forklart senere.

2.4 Sannsynlighet og diagnostiske tester

For å forstå hvordan diagnostiske tester fungerer, kan det være nyttig å ha en grunnleggende kjennskap til sannsynlighetsberegninger. For de fleste sykdommer kan en pasient ha to tilstander: syk eller frisk. En diagnostisk metode derimot vil kunne ha fire forskjellige utfall: Sann positiv (SP), falsk positiv (FP), sann negativ (SN) og falsk negativ (FN). En syk pasient som får et positivt testresultat (syk) kalles en sann positiv. Får den syke pasienten et negativt testresultat (frisk) kalles det en falsk negativ. En pasient som ikke har sykdommen, men som likevel får et positivt testresultat (syk) betegnes som en falsk positiv. Dersom den friske personen får et negativt testresultat (frisk) er det en sann negativ [92]. Dette kan videre brukes til å forstå begrepene sensitivitet og spesifisitet. Sensitivitet, gitt ved formel 2.6, gir et mål på hvor god en test er til å klassifisere syke som syke. Spesifisitet, gitt ved formel 2.7, gir et mål på hvor god testen er på å klassifisere de friske som friske.

$$\text{Sensitivitet} = \frac{SP}{SP + FN} \quad (2.6)$$

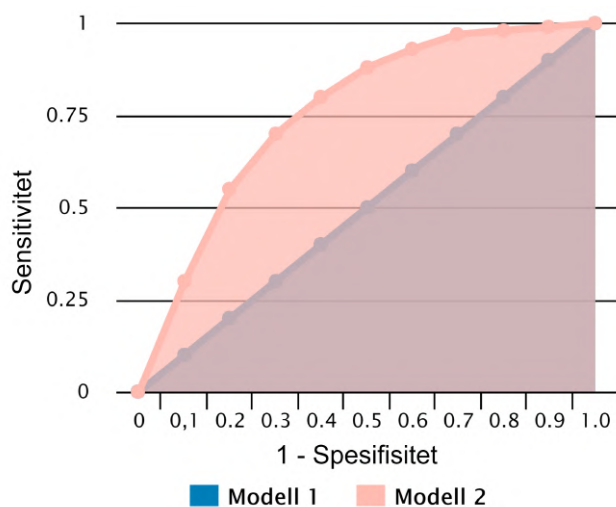
$$\text{Spesifisitet} = \frac{SN}{SN + FP} \quad (2.7)$$

I tillegg brukes presisjon i mange sammenhenger. Presisjon er et mål på den diagnostiske testens evne til å kun utpeke sanne positive, uten å peke ut falske positive. Fra formel 2.8 ser man at en test som kun utpeker sanne positive vil få en skår på 1, mens en skår på 0,5 betyr at testen utpeker like mange sanne positive som falske positive.

$$\text{Presisjon} = \frac{SP}{SP + FP} \quad (2.8)$$

Area under the curve (AUC) er mye brukt i diagnostiske tester, og spesielt metoden som kalles Receiver operating characteristic (ROC). Dette er en skåringsalgoritme som bruker sensitivitet og spesifisitet for å beregne en skår mellom 0 og 1. For en praktisk tilnærming til AUC-ROC-utregningen kan et enkelt diagram benyttes slik som i figur 2.18. I dette diagrammet viser y-aksen sensitivitet og x-aksen viser 1 – spesifisitet. Punktene som utgjør de to grafene i figur 2.18, er fra to ulike modeller; *Modell 1* og *Modell 2*. Punktene representerer sensitivitet og spesifisitet for ulike prediksjonsterkskler for *Modell 1* og *Modell 2*. I området der prediksjonsterkskelen er høy blir oftest sensitiviteten lav. Dette er fordi modellen må gi en svært høy sannsynlighet for et gitt utfall dersom sannsynligheten skal overstige prediksjonsterkskelen. Der modellene har høy prediksjonsterkskel og lav sensitivitet er gjerne spesifisiteten høy, fordi modellen må gi en høy grad av sannsynlighet for at utfallet skal få en positiv prediksjon. Dermed utpekes færre falske positive.

I andre enden av diagrammet (figur 2.18) er det motsatt. Der prediksjonsterkskelen er lav, blir sensitiviteten ofte høy, men det gir også en større risiko for å utpeke flere falske positive. Spesifisiteten blir dermed lav.



Figur 2.18: AUC-kurver for to ulike modeller. Figuren er laget i Mind the Graph (<https://mindthegraph.com/>). CC BY-SA 4.0 [35]

Modell 2 i figur 2.18 har et større areal under kurven enn *Modell 1*. Dette betyr at *Modell 2* er bedre enn *Modell 1*. *Modell 1* har en kurve som går diagonalt gjennom diagrammet og siden x og y-aksen går fra 0 til 1 blir arealet under kurven eller AUC-skåren = 0,5. Dette betyr at modellen klassifiserer riktig i 50% av tilfellene, som tilsvarer tilfeldig gjetting. *Modell 2* derimot har et større areal under kurven, som betyr at modellen klassifiserer rett i mer enn 50% av tilfellene og dette gir en AUC-skår $\geq 0,5$. En perfekt modell vil ha en AUC-skår = 1.

Falsk positiv paradokset

Falsk positiv paradokset viser hvordan en diagnostisk test, med både høy sensitivitet og spesifisitet, kan gi et feil bilde av hvor god testen er. Dette gjelder spesielt for diagnoser med lav prevalens. Som et eksempel kan man se for seg å teste alle norske idrettsutøvere, som tilsvarer omtrent 325 000 personer [33], for HCM, som har en prevalens på omtrent 1 : 500. Videre kan det antas at en kardiolog analyserer disse EKGene, og at analysen hadde en sensitivitet på 99% og spesifisitet på 99%.

Av de 325 000 idrettsutøverne, som ble testet ville statistisk sett 650 av de testede ha HCM og kardiologen ville klart å oppdage 643 av tilfellene, mens 7 personer ville ikke få avdekket sykdommen sin. For de resterende 324 350 personene som ikke har HCM vil 1% av disse, som tilsvarer 3244 personer, feilaktig bli oppfattet som syke. Så til tross for en sensitivitet på 99% og spesifisitet på 99% er det likevel bare

$$\frac{643}{643 + 3244} = 0,17$$

17% sannsynlighet for at en pasient har HCM dersom kardiologen stiller diagnosen. Denne utfordringen gjelder for alle diagnostiske tester som brukes på

sykdommer med lav prevalens.

2.5 Kunstig intelligens

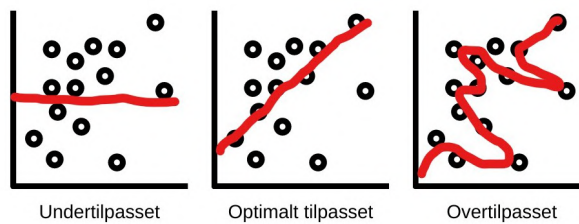
Kunstig intelligens blir av noen definert som algoritmer som er i stand til å justere sin egen aktivitet, og som derfor tilsynelatende framstår som intelligent [93]. Dette er på mange måter motsetningen til regelbaserte algoritmer, også kalt ekspertssystemer [93], som er programmert ned til minste detalj og framstår som mer forutsigbare.

Kunstig intelligens er et vidt begrep som omfavner et stort fagfelt med mange underkategorier. En av de mest omtalte underkategoriene er maskinlæring hvor man bruker statistiske metoder for å la dataprogrammer finne mønstre i store datamengder [94]. En maskinlæringsmodell programmeres ved å mate modellen med eksempler innenfor et gitt tema. Eksemplene som mates inn kan blant annet være bilder, tabell-data og tidsserier for nevne noen. Mange av metodene som brukes innenfor maskinlæring er gamle statistiske metoder, som har fått ny anvendelse og relevans ettersom tilgangen på regnekraft har økt.

2.5.1 Veiledet læring

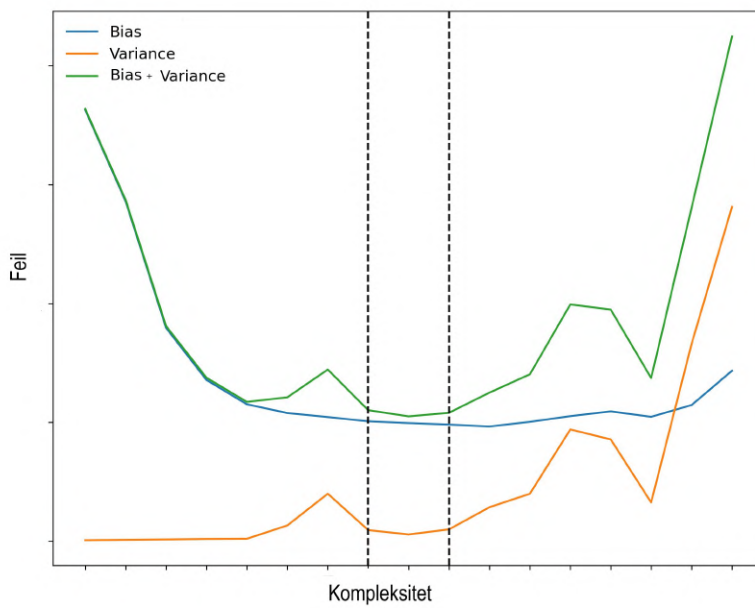
Innen maskinlæring finnes det hovedsakelig tre måter for algoritmene å lære på; forsterket læring, veiledet læring og ikke-veiledet læring [94]. I denne masteroppgaven vil det fokuseres på veiledet læring. Prinsippet for veiledet læring er at hvert eksempel eller datapunkt, som mates inn til algoritmen har en ”merkelapp”. Merkelappen er fasiten for det aktuelle eksempelet, og er det modellen skal forsøke å forutsi. Maskinlæringsmodellen justerer sine koeffisienter, slik at svaret fra modellen stemmer med merkelappen for det aktuelle eksempelet. Ved å repetere dette med flere forskjellige eksempler, så trenes modellen. Etter at modellen er trent er hensikten at modellen skal kunne forutsi merkelappen bare ved å se et nytt eksempel.

Et annet aspekt ved læringen er at det ikke er ønskelig at modellen skal lære for mye eller for lite av treningsdataene. En modell som har lært for mye av treningsdataene kalles en overtilpasset modell. Konsekvensen ved overtilpassning er at modellen vil være dårlig på å gjenkjenne nye og usette data. Motsetningen til en overtilpasset modell er en undertilpasset modell. En undertilpasset modell har så liten grad av tilpassning til treningsdataene at den hverken klarer å gjenkjenne treningsdataene eller usette data. Den optimalt trente modellen er et sted midt i mellom den undertilpassede og overtilpassede. Figur 2.19 viser eksempler på en undertilpasset, overtilpasset og optimalt tilpasset regresjonsmodell.



Figur 2.19: Tre spredningsplott med hvert sitt eksempel på kurvetilpasning (regresjon) til punktene. Plottet lengst til venstre viser en kurve som er undertilpasset til punktene i diagrammet. Plottet lengst til høyre viser en kurve som er overtilpasset til punktene i diagrammet. Plottet i midten viser en kurve som er nærmere optimalt tilpasset med tanke på å forutsi usette data.

Å trene modellen optimalt handler om å finne den riktige balansen mellom bias og varians, også kalt bias-varians kompromisset. I denne sammenhengen er bias et uttrykk for hvor godt modellen tilpasser seg til treningsdataene. En høy bias betyr at modellen ikke er godt tilpasset treningsdataene, altså er modellen undertilpasset. En modell med høy varians betyr at det er stor variabilitet i prediksjonene fra modellen. Høy varians kan assosieres med overtilpasning. Den optimalt trente modellen finner man der summen av varians og bias er lavest. Figur 2.20 viser et diagram med bias og varians som funksjon av kompleksiteten til modellen. Figuren viser hvordan den totale feilen, representert ved $\text{bias} + \text{varians}$, er dominert av bias, ved lav kompleksitet. Etter hvert som kompleksiteten øker, øker også variansen og tar over dominansen for den totale feilen.



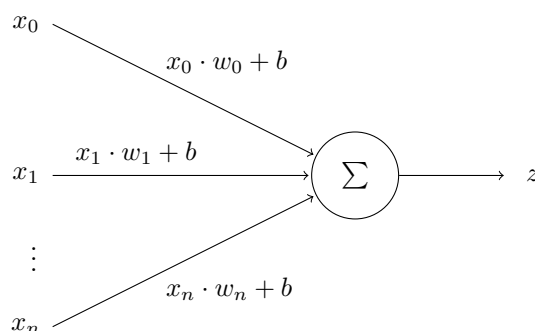
Figur 2.20: Bias-variens kompromisset. Bias, varians og en ureducerbar feil er de tre typene feil som ofte finnes i en maskinlæringsmodell. Figuren viser en typisk trend for hvordan bias dominerer den totale feilen ved lav kompleksitet eller ved lite trening. Etter hvert som kompleksiteten øker avtar bias, og variansen tar over dominansen av den totale feilen til modellen. Området mellom de stiplede linjene markerer den optimale kompleksiteten for modellen for det gitte problemet.

Nevrale nettverk

Nevrale nettverk er en maskinlæringsmetode som er inspirert av hjernen og dens nevroner. Et kunstig nevron ble først beskrevet av McCulloch og Pitts i 1943 [95], og senere videreutviklet av Frank Rosenblatt på 50-tallet [96]. Et enkelt nevron kan beskrives med formel 2.9, hvor w_i er vektallene og x_i inngangsverdiene fra i -antall inngangssignaler. Nevronets bias er uttrykt ved b .

$$z = \sum_{i=1}^n w_i x_i + b \quad (2.9)$$

Figur 2.21 kan gi en mer intuitiv forståelse av det samme nevronet, som er beskrevet med formel 2.9.



Figur 2.21: Et eksempel på et kunstig nevron. Figuren er en tilpasset utgave av Single Neuron and Perceptron (figure 1) hentet fra [97] av David Stutz[©], 2020. Gjengitt med tillatelse.

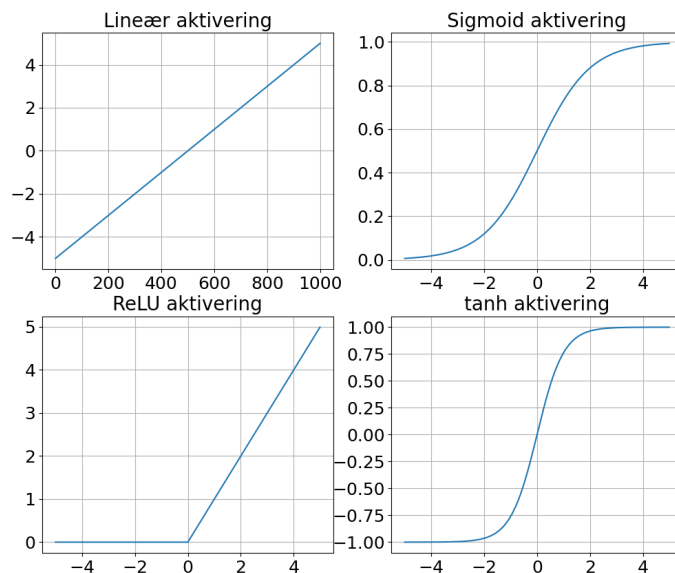
Resultatet, z , fra nevronet blir ofte behandlet av en aktiveringsfunksjon, f , slik at

$$a = f(z)$$

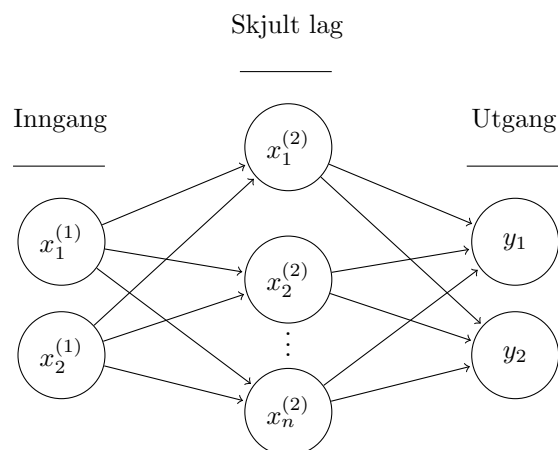
Aktiveringsfunksjonen, f , kan for eksempel være en Sigmoid-aktivering. Sigmoid-funksjonen kan beskrives ved formel 2.10, og er vist i figur 2.22.

$$S(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

De kunstige nevronene kan bygges opp lagvis, slik som i figur 2.23, hvor modellen har 2 nevroner i inngangslaget, et skjult lag med n nevroner og et utgangslag med 2 nevroner. Dersom modellen består av flere skjulte lag kalles modellen for et dypt nevralt nettverk. Dette blir omtalt som en egen underkategori av maskinlæring.



Figur 2.22: Fire eksempler på aktiveringsfunksjoner. Diagrammet øverst til venstre viser en lineær aktiveringsfunksjon. Diagrammet øverst til høyre er en Sigmoid aktiveringsfunksjon. En ReLU arkiveringsfunksjon vises i nederst vil venstre i figuren og i nederste høyre hjørne er vises en tanh-funksjon.



Figur 2.23: Et nevralt nettverk med et to inngangsnoder, et skjult lag med n noder og et utgangslag med to noder. Figuren er en tilpasset utgave av Multi-Layer Perceptrons (figure 3) hentet fra [97] av David Stutz[©], 2020. Gjengitt med tillatelse.

2.5.2 Dype nevrale nettverk

Dype nevrale nettverk er nevrale nettverk som består av flere skjulte lag. De dype nevrale nettverkene kan igjen deles inn i flere undergrupper. Den første undergruppen er de generelle dype nevrale nettverkene, som består av vanlige nevroner, som tidligere er beskrevet, tillegg finnes det konvolusjonelle nevrale nettverk, tilbakevendende eller gjentakende nevrale nettverk og dype Boltzmann maskiner. I denne masteroppgaven vil de konvolusjonelle nevrale nettverkene få mest oppmerksomhet.

Trening av nevrale nettverk gjøres i iterasjoner der det veksles mellom å oppdatere vektene i modellen og teste om modellen har blitt bedre. Treningssettet deles ofte inn i batcher, som er små delsett av det totale treningssettet. Modellen testes på batchen, og deretter oppdateres vektene til modellen i henhold til modellens ytelse på batchen. Modellen får ofte trene på hele treningssettet flere ganger. Én treningsrunde på hele treningssettet kalles en epoke.

Loss og Cost-funksjon

For å måle hvor godt det nevrale nettverket presterer brukes en cost-funksjon. Cost-funksjonen beregner modellens klassifiseringsfeil, eller forskjellen mellom predikert verdi og faktisk verdi. Det er vanlig å skille mellom loss-funksjon og cost-funksjon ved at loss-funksjonen brukes til å måle feilen for et enkelt eksempel, mens cost-funksjonen brukes til å beregne feilen for en batch eller hele treningssettet.

Dersom formålet er å trene opp én eller flere klasser, individuelt på et treningssett, er det vanlig å bruke binær kryssentropi som loss/cost-funksjon. Loss-funksjonen for binær kryssentropi er gitt ved følgende formel:

$$\mathcal{L}(y, \tilde{y}) = -(y \log \tilde{y} + (1 - y) \log(1 - \tilde{y})) \quad (2.11)$$

hvor y er den faktiske verdien, mens \tilde{y} er modellens klassifiserte verdi som et tall mellom 0 og 1.

Cost-funksjonen for binær kryssentropi kan skrives på følgende måte

$$\mathcal{C}(\tilde{Y}, Y) = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\tilde{y}_i) + (1 - y_i) \log(1 - \tilde{y}_i)), \quad (2.12)$$

hvor Y inneholder alle faktiske verdier y_i , \tilde{Y} inneholder alle \tilde{y}_i klassifisert av modellen og n er antallet samples i Y og \tilde{Y} .

Optimizer

Optimiserens oppgave er å minimere cost-funksjonen. Prinsippet bak en optimizer kan forklares ved å studere gradient decent-algoritmen, som er gitt ved formel 2.13. Gradient decent-algoritmen har som mål å finne den optimale koeffisienten, θ , for å minimere cost-funksjonen \mathcal{C} . Læringsraten, η , bestemmer størrelsen på stegene i endringen $\frac{\partial \mathcal{C}}{\partial \theta_i}$. En liten η vil føre til sakte konvergens mot minimum av cost-funksjonen, mens en høy η vil kunne føre til ustabil læring.

$$\theta_{i+1} = \theta_i - \eta \frac{\partial \mathcal{C}}{\partial \theta_i} \quad (2.13)$$

Backpropagation

For å beregne gradientene, som optimizere bruker for å minimere cost-funksjonen, brukes backpropagation-algoritmen i nevralt nettverk. Backpropagation ble først foreslått i doktorgradsavhandlingen til Paul John Werbos i 1974 [98], men ble for første gang tatt i bruk i et nevralt nettverk i 1986 [99]. Backpropagation-algoritmen oppdaterer vektene i det nevralt nettverket på en mer effektiv måte enn hva som var mulig tidligere. Den økte effektiviteten gjør det mulig å trene dype nevralt nettverk.

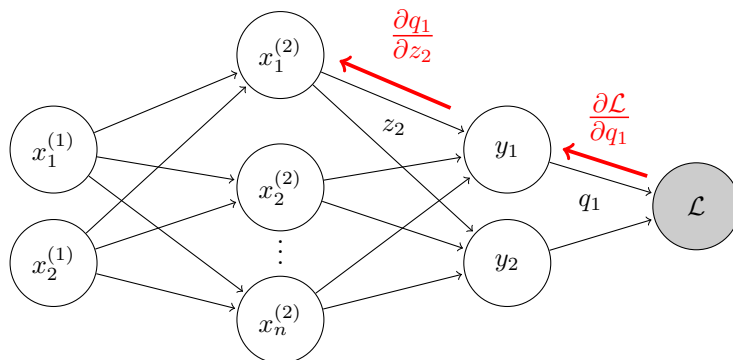
Backpropagation-algoritmen beregner gradienten til hvert av nevronene fra siste lag i nettverket, og mot det første laget [100]. Metoden som benyttes for å beregne seg bakover i nettverket er kjerneregelen (formel 2.14).

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial u} \frac{\partial u}{\partial x} \quad (2.14)$$

I figur 2.24 er det illustrert hvordan kjerneregelen kan tas i bruk på et nevralt nettverk for å regne ut $\frac{\partial \mathcal{L}}{\partial z_2}$ og dermed beregne gradienten til utgangen fra $x_1^{(2)}$:

$$\frac{\partial \mathcal{L}}{\partial z_2} = \frac{\partial \mathcal{L}}{\partial q_1} \frac{\partial q_1}{\partial z_2}$$

Eksempelet i figur 2.24 er forenklet ved å anta at nevronene kun benytter lineær aktiveringsfunksjon.



Figur 2.24: En illustrasjon av backpropagation i et nevralt nettverk. Figuren er en tilpasset utgave av Backpropagation (figure 4) hentet fra [97] av David Stutz[©], 2020. Gjengitt med tillatelse.

Hyperparametre

I dype nevralt nettverk, og generelt for de fleste maskinlæringsalgoritmer, er det visse parametre som kan endres, og som kan påvirke utfallet av treningen. Eksempler på slike parametre er optimizer, læringsrate, batch-størrelse og ulike regulariseringsparametre for å nevne noen. Disse kalles hyperparametre og kan optimaliseres for å maksimere ytelsen til en modell for en gitt oppgave.

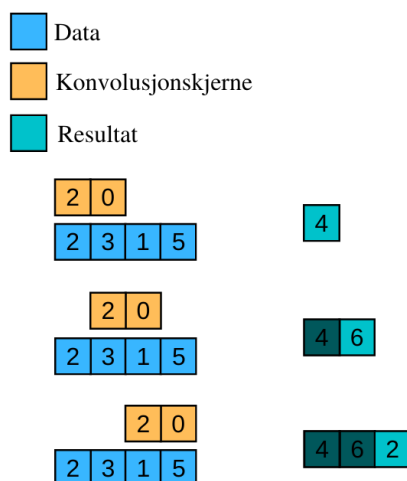
2.5.3 Konvolusjonelle nevralt nettverk

Konvolusjonelle nevralt nettverk (CNN) vil nesten alltid regnes som dype nevralt nettverk fordi de oftest består av mange lag og mange matematiske operasjoner. CNN-modeller er først og fremst kjent for sin evne til å utføre bilde-gjenkjenning i både det 2-dimensjonale og i det 3-dimensjonale rom, men det kan også brukes på 1-dimensjonale data, som for eksempel naturlig språkbehandling og tidsserier.

Konvolusjon

For å forstå CNN-modeller og hvordan disse fungerer er det viktig å forstå hva konvolusjon er. I denne studien er 1-dimensjonelle CNN brukt. En 1-dimensjonell konvolusjonell matematisk operasjon gjøres ved beregne dotproduktet (formel 2.15) mellom to tallrekker. For et konvolusjonelt lag i en CNN-modell kan dette vises ved figur 2.25, hvor en konvolusjonskjerne bestående av to elementer brukes på data bestående av ett konvolusjonsfilter.

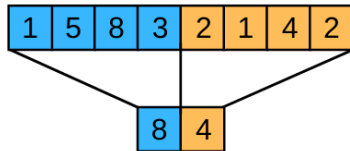
$$a \cdot b = \sum_{i=1}^n a_i b_i \quad (2.15)$$



Figur 2.25: Eksempel på 1-dimensjonal konvolusjon. Den blå tallrekken, bestående fire tall, er dataene som skal behandles av den gule konvolusjonskjernen, bestående av to tall. Ved å beregne dotproduktet mellom konvolusjonskjerner og de to første tallene i data-tallrekken blir resultatet det første tallet i konvolusjonen. Etter denne matematiske operasjonen forflyttes konvolusjonskjernen ett tall videre i data-tallrekken og gjør en ny beregning av dotproduktet. Den grønne resulterende tallrekken er resultatet av konvolusjonen.

Maks pooling

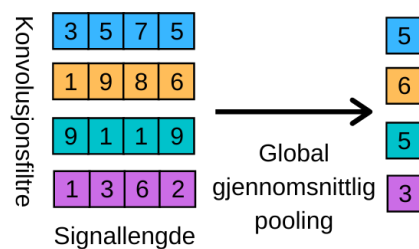
Maks pool-laget brukes i sammenheng med konvolusjonelle lag. Egenskapen til maks pool laget er å redusere størrelsen til dataene som behandles av nettverket. Dette resulterer i færre beregninger og raskere trening av nettverket, men kan også føre til tap av viktig informasjon. Figur 2.26 viser et eksempel på en maks pool operasjon med en pool-størrelse på fire.



Figur 2.26: Et eksempel på en 1-dimensjonal maks pool operasjon. I det aktuelle eksempelet har maks pool operasjonen en pool-størrelse = 4. Dette betyr at tallrekken som blir behandlet av maks pool-operasjonen blir delt opp i sekvenser på fire. Deretter blir alt bortsett fra det høyeste tallet fjernet.

Global gjennomsnittlig pooling

Global gjennomsnittlig pooling (fritt oversatt fra global average pooling) er et lag som oftest brukes helt mot slutten av et konvolusjonelt nevralt nettverk. Global gjennomsnittlig pooling tar gjennomsnittet av alle egenskapene på tvers av alle konvolusjonsfiltrene fra tidligere konvolusjonelle operasjoner i nettverket. Gjennomsnittsverdiene danner en 1-dimensjonell vektor som kan brukes til en endelig klassifisering av en aktiveringsfunksjon [101]. Et eksempel på en global gjennomsnittlig pooling-operasjon er vist i figur 2.27.



Figur 2.27: Et eksempel på en 1-dimensjonal global gjennomsnittlig pooling-operasjon. I det aktuelle eksemplet blir global gjennomsnittlig pooling-operasjonen brukt på data med fire konvolusjonsfiltre med en signallengde på fire. Resultatet er at det for hvert konvolusjonsfilter blir beregnet en gjennomsnittsverdi, som resulterer i en ny vektor med redusert dimensjonalitet.

Normaliseringslag

I konvolusjonelle nevrale nettverk er det vanlig å legge et normaliseringslag mellom et konvolusjonslag og aktiveringsfunksjonen. μ og σ er henholdsvis gjennomsnittet og standardavviket til X . La X være en vektor, slik at $X \in x_1, x_2, x_3, \dots, x_n$. Da kan den normaliserte verdien for x_i ; z_i beregnes ved formel 2.16.

$$z_i = \frac{x_i - \mu}{\sigma} \quad (2.16)$$

En av normaliseringsteknikkene er kalt batch-normalisering. Denne metoden beregner gjennomsnittet og variansen for alle egenskapene i en batch, og normaliserer alle egenskapene. Studier viser at batch-normalisering stabiliserer treningen, slik at treningen kan gjøres med høyere læringsrate, som fører til raskere konvergens mot optimal cost-funksjon, og dermed korter ned treningstiden [102].

Der batch-normaliseringslaget normaliserer alle egenskapene, også på tvers av kanalene over en hel batch, gjør instance normaliseringslaget det noe annerledes ved å normalisere egenskapene individuelt for hver konvolusjonsfilter i hver batch [103].

Dropout

Dropout-laget ble for første gang beskrevet av Srivastava & Hinton i 2014. De beskrev at hensikten med dropout-laget er å unngå overtilpasning av modellen ved å utelate noen av nevronene under treningen av et nevralt nettverk [104]. Hvilke nevroner som blir utelatt blir trukket tilfeldig før hver epoke. I TensorFlow [105] er Dropout-laget implementert, slik at en kan bestemme hvor mange nevroner som skal utelates i hvert dropout-lag. I tabell 2.1 vises et eksempel på hvordan et dropout-lag kan brukes i TensorFlow. For det aktuelle eksempelet vil 16 tilfeldige nevroner utelates av de totalt 32 nevronene i laget før dropout-laget.

Tabell 2.1: Kodeutklippet viser hvordan en dropout-operasjon kan brukes i et nevralt nettverk. Halvparten av de 32 nevronene i laget før dropout-operasjonen vil utelates under treningen. Hvilke av nevronene som utelates trekkes tilfeldig før hver epoke.

```
model.add(Dense(32))
model.add(Dropout(0.5))
model.add(Dense(1))
```

Oppmerksomhetsmekanisme

En oppmerksomhetsmekanisme (fritt oversatt fra Attention Mechanism) i en 1-dimensjonell CNN-modell, for tidsserie-klassifisering, brukes for å lære hvilke deler av tidsserien som er viktig for klassifiseringen. Dette har vist seg å være nyttig i disiplinen språkgjenkjenning og i transformer nettverk som brukes av DeepMind i AlphaFold [106].

For å forstå hvordan en oppmerksomhetsmekanisme fungerer må man forstå softmax. Softmax er en aktiveringsfunksjon der summen av aktiveringene, σ , for alle inngangsverdiene, z_i , kan summeres til 1. Softmax-funksjonen er gitt ved formel 2.17, der $\sigma(Z)_i$ gir sannsynligheten for elementet i av totalt K -elementer i vektoren Z

$$\sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.17)$$

Softmax brukes typisk som siste aktivering i et klassifiseringsproblem der det kun finnes ett korrekt svar blant to eller flere klasser. Softmax-laget kan også brukes til å vekte ulike egenskaper i et nevralt nettverk, slik det gjør i en oppmerksomhetsmekanisme.

En oppmerksomhetsmekanisme for tidsserier fungerer ved å multiplisere vektorer av samme lengde. *Vektor 1* føres først gjennom et softmax-lag, før den multipliseres med *Vektor 2*. Den resulterende vektoren, *Vektor 3*, er en vektning av elementene i *Vektor 2*, basert på aktiveringen av alle elementene i *Vektor 1*.

2.5.4 Ensemble modeller

Ensemble modell er et begrep innen maskinlæring og betyr at man kombinerer klassifiseringsresultat fra flere modeller. I motsetning til enkeltstående maskinlæringsmodeller, som har én hypotese for å løse klassifiseringsproblemet, har ensemble modellen flere hypoteser, som til sammen stemmer over den beste løsningen på problemet [107]. Målet er at den samlede prediksjonen skal overgå ytelsen til en enkelt modell. Ensemble modeller har vist seg spesielt nyttig på klassifiseringsproblemer med mange klasser der hvert resultat kan ha en positiv prediksjon for flere klasser samtidig (multi-label klassifisering). For klassifiseringsproblemer med mange klasser har det vist seg nyttig å dele klassifiseringen i flere små klassifiseringsproblemer [108].

Label space partitioning classifier

Label space partitioning classifier er en klyngealgoritme som deler det totale antall klasser inn i flere små klassifiseringsproblemer. Hver klynge kan betraktes som et uavhengig klassifiseringsproblem og klyngene kan bestå av de samme klassene flere ganger. Resultatet av klassifiseringen er \cup av resultatene fra hver av de mindre klassifiseringsproblemer [108].

Klassifiseringskjeder

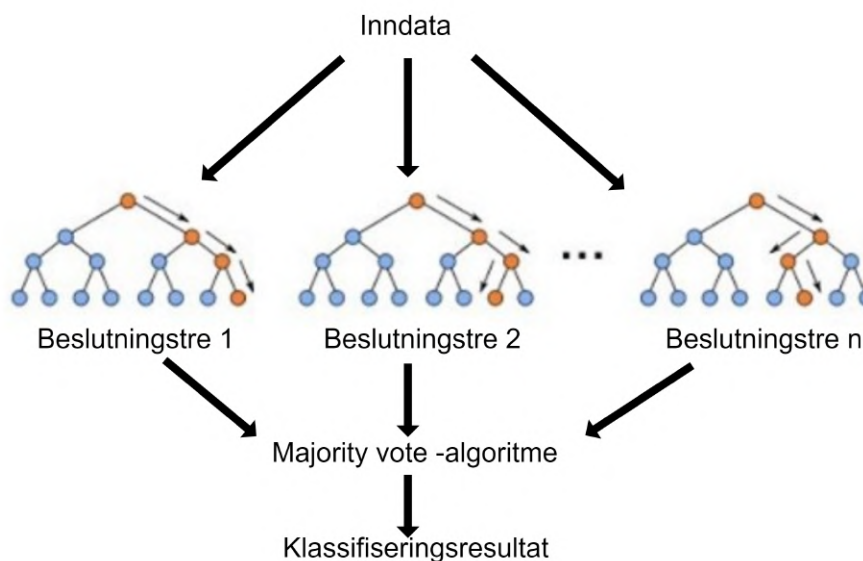
Denne metoden er basert på J. Read et al 2009 [109]. Klassifiseringskjeder bygger på Binær relevans-metoden, som er en godt egnet metode for multi-label klassifisering [110]. Klassifiseringskjeden konstruerer en sannsynlighetsbasert kjede av klassifiseringer per klasse. For L antall klasser trener den L klassifiseringsmodeller. Den første modellen blir trent til å bare klassifisere den første klassen, basert på alle tilgjengelige egenskaper. Deretter blir den neste modellen trent til å klassifisere den neste klassen, basert alle tilgjengelige egenskaper og klassifiseringsresultatet fra den første modellen. Slik fortsetter det helt det er trent en modell for samtlige L klasser. Treningsrekkefølgen for klassifiseringskjeder følger samme rekkefølge som klassene er gitt i treningssettet.

Tilfeldige beslutningsskoger

Tilfeldige beslutningsskoger ble først foreslått av Tin Kam Ho i 1995 [111], og baserer seg på beslutningstrær. Tilfeldige beslutningsskoger er i seg selv en ensemble modell, fordi den består av flere beslutningstrær som stemmer på en endelig klassifisering.

Et av suksesskriteriene for tilfeldige beslutningsskoger er å redusere korrelasjonen mellom beslutningstrærne, og på den måten reduseres også variansen til modellen [112]. Dette gjør at kompleksiteten til modellen kan økes uten at modellen blir overtilpasset, sammenlignet med et enkelt beslutningstre. I figur 2.28 er det illustrert et eksempel på en tilfeldig beslutningsskog-algoritme, bestående av n beslutningstrær. Resultatene fra beslutningstrærne blir evaluert av en majority vote-algoritme, som summerer opp sannsynlighetene for de forskjellige klassene og gjør en endelig prediksjon.

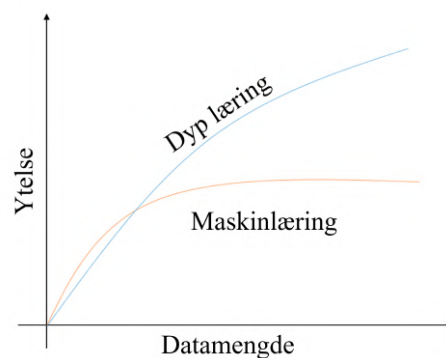
En majority vote-algoritme fungerer ved å sammenstille klassifiseringsresultater fra flere modeller, som klassifiserer de samme klassene. Den endelige klassifiseringen, basert på majority vote-algoritmen blir avgjort ved at mer enn halvparten (majoriteten) av modellene stemmer på et utfall.



Figur 2.28: En tilfeldig beslutningsskog som består av n beslutningstrær, og som tilsammen danner en beslutningsskog. Klassifiseringsresultatet for hvert beslutningstre blir evaluert av en majority vote-klynge, som gjør en endelig prediksjon. Figuren er en tilpasset utgave av illustrasjonen laget av Venkata Jagannath og hentet fra [113]. Figuren er modifisert i Mind the Graph (<https://mindthegraph.com/>). CC BY-SA 4.0 [35].

2.5.5 Maskinl ring og datamengder

Hvor mye data som trengs for   trene maskinl ringsmodeller og oppn  en akseptabel sk r er avhengig av flere ting. Blant annet kvaliteten p  dataene og kvaliteten p  annoteringene (ved veiledet l ring). I tillegg er kompleksiteten til modellen som skal trenes avgj rende for hvor mye data som trengs. Figur 2.29 illustrerer hvordan tradisjonelle maskinl ringsmetoder viser seg   ha bedre ytelse enn de mer komplekse modellene, som er basert p  dyp l ring, ved sm  datamengder. N r datamengden  ker ser det ut til at ytelsen til maskinl ringsmodellene avtar, mens modellene basert p  dyp l ring fortsetter   stige, og presterer etter hvert bedre enn de enklere maskinl ringsmodellene.



Figur 2.29: En illustrasjon p  hvordan man forventer at modeller basert p  dyp l ring skal overg  gamle og tradisjonelle maskinl ringsmodeller dersom en viss mengde data er gitt. Figuren er en tilpasset utgave av Figur 7 i Alom, Md Z et al. 2019 [114]. CC BY 4.0 [35]

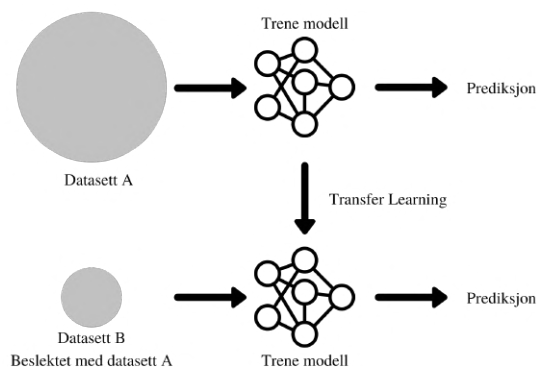
I noen tilfeller vil det v re vanskelig   skaffe nok data innenfor et domene til   oppn   nsket ytelse av maskinl ringsmodellen. En mulighet er   trene en modell p  et lignende domene der tilgangen p  data er tilstrekkelig, og deretter trene den samme modellen p  det mindre datagrunnlaget til det opprinnelige domenet. Dette er en metode som gir muligheten til   oppn  akseptable resultater selv n r datagrunnlaget er lite. Dette kalles Transfer Learning.

Transfer Learning

Transfer Learning er et begrep som brukes innen maskinl ring og dyp l ring. Transfer Learning dreier seg om   bruke kunnskap fra et domene til   l se et annet, men beslektet problem (illustrert i figur 2.30). Denne metoden er allerede mye brukt blant annet ved bildegjenkjenning. Ved bruk av TensorFlow [105] kan man for eksempel laste ned populære bildegjenkjenningsmodeller, som ResNet50 og VGG16, med ferdig trente vektst rrelser. Modellene er trent p  en rekke forskjellige bilder i ImageNet-databasen [115]. Fordelen er at de forh ndstrente modellene allerede har l rt seg   kjenne igjen objekter, kanter og kontraster i et bilde. Dette gj r at modellen raskere kommer opp p  et akseptabelt niv  ved trening p  nye objekter.

Transfer learning har ogs  blitt brukt p  andre domener enn bildegjenkjenning. For eksempel har en studie vist hvordan en modell, trent til   gjenkjenne

ulike arytmier i EKG fra mennesker, kan “re-trenes” på EKG fra hester, og gi en akseptabel ytelse i å gjenkjenne ulike arytmier hos hester [116].



Figur 2.30: En illustrasjon av prinsippet bak Transfer Learning. Modellen kan trenes på et stort datasett A, og deretter trenes på et mindre, men beslektet datasett B. På denne måten kan det oppnås et bedre klassifiseringsresultat enn dersom modellen kun var trent på datasett B.

2.5.6 Forklarbar kunstig intelligens

I innledningen ble det gitt en kort introduksjon til forklarbar kunstig intelligens. Blant annet ble det forklart at GDPR gir en person, som er underlagt en avgjørelse fra en algoritme, retten til en menneskelig gjennomgang av resultatene [29]. En annen grunn til å implementere forklarbar kunstig intelligens er for å verifisere at grunnlaget for beslutningen er riktig. Komplekse modeller vil kunne se sammenhenger der mennesker ikke kan se dem, men av og til kan man med sikkerhet si at modellen tar feil. En slik feil kan illustreres med et eksempel, som ble presentert i Ribeiro et al. 2016 [117]. I denne studien ble en enkel modell trent opp på et datasett med bilder av huskyer og ulver. Formålet med modellen var å kunne skille mellom bilder av huskyer og bilder av ulver. Modellen ble testet på et balansert testsett, bestående av 10 bilder, hvorav modellen klassifiserte 8 riktig og 2 uriktig. 80% nøyaktighet kan tilsynelatende se ut som en god modell, men forklaringen til prediksjonen, vist i figur 2.31b, viser at beslutningsgrunnlaget til modellen ikke er tuftet på ulven eller huskyens utseende, men snøen i bakgrunnen.



Figur 2.31: Forklarbar kunstig intelligens på husky og ulvehoder. (a) Viser et bilde av en husky som ukorrekt ble klassifisert som en ulv av en husky/ulv- klassifiseringsmodell. (b) Viser modellens beslutningsgrunnlag, og det som er farget med grått var ikke viktig for prediksjonen, i følge modellen. Resultatet er at det ser ut til at snøen i bakgrunnen var det viktigste for prediksjonen. Figurene er hentet fra Ribeiro et al. 2016 [117], Figure 11. Figuren er gjengitt med tillatelse fra Ribeiro et al. 2016.

Forskjellige forklaringsmetoder

Forklarbar KI kan deles inn i to hovedkategorier; globale forklaringsmodeller og lokale forklaringsmodeller [118]. De globale forklaringsmodellene har som formål å forklare hvordan modellen fungerer for alle mulige eksempler, mens de lokale forklaringsmodellene forklarer hver enkelt prediksjon [119], [120].

Lokale forklaringsmodeller kan igjen deles inn i to underkategorier; modell-spesifikk og modell-agnostisk [118]. Den modell-spesifikke forklaringsmodellen er knyttet til selve modellen, mens modell-agnostiske forklaringsmetoder skiller forklaringen fra modellen, og behandler modellen som en svart boks [117].

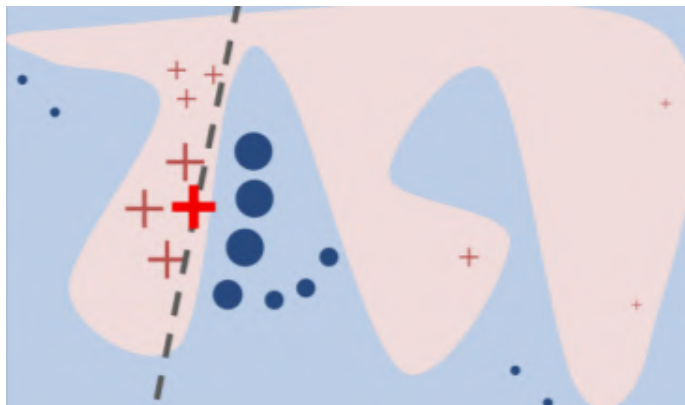
LIME (Local Interpretable Model-Agnostic Explanations)

Et eksempel på et populært rammeverk i Python, som kan brukes til å implementere lokale modell-agnostiske forklaringer, er LIME. LIME trener en lineær og forklarbar surrogatmodell, ved å observere klassifiseringsresultatet til modellen den skal forklare, samtidig som den gjør små endringer i input-dataene. På denne måten avdekkes viktigheten av de forskjellige egenskapene i input-dataene for det aktuelle klassifiseringseksempel [117].

Ribeiro et al. 2016 forklarer LIME ved å si at dersom forklaringsmodellen $g \in G$ hvor G er en potensielt forklarbar modell, så kan forklaringen, ξ , for eksempelet, x , finnes ved å minimere loss-funksjonen $\mathcal{L}(f, g, \pi_x)$ [117]. I loss-funksjonen \mathcal{L} er f den originale modellen som skal forklares, g er forklaringsmodellen og π_x er et mål på avstanden til "nærmeste nabo" for eksempelet x

for den originale modellens komplekse klassifiseringsfunksjon [117], [121]. Figur 2.32 viser den komplekse klassifiseringsmodellen på en intuitiv måte. I tillegg er forklaringen, ξ , avhengig av modellens kompleksitet gitt ved $\Omega(g)$, slik at ξ kan uttrykkes slik ved formel 2.18.

$$\xi(x) = \arg \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (2.18)$$



Figur 2.32: Den rosa og blå bakgrunnen illustrerer beslutningen til en kompleks og ulineær klassifiseringsmodell [117]. Den stiplede linjen representerer den lineære forklaringsmodellen, som er trent på datasettet \mathcal{Z} , representert ved de røde kryssene og de blå runde prikkene. Det uthevede røde krysset er eksemplet som blir forklart av modellen. Figuren er hentet fra Ribeiro et al. 2016 [117], Figure 3. Figuren er gjengitt med tillatelse fra Ribeiro et al. 2016.

$\mathcal{L}(f, g, \pi_x)$ kan uttrykkes ved formel 2.19

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2 \quad (2.19)$$

hvor

$$\pi_x(z) = \exp\left(\frac{-D(x, z)^2}{\sigma^2}\right)$$

som av Ribeiro et al. 2016 er definert som en “eksponentiell kernel definert med en avstandsfunksjon D og med bredde σ ” [117]. Videre er $f(z)$ den originale modellens klassifisering av z og $g(z') = w_g \cdot z'$ som representerer forklaringsmodellens prediksjon av z basert på forklaringsmodellens treningssett \mathcal{Z} [117].

3. Metode

Dette kapitlet tar for seg fremgangsmåten og utstyret, som ble brukt for å oppnå resultatene i denne oppgaven. Første del av kapitlet redegjør for hvordan data fra toppidrettsutøvere ble samlet inn, og hvordan disse ble analysert. I andre del forklares stegene i å trene en maskinlæringsmodell basert på åpne EKG fra den generelle befolkning.

3.1 EKG-registreringer på idrettsutøvere

I denne masteroppgaven undersøkes det blant annet hvorvidt dagens innebygde EKG-tolkningsalgoritmer gjør feil når de analyserer EKG fra idrettsutøvere. For å finne ut av dette ble idrettsutøvere fra fem ulike idretter invitert til å delta. Det ble samlet inn 10 sekunders 12-avlednings hvile-EKG, samt noen personopplysninger fra hver av deltakerne.

3.1.1 Utstyr

GE MAC VU360TM

En moderne elektrokardiograf fra General Electric (GE) av typen, GE MAC VU360TM, ble brukt til å registrere EKG på idrettsutøverne. Apparatet var utstyrt med 10 ledninger som gjorde det mulig å utføre et 12-avlednings EKG. Konfigurasjonen som ble brukt, på GE MAC VU360TM, er lagt ved i appendiks A. Et bilde av elektrokardiografen er vist i figur 3.1.



Figur 3.1: GE MAC VU360TM elektrokardiograf sett fra operatørens perspektiv. GE Healthcare[©]. Figuren er gjengitt med tillatelse fra GE Healthcare Norge.

Elektroder og plassering

GE MAC VU360TM har en digital plasseringsassistent, som benytter seg av impedansen mellom elektrodene for å avgjøre om elektrodene er plassert riktig på kroppen. Denne funksjonen ble benyttet for å kvalitetssikre at plasseringen av elektrodene ble gjort riktig. Elektrodene som ble benyttet til ekstremitetsavledningene ble plassert på håndleddene og ankene. Elektrodene for de prekordiale avledningene ble plassert slik Figur 2.6 viser.

Elektrodene som ble brukt under EKG-registreringen er av typen Ambu[©] BlueSensorTM Q [122] fra Ballerup i Danmark. Elektroden er vist i figur 3.2, og er en engangselektrode ment for EKG-registreringer.



Figur 3.2: Ambu[©] BlueSensorTM Q. Elektroden ble brukt til å registrere EKG på toppidrettsutøvere i arbeidet med denne masteroppgaven. Figuren er gjengitt med tillatelse fra Ambu[©].

GE MarquetteTM SL12

GE MAC VU360TM-apparatet som ble brukt i denne oppgaven er utstyrt med en innbygget tolkningsalgoritme, GE MarquetteTM SL12 (versjon 23). Algoritmen tolket EKGGet automatisk etter at hver enkelt registrering var utført.

Hvert EKG ble automatisk lagret på GE MAC VU360TM-apparatet i filformatet .ecg. Filene ble overført via USB til en frittstående PC med programvaren CardioSoftTM (Versjon V6.73).

CardioSoftTM

CardioSoftTM er en programvare fra GE som kan åpne og behandle EKG-filene fra GE MAC VU360TM. CardioSoftTM gir brukeren mulighet til blant annet å eksportere EKG-data til andre filformat som Excel, XML, PDF og Word. I denne oppgaven ble eksportering til XML og PDF benyttet. PDF-formatet ble senere benyttet av kardiologen for å analysere EKGene. XML-formatet ble valgt fordi det ga mest informasjon. XML-formatet var det eneste filformatet som inneholdt rådataene fra EKGGet. PDF og XML-filene som ble eksportert fra

CardioSoftTM ble lagt i en kryptert mappe, deretter komprimert og importert til TSD (Tjenester for Sensitive Data).

3.1.2 Planlegging, rekruttering og utføring av registreringene

Søknadsprosess

For å innhente fysiologiske data og lagre persondata trengs godkjenning fra Norsk senter for forskningsdata (NSD). I tillegg trengs godkjenning fra Regional etisk komite (REK) for prosjekter der formålet er å skaffe ny kunnskap om helse og sykdommer. Dessuten har Universitetet i Oslo (UiO) egne rutiner for innhenting og oppbevaring av persondata. Figur 3.3 illustrer prosessen i denne masteroppgaven fra REK-søknad til rekruttering av idrettsutøvere kunne begynne.

Prosjektet falt ikke under REKs mandat, derfor gjorde de heller ingen vurdering av prosjektet. Derimot er prosjektet godkjent og registrert hos NSD og Forskpro¹. I tillegg er TSD benyttet for å oppbevare og analysere de innsamlede dataene under prosjektperioden. TSD gir brukeren et sikkert prosjektområde på en virtuell maskin, som oppfyller lovkravene om behandling og lagring av sensitive data.

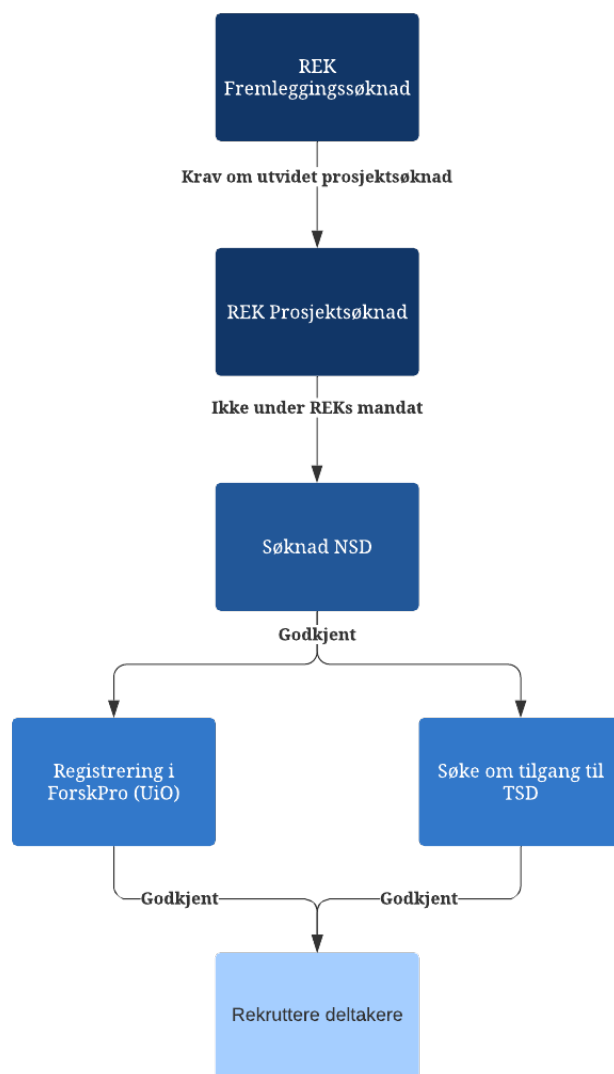
Rekruttering og kriterier

Kriteriene for deltakelse i denne studien var at deltakeren måtte være tilknyttet et landslag innen idrettene roing, padling, landeveis sykkel, triatlon eller lengdeløpsskøyter. Utøverne måtte være aktive eller ha lagt opp under ett år før studien startet. Deltakere ble rekruttert ved å sende ledelsen i de forskjellige særforbundene et informasjonsskriv med invitasjon til å delta i studien. Særforbundene som ble kontaktet var roing, padling, landeveis sykkel, triatlon og lengdeløpsskøyter. Utøverne som ønsket å være med ga et uformelt svar, og senere ble det avtalt en tid for når registreringen skulle finne sted.

Testdag

EKG-registreringene fant sted i 4.etasje i Sem Sælands vei 24 (fysikkbygget) på Blindern ved UiO. Registreringene ble utført i tidsrommet 13. februar til 3. mars 2020. Deltakerne ble på ny informert om studien ved oppmøte på testdagen, og samtlige skrev under på samtykkeskjemaet. Deltakerne ga informasjon om alder, kjønn, idrett og treningstimer siste 3 år. I tillegg ble det utført et 10 sekunders 12-avlednings hvile-EKG. Hver deltaker måtte også skrive navn og kontaktopplysninger på et eget dokument. En unik fire-sifferet koblingsnøkkel ble skrevet på dokumentet med kontaktinformasjonen, og på det andre dokumentet med de resterende personopplysningene. Dokumentet med navn, kontaktopplysninger og koblingsnøkler ble arkivert etter UiO's retningslinjer og låst inn i en safe. Dokumentet med de resterende personopplysningene ble lagt over i TSD, og deretter ble papirversjonen ble makulert. Denne prosessen sikret at alle personopplysninger ble pseudonymiserte.

¹Dette prosjektet er registrert i Forskpro: <https://www.uio.no/for-ansatte/arbeidstotte/fa/forskpro/prosjekter/mn/fysikk/elektronikk/elektrokardiografisk-tolkning-av-idrettshjarter-ved-bruk-av-algoritmer-en-pilotstudie/index.html>



Figur 3.3: Flytskjemaet illustrerer søknadsprosessen i denne studien.

Oppbevaring av data etter prosjektslutt

I samtykkeskjemaet ble deltakerne i studien spurt om deres EKG kunne publiseres, anonymisert, etter prosjektslutt. Samtlige deltakere samtykket til publisering av sitt EKG. Alle EKGene vil derfor publiseres på en åpen database for fysiologiske data etter prosjektslutt². Så langt jeg vet vil dette bli den første åpne databasen med EKG fra toppidrettsutøvere.

3.1.3 Analyse av innhentede EKG-data

Analyse utført av MarquetteTM SL12

MarquetteTM SL12-algoritmen utførte en analyse av EKGene automatisk rett etter at hver enkelt registrering var utført. Analysen ble gitt i form av en tolkningstekst, som kan leses av på det utskrevne EKGGet eller i et av de andre filformatene, som kan eksporteres fra CardioSoftTM. Algoritmene i MarquetteTM SL12 (versjon 23), som ble benyttet i denne studien er beskrevet i 12SL Physician's Guide [123].

Analyse utført av kardiologene

EKG-analysen ble utført i to steg av kardiologene. Først utførte en kardiolog en analyse av hvert enkelt EKG direkte etter at det var registrert. Hensikten med dette var å kunne oppdage eventuelle sykdomstegn hos deltakerne tidlig. Steg to ble utført av en annen kardiolog. Denne kardiologen fikk tilgang til alle EKG-opptakene i PDF-format. PDF-dokumentet inkluderte både signaler fra de 12 avledningene og tolkningstekstene fra GE Marquette SL 12. Kardiologen tolket hvert av EKGene både som om de tilhørte idrettsutøvere, og som om de tilhørte personer fra den generelle befolkning. EKGene, som ble analysert av kardiologen, var merket med pseudonymiseringskoden slik at det skulle være mulig å finne tilbake til deltakeren dersom noe patologisk ble funnet.

Sammenligning mellom MarquetteTM SL12 og kardiologen

For å kunne sammenligne kardiologens tolkningsrapport med tolkningene fra MarquetteTM SL12 måtte de mer nyanserte tolkningene fra kardiologen omformes til et enklere format, som gjorde de sammenlignbare med MarquetteTM SL12 algoritmen. Kardiologens tolkninger ble derfor binært klassifisert innenfor alle diagnosene som var gitt av MarquetteTM SL12 på de 28 EKGene. I tillegg ble hvert enkelt EKG kategorisert som normalt, i grenseområdet eller unormalt av både MarquetteTM SL12 og av kardiologen. Kardiologen klassifiserte hvert EKG som normalt, i grenseområdet eller unormalt både som om de tilhørte idrettsutøvere, og som om de var fra den generelle befolkning.

²EKGene innhentet i denne masteroppgaven vil publiseres her: <https://physionet.org/>

3.1.4 Skåringsfunksjoner

For å sammenligne enigheten mellom kardiologen og MarquetteTM SL12 for de to forskjellige tolkningene fra kardiologen er F_1 (formel 3.1), F_2 (formel 3.2) og G_2 -skår (formel 3.3) benyttet.

$$F_1 = 2 \cdot \frac{\textit{Presisjon} \cdot \textit{Sensitivitet}}{\textit{Presisjon} + \textit{Sensitivitet}} \quad (3.1)$$

$$F_2 = (1 + 2^2) \cdot \frac{\textit{Presisjon} \cdot \textit{Sensitivitet}}{(2^2 \cdot \textit{Presisjon}) + \textit{Sensitivitet}} \quad (3.2)$$

$$G_2 = \frac{SP}{SP + FP + 2 \cdot FN} \quad (3.3)$$

F_1 , F_2 og G_2 -skårene er beregnet som makro-skårer. Ved å bruke F_1 -skår som eksempel betyr dette at F_1 -skåren er beregnet for hver klasse n , og $F_{1_{makro}}$ er gjennomsnittet av F_1 -skårene for alle n klasser (formel 3.4).

$$F_{1_{makro}} = \frac{1}{n} \sum_{i=1}^n \frac{F_{1_n}}{n} \quad (3.4)$$

Videre i oppgaven vil $F_{1_{makro}}$, $F_{2_{makro}}$ og $G_{2_{makro}}$ kun bli omtalt som F_1 , F_2 og G_2 -skår.

3.2 Maskinl ring p  EKG fra den generelle befolkning

EKGene fra idrettsut verne er ikke tilstrekkelig for   trene eller validere en maskinl ringsmodell. Derfor er det, i denne delen av oppgaven, tatt i bruk et datasett fra den generelle befolkning for   trene og evaluere forskjellige metoder for maskinl ring p  EKG-data. Datasettet er fra PhysioNet/Computation in Cardiology (CinC) Challenge 2020, som er en  rlig konkurranse i   lage tolkingsmodeller for hjertelaterte data. I 2020 var utfordringen   klassifisere forskjellige diagnoser basert p  12-avlednings EKG, alder og kj nn. [28]. I dette delkapittelet beskrives metodene som ble brukt i mitt bidrag til PhysioNet/CinC Challenge 2020 [124]. I tillegg er en ny Ensemble-modell beskrevet, og et forslag til implementering av forklarbar KI for disse modellene. Koden som viser hvordan disse modellene ble implementert i TensorFlow kan ses i Appendix E. Publikasjonen som ble skrevet og posteren som ble laget i forbindelse med mitt bidrag til CinC kan ses i appendix G og appendix H.

3.2.1 Data

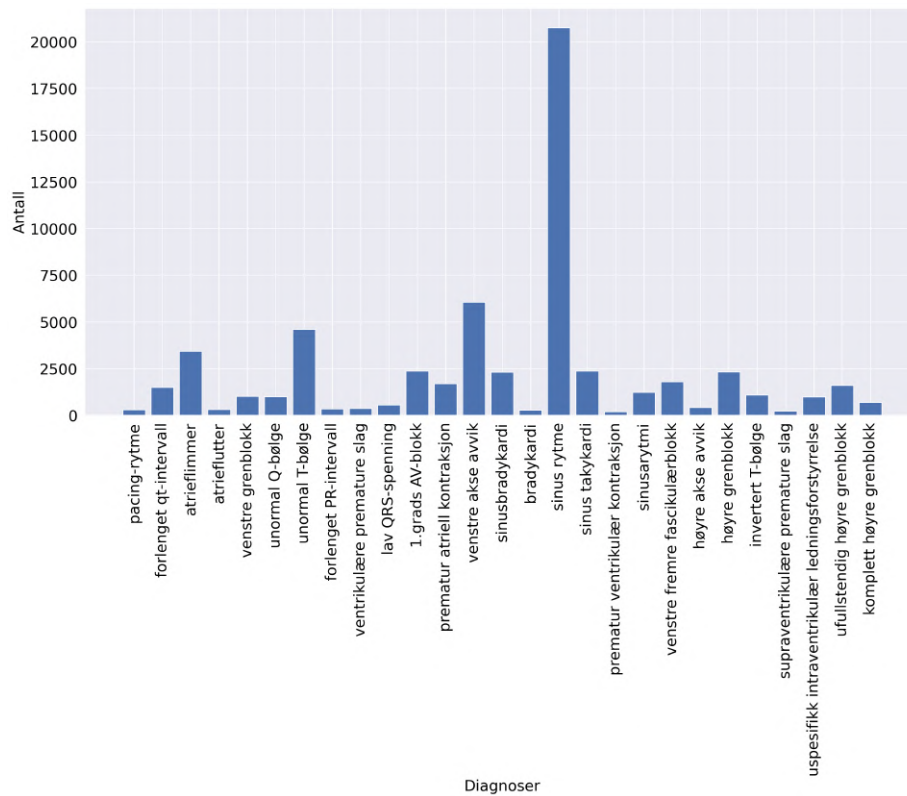
Totalt ble 43 101 EKG fra fire forskjellige kilder tilgjengeliggjort som utviklingsdata. Organisat rene av PhysioNet/CinC Challenge 2020 har i tillegg et skjult testsett p  henholdsvis 6630 og 16 630 EKG, benyttet i et skjult valideringssett og et skjult testsett. I denne masteroppgaven er det f rst og fremst utviklingsdataene som vil bli beskrevet. Tabell 3.1 viser de ulike kildene i utviklingsdataene.

Tabell 3.1: Oversikt over utviklingsdataene benyttet i denne masteroppgaven. China Physiological Signal Challenge 2018 (CPSC) består opprinnelig av 6877 EKG [125], men i PhysioNet/CinC Challenge 2020 er 3453 ekstra EKG lagt til fra et datasett kalt CPSC2018 Extra. EKGene fra Physikalisch-Technische Bundesanstalt (PTB) er fra samme opphav, men er i denne tabellen beskrevet som to kilder, blant annet fordi de har forskjellig samplingsfrekvens [126], [127]. Fra St. Petersburg Institute of Cardiological Technics (INCART) er 74 av de opprinnelige 75 opptakene brukt i dette utviklingssettet [128], [129]. Georgia 12-lead ECG Challenge (G12EC) er database fra Emory University i Atlanta, Georgia i USA [28].

Database	Antall EKG	Gj.snittlig opptakslengde (sekunder)	Samplingsfrekvens (Hz)	Kj�nn (kvinner /menn)	Gj.snittsalder (�r)
CPSC2018	10330	15,9	500	47%/53%	61,1
PTB D	516	110,8	1000	27%/73%	56,3
PTB-XL	21837	10	500	48%/52%	59,8
INCART	74	1800	257	46%/54%	58,0
G12EC	10344	10	500	46%/54%	60,5

Diagnoser i utviklingssettet

Det var totalt 111 unike diagnoser i utviklingssettet. Hvert EKG hadde minst én diagnose, men mange hadde også flere. Målet med PhysioNet/CinC Challenge 2020 var å klassifisere 27 av de 111 unike diagnosene, som var til stede i utviklingssettet. Fordelingen mellom de 27 ulike diagnosene vises i Figur 3.4.



Figur 3.4: Fordelingen mellom de 27 diagnosene i utviklingsdataene, som brukes i PhysioNet/CinC Challenge 2020.

3.2.2 Preprosessering av data

Rådataene fra EKGene var lagret i hver sin .mat-fil. I denne filen var det tolv tallrekker som representerte de tolv avledningene i EKGet. Hver .mat-fil hadde en korresponderende fil med metadata (.hea-format). Appendiks B viser en metadata-fil for en av pasientene.

CNN-modellene benyttet seg av rådataene samt alder og kjønn fra metadataene, mens *Ensemble-modellen* benyttet seg av forhåndsekstraherte egenskaper fra EKGet. Derfor ble to forskjellige preprosesseringsmetoder benyttet; én for CNN-modellene og én for *Ensemble-modellen*.

Preprosessering for CNN

Ved bruk av CNN-modeller må inngangsverdiene ha en gitt dimensjon. For CNN-modellene, benyttet i denne oppgaven ble dimensjonen på inngangsverdiene satt til å være $(n, 5000, 12)$, hvor n er et fleksibelt antall EKG, 5000 representerer antall samples i EKG-opptaket, og tallet 12 representerer antallet avledninger i EKGet. Lengden på EKGene, m , ble kuttet ned ved å fjerne alt etter 5000 samples dersom $m > 5000$ samples. Der $m < 5000$ samples ble det lagt til $5000 - m$ antall nuller slik at signalet ble 5000 samples langt.

Preprosessering for *Ensemble-modellen*

For å trekke ut egenskaper fra de 43 101 EKG-opptakene i utviklingsdataene ble en ny Python-pakke, ECG-featurizer [130], utviklet. ECG-featurizer bygger på Neurokit2 [131], Waveform-database (WFBDB) og Numpy [132].

42 720 av de 43 101 EKGene ble korrekt analysert, og deretter transformert til egenskaper av ECG-featurizer. 146 av de 42 720 EKGene ble imidlertid fjernet på grunn av manglende verdier. Dette ga et totalt datasett på 42 574 rader med korrekt ekstraherte EKG-egenskaper. ECG-featurizer ekstraherte 112 egenskaper fra hver av de 42 574 EKGene slik at det totale datasettet var en matrise $= 112 \times 42\,574$.

3.2.3 Skåringsfunksjoner

Organisatørene bak PhysioNet/CinC Challenge 2020 utviklet en spesiallaget skåringsalgoritme, PhysioNet/CinC Challenge skår, som også er brukt i denne masteroppgaven. Hensikten med PhysioNet/CinC Challenge skår var å gi poeng, slik at den reflekterte klinisk diagnostikk, hvor enkelte sykdommer ikke bør forveksles eller feiltolkes. Andre diagnoser har like behandlingsforløp eller utredningsmetoder, som gjør at konsekvensene ved forveksling er mindre. Skåringsmetoden baserte seg på å gi hver diagnose et vektall, som ga full skår for riktig diagnose, og noe poeng ved feil diagnose. Figur 3.5 viser en forvirringsmatrise W med vektallene for de ulike diagnosene.

	PR	LQT	AF	AFL	LBBB	QAb	Tab	LPR	VPB	LQRSV	IABV	PAC	LAD	SB	Brady	SNR	Stach	PVC	SA	LAnFB	RAD	RBBB	Tinv	SVPB	NSIVCB	IRBBB	CRBBB
PR	1	0.3	0.3	0.5	0.4	0.5	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	0.5	0.35	0.2	0.45	0.4	0.5	0.5	0.45	0.42	0.46	0.3	0.3	0.42
LQT	0.3	1	0.5	0.3	0.4	0.3	0.35	0.35	0.47	0.42	0.35	0.38	0.34	0.38	0.3	0.45	0.4	0.35	0.4	0.3	0.3	0.25	0.38	0.34	0.5	0.5	0.38
AF	0.3	0.5	1	0.3	0.4	0.3	0.35	0.35	0.47	0.42	0.35	0.38	0.34	0.38	0.3	0.45	0.4	0.35	0.4	0.3	0.3	0.25	0.38	0.34	0.5	0.5	0.38
AFL	0.5	0.3	0.3	1	0.4	0.5	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	0.5	0.35	0.2	0.45	0.4	0.5	0.5	0.45	0.42	0.46	0.3	0.3	0.42
LBBB	0.4	0.4	0.4	0.4	1	0.4	0.45	0.45	0.42	0.47	0.45	0.47	0.44	0.47	0.4	0.45	0.3	0.45	1	0.4	0.4	0.35	0.47	0.44	0.4	0.4	0.47
QAb	0.5	0.3	0.3	0.5	0.4	1	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	0.5	0.35	0.2	0.45	0.4	0.5	0.5	0.45	0.42	0.46	0.3	0.3	0.42
Tab	0.45	0.35	0.35	0.45	0.45	0.45	1	0.5	0.38	0.42	0.5	0.47	0.49	0.47	0.45	0.4	0.25	0.5	0.45	0.45	0.45	0.4	0.47	0.49	0.35	0.35	0.47
LPR	0.45	0.35	0.35	0.45	0.45	0.45	0.5	1	0.38	0.42	0.5	0.47	0.49	0.47	0.45	0.4	0.25	0.5	0.45	0.45	0.45	0.4	0.47	0.49	0.35	0.35	0.47
VPB	0.33	0.47	0.47	0.33	0.42	0.33	0.38	0.38	1	0.45	0.38	0.4	0.36	0.4	0.33	0.47	0.38	0.38	0.42	0.33	0.33	0.28	0.4	0.36	0.47	0.47	0.4
LQRSV	0.38	0.42	0.42	0.38	0.47	0.38	0.42	0.42	0.45	1	0.42	0.45	0.41	0.45	0.38	0.47	0.33	0.42	0.47	0.38	0.38	0.33	0.45	0.41	0.42	0.42	0.45
IABV	0.45	0.35	0.35	0.45	0.45	0.45	0.5	0.5	0.38	0.42	1	0.47	0.49	0.47	0.45	0.4	0.25	0.5	0.45	0.45	0.45	0.4	0.47	0.49	0.35	0.35	0.47
PAC	0.42	0.38	0.38	0.42	0.47	0.42	0.47	0.47	0.4	0.45	0.47	1	0.46	0.5	0.42	0.42	0.28	0.47	0.47	0.42	0.42	0.38	0.5	0.46	0.38	0.38	0.5
LAD	0.46	0.34	0.34	0.46	0.44	0.46	0.49	0.49	0.36	0.41	0.49	0.46	1	0.46	0.46	0.39	0.24	0.49	0.44	0.46	0.46	0.41	0.46	1	0.34	0.34	0.46
SB	0.42	0.38	0.38	0.42	0.47	0.42	0.47	0.47	0.4	0.45	0.47	0.5	0.46	1	0.42	0.42	0.28	0.47	0.47	0.42	0.42	0.38	0.5	0.46	0.38	0.38	1
Brady	0.5	0.3	0.3	0.5	0.4	0.5	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	1	0.35	0.2	0.45	0.4	0.5	0.5	0.45	0.42	0.46	0.3	0.3	0.42
SNR	0.35	0.45	0.45	0.35	0.45	0.35	0.4	0.4	0.47	0.47	0.4	0.42	0.39	0.42	0.35	1	0.35	0.4	0.45	0.35	0.35	0.3	0.42	0.39	0.45	0.45	0.42
Stach	0.2	0.4	0.4	0.2	0.3	0.2	0.25	0.25	0.38	0.33	0.25	0.28	0.24	0.28	0.2	0.35	1	0.25	0.3	0.2	0.2	0.15	0.28	0.24	0.4	0.4	0.28
PVC	0.45	0.35	0.35	0.45	0.45	0.45	0.5	0.5	0.38	0.42	0.5	0.47	0.49	0.47	0.45	0.4	0.25	1	0.45	0.45	0.45	0.4	0.47	0.49	0.35	0.35	0.47
SA	0.4	0.4	0.4	0.4	1	0.4	0.45	0.45	0.42	0.47	0.45	0.47	0.44	0.47	0.4	0.45	0.3	0.45	1	0.4	0.4	0.35	0.47	0.44	0.4	0.4	0.47
LAnFB	0.5	0.3	0.3	0.5	0.4	0.5	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	0.5	0.35	0.2	0.45	0.4	1	0.5	0.45	0.42	0.46	0.3	0.3	0.42
RAD	0.5	0.3	0.3	0.5	0.4	0.5	0.45	0.45	0.33	0.38	0.45	0.42	0.46	0.42	0.5	0.35	0.2	0.45	0.4	0.5	1	0.45	0.42	0.46	0.3	0.3	0.42
RBBB	0.45	0.25	0.25	0.45	0.35	0.45	0.4	0.4	0.28	0.33	0.4	0.38	0.41	0.38	0.45	0.3	0.15	0.4	0.35	0.45	0.45	1	0.38	0.41	0.25	0.25	0.38
Tinv	0.42	0.38	0.38	0.42	0.47	0.42	0.47	0.47	0.4	0.45	0.47	0.5	0.46	0.5	0.42	0.42	0.28	0.47	0.47	0.42	0.42	0.38	1	0.46	0.38	0.38	0.5
SVPB	0.46	0.34	0.34	0.46	0.44	0.46	0.49	0.49	0.36	0.41	0.49	0.46	1	0.46	0.46	0.39	0.24	0.49	0.44	0.46	0.46	0.41	0.46	1	0.34	0.34	0.46
NSIVCB	0.3	0.5	0.5	0.3	0.4	0.3	0.35	0.35	0.47	0.42	0.35	0.38	0.34	0.38	0.3	0.45	0.4	0.35	0.4	0.3	0.3	0.25	0.38	0.34	1	0.5	0.38
IRBBB	0.3	0.5	0.5	0.3	0.4	0.3	0.35	0.35	0.47	0.42	0.35	0.38	0.34	0.38	0.3	0.45	0.4	0.35	0.4	0.3	0.3	0.25	0.38	0.34	0.5	1	0.38
CRBBB	0.42	0.38	0.38	0.42	0.47	0.42	0.47	0.47	0.4	0.45	0.47	0.5	0.46	1	0.42	0.42	0.28	0.47	0.47	0.42	0.42	0.38	0.5	0.46	0.38	0.38	1

Figur 3.5: Forvirringsmatrisen viser vektallene som ble brukt for å skåre modellene med PhysioNet/CinC Challenge skår. Modellen får full skår dersom den predikerer riktig, men feil diagnoser vil også gi noen poeng.

Den spesiallagde skåringsalgoritmen for PhysioNet/CinC Challenge 2020 er gitt ved $s_{normalisert}$, som vises i formel 3.5. $s_{normalisert}$ er en score fra -1 til +1, og beregnes ved $s_{observert}$, $s_{normalklasse}$ og s_{sann} .

$$s_{normalisert} = \frac{s_{observert} - s_{normalklasse}}{s_{sann} - s_{normalklasse}} \quad (3.5)$$

$s_{observert}$ beregnes ved summere opp den elementvise multiplikasjon av de to matrisene A og W med identisk dimensjon. A er en forvirringsmatrise, som inneholder elementene a_{ij} , og vektallsmatrisen W , inneholder vektene w_{ij} . Forvirringsmatrisen A , er beregnet ut i fra modellens predikerte diagnoser opp mot de sanne diagnosene. Summen av alle produktene $a_{ij} \cdot w_{ij}$, gir skalaren $s_{observert}$, slik det er vist i formel 3.6.

$$s_{observert} = \sum_i^n \sum_j^n w_{ij} a_{ij} \quad (3.6)$$

$s_{normalklasse}$ representerer en inaktiv modell. En modell regnes som inaktiv dersom den klassifiserer alle EKGene som normale (normal sinusrytme). Matrisen B , som består av elementene b_{ij} , er en forvirringsmatrise der prediksjonene kun består av normalklassen (normal sinusrytme), og har identisk dimensjon som W . Produktet av elementene b_{ij} og vektallene w_{ij} summeres, og gir skalaren $s_{normalklasse}$ som vist i formel 3.7.

$$s_{normalklasse} = \sum_i^n \sum_j^n w_{ij} b_{ij} \quad (3.7)$$

s_{sann} , gitt ved formel 3.8, beregnes ved summen av den elementvise multiplikasjonen av w_{ij} og c_{ij} , hvor c_{ij} er elementene i matrisen C . C er en forvirringsmatrise med kun korrekte klassifiseringer, og har samme dimensjon som W . s_{sann} er en skalar, som representerer skåren til en modell som bare klassifiserer riktig.

$$s_{sann} = \sum_i^n \sum_j^n w_{ij} c_{ij} \quad (3.8)$$

PhysioNet/CinC Challenge skår gjør at en modell som klassifiserer mange falske positive vil få en negativ skår. Dette reflekterer klinikken der falske-positive kan utgjøre en byrde for helsesystemet, og ikke minst også for pasientene.

I tillegg til PhysioNet/CinC Challenge skår ble også F_1 -skår (formel 3.1), F_2 -skår (formel 3.2) og G_2 -skår (formel 3.3) beregnet for de ulike modellene i denne delen av oppgaven. I likhet med skårene beskrevet i kapittel 3.1.4, er F_1 , F_2 og G_2 beregnet som makro skårer.

3.2.4 Trening og valideringsdata

For å gi en indikasjon på hvor god modellene er til å klassifisere usette data, ble modellene kryssvalidert på utviklingsdataene. Utviklingsdataene ble delt opp i trenings- (90%) og valideringsdata (10%), ved å bruke stratifisert 10-ganger kryssvalidering [133]. Den stratifiserte kryssvalideringen gjorde at forholdet mellom diagnosene ble likt i trenings- og valideringsdataene. Modellene ble for hver runde i kryssvalideringen testet på både treningsdataene og valideringsdataene.

3.2.5 Konvolusjonelle nevralt nettverk

Åtte ulike kombinasjoner av CNN-modeller ble utviklet ved bruk av Tensorflow rammeverket [105] i Python. Fire modellarkitekturer er brukt for å lage de åtte kombinasjonene. To av arkitekturerne var CNN-modeller, ett flerlagsperceptron (MLP) og én regelbasert algoritme.

De to CNN-arkitekturerne ble valgt ut på bakgrunn av en gjennomgangsstudie av Fawaz HI et al 2019 [134], som så på prestasjonen til 1-dimensjonale CNN-modeller på forskjellige datasett, blant annet på EKG. De to CNN-arkitekturerne som presterte best på EKGene i gjennomgangsstudien ble tatt i bruk i denne masteroppgaven. Disse to modellarkitekturerne kalles Fully Convolutional Networks (FCN) og Encoder.

De åtte ulike kombinasjonene er listet opp i tabell 3.2

Tabell 3.2: De åtte CNN-modellene som er listet opp i denne tabellen ble utviklet og testet i arbeidet med PhysioNet/CinC Challenge 2020 [124].

Model
1) FCN
2) Encoder
3) FCN alder, kjønn
4) Encoder alder, kjønn
5) Encoder FCN
6) Encoder FCN alder, kjønn
7) Encoder FCN + Regelbasert modell
8) Encoder FCN alder, kjønn + Regelbasert modell

MLP-modell

Hensikten med *MLP-modellen* var å benytte seg av personenes alder og kjønn. Modellen hadde 2 inngangsnevner, 50 nevrner i det skjulte laget og 2 nevrner i utgangslaget. Mellom inngangslaget og det skjulte laget ble det benyttet ReLU aktiveringsfunksjon, videre ble det benyttet Sigmoid aktivering i utgangslaget. *MLP-modellen* ble kun benyttet i parallell med en CNN-modell og slått sammen i nest siste lag av CNN-modellen.

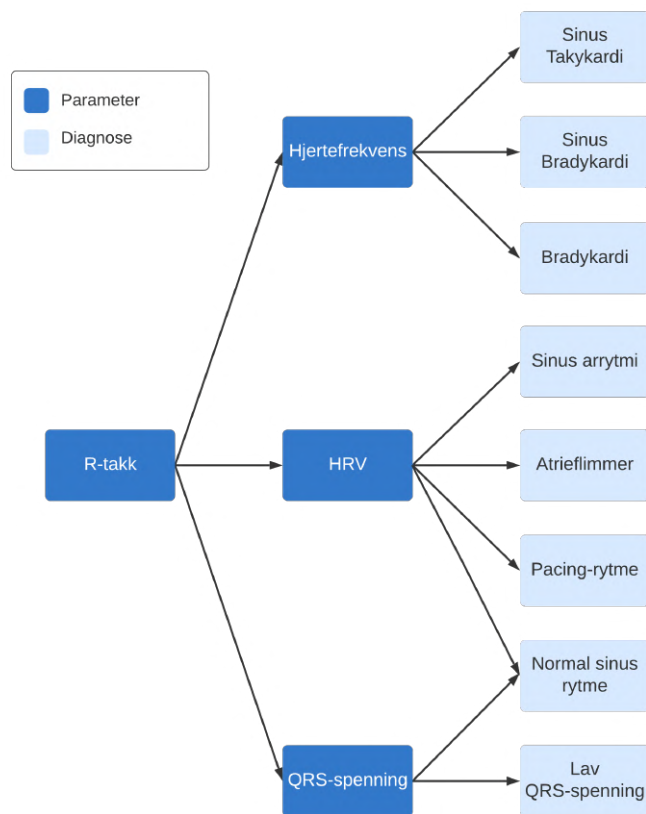
Regelbasert algoritme

Den *regelbaserte algoritmen* benyttet seg av Pan-Tompkins algoritme [135], for å detektere R-takkene i avledning *II* i EKG. Basert på R-takkene ble hjertefrekvens, maksimal QRS-spenning og hjerteratevariabilitet (HRV) beregnet. Algoritmen Root Mean Square of Successive Differences (RMSSD) mellom normale hjerteslag ble benyttet for å beregne HRV. Formel 3.9 viser utregningen av RMSSD hvor *RR* er intervallet mellom R-takkene, og *N* er antallet R-takker.

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{N-1} (RR_i - RR_{i+1})^2}{N-1}} \quad (3.9)$$

Den *regelbaserte algoritmen* ble brukt i kombinasjon med en CNN-modell eller en kombinert CNN og MLP-modell. Likevel klassifiserte den *regelbaserte algoritmen* uavhengig av CNN og MLP-modellene. I de tilfellene der det var uenighet mellom den *regelbaserte algoritmen* og CNN-modellen, overskrev den *regelbaserte algoritmen* prediksjonen fra CNN-modellen.

8 av 27 diagnoser ble klassifisert av den *regelbaserte algoritmen*. Figur 3.6 viser relasjonen mellom beregnede parameterne og diagnosene, klassifisert av den *regelbaserte algoritmen*. Beregningene av parameterne ble gjort ved å bruke en try-except-struktur, slik som i tabell 3.3. Dette var for å unngå at programmet skulle slutte å kjøre dersom det ikke klarte å beregne et parameter. Dersom et parameter (figur 3.6) ikke lot seg beregne ble heller ikke de tilhørende diagnosene (figur 3.6) klassifisert av den *regelbaserte algoritmen*.



Figur 3.6: Relasjonen mellom parametere og diagnoser i den *regelbaserte algoritmen*. Sinus takykardi, sinus bradykardi og bradykardi ble klassifisert basert på hjertefrekvens. Sinusarytmi, atrieflimmer og pacing-rytme ble klassifisert av HRV-beregningen. Lav QRS-spenning ble bestemt av parameteren QRS-spenning. Normal sinusrytme ble klassifisert av både HRV-beregningen og QRS-spenning

Tabell 3.3: Try-except-struktur for beregning av RMSSD

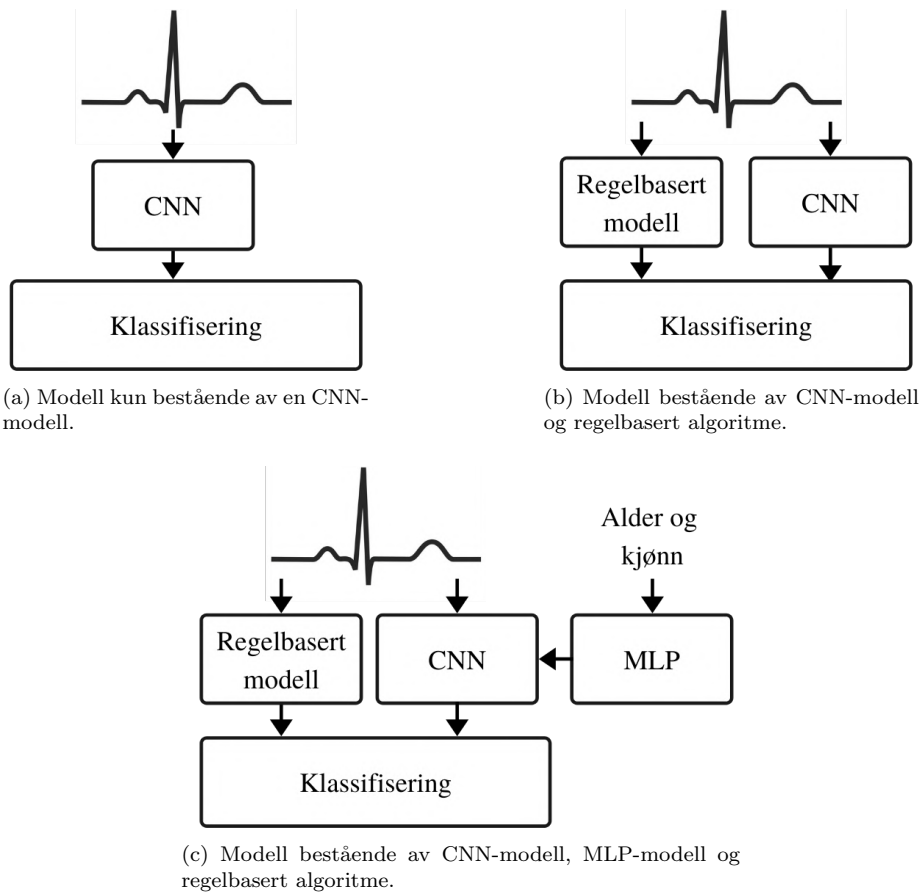
```

try:
    rmssd = np.mean(np.square(np.diff(rr_interval)))
except:
    print("did not manage to compute RMSSD")

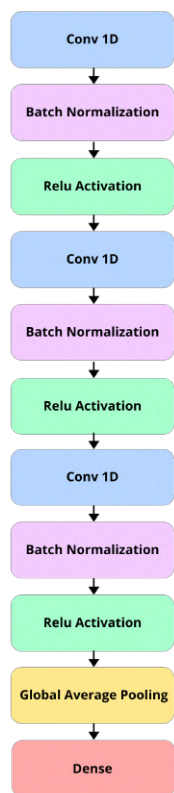
```

Sammensetning av de ulike modellene

Figur 3.7 viser hvordan CNN-modeller, *MLP-modellen* og den *regelbaserte algoritmen* ble satt sammen i de ulike kombinasjonene. I figur 3.7 kan boksen med navnet CNN bestå av *FCN-modellen*, *Encoder-modellen* eller en kombinasjon av de to.



Figur 3.7: Oppbygningen til de ulike CNN-baserte modellene. CNN-modeller, *MLP-modellen* og den *regelbaserte algoritmen* ble kombinert på åtte forskjellige måter. CNN-boksen i a, b og c kunne bestå av en *FCN-modell*, en *Encoder-modell* eller en kombinasjon av de to modellarkitekturene.



Figur 3.8: Arkitekturen for en FCN-modell. Hver blokk illustrerer et lag eller en matematisk operasjon. Conv 1D er et 1-dimensjonalt konvolusjonelt lag, og Dense er modellens siste lag med 27 utgangsnevroner og sigmoid-aktivering.

sjonslaget ble det benyttet instance normalization, PReLU-aktivering og et dropout-lag med dropout-rate = 0,2. Dropout-lagets utgang gikk til en oppmerksomhetsmekanisme. Ved oppmerksomhetsmekanismen var signallengden til EKGGet redusert fra 5000 til 1250, og antall filter ekspandert fra 12 til 512. I oppmerksomhetsmekanismen ble de ekstraherte egenskapene delt i to med hensyn på antall filter. Det ene settet ble brukt som inngang til softmax-laget, og deretter multiplisert med det andre settet, slik det er forklart i delkapittel 2.5.3.

FCN

FCN-modellen bestod av tre konvolusjonelle lag med batch normalisering og ReLU-aktivering, slik figur 3.8 viser. Dybden på konvolusjonsfiltrene i de konvolusjonelle lagene var på henholdsvis 128, 256 og 128, og størrelsene på konvolusjonskjernene var på henholdsvis 8, 5 og 3. Videre ble et globalt gjennomsnittlig pooling-lag brukt i det nest siste laget for å redusere dimensjonaliteten til dataene. Til slutt ble Sigmoid-aktivering brukt til å gjøre prediksjoner innenfor 27 klasser i det siste Dense-laget. De 27 klassene tilsvarer de 27 diagnosene som skulle klassifiseres i denne oppgaven.

Encoder

Arkitekturen til *Encoder-modellen* er illustrert ved figur 3.9. Modellen er bygget opp av to blokker, slik de stiplede linjene i figur 3.9 viser. Hver blokk besto av et konvolusjonslag med instance normalization, PReLU-aktivering, et dropout-lag og et maks pool-lag. I første blokk hadde konvolusjonslaget en filterdybde = 128, og en konvolusjonskjerne = 5. I andre blokk var filterdybden = 256 med en konvolusjonskjerne = 11. Begge blokkene hadde en dropout-rate = 0,2 og maks pool-størrelse = 2.

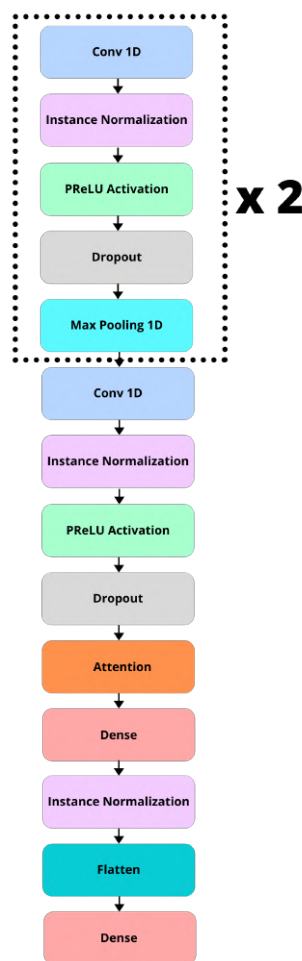
Etter de to blokkene fulgte et tredje konvolusjonslag med en filterdybde = 512 og en konvolusjonskjerne = 21. Etter konvolu-

Utgangen av oppmerksomhetsmekanismen ble brukt som inngang til et dense-lag med 256 nevroner med sigmoid-aktivering, etterfulgt av et instance normalization-lag. Etter dette ble et flatten-lag brukt for å flate ut matrisen med egenskaper. Dette gjorde at egenskapene fra hvert filter ble lagt etter hverandre i en lang tallrekke (vektor). Til slutt fulgte det avsluttende dense-laget med sigmoid-aktivering, som hadde 27 nevroner hvor hvert nevron representerte en av de 27 diagnose-

Terskeloptimalisering

De CNN-baserte modellene trengte prediksjonsterskler for å gjøre om utgangen fra det siste laget med Sigmoid-aktivering, til en binær klassifisering. Nelder-Mead downhill simplex method [136], [137], ble brukt for å optimalisere terskelverdiene individuelt med mål om å finne maksimal verdi av PhysioNet/CinC Challenge skår. Denne optimaliseringsmetoden krever mye regnekraft, og derfor ble kun $\frac{1}{10}$ av treningssettet brukt for å optimalisere tersklene. For å trekke ut en representativ del av treningssettet ble en stratifisert uttrekning brukt. Den stratifiserte uttrekningen gjorde at terskeloptimaliseringssettet fikk lik fordeling av diagnoser som treningssettet.

Nelder-Mead downhill simplex-metoden brukes til å finne det lokale minimumet for en funksjon, ved hjelp av selve funksjonen og en innledende gjetning av funksjonens optimale variabel. I den første runden av 10-ganger kryssvalideringen ble den første gjetningen for Nelder-Mead optimaliseringsalgoritmen funnet ved å gi en 27 elementer lang vektor verdien 1, og multiplisere vektoren med en variabel. Variabelen ble først gitt verdien 0, deretter økt med 0,05 helt opp til 1. For hver økning ble variabelen multiplisert med den 27 element lange vektoren. Vektoren ble deretter testet som prediksjonsterskel, og PhysioNet Challenge-skåren ble registrert. De neste rundene i 10-ganger kryssvalideringen ble den optimale terskelen fra forrige kryssvalideringsrunde brukt som første gjetning for Nelder-Mead-optimaliseringsalgoritmen.



Figur 3.9: Arkitektur for en Encoder modell. Hver blokk illustrerer et lag eller en matematisk operasjon.

Vektet læring

Som en følge av at treningsdataene var skjevfordelte med hensyn på diagnoser, slik det er vist i figur 3.4, ble hver diagnose gitt et vektall ut i fra hvor ofte diagnosen opptrådte i utviklingssettet. Vektallet ble brukt i beregningen av cost-funksjonen. Hensikten med vektallet var å balansere læringen, slik at modellene lærte like mye fra hver diagnose uavhengig av hyppigheten til diagnosen i utviklingssettet. Diagnoser med lav prevalens i utviklingssettet fikk et høyt vektall, mens diagnoser med høy prevalens fikk et lavt vektall.

Adaptiv læringsrate og tidlig treningsstans

Læringsrate for alle modeller, trent i denne oppgaven, startet på 0.001 og ble endret underveis i treningen ved hjelp av adaptiv læringsrate. Læringsraten ble senket med en faktor på 10 hver gang AUC-skåren for valideringsdataene ikke forbedret seg. Til slutt ble hele treningen stoppet ved hjelp av en tidlig treningsstans-funksjon. treningsstans-funksjonen overvåket AUC-skåren på valideringsdataene, og dersom AUC-skåren ikke forbedret seg eller ble dårligere to epoker på rad ble treningen stoppet. Dette var en metode for å motvirke overtilpasning på treningssettet.

Øvrige parametere

Alle CNN-modellene ble trent med batch size = 30 og Adam-optimizer. Adam-optimizere fungerer etter samme prinsipp som gradient decent-algoritmen som er forklart i kapittel 2.5.2, men i tillegg er parametrene; v_t og s_t lagt til. v_t beregner det eksponentielt fallende gjennomsnittet av tidligere gradienter, og s_t beregner det eksponentielt fallende gjennomsnitt av tidligere kvadratiske gradienter (formel 3.11) [138].

$$v_t = \beta_1 \cdot v_{t-1} - (1 - \beta_1) \cdot \nabla_{\mathcal{C}} \quad (3.10)$$

$$s_t = \beta_2 \cdot s_{t-1} - (1 - \beta_2) \cdot \nabla_{\mathcal{C}}^2 \quad (3.11)$$

Adam-optimizere kan beskrives ved formel 3.12:

$$\theta_{i+1} = \theta_i - \eta \frac{v_t}{\sqrt{s_t + \epsilon}} \frac{\partial \mathcal{C}}{\partial \theta_i} \quad (3.12)$$

Binær kryssentropi ble benyttet som cost-funksjon for samtlige CNN-modeller. Binær kryssentropi er forklart i kapittel 2.5.2.

For økt ytelse, og tilgang til Graphics Processing Unit (GPU) ble utregningsressurser i Google Colab Pro benyttet. CNN-modellene ble trent i Google Colab Pro med Tesla P100-PCIE 16 GB GPU og 26,30 GB RAM.

3.2.6 Ensemble-modell

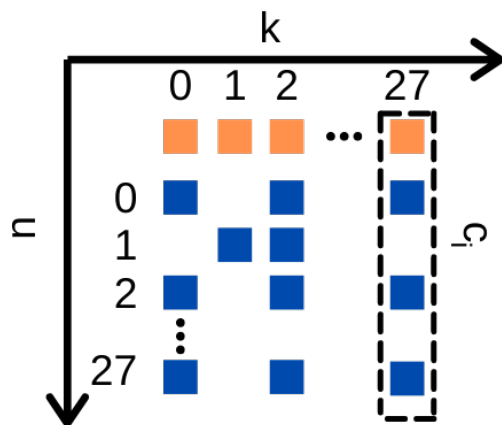
I tillegg til de CNN-baserte modellene, ble det utviklet en *ensemble-modell* ved bruk av Scikit-multilearn [108], som bygger på Scikit-learn rammeverket [133]. *Ensemble-modellen* bestod av en label space partitioning classifier [139], klassifiseringskjeder [109] og tilfeldige beslutningsskoger [111], som er beskrevet

i kapittel 2.5.4. *Ensemble-modellen* ble trent på forhåndsekstraherte egenskaper fra EKGene. Egenskapene ble ekstrahert ved bruk av ECG-featurizer [130].

Klyngene i label space partitioning classifier ble bestemt ved å bruke fixed label space clusterer, slik at klyngene kunne defineres manuelt. Klyngene ble definert ved å søke gjennom de 27 klassene K , i treningsdataene. For hver klasse k_i var det n_i av de resterende 26 klassene, som på tvers av hele treningssettet, eksisterte sammen med k_i . Vektoren c_i kan skrives:

$$c_i = k_i + n_i$$

der c_i er en av 27 vektorer som danner klyngen C . Dette er illustrert i figur 3.10



Figur 3.10: Oppdeling av klynger til *Ensemble-modellen*. K er de 27 forskjellige klassene, n er de forskjellige klassene som opptrådte sammen med hver enkelt k_i på tvers av hele treningssettet. c_i er én vektor av de totalt 27 vektorene i klyngen C .

27 nye klynger, med ulikt antall klasser i hver klynge, ble bestemt for hver av de ti kryssvalideringsrundene.

En klassifiseringskjede ble trent for hver av de 27 klyngene, bestemt av label space partitioning classifier-algoritmen. For hvert klassifiseringsproblem i klassifiseringskjeden, ble det trent en tilfeldig beslutningsskog med 5 beslutningstrær (estimatorer) i hver tilfeldige beslutningsskog. Rekkefølgen på klassene i klassifiseringskjedene var den samme som rekkefølgen på klassene i den genererte klyngen, C .

Ensemble-modellen ble trent på alle tilgjengelige prosessorer ($n_jobs = -1$) i Kaggle med 16 GB RAM tilgjengelig.

3.2.7 Forklaringsmodeller

For å forklare resultatet fra EKG-klassifiseringsmodellene ble LIME brukt [117]. Forklaringsmodeller og LIME er gjort rede for i kapittel 2.5.6.

Klassifiseringsmodellene som ble forklart var *Ensemble-modellen* og *Encoder-modellen*. Siden disse modellene bruker to forskjellige inngangsdata ble også forklaringsmetodene for disse to modellene forskjellig.

Ensemble-modellen ble først trent på 90% av utviklingsdataene, ved bruk av stratifisert splitt av dataene. Deretter ble en forklaringsmodell trent på prediksjonene fra *Ensemble-modellen*, og 5000 av treningsdataene ble brukt til å trene forklaringsmodellen. Til slutt ble LIME brukt til å forklare klassifiseringsresultatet til to av EKGene som ikke ble brukt som treningsdata. Den sanne diagnosen for de to EKGene var sinustakykardi og T-bølgeinversjon. Forklaringsmodellene ble trent i Kaggle med 16 GB RAM tilgjengelig.

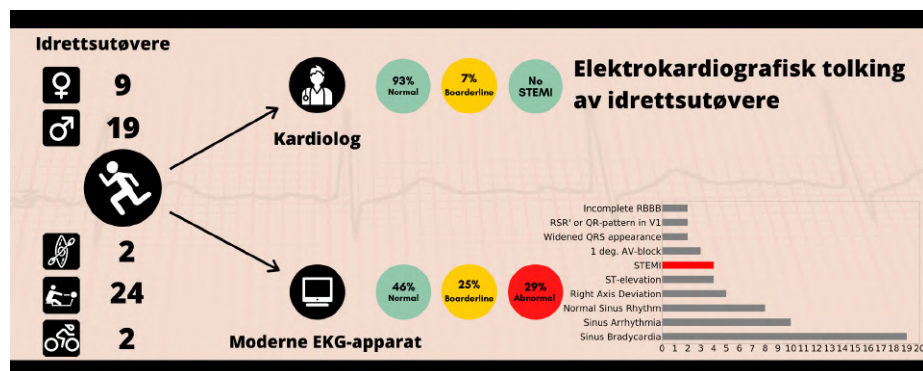
Encoder-modellen ble først trent på 90% av utviklingsdataene, ved bruk av stratifisert splitt av dataene. Deretter ble en forklaringsmodell trent på klassifiseringene til *Encoder-modellen*, og 4000 av treningsdataene ble brukt til å trene forklaringsmodellen. LIME ble brukt til å forklare klassifiseringsresultatet til to av EKGene, som ikke ble brukt som treningsdata. Den sanne diagnosen for de to EKGene var sinustakykardi og T-bølgeinversjon. Forklaringsmodellene ble trent i Google Colab Pro med Tesla P100-PCIE-16GB GPU og 26, 30 GB RAM tilgjengelig.

4. Resultater

Dette kapittelet tar for seg resultatene som ble innhentet i denne studien. Resultatkapittelet er to-delt og den første delen tar for seg resultatene som har med idrettsutøverne å gjøre. Del to dreier seg om resultatene fra modellene som ble trent og testet på EKG fra den generelle befolkning.

4.1 Tolkning av EKG fra idrettsutøvere

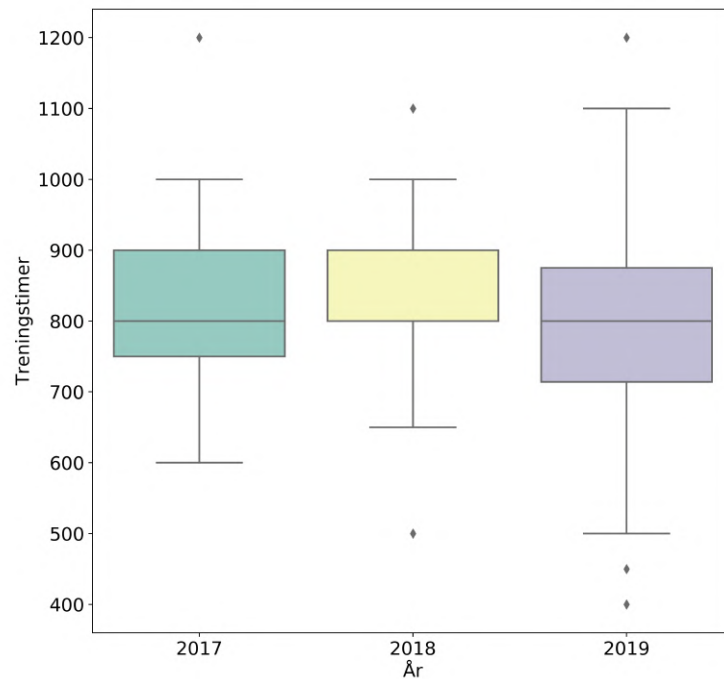
Figur 4.1 viser et grafisk sammendrag av registreringene og tolkningene av EKGene fra idrettsutøverne, som deltok i denne studien.



Figur 4.1: Et grafisk sammendrag av registreringene og tolkningene av EKGene fra idrettsutøverne, som deltok i denne studien. EKGene til 28 idrettsutøvere ble analysert av en kardiolog, og av tolkningsalgoritmen til en moderne elektrokardiograf. Tolkningsalgoritmen tolket 46% av alle EKGene som normale, 25% som i grenseområdet og 29% som unormale, hvorav fire av disse ble diagnostisert som hjerteinfarkt med ST-hevning (STEMI). STEMI er en kritisk diagnose som krever videre utredning. Kardiologen fant ingen kritiske tegn da han tolket EKGene etter idrettskardiologiske retningslinjer. To av funnene ble kategorisert som i grenseområdet av kardiologen, men dette endret ingenting med tanke på videre utredning.

4.1.1 Deltakerne i studien

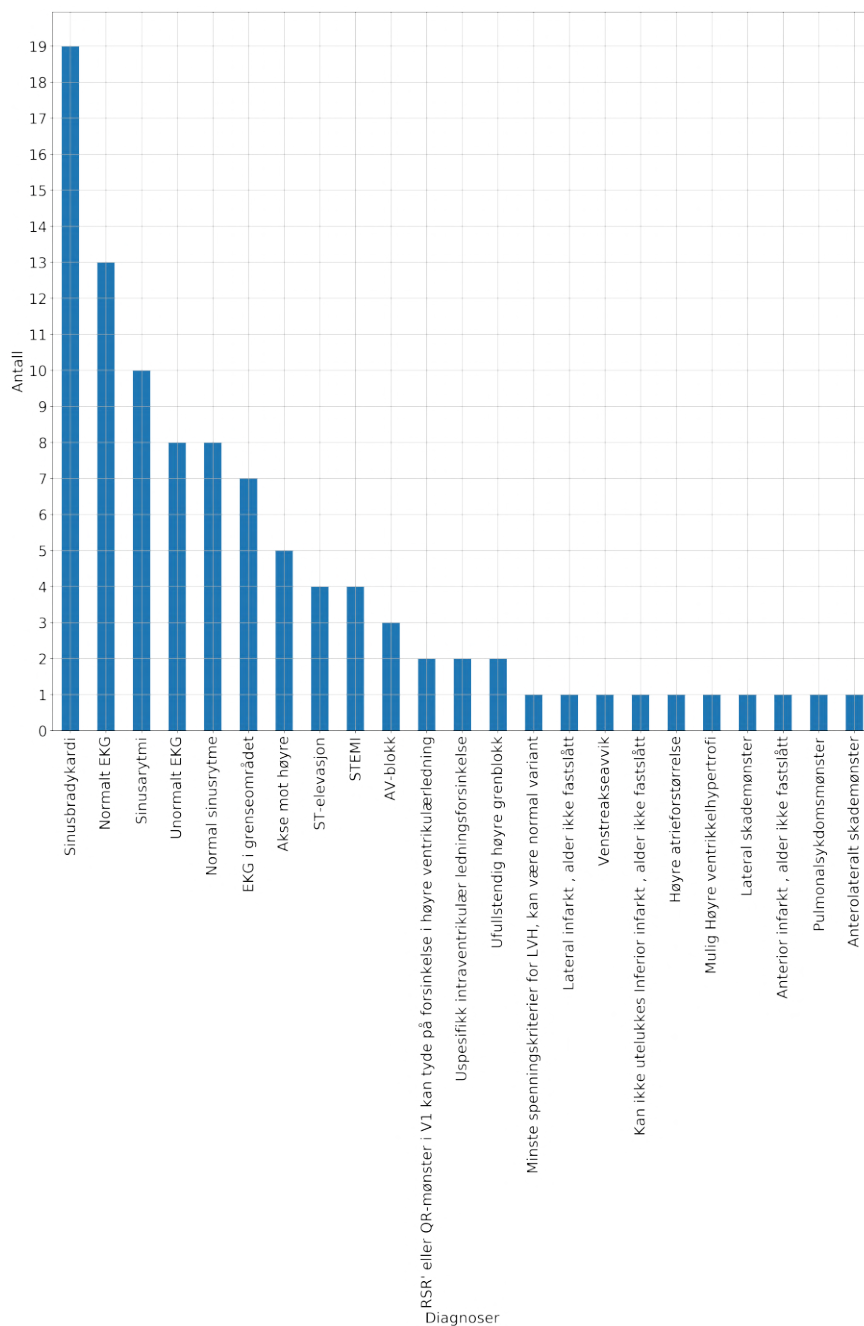
Alle deltakerne som ønsket å delta i denne studien innfridde inkluderingskriteriene. Deltakerne representerte tre ulike idretter; Roing (24), sykkel (2) og padling (2). Av de 28 deltakerne var 19 menn og 9 kvinner. Alderen blant deltakerne var 26.2 ± 4.8 (median = 25). Alle deltakerne oppga antall treningstimer de trente i 2017, 2018 og 2019. Spredningen i treningstimer blant idrettsutøverne, for hvert av de tre årene, er vist i figur 4.2.



Figur 4.2: Boks-diagrammet fremstiller antall treningstimer i 2017, 2018 og 2019 for de 28 idrettsutøverne som deltok i denne studien.

4.1.2 GE Marquette SL12s tolkning

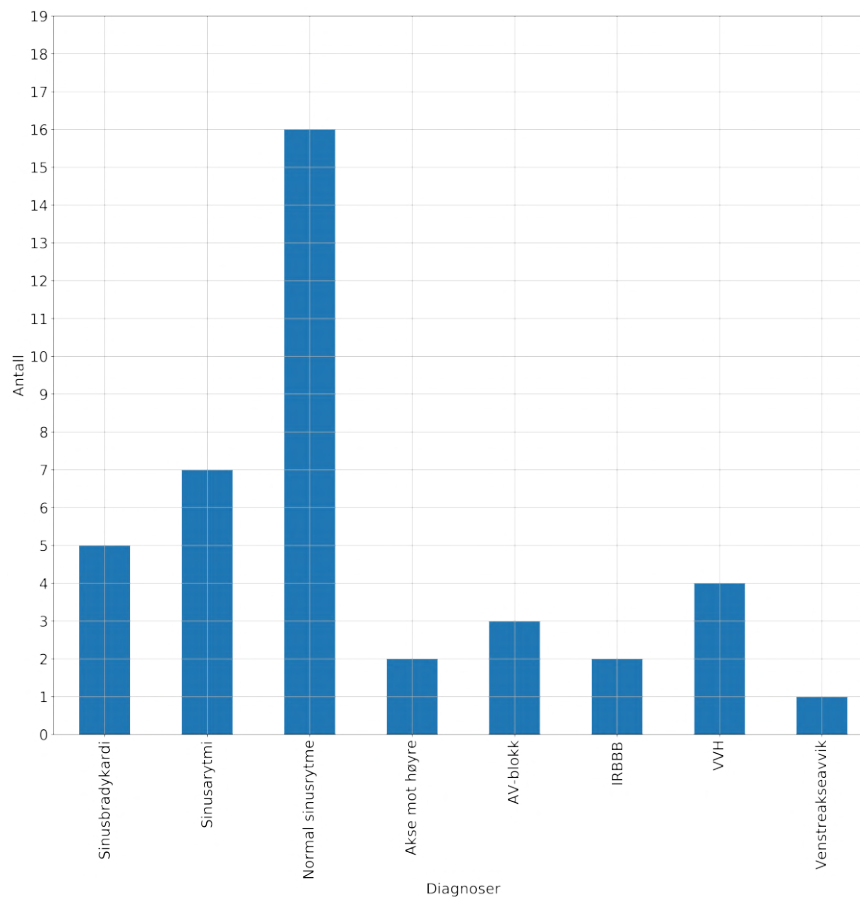
Tolkningsalgoritmen i GE MAC VU360, MarquetteTM SL12, ga totalt 23 forskjellige tolkningstekster for de 28 utøverne. Flere av de 23 tolkningstekstene ble gitt til flere av utøverne slik figur 4.3 viser.



Figur 4.3: Hyppigheten av alle de forskjellige tolkningene som ble gitt av GE MarquetteTM SL12 algoritmen fra samtlige av idrettsutøvernes EKG.

4.1.3 Kardiologens tolkning

De 28 EKGene fra idrettsutøverne ble analysert av en kardiolog med kunnskap innen idrettskardiologi. Kardiologen tok et repetisjonskurs i de siste tolkningskriteriene for EKG tolkning av idrettshjarter [15], før han utførte tolkningen. På samme måte som MarquetteTM SL12, ga kardiologens tolkning av og til flere diagnoser til samme person. Figur 4.4 viser hyppigheten alle unike tolkninger som ble gitt av kardiologen.



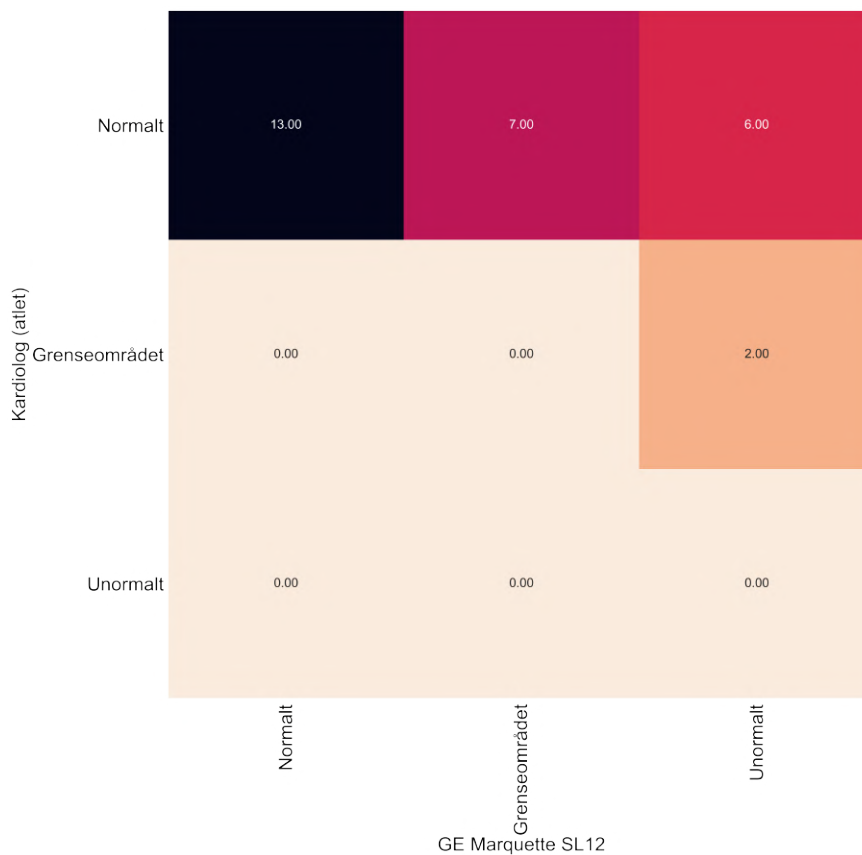
Figur 4.4: Hyppigheten av alle forskjellige diagnoser gitt av kardiologen etter å ha tolket samtlige av de 28 idrettsutøvernes EKG.

4.1.4 Sammenligning mellom Marquette SL12 og kardiologens tolkning

Samtlige EKG ble tolket og klassifisert av både kardiologen og MarquetteTM SL12 (se forvirringsmatrise i appendiks D). I tillegg ble EKGene kategorisert inn i kategoriene normalt, i grenseområdet eller unormalt av både MarquetteTM SL12 og av kardiologen. Kardiologen tolket EKGene både fra et idrettskardiologisk perspektiv og fra et ikke-idrettskardiologisk perspektiv.

Kategorisering etter idrettskardiologiske retningslinjer

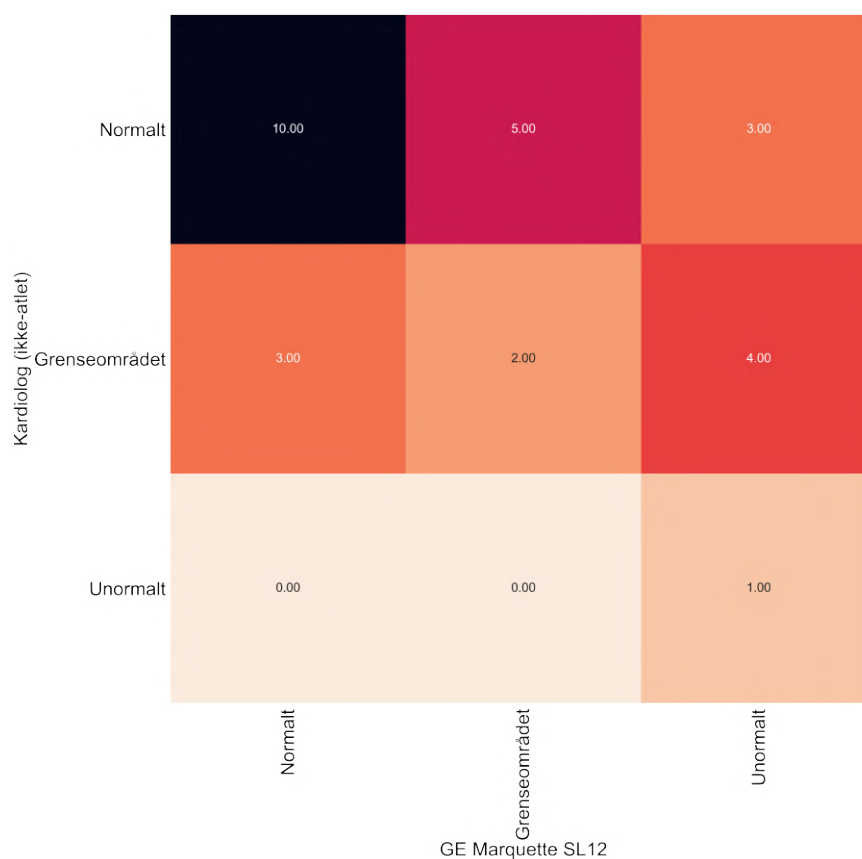
I figur 4.5 er MarquetteTM SL12s tolkninger sammenlignet med kardiologens tolkninger fra et idrettskardiologisk perspektiv. MarquetteTM SL12s tolkning, når kardiologens tolkning regnes som korrekt, gir $F_1 = 0,22$, $F_2 = 0,18$ og $G_2 = 0,11$. Utregningene av skårene er vist i appendiks F.



Figur 4.5: Forvirringsmatrise med kardiologens kategorisering og MarquetteTM SL12s kategorisering. Kardiologen har kategorisert EKGene etter idrettskardiologiske retningslinjer.

Kategorisering etter ikke-idrettskardiologiske retningslinjer

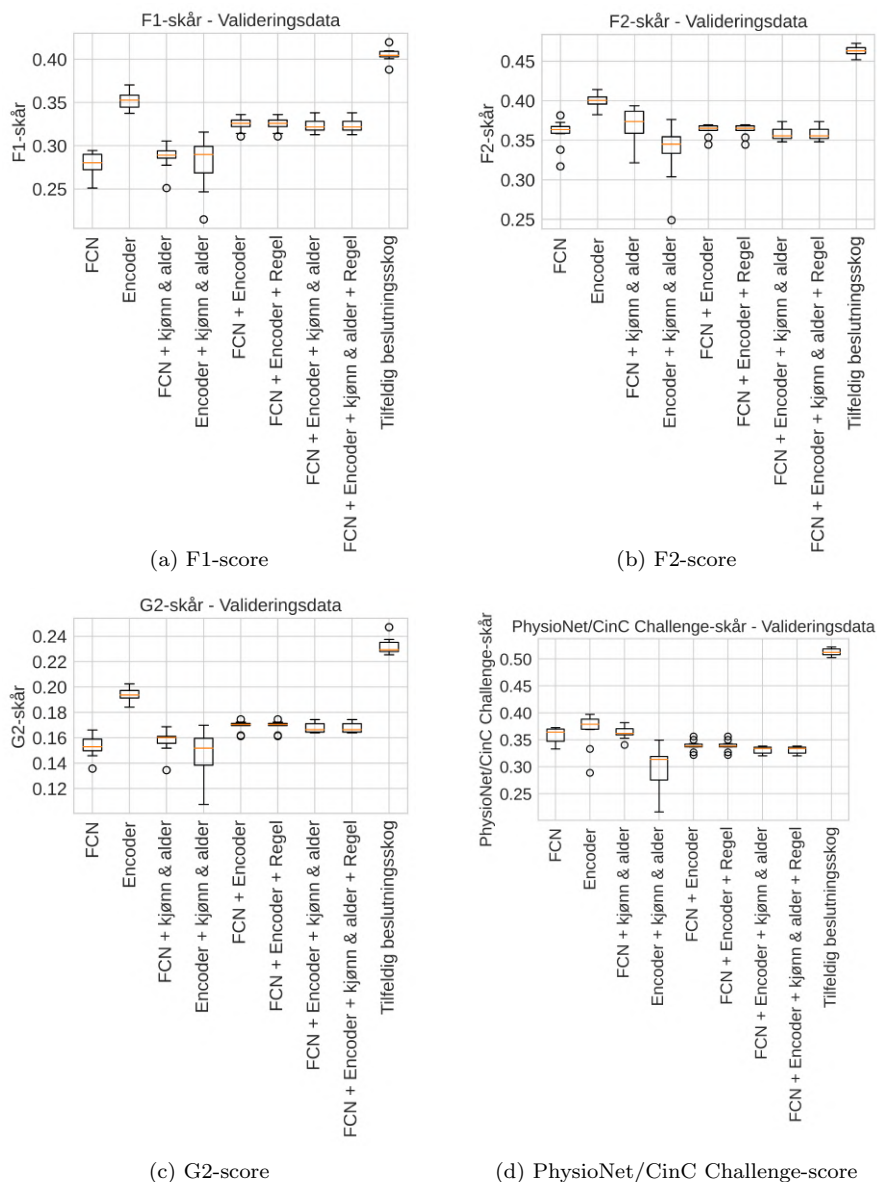
I figur 4.6 er MarquetteTM SL12s tolkninger sammenlignet med kardiologens tolkninger fra et ikke-idrettskardiologisk perspektiv. MarquetteTM SL12s tolkning, når kardiologens tolkning regnes som korrekt, gir $F_1 = 0,38$, $F_2 = 0,42$ og $G_2 = 0,22$. Utregningene av skårene er vist i appendiks F.



Figur 4.6: Forvirringsmatrise med sammenligning kardiologens kategorisering og Marquettes kategorisering. Kardiologen kategoriserte EKGene som om de ikke tilhørte idrettsutøvere

4.2 Resultater fra maskinl ring p  EKG fra den generelle befolkning

Figur 4.7 viser 10-ganger kryssvaliderte resultater p  valideringsdataene i utviklingssettet. De kryssvaliderte resultatene p  treningsdataene kan ses i appendiks C.

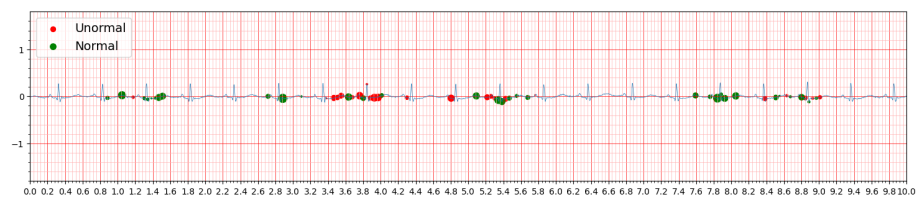


Figur 4.7: Boks-diagrammene viser sk rene oppn dd av ni modeller ved bruk av 10-ganger kryssvalidering og prediksjon p  valideringsdataene. (a) viser F1-poengsum, (b) viser F2-poengsum, (c) viser G2-poengsum og (d) viser PhysioNet / CinC Challenge sk r.

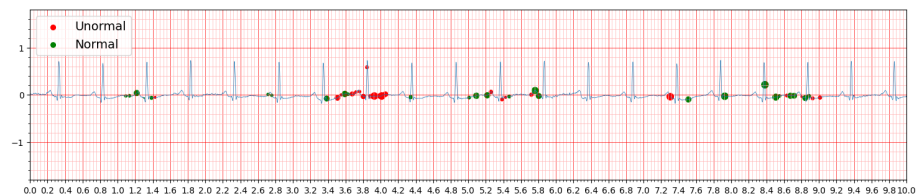
4.2.1 Forklarbar KI for *Encoder-modellen*

Sinustakykardi

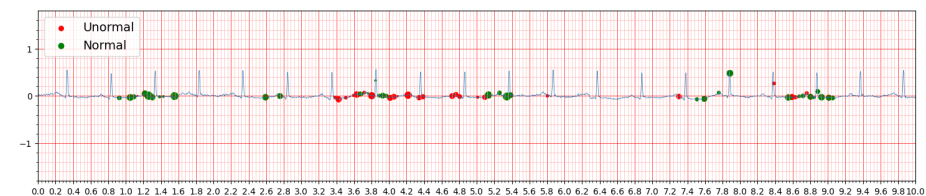
LIME ble brukt til å forklare klassifiseringen av et EKG med sinustakykardi. EKGGet ble først klassifisert av *Encoder-modellen* som klassifiserte sinustakykardi med en sannsynlighet på 0,76. 500 av de viktigste egenskapene for prediksjonen, fordelt utover de 12-avledningene i EKGGet, ble ekstrahert. Fire av de 12 avledningene kan ses i figur 4.8.



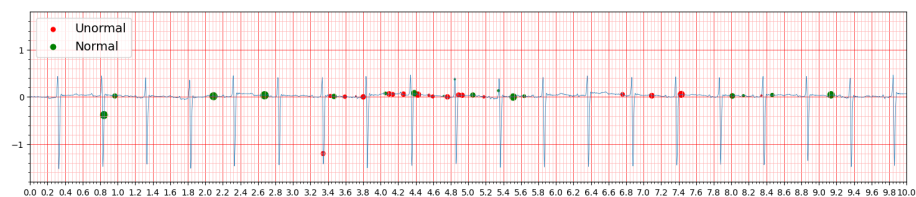
(a) Avledning I



(b) Avledning II



(c) Avledning III

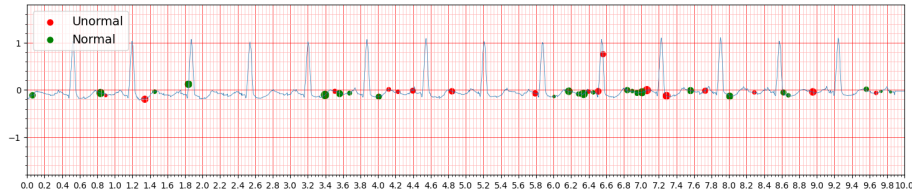


(d) Avledning V2

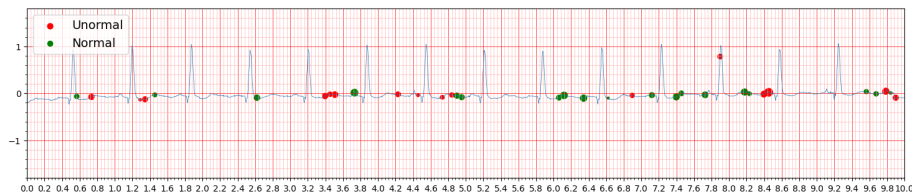
Figur 4.8: Forklarbarhet for *Encoder-modellen*. EKG-opptaket er hentet fra en pasient diagnostisert med sinustakykardi. Forklaringsmodellen ser på de 500 viktigste egenskapene for prediksjonen, fordelt på alle 12 EKG-avledningene. Viktigheten av parameterne er rangert ved størrelsen på prikkene. Store prikker betyr større viktighet enn små prikker.

T-bølgeinversjon

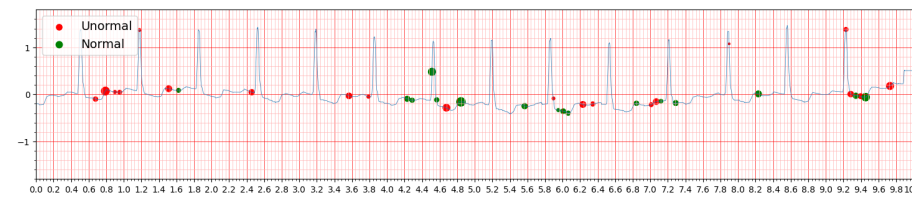
LIME ble brukt til å forklare klassiferingen av et EKG med T-bølgeinversjon. EKGet ble først klassifisert av *Encoder-modellen*, som ga en sannsynlighet på 0,02 for T-bølgeinversjon. 500 av de viktigste egenskapene for prediksjonen, fordelt utover de 12-avledningene i EKGet, ble ekstrahert. Fire av de tolv avledningene kan ses i figur 4.9.



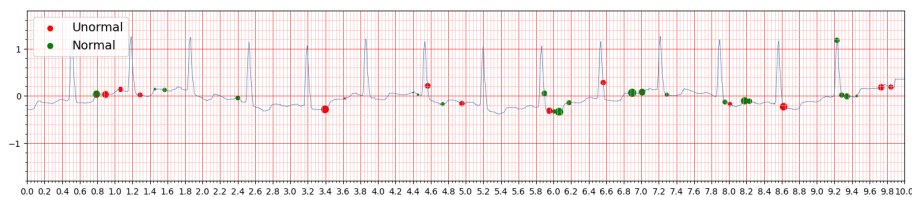
(a) Avledning I



(b) Avledning aVL



(c) Avledning V5



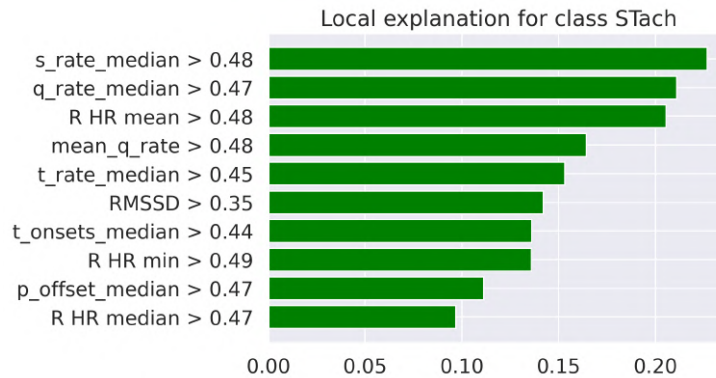
(d) Avledning V6

Figur 4.9: Forklarbarhet for *Encoder-modellen*. EKG-opptaket er hentet fra en pasient diagnostisert med T-bølgeinversjon. Forklaringsmodellen ser på de 500 viktigste egenskapene for prediksjonen fordelt på alle 12 EKG-avledningene. Viktigheten av parameterne er rangert ved størrelsen på prikkene. Store prikker betyr større viktighet enn små prikker.

4.2.2 Forklarbar KI for *Ensemble-modellen*

Sinustakykardi

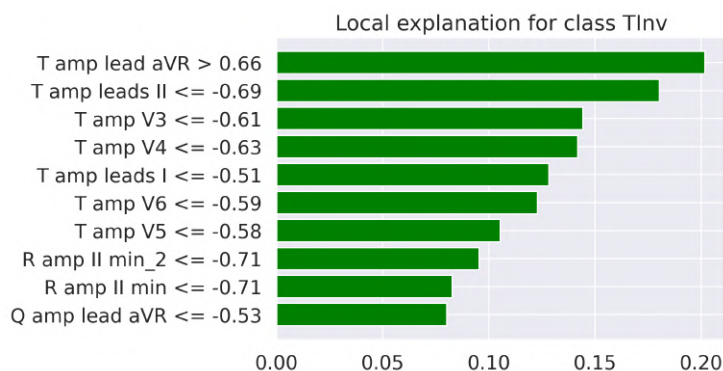
LIME ble brukt til å forklare klassifiseringen av et EKG med sinustakykardi. EKGGet ble først klassifisert av *Ensemble-modellen*, som klassifiserte EKGGet korrekt med sinustakykardi. Av totalt 112 egenskaper i datasettet er de 10 viktigste egenskapene vist i figur 4.10.



Figur 4.10: Lokale forklaringer ved bruk av de 10 mest relevante egenskapene fra et EKG diagnostisert med sinustakykardi (STach). EKG ble riktig klassifisert av *Ensemble-modellen*. De grønne stolpene indikerer bidraget til en positiv klassifisering av sinus takykardi. Røde stolper i motsatt retning ville blitt vist dersom noen av egenskapene bidro mot en negativ prediksjon av sinus takykardi. I etikettene på den vertikale aksene er HR en forkortelse for hjerterate (hjerterefrekvens).

T-bølgeinversjon

LIME ble brukt til å forklare klassifiseringen av et EKG med T-bølgeinversjon. EKGGet ble først klassifisert av *Ensemble-modellen*, som klassifiserte EKGGet korrekt med T-bølgeinversjon. Av totalt 112 egenskaper i datasettet er de 10 viktigste egenskapene vist i figur 4.11.



Figur 4.11: Lokale forklaringer ved bruk av de 10 mest relevante egenskapene fra et EKG diagnostisert med T-bølgeinversjon (TInv). EKG ble riktig klassifisert av *Ensemble-modellen*. De grønne stolpene indikerer bidraget til en positiv klassifisering av T-bølgeinversjon. Røde stolper i motsatt retning ville blitt vist dersom noen av egenskapene bidro mot en negativ prediksjon av T-bølgeinversjon. I etikettene på den vertikale aksene er amp en forkortelse for amplitude.

4.3 Tilgjengelig kode

All kode som er nødvendig for å gjenskape resultatene i denne studien er gjort tilgjengelig i et GitHub-repository¹. Koden er tilgjengeliggjort som åpen kildekode, lisensiert under Apache License 2.0 [140]. I tillegg er koden til Python-pakken; ECG-featurizer, offentliggjort i et eget åpent GitHub-repository². Pakken er tilgjengeliggjort som åpen kildekode, lisensiert under GNU General Public License v3.0 [141].

¹GitHub-repository til koden utviklet i denne studien: <https://github.com/Bsingstad/IdrettsEKG>

²GitHub-repository for ECG-featurizer: <https://github.com/ECG-featurizer/ECG-featurizer>

5. Diskusjon

I forrige kapittel ble funnene i denne studien presentert. I dette kapittelet vil resultatene drøftes, tolkes og sammenlignes med andre studier. Først vil resultatet fra tolkningen av EKGene fra idrettsutøverne diskuteres. Deretter vil resultatet fra maskinlæringsmodellene, og implementeringen av forklarbar KI på to av maskinlæringsmodellene diskuteres. Til slutt vil det vurderes hvordan maskinlæring kan brukes på EKG fra idrettsutøvere.

5.1 Tolkning av EKG fra idrettsutøvere

5.1.1 Sammenligning av tolkning fra Marquette SL12 og kardiologen

Tolkningene fra MarquetteTM SL12 kategoriserte EKGene fra idrettsutøverne inn i 13 (46%) normale, 7 (25%) i grenseområdet og 8 (29%) unormale. Tolkningene fra kardiologen ga derimot 18 (64%) normale, 9 (32%) i grenseområdet og 1 (4%) unormale ved ikke-idrettskardiologiske retningslinjer, og 26 (93%) normale, 2 (7%) i grenseområdet og 0 (0%) unormale ved idrettskardiologiske retningslinjer.

I figur 4.5 er MarquetteTM SL12s tolkning sammenlignet med kardiologens tolkninger etter idrettskardiologiske retningslinjer. Denne sammenligningen gir MarquetteTM SL12 skårene; $F_1 = 0,22$, $F_2 = 0,18$ og $G_2 = 0,11$, når kardiologens tolkning brukes som fasit.

I figur 4.6 er MarquetteTM SL12s tolkning sammenlignet med kardiologens tolkninger etter ikke-idrettskardiologiske retningslinjer. Denne sammenligningen gir MarquetteTM SL12 skårene; $F_1 = 0,38$, $F_2 = 0,42$ og $G_2 = 0,22$, når kardiologens tolkning brukes som fasit.

Sammenligningen mellom tolkningen fra MarquetteTM SL12 og de to forskjellige tolkningene av kardiologen, basert på F_1 , F_2 og G_2 -skår, indikerer at GE MarquetteTM SL12 er mer tilpasset til tolkninger rettet mot vanlige retningslinjer enn til idrettskardiologiske tolkninger. På den annen side viser forvirringsmatrisene i figur 4.5 og figur 4.6 at like mange av idrettsutøverne (13 av 28) ble korrekt klassifisert av MarquetteTM SL12 uavhengig av om kardiologen klassifiserte idrettsutøverne etter idrettskardiologiske retningslinjer eller vanlige retningslinjer. En større populasjon med flere sanne tilfeller i de forskjellige kategoriene trengs for å kunne si noe om ytelsen til MarquetteTM SL12s kategorisering av idrettsutøvere.

En begrensning ved å sammenligne en kardiologs tolkning med tolkningen fra MarquetteTM SL12, er at kardiologens tolkninger var mer nyansert enn

MarquetteTM SL12. For å kunne sammenligne de to tolkningene i en forvirringsmatrise, som i figur 4.5, 4.6, måtte de nyanserte tolkningene endres til et format som var sammenlignbart med MarquetteTM SL12 sine tolkninger.

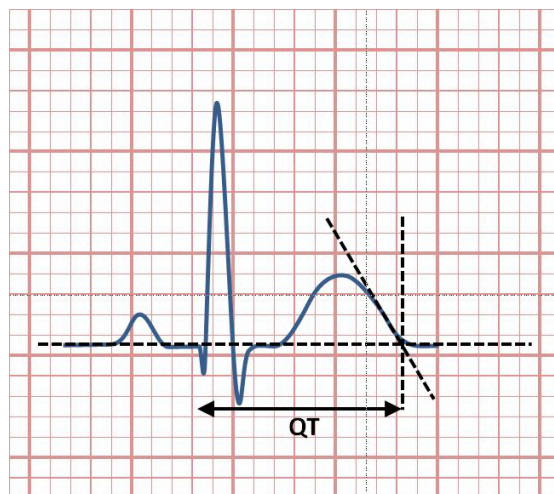
5.1.2 Beregning av intervaller og segmenter

Varigheten på noen av segmentene og intervallene i EKGene ble beregnet forskjellig av MarquetteTM SL12 og kardiologens visuelle måling. Kardiologen fant for eksempel at det var en systematisk lengre QRS-varighet i beregningene av MarquetteTM SL12 algoritmen enn ved sin egen visuelle måling. En mulig grunn til denne systematiske forskjellen er at MarquetteTM SL12 algoritmen beregner QRS-varigheten ved å ta tiden fra første tegn til start på en Q-bølge, i en hvilken som helst avledning, til den seneste avslutningen av S-bølgen, i hvilken som helst avledning [123]. Det er derfor sannsynlig at den algoritmemålte QRS-varigheten blir lengere enn kardiologens visuelle måling, som leser av intervallet i en avledning. I en studie som sammenlignet en algoritme med en kardiolog viste svakest korrelasjon mellom visuelt målt og algoritmemålt QRS-varighet av alle de algoritmemålte intervallene som studien så på [20].

I denne masteroppgaven ble to av personene klassifisert med uspesifikk intraventrikulær ledningsforsinkelse av MarquetteTM SL12 (figur 4.3). Dette er en diagnose som blir stilt basert på QRS-tiden ($QRS > 110ms$ visuelt målt [142]). Kardiologen var uenig med MarquetteTM SL12, og begrunnet feilklassifiseringen med overestimering av QRS-tiden. På den annen side kan det se ut som at dette er justert for i dokumentasjonen av MarquetteTM SL12 der uspesifikk intraventrikulær ledningsforsinkelse er definert ved $118ms < QRS < 124ms$ [123].

QT/QTc-tiden er et annet parameter der MarquetteTM SL12 algoritmen systematisk beregnet tiden lengre (5-10%) enn kardiologen. Denne forskjellen beskrives også av Drezner J.A et al. 2017 [15], der nøyaktigheten til algoritmemålte QTc-verdier var omtrent 90–95% i forhold til kardiologen. I denne oppgaven beregnet Kardiologen QT-tiden fra starten av en q-takk til krysningspunktet mellom grunnlinjen i EKG et og tangenten til bratteste del av T-bølgen. Denne QT-målemetoden er vist i figur 5.1. MarquetteTM SL12 algoritmen beregner QT-intervallet fra første tegn til depolarisering i hvilken som helst avledning, til den siste påvisning av repolarisering i hvilken som helst avledning [123]. Deretter brukes en QTc-korreksjonsalgoritme, MarquetteTM SL12 algoritmen brukte i denne studien Friderica-metoden for å beregne QTc-tiden (appendiks A). Til tross for den algoritme-beregnete QTc-tiden er systematisk var lengere enn den kardiolog-målte QT-tiden er kriteriene for forlenget QT-intervall det samme for kardiologen og algoritmen, $QTc > 480ms$ [123]. Selv om det ikke var noen tilfeller av forlenget QT-intervall i denne oppgaven er det mulig at dette kan føre til overdiagnostikk i andre populasjoner.

Et annet viktig parameter, som også ser ut til å feilvurderes av MarquetteTM SL12 algoritmen, er ST-segmentet. Et høyt J-punkt og tidlig repolarisering, som er et svært vanlig funn hos idrettsutøvere [76] kan forveksles med STEMI [144]. Derimot vil et oppadgående konkavt ST-segment signalisere en normal variant [144]. En studie, som sammenlignet en annen EKG-tolkningsalgoritme med en kardiologs vurdering, viste lav korrelasjon mellom de to tolkningene av tidlig repolarisering [20], og en annen studie viste at tidlig repolarisering var blant de viktigste årsakene til feilklassifisering av STEMI [145]. Dette gjør at feiltolk-



Figur 5.1: Visuell måling av QT-tid. Figuren er hentet fra [143], av PeaBrainC, 2018. CC BY-SA 4.0 [35].

ning av tidlig repolarisering kan være en plausibel forklaring på de tilfellene der MarquetteTM SL12 ga diagnosen STEMI i denne oppgaven. På den annen side har to studier vist at MarquetteTM SL12 har høy spesifisitet for STEMI med 98% og 100% [146], [147]. I denne studien er imidlertid 4 av de 28 deltakerne uriktig diagnostisert med STEMI av MarquetteTM SL12 (se figur 4.3). Dette gir en spesifisitet på 86%. En forklaring på forskjellen mellom denne studien og de tidligere rapporterte resultatene kan være at deres populasjoner var fra legevakten. Den ene studien rapporterte også om en gjennomsnittsalder på 64år [146], som står i kontrast til populasjonen i denne masteroppgaven hvor gjennomsnittsalderen var på 26år og alle var antatt friske. Til tross for at kardiologer er kjent med faren for feilklassifisering av STEMI [12] kan potensielt en slik feilklassifisering adapteres av mindre trent personell og skape unødvendig stress for pasienten. Schlapfer et al. 2017 foreslår blant annet å implementere kjønns og aldersspesifikke tolkningskriterier for STEMI [12]. Dette er allerede implementert i MarquetteTM SL12, men med lav prioritet ifølge dokumentasjonen [123]. Basert på funnene i denne studien foreslår jeg å øke prioriteten av alder, og implementere en kvantifisering av fysisk form i algoritmene for å øke spesifisiteten for STEMI ovenfor idrettsutøvere.

5.1.3 Feilplassering av elektroder

Ett av de 28 EKGene hadde en eller flere feilplasserte prekordiale elektroder. Kardiologen identifiserte denne feilplasseringen, som ble gjort til tross for at apparatet hadde en innebygget plasseringsassistent. EKGGet, som ble tatt i tilfellet med feilplasserte elektroder, ga diagnoser som; STEMI, anterior infarkt og lateralt skademønster. Kardiologen mente at en plausibel forklaring på disse diagnosene kunne være feilplasseringen. Drezner et al. 2017 diskuterer også at feilplassering av prekordiale avledninger kan simulere myokardiel skade [15].

5.2 Maskinl ring p  EKG fra den generelle befolkning

5.2.1 Sammenligning av maskinl ringsmodellene

Denne masteroppgaven viser et eksempel p  hvordan CNN-modeller og en ensemble-modell kan brukes til   klassifisere 27 forskjellige diagnoser basert p  12-lede EKG fra 43 101 pasienter. Kryssvaliderte resultater viser at *Ensemble-modellen* presterte bedre enn CNN-modellene p  utviklingsdataene. Modellene ble evaluert ved bruk av F_1 , F_2 , G_2 og PhysioNet/CinC Challenge sk r. *Ensemble-modellen* presterte bedre p  alle fire sk ringsalgoritmer.

Vinneren av PhysioNet/CinC Challenge 2020 rapporterte en kryssvalidert PhysioNet/CinC Challenge-sk r p  $0,533 \pm 0,046$ p  utviklingsdataene [148]. Dette er bare $0,021$ bedre enn den beste poengsummen oppn dd i denne masteroppgaven: $0,512 \pm 0,006$. Likevel er det viktig   huske at PhysioNet/CinC Challenge 2020 ble avgjort basert p  modellens ytelse p  et skjult testsett. De kryssvaliderte sk rene som er oppn dd p  utviklingssettet b r sammenlignes med forsiktighet med de rapporterte sk rene p  PhysioNet/CinC Challenge 2020's skjulte testsett fra andre studier [124], [148]–[179]. Noen f  av studiene viste godt samsvar mellom kryssvaliderte resultater p  utviklingssettet og resultatene oppn dd p  det skjulte testsettet [148], [178], men de fleste rapporterte om betydelig reduksjon i sk r fra validering p  utviklingssettet og resultatene oppn dd p  det skjulte testsettet [149]–[165].

Overraskende nok presterte *Encoder-modellen* best sammenlignet med resten av CNN-modellene. Dette var overraskende fordi *Encoder-modellen* var en av modellene med lavest kompleksitet. *Encoder-modellen* var betydelig bedre m lt p  F_1 , F_2 og G_2 -sk r (figur 4.7 a og c), men tett fulgt av *FCN* og *FCN || Kj nn & Alder*, n r man ser p  PhysioNet/CinC Challenge sk r (figur 4.7 d). I Singstad & Tronstad 2020 viste vi derimot at *Encoder-modellen* presterte d rligere sammenlignet med flere av de nevnte modellene p  et delsett av det skjulte testsettet [124]. Dette understreker at man b r v re forsiktig n r man sammenligner kryssvaliderte sk rer med sk rer oppn dd p  det skjulte testsettet.

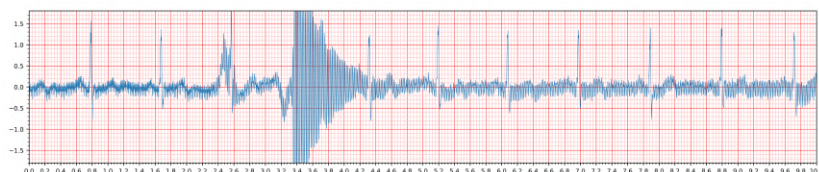
Ingen signifikante forskjeller ble funnet mellom *FCN || Encoder* og *FCN || Encoder || kj nn & alder*, n r den *regelbaserte algoritmen* ble lagt til (figur 4.7). En mulig forklaring p  dette kan v re at den *regelbaserte algoritmen* alltid resulterte i lik klassifisering som CNN-modellene, og dermed forble den endelige klassifiseringen uendret. En annen mulig forklaring er at den *regelbaserte algoritmen* ikke klarte   analysere EKGene, og at dette resulterte i at den *regelbaserte algoritmen* ikke klarte   klassifisere noen av EKGene. Et viktig poeng   nevne her er at den *regelbaserte algoritmen* i denne studien var sv rt enkel. P  bakgrunn av dette kan det ikke konkluderes med hvorvidt regelbaserte algoritmer kan brukes til   forbedre konvolusjonelle nevralt nettverk i klassifiseringen av EKG eller ikke. I et fremtidig arbeid vil det sannsynligvis gi et bedre resultat hvis den endelige beslutningen (fra en CNN-modell og en regelbasert algoritme) er basert p  en vektning mellom de to klassifiseringsresultatene.

En mulig grunn til at *Ensemble-modellen* presterte bedre enn CNN-modellene kan v re at datamengden var for liten for de komplekse CNN-modellene. Studier viser at dype nevralt nettverk ofte trenger store datamengder for   kunne overg 

enkler maskinlæringsmodeller [114].

5.2.2 Begrensninger og mulige feilkilder

Noen av EKGene inneholdt mye støy, som for eksempel EKGet vist i figur 5.2. En mulig feilkilde i denne delen av oppgaven er at EKGene ikke ble filtrert før de ble matet inn i modellen, eller før egenskaper ble ekstrahert av ECG-featurizer [130]. Det ble forsøkt å implementere filtrering, ved bruk av NeuroKit2 [131], i batch-generatoren, men begrensninger i form av regnekraft satte en stopper for dette. Ytterligere studier er nødvendige for å avgjøre om filtrering av EKG-signalene vil forbedre ytelsen til modellene som ble brukt i denne studien.



Figur 5.2: EKGet i dette eksempelet fra utviklingsdatasettet er kontaminert av støy. Signalet som er vist i dette eksempelet er fra avledning *II*. Figuren er laget ved bruk av `ecg plot` [54].

En feilkilde som kan ha påvirket de CNN-baserte modellene er at samtlige signaler ble klippet ned til 5000 samples. Dette kan ha forårsaket tap av informasjon. Noen få av EKGene var omtrent 30 minutter lange, og sannsynligheten for at viktig informasjon ble fjernet fra disse er stor. Spesielt er EKG med VES sårbart [25], da dette diagnostiseres på bakgrunn av enkelte hjerteslag, som kan opptre tilfeldig i EKGet.

5.3 Forklarbar KI på EKG

Klassene; Sinustakykardi og T-bølgeinversjon ble valgt ut til videre analyser med forklarbar KI, fordi disse diagnosene representerer to forskjellige måter å analysere EKGet på. Sinustakykardi er definert ved sinusrytme over 100 slag i minuttet. I dette tilfelle må derfor rytmen analyseres. Den enkleste måten er å se på tidsintervallene mellom R-takkene. T-bølgeinversjon representerer en mer morfologisk analyse av EKGet, da T-bølgeinversjon bestemmes av formen til T-bølgen i forskjellige avledninger. Den første hypotesen er at *Ensemble-modellen* skal gjøre det bedre enn CNN-modellen på rytme-klassifiseringer fordi flere av de ekstraherte EKG-egenskapene, som ble brukt av *Ensemble-modellen*, er basert på intervaller mellom forskjellige bølgetopper. Den andre hypotesen er at de CNN-baserte modellene skulle prestere bedre på de morfologiske klassifiseringene, fordi de kan bruke alle samples fra EKGet som beslutningsgrunnlag. Dessuten har CNN-modeller vist seg å være gode på å detektere former og mønstre i bilder og tidsserier.

To EKG, som korrekt ble klassifisert av *Ensemble-modellen* med henholdsvis sinustakykardi og T-bølgeinversjon, ble plukket ut for at de skulle bli forklart med LIME. Den lokale forklaringen fra LIME viste at de tre parameterne

med størst betydning for klassifiseringen av sinustakykardi var medianen av S-intervallene, medianen av Q-intervallene og gjennomsnittet av hjertefrekvensen. Dette gir fysiologisk sett mening, siden sinustakykardi er definert ved hjertefrekvens ≥ 100 . Ved forklaring av T-bølgeinversjon var det heller ikke uventet at de forskjellige parameterne relatert til T-bølgen var rangert som de viktigste. Den viktigste parameteren viste seg å være amplituden til T-bølgen i aVR-avledningen etterfulgt av amplituden til T-bølgen i II og V3. Dette stemmer med de diagnostiske kriteriene, der polariteten til T-bølgen benyttes for å diagnostisere T-bølgeinversjon. I tillegg er det verdt å merke seg at forklaringsmodellen vektlegger at T-bølgen er positiv i aVR. T-bølgen i et normalt EKG er vanligvis negativ i aVR, men i tilfeller av T-bølgeinversjon er T-bølgen invertert til positiv.

Encoder-modellen er valgt til å representere CNN-modellene, fordi den presterte best på de kryssvaliderte resultatene. Likt som for *Ensemble-modellen*, ble to EKGer med sinustakykardi og T-bølgeinversjon valgt ut for videre analyser med forklarbar KI. Det var forventet at forklaringen av EKGet klassifisert med sinustakykardi (Figur 4.8) skulle fremheve punkter i EKGet, som periodisk gjentar seg. Avledning V2, i figur 4.8, ser ut til å fremheve noen av de rytmiske egenskapene i EKGet, men helhetsinntrykket er at dette ikke er en spesifikk nok forklaring. For T-bølgeinversjon fremhever den lokale forklaringen fra LIME i stor grad T-bølgene. I Figur 4.9c, er den siste T-bølgen i avledning V5 markert med både røde og grønne prikker. Dette betyr at dette området på EKGet bidro både til en positiv og en negativ prediksjon av klassifiseringen T-bølgeinversjon.

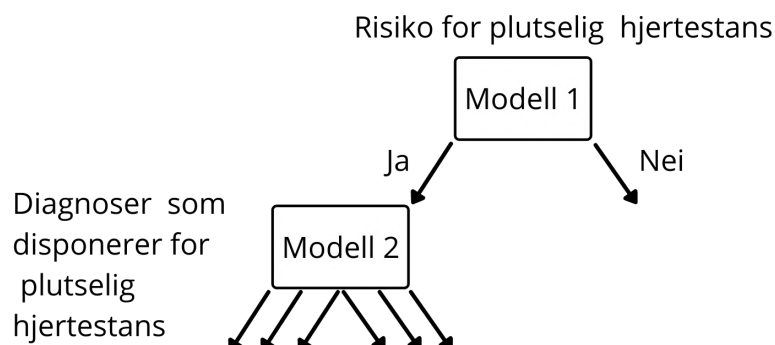
Forklarbar KI er et relativt ungt fagfelt, og det er kun noen få studier som har undersøkt forklarbar KI på EKG-klassifiseringsmodeller [180]–[183]. Så vidt jeg vet er denne masteroppgaven den første studien som tar i bruk LIME-rammeverket til forklaring av maskinlæringsbasert klassifisering av EKG.

5.4 Maskinlæring på EKG fra idrettsutøvere

De 28 EKG, som ble samlet inn i denne masteroppgaven, er for få til å brukes til trening av en maskinlæringsmodell. Dessuten representerer EKGene, i følge kardiologens tolkning, kun friske samt to funn i grenseområdet. En mulighet for å kunne bruke EKGene er å kombinere de med andre datasett. EKGene som ble samlet inn i denne studien vil bli publisert på Physionet.org, som er en åpen database. Foreløpig ser det ut til at EKGene samlet inn i denne studien vil utgjøre den første åpne databasen bestående av EKGer fra idrettsutøvere.

Gitt at man har klarer å skaffe nok relevante EKGer til å trene en maskinlæringsmodell for å klassifisere idrettsutøvere med risiko for plutselig hjertestans, foreslår jeg en klassifisering i to trinn (figur 5.3). *Modell 1* bør klassifisere EKGet med to utfall: positiv eller negativ prediksjon for risiko for plutselig hjertestans. Ved positiv prediksjon bør en ny modell (*Modell 2*) klassifisere EKGet innenfor de ulike diagnosene som er kjent for å kunne disponere for plutselig hjertestans. Fordelen med denne fremgangsmåten er at *Modell 1* kan trenes på EKG fra personer som har opplevd plutselig hjertestans (faktiske hendelser). På den annen side vil en positiv prediksjon være knyttet til mange forskjellige diagnoser, og dagens forklaringsmodeller vil sannsynligvis ikke kunne gi en så god forklaring at legen forstår hvilken diagnose det er. Derfor vil sannsynligvis *Modell 1* opptre som en “svart boks”, som igjen betyr at *Modell 1*, alene, vil være klinisk

irrelevant. *Modell2* vil klassifisere EKGene innenfor diagnosene som er kjent fra litteraturen til å kunne disponere for plutselig hjertestans. Dette gir legen en idé om hva som skal være videre utredning, i tillegg til at legen lettere vil kunne forstå forklaringen fra en forklaringsmodell, når den kun er relatert til én diagnose om gangen.



Figur 5.3: En foreslått modellarkitektur for fremtidig arbeid

En mulighet for videre arbeid, uten tilgang til mer data, er å simulere EKGer med friske og ulike patologiske trekk, som kan være vanskelig å skille fra idretts-hjarter. Dette kan gjøres på flere forskjellige måter, enten ved bruk av konvensjonelle simuleringmetoder [184], eller for eksempel ved bruk av generative adversarial network (GAN) [185]. Deretter kan de simulerte EKGene blandes med de innhentede EKGene fra denne masteroppgaven, og deretter se hvorvidt en algoritme kan klare å skille de simulerte patologiske EKGene, fra EKGene fra de friske idrettsutøverne. Enda bedre vil det være dersom det kan suppleres med EKG fra noen syke idrettsutøvere.

For å øke antallet EKGer fra idrettsutøvere kan et internasjonalt samarbeid være løsningen. En dansk studie, Tischer et al.2016 [186], foreslår nettopp dette. Danskene har omtrent 500 EKG fra danske toppidrettsutøvere, og foreslår et nordisk samarbeid for deling av EKG-data fra toppidrettsutøvere.

Det er flere utfordringer ved innsamling av EKGer fra idrettsutøvere i Norge. En av utfordringene ligger i dagens praksis i flere norske sykehus, hvor EKGene skrives ut, deretter skannes inn og legges til i pasientjournalen. De skannede EKGene gjør det mer komplisert å utføre maskinlæring på disse EKGene, sett at de skulle blitt tilgjengelig for forskning. Det vil selvsagt være mulig å bruke bildeklassifiseringsmodeller på slike data, men det vil sannsynligvis kreve veldig høy oppløsning på bildedataene for at det skal kunne måle seg med mulighetene det gir å kunne utføre maskinlæring direkte på rådataene til EKGet. Dessuten kreves det mer lagringsplass og regnekraft for behandle bildedata i motsetning til rådataene fra EKGet.

En annen utfordring er hva som skal brukes som fasit for en modell som skal

klassifisere EKG med risiko for plutselig hjertestans. I mange studier brukes tolkningen fra en gruppe med kardiologer som fasit, men dette åpner for menneskelige feil. Et bedre alternativ vil være å bruke faktiske hendelser. I Norge anslås det at omtrent tre norske idrettsutøvere dør som følge av plutselig hjertestans årlig [33]. Det er selvsagt bra at dette tallet er lavt, men dette gjør også at det å bygge opp en database bestående av EKG'er fra personer som senere opplever plutselig hjertestans, vil ta veldig lang tid. Et slikt alternativ vil derfor bare være aktuelt på internasjonalt nivå.

En annen tilnærming på problemet med manglende data er å bruke transfer learning. Et eksempel kan være å trene en maskinlæringsmodell på EKG'er fra den generelle befolkning, som senere har opplevd plutselig hjertestans. Dette er EKG'er som viser seg å være lettere å skaffe store mengder av. Et eksempel på det er denne danske studien med EKG fra 326 227 personer hvor 2667 opplevde plutselig hjertestans [187]. En modell trent på et slikt datasett, og videre på et mindre datasett med EKG fra idrettsutøvere (transfer learning) kan kalibrere modellen mot å bli bedre på å skille friske idrettshjertener fra idrettshjertener med risiko for plutselig hjertestans.

Den lave prevalensen blant mange av sykdommene som disponerer for plutselig hjertestans er en utfordring for enhver modell. Det bør være en svært høy spesifisitet (≈ 1), i klassifiseringen for å forhindre effekten av falske-positive paradokset, der selv en ganske høy spesifisitet vil gi mange falske positive. Samtidig ønsker man en viss sensitivitet, slik at man faktisk klarer å fange opp noen syke. Dette bør man ta i betraktning når man trener og skårer maskinlæringsmodeller for å avdekke sjeldne patologiske trekk.

Maskinlæringsmodeller har vist at de kan være bedre enn kardiologer til å tolke EKG'er i visse sammenhenger [23], men det gjenstår fortsatt å teste maskinlæringsmodellene opp mot en toppmoderne og klinisk anvendt EKG-tolkningsalgoritme på et datasett med stor variasjon i diagnoser.

6. Konklusjon

Denne oppgaven hadde som mål å belyse problematikken rundt plutselig hjertestans blant idrettsutøvere, og hvordan nye metoder kan forbedre hjerte-diagnostikken av denne gruppen. Hypotesen var at maskinlæring kan brukes på EKG for å identifisere idrettsutøvere med risiko for plutselig hjertestans. I denne studien er det høstet erfaring basert på innsamlede EKG fra idrettsutøvere og maskinlæring på EKG fra den generelle befolkning.

En av hensiktene med denne oppgaven var å undersøke om tolkningsalgoritmene, i en klinisk anvendt elektrokardiograf, er bedre tilpasset EKG fra den generelle befolkning enn idrettsutøvere. For å undersøke dette ble det målt EKG på 28 idrettsutøvere. Disse EKGene ble videre tolket av både elektrokardiografens tolkningsalgoritme, og av en kardiolog med erfaring innen tolkning av idrettshjerter. Ut i fra sammenligningen mellom de algoritme-tolkede og kardiolog-tolkede resultatene ble det identifisert noen svakheter ved tolkningsalgoritmen. Flere av disse svakhetene er også observert i andre studier, som har sett på andre tolkningsalgoritmer [12], [15], [20], [145]. Dette kan tyde på at det eksisterer noen generelle svakheter i dagens EKG-tolkningsalgoritmer. En av tilstandene, som så ut til å bli feiltolket, var tidlig repolarisering. Tidlig repolarisering er et svært vanlig funn blant unge idrettsutøvere med en prevalens opp mot 90% [76]. I følge kardiologen så det ut til at tidlig repolarisering ble tolket som STEMI, og dette støttes også av andre studier [12], [145]. Dette indikerer at noen av de generelle svakhetene ved tolkningsalgoritmene kan bli mer fremtredende for unge idrettsutøvere, men det er fortsatt flere klinisk anvendte tolkningsalgoritmer som ikke er testet enda.

Den andre hensikten med denne oppgaven var å undersøke om moderne metoder, som KI, kan brukes til å forbedre dagens klinisk anvendte tolkningsalgoritmer, slik at de blir bedre kalibrert for idrettsutøvere. I mangel på nok relevante EKG fra idrettsutøvere, ble det tatt i bruk et åpent datasett fra den generelle befolkning. Dette var det mest relevante datasettet som var tilgjengelig i tilstrekkelig størrelse for maskinlæringsformål. På datasettet ble det trent ni ulike maskinlæringsmodeller. 10-ganger kryssvaliderte resultater på datasettet viste at *ensemble-modellen* presterte best av de ni modellene. Denne masteroppgaven viser at maskinlæring kan brukes til å klassifisere forskjellige klinisk relevante diagnoser basert på EKG. Det er fortsatt usikkert hvordan modellene, utviklet i denne oppgaven, vil prestere i forhold til kardiologer, eller tolkningsalgoritmene i klinisk anvendte elektrokardiografer. Dessuten må modellen trenes og valideres på relevante EKG, for å avgjøre om KI kan forbedre dagens kliniske anvendte tolkningsalgoritmer med hensyn til idrettsutøvere.

En av forutsetningene for å kunne bruke avanserte maskinlæringsmodeller og dyplæringsmodeller, i klinisk sammenheng, er at de er forklarbare. Derfor er

det i denne oppgaven undersøkt hvorvidt prediksjonene fra modellene, utviklet i denne masteroppgaven, lar seg forklare ved bruk av LIME. Prediksjonene fra både *Encoder-modellen* og *Ensemble-modellen* lot seg forklare, men det gjenstår fortsatt å validere nyttigheten av forklaringene. Sannsynligvis må et tettere tverrfaglig samarbeid til, for å skape verdi ut av forklaringsmodellene.

Arbeidet presentert i denne oppgaven viser at det eksisterer svakheter ved de eksisterende klinisk anvendte tolkningsalgoritmene. I denne oppgaven er det også bekreftet at KI og forklarbar KI kan brukes på EKG fra den generelle befolkning for å klassifisere en rekke diagnoser. Det er grunn til å tro dette vil kunne fungere på EKG fra idrettsutøvere også dersom datagrunnlaget er godt nok. Mangelen på data er den største begrensningen innenfor dette fagfeltet. Hovedprioriteten for å gjøre fremskritt på dette området bør derfor være å skaffe tilgang til flere relevante EKG. Plutselig hjertestans blant idrettsutøvere er sjeldent, men likevel svært tragisk når det skjer. Jeg håper flere ønsker å dele data slik de 28 idrettsutøverne har gjort i denne oppgaven. Mer åpne data, og økt oppmerksomhet vil kunne forbedre diagnostikken av idrettsutøvere, som igjen kan føre til færre unødvendige utredninger og forhåpentligvis redde liv.

Bibliografi

- [1] Harmon Kimberly G., Asif Irfan M., Klossner David og Drezner Jonathan A., "Incidence of Sudden Cardiac Death in National Collegiate Athletic Association Athletes", *Circulation*, apr. 2011. DOI: 10.1161/CIRCULATIONAHA.110.004622.
- [2] M. P. Suárez-Mier og B. Aguilera, "Causes of Sudden Death During Sports Activities in Spain", *Revista Española de Cardiología (English Edition)*, apr. 2002.
- [3] K. Bille, D. Figueiras, P. Schamasch, L. Kappenberger, J. I. Brenner, F. J. Meijboom og E. J. Meijboom, "Sudden cardiac death in athletes: the Lausanne Recommendations", *European Journal of Cardiovascular Prevention & Rehabilitation*, des. 2006. DOI: 10.1097/01.hjr.0000238397.50341.4a.
- [4] D. Corrado, C. Basso, A. Pavei, P. Michieli, M. Schiavon og G. Thiene, "Trends in Sudden Cardiovascular Death in Young Competitive Athletes After Implementation of a Preparticipation Screening Program", *JAMA*, okt. 2006. DOI: 10.1001/jama.296.13.1593.
- [5] N. Chandra, R. Bastiaenen, M. Papadakis og S. Sharma, "Sudden Cardiac Death in Young Athletes", *Journal of the American College of Cardiology*, mar. 2013. DOI: 10.1016/j.jacc.2012.08.1032.
- [6] M. T. Stokstad, H. M. Berge og K. Gjesdal, "Hjertescreening av unge idrettsutøvere", *Tidsskrift for Den norske legeförening*, 2013. DOI: 10.4045/tidsskr.13.0016.
- [7] M. C. Koester, "A Review of Sudden Cardiac Death in Young Athletes and Strategies for Preparticipation Cardiovascular Screening", *Journal of Athletic Training*, 2001.
- [8] S. Sharma, A. Merghani og L. Mont, "Exercise and the heart: the good, the bad, and the ugly", *European Heart Journal*, jun. 2015. DOI: 10.1093/eurheartj/ehv090.
- [9] A. D'Silva og M. Papadakis, "Sudden Cardiac Death in Athletes", *European Cardiology Review*, jul. 2015. DOI: 10.15420/ocr.2015.10.01.48.
- [10] M. Bickerton og A. Pooler, "Misplaced ECG Electrodes and the Need for Continuing Training", *British Journal of Cardiac Nursing*, mar. 2019. DOI: 10.12968/bjca.2019.14.3.123.

- [11] J. A. Drezner, “18 highlights from the International Criteria for ECG interpretation in athletes”, *British Journal of Sports Medicine*, feb. 2020. DOI: 10.1136/bjsports-2019-101537.
- [12] J. Schläpfer og H. J. Wellens, “Computer-Interpreted Electrocardiograms”, *Journal of the American College of Cardiology*, aug. 2017. DOI: 10.1016/j.jacc.2017.07.723.
- [13] H. Smulyan, “The Computerized ECG: Friend and Foe”, *The American Journal of Medicine*, feb. 2019. DOI: 10.1016/j.amjmed.2018.08.025.
- [14] B. Morisbak og K. Gjesdal, “[Computer-based interpretation of ECG—guiding or misleading?]”, *Tidsskrift for Den Norske Laegeforening: Tidsskrift for Praktisk Medicin, Ny Raekke*, sep. 1999.
- [15] J. A. Drezner, S. Sharma, A. Baggish, M. Papadakis, M. G. Wilson, J. M. Prutkin, A. L. Gerche, M. J. Ackerman, M. Borjesson, J. C. Salerno, I. M. Asif, D. S. Owens, E. H. Chung, M. S. Emery, V. F. Froelicher, H. Heidbuchel, C. Adamuz, C. A. Asplund, G. Cohen, K. G. Harmon, J. C. Marek, S. Molossi, J. Niebauer, H. F. Pelto, M. V. Perez, N. R. Riding, T. Saarel, C. M. Schmied, D. M. Shipon, R. Stein, V. L. Vetter, A. Pelliccia og D. Corrado, “International criteria for electrocardiographic interpretation in athletes: Consensus statement”, *British Journal of Sports Medicine*, mai 2017. DOI: 10.1136/bjsports-2016-097331.
- [16] H. M. Berge, K. Gjesdal, T. E. Andersen, E. E. Solberg og K. Steine, “Prevalence of abnormal ECGs in male soccer players decreases with the Seattle criteria, but is still high”, *Scandinavian Journal of Medicine & Science in Sports*, 2015. DOI: 10.1111/sms.12274.
- [17] J. A. Drezner, M. J. Ackerman, J. Anderson, E. Ashley, C. A. Asplund, A. L. Baggish, M. Börjesson, B. C. Cannon, D. Corrado, J. P. DiFiori, P. Fischbach, V. Froelicher, K. G. Harmon, H. Heidbuchel, J. Marek, D. S. Owens, S. Paul, A. Pelliccia, J. M. Prutkin, J. C. Salerno, C. M. Schmied, S. Sharma, R. Stein, V. L. Vetter og M. G. Wilson, “Electrocardiographic interpretation in athletes: the ‘Seattle Criteria’: Table 1”, *British Journal of Sports Medicine*, feb. 2013. DOI: 10.1136/bjsports-2012-092067.
- [18] Uberoi Abhimanyu, Stein Ricardo, Perez Marco V., Freeman James, Wheeler Matthew, Dewey Frederick, Peidro Roberto, Hadley David, Drezner Jonathan, Sharma Sanjay, Pelliccia Antonio, Corrado Domenico, Niebauer Josef, Estes N.A. Mark, Ashley Euan og Froelicher Victor, “Interpretation of the Electrocardiogram of Young Athletes”, *Circulation*, aug. 2011. DOI: 10.1161/CIRCULATIONAHA.110.013078.
- [19] Sheikh Nabeel, Papadakis Michael, Ghani Saqib, Zaidi Abbas, Gati Sabiha, Adami Paolo Emilio, Carré François, Schnell Frédéric, Wilson Mathew, Avila Paloma, McKenna William og Sharma Sanjay, “Comparison of Electrocardiographic Criteria for the Detection of Cardiac Abnormalities in Elite Black and White Athletes”, *Circulation*, apr. 2014. DOI: 10.1161/CIRCULATIONAHA.113.006179.
- [20] H. M. Berge, K. Steine, T. E. Andersen, E. E. Solberg og K. Gjesdal, “Visual or computer-based measurements: important for interpretation of athletes’ ECG”, *British Journal of Sports Medicine*, mai 2014. DOI: 10.1136/bjsports-2014-093412.

- [21] B. Pyakillya, N. Kazachenko og N. Mikhailovsky, “Deep Learning for ECG Classification”, *Journal of Physics: Conference Series*, okt. 2017. DOI: 10.1088/1742-6596/913/1/012004.
- [22] S. M. Mathews, C. Kambhamettu og K. E. Barner, “A novel application of deep learning for single-lead ECG classification”, *Computers in Biology and Medicine*, aug. 2018. DOI: 10.1016/j.combiomed.2018.05.013.
- [23] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia og A. Y. Ng, “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”, *Nature Medicine*, jan. 2019. DOI: 10.1038/s41591-018-0268-3.
- [24] A. H. Ribeiro, M. H. Ribeiro, G. M. M. Paixão, D. M. Oliveira, P. R. Gomes, J. A. Canazart, M. P. S. Ferreira, C. R. Andersson, P. W. Macfarlane, W. M. Jr, T. B. Schön og A. L. P. Ribeiro, “Automatic diagnosis of the 12-lead ECG using a deep neural network”, *Nature Communications*, apr. 2020. DOI: 10.1038/s41467-020-15432-4.
- [25] Q. Yao, R. Wang, X. Fan, J. Liu og Y. Li, “Multi-class Arrhythmia detection from 12-lead varied-length ECG using Attention-based Time-Incremental Convolutional Neural Network”, *Information Fusion*, jan. 2020. DOI: 10.1016/j.inffus.2019.06.024.
- [26] D. Li, H. Wu, J. Zhao, Y. Tao og J. Fu, “Automatic Classification System of Arrhythmias Using 12-Lead ECGs with a Deep Neural Network Based on an Attention Mechanism”, *Symmetry*, nov. 2020. DOI: 10.3390/sym12111827.
- [27] T.-M. Chen, C.-H. Huang, E. S. Shih, Y.-F. Hu og M.-J. Hwang, “Detection and Classification of Cardiac Arrhythmias by a Challenge-Best Deep Learning Neural Network Model”, *iScience*, mar. 2020. DOI: 10.1016/j.isci.2020.100886.
- [28] E. A. Perez Alday, A. Gu, A. J. Shah, C. Robichaux, A.-K. I. Wong, C. Liu, F. Liu, A. B. Rad, A. Elola, S. Seyedi, Q. Li, A. Sharma, G. D. Clifford og M. A. Reyna, “Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020”, *Physiological Measurement*, nov. 2020. DOI: 10.1088/1361-6579/abc960.
- [29] N. Vollmer, *Article 22 EU General Data Protection Regulation (EU-GDPR)*, mai 2020. side: <https://www.privacy-regulation.eu/en/article-22-automated-individual-decision-making-including-profiling-GDPR.htm> (sjekket 19.12.2020).
- [30] D. Castelvechi, “Can we open the black box of AI?”, *Nature News*, okt. 2016. DOI: 10.1038/538020a.
- [31] H. Zhang og M. Wang, “Search for the smallest random forest”, *Statistics and its interface*, jan. 2009.
- [32] H. B. Benestad, “Tolkning av EKG med kunstig intelligens”, *Tidsskrift for Den norske legeforening*, 2019. DOI: 10.4045/tidsskr.19.0119.
- [33] R. Bahr, “Can electrocardiographic screening prevent sudden death in athletes? No”, *BMJ*, sep. 2010. DOI: 10.1136/bmj.c4914.

- [34] H. Arnesen, *hjertet*, feb. 2020. side: <http://sml.snl.no/hjertet> (sjekket 30.01.2021).
- [35] *Creative Commons — Attribution-ShareAlike 4.0 International — CC BY-SA 4.0*. side: <https://creativecommons.org/licenses/by-sa/4.0/> (sjekket 20.01.2021).
- [36] *Hjertets ledningssystem*, 2020. side: https://sml.snl.no/hjertets_ledningssystem (sjekket 02.12.2020).
- [37] Unknown, *His' bunt*, 2020. side: https://sml.snl.no/His'_bunt (sjekket 02.12.2020).
- [38] A. D. Waller, "A Demonstration on Man of Electromotive Changes accompanying the Heart's Beat", *The Journal of Physiology*, okt. 1887.
- [39] S. S. Barold, "Willem Einthoven and the birth of clinical electrocardiography a hundred years ago", *Cardiac Electrophysiology Review*, jan. 2003. DOI: 10.1023/a:1023667812925.
- [40] S. Samstad, *elektrokardiograf*, jan. 2019. side: <https://sml.snl.no/elektrokardiograf>.
- [41] W. Einthoven, G. Fahr og A. De Waart, "Über die Richtung und die manifeste Grösse der Potentialschwankungen im menschlichen Herzen und über den Einfluss der Herzlage auf die Form des Elektrokardiogramms", *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere*, 1913. DOI: <https://doi.org/10.1007/BF01697566>.
- [42] Hurst J. Willis, "Naming of the Waves in the ECG, With a Brief Account of Their Genesis", *Circulation*, nov. 1998. DOI: 10.1161/01.CIR.98.18.1937.
- [43] U. D. of Health and Human Services, *National Ambulatory Medical Care Survey: 2015 State and National Summary Tables*, 2015. side: https://www.cdc.gov/nchs/data/ahcd/namcs_summary/2015_namcs_web_tables.pdf (sjekket 31.01.2021).
- [44] Unknown, *Willem Einthoven ECG*. side: https://commons.wikimedia.org/wiki/File:Willem_Einthoven_ECG.jpg (sjekket 20.01.2021).
- [45] *CC0*. side: <https://creativecommons.org/share-your-work/public-domain/cc0/> (sjekket 20.01.2021).
- [46] Npatchett, *English: Derivation of the limb leads*, mar. 2015. side: https://commons.wikimedia.org/wiki/File:Limb_leads_of_EKG.png (sjekket 20.01.2021).
- [47] M. K. Huseby, "FPGA Based Development Platform for Biomedical Measurements", masteroppg., University of Oslo, Oslo, okt. 2013. side: <https://www.duo.uio.no/handle/10852/37175>.
- [48] H. Arnesen, *EKG*, okt. 2018. side: <http://sml.snl.no/EKG> (sjekket 04.06.2020).
- [49] Twisp, *Precordial leads*, jan. 2008. side: <https://commons.wikimedia.org/w/index.php?curid=3475996> (sjekket 20.01.2021).
- [50] G. Yan, S. Liang, Y. Zhang og F. Liu, "Fusing Transformer Model with Temporal Features for ECG Heartbeat Classification", i *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, nov. 2019. DOI: 10.1109/BIBM47256.2019.8983326.

- [51] A. C. V. Maggio, M. P. Bonomini, E. L. Leber og P. D. Arini, “Quantification of Ventricular Repolarization Dispersion Using Digital Processing of the Surface ECG”, *Advances in Electrocardiograms - Methods and Analysis*, jan. 2012. DOI: 10.5772/23050.
- [52] N. Pan, V. Mang, M. P. Un og P. S. hang, “Accurate Removal of Baseline Wander in ECG Using Empirical Mode Decomposition”, i *2007 Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*, okt. 2007. DOI: 10.1109/NFSI-ICFBI.2007.4387719.
- [53] Y. Luo, R. H. Hargraves, A. Belle, O. Bai, X. Qi, K. R. Ward, M. P. Pfaffenberger og K. Najarian, “A Hierarchical Method for Removal of Baseline Drift from Biomedical Signals: Application in ECG Analysis”, *The Scientific World Journal*, 2013. DOI: 10.1155/2013/896056.
- [54] dy1901, *ecg-plot: Plot standard ECG chart from data*. side: https://github.com/dy1901/ecg_plot (sjekket 10.10.2020).
- [55] E. Rossen, *ekspertsystem*, aug. 2015. side: <https://snl.no/ekspertsystem>.
- [56] J. S. Alpert, “Can You Trust a Computer to Read Your Electrocardiogram?”, *The American Journal of Medicine*, jun. 2012. DOI: 10.1016/j.amjmed.2012.02.001.
- [57] J. Morganroth, B. J. Maron, W. L. Henry og S. E. Epstein, *Comparative left ventricular dimensions in trained athletes*. *Ann Intern Med*.
- [58] Levine Benjamin D., Baggish Aaron L., Kovacs Richard J., Link Mark S., Maron Martin S. og Mitchell Jere H., “Eligibility and Disqualification Recommendations for Competitive Athletes With Cardiovascular Abnormalities: Task Force 1: Classification of Sports: Dynamic, Static, and Impact”, *Circulation*, des. 2015. DOI: 10.1161/CIR.0000000000000237.
- [59] J. H. Mitchell, W. Haskell, P. Snell og S. P. Van Camp, “Task Force 8: Classification of sports”, *Journal of the American College of Cardiology*, apr. 2005. DOI: 10.1016/j.jacc.2005.02.015.
- [60] I. Moon, S.-Y. Lee, H.-K. Kim, K.-D. Han, S. Kwak, M. Kim, H.-J. Lee, I.-C. Hwang, H. Lee, J.-B. Park, Y. E. Yoon, Y.-J. Kim og G.-Y. Cho, “Trends of the prevalence and incidence of hypertrophic cardiomyopathy in Korea: A nationwide population-based cohort study”, *PLOS ONE*, jan. 2020. DOI: 10.1371/journal.pone.0227012.
- [61] M. M. Wasfy, A. M. Hutter og R. B. Weiner, “Sudden Cardiac Death in Athletes”, *Methodist DeBakey Cardiovascular Journal*, apr. 2016. DOI: 10.14797/mdcj-12-2-76.
- [62] T. K. Chin, J. K. Perloff, R. G. Williams, K. Jue og R. Mohrmann, “Isolated noncompaction of left ventricular myocardium. A study of eight cases.”, *Circulation*, aug. 1990. DOI: 10.1161/01.CIR.82.2.507.

- [63] B. Bauce, G. Frigo, G. Benini, P. Michieli, C. Basso, A. F. Folino, I. Rigato, E. Mazzotti, L. Daliento, G. Thiene og A. Nava, "Differences and similarities between arrhythmogenic right ventricular cardiomyopathy and athlete's heart adaptations", *British Journal of Sports Medicine*, feb. 2010. DOI: 10.1136/bjism.2007.042853.
- [64] Cipriani Alberto, Bauce Barbara, De Lazzari Manuel, Rigato Ilaria, Bariani Riccardo, Meneghin Samuele, Pilichou Kalliopi, Motta Raffaella, Aliberti Camillo, Thiene Gaetano, McKenna William J., Zorzi Alessandro, Iliceto Sabino, Basso Cristina, Perazzolo Marra Martina og Corrado Domenico, "Arrhythmogenic Right Ventricular Cardiomyopathy: Characterization of Left Ventricular Phenotype and Differential Diagnosis With Dilated Cardiomyopathy", *Journal of the American Heart Association*, mar. 2020. DOI: 10.1161/JAHA.119.014628.
- [65] K. Haugaa, *Idrettsutøveren: hvor går grensen mot kardial patologi?*, 2012. side: <https://www.legeforeningen.no/contentassets/06b9b057542644b99ecfe60271155955/2012-nsc-hostmote-k-haugaa-til-ola.pdf>.
- [66] S. N. Shah, K. K. Umapathi og T. I. Oliver, "Arrhythmogenic Right Ventricular Cardiomyopathy", i *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2020.
- [67] J. L. Jefferies og J. A. Towbin, "Dilated cardiomyopathy", *The Lancet*, feb. 2010. DOI: 10.1016/S0140-6736(09)62023-7.
- [68] J. Basu og A. Malhotra, "Interpreting the Athlete's ECG: Current State and Future Perspectives", *Current Treatment Options in Cardiovascular Medicine*, 2018. DOI: 10.1007/s11936-018-0693-0.
- [69] Knut Gjesdal, *EKG I HJERTEFORUM - Venstre ventrikelhypertrofi*. side: <https://www.legeforeningen.no/contentassets/3a84001a3a6b42a1a94cf458a10047cf/webhjerteforum-2.2020-10-ekg-i-hjerteforum.pdf> (sjekket 19.11.2020).
- [70] M. Sokolow og T. P. Lyon, "The ventricular complex in left ventricular hypertrophy as obtained by unipolar precordial and limb leads", *American Heart Journal*, feb. 1949. DOI: 10.1016/0002-8703(49)90562-1.
- [71] S. Sharma, G. Whyte, P. Elliott, M. Padula, R. Kaushal, N. Mahon og W. J. McKenna, "Electrocardiographic changes in 1000 highly trained junior elite athletes.", *British Journal of Sports Medicine*, okt. 1999. DOI: 10.1136/bjism.33.5.319.
- [72] M. Papadakis, F. Carre, G. Kervio, J. Rawlins, V. F. Panoulas, N. Chandra, S. Basavarajaiah, L. Carby, T. Fonseca og S. Sharma, "The prevalence, distribution, and clinical outcomes of electrocardiographic repolarization patterns in male athletes of African/Afro-Caribbean origin", *European Heart Journal*, sep. 2011. DOI: 10.1093/eurheartj/ehrl40.

- [73] A. Zaidi, S. Ghani, N. Sheikh, S. Gati, R. Bastiaenen, B. Madden, M. Papadakis, H. Raju, M. Reed, R. Sharma, E. R. Behr og S. Sharma, “Clinical significance of electrocardiographic right ventricular hypertrophy in athletes: comparison with arrhythmogenic right ventricular cardiomyopathy and pulmonary hypertension”, *European Heart Journal*, des. 2013. DOI: 10.1093/eurheartj/ehz391.
- [74] H. Arnesen, *grenblokk*, jan. 2020. side: <http://sml.sn1.no/grenblokk> (sjekket 05.01.2021).
- [75] J. H. Kim, P. A. Noseworthy, D. McCarty, K. Yared, R. Weiner, F. Wang, M. J. Wood, A. M. Hutter, M. H. Picard og A. L. Baggish, “Significance of Electrocardiographic Right Bundle Branch Block in Trained Athletes”, *American Journal of Cardiology*, apr. 2011. DOI: 10.1016/j.amjcard.2010.11.037.
- [76] P. Aagaard, B. Baranowski, P. Aziz og D. Phelan, “Early Repolarization in Athletes”, *Circulation: Arrhythmia and Electrophysiology*, mar. 2016. DOI: 10.1161/CIRCEP.115.003577.
- [77] J. Basu, A. Malhotra, V. Styliandis, H. D. Miles, G. Parry-Williams, M. Tome, S. Sharma og M. Papadakis, “71 Prevalence and progression of the juvenile pattern in the electrocardiogram of adolescents”, *Heart*, jun. 2018. DOI: 10.1136/heartjnl-2018-BCS.71.
- [78] H. Arnesen, *Sinusarytmi*, jan. 2020. side: <https://sml.sn1.no/sinusarytmi> (sjekket 21.12.2020).
- [79] M. Zehender, T. Meinertz, J. Keul og H. Just, “ECG variants and cardiac arrhythmias in athletes: Clinical relevance and prognostic importance”, *American Heart Journal*, jun. 1990. DOI: 10.1016/S0002-8703(05)80189-9.
- [80] K. Gjesdal, M. Bjaanes og M. K. Stokke, *Eløring fra med.fak. UiO*. side: <http://meddev.uio.no/elaring/fag/hjertesykdommer/index.shtml> (sjekket 20.01.2021).
- [81] M. Mysid, *The hexaxial reference system*. Aug. 2007. side: https://commons.wikimedia.org/wiki/File:Hexaxial_reference_system.svg (sjekket 20.01.2021).
- [82] S. Gati, N. Sheikh, S. Ghani, A. Zaidi, M. Wilson, H. Raju, A. Cox, M. Reed, M. Papadakis og S. Sharma, “Should axis deviation or atrial enlargement be categorised as abnormal in young athletes? The athlete’s electrocardiogram: time for re-appraisal of markers of pathology”, *European Heart Journal*, des. 2013. DOI: 10.1093/eurheartj/ehz390.
- [83] C. Ng, A. Ahmad, D. R. Budhram, M. He, N. Balakrishnan og T. Mondal, “Accuracy of Electrocardiography and Agreement with Echocardiography in the Diagnosis of Pediatric Left Atrial Enlargement”, *Scientific Reports*, jun. 2020. DOI: 10.1038/s41598-020-66987-7.
- [84] B.-J. Singstad, *Brugadas syndrom*, jun. 2020. side: https://sml.no/Brugadas_syndrom (sjekket 31.01.2021).
- [85] W. Akhtar, K. S. Rathod, J. Alexander og F. Farooqi, “Profound first-degree atrioventricular block”, *Case Reports*, sep. 2013. DOI: 10.1136/bcr-2013-010474.

- [86] M. A. Mangi, W. M. Jones, M. K. Mansour og L. Napier, “Atrioventricular Block Second-Degree”, i *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2020.
- [87] V. Knabben, L. Chhabra og M. Slane, “Third-Degree Atrioventricular Block”, i *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2020.
- [88] K. Farzam og J. R. Richards, “Premature Ventricular Contraction”, i *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2020.
- [89] M. Liwanag og C. Willoughby, “Atrial Tachycardia”, i *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2020.
- [90] H. Arnesen, *supraventrikulær takykardi*, jan. 2020. side: http://snl.no/supraventrikul%C3%A6r_takykardi (sjekket 04.01.2021).
- [91] R. M. John, U. B. Tedrow, B. A. Koplan, C. M. Albert, L. M. Epstein, M. O. Sweeney, A. L. Miller, G. F. Michaud og W. G. Stevenson, “Ventricular arrhythmias and sudden cardiac death”, *The Lancet*, okt. 2012. DOI: 10.1016/S0140-6736(12)61413-5.
- [92] S. Lydersen, “Hva er sannsynligheten for riktig resultat av en diagnostisk test?”, *Tidsskrift for Den norske legeforening*, okt. 2017. DOI: 10.4045/tidsskr.17.0409.
- [93] A. Tidemann, *kunstig intelligens*, jan. 2020. side: http://snl.no/kunstig_intelligens (sjekket 22.06.2020).
- [94] A. Tidemann og A. C. Elster, *maskinl ring*, jun. 2019. side: <http://snl.no/maskinl%C3%A6ring> (sjekket 22.06.2020).
- [95] W. S. McCulloch og W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysics*, des. 1943. DOI: 10.1007/BF02478259.
- [96] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain”, *Psychological Review*, 1958. DOI: 10.1037/h0042519.
- [97] D. Stutz, *Illustrating (Convolutional) Neural Networks in LaTeX with TikZ* • David Stutz, jun. 2020. side: <https://davidstutz.de/illustrating-convolutional-neural-networks-in-latex-with-tikz/> (sjekket 20.01.2021).
- [98] P. J. Werbos, “Beyond regression: new tools for prediction and analysis in the behavioral sciences”, English, OCLC: 77001455, ph.d.-avh., 1975.
- [99] D. E. Rumelhart, G. E. Hinton og R. J. Williams, “Learning representations by back-propagating errors”, *Nature*, okt. 1986. DOI: 10.1038/323533a0.
- [100] Stanford University School of Engineering, *Lecture 4 — Introduction to Neural Networks*, aug. 2017. side: <https://www.youtube.com/watch?v=d14TUNcBn1k> (sjekket 24.01.2021).
- [101] M. Lin, Q. Chen og S. Yan, “Network In Network”, *arXiv:1312.4400 [cs]*, mar. 2014.
- [102] N. Bjorck, C. P. Gomes, B. Selman og K. Q. Weinberger, “Understanding Batch Normalization”, i *Advances in Neural Information Processing Systems*, bd. 31, 2018.

- [103] D. Ulyanov, A. Vedaldi og V. Lempitsky, “Instance Normalization: The Missing Ingredient for Fast Stylization”, *arXiv:1607.08022 [cs]*, nov. 2017.
- [104] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever og R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, *Journal of Machine Learning Research*, 2014.
- [105] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu og X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”, *arXiv:1603.04467 [cs]*, mar. 2016.
- [106] E. Callaway, “‘It will change everything’: DeepMind’s AI makes gigantic leap in solving protein structures”, *Nature*, nov. 2020. DOI: 10.1038/d41586-020-03348-4.
- [107] T. G. Dietterich, “Ensemble Methods in Machine Learning”, i *Multiple Classifier Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.
- [108] P. Szymanski og T. Kajdanowicz, “Scikit-multilearn: a scikit-based Python environment for performing multi-label classification”, *The Journal of Machine Learning Research*, jan. 2019.
- [109] J. Read, B. Pfahringer, G. Holmes og E. Frank, “Classifier chains for multi-label classification”, i *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2009.
- [110] M.-L. Zhang, Y.-K. Li, X.-Y. Liu og X. Geng, “Binary relevance for multi-label learning: an overview”, *Frontiers of Computer Science*, apr. 2018. DOI: 10.1007/s11704-017-7031-7.
- [111] Tin Kam Ho, “Random decision forests”, i *Proceedings of 3rd International Conference on Document Analysis and Recognition*, bd. 1, aug. 1995. DOI: 10.1109/ICDAR.1995.598994.
- [112] T. Hastie, R. Tibshirani og J. Friedman, “Random Forests”, i *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer Series in Statistics, T. Hastie, R. Tibshirani og J. Friedman, red., New York, NY: Springer, 2009, s. 587–604. DOI: 10.1007/978-0-387-84858-7_15.
- [113] V. Jagannath, *Bahasa Indonesia: Random Forest*, mar. 2017. side: https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png (sjekket 20.01.2021).
- [114] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal og V. K. Asari, “A State-of-the-Art Survey on Deep Learning Theory and Architectures”, *Electronics*, mar. 2019. DOI: 10.3390/electronics8030292.

- [115] J. Deng, W. Dong, R. Socher, L. Li, Kai Li og Li Fei-Fei, “ImageNet: A large-scale hierarchical image database”, i *2009 IEEE Conference on Computer Vision and Pattern Recognition*, ISSN: 1063-6919, jun. 2009. DOI: 10.1109/CVPR.2009.5206848.
- [116] G. Van Steenkiste, G. van Loon og G. Crevecoeur, “Transfer Learning in ECG Classification from Human to Horse Using a Novel Parallel Neural Network Architecture”, *Scientific Reports*, jan. 2020. DOI: 10.1038/s41598-019-57025-2.
- [117] M. T. Ribeiro, S. Singh og C. Guestrin, ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”, *arXiv:1602.04938 [cs, stat]*, aug. 2016.
- [118] K. Aas, M. Jullum og A. Løland, “Explaining individual predictions when features are dependent: More accurate approximations to Shapley values”, *arXiv:1903.10464 [cs, stat]*, feb. 2020.
- [119] M. Du, N. Liu og X. Hu, “Techniques for Interpretable Machine Learning”, *arXiv:1808.00033 [cs, stat]*, mai 2019.
- [120] L. Kopitar, L. Cilar, P. Kocbek og G. Stiglic, “Local vs. Global Interpretability of Machine Learning Models in Type 2 Diabetes Mellitus Screening”, i *Artificial Intelligence in Medicine: Knowledge Representation and Transparent and Explainable Systems*, M. Marcos, J. M. Juarez, R. Lenz, G. J. Nalepa, S. Nowaczyk, M. Peleg, J. Stefanowski og G. Stiglic, red., Cham: Springer International Publishing, 2019.
- [121] S. M. Lundberg og S.-I. Lee, “A Unified Approach to Interpreting Model Predictions”, *Advances in Neural Information Processing Systems*, 2017.
- [122] *Ambu BlueSensor Q*. side: <https://www.ambu.com/cardiology/ecg-electrodes/product/ambu-bluesensor-q> (sjekket 23.11.2020).
- [123] *Marquette™ 12SL™ ECG Analysis Program Physician’s Guide, 12SL version 23, revision B*, jul. 2015.
- [124] B.-J. Singstad og C. Tronstad, “Convolutional Neural Network and Rule-Based Algorithms for Classifying 12-lead ECGs”, des. 2020. DOI: 10.22489/CinC.2020.227.
- [125] F. Liu, C. Liu, L. Zhao, X. Zhang, X. Wu, X. Xu, Y. Liu, C. Ma, S. Wei, Z. He, J. Li og E. N. Yin Kwee, “An Open Access Database for Evaluating the Algorithms of Electrocardiogram Rhythm and Morphology Abnormality Detection”, *Journal of Medical Imaging and Health Informatics*, sep. 2018. DOI: 10.1166/jmihi.2018.2442.
- [126] R. Bousseljot, D. Kreiseler og A. Schnabel, “Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet”, *Biomedizinische Technik/Biomedical Engineering*, jul. 2009. DOI: 10.1515/bmte.1995.40.s1.317.
- [127] P. Wagner, N. Strodthoff, R. Bousseljot, W. Samek og T. Schaeffter, “PTB-XL, a large publicly available electrocardiography dataset (version 1.0.1). PhysioNet.”, 2020. DOI: <https://doi.org/10.13026/x4td-x982..>

- [128] Tihonenko, V., Khaustov, A, Ivanov, S og Rivin, A., “St.-Petersburg Institute of Cardiological Technics 12-lead Arrhythmia Database [Data set].”, 2007. DOI: <https://doi.org/10.13026/C2V88N>.
- [129] A. L. Goldberger og et al., “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals”, *Circulation*, jun. 2000. DOI: 10.1161/01.CIR.101.23.e215.
- [130] Bjørn-Jostein Singstad, *ECG-featurizer*. side: <https://doi.org/10.5281/ZENODO.4265151>.
- [131] D. Makowski, T. Pham, Z. J. Lau, J. C. Brammer, F. Lespinasse, H. Pham, C. Schölzel og A. S H Chen, *NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing*. Zenodo, 2020. DOI: 10.5281/ZENODO.3597887.
- [132] S. Van Der Walt, S. C. Colbert og G. Varoquaux, “The NumPy array: a structure for efficient numerical computation”, *Computing in Science & Engineering*, 2011.
- [133] F. Pedregosa og et al., “Scikit-Learn: Machine Learning in Python”, *Journal of Machine Learning Research*, 2011.
- [134] H. I. Fawaz og et al., “Deep Learning for Time Series Classification: A Review”, *Data Mining and Knowledge Discovery*, jul. 2019. DOI: 10.1007/s10618-019-00619-1.
- [135] J. Pan og W. J. Tompkins, “A Real-Time QRS Detection Algorithm”, *IEEE Transactions on Biomedical Engineering*, mar. 1985. DOI: 10.1109/TBME.1985.325532.
- [136] J. A. Nelder og R. Mead, “A Simplex Method for Function Minimization”, *The Computer Journal*, jan. 1965. DOI: 10.1093/comjnl/7.4.308.
- [137] P. Virtanen og et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”, *Nature Methods*, mar. 2020. DOI: 10.1038/s41592-019-0686-2.
- [138] D. P. Kingma og J. Ba, “Adam: A Method for Stochastic Optimization”, *arXiv:1412.6980 [cs]*, jan. 2017.
- [139] P. Szymański, T. Kajdanowicz og K. Kersting, “How Is a Data-Driven Approach Better than Random Choice in Label Space Division for Multi-Label Classification?”, *Entropy*, 2016. DOI: 10.3390/e18080282.
- [140] *Apache License, Version 2.0*. side: <https://www.apache.org/licenses/LICENSE-2.0> (sjekket 20.01.2021).
- [141] *GNU license*. side: <https://www.gnu.org/licenses/gpl-3.0.en.html> (sjekket 20.01.2021).
- [142] J.-S. Uhm, Y. Lee, Y. H. Roh, J. Lee, D. Kang, M.-N. Jin, I.-S. Kim, H. T. Yu, T.-H. Kim, J.-Y. Kim, B. Joung, H.-N. Pak og M.-H. Lee, “Nonspecific intraventricular conduction delay is associated with future occurrence of atrial fibrillation in patients with structurally normal heart”, *European Journal of Internal Medicine*, feb. 2020. DOI: 10.1016/j.ejim.2019.11.006.

- [143] PeaBrainC, *Cartoon showing measurement of the QT interval*, nov. 2018. side: https://commons.wikimedia.org/wiki/File:QT_interval.jpg (sjekket 28.01.2021).
- [144] S. W. Smith, A. Khalil, T. D. Henry, M. Rosas, R. J. Chang, K. Heller, E. Scharrer, M. Ghorashi og L. A. Pearce, “Electrocardiographic Differentiation of Early Repolarization From Subtle Anterior ST-Segment Elevation Myocardial Infarction”, *Annals of Emergency Medicine*, jul. 2012. DOI: 10.1016/j.annemergmed.2012.02.015.
- [145] N. Bosson, S. Sanko, R. E. Stickney, J. Niemann, W. J. French, J. G. Jollis, M. C. Kontos, T. G. Taylor, P. W. Macfarlane, R. Tadeo, W. Koenig og M. Eckstein, “Causes of Prehospital Misinterpretations of ST Elevation Myocardial Infarction”, *Prehospital Emergency Care*, mai 2017. DOI: 10.1080/10903127.2016.1247200.
- [146] A. J. J. McClelland, C. G. Owens, I. B. A. Menown, M. Lown og A. A. J. Adgey, “Comparison of the 80-lead body surface map to physician and to 12-lead electrocardiogram in detection of acute myocardial infarction”, *The American Journal of Cardiology*, aug. 2003. DOI: 10.1016/S0002-9149(03)00619-2.
- [147] D. Massel, J. A. Dawdy og L. J. Melendez, “Strict reliance on a computer algorithm or measurable ST segment criteria may lead to errors in thrombolytic therapy eligibility”, *American Heart Journal*, aug. 2000. DOI: 10.1067/mhj.2000.108240.
- [148] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij og J. Rubin, “A Wide and Deep Transformer Neural Network for 12-Lead ECG Classification”, des. 2020. DOI: 10.22489/CinC.2020.107.
- [149] S. Yang, H. Xiang, Q. Kong og C. Wang, “Multi-label Classification of Electrocardiogram With Modified Residual Networks”, des. 2020. DOI: 10.22489/CinC.2020.007.
- [150] w. jia wenzhao, X. Xu, X. Xu, Y. Sun og X. Liu, “Arrhythmia Detection and Classification of 12-lead ECGs Using a Deep Neural Network”, des. 2020. DOI: 10.22489/CinC.2020.035.
- [151] J. Chen, T. Chen, B. Xiao, X. Bi, Y. Wang, W. Li, H. Duan, J. Zhang og X. Ma, “SE-ECGNet: Multi-scale SE-Net for Multi-lead ECG Data”, des. 2020. DOI: 10.22489/CinC.2020.085.
- [152] Y. Feng og E. Vigmond, “Deep Multi-Label Multi-Instance Classification on 12-Lead ECG”, des. 2020. DOI: 10.22489/CinC.2020.095.
- [153] Z. Zhao, H. Fang, S. Relton, R. Yan, Y. Liu, Z. Li, J. Qin og D. Wong, “Adaptive lead weighted ResNet trained with different duration signals for classifying 12-lead ECGs”, des. 2020. DOI: 10.22489/CinC.2020.112.
- [154] M. Alkhodari, L. J. Hadjileontiadis og A. H. Khandoker, “Identification of Cardiac Arrhythmias from 12-lead ECG using Beat-wise Analysis and a Combination of CNN and LSTM”, des. 2020. DOI: 10.22489/CinC.2020.127.
- [155] P.-Y. Hsu, P.-H. Hsu, T.-H. Lee og H.-L. Liu, “Multi-label Arrhythmia Classification From 12-Lead Electrocardiograms”, des. 2020. DOI: 10.22489/CinC.2020.134.

- [156] Z. Jiang, T. Paggi de Almeida, F. Schlindwein, G. André Ng, H. Zhou og X. Li, “Diagnostic of Multiple Cardiac Disorders from 12-lead ECGs Using Graph Convolutional Network Based Multi-label Classification”, des. 2020. DOI: 10.22489/CinC.2020.135.
- [157] M. Baumgartner, D. Hayn, A. Ziegl, A. Eggerth og G. Schreier, “Multi-Stream Deep Neural Network for 12-Lead ECG Classification”, des. 2020. DOI: 10.22489/CinC.2020.148.
- [158] N. Fayyazifar, S. Ahderom, D. Suter, A. Maiorana og G. dwivedi, “Impact of Neural Architecture Design on Cardiac Abnormality Classification Using 12-lead ECG Signals”, des. 2020. DOI: 10.22489/CinC.2020.161.
- [159] H. Sigurthorsdottir, J. Van Zaen, R. Delgado-Gonzalo og M. Lemay, “ECG Classification With a Convolutional Recurrent Neural Network”, des. 2020. DOI: 10.22489/CinC.2020.198.
- [160] L. Weber, M. Gaiduk, W. Daniel Scherz og R. Seepold, “Cardiac Abnormality Detection in 12-lead ECGs with Deep Convolutional Neural Networks Using Data Augmentation”, des. 2020. DOI: 10.22489/CinC.2020.229.
- [161] P. Samuel Ignacio, J.-A. Bulauan og J. Rick Manzanares, “A Topology Informed Random Forest Classifier for ECG Classification”, des. 2020. DOI: 10.22489/CinC.2020.297.
- [162] S. Min, H.-S. Choi, H. Han, M. Seo, J.-K. Kim, J. Park, S. Jung, I.-Y. Oh, B. Lee og S. Yoon, “Bag of Tricks for Electrocardiogram Classification with Deep Neural Networks”, des. 2020. DOI: 10.22489/CinC.2020.328.
- [163] C. Zisou, A. Sochopoulos og K. Kitsios, “Convolutional Recurrent Neural Network and LightGBM Ensemble Model for 12-lead ECG Classification”, des. 2020. DOI: 10.22489/CinC.2020.417.
- [164] D. Nankani, P. Saikia og R. Dutta Baruah, “Automatic Concurrent Arrhythmia Classification Using Deep Residual Neural Networks”, des. 2020. DOI: 10.22489/CinC.2020.424.
- [165] H. Hasani, A. Bitarafan og M. Soleymani, “Classification of 12-lead ECG Signals With Adversarial Multi-Source Domain Generalization”, des. 2020. DOI: 10.22489/CinC.2020.445.
- [166] P. Nejedly, A. Ivora, I. Viscor, J. Halamek, P. Jurak og F. Plesinger, “Utilization of Residual CNN-GRU with Attention Mechanism for Classification of 12-lead ECG”, des. 2020. DOI: 10.22489/CinC.2020.032.
- [167] M. Żyliński og G. Cybulski, “Selected Features for Classification of 12-lead ECGs”, des. 2020. DOI: 10.22489/CinC.2020.061.
- [168] g. bortolan giovanni, I. Christov og I. Simova, “Rule-Based methods and Deep Learning Networks for Automatic Classification of ECG”, des. 2020. DOI: 10.22489/CinC.2020.116.
- [169] M. Soliński, M. Lepek, A. Pater, K. Muter, P. Wiszniewski, D. Kokosińska, J. Salamon og Z. Puzio, “12-lead ECG Arrhythmia Classification Using Convolutional Neural Network for Mutually Non-Exclusive Classes”, des. 2020. DOI: 10.22489/CinC.2020.124.

- [170] A. Demonbreun og G. Mirsky, “Automated Classification of Electrocardiograms Using Wavelet Analysis and Deep Learning”, des. 2020. DOI: 10.22489/CinC.2020.138.
- [171] R. Smisek, A. Nemcova, L. Marsanova, L. Smital, M. Vitek og J. Kozumplik, “Cardiac Pathologies Detection and Classification in 12-lead ECG”, des. 2020. DOI: 10.22489/CinC.2020.171.
- [172] H. Rajpal, M. Sas, R. Joakim, C. Lockwood, N. S. Peters og M. Falkenberg, “Interpretable XGBoost Based Classification of 12-lead ECGs Applying Information Theory Measures From Neuroscience”, des. 2020. DOI: 10.22489/CinC.2020.185.
- [173] A. Kirodiwal, A. Srivastava, A. Dash, A. Saha, G. Vamsi Penaganti, S. Pratiher, s. alam sazedul, A. Patra, N. Ghosh og N. Banerjee, “A Bio-toolkit for Multi-Cardiac Abnormality Diagnosis Using ECG Signal and Deep Learning”, des. 2020. DOI: 10.22489/CinC.2020.225.
- [174] R. Pardasani og N. Awasthi, “Classification of 12 Lead ECG Signal Using 1D-CNN With Class Dependent Threshold”, des. 2020. DOI: 10.22489/CinC.2020.277.
- [175] Z. Zhu, H. Wang, T. Zhao, Y. Guo, Z. Xu, Z. Liu, S. Liu, X. Lan, X. Sun og M. Feng, “Classification of Cardiac Abnormalities From ECG Signals Using SE-ResNet”, des. 2020. DOI: 10.22489/CinC.2020.281.
- [176] R. Duan, X. He og O. Zhuoran, “MADNN: A Multi-scale Attention Deep Neural Network for Arrhythmia Classification”, des. 2020. DOI: 10.22489/CinC.2020.282.
- [177] M. Sallem, A. Saadaoui, A. Ghrissi og V. Zarzoso, “Detection of Cardiac Arrhythmias From Varied Length Multichannel Electrocardiogram Recordings Using Deep Convolutional Neural Networks”, des. 2020. DOI: 10.22489/CinC.2020.339.
- [178] P. Sodmann og M. Vollmer, “ECG Segmentation using a Neural Network as the Basis for Detection of Cardiac Pathologies”, des. 2020. DOI: 10.22489/CinC.2020.356.
- [179] M. Bodini, M. W Rivolta og R. Sassi, “Classification of 12-lead ECG With an Ensemble Machine Learning Approach”, des. 2020. DOI: 10.22489/CinC.2020.406.
- [180] N. Strodthoff, P. Wagner, T. Schaeffter og W. Samek, “Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL”, *arXiv:2004.13701 [cs, stat]*, apr. 2020.
- [181] D. Zhang, X. Yuan og P. Zhang, “Interpretable Deep Learning for Automatic Diagnosis of 12-lead Electrocardiogram”, *arXiv:2010.10328 [cs, eess]*, okt. 2020.
- [182] van de Leur Rutger R., Blom Lennart J., Gavves Efstratios, Hof Irene E., van der Heijden Jeroen F., Clappers Nick C., Doevendans Pieter A., Hassink Rutger J. og van Es René, “Automatic Triage of 12-Lead ECGs Using Deep Convolutional Neural Networks”, *Journal of the American Heart Association*, mai 2020. DOI: 10.1161/JAHA.119.015138.

- [183] Y.-C. Lin, Y.-C. Lee, W.-C. Tsai, W.-K. Beh og A.-Y. Wu, “Explainable Deep Neural Network for Identifying Cardiac Abnormalities Using Class Activation Map”, des. 2020. DOI: 10.22489/CinC.2020.072.
- [184] F. Mellestrand, “Simulation of 12 Lead ECG”, eng, Accepted: 2017-09-19T13:32:56Z Publisher: University of Stavanger, Norway, masteroppg., jun. 2017. side: <https://uis.brage.unit.no/uis-xmlui/handle/11250/2455552> (sjekket 05.01.2021).
- [185] A. M. Delaney, E. Brophy og T. E. Ward, “Synthesis of Realistic ECG using Generative Adversarial Networks”, *arXiv:1909.09150 [cs, eess, stat]*, sep. 2019.
- [186] S. G. Tischer, N. Mattsson, M. Storgaard, D. E. Høfsten, N. B. Høst, L. J. Andersen, E. Prescott og H. K. Rasmusen, “Results of voluntary cardiovascular examination of elite athletes in Denmark: Proposal for Nordic collaboration”, *Scandinavian Journal of Medicine & Science in Sports*, jan. 2016. DOI: 10.1111/sms.12405.
- [187] M. M. Søndergaard, J. B. Nielsen, R. N. Mortensen, G. Gislason, L. Køber, F. Lippert, C. Graff, S. Haunsø, J. H. Svendsen, K. H. Kragholm, A. H. Pietersen, B. S. Lind, S. P. Hjortshøj, A. G. Holst, J. J. Struijk, C. Torp-Pedersen og S. M. Hansen, “Associations between common ECG abnormalities and out-of-hospital cardiac arrest”, *Open Heart*, mai 2019. DOI: 10.1136/openhrt-2018-000905.

Appendiks A

Innstillinger EKG MAC VU360TM og MarquetteTM SL12

Innstillinger EKG GE MAC VU360TM og MarquetteTM SL12

Innstillinger EKG MAC VU360 TM og Marquette TM SL12	
Innstillinger for bølgeform og skjerminnstillinger	
Standard hastighet	25.0 mm/s
Standard gevinst / forsterkning	10mm/mV
Skjermfilter	150 Hz
12SL-innstillinger	
QTc-metode	Fridericia
Vekselstrømfilter	50Hz
Avledningsmerketype	IEC
Påvisningskonfigurasjon	
STEMI	Ja
Iskemi	Ja
AV-blokk	Ja
Arytmi	Ja
Programvareversjon	
Programvare fullversjon	1.01 SP04
12SL-versjon	12SL 23

Appendiks B

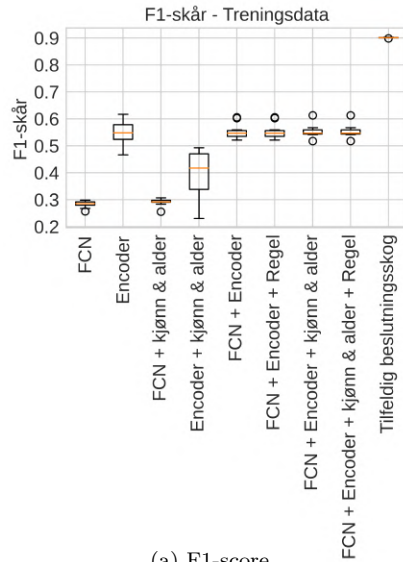
Metadata-fil fra PhysioNet/CinC Challenge 2020

Tabellen viser et eksempel på metadata-fil. Første linje i tabellen viser nummer på opptaket (A0001), antall EKG-avledninger (12), samplingsfrekvens (500Hz), antall samples (7500), dato (12-May-2020) og klokkeslett (12:33:59). De neste 12 radene viser igjen filnavnet, hvert signal ble skrevet med 16 bit og 24bits offset, oppløsningen på spenningssignalet (1000/mV), EKG-apparatets Analog-til-digital konverter verdier (16+24), grunnverdien til signalet er 0 for alle ledere, den første verdien til signalet, checksum og tilslutt navnet på avledningen. I de 6 siste radene gis det informasjon om alder, kjønn, diagnose (Dx) kodet/kryptert med SNOMED-CT kode, Resept (Rx), pasienthistorie (Hx) og symptom eller operasjon (Sx).

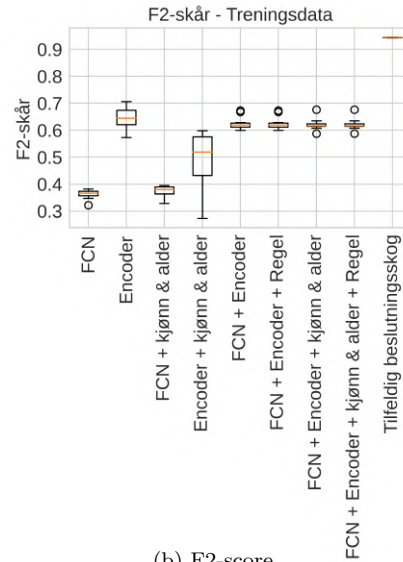
```
A0001 12 500 7500 12-May-2020 12:33:59
A0001.mat 16+24 1000/mV 16 0 28 -1716 0 I
A0001.mat 16+24 1000/mV 16 0 7 2029 0 II
A0001.mat 16+24 1000/mV 16 0 -21 3745 0 III
A0001.mat 16+24 1000/mV 16 0 -17 3680 0 aVR
A0001.mat 16+24 1000/mV 16 0 24 -2664 0 aVL
A0001.mat 16+24 1000/mV 16 0 -7 -1499 0 aVF
A0001.mat 16+24 1000/mV 16 0 -290 390 0 V1
A0001.mat 16+24 1000/mV 16 0 -204 157 0 V2
A0001.mat 16+24 1000/mV 16 0 -96 -2555 0 V3
A0001.mat 16+24 1000/mV 16 0 -112 49 0 V4
A0001.mat 16+24 1000/mV 16 0 -596 -321 0 V5
A0001.mat 16+24 1000/mV 16 0 -16 -3112 0 V6
#Age: 74
#Sex: Male
#Dx: 59118001
#Rx: Unknown
#Hx: Unknown
#Sx: Unknown
```

Appendiks C

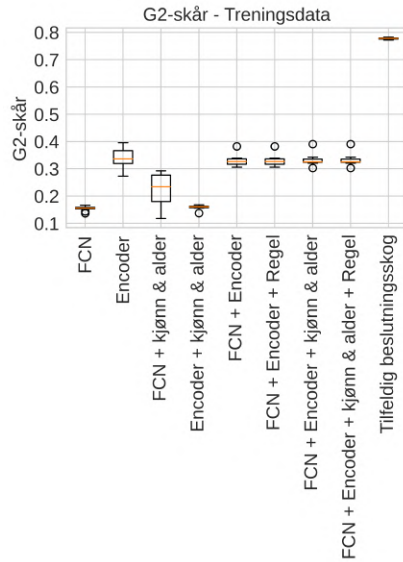
10-ganger kryssvalidert skår på treningsdataene



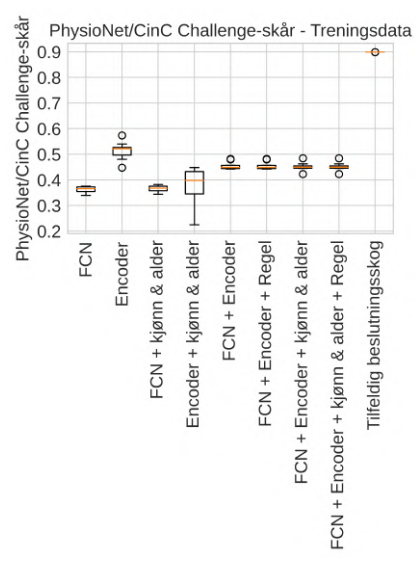
(a) F1-score



(b) F2-score



(c) G2-score



(d) PhysioNet/CinC Challenge-score

Poengskårene oppnådd av ni forskjellige modeller ved prediksjon på treningsdataene og bruk av 10 ganger kryssvalidering. Øverst til venstre viser F1-poengsum, øverst til høyre viser F2-poengsum, nederst til venstre viser G2-poengsum og nederst til høyre viser PhysioNet / CinC Challenge-poengsum.

Appendiks D

Sammenligning mellom diagnostisering av kardiologen og MarquetteTM SL12s

Kardiolog	Sinusbradykardi	5,00	1,00	0,00	1,00	3,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	
	Sinusarytmi	5,00	7,00	2,00	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Normal sinusrytme	9,00	1,00	7,00	2,00	1,00	3,00	2,00	2,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Akse mot høyre	2,00	1,00	0,00	2,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	ST-heving	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	STEMI	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	AV-blokk	2,00	1,00	1,00	1,00	0,00	0,00	3,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	RSR' eller QR-mønster i V1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Uspesifikk I.V. ledningsforsinkelse	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	IRBBB	1,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	0,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	VVH	4,00	1,00	0,00	0,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00
	Lateralt infarkt	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Venstreakseavvik	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Kan ikke utelukke inferior infarkt	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Høyre atrieforstørr.	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Mulig HVH	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Lateralt skademønster	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Anterior infarkt	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pulmonalsykdomsmønster	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Anterolateralt skademønster	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

GE Marquette SL12

Forvirringsmatrise med kardiologens diagnostisering og MarquetteTM SL12s diagnostisering.

Appendiks E

FCN

```
from tensorflow import keras

def FCN():
    inputlayer = keras.layers.Input(shape=(5000,12))

    conv1 = keras.layers.Conv1D(filters=128,
        kernel_size=8,input_shape=(5000,12),
        padding='same')(inputlayer)
    conv1 = keras.layers.BatchNormalization()(conv1)
    conv1 = keras.layers.Activation(activation='relu')(conv1)

    conv2 = keras.layers.Conv1D(filters=256,
        kernel_size=5, padding='same')(conv1)
    conv2 = keras.layers.BatchNormalization()(conv2)
    conv2 = keras.layers.Activation('relu')(conv2)

    conv3 = keras.layers.Conv1D(filters=128,
        kernel_size=3,padding='same')(conv2)
    conv3 = keras.layers.BatchNormalization()(conv3)
    conv3 = keras.layers.Activation('relu')(conv3)

    gap_layer = keras.layers.GlobalAveragePooling1D()(conv3)

    outputlayer = keras.layers.Dense(27, activation='sigmoid')(gap_layer)

    model = keras.Model(inputs=inputlayer, outputs=outputlayer)
```

Encoder

```
from tensorflow import keras
import tensorflow_addons as tfa

def encoder_model():
    input_layer = keras.layers.Input(shape=(5000, 12))
    # conv block 1
    conv1 = keras.layers.Conv1D(filters=128,
                                  kernel_size=5, padding='same')(input_layer)
    conv1 = tfa.layers.InstanceNormalization()(conv1)
    conv1 = keras.layers.PReLU(shared_axes=[1])(conv1)
    conv1 = keras.layers.Dropout(rate=0.2)(conv1)
    conv1 = keras.layers.MaxPooling1D(pool_size=2)(conv1)
    # conv block 2
    conv2 = keras.layers.Conv1D(filters=256,
                                  kernel_size=11, padding='same')(conv1)
    conv2 = tfa.layers.InstanceNormalization()(conv2)
    conv2 = keras.layers.PReLU(shared_axes=[1])(conv2)
    conv2 = keras.layers.Dropout(rate=0.2)(conv2)
    conv2 = keras.layers.MaxPooling1D(pool_size=2)(conv2)
    # Final layers
    conv3 = keras.layers.Conv1D(filters=512,
                                  kernel_size=21, padding='same')(conv2)
    conv3 = tfa.layers.InstanceNormalization()(conv3)
    conv3 = keras.layers.PReLU(shared_axes=[1])(conv3)
    conv3 = keras.layers.Dropout(rate=0.2)(conv3)
    # split for attention
    attention_data = keras.layers.Lambda(
        lambda x: x[:, :, :256])(conv3)
    attention_softmax = keras.layers.Lambda(
        lambda x: x[:, :, 256:])(conv3)
    # attention mechanism
    attention_softmax = keras.layers.Softmax()(attention_softmax)
    multiply_layer = keras.layers.Multiply()(
        [attention_softmax, attention_data])

    dense_layer = keras.layers.Dense(units=256,
                                       activation='sigmoid')(multiply_layer)
    dense_layer = tfa.layers.InstanceNormalization()(dense_layer)
    # output layer
    flatten_layer = keras.layers.Flatten()(dense_layer)
    output_layer = keras.layers.Dense(units=27,
                                       activation='sigmoid')(flatten_layer)

    model = keras.models.Model(inputs=input_layer,
                                outputs=output_layer)
```

FCN + kjønn og alder

```
from tensorflow import keras

def FCN_demo():
    inputA = keras.layers.Input(shape=(5000,12))
    inputB = keras.layers.Input(shape=(2,))

    conv1 = keras.layers.Conv1D(filters=128,
                                  kernel_size=8,input_shape=(5000,12),
                                  padding='same')(inputA)
    conv1 = keras.layers.BatchNormalization()(conv1)
    conv1 = keras.layers.Activation(activation='relu')(conv1)

    conv2 = keras.layers.Conv1D(filters=256,
                                  kernel_size=5, padding='same')(conv1)
    conv2 = keras.layers.BatchNormalization()(conv2)
    conv2 = keras.layers.Activation('relu')(conv2)

    conv3 = keras.layers.Conv1D(128,
                                  kernel_size=3,padding='same')(conv2)
    conv3 = keras.layers.BatchNormalization()(conv3)
    conv3 = keras.layers.Activation('relu')(conv3)

    gap_layer = keras.layers.GlobalAveragePooling1D()(conv3)
    model1 = keras.Model(inputs=inputA, outputs=gap_layer)
    # MLP model
    mod3 = keras.layers.Dense(50, activation="relu")(inputB)
    mod3 = keras.layers.Dense(2, activation="sigmoid")(mod3)
    model3 = keras.Model(inputs=inputB, outputs=mod3)
    # Combining MLP and FCN
    combined = keras.layers.concatenate([model1.output, model3.output])
    final_layer = keras.layers.Dense(27, activation="sigmoid")(combined)
    model = keras.models.Model(inputs=[inputA,inputB], outputs=final_layer)
```

Encoder + kjønn og alder

```
from tensorflow import keras
import tensorflow_addons as tfa

def encoder_model_demo():
    inputA = keras.layers.Input(shape=(5000, 12))
    inputB = keras.layers.Input(shape=(2,))
    # conv block 1
    conv1 = keras.layers.Conv1D(filters=128,
                                  kernel_size=5, padding='same')(inputA)
    conv1 = tfa.layers.InstanceNormalization()(conv1)
    conv1 = keras.layers.PReLU(shared_axes=[1])(conv1)
    conv1 = keras.layers.Dropout(rate=0.2)(conv1)
    conv1 = keras.layers.MaxPooling1D(pool_size=2)(conv1)
    # conv block 2
    conv2 = keras.layers.Conv1D(filters=256,
                                  kernel_size=11, padding='same')(conv1)
    conv2 = tfa.layers.InstanceNormalization()(conv2)
    conv2 = keras.layers.PReLU(shared_axes=[1])(conv2)
    conv2 = keras.layers.Dropout(rate=0.2)(conv2)
    conv2 = keras.layers.MaxPooling1D(pool_size=2)(conv2)
    conv3 = keras.layers.Conv1D(filters=512,
                                  kernel_size=21, padding='same')(conv2)
    conv3 = tfa.layers.InstanceNormalization()(conv3)
    conv3 = keras.layers.PReLU(shared_axes=[1])(conv3)
    conv3 = keras.layers.Dropout(rate=0.2)(conv3)
    # split for attention
    attention_data = keras.layers.Lambda(
        lambda x: x[:, :, :256])(conv3)
    attention_softmax = keras.layers.Lambda(
        lambda x: x[:, :, 256:])(conv3)
    # attention mechanism
    attention_softmax = keras.layers.Softmax()(attention_softmax)
    multiply_layer = keras.layers.Multiply()(
        [attention_softmax, attention_data]
    )
    # last layer
    dense_layer = keras.layers.Dense(units=256,
                                       activation='sigmoid')(multiply_layer)
    dense_layer = tfa.layers.InstanceNormalization()(dense_layer)
    # output layer
    output_layer = keras.layers.Flatten()(dense_layer)
    mod1 = keras.Model(inputs=inputA, outputs=output_layer)
    mod2 = keras.layers.Dense(50, activation="relu")(inputB)
    mod2 = keras.layers.Dense(2, activation="sigmoid")(mod2)
    mod2 = keras.models.Model(inputs=inputB, outputs=mod2)
    combined = keras.layers.concatenate([mod1.output, mod2.output])
    z = keras.layers.Dense(27, activation="sigmoid")(combined)
    model = keras.models.Model(inputs=[mod1.input, mod2.input], outputs=z)
```

Encoder + FCN

```
from tensorflow import keras
import tensorflow_addons as tfa

def FCN_Encoder():
    inputA = tf.keras.layers.Input(shape=(5000,12))
    conv1 = keras.layers.Conv1D(filters=128,
        kernel_size=8,input_shape=(5000,12), padding='same')(inputA)
    conv1 = keras.layers.BatchNormalization()(conv1)
    conv1 = keras.layers.Activation(activation='relu')(conv1)
    conv2 = keras.layers.Conv1D(filters=256,
        kernel_size=5, padding='same')(conv1)
    conv2 = keras.layers.BatchNormalization()(conv2)
    conv2 = keras.layers.Activation('relu')(conv2)
    conv3 = keras.layers.Conv1D(
        128, kernel_size=3,padding='same')(conv2)
    conv3 = keras.layers.BatchNormalization()(conv3)
    conv3 = keras.layers.Activation('relu')(conv3)
    gap_layer = keras.layers.GlobalAveragePooling1D()(conv3)
    model1 = keras.Model(inputs=inputA, outputs=gap_layer)

    conv1 = keras.layers.Conv1D(filters=128,
        kernel_size=5,padding='same')(inputA)
    conv1 = tfa.layers.InstanceNormalization()(conv1)
    conv1 = keras.layers.PReLU(shared_axes=[1])(conv1)
    conv1 = keras.layers.Dropout(rate=0.2)(conv1)
    conv1 = keras.layers.MaxPooling1D(pool_size=2)(conv1)
    # conv block -2
    conv2 = keras.layers.Conv1D(filters=256,
        kernel_size=11,padding='same')(conv1)
    conv2 = tfa.layers.InstanceNormalization()(conv2)
    conv2 = keras.layers.PReLU(shared_axes=[1])(conv2)
    conv2 = keras.layers.Dropout(rate=0.2)(conv2)
    conv2 = keras.layers.MaxPooling1D(pool_size=2)(conv2)
    # conv block -3
    conv3 = keras.layers.Conv1D(filters=512,
        kernel_size=21,padding='same')(conv2)
    conv3 = tfa.layers.InstanceNormalization()(conv3)
    conv3 = keras.layers.PReLU(shared_axes=[1])(conv3)
    conv3 = keras.layers.Dropout(rate=0.2)(conv3)
    # split for attention
    attention_data = keras.layers.Lambda(lambda x: x[:, :, :256])(conv3)
    attention_softmax = keras.layers.Lambda(lambda x: x[:, :, 256:])(conv3)
    # attention mechanism
    attention_softmax = keras.layers.Softmax()(attention_softmax)
    multiply_layer = keras.layers.Multiply()(
        [attention_softmax,attention_data]
    )
    # last layer
```

```
dense_layer = keras.layers.Dense(units=256,  
                                  activation='sigmoid')(multiply_layer)  
dense_layer = tf.keras.layers.InstanceNormalization()(dense_layer)  
# output layer  
flatten_layer = keras.layers.Flatten()(dense_layer)  
model2 = keras.Model(inputs=inputA, outputs=flatten_layer)  
combined = keras.layers.concatenate([model1.output, model2.output])  
final_layer = keras.layers.Dense(27, activation="sigmoid")(combined)  
model = keras.models.Model(inputs=inputA, outputs=final_layer)
```

Encoder + FCN + Alder og kjønn

```
from tensorflow import keras
import tensorflow_addons as tfa

def FCN_Encoder_demo():
    inputA = keras.layers.Input(shape=(5000,12))
    inputB = keras.layers.Input(shape=(2,))
    conv1 = keras.layers.Conv1D(filters=128,
        kernel_size=8,input_shape=(5000,12), padding='same')(inputA)
    conv1 = keras.layers.BatchNormalization()(conv1)
    conv1 = keras.layers.Activation(activation='relu')(conv1)
    conv2 = keras.layers.Conv1D(filters=256,
        kernel_size=5, padding='same')(conv1)
    conv2 = keras.layers.BatchNormalization()(conv2)
    conv2 = keras.layers.Activation('relu')(conv2)
    conv3 = keras.layers.Conv1D(128,
        kernel_size=3,padding='same')(conv2)
    conv3 = keras.layers.BatchNormalization()(conv3)
    conv3 = keras.layers.Activation('relu')(conv3)
    gap_layer = keras.layers.GlobalAveragePooling1D()(conv3)
    model1 = keras.Model(inputs=inputA, outputs=gap_layer)
    conv1 = keras.layers.Conv1D(filters=128,
        kernel_size=5,padding='same')(inputA)
    conv1 = tfa.layers.InstanceNormalization()(conv1)
    conv1 = keras.layers.PReLU(shared_axes=[1])(conv1)
    conv1 = keras.layers.Dropout(rate=0.2)(conv1)
    conv1 = keras.layers.MaxPooling1D(pool_size=2)(conv1)
    # conv block -2
    conv2 = keras.layers.Conv1D(filters=256,
        kernel_size=11,padding='same')(conv1)
    conv2 = tfa.layers.InstanceNormalization()(conv2)
    conv2 = keras.layers.PReLU(shared_axes=[1])(conv2)
    conv2 = keras.layers.Dropout(rate=0.2)(conv2)
    conv2 = keras.layers.MaxPooling1D(pool_size=2)(conv2)
    # conv block -3
    conv3 = keras.layers.Conv1D(filters=512,
        kernel_size=21,padding='same')(conv2)
    conv3 = tfa.layers.InstanceNormalization()(conv3)
    conv3 = keras.layers.PReLU(shared_axes=[1])(conv3)
    conv3 = keras.layers.Dropout(rate=0.2)(conv3)
    # split for attention
    attention_data = keras.layers.Lambda(lambda x: x[:, :, :256])(conv3)
    attention_softmax = keras.layers.Lambda(lambda x: x[:, :, 256: ])(conv3)
    # attention mechanism
    attention_softmax = keras.layers.Softmax()(attention_softmax)
    multiply_layer = keras.layers.Multiply()(
        [attention_softmax,attention_data]
    )
    # last layer
```

```
dense_layer = keras.layers.Dense(
    units=256, activation='sigmoid')(multiply_layer)
dense_layer = tf.keras.layers.InstanceNormalization()(dense_layer)
# output layer
flatten_layer = keras.layers.Flatten()(dense_layer)
model2 = keras.Model(inputs=inputA, outputs=flatten_layer)
mod3 = keras.layers.Dense(50, activation="relu")(inputB)
mod3 = keras.layers.Dense(2, activation="sigmoid")(mod3)
model3 = keras.Model(inputs=inputB, outputs=mod3)
combined = keras.layers.concatenate([model1.output,
    model2.output, model3.output])
final_layer = keras.layers.Dense(27,
    activation="sigmoid")(combined)
model = keras.models.Model(inputs=[inputA, inputB],
    outputs=final_layer)
```


Ensemble modell

```
from skmultilearn.problem_transform import ClassifierChain
from skmultilearn.ensemble import LabelSpacePartitioningClassifier
from skmultilearn.cluster import FixedLabelSpaceClusterer
from sklearn.ensemble import RandomForestClassifier

print("Making the clusters")
my_cluster = []
for i in range(len(y_train.T)):
    my_cluster.append(np.unique(
        np.where(y_train[np.where(y_train.T[i]==1)])[1])
    )
print("Making the model")
classifier = LabelSpacePartitioningClassifier(
    classifier = ClassifierChain(
        classifier= RandomForestClassifier(
            n_jobs=-1,n_estimators=5, verbose=0
        ),
        require_dense = [False, True]
    ),
    require_dense = [True, True],
    clusterer = FixedLabelSpaceClusterer(clusters=my_cluster)
)
```

Regelbasert modell

```
def DetectRWithPanTompkins (signal, signal_freq):
    '''signal=ECG signal (type=np.array), signal_freq=sample frequency'''
    lowcut = 5.0
    highcut = 15.0
    filter_order = 2

    nyquist_freq = 0.5 * signal_freq
    low = lowcut / nyquist_freq
    high = highcut / nyquist_freq

    b, a = butter(filter_order, [low, high], btype="band")
    y = lfilter(b, a, signal)

    diff_y=np.ediff1d(y)
    squared_diff_y=diff_y**2

    integrated_squared_diff_y =np.convolve(squared_diff_y,np.ones(5))

    normalized = (integrated_squared_diff_y-min(integrated_squared_diff_y))/
        (max(integrated_squared_diff_y)-min(integrated_squared_diff_y))

    peaks, metadata = find_peaks(normalized,
        distance=signal_freq/5 ,
        height=0.5,
        width=0.5
    )

    return peaks

HeartRate = []
TimeBetweenBeat= []
for index, item in enumerate(r_time,-1):
    HeartRate.append(60/((r_time[index+1]-r_time[index])/sampfreq))
    TimeBetweenBeat.append((r_time[index+1]-r_time[index])/sampfreq)
del HeartRate[0]
avgHr = sum(HeartRate)/len(HeartRate)
TimeBetweenBeat= np.asarray(TimeBetweenBeat)
TimeBetweenBeat=TimeBetweenBeat * 1000 # sec to ms
TimeBetweenBeat = TimeBetweenBeat[1:] # remove first element
return TimeBetweenBeat, avgHr

'''signal = ECG signal, peaks = uncorrected R peaks'''
peaks_corrected, metadata = find_peaks(signal,
    distance=min(np.diff(peaks)))
return peaks_corrected
```

```

for i in range(len(val_data)):
    data , header_data = load_challenge_data(
        ecgfilenames[val_data[i]]
    )

    avg_hr = 0
    peaks = 0
    rmssd = 0
    qrs_voltage = 0
    try:
        peaks = DetectRWithPanTompkins(
            data[1],int(header_data[0].split()[2])
        )

        try:
            peaks = R_correction(data[1], peaks)
        except:
            print("Did not manage to do R_correction")
    except:
        print("Did not manage to find any peaks using Pan Tomkins")
    try:
        rr_interval, avg_hr = heartrate(
            peaks,int(header_data[0].split()[2])
        )
        try:
            rmssd = np.mean(np.square(np.diff(rr_interval)))
        except:
            print("did not manage to comp rmssd")
    except:
        print("not able to calculate heart rate")
        rr_interval = 0
        avg_hr = 0
    try:
        qrs_voltage = np.mean(data[1][peaks])
    except:
        print("Could not calculate mean QRS peak voltage")
if avg_hr != 0:
    # bare gjør disse endringene dersom vi klarer å beregne puls
    if 60 < avg_hr < 100:
        dnn_prediction[i][16] = 0
        dnn_prediction[i][14] = 0
        dnn_prediction[i][13] = 0
    elif avg_hr < 60 & dnn_prediction[i][15] == 1:
        dnn_prediction[i][13] = 1
    elif avg_hr < 60 & dnn_prediction[i][15] == 0:
        dnn_prediction[i][14] = 1
    elif avg_hr > 100:
        dnn_prediction[i][16] = 1

```

Appendiks F

Beregninger - makro-skår

Marquette SL12 vs Kardiolog (idrettskardiologiske retningslinjer)

Normal:

$$SP = 13, SN = 2, FP = 0, FN = 13$$

$$Presisjon = \frac{13}{13 + 0} = 1.0$$

$$Sensitivitet = \frac{13}{13 + 13} = 0.5$$

$$Spesifisitet = \frac{2}{2 + 0} = 1.0$$

$$F_{1normal} - skår = 2 \cdot \frac{1 \cdot 0.5}{1 + 0.5} = 0.667$$

$$F_{2normal} - skår = (1 + 2^2) \cdot \frac{1 \cdot 0.5}{0.5 + 1 \cdot 2^2} = 0.55$$

$$G_{2normal} - skår = \frac{13}{13 + 0 + 2 \cdot 13} = 0.33$$

I grenseområdet:

$$SP = 0, SN = 19, FP = 7, FN = 2$$

$$Presisjon = \frac{0}{0 + 7} = 0$$

$$Sensitivitet = \frac{0}{0 + 2} = 0$$

$$Spesifisitet = \frac{19}{19 + 7} = 0.73$$

$$F_{1grenseområdet} - skår = 0 = 2 \cdot \frac{0 \cdot 0}{0 + 0} = 0$$

$$F_{2grenseområdet} - skår = (1 + 2^2) \cdot \frac{0 \cdot 0}{0 + 0 \cdot 2^2} = 0$$

$$G_{2grenseområdet} - skår = \frac{0}{0 + 7 + 2 \cdot 2} = 0$$

Unormal:

$$SP = 0, SN = 20, FP = 8, FN = 0$$

$$Presisjon = \frac{0}{0+8} =$$

$$Sensitivitet = \frac{0}{0+0} = 0$$

$$Spesifisitet = \frac{20}{20+8} = 0.71$$

$$F_{1_{unormal}} - skår = 2 \cdot \frac{0 \cdot 0}{0+0} = 0$$

$$F_{2_{unormal}} - skår = (1 + 2^2) \cdot \frac{0 \cdot 0}{0+0 \cdot 2^2} = 0$$

$$G_{2_{unormal}} - skår = \frac{0}{0+8+2 \cdot 0} = 0$$

Makro-skår:

$$F_{1_{makro}} = \frac{0.667 + 0 + 0}{3} = 0.22$$

$$F_{2_{makro}} = \frac{0.55 + 0 + 0}{3} = 0.18$$

$$G_{2_{makro}} = \frac{0.33 + 0 + 0}{3} = 0.11$$

Marquette SL12 vs Kardiolog (ikke-idrettskardiologiske retningslinjer)**Normal:**

$$SP = 10, SN = 7, FP = 3, FN = 8$$

$$Presisjon = \frac{10}{10+3} = 0.77$$

$$Spesifisitet = \frac{7}{7+3} = 0.70$$

$$Sensitivitet = \frac{10}{10+8} = 0.56$$

$$F_{1_{normal}} - skår = 2 \cdot \frac{0.77 \cdot 0.56}{0.77 + 0.56} = 0.65$$

$$F_{2_{normal}} - skår = (1 + 2^2) \cdot \frac{0.77 \cdot 0.56}{0.56 + 0.77 \cdot 2^2} = 0.59$$

$$G_{2_{normal}} - skår = \frac{10}{10+3+2 \cdot 8} = 0.34$$

I grenseområdet:

$$SP = 2, SN = 14, FP = 5, FN = 7$$

$$Presisjon = \frac{2}{2+5} = 0.29$$

$$Spesifisitet = \frac{14}{14+5} = 0.74$$

$$Sensitivitet = \frac{2}{2+7} = 0.22$$

$$F_{1_{grenseområdet}} - skår = 2 \cdot \frac{0.29 \cdot 0.22}{0.29 + 0.22} = 0.25$$

$$F_{2_{grenseområdet}} - skår = (1 + 2^2) \cdot \frac{0.29 \cdot 0.22}{0.22 + 0.29 \cdot 2^2} = 0.23$$

$$G_{2_{grenseområdet}} - skår = \frac{2}{2+5+2 \cdot 7} = 0.10$$

Unormal:

$$SP = 1, SN = 20, FP = 7, FN = 0$$

$$Presisjon = \frac{1}{1+7} = 0.13$$

$$Spesifisitet = \frac{20}{20+7} = 0.74$$

$$Sensitivitet = \frac{1}{1+0} = 1$$

$$F_{1_{unormal}} - skår = 2 \cdot \frac{0.13 \cdot 0}{0.13 + 1} = 0.23$$

$$F_{2_{unormal}} - skår = (1 + 2^2) \cdot \frac{0.13 \cdot 1}{1 + (0.13 \cdot 2^2)} = 0.43$$

$$G_{2_{unormal}} - skår = \frac{1}{1+7+2 \cdot 0} = 0.13$$

Makro-skår:

$$F_{1_{makro}} = \frac{0.65 + 0.25 + 0.23}{3} = 0.38$$

$$F_{2_{makro}} = \frac{0.59 + 0.23 + 0.43}{3} = 0.42$$

$$G_{2_{makro}} = \frac{0.34 + 0.10 + 0.13}{3} = 0.19$$

Appendiks G

Convolutional Neural Network and Rule-Based Algorithms for Classifying 12-lead ECGs

Bjørn-Jostein Singstad¹, Christian Tronstad²

¹University of Oslo, Oslo, Norway

²Oslo University Hospital, Oslo, Norway

Abstract

The objective of this study was to classify 27 cardiac abnormalities based on a data set of 43101 ECG recordings. A hybrid model combining a rule-based algorithm with different deep learning architectures was developed.

We compared two different Convolutional Neural Networks, a Fully Convolutional Neural Network and an Encoder Network, a combination of both, and with the addition of another neural network using age and gender as input. Two of these combinations were finally combined with a rule-based model using derived ECG features. The performance of the models was evaluated on validation data during model development using hold-out validation. Finally, the models were deployed to a Docker image, trained on the provided development data, and tested on the Challenge validation set. The model that performed best on the Challenge validation set was then deployed and tested on the full Challenge test set. The performance was evaluated based on a particular Challenge score.

Our team, TeamUIO, achieved a Challenge validation score of 0.377, and a full test score of 0.206 for our best model. The score on the full test set placed us at 20th out of 41 teams in the official ranking.

1. Introduction

The ECG reflects the electrical activity of the heart, and the interpretation of this recording can reveal numerous pathologies of the heart. An ECG is recorded using an electrocardiograph, where modern clinical devices usually contain automatic interpretation software that interprets the ECGs directly after recording. Although automatic ECG interpretation started in the 1950s, there are still some limitations [1, 2]. Because of the errors done by the automatic interpretation software, doctors have to read over the ECGs [3]. This is time-consuming for the doctors and requires a high degree of expertise [4]. There is clearly a need for better ECG interpretation algorithms.

Recent years have shown a rapid improvement in the

field of machine learning. A sub-field of machine learning is called deep learning, where more complex architectures of neural networks are better able to scale with the amount of data in terms of performance. This type of machine learning has shown promising performance in many fields including medicine, and in this study, we have explored the usefulness of deep learning in classifying 12-lead ECGs.

As a starting point for our model architecture, we chose to use the two best performing Convolutional Neural Networks (CNN) used on ECG data in Fawaz HI et al 2019 [5]. They reported that Fully Convolutional Neural Networks (FCN) outperformed eight other CNN architectures compared. We also wanted to test the second-best architecture from their study which was an Encoder network. Finally, we assessed the integration of a rule-based algorithm within these models to test the performance of a CNN and rule-based hybrid classifier.

This study is a part of the PhysioNet/Computing in Cardiology (CinC) Challenge 2020, where the aim was to develop an automated interpretation algorithm for the identification of multiple clinical diagnoses from 12-lead ECG recordings.

2. Methods

2.1. Data

To train the CNN models a data set containing 43.101 ECG recordings with corresponding information files describing the recording, patient attributes, and the diagnosis was used [6, 7]. The recording lengths varied across the different ECG signals, 83.4% were 5000 samples long. 98.5% of the recordings were sampled at a frequency of 500Hz, 1.3% signals sampled at 1kHz and 0.2% signals sampled at 257Hz.

2.2. Preprocessing

According to the goal of this Challenge, we aimed to classify 27 of the 111 diagnoses [6]. The 27 labels to classify were One-Hot encoded, with each diagnosis rep-

resented as a bit in a 27-bit long array. All recordings were padded and truncated to a signal length of 5000 samples. Padding and truncation were done by removing any parts longer than 5000 samples and adding a tail of $5000 - n$ zeros to any recording of length $n < 5000$.

2.3. CNN architectures

As a starting point for classifying the ECG-signals, we employed FCN and Encoder types of CNN models as described in Fawaz HI et al 2019 [5]. Two models were tested without any modifications to the architecture other than changing the input and output layers to fit our input data and output classes. All output layers of each model used a Sigmoid activation function.

To make use of the provided age and gender data, a simpler neural network model with 2 inputs, one hidden layer of 50 units, and 2 outputs in the final layer was added. This new model was combined with our FCN and Encoder models by concatenation of the last layer of the CNNs.

Age and gender data were passed into the simple neural network as integers, but in some information files, the age of the patient was not given and was assigned a value of -1. The gender data was transformed into integers, where a male was set equal to 0, female equal to 1, and unknown was set to 2.

The two CNN models (FCN and Encoder) were combined as parallel models, concatenated on the second last layer. This model was tested with and without a parallel dense layer¹.

2.4. Rule-based model

The rule-based algorithm used the raw ECG signal, without any padding or truncating, as input. R-peak detection [8], and heart rate variability (HRV) analysis was programmed to add relevant derived features to the algorithm. An HRV-score was obtained by computing the root mean square of successive differences between normal heartbeats (RMSSD) using the detected R-peaks as timing indicators of each heartbeat.

The rule-based algorithm was able to classify eight different diagnoses: atrial fibrillation, bradycardia, low QRS-complex, normal sinus rhythm, pacing rhythm, sinus arrhythmia, sinus bradycardia, and sinus tachycardia.

The rule-based algorithm performed classification independent of the deep learning models. If there was disagreement between the rule-based algorithm and the CNN model, the rule-based algorithm overwrote the classification from the CNN model.

¹All models and algorithms are available here: <https://www.kaggle.com/bjoernjostein/physionet-challenge-2020>

2.5. Model development

The models were trained and validated on the development data using hold-out validation with a split of 90% for training and 10% for validation. The first fold in a stratified K-fold was used with a random seed of 42 [9]. The splitting was arranged such that the distribution of diagnoses was the same in both the train and validation data.

During training, the Area Under the Curve (AUC) score on the validation set was used to determine if the learning rate should drop or stay. The learning rate was initially set to 0.001 for all models and decreased by a factor of 10, using the reduce on plateau method [10], for each epoch that the AUC score did not improve. Early stopping [10] was triggered when the AUC score on the validation data did not improve over two successive epochs.

2.6. Threshold optimization

The prediction thresholds were optimized during model development. This was done by running the classifier on the hold-out validation data and receiving a score between 0 and 1 for each of the classes. The Nelder-Mead downhill simplex method [11, 12] was applied to optimize the threshold individually for the 27 classes. The Nelder-Mead downhill simplex method is used to find the local minimum of a function using the function itself and an initial guess of the variable of the function. The 27-element long array was optimized using the negative of the PhysioNet/CinC Challenge score [6]. To increase the possibility of finding the global maximum of the PhysioNet/CinC Challenge score, all elements in the 27-element long array was given a value of 1 and multiplied it with a variable that was given values from 0 to 1, with a step size of 0.05. The value that gave the highest PhysioNet/CinC Challenge score was used as the initial guess for the Nelder-Mead downhill simplex method.

2.7. Model deployment

To obtain a valid score in the PhysioNet/CinC Challenge we submitted the models to the PhysioNet/CinC committee for testing on a Challenge validation and test set [6].

A Docker image was used to create a virtual Python environment for the model to be tested. During model deployment, the model was trained on the whole development set. The first three Challenge validation scores were obtained using AUC on the development data to schedule the reduction of the learning rate.

The two last Challenge validation scores were obtained using a learning rate scheduler. The learning rate schedule was programmed to be the same as in model development.

Model ID and name	Rule-based model	AUC	F1	F2	G2	Challenge score
A) FCN	No	0.875	0.381	0.446	0.230	0.348
B) Encoder	No	0.866	0.396	0.429	0.228	0.398
C) FCN + age, gender	No	0.877	0.368	0.438	0.222	0.385
D) Encoder + age, gender	No	0.828	0.334	0.389	0.190	0.333
E) Encoder + FCN	No	0.872	0.399	0.436	0.237	0.409
F) Encoder + FCN	Yes	0.872	0.361	0.413	0.203	0.348
G) Encoder + FCN + age, gender	No	0.866	0.400	0.434	0.233	0.395
H) Encoder + FCN + age, gender	Yes	0.866	0.356	0.405	0.198	0.338

Table 1. Scores were obtained by eight different models during model development. The models were evaluated by five different metrics, AUC, F1, F2, G2, and the PhysioNet/CinC Challenge score, during model development.

2.8. General parameters for both validation and testing procedures

For all models in both development and deployment, we used Adam optimizer, a batch size of 30, and binary cross-entropy as the loss function. A batch generator was used to feed the model with data during training, programmed to shuffle the order of data for each epoch.

Weights based on the number of occurrences of the different classes were calculated to deal with the skewed classes in the development data [9]. The calculated weights were passed to the model during training to give higher priority to rare diagnoses and lower priority to diagnoses that occurred more frequently.

3. Results

3.1. Scoring metrics

During model development, all models were validated on a subset of the development data using the metrics AUC (Eq 1), F_1 -score (Eq 2), F_2 -score (Eq 3), G_2 -score (Eq 4), and the PhysioNet/CinC Challenge score, as seen in Table 1. On the Challenge validation set, we only obtained the PhysioNet/CinC Challenge score as seen in Table 2. After the evaluation of the performance on the full Challenge test set we were provided AUC (Eq 1), F_1 -score (Eq 2), PhysioNet/CinC Challenge score, an Area Under the Precision-Recall Curve (AUPRC) score, and an accuracy score.

$$AUC_{(t_i - t_{i-1})} = (t_i - t_{i-1}) \times \frac{f(t_i) + f(t_{i-1})}{2} \quad (\text{Eq 1})$$

$$F_1 = \frac{2 \times TP}{2 * TP + FP + FN} \quad (\text{Eq 2})$$

$$F_2 = \frac{(1 + 2^2) \times TP}{(1 + 2^2) \times TP + FP + 2^2 \times FN} \quad (\text{Eq 3})$$

$$G_2 = \frac{TP}{TP + FP + 2 \times FN} \quad (\text{Eq 4})$$

3.2. Classification performance

Five out of the eight models tested during the development phase, as seen in Table 1, were successfully deployed and obtained a score on the Challenge validation set, presented in Table 2.

Model ID and name	Rule-based model	Challenge score
B) Encoder	No	0.229
C) FCN + age, gender	No	0.302
D) Encoder + age, gender	No	0.272
F) Encoder + FCN	Yes	0.377
H) Encoder + FCN + age, gender	Yes	0.364

Table 2. The scores are obtained on the Challenge validation set and only the PhysioNet/CinC Challenge score was given. The Challenge validation set is a subset of the Challenge test set and not the final score in the challenge. The scores achieved on the Challenge validation set was used to select one model for deployment on the full Challenge test set.

The best score on the Challenge validation set was achieved by model H, an Encoder in parallel with an FCN with the rule-based algorithms added, as seen in Table 2. Model H was finally deployed and scored on the full Challenge test set [6]. The model achieved an AUC-score of 0.728, an F_1 -score of 0.233, and a PhysioNet/CinC Challenge score of 0.206. This score brought us, TeamUIO, to 20th place in the PhysioNet/CinC Challenge 2020.

4. Discussion and conclusion

We chose to pad and truncate the signals to 5000 samples which were necessary to be able to feed the signal to

the CNN. The disadvantage was that some important information from segments of the ECG recordings might have been omitted in training the models. On the other hand, the derived features used in the rule-based implementation were based on complete recordings. Thus, the models that combined both CNN and rule-based algorithms used the entire signal when classifying the ECG.

Deployment of the models was done using two different ways of controlling the learning rate. The scores of models B, C, and D, on the Challenge validation set (Table 2), were obtained by using AUC on the development data to schedule the reduction of the learning rate. This might have contributed to overfitting indicated by the difference of the Challenge score of models B, C, and D in Table 1 compared with the same models in Table 2. The Challenge score achieved on the Challenge validation data by model F and G (Table 2), were obtained using a learning rate scheduler [10]. The PhysioNet/CinC Challenge scores achieved on the Challenge validation data by model F and G (Table 2) are more consistent with the PhysioNet/CinC Challenge score obtained on the development data in Table 1 for the same models. In summary, our result indicates that the models, deployed on the Challenge validation set, which kept the same training schedule as in the development model, seem to avoid overfitting and perform better on unseen data.

During the model development, we observed that the Encoder (model B) performed better than the FCN (model A) on the PhysioNet/CinC Challenge score as seen in Table 1. A plain FCN (model A) was not scored on the Challenge validation set and thus it remains unclear which of a plain FCN or a plain Encoder perform best on unseen data like the Challenge validation data.

The Encoder (model B) decreased in performance when a parallel model for age and gender was added (model D) during model development (Table 1). However, the performance increased when the Encoder (model B) was added a parallel model for age and gender (model D) when scoring the models on the Challenge validation set (Table 2). Based on the PhysioNet/CinC Challenge score, during model development (Table 1), the FCN (model A) improved in performance when adding a parallel model for age and gender (model C). However, we did not deploy a plain FCN (model A) to the Challenge validation set and thus it remains unclear if the FCN + age and gender (model C) would outperform the FCN (model A) on the Challenge validation set.

During model development (Table 1), the Encoder + FCN (model E) and the Encoder + FCN + age, gender (model G), decreased in performance when adding the rule-based model (model F and H). However, the PhysioNet/CinC Challenge score, achieved by model F and G on the Challenge validation set (Table 2), was better than

the PhysioNet/CinC Challenge score achieved by the same models during model development (Table 1). Our results indicate that the hybridization of CNN with a rule-based model could improve the diagnostic classification of ECG, but further analysis is needed to confirm whether, and to which extent such implementation improves the performance of the proposed CNN models.

References

- [1] Schläpfer J, Wellens HJ. Computer-Interpreted Electrocardiograms. *Journal of the American College of Cardiology* August 2017;70(9):1183–1192.
- [2] Smulyan H. The Computerized ECG: Friend and Foe. *The American Journal of Medicine* February 2019;132(2):153–160.
- [3] Alpert JS. Can You Trust a Computer to Read Your Electrocardiogram? *The American Journal of Medicine* June 2012;125(6):525–526.
- [4] Bickerton M, Pooler A. Misplaced ECG Electrodes and the Need for Continuing Training. *British Journal of Cardiac Nursing* March 2019;14(3):123–132.
- [5] Fawaz HI, et al. Deep Learning for Time Series Classification: A Review. *Data Mining and Knowledge Discovery* July 2019;33(4):917–963.
- [6] Alday EAP, Gu A, Shah A, Robichaux C, Wong AKI, Liu C, Liu F, Rad AB, Elola A, Seyedi S, Li Q, Sharma A, Clifford GD, Reyna MA. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. *Physiol Meas* 2020;(Under Review).
- [7] Goldberger AL, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* June 2000;101(23).
- [8] Pan J, Tompkins WJ. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering* March 1985;BME-32(3):230–236.
- [9] Pedregosa F, et al. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011; 12(85):2825–2830.
- [10] Martín Abadi, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.
- [11] Nelder JA, Mead R. A Simplex Method for Function Minimization. *The Computer Journal* January 1965;7(4):308–313.
- [12] Virtanen P, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* March 2020;17(3):261–272.

Address for correspondence:

Bjørn-Jostein Singstad
Sem Sælands vei 24, 0371 Oslo, Norway
bj.singstad@fys.uio.no

Convolutional Neural Network and rule-based algorithms for classifying 12-lead ECGs

Bjørn-Jostein Singstad¹ Christian Tronstad²

¹University of Oslo ²Oslo University Hospital

Introduction

Interpretation of 12 lead electrocardiogram (ECG) is mainly done by doctors and rule-based algorithms. Recent years advances in deep learning and the growing number of digitally available ECGs has opened up possibilities for improving automated interpretation using deep learning. In this study we employed three different deep learning architectures based on *H. Ismail Fawaz et al 2019* for classification of 27 diagnostic labels in the Physionet Challenge 2020 dataset.cipy optimization

Method

A Fully convolutional neural network (FCN), a Residual network (ResNet), an Encoder architecture and a ResNet with separable layers were implemented with model selection and optimization according to figure 1. The development data set was split into training and validation data. The validation data was used for hyperparameter tuning and model selection. A rule-based algorithm based on derived ECG parameters were included in in a hybrid model. Finally the thresholds for each class where optimized individually using Nelder-Mead simplex algorithm (Nelder, J.A. and Mead, R. 1965).

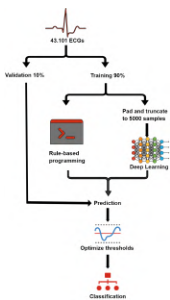


Figure 1: flow diagram for optimizing

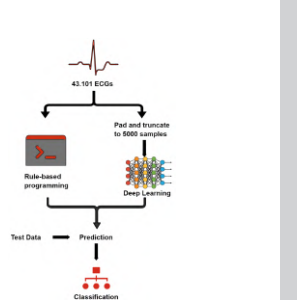


Figure 2: flow diagram for final model and selecting models

The final model, used for classification of the test set, was trained as described in figure 2.

Results

Ten different models were trained and validated, of which five were tested on the test set giving the scores presented in Table 1

Table 1: Validation and test results for all models. The metric used for scoring is the Physionet Challenge 2020 scoring metrics

Model	Rule-based model	Val score	Test score
ResNet	No	0.277	-
FCN	No	0.348	-
Encoder	No	0.398	0.229
ResNet DNN	No	0.221	-
FCN DNN	No	0.385	0.302
Encoder DNN	No	0.333	0.272
Encoder FCN	No	0.409	-
Encoder FCN	Yes	0.348	0.377
Encoder FCN DNN	No	0.395	-
Encoder FCN DNN	Yes	0.338	0.364
ResNet revised	No	0.410	-
ResNet sep.conv/conv	No	0.451	-

The best performing model was the ResNet with separable layers. Figure 3 shows the confusion matrix based on the performance on the validation data using the ResNet with separable layers.

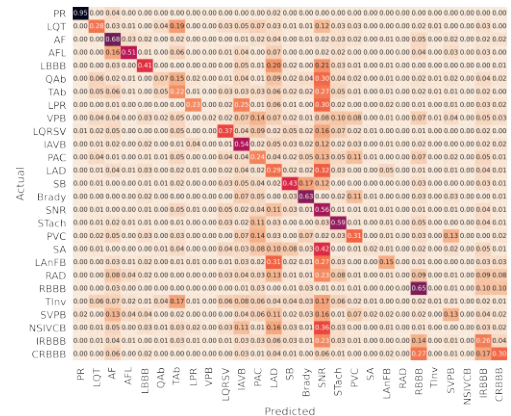


Figure 3: Confusion matrix for all diagnoses classified by the ResNet with separable convolutional layers

Conclusions

A parallel dense model, with age and gender as input, improved the performance of the model in 1 out of 3 cases. The implementation of a rule-based model decreased the performance of the models on the validation set, but there were better agreement between test and validation set. Separable convolutional layers in combination with normal convolutional layers seems to improve the performance of the ResNet when classifying 12-lead ECGs.

