



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2021 30 stp
Fakultet for realfag og teknologi

Klassifisering og filtrering av Sentinel-3 PEACHI-returpulser ved bruk av maskinlæring

Classification and filtering of Sentinel-3
PEACHI-return pulses using machine learning

Alf Georg Øvland
Institutt for Geomatikk

Forord

Jeg vil begynne med å takke mine veiledere Kristian Breili og Vegard Ophaug for ressurser og støtte i altimetrifagfeltet, og i skrivingen av oppgaven. Jeg vil også takke Oliver Tomic for veiledning i opparbeidelse av maskinlæringsmodeller. I tillegg vil jeg takke AVISO for problemfri tilgang til altimetridata. I programmeringsarbeidet har Python, SKlearn, Tensorflow vært gode verktøy. For visualisering og preprosessering har Geopandas og Openstreetmaps vært svært nyttige. Ellers vil jeg takke alle mine medstudenter på universitetet for god støtte, både faglig og sosialt.

Ås, 1. Juni 2021

Alf Georg Øvland

Sammendrag

I denne gradsoppgaven utforskes potensialet til Sentinel-3 PEACHI-data langs Norskekysten. To filtreringsmetoder blir utprøvd for å fjerne observasjoner som forventes å gi feil avstandsmåling, hovedsakelig forårsaket av landkontaminering og ulike refleksjonsegenskaper på havoverflata. Et kystsnitt blir beregnet for å finne den beste filtreringsmetoden for hele kysten. Den ene filtreringsmetoden baserer seg på grovfeilsøk på avstandsmålingen med Student's multippel-t test. Den andre metoden baserer seg på klassifisering av returpulsens form ved bruk av maskinlæring. For sistnevnte blir tre ulike maskinlæringsmodeller testet for å finne den beste for oppgaven. I tillegg blir en cluster-modell testet for å se om den karer å finne ønskede returpuls-egenskaper.

Maskinlæringsmodellene *Dense*, *CNN HSI* og *CNN* oppnår klassifiserings-nøyaktighet på henholdsvis 66.3%, 63.4% og 68.2%. Cluster-algoritmen K-Means fant ikke ønskede egenskaper og ble derfor ikke benyttet i videre beregninger. Etter evaluering av maskinlæringsmodellene ved bruk av forvirringsmatriser ble *CNN*-modellen valgt til å predikere pulsklasser i valideringsområder rundt vannstandsmålerne.

Filtreringsmetodene blir hver for seg validert ved dynamisk havtopografi med fasit fra permanente vannstandsmålere. Deretter sammenliknet med hverandre. Enkelte vannstandsmålere måtte utelates av kystsnittet på grunn av få observasjoner etter filtreringsmetoden som bruker maskinlæring. Student's multippel-t test gir differanse, standardavvik, *median absolute deviation* (MAD) og Spearman-korrelasjon på henholdsvis 21.6 mm, 111.8 mm og 0.8724. Klassifisering basert på returpulsens form gir differanse, standardavvik, MAD og Spearman-korrelasjon på henholdsvis 9.3 mm, 103.0 mm og 0.8786. Det tilsvarer en prosentvis endring på -57.1%, -7.9%, 0.7% og -3.4%, for henholdsvis differanse, standardavvik, Spearman-korrelasjon og MAD.

Resultatene indikerer at filtrering ved maskinlæringsmetoden egner seg bedre for kystaltimetri langs Norskekysten. Sentinel-3 SAR-observasjoner gir bedre resultater enn Sentinel-3 PEACHI-observasjoner når man sammenlikner med liknende tidligere arbeid.

Abstract

In this thesis, the potential of Sentinel-3 PEACHI data along the Norwegian coast line is explored. Two filtration methods are being tested to remove observations that are expected to give incorrect distance measurement, mainly caused by land contamination and various reflection properties of the sea surface. A coastal mean is calculated to find the best filtration method for the entire coast. One filtering method is based outlier detection, using the distance measurement with Student's multiple-t test. The second method is based on the classification of the shape of the return pulse using machine learning. For the latter, three different machine learning models are tested to find the best one for this task. In addition, a cluster model is tested to see if it can find the desired return pulse properties.

The machine learning models, *Dense*, *CNN HSI* og *CNN* achieves classification accuracy of 66.3 %, 63.4 % and 68.2 %, respectively. The cluster algorithm K-Means did not find the desired properties and was therefore not used in further calculations. After evaluating the machine learning models using confusion matrices, the *textit* CNN model was chosen to predict pulse-classes in validation areas around the water level gauges.

The filtration methods are individually validated by dynamic sea topography with results from permanent water level gauges. Then compared with each other. Some water level gauges had to be omitted from the coastal mean due to few observations according to the filtration method that uses machine learning. Student's multiple-t test gives difference, standard deviation, *textit* median absolute deviation (MAD) and Spearman correlation of 21.6 mm, 111.8 mm and 0.8724, respectively. Classification based on the shape of the return pulse gives the difference, standard deviation, MAD and Spearman correlation of 9.3 mm, 103.0 mm and 0.8786, respectively. This corresponds to a percentage change of -57.1 %, -7.9 %, 0.7 % and -3.4 %, for difference, standard deviation, Spearman correlation and MAD, respectively.

The results indicate that filtering by the machine learning method is better suited for coastal altimetry along the Norwegian coast. Sentinel-3 SAR observations give better results than Sentinel-3 PEACHI observations when compared to similar previous work.

Innhold

Figurer	vi
Tabeller	viii
1 Innledning	1
1.1 Problemstilling og aktualitet	1
1.2 Avgrensninger	2
2 Teori	4
2.1 Altimetri	4
2.1.1 Avstandsmåling - SAR	5
2.1.2 Sentinel 3	6
2.1.3 Korreksjoner	9
2.1.4 Referanserammer	11
2.1.5 Dynamisk havtopografi	11
2.2 Maskinlæring	12
2.2.1 Styrt klassifisering	12
2.2.2 Ikke-styrt klassifisering	14
2.2.3 Attributtredigering	15
2.2.4 Validering av maskinlæringsalgoritmer	17
3 Datasett og metoder	20
3.1 LR RMC	20
3.2 Kystlinje	20
3.3 Vannstandsmålere	20
3.4 Geodetiske modeller	22
3.4.1 NKG2015	22
3.4.2 HREF2018B-NKG2015	22
3.5 Maskinlæringsmodeller	22

3.6	Testområder	24
3.7	Klassifisering av returpulsene	25
3.8	Valideringsområder	28
3.9	Validering ved dynamisk havtopografi	30
4	Resultater	31
4.1	Klassifiseringsnøyaktighet	31
4.1.1	<i>Dense</i>	32
4.1.2	<i>HSI CNN</i>	33
4.1.3	<i>CNN</i>	34
4.1.4	Forvirringsmatriser	34
4.1.5	Clustering resultater	37
4.2	Dynamisk havtopografi	40
5	Diskusjon	45
6	Konklusjoner og anbefalinger	47
	Bibliografi	49
A	Kode	51
A.1	<i>Dense</i>	51
A.2	<i>HSI-CNN</i>	52
A.3	<i>CNN</i>	53
A.4	<i>K-Means</i>	54
B	Figurer	55

Figurer

2.1	Øverst vises bølgefrontens tidsutvikling på jordoverflata ved bruk av tradisjonell altimetri. Nederst vises mottatt signal fra tilhørende fotavtrykk. Figur hentet fra Ophaug [2017].	4
2.2	Forklaring av pulsens tidsforløp i SAR-modus. Hentet fra : http://www.altimetry.info/radar-altimetry-tutorial/how-altimetry-works/delay-doppler-or-sar-altimetry/	5
2.3	Tradisjonell radar altimeter(venstre) vs SAR altimeter(høyre) Hentet fra : http://www.altimetry.info/radar-altimetry-tutorial/how-altimetry-works/delay-doppler-or-sar-altimetry/	6
2.4	Nominelle bakkespor for S3 over Norge. Som følge av inklinasjonsvinkelen på 98.65 grader er det altimetriobservasjoner langs hele Norskekysten.	7
2.5	Observasjonsinstrumenter (rød) og navigasjonsinstrumenter (blå) ombord Sentinel-3 satellittene. Hentet fra: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3/overview - Sist lastet: 22/02 2021.	8
2.6	Prosesseringsmetode (øverst) og foravtrykk (nederst) for SAR (venstre) og LR-RMC (høyre). Figur hentet fra Moreau et al. [2021]	9
2.7	Oppbyggingen til nevroner i NN. Tilpasset fra Raschka and Mirjalili [2019].	12
2.8	Eksempel på NN hvor sirkler representerer nevroner og pilene representerer vektcoeffisientene. Hentet fra Raschka and Mirjalili [2019]	13
2.9	Eksempel på samlinger som enkelt kan skilles fra hverandre. Hentet fra Raschka and Mirjalili [2019].	14
2.10	Summert varians som hver prinsippal komponent forklarer. Denne grafen er produsert med S3-LR-RMC-data og er unik for datasettet.	16
2.11	Visuell beskrivelse av PCA. Her er x_1 og x_2 originale attributtakser. Hentet fra Raschka and Mirjalili [2019].	17
2.12	Holdout-metode for validering av maskinlæringsmodeller. Hentet fra Raschka and Mirjalili [2019].	18
2.13	Forvirringsmatrise for binært problem. Hentet fra Raschka and Mirjalili [2019].	18
2.14	Forvirringsmatrise for flerklasseproblem.	19
3.1	Permanente vannstandsmålere i Norge.	21
3.2	Testområder for klassifisering vist i fiolett.	24
3.3	Eksempelpulser brukt som hjelpemiddel i klassifiseringsfasen.	25
3.4	Kartutsnitt som viser hvor den aktuelle pulsen som skal klassifiseres befinner seg(lyseblå), hvilke pulser som har blitt valgt ut for klassifisering i passet(rød) og fotavtrykket til satellitten(mørkeblå). Testområde: Sognefjorden	26

3.5	Klassefordeling i datasettet	27
3.6	Klasser fra manuell klassifisering.	28
3.7	Vannstandsmålere med tilhørende valideringsområder.	29
4.1	Utvikling for <i>Dense</i> -modell.	32
4.2	Utvikling for <i>HSI CNN</i> -modell.	33
4.3	Utvikling for <i>CNN</i> -modell.	34
4.4	Forvirringsmatrise for <i>Dense</i> -modell.	35
4.5	Forvirringsmatrise for <i>CNN HSI</i> -modell.	36
4.6	Forvirringsmatrise for <i>CNN</i> -modell.	36
4.7	Prinsippal komponenter fargelagt med klasser fra manuell klassifisering.	38
4.8	Prinsippal komponenter fargelagt med klasser predikert av Kmeans.	39
4.9	Klasser basert på cluster.	40
4.10	Inkluderte observasjoner for hele tidsperioden	41
4.11	Inkluderte pass for hele tidsperioden	41
4.12	Differanse for hele tidsperioden	42
4.13	Standardavvik for hele tidsperioden	42
4.14	Spearman korrelasjon for hele tidsperioden	43
4.15	MAD per vannstandsmåler for hele tidsperioden	43

Tabeller

2.1	Observasjonsinstrumenter ombord Sentinel 3	6
3.1	Vannstandsmålere langs Norges kyst. Koder for referanse i kommende plott.	22
3.2	Havareal og forventet antall punkter per satellittsyklus. På grunn av variasjoner i satellittbanen kan antall observasjonspunkter variere.	24
3.3	Hjelpemiddel for klassifisering av pulsene.	26
4.1	Resultater fra trening av maskinlæringsmodeller på testdatasett med endelig antall epoker.	34
4.2	Kystsnitt og prosentvis endring.	44

1 Innledning

Satellittaltimetri er en jordobservasjonsteknikk som måler havets høyde over en ellipsoide ved bruk av et radarsignals gangtid. Observasjonsserier over lengre tid, gjerne kombinert med modeller eller bakkebaserte målinger, kan man utlede flere størrelser. Eksempler er sesongvariasjoner, havnivåets høyde, dynamisk havtopografi og overvåking av miljø og klima. Altimetri har siden de første dedikerte altimetrisatellittene på 1970-tallet gitt nærmest global dekning og enorme mengder data, som bidrar til forståelse av globale systemer i havene som ellers er svært vanskelig å observere med andre metoder.

1.1 Problemstilling og aktualitet

Et fundamentalt problem i kystaltimetri er å skille mellom gode og dårlige observasjoner. Problemene oppstår når altimetrisatellittene nærmer seg kysten og får retursignaler delvis fra land eller grunner. Landområder og grunne havområder har andre refleksjonsegenskaper sammenliknet med åpent hav. Når deler av retursignalet kommer fra land kalles det landkontaminering. Havet kan også ha andre refleksjonsegenskaper nærme land, spesielt i skjærgård, små havbasseng og fjorder hvor havet nærmest speiler radarsignalet. Retursignalet vil da opptre på en annen måte enn det som er forventet. På åpent hav er refleksjonsegenskapene kjent og nærmest uniformt i området bestrålt av radarpulsen til satellitten. I kystsonen endrer havets overflate seg raskt og lokalt. Tidligere har altimetriobservasjoner blitt prosessert med metoder optimalisert for åpent hav, derfor vil det i kystsonen være vanskelig å skille mellom støy og informasjon. Langs Norskekysten er det ekstra utfordrende på grunn av flere tusen øyer og lange smale fjorder.

Flere omfattende studier har blitt gjennomført langs Norskekysten for å øke presisjonen til kystaltimetri de siste årene. Breili [2021] tar for seg Sognefjorden og omkringliggende områder med SAR-data (Synthetic Aperture Radar) fra Sentinel-3. Student's multippel-t test blir brukt for å eliminere grovfeil i avstandsmålingene før dynamisk havtopografi blir beregnet. Dynamisk havtopografi fra altimetriobservasjoner blir sammenliknet med dynamisk havtopografi fra vannstandsmålere. I ytre kystsoner er standardavviket 6.9 cm med en romlig korrelasjon på 0.63. I Sognefjorden er standardavviket 10.4 cm med romlig korrelasjon på 0.29. Konklusjonen er at landkontaminering fører til dårligere resultater i Sognefjorden.

Idžanović et al. [2018] bruker nygenerasjons altimetri med SARIn (interferometric SAR) observasjoner fra CryoSat-2 for å bestemme midlet dynamisk havtopografi langs Norskekysten. Her brukes også en multippel-t test til grovfeilsøk og sammenlikning med vannstandsmålere i den ytre kystsonen, som resulterer i et standardavvik på 9 cm. For vannstandsmålere i den indre kystsonen er standardavviket 20 cm.

Felles for Breili [2021] og Idžanović et al. [2018] er at observasjonene kommer fra nygenerasjons altimetrisatellitter som opererer i en form for SAR-modus og man bruker Student's multippel-t test for å eliminere grovfeil i avstandsmålingene fra altimetrisatellittene. Konvensjonelle grovfeilsøk kan føre til at man filtrerer bort gode data eller blir sittende igjen med for mange dårlige observasjoner i det endelige datasettet. Dette gjelder særlig for kystaltimetri om altimetrisatellitten går i repeterende baner. Da vil en øy ligge på tilnærmet samme plass, relativt til satellitten, for hver overflygning satellitten gjør. Øya kan føre til et systematisk avvik som forskyver samlingen og man risikerer at mange gode observasjoner kan bli forkastet.

Et alternativ til grovfeilsøk med Student's multippel-t test er å bruke returpulsens form til å bestemme om observasjonen er god eller dårlig. En metode som egner seg godt til dette er maskinlæring. Derfor skal denne oppgaven undersøke hvordan maskinlæringsmodeller filtrerer altimetriobservasjoner i kystsonen. Maskinlæringsmodeller trent på manuelt klassifiserte returpulser vil forsøke å predikere om observasjonen kan brukes i beregning av dynamisk havtopografi. Maskinlæring har tidligere blitt benyttet til altimetri med gode resultater. Da har man fokusert på klassifisering av ulike pulsformer, og har i tillegg inkludert parametere i prosessen som beregner avstanden basert på returpulsens form (retracking) i maskinlæringsmodellen, slik at de tilpasser seg de klassifiserte pulsene.

Mattes [2019] bruker nevralt nett for å finne en optimal retracker til returpulser fra innlandsvann og elver som inneholder landkontaminering. Her blir den rå returpuls, sammen med observert vannstand fra vannstandsmålere, brukt til å trene maskinlæringsmodellen. Det nevralt nettet klarer å finne en retracker som klarer å se sesongvariasjoner og beregner vannstand med 1.04 m gjennomsnittsfel.

Dettmering et al. [2018] utforsker hvordan maskinlæring kan finne leder i havis med SAR-data fra CryoSat-2. Konklusjonen er at en cluster-algoritme introdusert av Mueller et al. [2017] fungerer best til å finne leder, med 97 prosent nøyaktighet. Valideringen blir utført med flybilder som ble tatt maks 90 minutter før eller etter satellitten passerte. Et liknende arbeid finner man i Boergens et al. [2017] hvor samme datasett skal brukes til å avlede vannstand i elvebasseng i Mekong elven etter en cluster-algoritme skiller mellom observasjoner over land og vann. Resultatene sammenliknes med observasjoner filtrert ved bruk av elvebredde fra satellittbilder. De finner at cluster-algoritmen presterer bedre enn filtrering ved satellittbilder, med henholdsvis 0.76 m og 0.83 m medianverdi for årlig avvik. For å validere resultatene bruker de vannstandsmålere.

Moreau et al. [2021] undersøker et nytt datasett kalt PEACHI, hvor en ny prosesseringsmetode har blitt benyttet behandlingen av returpulsene fra altimetrisatelliten. Prosesseringsmetoden kalles LR-RMC (Low Resolution with Range Migration Correction), og brukes på SAR-altimetridata fra Sentinel-3. Prosesseringsmetoden reduserer støy fra krusninger på åpent hav i global skala, men man er ikke kjent med hvordan datasettet fungerer regionalt og i kystsonen. Datasettet er laget av CLS (Collecte Localisation Satellite) og er en del av PEACHI-prosjektet (Prototype for Expertise on Altimetry for Coastal, Hydrology and Ice), med støtte fra CNES (Centre National d'Études Spatiales) som er en Fransk statlig organisasjon som former og gjennomfører Frankrikes rompolitikk i EU [AVISO, 2021].

Måten denne oppgaven skiller seg fra tidligere arbeid er hovedsakelig det eksperimentelle PEACHI-datasettet for bruk i kystsonen, og bruken av maskinlæring for å filtrere observasjonene basert på returpulsens form.

1.2 Avgrensninger

Altimetridataene som skal brukes i denne oppgaven har nærmest global dekning. For å gjøre oppgaven overkommelig vil en begrensning til fem testområder langs Norskekysten bli gjort i klassifiseringsfasen. Sentinel-3A PEACHI data fra syklus 20-24 innenfor testområdene blir inkludert. Det opprettes valideringsområder rundt vannstandsmålerne langs Norskekysten til beregning av dynamisk havtopografi, totalt 23. All tilgjengelig Sentinel-3A PEACHI data innenfor valideringsområder vil bli predikert (Juli 2016 - August 2018, syklus 6-34). Det vil bli utfordrende å være konsekvent med den manuelle klassifiseringen. I utarbeidelsen av klasser er det de mest representerte pulsformene som har fått sin egen klasse. Returpulser som faller mellom klasser blir plassert i den som likner mest. Posisjon, LR-RMC-returpuls, avstand og korreksjoner er parametere som blir hentet fra PEACHI-data.

Ofte er treningen av maskinlæringsmodeller en svært tidkrevende prosess med mindre man bruker spesiallaget, dyr maskinvare. Norges miljø- og biovitenskapelige universitet (NMBU) har en egen High-Performance Computing Cluster (HPC cluster) lokalt på universitetet, men man må ha god kjennskap til systemene som ligger til grunn for at det skal gi gevinst i forhold til tidsbruk. Data-

settet som er klassifisert og predikeres er forholdsvis små. Tidsbesparelsen er ikke stor nok til å sette seg inn i systemene til HPC clusteren. Derfor vil modellene og øvrige beregninger utføres på en datamaskin med Intel i7-8750H prosessor og Nvidia GeForce GTX 1060 grafikk-kort som gjør beregningene innenfor tilfredsstillende tid.

Maskinlæringspakkene jeg skal bruke er SKLearn og Tensorflow. SKLearn brukes til preprosessering og inndeling av data. Tensorflow brukes til å lage, trene og validere maskinlæringsmodeller. Tensorflow utnytter grafikk-kort for å korte ned beregningstiden. Øvrige beregninger blir utført i Python. Maskinlæringspakkene egner seg til prototypebygging og -testing av maskinlæringsmodeller. Brukergrensesnittet er enkelt, men man kan også stille på alle parametere om man ønsker det.

Det vil bli testet tre ulike maskinlæringsmodeller basert på styrt klassifisering. To av disse er utviklet i forarbeidet til oppgaven. Den tredje har vist gode resultater for klassifisering av hyperspektrale bilder. Modellene som er laget i forarbeidet til oppgaven har i liten grad blitt fininnstilt, og har i oppgaven låste parametere, foruten antall treningsepoker. Modellene vil bli validert med prinsipper fra holdout-metoden og ved å se på nøyaktighetstall og forvirringsmatriser. Returpulsene vil bli normalisert slik at maskinlæringsmodellene ikke får utfordringer med ulik signalstyrke. Slik blir klassifiseringen bare avhengig av formen til returpulsene for å bestemme riktig klasse. Dette vil kun gjelde for signalstyrken til returpulsene, eventuelle tidsforskyvninger vil ikke bli korrigert.

En enkel cluster-algoritme vil bli testet. Resultater vil ikke bli sammenliknet med egendefinerte klasser, men returpulsformens egenskaper i de ulike cluster-klassene vil bli evaluert.

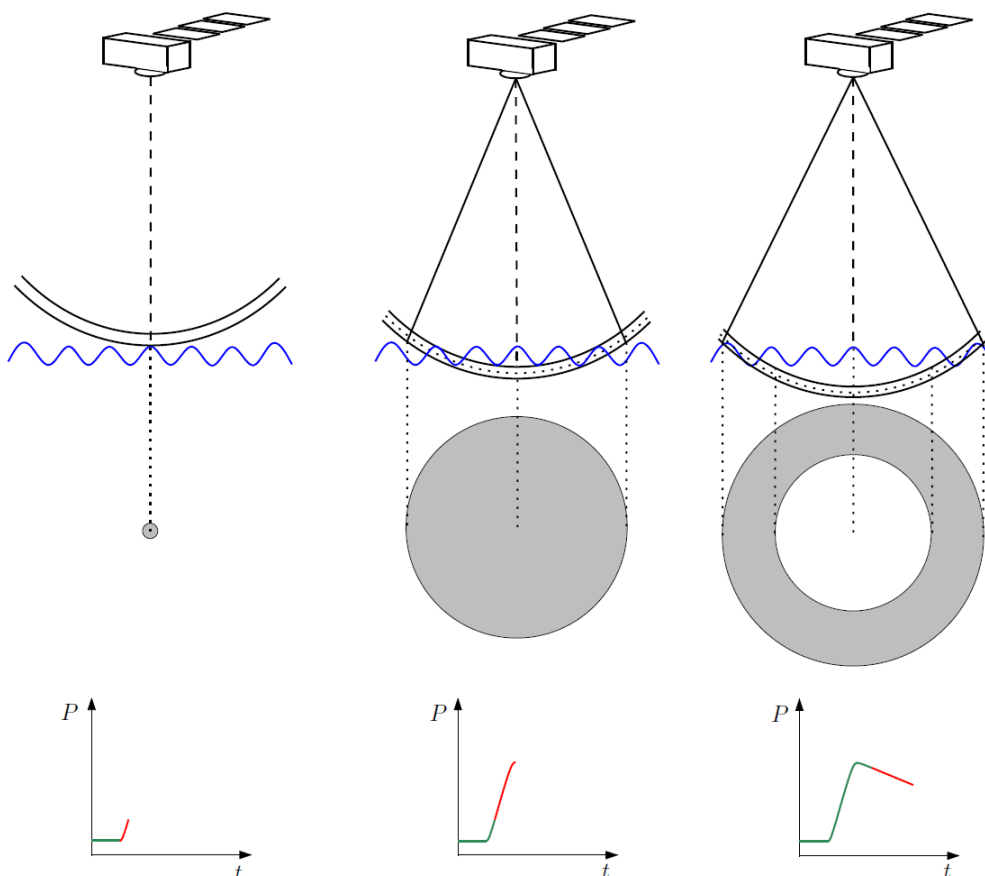
Grovfeilsøk ved Student's multippel-t test vil bli beregnet på samme måte som i Revhaug [2007] slik at Sentinel-3 PEACHI-datasettet kan sammenliknes med tidligere arbeid utført langs Norskekysten.

2 Teori

2.1 Altimetri

I tradisjonell satellitaltimetri sendes en radarbølge fra satellitten til jordoverflata. Størrelsen på det belyste området på jorden kalles satellittens fotavtrykk. Radarpulsen blir reflektert av jordoverflata og returnerer til satellitten. Ut fra tiden radarpulsen bruker kan avstanden avledes, se formel (2.1).

Radarsignalets bølgefront reflekteres først i nadir, før den brer seg ut rundt nadirpunktet, se figur 2.1. Tradisjonell altimetri er preget av grov romlig oppløsning fordi fotavtrykket må være stort nok til å glatte ut uregelmessigheter og små bølger. Samtidig må havoverflata være uniform i fotavtrykket.



Figur 2.1: Øverst vises bølgefrontens tidsutvikling på jordoverflata ved bruk av tradisjonell altimetri. Nederst vises mottatt signal fra tilhørende fotavtrykk. Figur hentet fra Ophaug [2017].

For å få avstanden (R) mellom satellitten og havet må man multiplisere lysfarten i vakuum (c) med halve gangtiden (Δt).

$$R = c \frac{\Delta t}{2} \quad (2.1)$$

Gangtiden Δt er fra signalet blir sendt til det mottas av satellitten. Derfor må man dele på to for å få gangtiden i en retning. For finne havoverflatas høyde over en ellipsoide må man subtrahere avstandsmålingen R fra satellittens høyde i forhold til ellipsoiden.

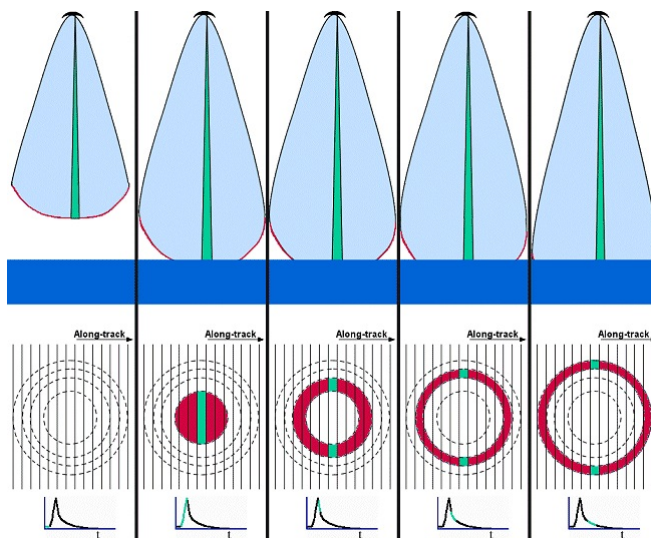
$$h = H - R \quad (2.2)$$

I likning (2.2) ser man at en feil i satellitthøyden H vil forplante seg direkte til h . Derfor er det viktig å bestemme banen med høy presisjon. Det er hovedsakelig den radielle komponenten i satellittbanen som vil føre til feil i h . De tidligste altimetrisatellittene hadde en dårligere bestemt flyhøyde sammenliknet med dagens satellitter. Banebestemmelsen har gått fra meter til centimeter presisjon [Stammer and Cazenave, 2017]. Dette har vært mulig ved å kombinere Satellite Laser Ranging (SLR), Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS) og Global Navigation Satellite System (GNSS). De geodetiske koordinatene for fotavtrykkets senter er også viktige, spesielt i kystsonen. Koordinatene kan brukes i preprocessering og filtrering. For eksempel kan de brukes til å klippe vekk pulsformer med nadirpunkt på land som hovedsakelig vil bestå av landkontaminering.

Satellittbanen er et kompromiss mellom flere hensyn. Ulike fenomen i havet har ulike perioder. For å observere et fenomen må syklusen være kortere enn perioden til fenomenet man ønsker å observere. Dekningsgraden påvirkes også når man velger bane. Om man ønsker å observere hav og is ved polene må satellitten passere over polene med ønsket frekvens. For å få observasjoner langs hele Norges kyst må satellitten ha en høy inklinasjonsvinkel. Inklinasjonsvinkelen er vinkelen mellom ekvatorplanet og planet satellittbanen spenner ut.

2.1.1 Avstandsmåling - SAR

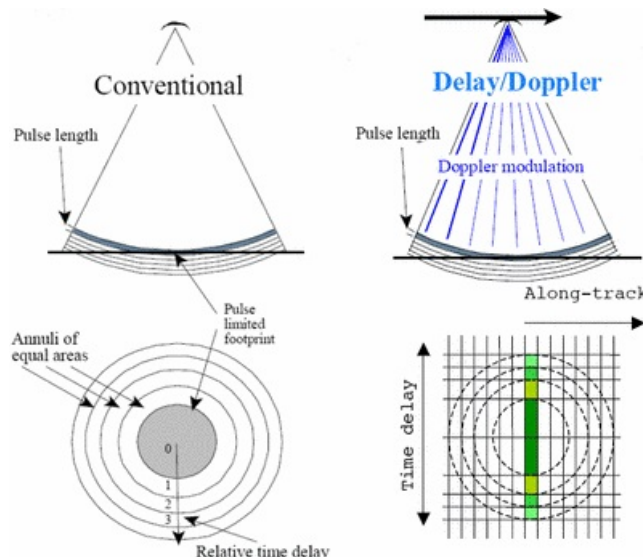
Hver nye altimetrisatellitt har arvet de beste egenskapene fra tidligere satellitter og forbedret seg. Dette kan gjelde korreksjoner på gangtid, presisjon på banebestemmelse eller prosesseringsteknikker. SAR-altimetri er en prosesseringsteknikk som har ført til store forbedringer i kystaltimetri.



Figur 2.2: Forklaring av pulsens tidsforløp i SAR-modus.

Hentet fra <http://www.altimetry.info/radar-altimetry-tutorial/how-altimetry-works/delay-doppler-or-sar-altimetry/>

Om en altimetrisatellitt opererer i SAR-modus innebærer det at radaren sender ut modulerte pulser med svært høy repetisjonsfrekvens. I prosesseringen av retursignalet utnyttes dopplerskiftet mellom pulser reflektert foran og bak satellittens til å isolere striper i de ellers sirkulære fotavtrykkene, se figur 2.2. Dette gjør at SAR-altimetri får langt bedre romlig oppløsning langs satellittsporet enn tradisjonell altimetri med sirkulære fotavtrykk. En følge av SAR-altimetri er økt signal-støyforhold og redusert sjans for refleksjon fra landområder. Se figur 2.3.



Figur 2.3: Tradisjonell radar altimeter(venstre) vs SAR altimeter(høyre) Hentet fra <http://www.altimetry.info/radar-altimetry-tutorial/how-altimetry-works/delay-doppler-or-sar-altimetry/>

2.1.2 Sentinel 3

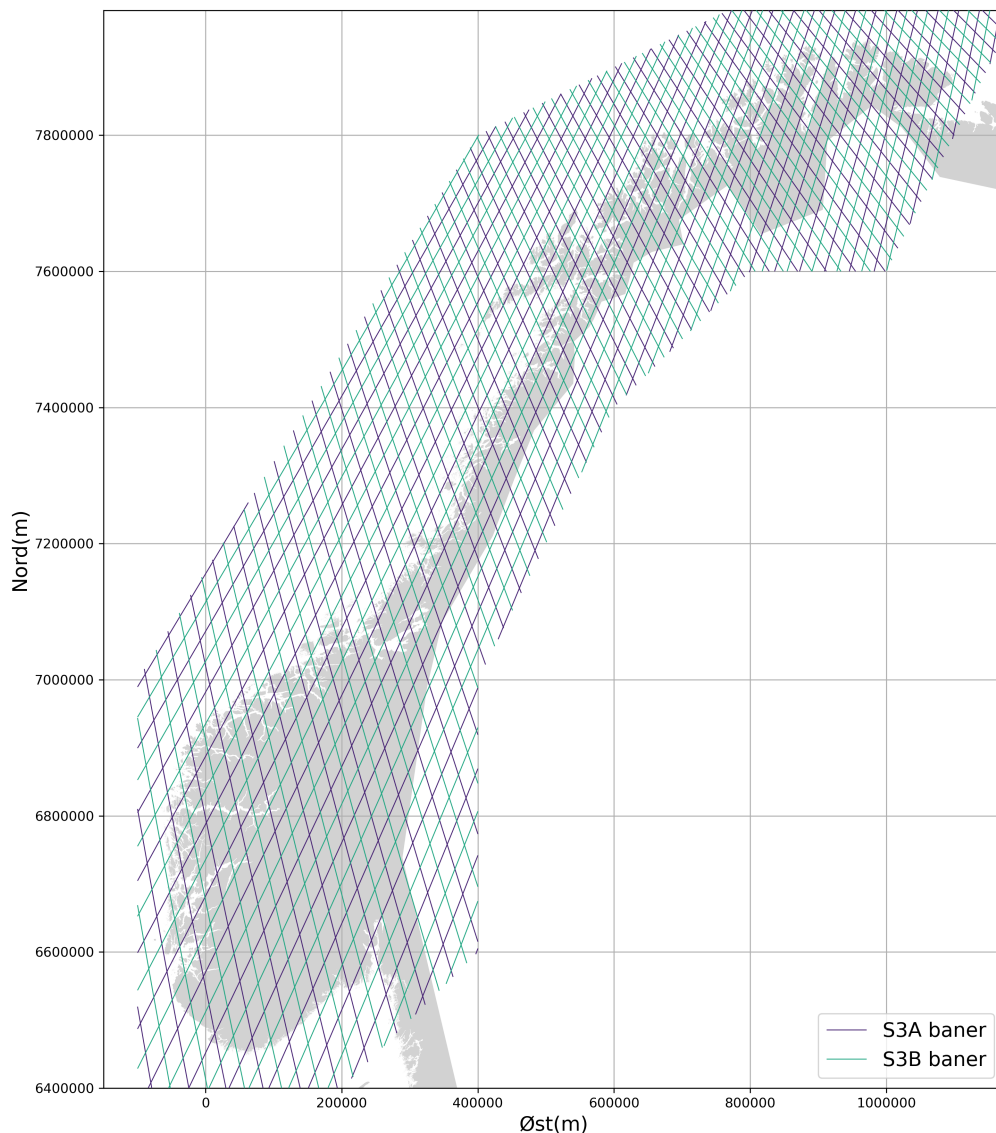
I forsøket på å klassifisere pulser skal det brukes Sentinel-3 data prosessert på en ny eksperimentell metode. Det finnes to identiske Sentinel-3 satellitter, Sentinel-3A og -3B, heretter omtalt som S3. Satellittene er en del av Copernicus, som er den europeiske unionens jordobservasjonsprogram. Sentinel-familien av satellitter er deres egne dedikerte jordobservasjonssatellitter. Disse har ulike sensorer og mål for romferden. Det er den europeiske romorganisasjonen ESA (European Space Agency) som skreddersyr og lager Sentinel satellittene, ofte i samarbeid med andre romorganisasjoner. Sentinel 6 er neste generasjons altimetrisatellitt. Den ble skutt opp i november 2020 og er resultatet av et samarbeid mellom ESA, EUMETSAT (European Organisation for the Exploitation of Meteorological Satellites), NOAA(National Oceanic and Atmospheric Administration) og NASA(National Aeronautics and Space Administration). Sentinel 6 har som hovedmål å måle havnivå.

Sentinel-3C og -3D er planlagt til å overta for 3A og 3B når de ikke lenger er operative. De vil ha identiske altimeterinstrumenter for å få en kontinuerlig måleserie på ca. 25 år. [ESA, 2021a]

Tabell 2.1: Observasjonsinstrumenter ombord Sentinel 3

Forkortelse	Beskrivelse
OLCI	Hav og land fargeinstrument
SLSTR	Hav og land overflatetemperatur instrument
SRAL	SAR radar altimeter
MWR	Microbølgeinstrument

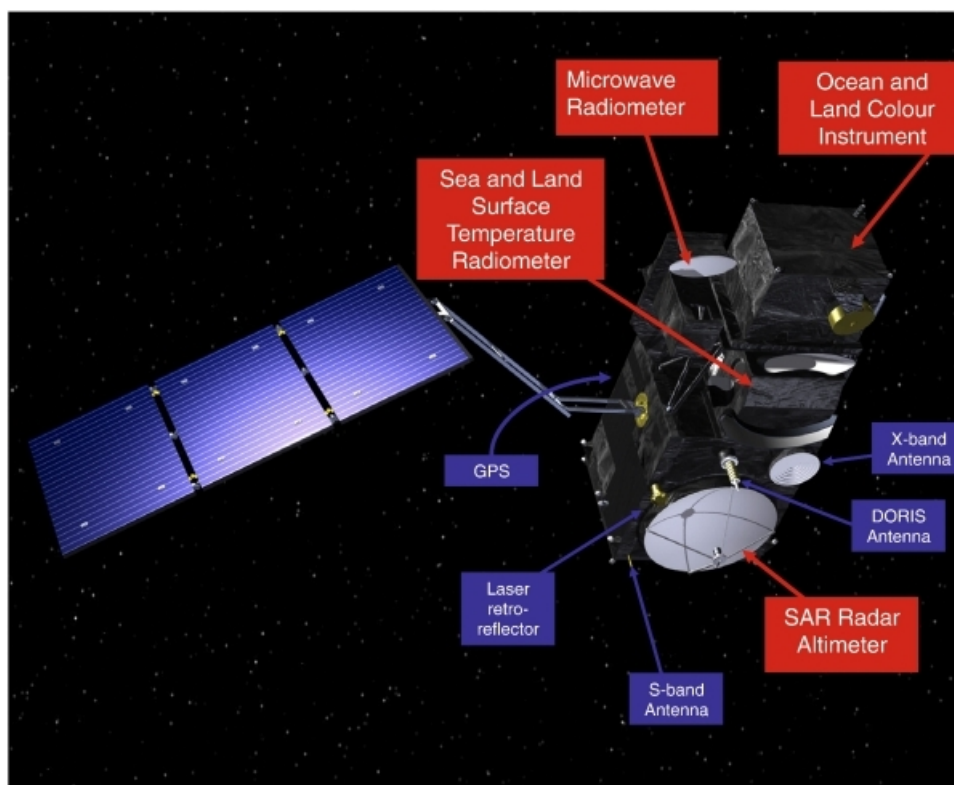
S3-satellittene går i nærpolar solsynkron bane og har en inklinasjonsvinkel på 98.65 grader [ESA, 2021b]. Det gjør at man får observasjoner til 81.4 grader nord og sør. Kombinert med de andre observasjonsinstrumentene om bord egner de seg til havmonitorering, samt miljø og klimaforskning, se tabell 2.1. Satellittene ligger i samme baneplan, hvor S3B ligger 140 grader bak S3A. På grunn av jordens rotasjon vil bakkesporet til S3B som følge av dette ligge ca. en halv grad vest for S3A bakkesporet ved ekvator. Satellittene har en repetisjons tid på 27 dager. Det vil si at de flyr over det samme området hver 27. dag. Satellittene flyr over det nominelle sporet med en nøyaktighet på ± 1 km [ESA, 2021b]. Dette fører til at nadirpunktet til pulsen kan havne på ulikt sted og vil kunne inneholde ulik grad av landkontaminering.



Figur 2.4: Nominelle bakkespor for S3 over Norge. Som følge av inklinasjonsvinkelen på 98.65 grader er det altimetriobservasjoner langs hele Norskekysten.

Figur 2.4 viser en syklus for S3 over Norge. I løpet av en syklus vil satellittene hele tiden fly over nye områder. I løpet av et pass vil summen av jordens rotasjon og satellittbanens rotasjon medføre en lengdegradsforflytning på ca. 25 grader. Et pass er en runde rundt jorden og regnes fra ekvator til ekvator på samme side. Satellittbanen er solsynkron, dette innebærer at baneplanet roterer i takt med jordas rotasjon rundt sola. Når hele syklusen er fullført er det ca. 104 km mellom krysningspunktene diamantmønsteret S3A-bakkesporene lager (se figur 2.5) ved ekvator. Ved 58 grader nord (Tregde) er det ca. 55 km mellom krysningspunktene. Ved 71 grader nord

(Honningsvåg) er det ca. 34 km mellom krysningspunktene. Om man inkluderer S3B-observasjoner halveres avstandene mellom krysningspunktene.



Figur 2.5: Observasjonsinstrumenter (rød) og navigasjonsinstrumenter (blå) ombord Sentinel-3 satellittene.

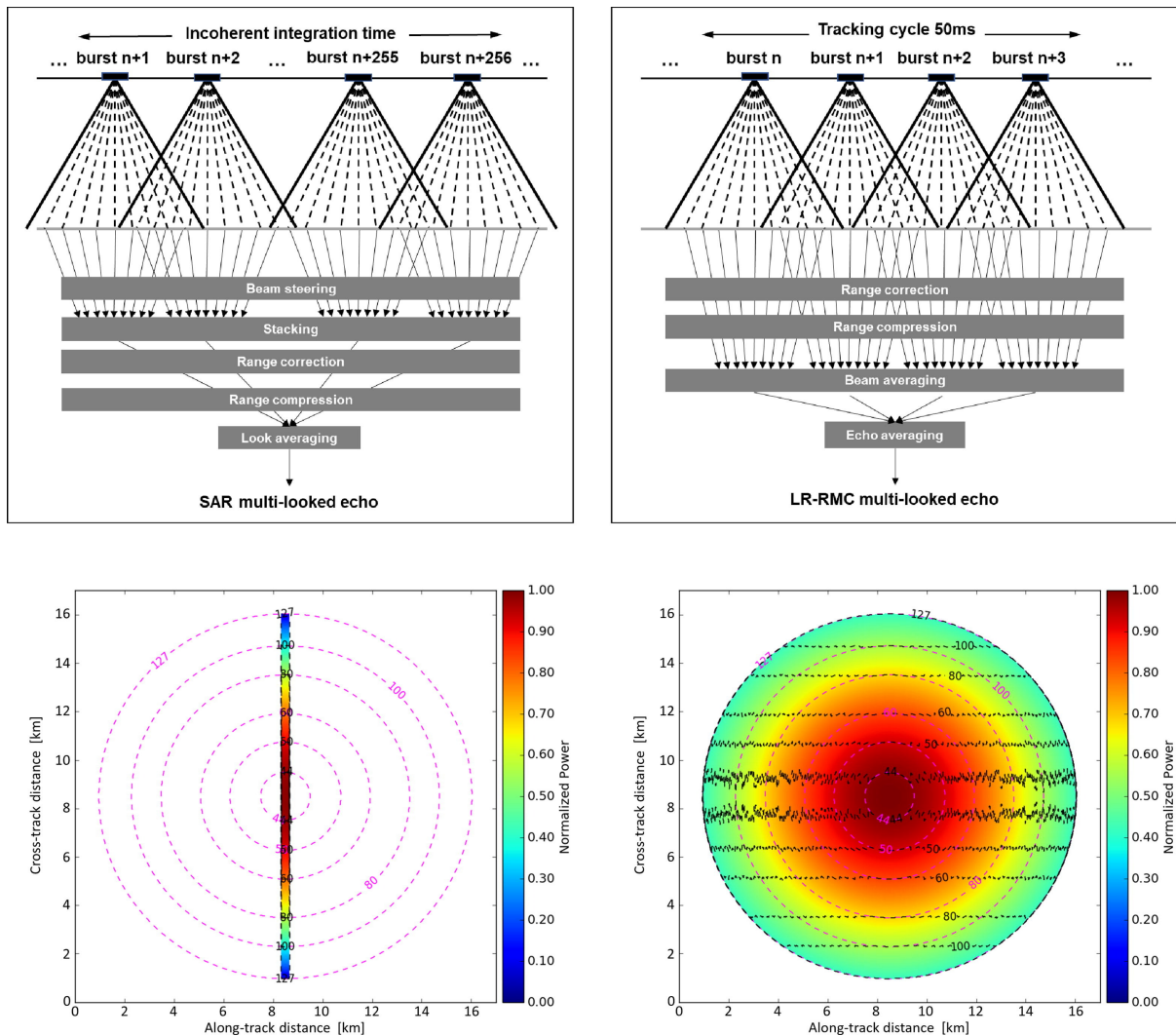
Hentet fra: <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3/overview> - Sist lastet: 22/02 2021.

LR-RMC-prosessering av Sentinel-3 data

Dataene brukt i denne oppgaven er prosessert på en annen måte enn originaldatasettet til Sentinel-3 satellittene og er kun tilgjengelig for Sentinel-3A. I figur 2.6 ser man prosesseringsmetoder og resulterende fotavtrykk. LR-RMC metoden fjerner ifølge Moreau et al. [2021] all støy fra krusninger ute på åpent hav, på en global skala. Det er ikke gjort noe arbeid for å undersøke hvordan metoden presterer regionalt eller i kystsonen med LR-RMC-data fra S3. Prosesseringsmetoden er tidligere anvendt på CryoSat-2 data [Phalippou and Demeestere, 2011].

Størrelsen på fotavtrykket ved bruk av denne metoden er sammenliknbart med fotavtrykket til tradisjonell altimetri. Derfor kan det stilles spørsmål om hvordan dataene vil prestere i kystsonen på tross av at AVISO har listet kystaltimetri som bruksområde.

Nederst i figur 2.6 ser man normalisert intensitet fordelt i fotavtrykket til satellitten for SAR-observasjoner og LR-RMC-observasjoner. Her ser man at LR-RMC-observasjoner er mer utsatt for landkontaminering enn SAR-observasjoner. Signalstyrkefordelingen i fotavtrykket vil ha innvirkning på forholdet mellom signal og støy til en puls som inneholder landkontaminering.



Figur 2.6: Prosesseringsmetode (øverst) og foravtrykk (nederst) for SAR (venstre) og LR-RMC (høyre). Figur hentet fra Moreau et al. [2021]

Tilgjengeligheten på dataene er svært god, da de kan hentes fra en File Transfer Protocol (FTP) server, enten via en nettleser eller et Application Programming Interface (API). Hvor sistnevnte har blitt benyttet i denne sammenhengen. Datasettet er mindre og lettere håndterlig enn offisielt S3-data fra Copernicus fordi det omfatter færre parametere enn offisielt S3A-produkt fra Copernicus. Dette er en fordel når det gjelder lokal lagringskapasitet, prosesseringskraft og minne(RAM).

2.1.3 Korreksjoner

For å bestemme havnivå (Sea Surface Height, SSH) nøyaktig fra altimetriobservasjoner må man korrigere for påvirkninger på signalets ferd gjennom atmosfæren og geofysiske signaler. SSH er høyden på havet over en ellipsoide. Flere av korreksjonene må takles annerledes når man er nær kysten eller i grunne havområder [Andersen and Scharroo, 2011]. Alle korreksjoner påført altimetriobservasjonene er hentet fra PEACHI-datasettet.

$$R_{korr} = R_{obs} - \Delta R_{tørr} - \Delta R_{våt} - \Delta R_{iono} - \Delta R_{ssb} \quad (2.3)$$

R_{korr} er den korrigerede avstanden mellom satellitt og jordoverflata, R_{obs} er den observerte avstanden, $\Delta R_{tørr}$ er korreksjonen for den tørre delen av troposfæren, $\Delta R_{våt}$ er korreksjonen for den våte

delen av troposfæren, ΔR_{iono} er korreksjonen for ionosfæren og ΔR_{ssb} er korreksjonen for effekter knyttet til vekselvirkning med havoverflata (Sea State Bias).

Satt inn i likning (2.2) for R får man

$$h = H - (R_{obs} - \Delta R_{t\ddot{t}orr} - \Delta R_{v\ddot{a}t} - \Delta R_{iono} - \Delta R_{ssb}) \quad (2.4)$$

Den tørre delen av troposfæren består av tørre gasser og er den største korreksjonen som påføres avstandsmålingen [Andersen and Scharroo, 2011].

$$R_{t\ddot{t}orr} \approx -0.2277 \cdot P_0(1 + 0.0026\cos(2\varphi)) \quad (2.5)$$

Her er P_0 lufttrykk ved havnivå og φ er breddegrad til observasjonspunktet. Lufttrykket ved havnivå bestemmes ved interpolering av lufttrykkobservasjoner fra enten European Center for Medium-Range Weather Forecasts (ECMWF) eller U.S. National Centers for Enviromental Prediction(NCEP).

Vanddampmengde i troposfæren varierer i større grad enn de tørre gassene og har høy romlig oppløsning. Korreksjonen for vanddampinnholdets påvirkning for signalets gangtid finner man ved å integrere vanddampinnholdet(ρ_{vap}) gjennom troposfæren og multiplisere med en empirisk bestemt parameter (β'_{vap}).

$$R_{v\ddot{a}t} \approx -\beta'_{vap} \int_0^R \rho_{vap}(z), dz \quad (2.6)$$

Et mikrobølge-radiometer ombord S3 måler troposfærens vanddampinnhold, og korrigerer gangtiden tilsvarende. Fotavtrykket til radiometeret er rundt 23 km [Frery et al., 2020]. Dermed inneholder den våte troposfærekorreksjonen landkontaminering lengre fra kysten enn altimetersignalet, som har et fotavtrykk på rundt 10 km. Et alternativ er å bruke metrologiske modeller (ECMWF eller NCEP) til å korrigere for våt troposfærisk forsinkelse, og er svært aktuelt i kystsonen hvor landområder forstyrrer radiometermålingene.

Ionosfærisk forsinkelse er avhengig av ionosfærens totale elektrontetthet (TEC) og frekvensen (f) på radarsignalet som skal korrigeres.

$$R_{iono} = -k \frac{TEC}{f^2} \quad (2.7)$$

En to-frekvent altimetriradar identifiserer ionosfærens påvirkning på gangtiden slik at dette også kan korrigeres for. Korreksjonen kan også komme fra modeller basert på to-frekvente GNSS-målinger.

I tillegg til korreksjonene for påvirkninger av signalets gangtid må man også korrigere for andre effekter som påvirker observasjonspunktets høyde over ellipsoiden. Dette gjelder effekter som tidevann, tidejord, GIA (Glacial Isostatic Adjustment - Postglasial heving) og tidevannets last. Her er det ikke avstanden mellom satellitten og jordoverflata som korrigeres, men høyden over ellipsoiden.

$$h_{SSH} = h + h_{geo} \quad (2.8)$$

Her er h_{SSH} havets høyde over ellipsoiden korrigert for geofysiske effekter og effekter som påvirker signalets gangtid gjennom atmosfæren. h er havets høyde over ellipsoiden kun korrigert for effekter som påvirker signalets gangtid. h_{geo} er summen av alle geofysiske effekter som påvirker observasjonspunktets høyde over ellipsoiden.

2.1.4 Referanserammer

Referansesystem og referanserammer er sentrale i all jordobservasjon. Satellittaltimetri observerer små endringer over lang tid og stiller derfor svært strenge krav til referanserammens nøyaktighet og stabilitet over tid. En referanseramme som ikke innfrir de strenge kravene, vil innføre systematiske feil i observasjonene og dermed se ut som en tilsynelatende endring i havoverflatens høyde.

Sammenlikning av forskjellige observasjonsteknikker forutsetter at observasjonene er i samme referansesystem. Ellers kan man risikere at man får et systematisk avvik. For å knytte sammen altimetriobservasjoner fra forskjellige oppdrag må de også kunne beregnes til samme referansesystem. Da er det viktig å benytte referanserammer med høy nøyaktighet og god stabilitet. Satellittenes levetid er begrenset fordi den går tom for drivstoff til å korrigere banen. Planlagt levetid for S3A og S3B er ca. syv år.

2.1.5 Dynamisk havtopografi

Dynamisk havtopografi skal brukes til å sammenlikne altimetriobservasjoner og vannstandsobservasjoner. Altimetriobservasjonenes posisjon er fordelt i henhold til satellittbanen, og sannsynligheten for at en altimetriobservasjon befinner seg på nøyaktig samme posisjon som en vannstandsmåler er usannsynlig. I tillegg ville altimetriobservasjonen vært preget av landkontaminering fordi vannstandsmålerne står på land. Dynamisk havtopografi (DT) kan beregnes ved å subtrahere en geoidmodell (N) fra havoverflatas høyde (Sea Level - SL) over ellipsoiden:

$$DT = SL - N \quad (2.9)$$

Det er viktig å transformere observasjonene som skal sammenliknes til samme referanse og at de ligger i samme tidevanns-system som geoidmodellen [Breili, 2021].

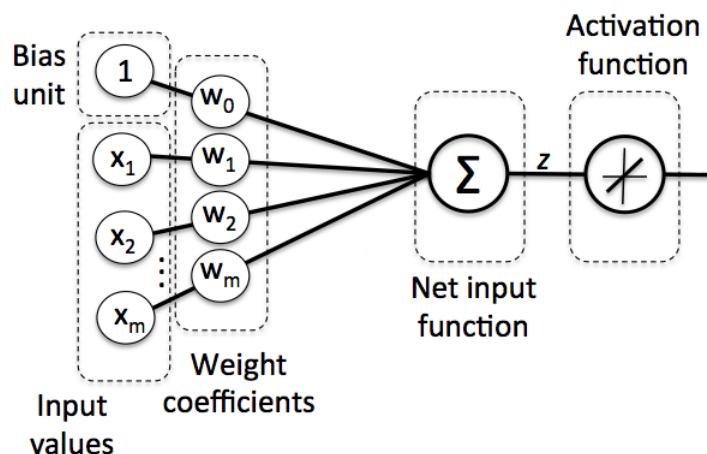
2.2 Maskinl ring

2.2.1 Styrt klassifisering

Målet med styrt klassifisering er   lære eksisterende data slik at en modell kan predikere nye data. For   gj re det m  man p  forh nd ha et datasett og en tilh rende fasit. For at man skal ha en fasit tilgjengelig forutsetter det at man p  forh nd har kunnskap om dataene. Det forutsetter ogs  at det eksisterer en m te   observere eller beregne fasiten. Styrt klassifisering egner seg til   l se problemer hvor datasettet inneholder egenskaper eller st rrelser man med enkelhet kan observere, i motsetning til fasiten.

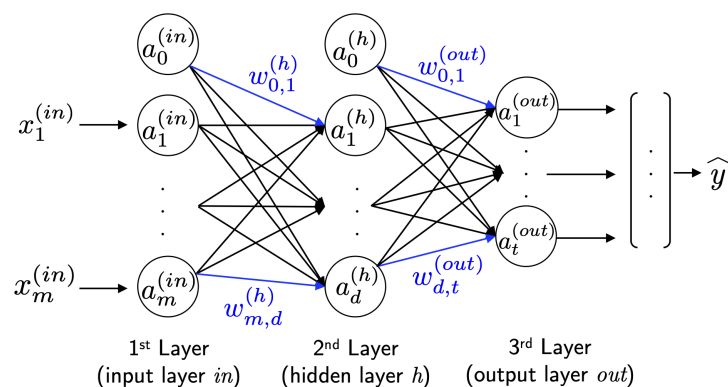
Et eksempel er Gou et al. [2013] hvor man fra et infrar dt bilde av vakuumpakket spekemat kunne predikere vann-, fett- og saltinnhold, med henholdsvis 1.34%, 1.36% og 0.71% feilprediksjon. Ved bruk av maskinl ring kan en ellers tidkrevende kvalitetskontroll, som ikke er mulig   utf re p  alle varene, bli en del av samleb ndet.

Dyp l ring er en underkategori av maskinl ring og har f tt mest oppmerksomhet i fagfeltet. Dyp l ring kan beskrives som effektiv trening av kunstige nevrane nett(NN - Neural Networks) med mange lag. NN er basert p  hypoteser og modeller om hvordan hjernen l ser kompliserte problemer [Raschka and Mirjalili, 2019].



Figur 2.7: Oppbygningen til nevrane i NN. Tilpasset fra Raschka and Mirjalili [2019].

I figur 2.7 ser man de ulike delene som en nevron best r av. Ved input(x) er en partisk enhet (bias unit), i tillegg til et sett med vekt-koeffisienter(W). Det er i oppdateringen av vekt-koeffisientene det er mulighet for l ring. Inputverdiene blir multiplisert med de tilh rende vekt-koeffisientene og resultatet blir summert(z), som deretter g r gjennom en aktiveringsfunksjon. N r flere av disse nevronene blir satt sammen f r man et nett. Et generelt eksempel er vist i figur 2.8.



Figur 2.8: Eksempel på NN hvor sirkler representerer nevroner og pilene representerer vekt-koeffisientene. Hentet fra Raschka and Mirjalili [2019]

Nettet i figur 2.8 kan utvides med flere skjulte lag (hidden layer) og nevroner. Antall nevroner beskriver dybden på laget. Dybden på lagene, sammen med antall skjulte lag sier noe om kompleksiteten på modellen. Input-lagets størrelse bestemmes av antall attributter datasettet består av. Attributtene er ofte størrelser eller egenskaper til et objekt som kan observeres med eksisterende metoder, gjerne enkle og billige. Resultat-lagets (output layer) størrelse bestemmes av antall klasser datasettet har. Datasettet som man bruker må inneholde både attributter og en tilhørende klasse til hvert objekt.

Det finnes flere NN-strukturer som egner seg til forskjellige problemer, for eksempel konvolusjonelle nevrale nett (Convolutional neural net - CNN) som egner seg til bildeklassifisering. Tilbakevendende nevrale nett (Recurrent neural net - RNN) egner seg til klassifisering av data hvor rekkefølgen i datasettet inneholder informasjon. Derfor brukes RNN ofte til tekstanalyse og tidsserier.

CNN gir datamaskiner evnen til å se, og er satt opp for å etterlikne det menneskelige synssystemet. CNN består ofte av to deler. Den første henter ut informasjon av datasettet og endrer på størrelse og form. Den andre delen består av et fullkoblet NN som predikerer.

Aktiveringsfunksjon

Aktiveringsfunksjoner skal basert på signaler fra tidligere lag avgjøre om nevronen skal sende et output-signal. Forskjellige aktiveringsfunksjoner har ulike egenskaper. Det er ikke uvanlig å ha flere funksjoner i en modell.

Hyperbolic tanget (Tanh) er et eksempel på en aktiveringsfunksjon man bruker i skjulte lag. ϕ kan få verdier fra -1 til 1 . Om nettoinput (z) fra nevroner i tidligere lag er stort vil Tanh returnere 1 , som kan føre til at modellen lærer sakte.

$$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.10)$$

Rectified linear unit activation (ReLU) er en annen aktiveringsfunksjon som brukes i skjulte lag. Funksjonen returnerer null for negative verdier og z for positive verdier. På den måten har modellen muligheten til å lære effektivt, og man får ikke samme utfordringer som med Tanh.

$$\phi(z) = \max(0, z) \quad (2.11)$$

Softmax funksjonen brukes til å estimere sannsynligheten for klassetilhørighet basert på nettoinput (z)

fra nevroner i tidligere skjulte lag. Sannsynlighet for alle klassene summerer opp til en.

$$\phi(z) = \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}} \quad (2.12)$$

Taps- og optimaliseringsfunksjon

Tapsfunksjonen beregner et mål på suksess for predikeringen. For et flerklasseproblem vil man bruke kategorisk kryssentropi (Categorical crossentropy).

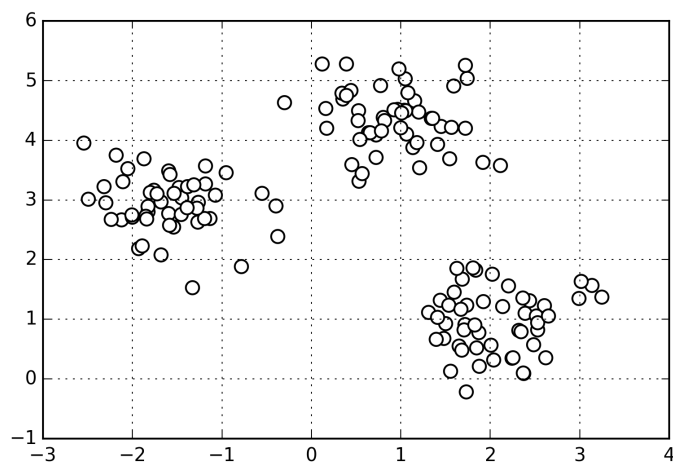
$$L = \sum_i^K y_i \log \hat{y}_i \quad (2.13)$$

Hvor y_i og \hat{y}_i er henholdsvis fasit og prediksjon for klasse K . Det er verdien fra tapsfunksjonen som optimaliseringsfunksjonen bruker i oppdateringen av modellen [Chollet et al., 2018]. Oppdatering av modellen kalles ofte for en epoke og er en treningsrunde som modellen foretar. Input-data går gjennom nevronene i modellens lag til man får en output og feilen i prediksjonene blir beregnet mot fasit. Deretter vil tapsfunksjonen oppdatere modellen. Oppdateringen innebærer en endringsverdi for vektskoeffisientene justert med en læringshastighet.

Optimaliseringsfunksjonen gjør slik at vektskoeffisientenes oppdatering fører til minimalisering av tapet. Det gjør den ved å finne den negative gradienten til tapsfunksjonen. *Adam* er en av flere optimaliseringsfunksjoner implementert i Tensorflow. Den skal ifølge Kingma and Ba [2014] håndtere store datasett godt, ikke ta opp mye datakraft eller minne i beregningen og prestere på linje med andre optimaliseringsfunksjoner.

2.2.2 Ikke-styrt klassifisering

Ikke-styrt klassifisering kan hjelpe oss å finne koblinger eller mønstre man ikke har sett selv. Man har ikke forutsetningen med tidligere kjennskap til dataene i motsetning til styrt klassifisering. Clustering er en form for ikke-styrt klassifisering som identifiserer grupper med liknende attributter, se figur 2.9. Ved å definere likhet som det motsatte av distanse kan man beregne hvor like to punkter er [Raschka and Mirjalili, 2019]. Prinsippal Component Analysis (PCA) kan hjelpe algoritmen å finne mønstre og er hyppig brukt som et preprosesseringssteg for clustering.



Figur 2.9: Eksempel på samlinger som enkelt kan skilles fra hverandre. Hentet fra Raschka and Mirjalili [2019].

K-Means cluster

K-Means er en cluster algoritme som er enkel å implementere og krever lite datakraft, dette har gjort den svært populær. K-Means faller under kategorien prototype-cluster hvor et prototype-objekt definerer senter i samlingen. Prototypeobjektet er et gjennomsnitt av hele samlingen for kontinuerlige variabler [Raschka and Mirjalili, 2019]. I figur 2.9 ser man et eksempel med to dimensjoner for visualisering, men K-Means leter etter grupperinger i alle dimensjonene til datasettet. Ved bruk av K-Means må man på forhånd definere antall samlinger. Det betyr at man må kjenne til datasettet på forhånd eller man kan prøve seg frem med flere K-Means-modeller.

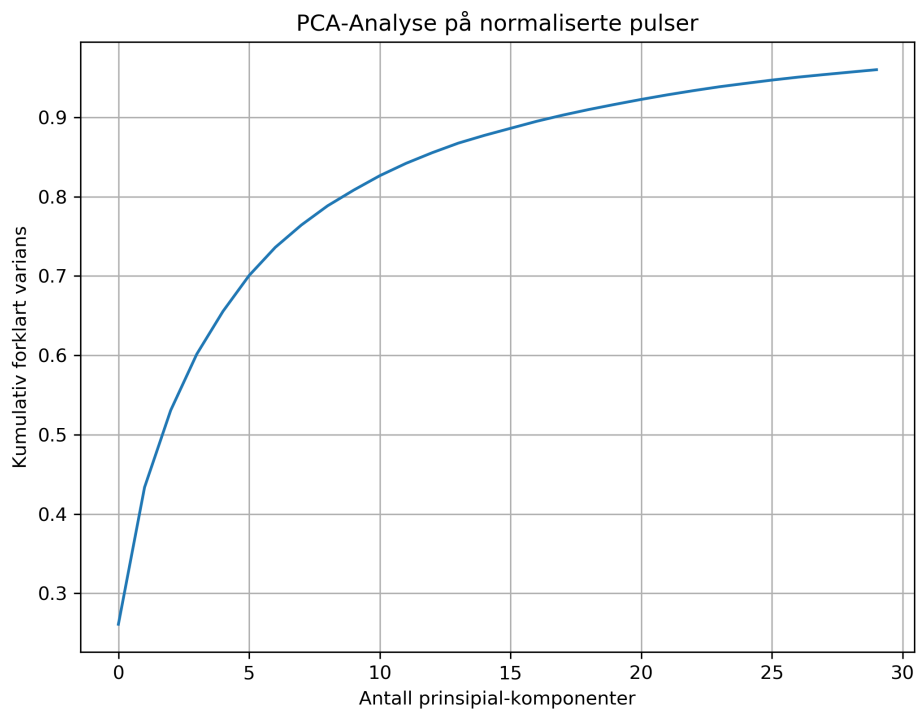
2.2.3 Attributtredigering

Attributtredigering omfatter alle endringer man gjør på datasettet før man setter dem inn i en maskinlæringsmodell. Det kan være skalering, transformering, håndtere uteliggere og manglende verdier samt takle skjevfordelt klassefordeling. Alle metodene skal bidra til at modellen lærer å predikere datasettet på best mulig måte. I denne oppgaven behøver man ikke ta hensyn til uteliggere eller manglende verdier fordi datasettet er produsert under kontrollerte forhold.

Normalisering skalerer individuelle objekt med hensyn til alle attributtene for å få en enhetsnorm [Pedregosa et al., 2011]. Det vil si at kvadratsummen av alle utvalgpunktene etter normalisering blir en. På den måten vil den individuelle returpulsen beholde formen og noe av signalstyrken.

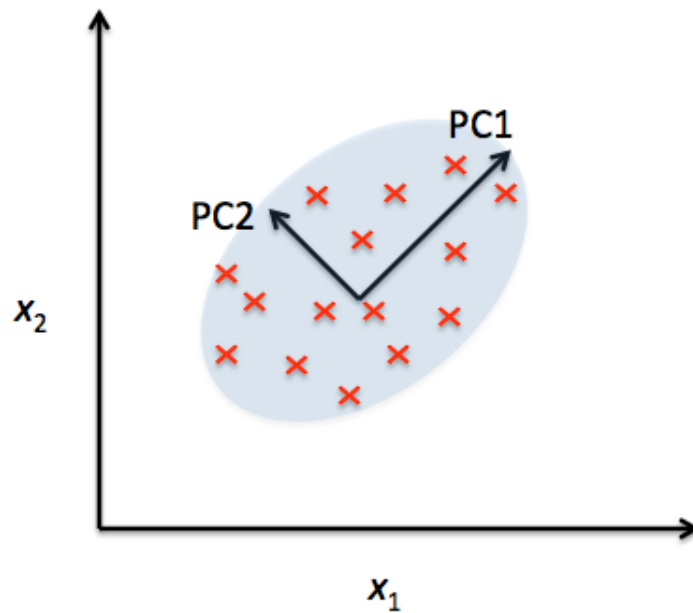
Prinsipal komponent-analyse

Med PCA ønsker man å hente ut viktig informasjon fra et datasett og representere det som par av ortogonale variabler kalt prinsipale komponenter. Prinsipal komponentene er lineærkombinasjoner beregnet fra det originale datasettet [Abdi and Williams, 2010]. I figur 2.10 ser man summert varians for hver prinsipal komponent beregnet ved SKlearn. Dette kan være et hjelpemiddel når man skal bestemme hvor mange prinsipale komponenter man ønske å inkludere. Ved å bare ta vare på den viktigste informasjonen blir datasettet mindre og enklere å bruke.



Figur 2.10: Summert varians som hver prinsipal komponent forklarer. Denne grafen er produsert med S3-LR-RMC-data og er unik for datasettet.

I figur 2.11 ser man hvordan PCA fungerer i prinsippet. Den første prinsipale komponenten (PC1) finner retningen med størst varians og vil derfor inneholde mest informasjon, den andre (PC2) må stå ortogonalt på PC1. Om man avbilder det originale datasettet til prinsipal komponenter kan man oppdage nye mønstre i datasettet.

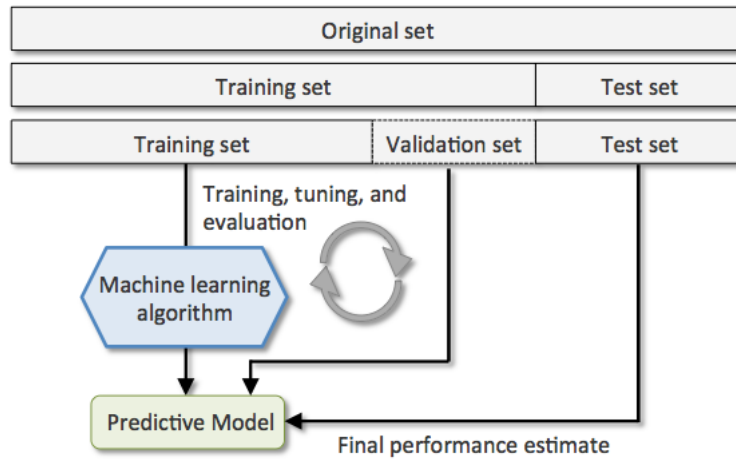


Figur 2.11: Visuell beskrivelse av PCA. Her er x_1 og x_2 originale attributtakser. Hentet fra Raschka and Mirjalili [2019].

2.2.4 Validering av maskinlæringsalgoritmer

En mye brukt metode for validering av maskinlæringsalgoritmer er Holdout-metoden. Først splittes datasettet man jobber med i to deler, se figur 2.12. Disse blir ofte kalt treningsdata og testdata. Treningsdata skal brukes til å trene modellen og bør ha en viss størrelse, hvis ikke kan modellen prestere dårlig. Testdata skal brukes til å teste prediksjonene modellen gjør. Det må være stort nok til å gi en pålitelig validering av maskinlæringsmodellen. Når man splitter et datasett med funksjon fra SKlearn får man flere valg. De viktigste er hvilken størrelse datasettene skal ha i forhold til hverandre og om oppdelingen skal ta hensyn til klassefordelingen. Om man velger å ta hensyn til klassefordelingen vil de nye datasettene få lik klassefordeling som det opprinnelige datasettet. På den måten sikrer man seg at alle klassene blir inkludert i treningen og predikteringen. Testdatasettet vil bli en simulering av hvordan en ferdig trent modell vil prestere på usett data.

Videre kan man gjøre en ytterligere oppdeling av treningsdataene. Oppdelingen skjer bare i treningsdatasettet, og det nye datasettet kalles valideringsdata, se figur 2.12. Valideringsdataene predikeres og testes for hver epoke modellen foretar i treningen. På den måten får man innblikk i hvordan modellen utvikler seg i løpet av treningen på usett data. Maskinlæringsmodellen forventes å overtilpasse seg treningsdata. Man kan med en overkompleks modell og nok epoker(tid) lage en tilsynelatende perfekt modell til ethvert problem, så lenge man predikerer samme data som modellen er trent på. Med resultater fra prediksjon av valideringsdataene vil man få et mer reelt bilde på modellens prestasjon. Man får også en indikasjon på når man bør stoppe treningen, for å unngå overtilpassing.



Figur 2.12: Holdout-metode for validering av maskinlæringsmodeller. Hentet fra Raschka and Mirjalili [2019].

Både SK-Learn og Tensorflow-Keras har mange innebygde metoder for å beregne prestasjonen til en modell. Det er viktig at man bruker prestasjonstall tilpasset problemstillingen. Det enkleste eksempelet på et prestasjonstall er nøyaktighet (accuracy).

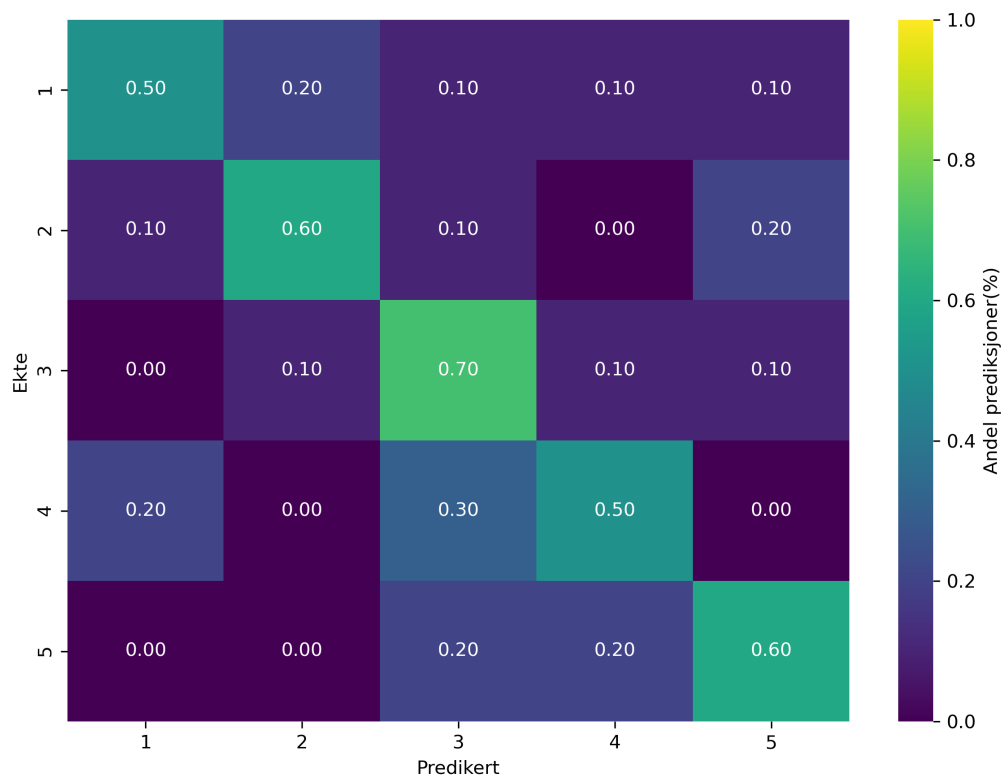
$$\text{nøyaktighet} = \frac{\text{korrekt predikert}}{\text{antall prediksjoner}} \quad (2.14)$$

Den beregnede nøyaktigheten i likning (2.14) tar ikke vare på informasjon om klassefordeling. Om klassefordelingen er veldig skjev kan nøyaktighetstallet være misvisende. På tross av dette er det et intuitivt prestasjonstall. I tilfelle nøyaktighetstallet alene ikke er tilstrekkelig kan man bruke andre metoder for å validere en maskinlæringsmodell. En metode som fungerer godt i kombinasjon med nøyaktighetstallet er forvirringsmatriser. I matrisene blir informasjon om klassefordeling ivaretatt og man får et nøyaktighetstall for hver klasse.

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Figur 2.13: Forvirringsmatrise for binært problem. Hentet fra Raschka and Mirjalili [2019].

I figur 2.13 ser man et eksempel for et binært problem, men metoden kan enkelt utvides til flere klasser. Om maskinlæringsmodellen presterer godt vil diagonalen få høyest verdi. Elementer utenfor diagonalen sier noe om hvor maskinlæringen gjør feil, se figur 2.14. Radene i matrisen summerer til en, og verdiene i diagonalelementene er det samme som nøyaktighetstall for hver enkelt klasse.



Figur 2.14: Forvirringsmatrise for flerklasseproblem.

3 Datasett og metoder

3.1 LR RMC

Etter manuell klassifisering består datasettet av 6900 klassifiserte returpulser. Ved bruk av holdout-metode med fordeling på 80/20 vil treningsdatasettet bestå av 4416 returpulser, testdatasettet av 1380 returpulser og valideringsdatasettet av 1104 returpulser.

Datasettet inneholder flere relevante parametere. Parameterne er korreksjoner, metrologiske data fra modeller og ulike flagg som beskriver målesituasjonen. For beregning av korrigert avstand benyttes korreksjoner og modeller fra PEACHI-datasettet

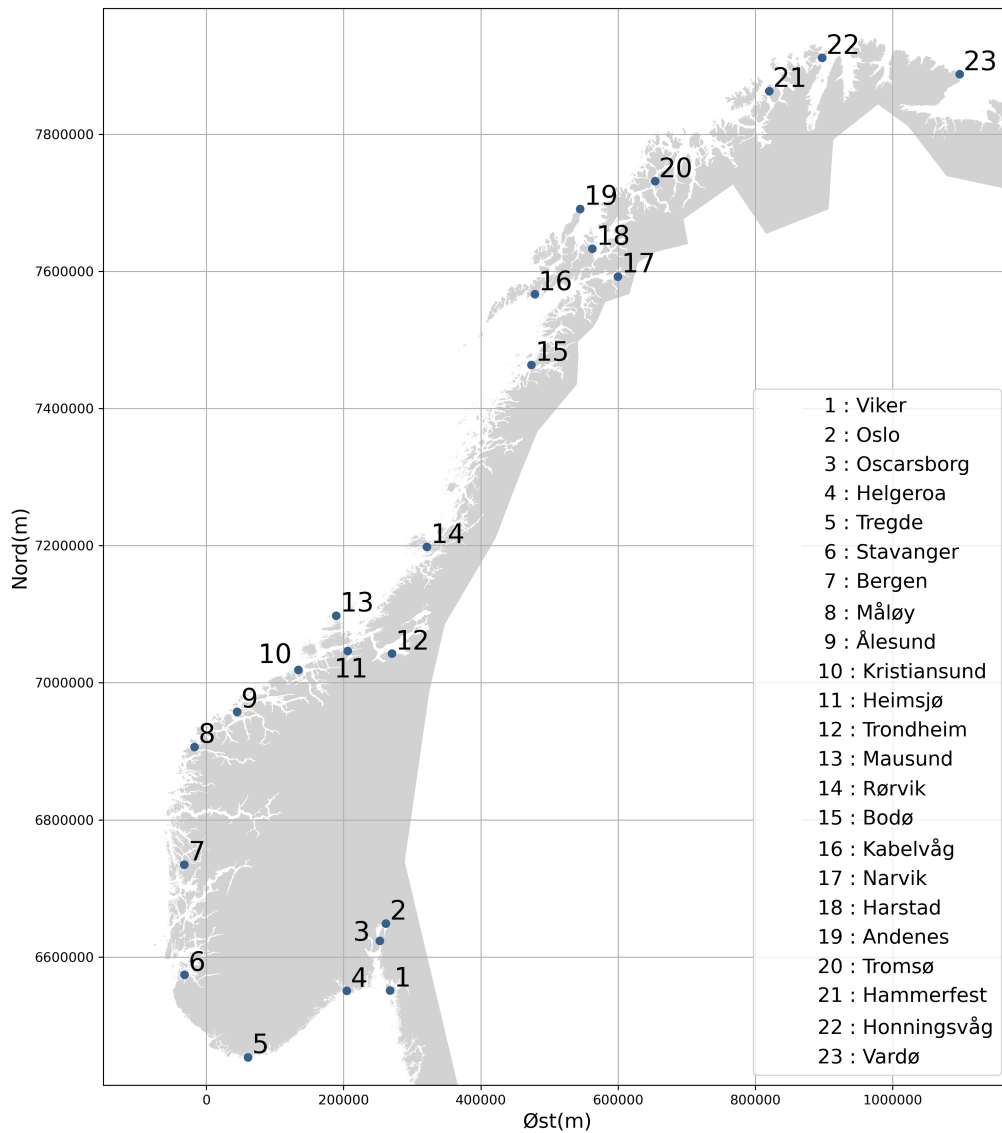
3.2 Kystlinje

Kystlinjen som er brukt for filtrering av altimetriobservasjoner med nadirpunkt på land er hentet fra openstreetmaps. Igjen et godt eksempel på ressursene som er tilgjengelig i opensource-miljøet. Selv om det er mye ressurser tilgjengelig må man være ekstra påpasselig med data som kommer fra opensource-miljøer. Det kan være feil i datasett og metadata. Som et eksempel kikket jeg på områder i den sørlandske skjærgården i modellen, samtidig som jeg kikket på flyfoto fra norgebilder.no [Kartverket, 2021]. På flybildene var det flere holmer og skjær som ikke blir inkludert i kystmodellen. De aktuelle holmene var større enn oppgitt romlig oppløsning på kystmodellen som er 1 m. Gjennomsnittsdiameteren på en av øyene som ikke ble inkludert var ca. 10 m, målt med linjalverktøy på nettsiden. Det er usikkert hvor mye en holme med diameter på 10 m vil utgjøre på avstandsmålingen og returpulsens form.

3.3 Vannstandsmålere

For å validere resultatene i denne oppgaven vil det bli brukt observasjoner fra alle permanente vannstandsmålere i Norge med unntak av måleren i Ny-Ålesund på Svalbard. En vannstandsmåler måler havets høyde i forhold til land. Vannstanden logges hvert sekund og blir sendt til Kartverket for videre prosessering. Kort tid etter blir 10 minuttss målinger tilgjengelig på nett. Observasjoner fra vannstandsmålere brukes til å beregne tidevannstabeller, studere havnivåendringer og beregne fremtidig havnivå.

Vannstandsmålere som logger vannstanden over lengre perioder har eksistert siden 1800-tallet. I Norge har man måleserier fra ca. 1915 for enkelte målere. De gamle måleseriene er sentrale i overvåking av havnivå- og klimaendringer [Kartverket, 2020].



Figur 3.1: Permanente vannstandsmålere i Norge.

Vannstandsmålerne har digitale avlesere og sender observasjoner til Kartverket automatisk. Det finnes flere typer vannstandsmålere langs Norskekysten, men flottørmåleren er mest brukt. Et rør (ofte omtalt som en brønn) med liten åpning i bunnen er plassert i havoverflata. Brønnen fungerer som et lav-pass filter, og vil fjerne de minste og bråeste endringene forårsaket av krusninger. I brønnen henger en flottør som er koblet til selve måleinstrumentet. Dette logger vannstanden i den frekvensen som er nødvendig for den aktuelle vannstandsmåleren. Med vannstandsmålerne hører det til et nivelleringspunkt. Dette blir brukt til å overvåke vannstandsmålerens høyde i en referanseramme.

Tabell 3.1: Vannstandsmålere langs Norges kyst. Koder for referanse i kommende plott.

Vannstandsmåler	Kode	Breddegrad	Lengdegrad
Viker	vike	59.0	10.9
Oslo	oslo	59.9	10.7
Oscarsborg	osca	59.7	10.6
Helgeroa	helg	59.0	9.9
Tregde	treg	58.0	7.6
Stavanger	stav	59.0	5.7
Bergen	berg	60.4	5.3
Måløy	malo	61.9	5.1
Ålesund	ales	62.5	6.2
Kristiansund	krin	63.1	7.7
Trondheim	tron	63.4	10.4
Heimsjø	heim	63.4	9.1
Mausund	maus	63.9	8.7
Rørvik	rorv	64.9	11.2
Bodø	bodo	67.3	14.4
Kabelvåg	kabe	68.2	14.5
Narvik	narv	68.4	17.4
Harstad	hars	68.8	16.5
Andenes	ande	69.3	16.1
Tromsø	trom	69.6	19.0
Hammerfest	hamm	70.7	23.7
Honningsvåg	honn	71.0	26.0
Vardø	vard	70.4	31.1

3.4 Geodetiske modeller

3.4.1 NKG2015

Geoiden som blir brukt i beregningen av dynamisk havtopografi er NKG2015. Det er NKG (Nordic Geodetic Commission) som står for utarbeidelsen av den og arbeidet startet i 2011. NKG er en organisasjon som består av de geodetiske fagmiljøene i landene Norge, Sverige, Danmark og Island, samt de baltiske landene. Geoidmodellen er en regional modell for de nordiske og baltiske landene. Modellen har et standardavvik på 0.028 m for hele området og 0.033 m for Norge Ågren et al. [2016], og regnes som den beste modellen tilgjengelig for Norskekysten og nærliggende områder i dag [Idžanović et al., 2017].

3.4.2 HREF2018B-NKG2015

Høydereferansemodellen HREF2018B-NKG2015 er produsert av Kartverket. En høydereferansemodell angir forskjellen mellom to høydesystemer og kobler i dette tilfellet sammen høyden for vannstandsmåler i nn2000 og ellipsoiden. Ulikt den offisielle HREF2018b som er beregnet ut fra geoidmodellen NMA2013v30 er denne beregnet fra NKG2015.

3.5 Maskinlæringsmodeller

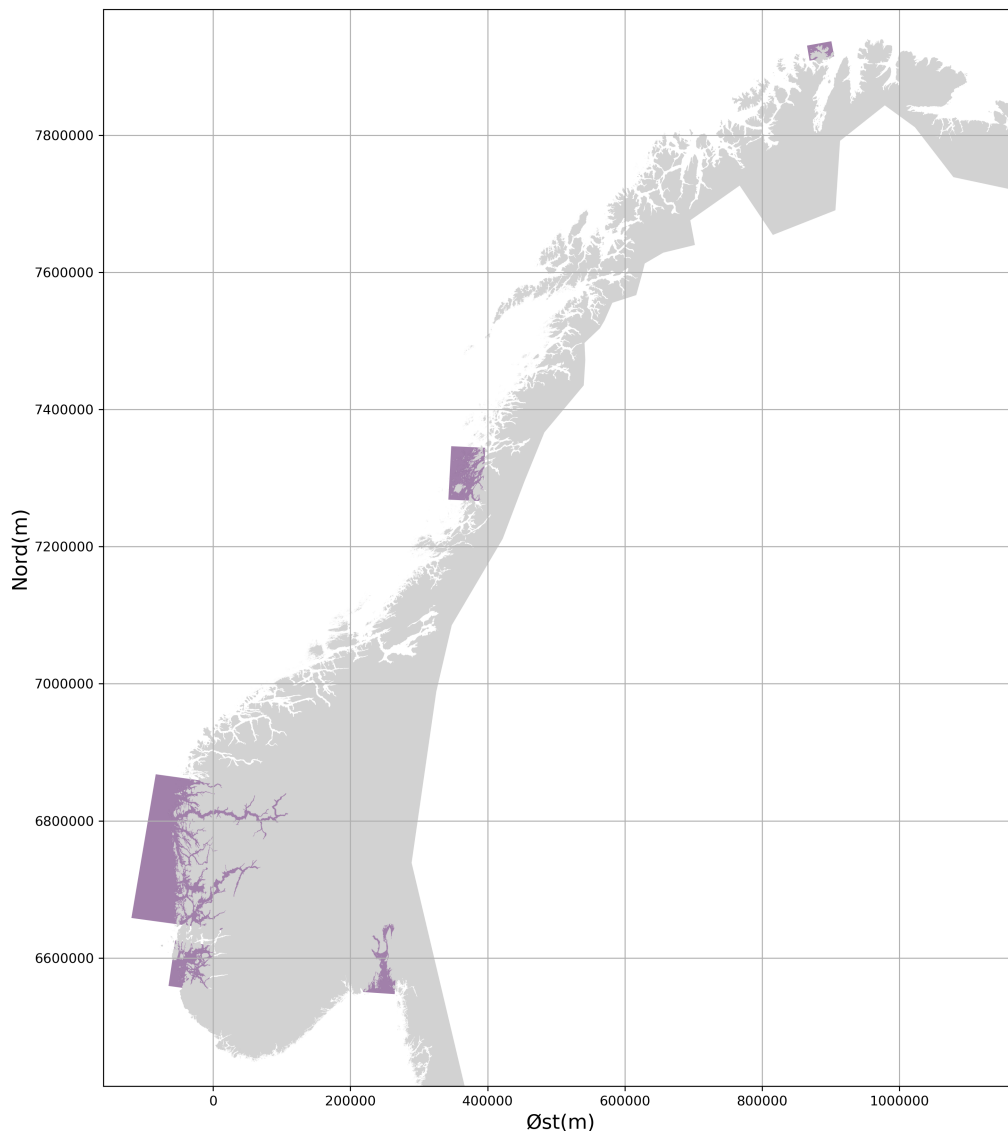
I denne oppgaven skal tre maskinlæringsmodeller undersøkes, de finner man i vedlegg A. Den første, *Dense*, er et fullt sammenkoblet nett og den enkleste av modellene. Deretter kommer *HSI-CNN* som er hentet fra Hu et al. [2015]. Den siste er *CNN*. *Dense* og *CNN* er utviklet i forarbeidene til

oppgaven. Alle modellene er forholdsvis enkle. Det kom fram ved utviklingen av *Dense* og *CNN* at mer komplekse modeller ikke gav bedre resultater, men heller førte til overtilpassing etter svært få epoker.

Årsaken til at hyperspektrale bilder (hyperspectral images, HSI) blir nevnt i denne oppgaven er likhetene mellom spekteret til en piksel i et HSI og pulsformen fra en altimetriobservasjon. Spekteret til en piksel består av signalstyrken til ulike bølgelengder. Styrt og ikke-styrt klassifisering av piksler basert på spekterets form eller verdier for enkelte bølgelengder i spekteret, er mye mer utbredt enn klassifisering av altimetridata. Det skal undersøkes om erfaringer fra HSI-klassifisering kan hjelpe å lage en god maskinlæringsmodell til klassifisering av altimetriobservasjoner.

3.6 Testområder

For å produsere et datasett som representerer store deler av Norskekysten uten å klassifisere alle pulser manuelt, har fem testområder blitt definert, se tabell 3.2. Både størrelse og topografi i testområdene varierer. De inkluderer fjorder, skjærgård, havbasseng og åpent hav. Også topografien rundt testområdene varierer, som vil påvirke returpulsens form. Det kan være høye fjell, svaberg, holmer og middels høye fjell for å nevne noe.



Figur 3.2: Testområder for klassifisering vist i fiolett.

Tabell 3.2: Havareal og forventet antall punkter per satellittsyklus. På grunn av variasjoner i satellittbanen kan antall observasjonspunkter variere.

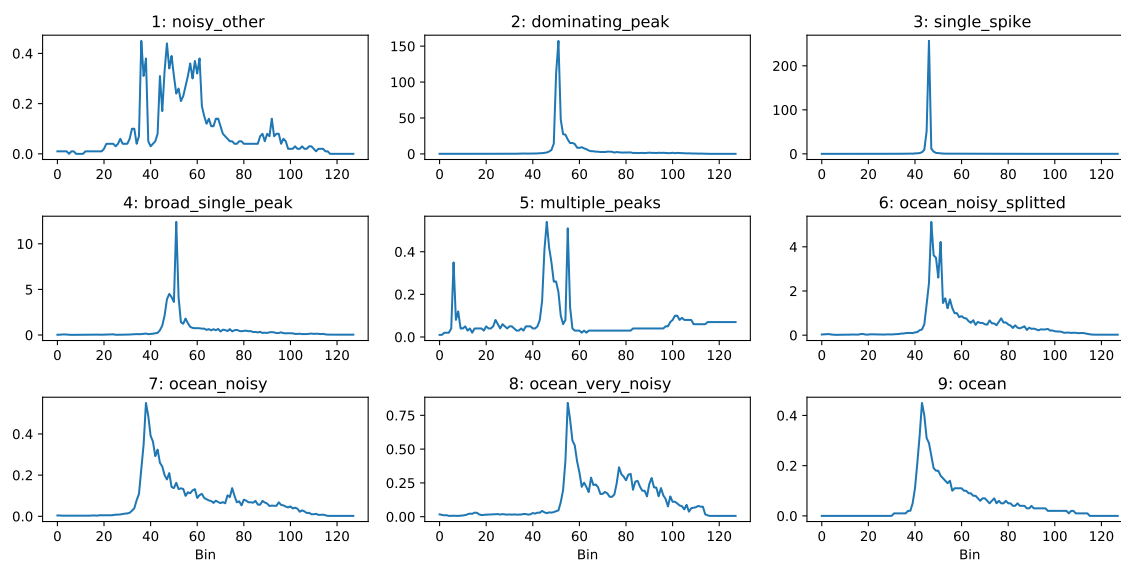
Område	Havareal(km ²)	Punkter per satellittsyklus
Sognefjorden	15600	680
Oslofjorden	1500	200
Vega	2900	240
Boknafjorden	2000	230
Finmark	460	100

For det største av de fem testområdene (Sognefjorden) kan altimetriobservasjoner opptil 60 km fra kysten bli inkludert. Dermed kunne noen av observasjonene være tjent med å ha korreksjonen utledet fra radiometerets beregninger. På tross av dette vil alle observasjoner bli behandlet likt for å være konsekvent og vil dermed få troposfæriske korreksjoner beregnet fra metrologisk modell.

3.7 Klassifisering av returpulsene

Programmet som blir brukt til å klassifisere returpulsene manuelt er utviklet av veileder. Programmet søker gjennom alle filene lastet ned fra FTP-serveren etter punkter innenfor de definerte testområdene. Returpulser med nadirpunkt på land blir filtrert bort ved hjelp av kystmodellen. Deretter må man plassere hver enkelt returpuls i riktig klasse. En figur med eksempelpulser fra hver klasse kommer automatisk frem når man starter programmet og legger til rette for å være mest mulig konsekvent med klassifiseringen, se figur 3.3. Som et ekstra hjelpemiddel har man et kart over det aktuelle passet man jobber med. Der vises også posisjonen til returpulsene som skal klassifiseres, se figur 3.4.

Når man har klassifisert en hel syklus for et gitt område kan man gå over igjen og sjekke returpulsene klassevis. Dette blir kalt reklassifisering. På den måten ser man fort om det er en returpuls som skiller seg ut og gir mulighet til å rette opp feil gjort i den opprinnelige klassifiseringen. Muligheten for feilklassifisering blir mindre. Det tydeliggjør også ulike grupperinger av returpulser innad i en klasse når man presenterer pulsene klassevis. De reklassifiserte returpulsene lagres i en tekstfil med posisjon, avstand til kyst, klasse, og deretter de 128 utvalgspunktene (samplingene) som representerer hver enkelt puls.

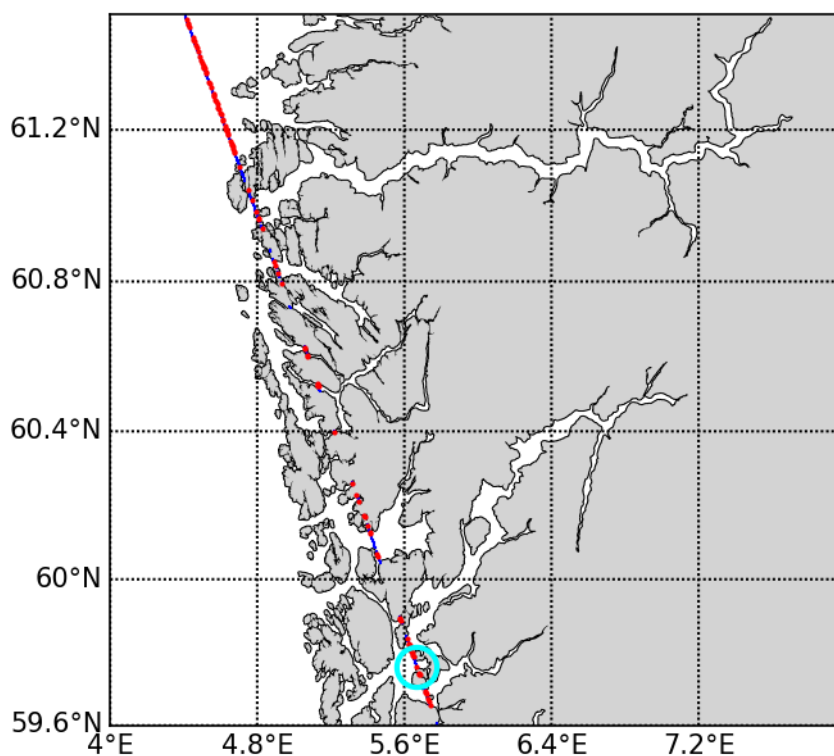


Figur 3.3: Eksempelpulser brukt som hjelpemiddel i klassifiseringsfasen.

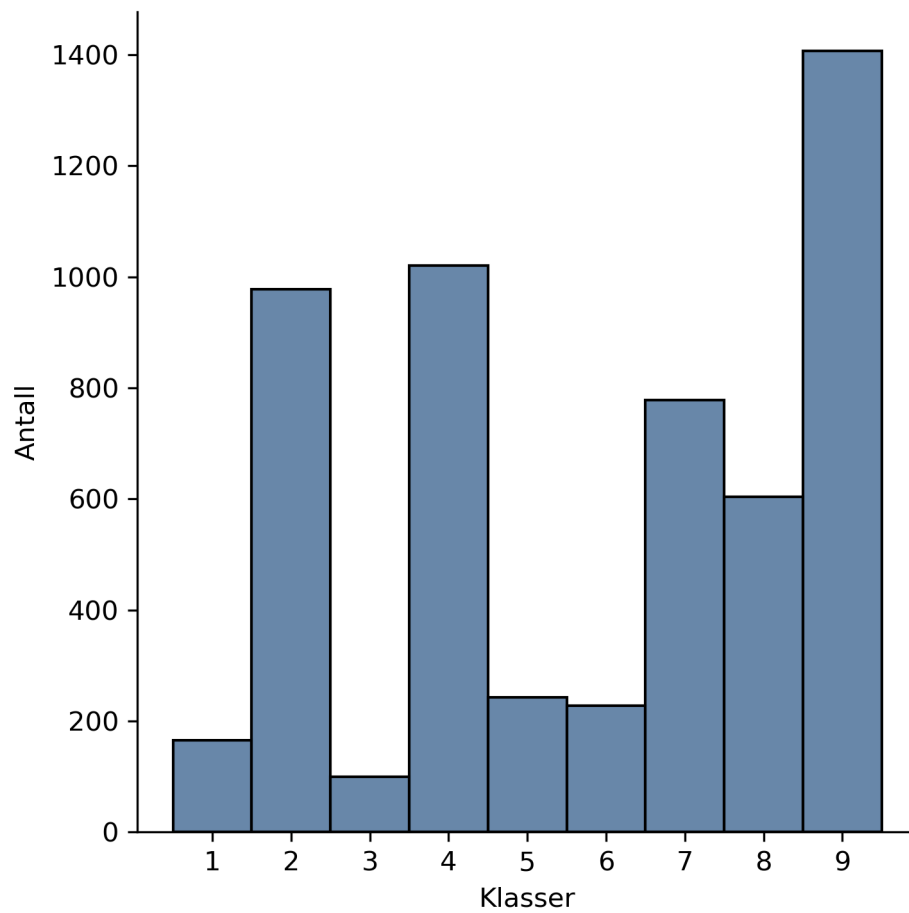
Det er ikke bare eksempelpulsene i figur 3.3 som danner et grunnlag for hvilken klasse en puls skal få, det er også en beskrivelse som inngår. Klassene har veileder definert basert på hvilke pulser man kan forvente å finne i kystsonen, se tabell 3.3. Ser man på klassefordelingen til treningsdatasettet i figur 3.5 er det tydelig at fordelingen er skjev.

Tabell 3.3: Hjelpemiddel for klassifisering av pulsene.

Klasse	Navn	Forklaring
1	Støy og annet	Pulsform som ikke tilhører noen andre klasser.
2	Dominerende peak	Pulsform med ren leading edge, dominerende peak og noe støy til høyre for denne.
3	Enkelt peak	En symmetrisk smal peak.
4	Bred enkeltpeak	Dominerende peak med støy til venstre og høyre.
5	Flere peaks	Flere peaks, spredt ut over utvalgspunktene.
6	Splittet støyete sjø	Ren leading edge, dominerende peak som er spittet, støyete trailing edge.
7	Støyete sjø	Ren leading edge, litt støyete trailing edge.
8	Veldig støyete sjø	Sjøform med mye støy over hele.
9	Sjø	Ren sjø, åpent hav.

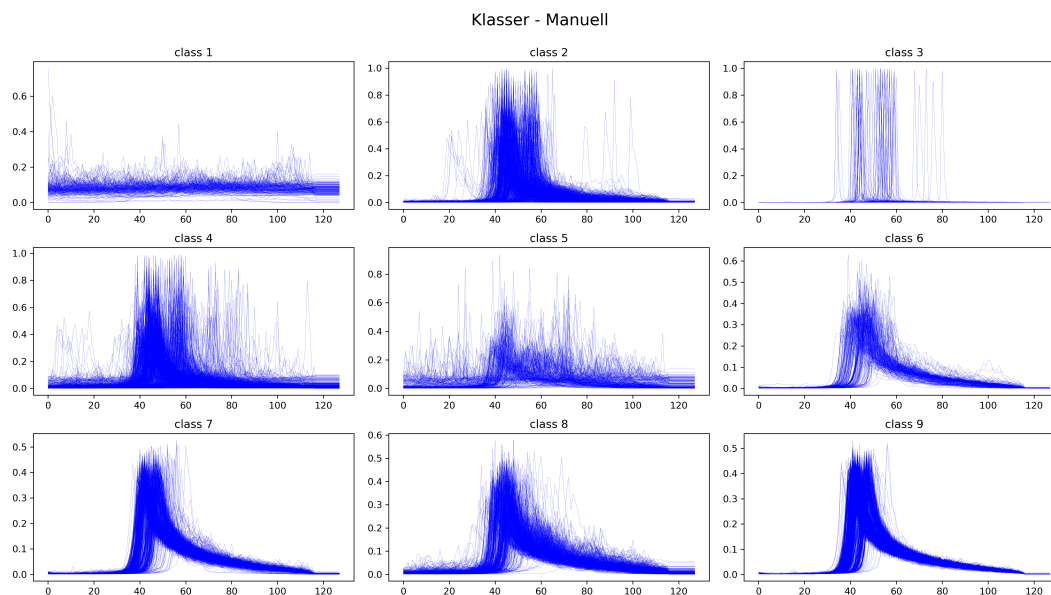


Figur 3.4: Kartutsnitt som viser hvor den aktuelle pulsen som skal klassifiseres befinner seg (lyseblå), hvilke pulser som har blitt valgt ut for klassifisering i passet (rød) og fotavtrykket til satellitten (mørkeblå). Testområde: Sognefjorden



Figur 3.5: Klassefordeling i datasettet

I figur 3.6 ser man at flere av klassene har pulser hvor starten av pulsen fordeler seg ut over utvalgs-punktene, eller gruppering av ulike pulsformer. Dette forventes å bli en utfordring for modellen. Samtidig forventes observasjonene i valideringsområdene å ha de samme grupperingene.



Figur 3.6: Klasser fra manuell klassifisering.

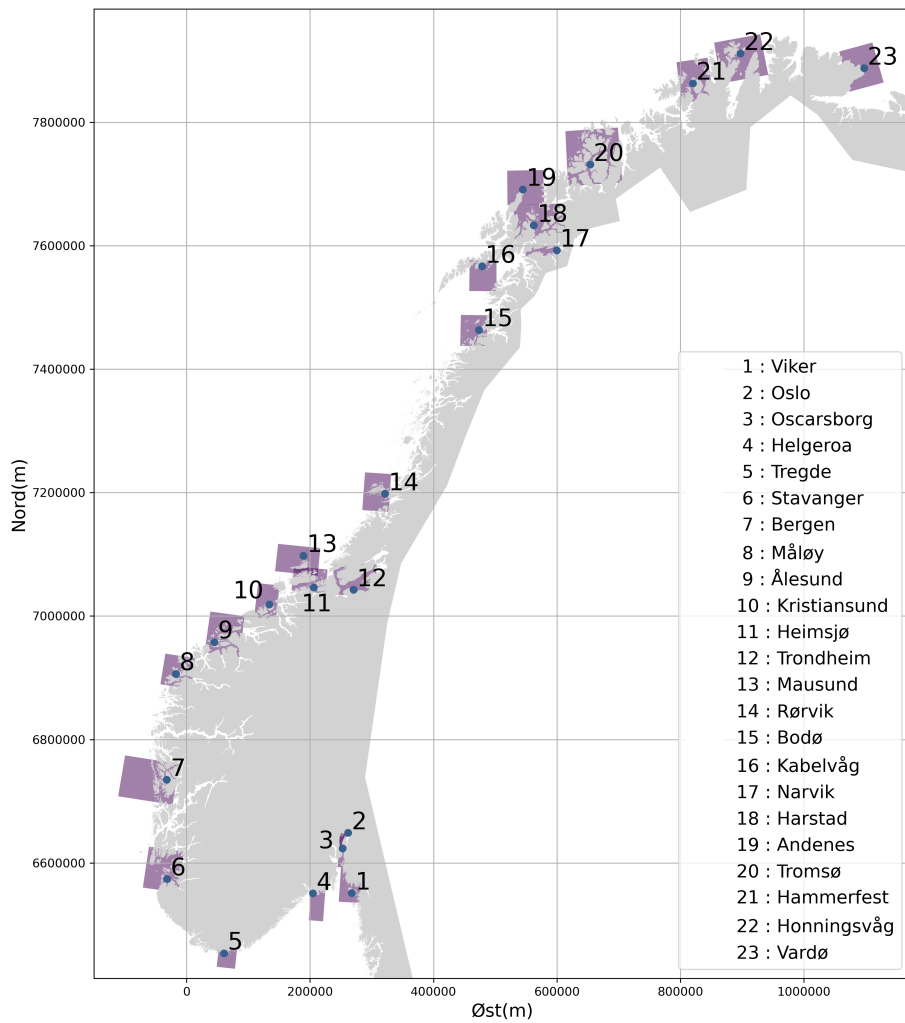
3.8 Valideringsområder

I valideringsområdene skal altimetriobservasjoner predikeres av en maskinlæringsmodell, samtidig som de går gjennom et grovfeilsøk med multipel-t test. Deretter vil dynamisk havtopografi bli beregnet basert på predikeringen og resultatet av grovfeilsøket. Når man definerer valideringsområder rundt en vannstandsmåler må man balansere avstand til vannstandsmåleren og antall observasjoner. Om man velger et stort område vil man få flere observasjoner. For observasjoner langt unna kan havtopografien, vær og oseanografiske forhold være annerledes enn ved vannstandsmåleren. Om man velger et lite område får man få observasjoner til å bestemme dynamisk havtopografi.

Breili et al. [2017] velger altimetriobservasjoner som er innenfor en sfærisk grad i forhold til vannstandsmåleren. På den måten vil alle vannstandsmålerne bli behandlet likt uavhengig av topografi rundt vannstandsmåleren. Idžanović et al. [2018] bruker en annen tilnærming for CryoSat-2 observasjoner. CryoSat-2 satellitten har en repetisjonsfrekvens på 369 dager, med undersykluser på ca. 30 dager. Det førte til at de brukte 45 km x 45 km bokser hvor vannstandsmåleren måtte minst være 0.2 sfæriske grader fra boksens grense. På denne måten balanserer boksen romlig oppløsning og tidsoppløsning.

Metoden brukt for denne oppgaven har ikke en fast rutine for å lage utvalgsområder som de to overnevnte metodene i Breili et al. [2017] og Idžanović et al. [2018]. På grunn av høy repetisjonsfrekvens og satellittspor som avviker lite (± 1 km), vil et pass som er inkludert i valideringsområdene en gang, bli inkludert for hver syklus. Valideringsområdene har blitt valgt ut manuelt med tanke på hvor bakkespor til satellitten befinner seg. Man har også unngått å ta med observasjoner fra andre fjorder eller havbasseng som ikke har en direkte kobling til vannstandsmåleren via et sund eller liknende. De fleste områdene vil ha tilgang til pulser på åpen sjø, men i noen tilfeller ligger vannstandsmåleren i bukter, eller har øyer og holmer som ikke gir tilgang til pulser på åpen sjø. Merk at vannstandsmåler 2, 3, 12 og 17 i figur 3.7 ikke forventes få pulser av klasse 7, 8 og 9 fordi det ikke er åpent hav i valideringsområdet.

Resultatene fra Breili [2021] indikerer at midlet dynamisk havtopografi stiger innover fjordsystemet med 5 til 10 cm. Det kan forklares ved at den norske kyststrømmen blir avvøyd mot land av corioliseffekten, og avrenning av ferskvann[Breili, 2021]. Topografien for de permanente vannstandsmålerne er ikke like ekstrem som i Sognefjorden, men en liknende effekt kan likevel være gjeldene. Tidevann kan også forsinkes av topografien, som vil innføre feil i sammenlikningen med dynamisk havtopografi fra vannstandsmålere.



Figur 3.7: Vannstandsmålere med tilhørende valideringsområder.

3.9 Validering ved dynamisk havtopografi

Valideringen vil foregå ved å sammenlikne dynamisk havtopografi, hvor fasiten er beregnet fra vannstandsmåler-observasjoner. På den måten vil både multippel-t test og maskinlærings-klassifisering som filtreringsmetoder bli validert individuelt. Samtidig kan metodene sammenliknes for å avgjøre hvilken som egner seg best til filtrering av S3-PEACHI data i kystsonen.

Dynamisk havtopografi for vannstandsmålerne blir beregnet i koordinatene til den aktuelle vannstandsmåleren. Dynamisk havtopografi for altimetriobservasjonene beregnes der observasjonene befinner seg innenfor valideringsområdene. For hvert pass satellitten gjør over et valideringsområde vil medianen av godkjente returpulser bli lagret. Med godkjente returpulser menes de som ikke har blitt forkastet i multippel-t test eller har en klasse som er en del av klassekombinasjonen som skal undersøkes. Dette gjentas for hvert pass, for hver vannstandsmåler og for hver klassekombinasjon som skal testes. Medianene fra begge metodene sammenliknes med fasit. Fra sammenlikningen blir differanse og standardavvik for hele tidsperioden beregnet. Prestasjonsmål vil for hver vannstandsmåler, samt et kystmiddel, bli gitt som: differanse, standardavvik, Spearman korrelasjon og MAD. Man kan ut fra dette avgjøre hvilken metode som egner seg til hver enkelt vannstandsmåler og for hele Norskekysten.

Med utgangspunkt i ni klasser vil antall unike kombinasjoner uten repetisjon bli 511. Beregning av dynamisk havtopografi vil ta svært lang tid lang tid. Basert på at klassene er valgt på gitte kriterier, vil klassekombinasjoner også bli valgt på gitte kriterier. For eksempel vil rene klasser være interessant å se på sammen. For eksempel vil kombinasjoner av klasse 2, 3 og 9 være av interesse.

4 Resultater

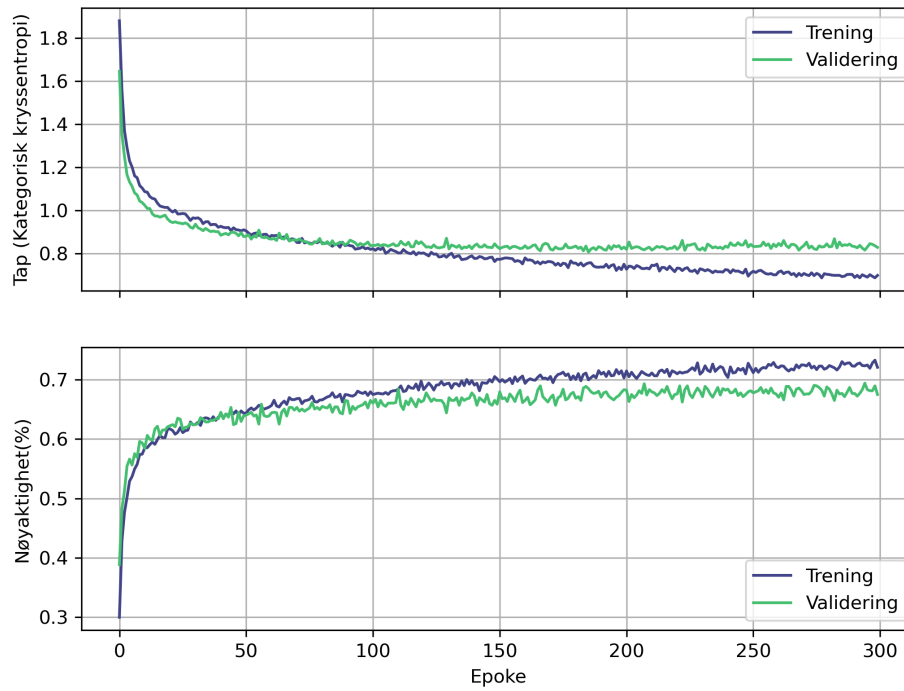
I dette kapittelet vil resultatene fra treningen av maskinlæringsmodeller. Ut fra resultatene skal en maskinlæringsmodell velges til å predikere altimetriobservasjoner fra valideringsområdene. De predikerte klassene vil bli brukt i ulike kombinasjoner til å beregne dynamisk havtopografi. Dynamisk havtopografi beregnet fra klassifiserte altimetriobservasjoner vil bli sammenliknet med dynamisk havtopografi beregnet fra vannstandsmåler-observasjoner. På den måten vil man avdekke hvilke klassekombinasjoner som egner seg best til beregning av dynamisk havtopografi langs Norskekysten. Til slutt vil validering ved dynamisk havtopografi bli presentert.

4.1 Klassifiseringsnøyaktighet

Kurver for tapsfunksjon og nøyaktighet som blir presentert er basert på 300 treningsepoker. Man forventer overtilpassing etter 300 epoker, men man får et godt innblikk i hvordan modellen utvikler seg. Utviklingen til modellen gir grunnlag for å bestemme hvor mange epoker den enkelte modellen bør trenes på for å oppnå det beste resultatet. Treningskurvene til tapsfunksjonen og nøyaktighet må sees i sammenheng med valideringskurvene. Når valideringskurven for tapsfunksjonen går over treningskurven indikerer det at modellen overtilpasser i epokene som kommer etter krysningspunktet. For nøyaktighetskurven gjelder det samme når treningskurven går over valideringskurven.

4.1.1 *Dense*

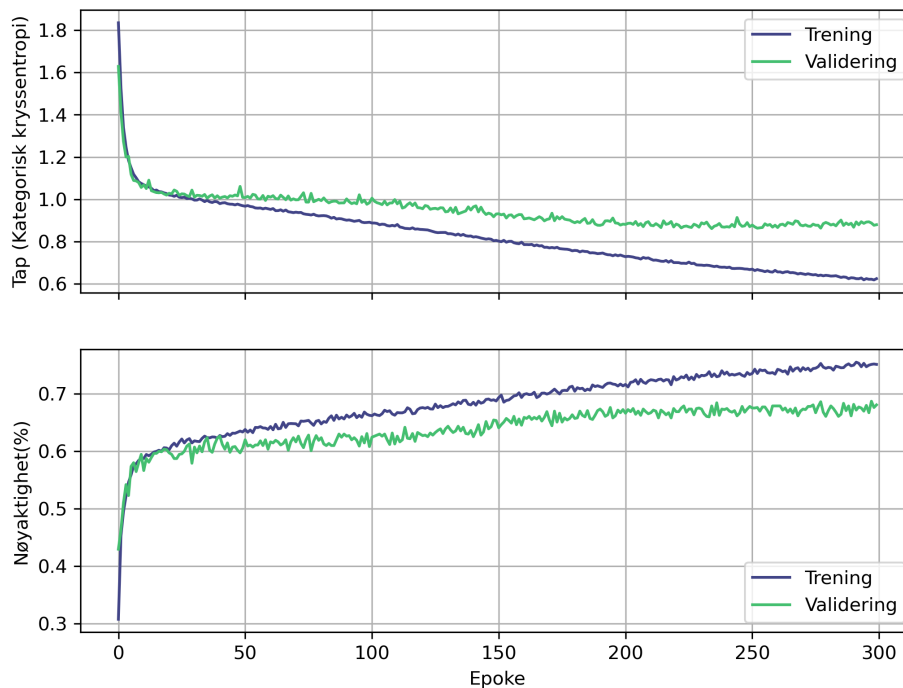
Figur 4.1 brå utvikling av *Dense*-modellen i de første epøkene før kurvene flater ut. Det kan se ut som tapsfunksjonene for treningsdata og validering krysser ved ca. 75 epoker. Forskjellen på nøyaktighet mellom treningsdata og valideringsdata er lite ved 75 epoker.



Figur 4.1: Utvikling for *Dense*-modell.

4.1.2 HSI CNN

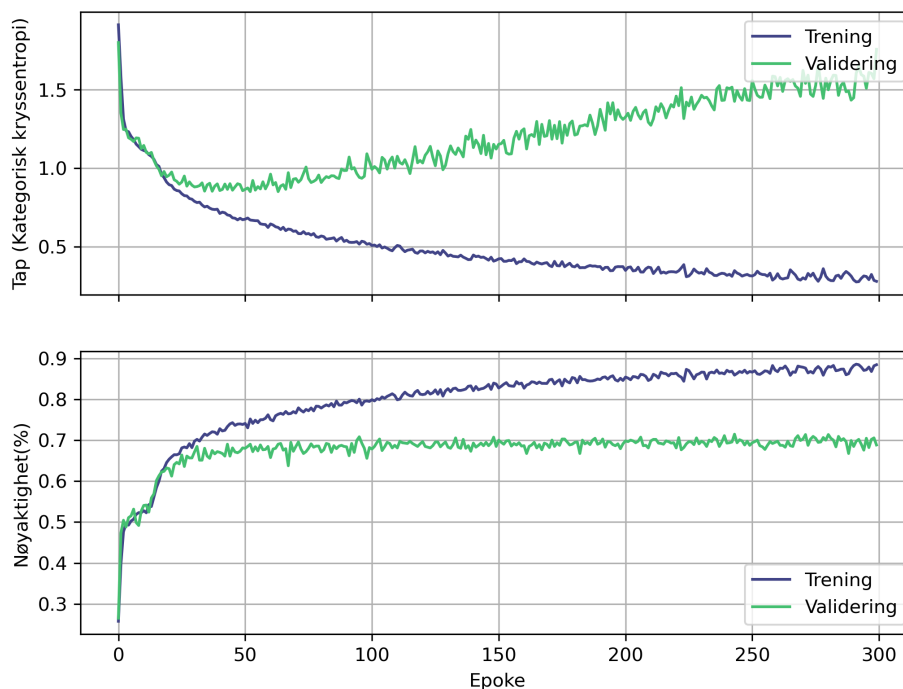
I figur 4.2 ser man en liten forbedring rundt epoke 150 i både tapsfunksjonen og nøyaktighetsfunksjonen for valideringsdata. Man risikerer noe overtilpassing, men forvirringsmatrisen vil avdekke om det gir betydelig utslag. Valideringsnøyaktigheten er uansett tilfredsstillende på ca. 65% ved epoke 150.



Figur 4.2: Utvikling for HSI CNN-modell.

4.1.3 CNN

Grafene i figur 4.3 skiller seg ut fra grafene i figur 4.1 og 4.3. Avstanden mellom grafene er større, og tapsfunksjonene stabiliserer seg ikke i forhold til hverandre. Her velger jeg å trene den endelige modellen på 50 epoker. Ved 50 epoker har modellen lavest verdi i tapsfunksjonen for valideringsdata. Det er heller ikke for stor avstand mellom treningsnøyaktighet og valideringsnøyaktighet.



Figur 4.3: Utvikling for CNN-modell.

Med antall treningsepoker bestemt blir modellene evaluert med testdatasettet, se tabell 4.1. Alle modellene predikerer med over 60% nøyaktighet. Videre vil resultatene fra predikering av testdata bli satt inn i forvirringsmatriser for å velge den beste modellen.

Tabell 4.1: Resultater fra trening av maskinlæringsmodeller på testdatasett med endelig antall epoker.

	Dense	HSI-CNN	CNN
Stoppet ved epoke	75	150	50
Test nøyaktighet (%)	66.3	63.4	68.2
Test tap	0.864	0.943	0.875

4.1.4 Forvirringsmatriser

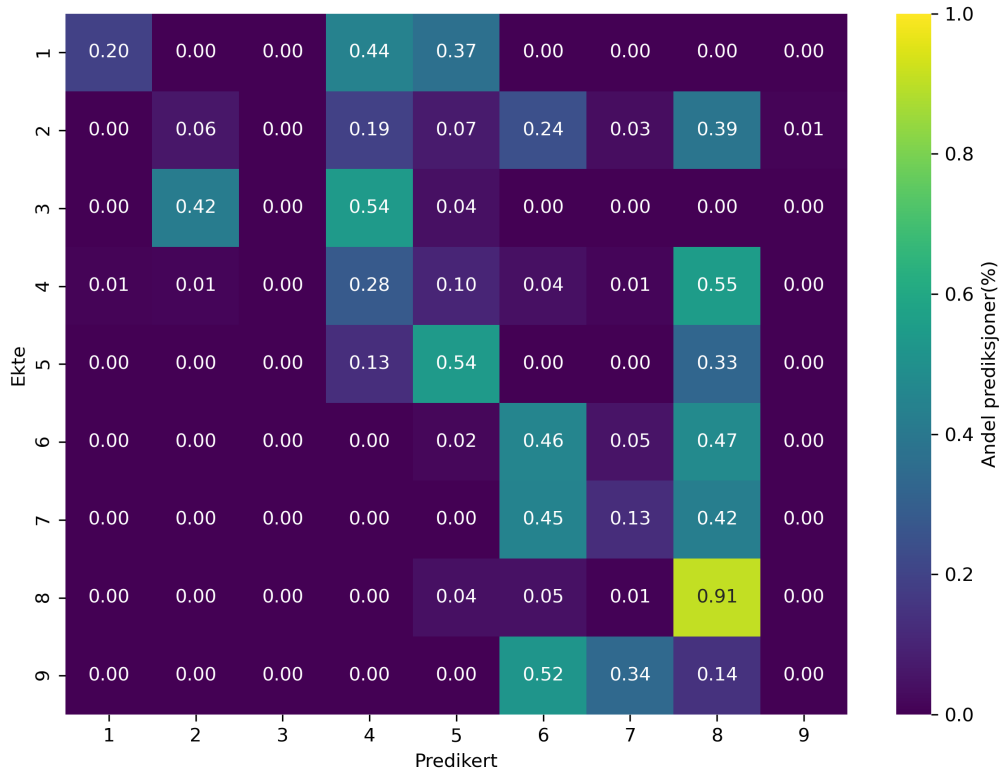
Forvirringsmatrisene må sammenliknes med hverandre for å velge ut den beste modellen. I tillegg må man ta hensyn til klassefordelingen i figur 3.5. Verdiene i matriseelementene er avrundet. Prediksjonene i forvirringsmatrisene er gjort på testdatasettet.

Matrisen for *Dense* i figur 4.4 har ikke diagonalmønsteret vi ønsker. Den største og viktigste klassen, klasse 9, har ingen riktige klassifiserte pulser. Foruten klasse 8 og 5 presterer modellen svært dårlig. *Dense*-modellen blir derfor ikke valgt til predikering av returpulser i valideringsområdene.

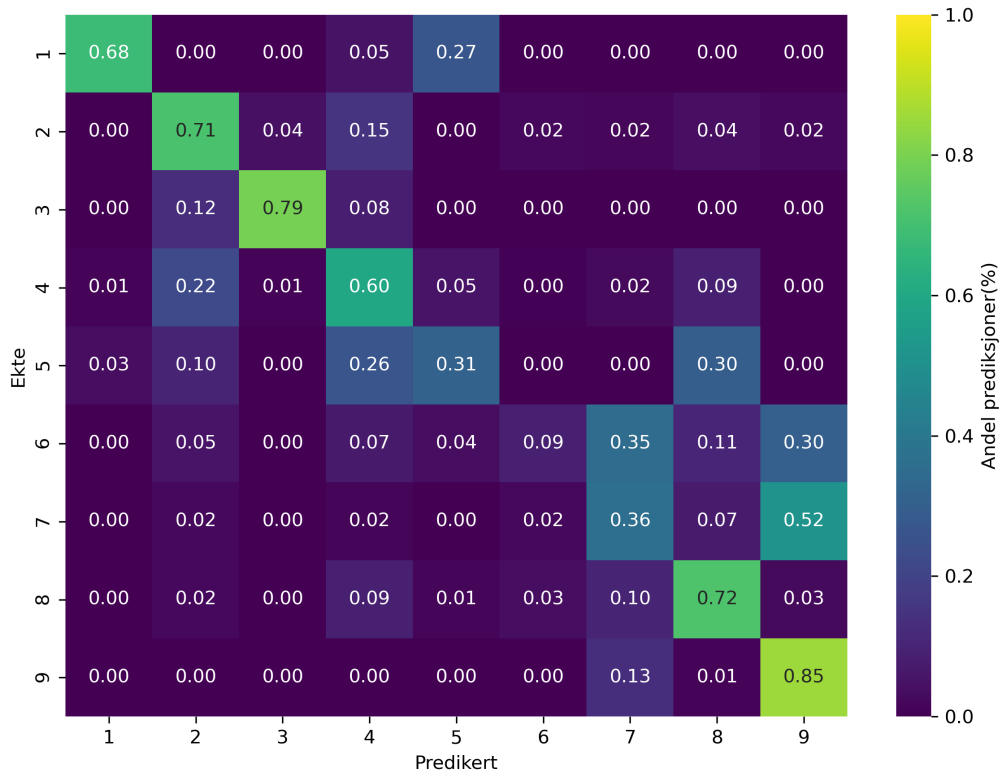
Dette er et eksempel på hvordan nøyaktighetstallet kan være misvisende når datasettet har skjev klassefordeling (se tabell 4.1).

Matrisen for *CNN HSI*-modellen i figur 4.5 er nærmere diagonalmønsteret vi ønsker enn i figur 4.4. Foruten klasse 5, 6 og 7 har *CNN HSI* predikert over 60% korrekt på testdata. Over 50% av pulsene i klasse 7 har blitt predikert som klasse 9. Denne feilpredikeringen var forventet fordi pulsene til klasse 7 og 9 er svært like. Klasse 2 og 4 smitter over i hverandre, men ikke i like stor grad som klasse 7 og 9.

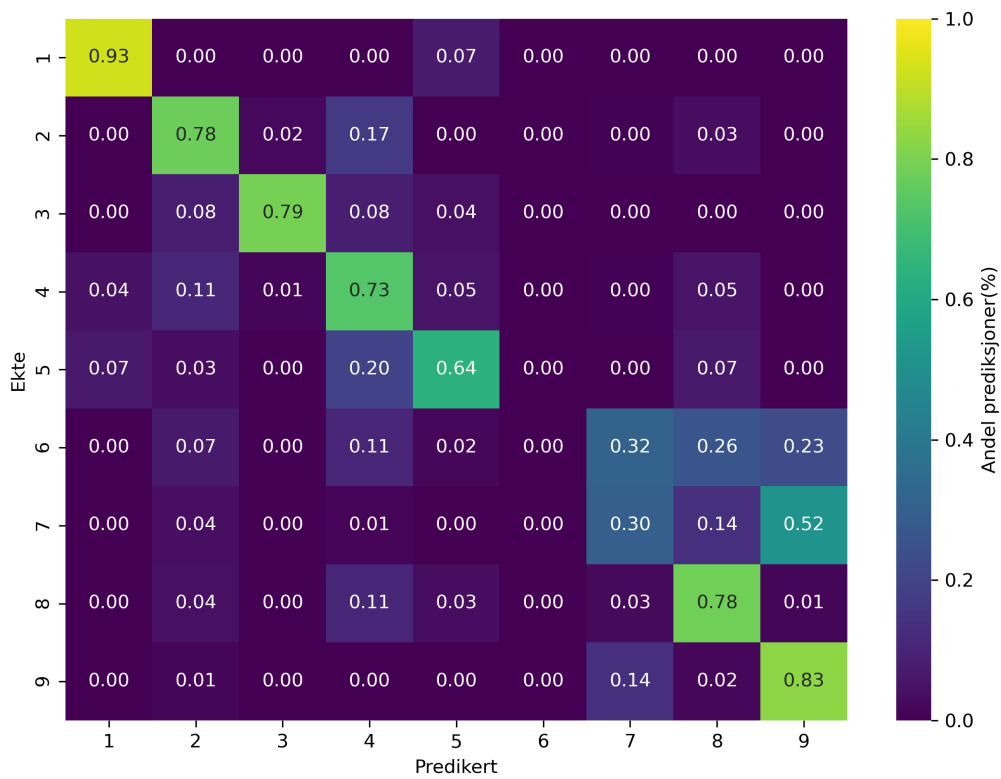
Om man sammenlikner matrisene til *CNN HSI* i figur 4.5 og *CNN* i figur 4.6 ser man klare likhetstrekk. Matrisen i figur 4.6 har høyere diagonalverdier enn i figur 4.5. Klasse 6 får tilsynelatende ingen korrekte prediksjoner og fordeles seg hovedsakelig ut i klasse 7-9.



Figur 4.4: Forvirringsmatrise for *Dense*-modell.



Figur 4.5: Forvirringsmatrise for *CNN HSI*-modell.



Figur 4.6: Forvirringsmatrise for *CNN*-modell.

Med hensyn til forvirringsmatrisene blir *CNN*-modellen valgt som den beste av modellene som baserer seg på styrt klassifisering. Det kan se ut som den presterer marginalt bedre enn *HSI-CNN*-modellen om man sammenlikner forvirringsmatrisene, til tross for betydelig bedre nøyaktighetstall i tabell 4.1.

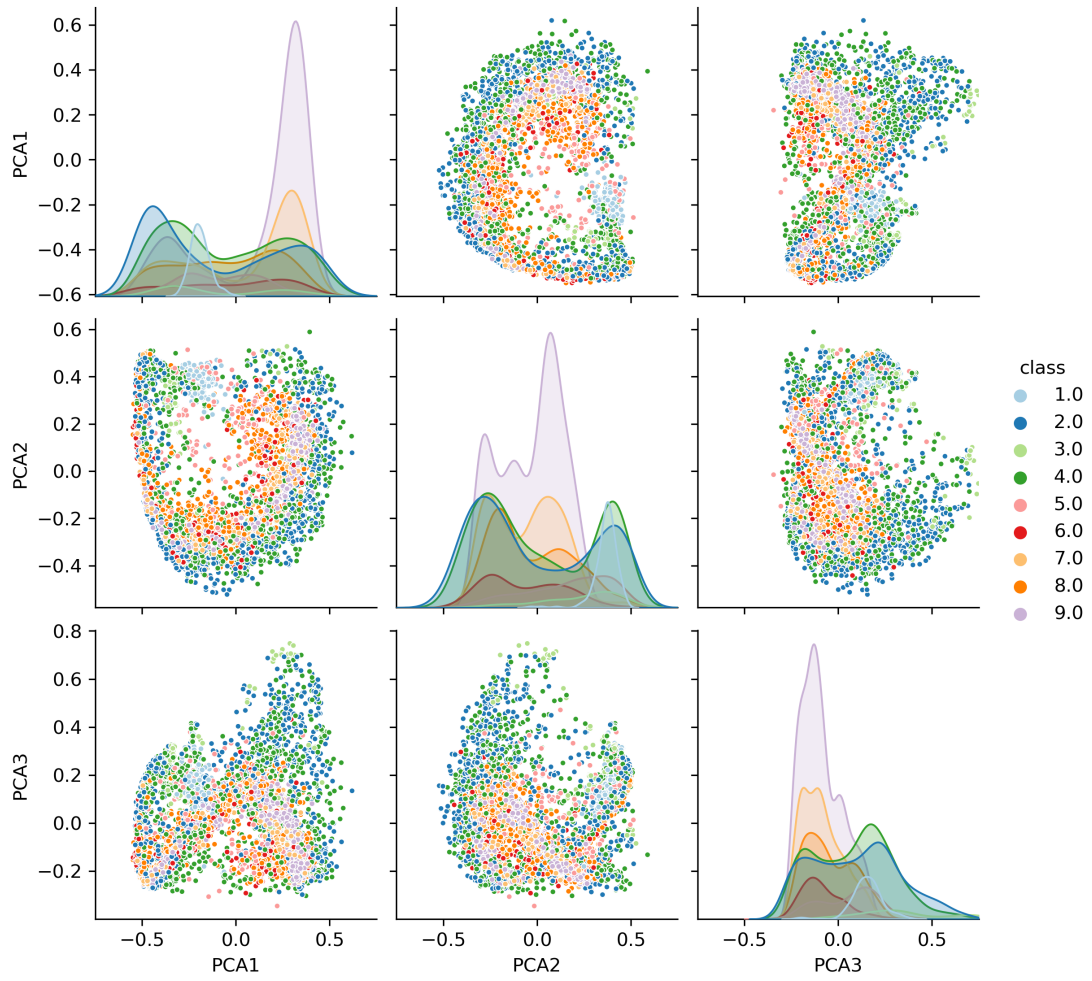
4.1.5 Clustering resultater

Her skal resultater for cluster-algoritmen K-Means presenteres. Fordi metoden er ikke-styrt skal grupperingene som algoritmen finner sammenliknes med de manuelt klassifiserte returpulsene i figur 3.6. I sammenlikningen vektlegges egenskaper til de definerte klassene i tabell 3.3.

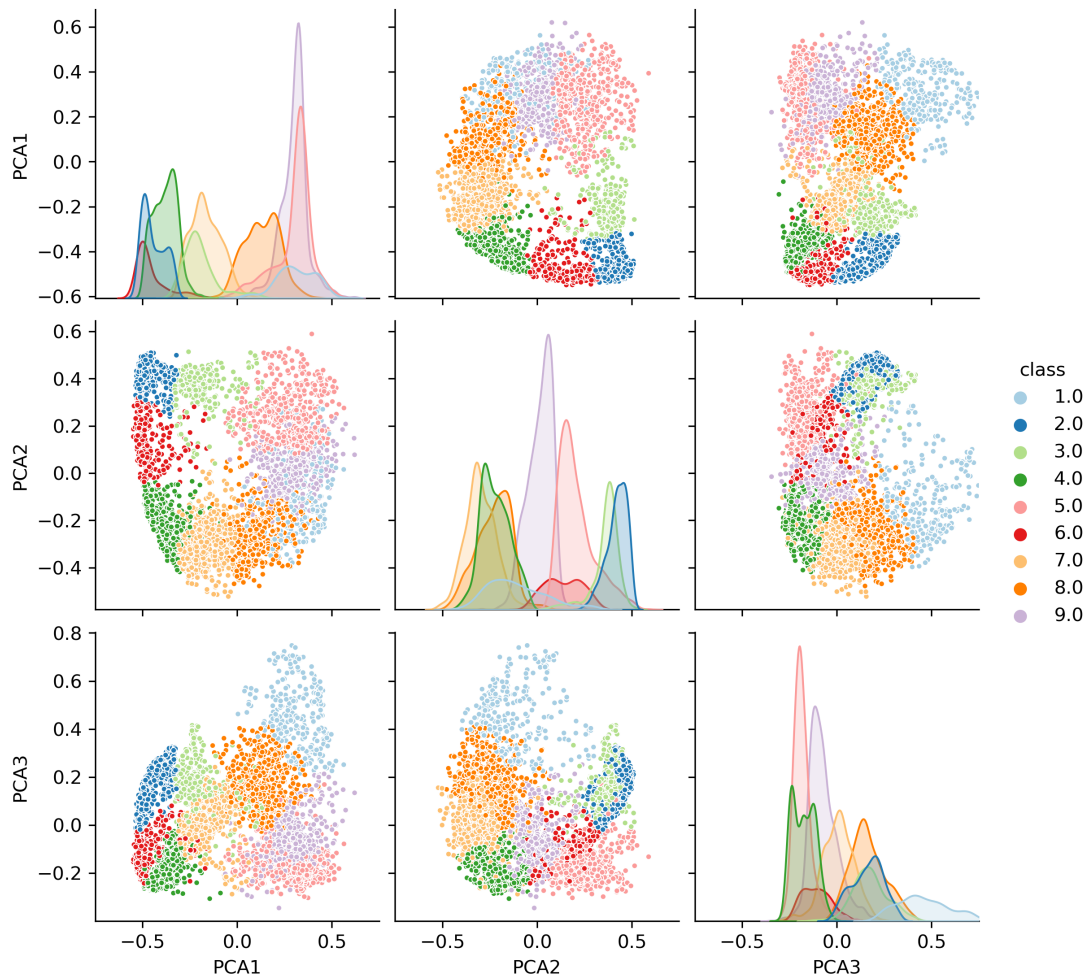
Allerede ved å se på prinsipal komponentene plottet i figur 4.7 ser man at det kun er en samling (cluster) av punkter. Punktene og grafene i figur 4.7 er fargelagt med de manuelt klassifiserte returpulsene. Diagonalplottene viser spredningen til klassene innenfor en prinsipal komponent. Grafene i diagonalplottene overlapper hverandre, som betyr at ingen klasse skiller seg ut fra de andre innenfor en prinsipal komponent. Plottene utenfor diagonalen viser prinsipale komponenter plottet mot hverandre. Det er i disse man ser en stor samling av punkter. Man ser tydelig at klassene fordeler seg ut over alle prinsipal komponentene i diagonalplottene, og at det ikke er en tydelig samling i plottene utenfor diagonalen.

I figur 4.8 er punkter og grafer fargelagt med klasser predikert av K-Means. Ideelt sett skulle man fått de samme mønstrene som i figur 4.7. I diagonalplottene til figur 4.8 ser man en mindre spredning innad i klassen, men grafene overlapper hverandre til en viss grad. Utenfor diagonalen ser man at gruppering av klasser ikke stemmer overens med grupperingene i figur 4.7.

Totalt ble 30 prinsipale komponenter beregnet og inkludert i Kmeans-algoritmen. På den måten tar man vare på det meste av informasjonen (ca. 95%), samtidig som man reduserer antall attributter cluster-algoritmen må ta hensyn til, se figur 2.10. I figur 4.8 ser man de samme prinsipal komponentene som i figur 4.7 men fargelagt med klasser funnet av cluster-modellen. Når man sammenlikner figurene ser man at modellen ikke finner de samme grupperingene som kommer av den manuelle klassifiseringen.

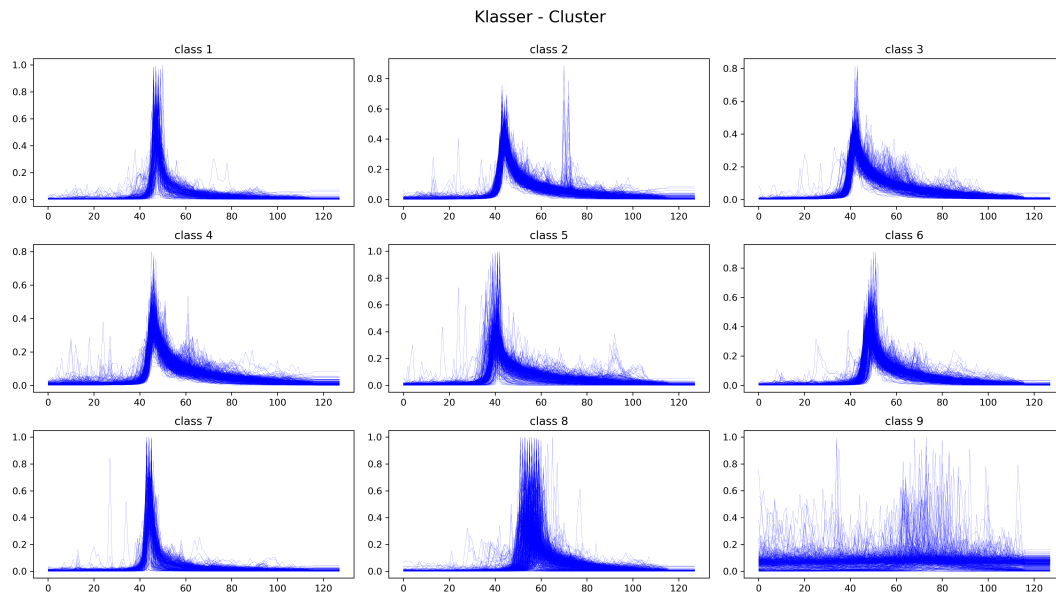


Figur 4.7: Prinsipal komponenter fargelagt med klasser fra manuell klassifisering.



Figur 4.8: Prinsipal komponenter fargelagt med klasser predikert av Kmeans.

Kmeans-algoritmen klarte ikke å finne samlinger av pulser med ønskede egenskaper, se tabell 3.3. I figur 4.9 ser man klassene (samlingene) som cluster-algoritmen har kommet frem til. Alle klassene ser relativt like ut med unntak av klasse 9. Klassenummerne i figur 4.9 relaterer seg ikke til klassenummerne fra den manuelle klassifisering. I klasser 1-8 ser man at det er relativt mye støy i utvalgspunktene mellom null og 30. Dette er ikke ønskelig. En ren start på returpulsen er et av kravene til flere klasser fra den manuelle klassifisering. I klasse 1-9 er det ulik utstrekning og startpunkt på returpulsen mellom klassene. Forskjell på startpunktet innad i en klasse er mindre for cluster-klasser enn manuelle klasser. Det indikerer at ulik plassering på startpunktet har ført til at pulser med relativ lik form har blitt plassert i forskjellige klasser av cluster-algoritmen.



Figur 4.9: Klasser basert på cluster.

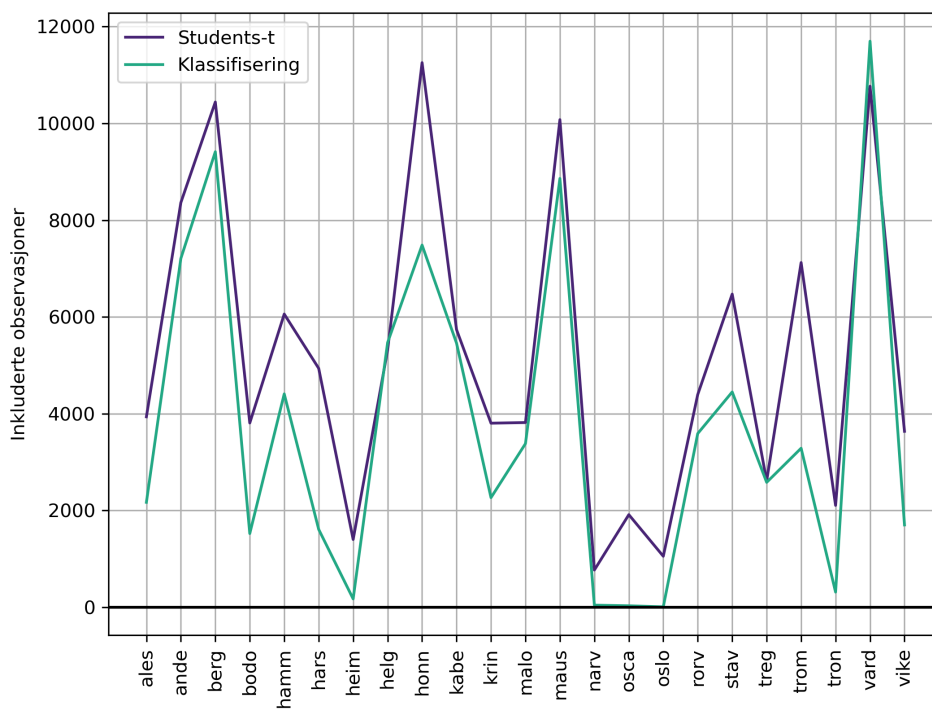
På grunn av den skjeve klassefordelingen i datasettet må man vektlegge forvirringsmatrisene mest når man skal gjøre dette valget. Selv om K-Means ga resultater har ingen av klassene de egenskapene vi i utgangspunktet ønsker for beregning av dynamisk havtopografi. Derfor blir *CNN* valgt til å predikere returpulsene i valideringsområdene.

4.2 Dynamisk havtopografi

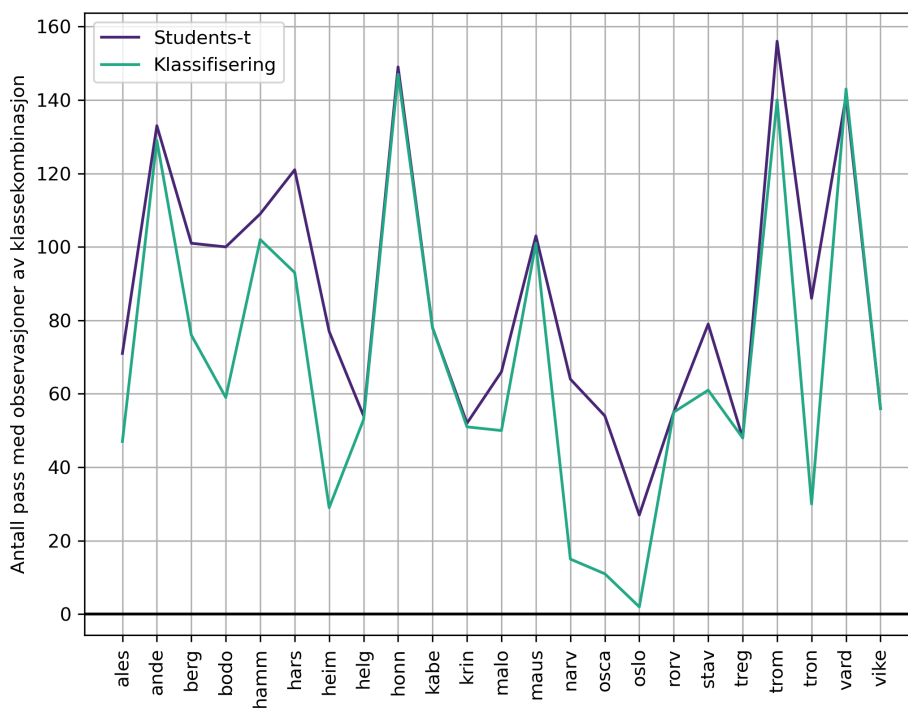
Her vil dynamisk havtopografi fra både Students multippel-t test og prediktering av returpulser ved *CNN* bli beregnet og sammenliknet med fasit fra vannstandsmålerne. Deretter vil metodene settes opp mot hverandre for å avgjøre hvilken som egner seg best for filtrering av S3-PEACHI-data langs Norskekysten.

Etter å ha testet flere klassekombinasjoner for å beregne dynamisk havtopografi ble det klart at det kun var klasse 9 som presterte bedre enn multippel-t test. Dette var også forventet med tanke på at klasse 9 skal representere åpen sjø, noe retrackeren er forventet å håndtere godt. Ut fra dette kan det virke som at retrackeren ikke klarer å hente ut korrekt avstand fra de andre pulsklassene.

Ved å kun bruke predikert klasse 9 returpulser i beregningen fikk enkelte valideringsområder svært få observasjoner, se figur 4.10. Dette var også forventet, da enkelte valideringsområder ikke inneholder åpen sjø, se figur 3.7. På tross av svært få returpulser i klasse 9 presterte disse bedre enn multippel-t test, som hadde betydelig flere observasjoner inkludert. De få observasjonene i klasse 9 kan komme av spesielle sjø- og værforhold, samt ulik plassering av fotavtrykket for hvert pass. En grense på 1000 observasjoner og 20 pass blir satt for videre presentering av resultater. På den måten vil de midlede resultatene for de resterende valideringsområdene være representative. Grensene kan virke noe konservative, men fordi datasettet består av observasjoner over lang tid er det nødvendig. Oslo, Oscarsborg, Narvik, Trondheim og Heimsjø blir som følge av dette fjernet fra beregning av prestasjonstall.

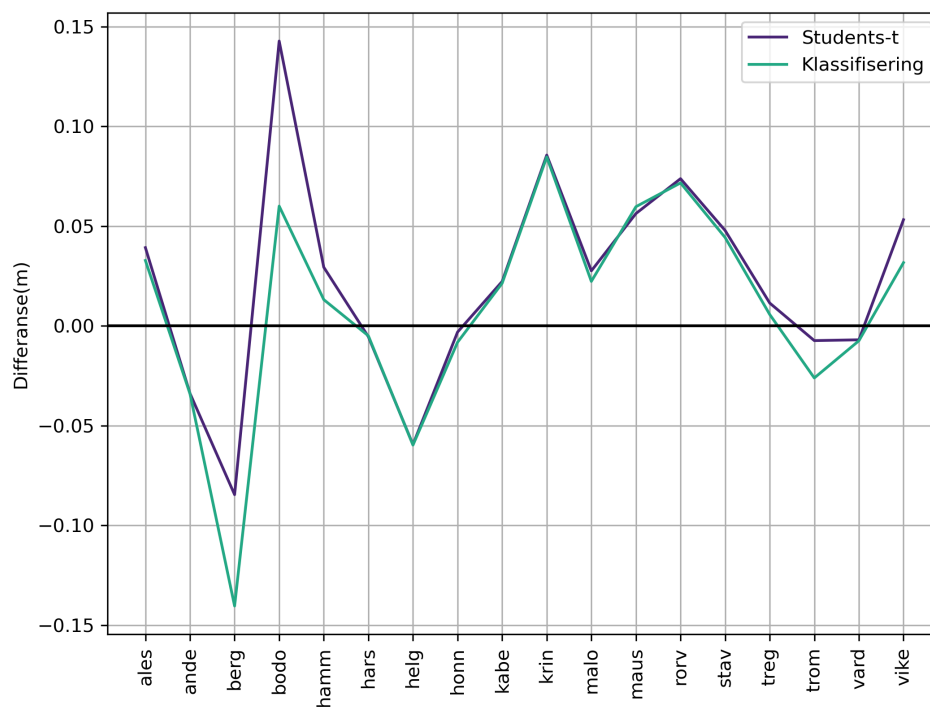


Figur 4.10: Inkluderte observasjoner for hele tidsperioden

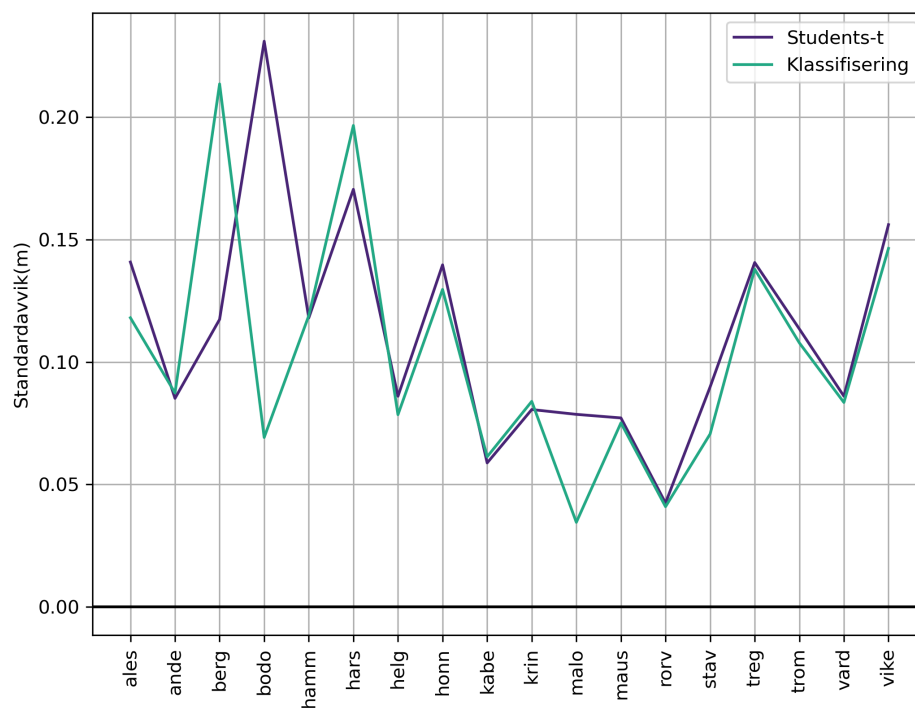


Figur 4.11: Inkluderte pass for hele tidsperioden

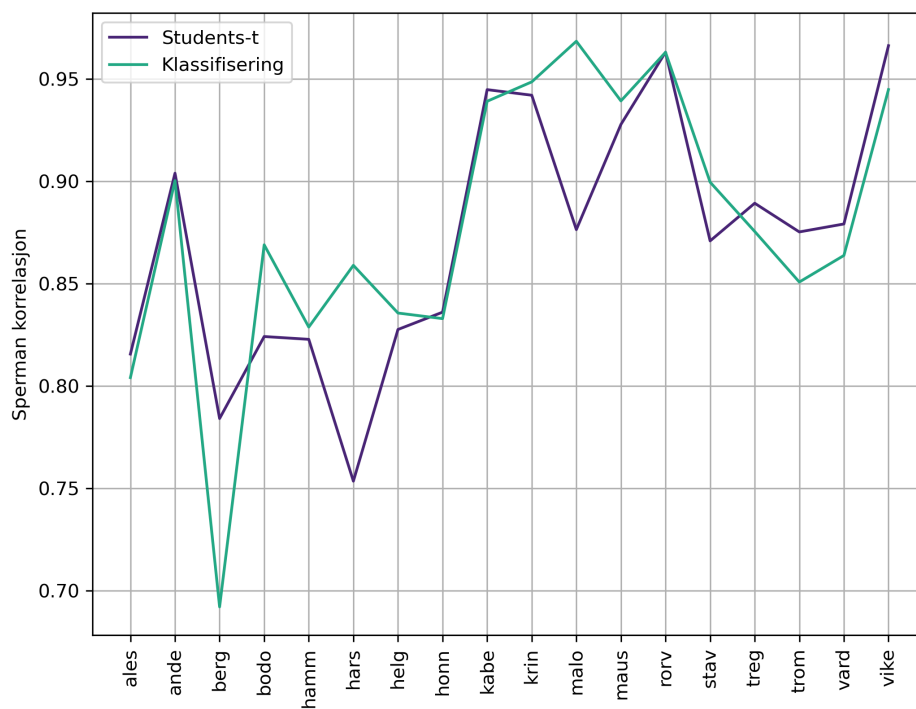
I figur 4.12-4.15 er valideringsområder med færre enn 1000 observasjoner og 20 pass ekskludert.



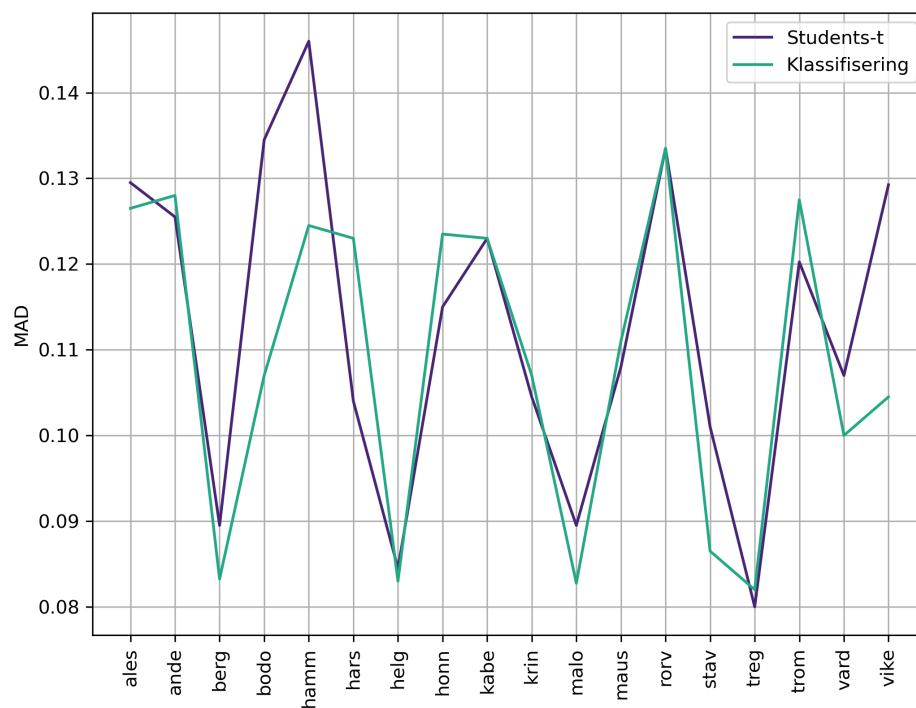
Figur 4.12: Differanse for hele tidsperioden



Figur 4.13: Standardavvik for hele tidsperioden



Figur 4.14: Spearman korrelasjon for hele tidsperioden



Figur 4.15: MAD per vannstandsmåler for hele tidsperioden

Resultatene fra valideringsområde rundt Bergen skiller seg ut i figur 4.12 og er stasjonen hvor maskinlæringsmetoden presterer dårligst i forhold til fasit. Multippel-t metoden har også utfordringer i områdene rundt Bergen. Det kan forklares av valideringsområdet i figur 3.7 hvor store deler av altimetriobservasjonene kommer fra åpen sjø. Om havet har en gradient i området kan det være årsaken. Om valideringsområdet ikke hadde strekt seg like langt ut på havet ville mest sannsynlig også Bergen kommet under minstekravet på 1000 observasjoner og 20 pass.

For å se på forskjellen mellom de to metodene for hele kysten blir resultater for de resterende valideringsområdene midlet og satt opp mot hverandre.

Tabell 4.2: Kystsnitt og prosentvis endring.

	Students t	Klassifisering	Prosentvis endring
Differanse(mm)	21.6	9.3	-57.1
Standardavvik(mm)	111.8	103.0	-7.9
Spearman korrelasjon	0.8724	0.8786	0.7
MAD	0.1125	0.1086	-3.4
Observasjoner inkludert	6253	4809	-23.1
Pass inkludert	93	83	-10.9

Vi ser fra tabell 4.2 at pulser fra klasse 9 reduserer differansen betydelig. Øvrige størrelser ser bare en marginal forbedring. For kystsnittet blir færre pass og observasjoner inkludert i beregningen av dynamisk havtopografi. I snitt blir 4809 observasjoner og 83 pass som inneholder observasjoner, inkludert for hver vannstandsmåler.

For å sammenlikne med SAR-altimetriobservasjoner fra CryoSat-2 i Idžanović et al. [2018] må man først midle resultatene fra indre og ytre kystzone, slik at man får et kystsnitt. Det midlede kystsnittet for standardavviket blir 14.5 cm. Sammenliknet med resultater i tabell 4.2 vil begge metodene egne bedre for kystaltimetri enn CryoSat-2 SAR-observasjoner. Det tyder på at det er PEACHI-datasettet som fører til forbedringen.

Om man sammenlikner med resultatene fra Breili [2021] må man midle indre og ytre kystzone på samme måte. Da blir standardavviket for kystsnittet 8.65 cm. Det indikerer at SAR-observasjoner fra Sentinel-3 egner seg bedre til kystaltimetri enn Sentinel-3 PEACHI data. Det stemmer med forventningene med tanke på fotavtrykkets størrelse for datasettene.

5 Diskusjon

En svært tidkrevende og utfordrende del av oppgaven har vært den manuelle klassifiseringen. Det innebærer også definering av klassene, som tidlig i prosessen ble låst på grunn av tidsbruk. Det ble foretatt en revisjon etter man ble kjent med datasettet. Da fikk sjø-pulsformene (7 og 9) fikk en ekstra klasse (8) og peak-klassene (2 og 3) fikk en ekstra klasse (4). Det medførte at den tidkrevende klassifiseringen måtte starte på nytt. For mengden treningsdata tilgjengelig vil en og annen feilklassifisert puls ikke utgjøre mye i maskinlæringsmodellen. Problemet man møter i denne sammenhengen er at grensene mellom klassene ikke er klare nok. Opprinnelig var det to personer som klassifiserte pulser fra ulike sykluser for å få mest mulig treningsdata. Når data fra begge personene ble kombinert presterte maskinlæringsmodellen betydelig dårligere, derfor ble bare data fra den ene personen benyttet i oppgaven. Det betyr ikke at det ene datasettet er bedre enn det andre, men at kombinasjonen er ugunstig for maskinlæringsmodellene i oppgaven. Selv om eksempelpulsene i figur 3.3 og beskrivelsene i tabell 3.3 definerer klassene, er det rom for tolkning. På bakgrunn av dette ble ikke-styrt klassifisering foreslått som en metode for å unngå manuell klassifisering.

I figur 3.6 ser man at starten på pulsene i klasse 7, 8 og 9 fordeler seg ut over utvalgspunktene. For klasse 3 ser man også pulser som befinner seg på forskjellige steder i forhold til utvalgspunktene. Årsaken til forskyvningen er usikker. En slik gruppering innad i en klasse kan forvirre maskinlæringsmodellen, på samme måte clusteren så ut til å bli forvirret av det (figur 4.9), og føre til feilpredikering. Om man hadde funnet en måte å samle starten på alle pulsene ville trolig resultater for styrt og ikke-styrt klassifisering blitt forbedret. Da kunne maskinlæringsmodellene kun fokusert på formen til returpulsen, istedenfor formen og plassering til returpulsen i utvalgspunktene.

Maskinlæringsmodellene i oppgaven er ikke fininnstilt og er forholdsvis enkle modeller. Ut fra forvirringsmatrisen til *CNN*, som ble brukt til å predikere returpulser i valideringsområdene må vi anta at bare 83% av pulser som var ren sjø havnet i klasse 9. I tillegg må man inkludere eventuelle feil som ble begått i den manuelle klassifiseringen. Med modeller som er fininnstilt kan man forvente bedre prestasjon, og det er stor sannsynlighet for at det ville forbedret resultatene fra dynamisk havtopografi. En alternativ måte å klassifisere returpulsene på er å bruke en RNN-modell. Da kunne man utnyttet informasjonen som ligger i etterfølgende observasjonspunkter i et pass. På grunn av det tilfeldige utvalget som blir gjort i den manuelle klassifiseringen lot det seg ikke gjøre.

At bare klasse 9 ga sammenliknbare resultater med multippel-t test, bekrefter at det er nødvendig med minst tre sjøklasser (7, 8, 9). Ellers ville forventet sammenliknbare resultater fra klasse 9 alene og klasse 7, 8 og 9 i ulike kombinasjoner. Internt i klasse 7 ser det ut som at det er et skille, se figur 4.6. 50 % av ekte klasse 7 pulser blir predikert som klasse 9 av *CNN*-modellen. Det er usikkert om dette har hatt positiv eller negativ effekt på resultatene, men returpulsene predikert som klasse 7 gir dårligere resultater.

Den ujevne klassefordelingen i datasettet (se figur 3.5) har gitt utfordringer i treningen av maskinlæringsmodellene, og ført til at man må ta ekstra hensyn i valideringen av maskinlæringsmodellene. Klassen med færrest returpulser er klasse 3, med ca. 100 returpulser. Om man skulle balansert klassefordelingen ville svært mye data blitt kastet bort, og datasettet ville bestått av kun 900 returpulser. Det ble forsøkt å trene en modell på de 900 returpulsene, men resultatene ble betydelig dårligere, og derfor har den skjeve klassefordelingen blitt værende.

All S3-PEACHI data som er tilgjengelig har blitt brukt i oppgaven. Om flere observasjoner skulle

blitt inkludert i beregningen av dynamisk havtopografi måtte man funnet en måte å bruke avstandsmålinger fra de andre returpulsklassene. I figur B.2 og B.3 fra vedlegg B, ser man predikerte pulser fra valideringsområder i Oslofjorden. I disse områdene har man mange returpulser predikert til klasse 2, 4, og 8. Om en retracker hadde blitt inkludert i oppgaven er det mulig at man kunne fått riktig avstandsmåling fra disse også. Det mest optimale ville vært å utvikle en retracker for hver klasse. Fra resultatene i denne oppgaven er det tydelig at retrackeren brukt for dette datasettet kun gir riktig avstand på klasse 9 returpulser. I følge Moreau et al. [2021] kan samme retracker benyttes for både S3-SAR og S3-LR-RMC. Man kan derfor anta at maskinæringsmodellen, trent på LR-RMC data, kan predikere SAR-data. SAR-data forventes mindre antall landkontaminerte returpulser i kystsonen forventer man mindre landkontaminerte returpulser på grunn av at SAR fotavtrykket er mindre, og vil derfor egne seg bedre til kystaltimetri. Man må være forberedt på at man kan møte de samme utfordringene i forhold til feil avstandsmåling fra pulser som fraviker åpen-sjø formen og ulikt startpunkt for returpulsen.

Testområdene (se figur 3.2) som ble benyttet i oppgaven, har på samme måte som de definerte returpulsklassene, blitt låst tidlig. De ble bestemt før valideringsområdene (se figur 3.7) var definert, og man finner derfor tilfeller hvor valideringsområder overlapper med testområder. Dette har ført til at observasjoner som skal predikeres har vært del av treningsdatasettet til modellen. Når det skjer, kan resultatene fra predikeringen bli kunstig gode. I dette tilfellet gjelder det kun for observasjoner i syklus 20-24, og man kan anta at det har påvirket resultatene i liten grad. Man kunne til fordel ekskludert de aktuelle syklusene fra predikeringen. Ideelt sett skulle testområdene vært definert i områder uten vannstandsmålere slik at valideringsområdene ikke overlappet. Det kunne også ha vært flere og mindre testområder, som kunne balansert klassefordelingen, se figur 3.5. Den skjeve klassefordelingen kan delvis være forårsaket av ulik størrelse på testområdene, samt testområdets utstrekning mot åpent hav. Sognefjorden er et testområde med mye åpent hav hvor man forventer mange klasse 9 pulser.

En annen måte å balansere klassefordelingen på ville vært å definere et eget valideringsområde på åpent hav, eller noe liknende som I Breili [2021] blir interesseområdene definert fra innerst i Sognefjorden og ut til åpent hav. Om noe liknende hadde vært gjort her kunne man undersøkt hvor nærme kysten S3-PEACHI data kan brukes, og hvordan datasettet gjør det regionalt. Et alternativ er å ha en fordeling av indre (innaskjærs) og ytre (utaskjærs) kystsoner. Da kunne man undersøkt hvor mye havet i ytre kystsoner skiller fra havet i indre, og samtidig testet datasettet i begge områdene. Valideringsområdet i Bergen demonstrerer dette til en viss grad, hvor store deler av området dekker hav som er utaskjærs. Resultatene fra Bergen peker mot at det ikke er gunstig å bruke observasjoner fra ytre kystsoner, men de kan være svartet av dårlige observasjoner i indre kystsoner.

Altimetridatasettet som har blitt brukt i oppgaven har hatt mye større fotavtrykk enn det som var forventet. Forventningen var at det skulle ha like stort fotavtrykk som offisielle S3-SAR observasjoner. LR-RMC prosesseringsmetoden resulterer i et fotavtrykk som kan sammenliknes med fotavtrykket til tradisjonell altimetri (ca.10 km i diameter). Årsaken til at datasettet ikke ble byttet ut er at man får muligheten til å undersøke ytelsen til Sentinel-3 PEACHI-data langs Norskekysten og sammenlikne med tidligere arbeid. Sammenlikningen vil ikke bli helt optimal på grunn av ulike valideringsområder i forhold til vannstandsmålerne.

6 Konklusjoner og anbefalinger

I denne oppgaven skulle potensialet til Sentinel-3 PEACHI-data langs Norskekysten utforskes. I tillegg skulle maskinlæring bli brukt til klassifiseringen av altimetriobservasjoner basert på returpulsens form. Samtidig gjennomføres et grovfeilsøk med multippel-t test på altimetriobservasjonene. De to metodene for filtrering av altimetriobservasjoner brukes parallelt til å beregne dynamisk havtopografi. Metodene sammenliknes med fasit, som består av dynamisk havtopografi beregnet fra vannstandsmåler-observasjoner. Ved å beregne et kystsnitt for hver metode vil man ut fra dette kunne avgjøre hvilken metode som egner seg for filtrering av altimetriobservasjoner langs Norskekysten.

Fire ulike maskinlæringsmodeller ble utprøvd for å finne den beste til å filtrere altimetriobservasjoner. *K-Means* var den eneste ikke-styrte klassifiseringsmetoden som ble testet. Visualisering av prinsippal komponentene i datasettet tyder på at det ikke finnes tydelige grupperinger i datasettet. Den fant ikke grupperinger i datasettet som hadde de egenskapene man ønsker i returpulsens form (se figur 4.9). Derfor ble ikke resultatet av *K-Means* benyttet for å beregne dynamisk havtopografi.

Maskinlæringsmodellene *Dense*, *CNN HSI* og *CNN* predikerte pulser med henholdsvis 66.3%, 63.4% og 68.2% klassifiseringsnøyaktighet. *CNN* ble på bakgrunn av forvirringsmatrisen valgt til å predikere altimetriobservasjoner i valideringsområdene. Det var kun returpulser i klasse 9 som klarte å beregne dynamisk havtopografi bedre enn multippel-t test. Andre klassekombinasjoner gjorde det betydelig dårligere enn multippel-t test. Valideringsområdene Oslo, Oscarsborg, Narvik, Trondheim og Heimsjø ble fjernet fra beregningene av kystsnitt fordi det var svært få observasjoner og pass som inneholdt observasjoner av klasse 9.

Filtrering ved bruk av Student's multippel-t test gir differanse, standardavvik, MAD og Spearman-korrelasjon på henholdsvis 21.6 mm, 111.8 mm og 0.8724. Klassifisering basert på returpulsens form gir differanse, standardavvik, MAD og Spearman-korrelasjon på henholdsvis 9.3 mm, 103.0 mm og 0.8786.

Kystsnittene presentert ovenfor indikerer at det er fordelene med å bruke en maskinlæringsmodell istedenfor multippel-t test for å avgjøre om altimetriobservasjoner kan brukes til beregning av dynamisk havtopografi. For kystmiddelet var den prosentvise endringen -57.1%, -7.9%, 0.7% og -3.4%, for henholdsvis differanse, standardavvik, Spearman-korrelasjon og MAD.

Resultatene peker mot at filtrering med maskinlæring presterer bedre enn multippel-t test for hele kysten. Når man sammenlikner med tidligere arbeid hvor man har benyttet Sentinel-3 SAR-observasjoner, peker resultatene på at SAR-observasjoner gir minst standardavvik, og egner seg bedre til kystaltimetri enn PEACHI-observasjoner. Derfor bør man bruke Sentinel-3 SAR-observasjoner i videre studier langs Norskekysten. Sentinel-3 SAR-datasettet som er tilgjengelig går over en lengre periode, og man kan også inkludere Sentinel-3B i analysene. Kombineringen mellom SAR-observasjoner og filtrering med maskinlæringsmetoder kan forbedre resultatene ytterligere.

Når det gjelder maskinlærings-metoden for filtrering av altimetriobservasjoner er det stort forbedringspotensial. Bare ved å fininnstille maskinlæringsmodellene kunne flere returpulser blitt klassifisert riktig. Metoden kan også dra nytte av å inkludere en retracker som tilpasses hver klasse, slik at de også kan brukes til å beregne dynamisk havtopografi. I tillegg kunne en annen type maskinlæringsmetode vært testet, for eksempel RNN, slik at informasjon mellom etterfølgende observasjoner kunne blitt utnyttet.

Testområder bør defineres slik at de ikke overlapper med valideringsområdene. De bør også være i samme størrelsesorden som valideringsområdene, samt spres mer ut langs Norskekysten. Da er det mulig at skjeve klassefordelingen balanseres noe.

Bibliografi

- Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.
- J Ågren, G Strykowski, M Bilker-Koivula, O Omang, S Mårdla, T Oja, I Liepinš, E Paršeliūnas, R Forsberg, J Kaminskis, et al. On the development of the new nordic gravimetric geoid model nkg2015. In IAG Symposium on Gravity, Geoid and Height Systems, Thessaloniki, Greece, pages 19–23, 2016.
- O. B. Andersen and R. Scharroo. Range and Geophysical Corrections in Coastal Regions: And Implications for Mean Sea Surface Determination. In S. Vignudelli, A. G. Kostianoy, P. Cipollini, and J. Benveniste, editors, Coastal Altimetry, pages 103–146. Springer, 2011. doi:10.1007/978-3-642-12796-0_5.
- AVISO. Aviso cnes download platform. <https://www.aviso.altimetry.fr/en/data/products/sea-surface-height-products/global/experimental-sentinel-3-products-peachi.html>, 2021. Accessed: 2021-01-01.
- Eva Boergens, Karina Nielsen, Ole Baltazar Andersen, Denise Dettmering, and Florian Seitz. River levels derived with CryoSat-2 SAR data classification—A case study in the Mekong River Basin. Remote Sensing, 9(12):1238, 2017.
- Kristian Breili. Dynamic ocean topography from Sentinel-3 and tide gauges along the Norwegian coast. Kart og Plan, 2021.
- Kristian Breili, Matthew JR Simpson, and Jan Even Øie Nilsen. Observed Sea-Level Changes along the Norwegian Coast. Journal of Marine Science and Engineering, 5(3):29, 2017.
- Francois Chollet et al. Deep learning with Python, volume 361. Manning New York, 2018.
- Denise Dettmering, Alan Wynne, Felix L Müller, Marcello Passaro, and Florian Seitz. Lead detection in polar oceans—A comparison of different classification methods for Cryosat-2 SAR data. Remote Sensing, 10(8):1190, 2018.
- ESA. Copernicus programme description. <https://www.copernicus.eu/en>, 2021a. Accessed: 2021-02-19.
- ESA. Sentinel sentinel-3 mission description. <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3>, 2021b. Accessed: 2021-02-19.
- Marie-Laure Frery, Mathilde Siméon, Christophe Goldstein, Pierre Féménias, Franck Borde, Alexandre Houpert, and Ana Olea Garcia. Sentinel-3 microwave radiometers: Instrument description, calibration and geophysical products performances. Remote Sensing, 12(16):2590, 2020.
- P Gou, E Santos-Garces, Martin Høy, JP Wold, KH Liland, and E Fulladosa. Feasibility of NIR interactance hyperspectral imaging for on-line measurement of crude composition in vacuum packed dry-cured ham slices. Meat science, 95(2):250–255, 2013.
- Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. Journal of Sensors, 2015, 2015.

-
- Martina Idžanović, Vegard Ophaug, and Ole Baltazar Andersen. The coastal mean dynamic topography in Norway observed by CryoSat-2 and GOCE. Geophysical Research Letters, 44(11): 5609–5617, 2017.
- Martina Idžanović, Vegard Ophaug, and Ole Baltazar Andersen. Coastal sea level from CryoSat-2 SARIn altimetry in Norway. Advances in Space Research, 62(6):1344–1357, 2018.
- Kartverket. Permanente vannstandsmålere. <https://www.kartverket.no/til-sjos/se-havniva/lar-om-tidevann-og-vannstand/permanente-vannstandsmalere>, 2020. Accessed: 2021-05-23.
- Kartverket. Norge i bilder. <https://www.norgebilder.no>, 2021. Accessed: 2021-04-15.
- Diederik P Kingma and Jimmy Ba. SAdam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- Dennis Frederic Mattes. Analysis of waveforms in the satellite altimetry by using neural networks. Master’s thesis, University of Stuttgart, 2019.
- T Moreau, E Cadier, F Boy, J Aublanc, P Rieu, M Raynal, S Labroue, P Thibaut, G Dibarboue, N Picot, et al. High-performance altimeter Doppler processing for measuring sea level height under varying sea state conditions. Advances in Space Research, 67(6):1870–1886, 2021.
- Felix L Mueller, Denise Dettmering, Wolfgang Bosch, and Florian Seitz. Monitoring the Arctic seas: How satellite altimetry can be used to detect open water in sea-ice regions. Remote Sensing, 9(6):551, 2017.
- Vegard Ophaug. Geodetic observations and modeling of gravity field, sea level, and ocean dynamics in the Norwegian coastal zone. PhD thesis, Norwegian University of Life Sciences, Ås, 2017.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.
- L Phalippou and F Demeestere. Optimal re-tracking of SAR altimeter echoes over open ocean: From theory to results for SIRAL2. Ocean Surf. Topogr. Sci. Team Meet, pages 1–18, 2011.
- Sebastian Raschka and Vahid Mirjalili. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd, 2019.
- Revhaug. Grovfeilsøk ved multippel testing med Students t-test og Fisher F-test. Kart og Plan, 67:101–107, 2007.
- Detlef Stammer and Anny Cazenave. Satellite altimetry over oceans and land surfaces. CRC Press, 2017.

A Kode

A.1 *Dense*

```
def build_model():
    model = models.Sequential([
        layers.Dense(64, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(64, activation='relu'),
        layers.Dense(9, activation='softmax')])

    model.compile(optimizer='Adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    return model

model = build_model()

historyD = model.fit(X_norm_train,
                    dummy_y_train,
                    epochs=75,
                    batch_size=32,
                    verbose=0,
                    validation_split=0.2)
```

A.2 HSI-CNN

```
def build_HSI_CNN():

    input_img = Input(shape=(1, X_train.shape[1]))

    c1 = Conv1D(filters = 20,
               kernel_size=k1,
               activation='tanh',
               padding='same')(input_img)

    p1 = MaxPooling1D(pool_size=k2, padding='same')(c1)

    de1 = Dense(n4, activation='tanh')(p1)

    f1 = Flatten()(de1)

    outputs = Dense(9, activation='softmax')(f1)

    model = Model(inputs=[input_img], outputs=[outputs])

    opt = RMSprop(lr=0.001, momentum=0.0, centered=False)

    model.compile(loss='categorical_crossentropy',
                  optimizer='Adam',
                  metrics=['accuracy'])

    return model

model = build_HSI_CNN()

model.summary()

history_HSI = model.fit(re_X_train,
                       dummy_y_train,
                       epochs=150,
                       batch_size=32,
                       verbose=0,
                       validation_split=0.2)
```

A.3 CNN

```
def build_CNN():  
  
    input_img = Input(shape=(1, X_train.shape[1]))  
    c1 = Conv1D(filters=96,  
               kernel_size=8,  
               activation='relu',  
               padding='same',  
               strides=2)(input_img)  
  
    c2 = Conv1DTranspose(filters=96,  
                        kernel_size=8,  
                        activation='relu',  
                        padding='same')(c1)  
  
    c3 = Conv1D(filters=64,  
               kernel_size=16,  
               activation='relu',  
               padding='same')(c2)  
  
    dr1 = Dropout(rate=0.2)(c3)  
  
    c4 = Conv1DTranspose(filters=64,  
                        kernel_size=16,  
                        activation='relu',  
                        padding='same')(dr1)  
  
    p1 = MaxPooling1D(pool_size=4,  
                     padding='same',  
                     strides=2)(c4)  
  
    c5 = Conv1D(filters=32,  
               kernel_size=16,  
               activation='relu',  
               padding='same')(p1)  
  
    f1 = Flatten()(c5)  
  
    outputs = Dense(9, activation='softmax')(f1)  
  
    model = Model(inputs=[input_img],  
                  outputs=[outputs])
```

```
model.compile(loss='categorical_crossentropy',
              optimizer='Adam',
              metrics=['accuracy'])

return model

model = build_CNN()

history_CNN = model.fit(re_X_train,
                       dummy_y_train,
                       epochs=50,
                       batch_size=32,
                       verbose=2,
                       validation_split=0.2)
```

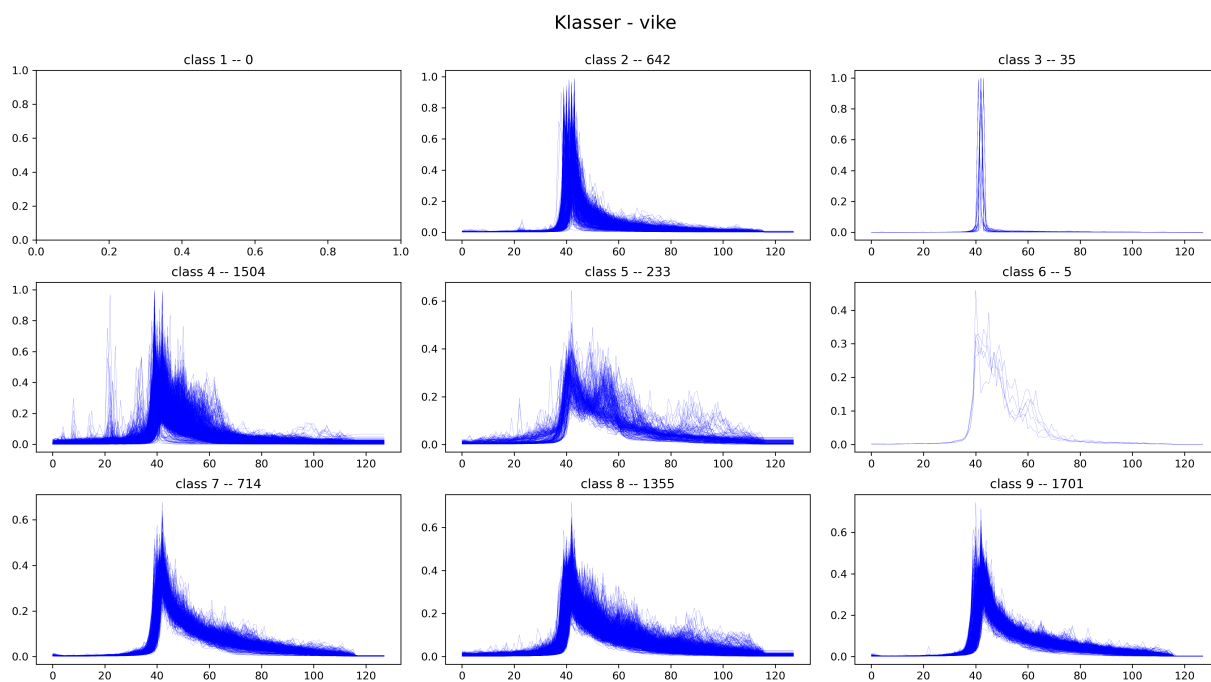
A.4 *K-Means*

```
km = KMeans(n_clusters=9,
            init='random',
            n_init=10,
            max_iter=750,
            tol=1e-06,
            random_state=0)

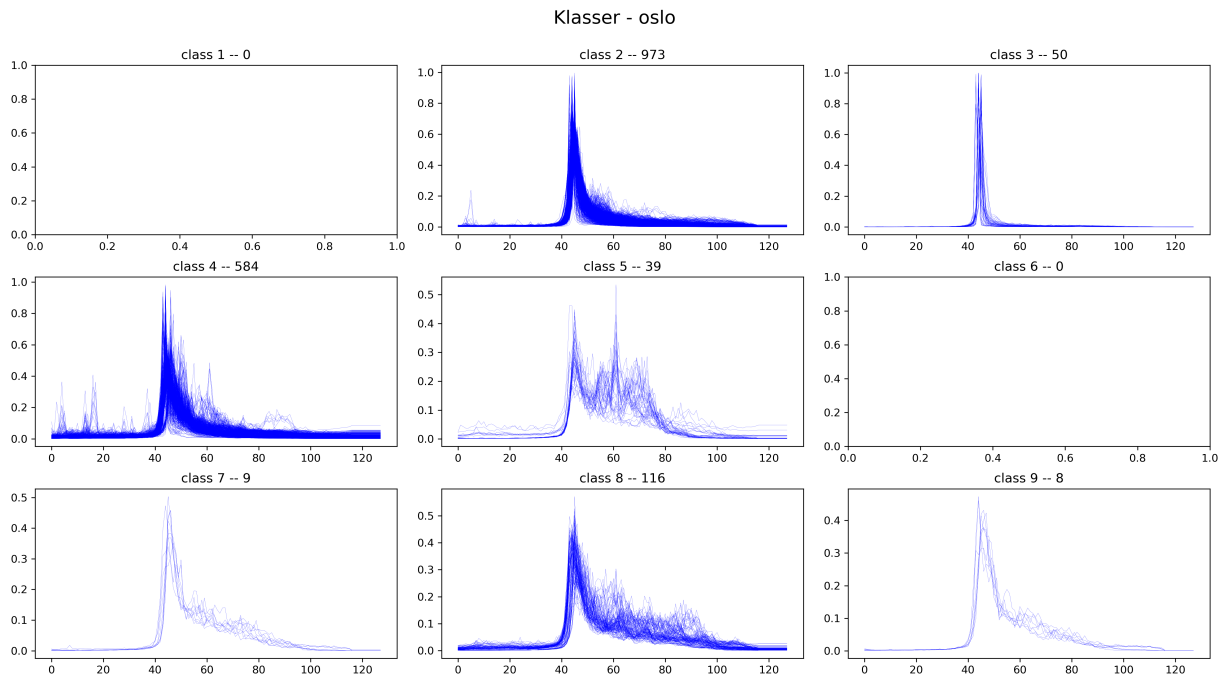
y_km = km.fit_predict(X_pca)
```

B Figurer

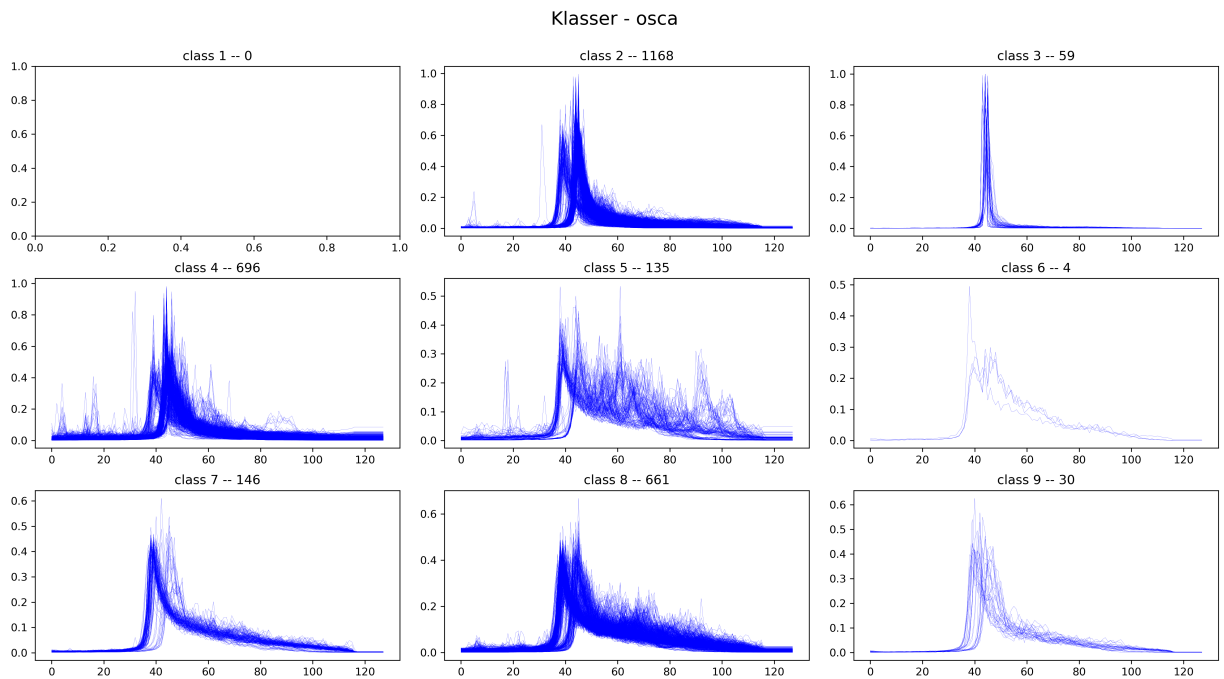
Figurer som viser klassefordeling etter prediksjon for hvert valideringsområde. Det er *CNN*-modellen som har gjort prediksjonene.



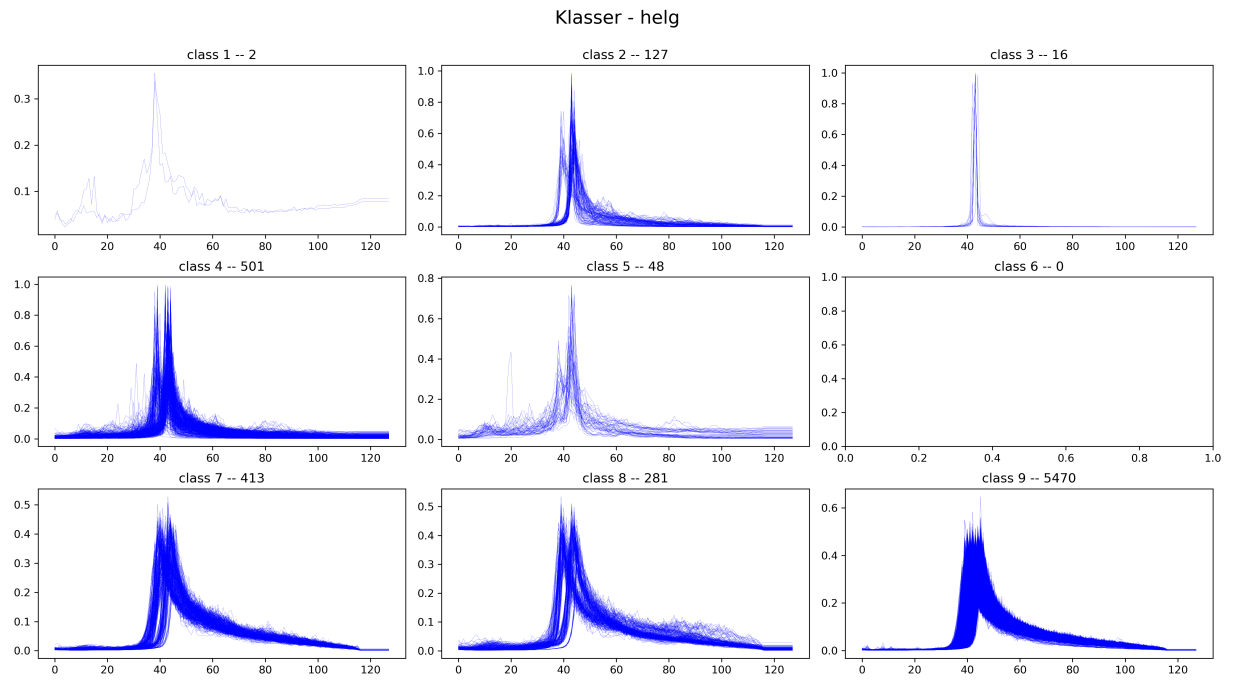
Figur B.1: Predikerte pulser fra valideringsområde rundt Vikar.



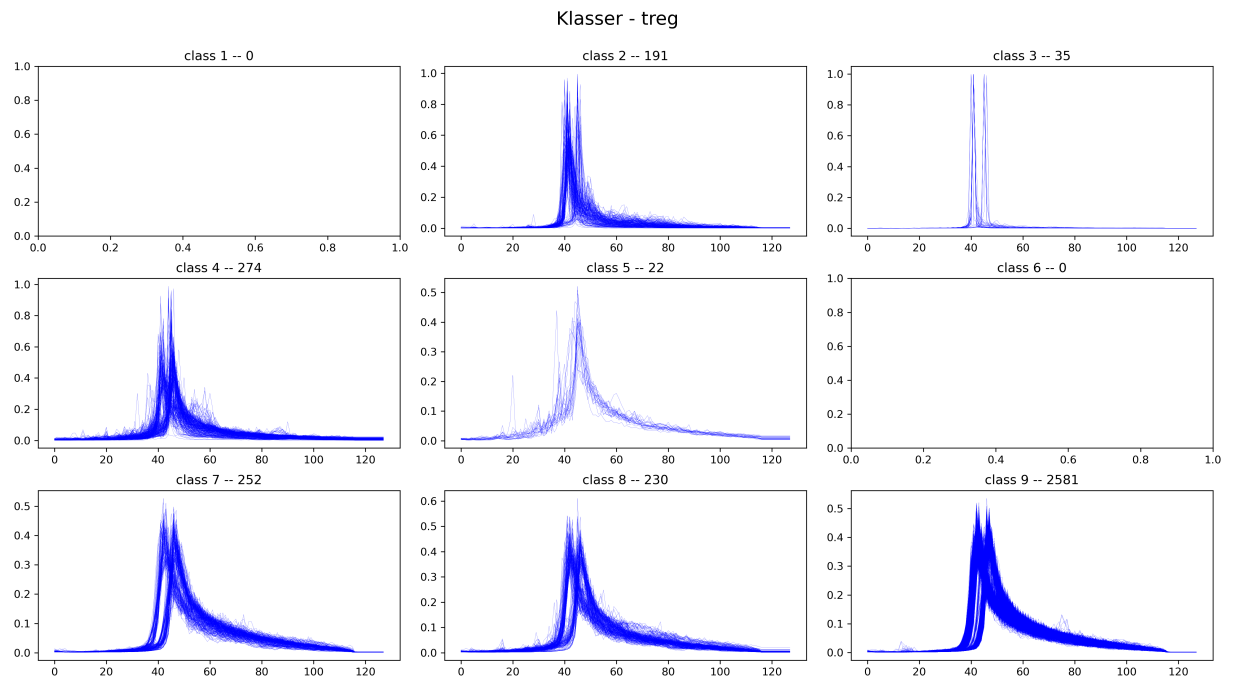
Figur B.2: Predikerte pulser fra valideringsområde rundt Oslo.



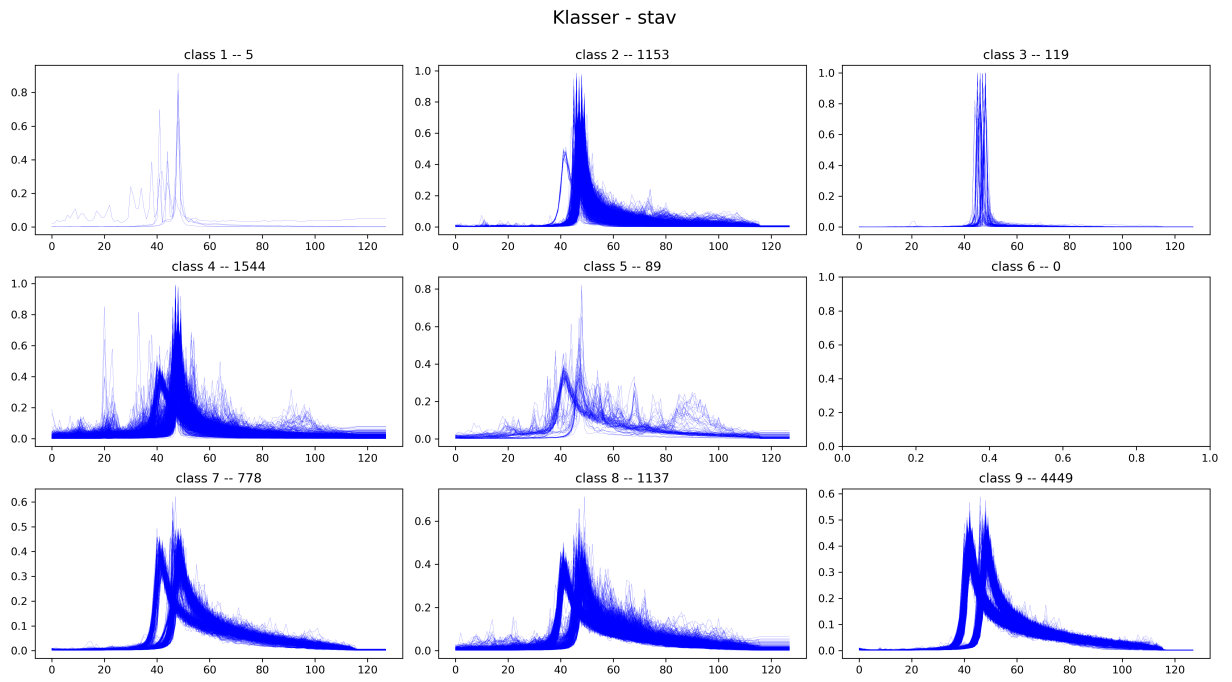
Figur B.3: Predikerte pulser fra valideringsområde rundt Oscarsborg.



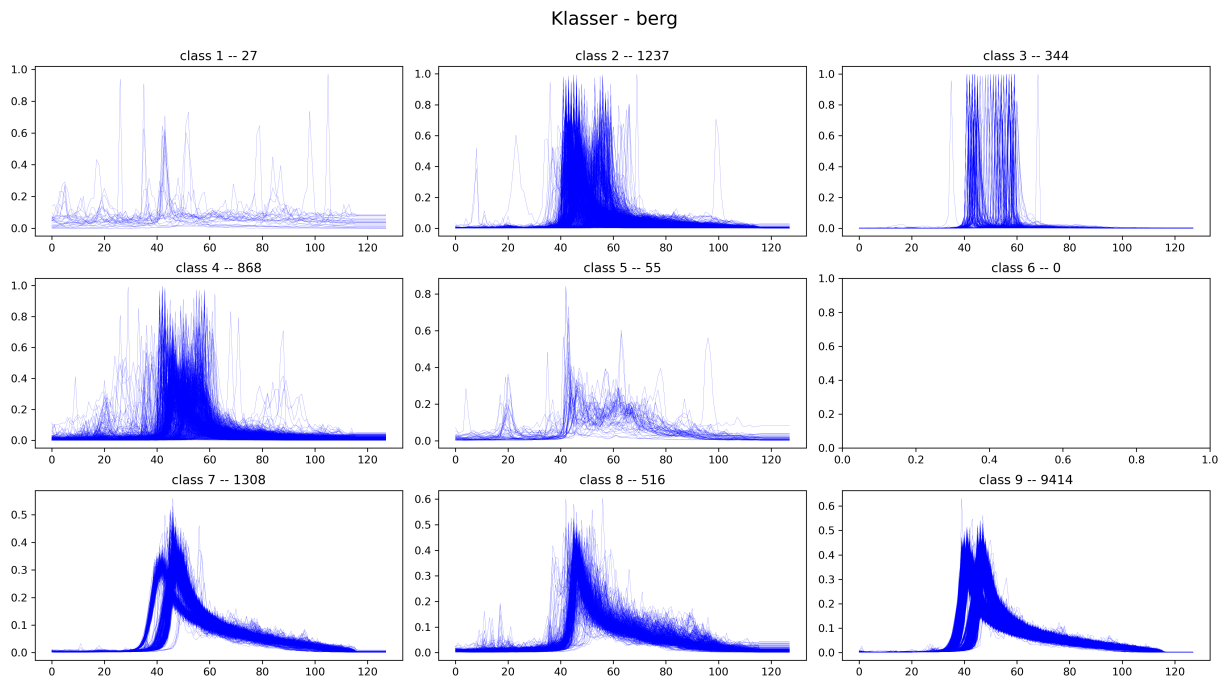
Figur B.4: Predikerte pulser fra valideringsområde rundt Helgeroa.



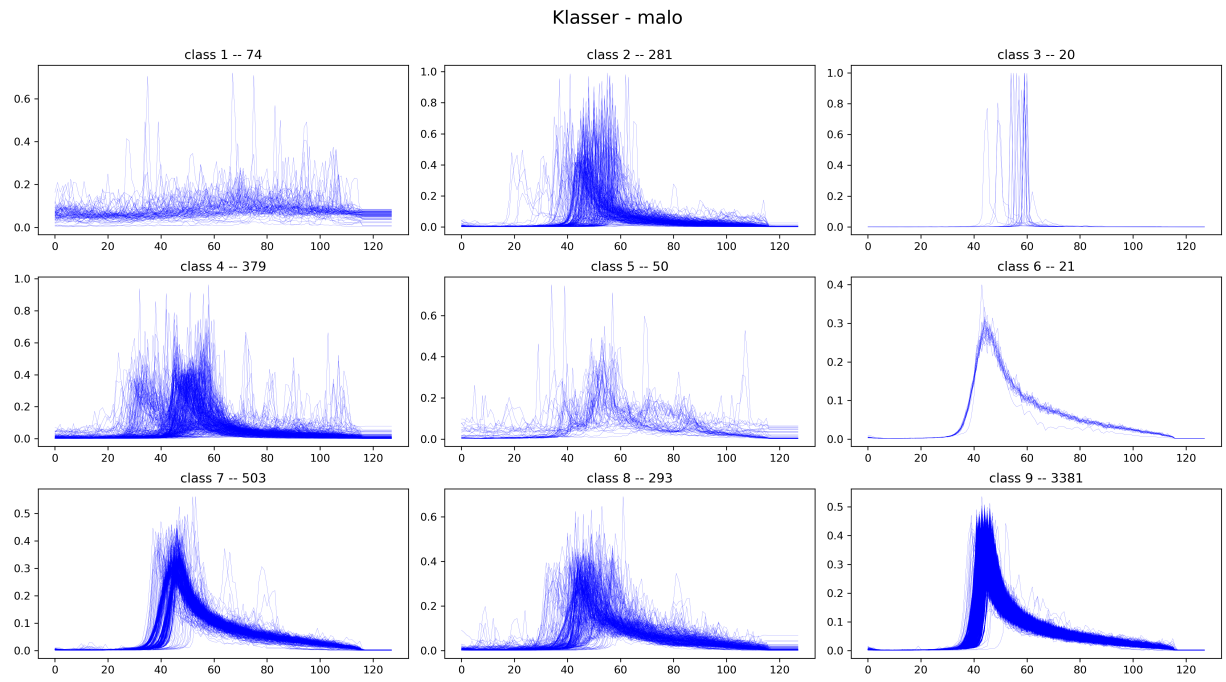
Figur B.5: Predikerte pulser fra valideringsområde rundt Tregde.



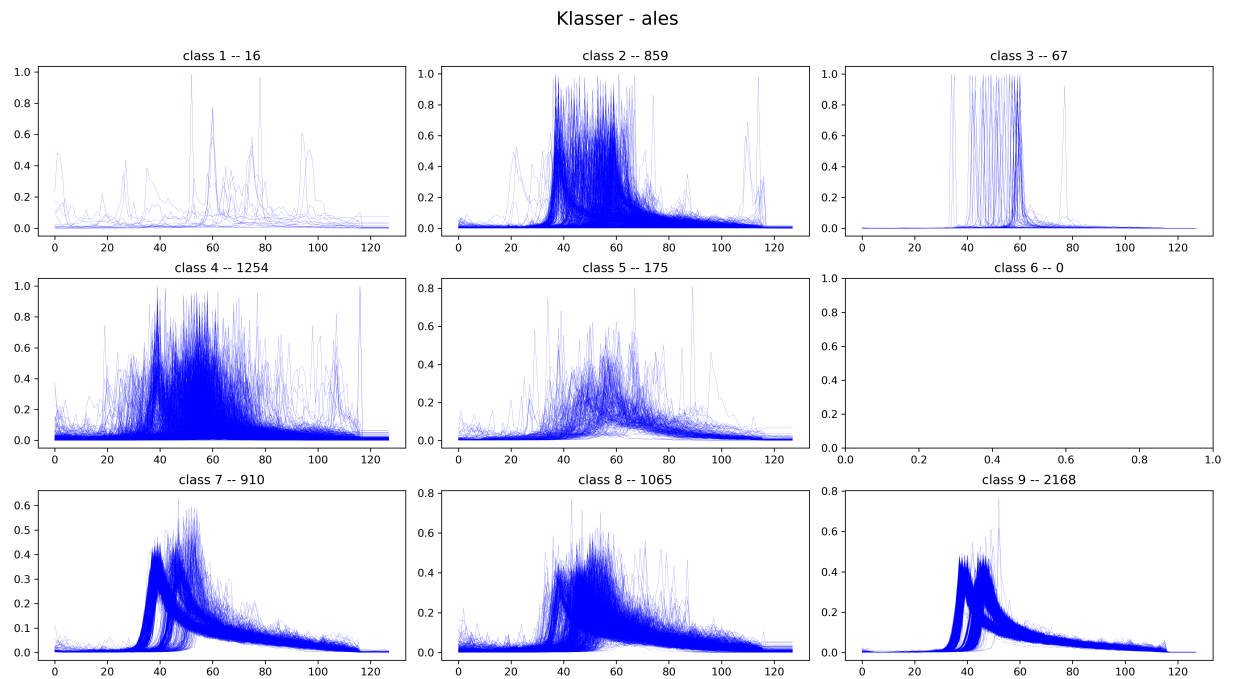
Figur B.6: Predikerte pulser fra valideringsområde rundt Stavanger.



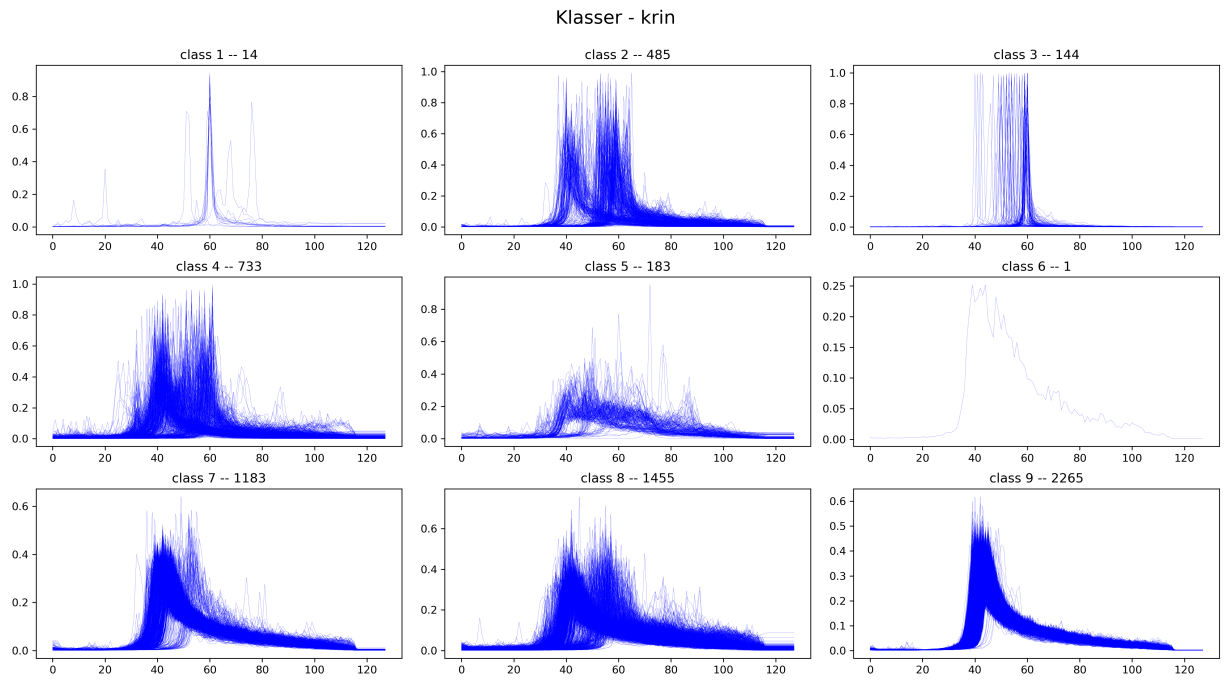
Figur B.7: Predikerte pulser fra valideringsområde rundt Bergen.



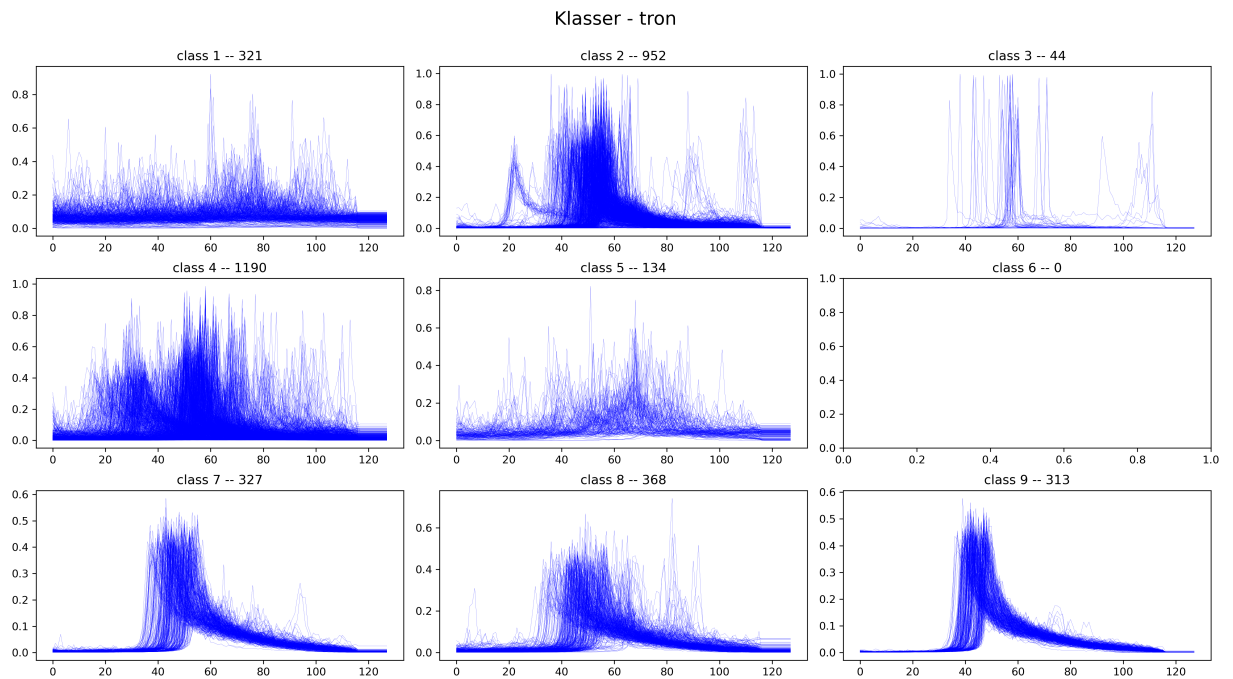
Figur B.8: Predikerte pulser fra valideringsområde rundt Måløy.



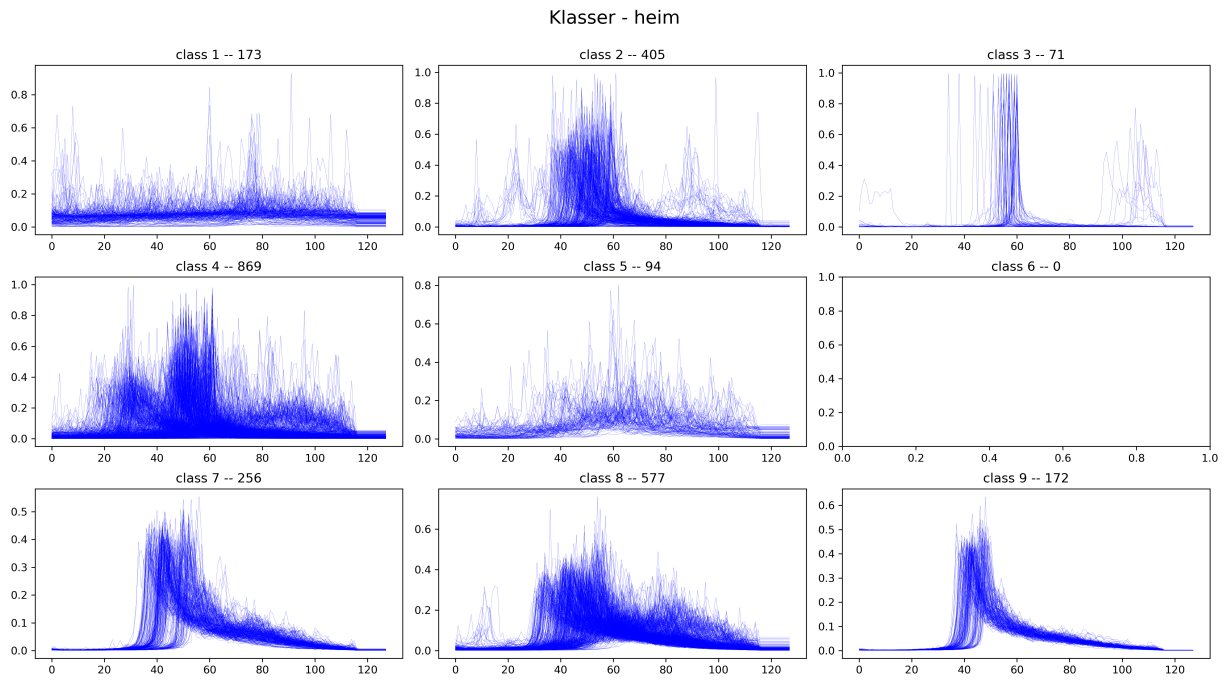
Figur B.9: Predikerte pulser fra valideringsområde rundt Ålesund.



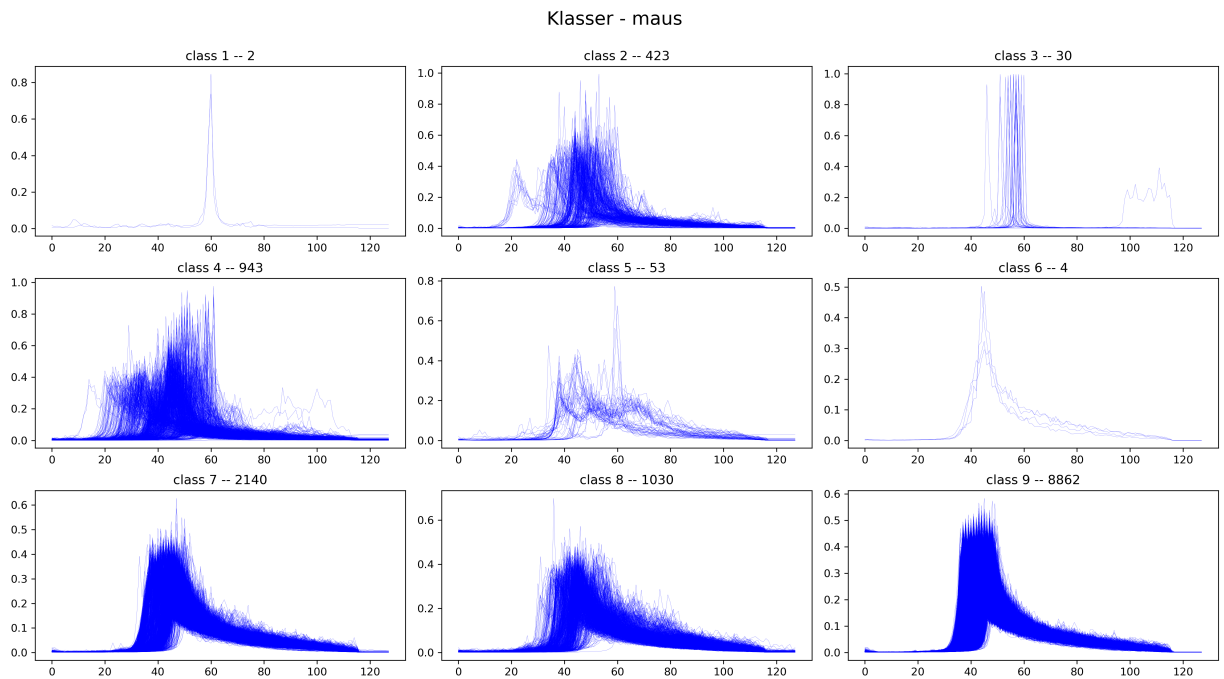
Figur B.10: Predikerte pulser fra valideringsområde rundt Kristiansund.



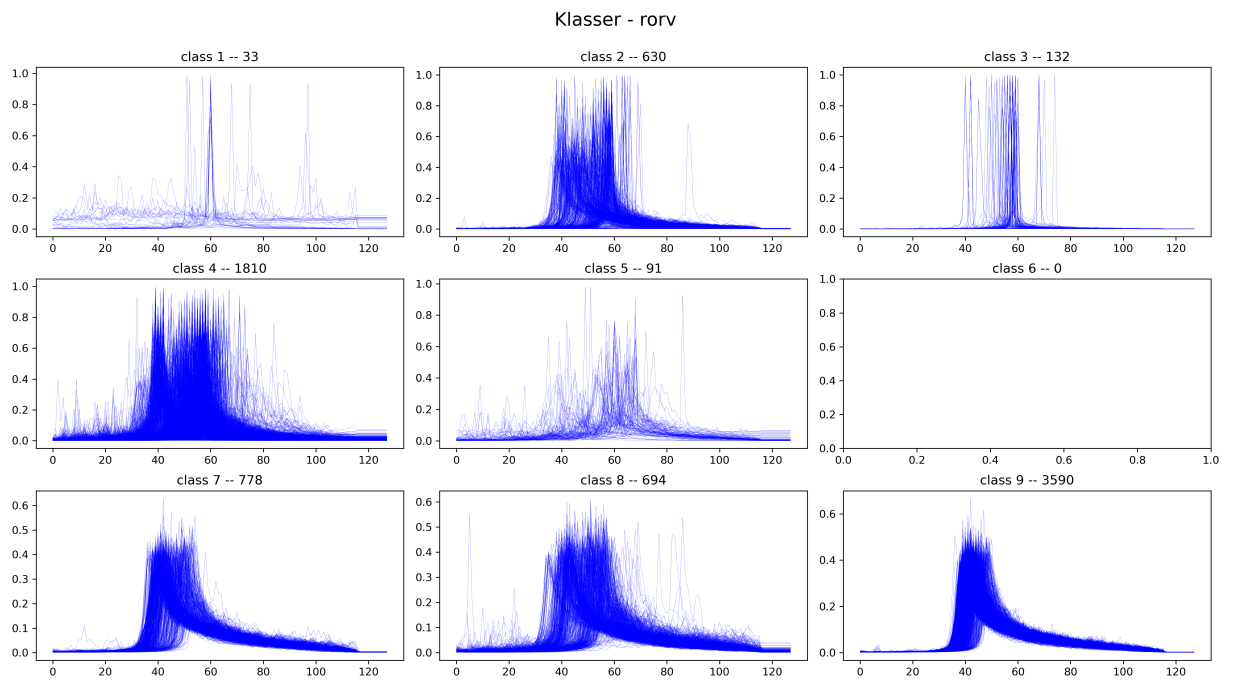
Figur B.11: Predikerte pulser fra valideringsområde rundt Trondheim.



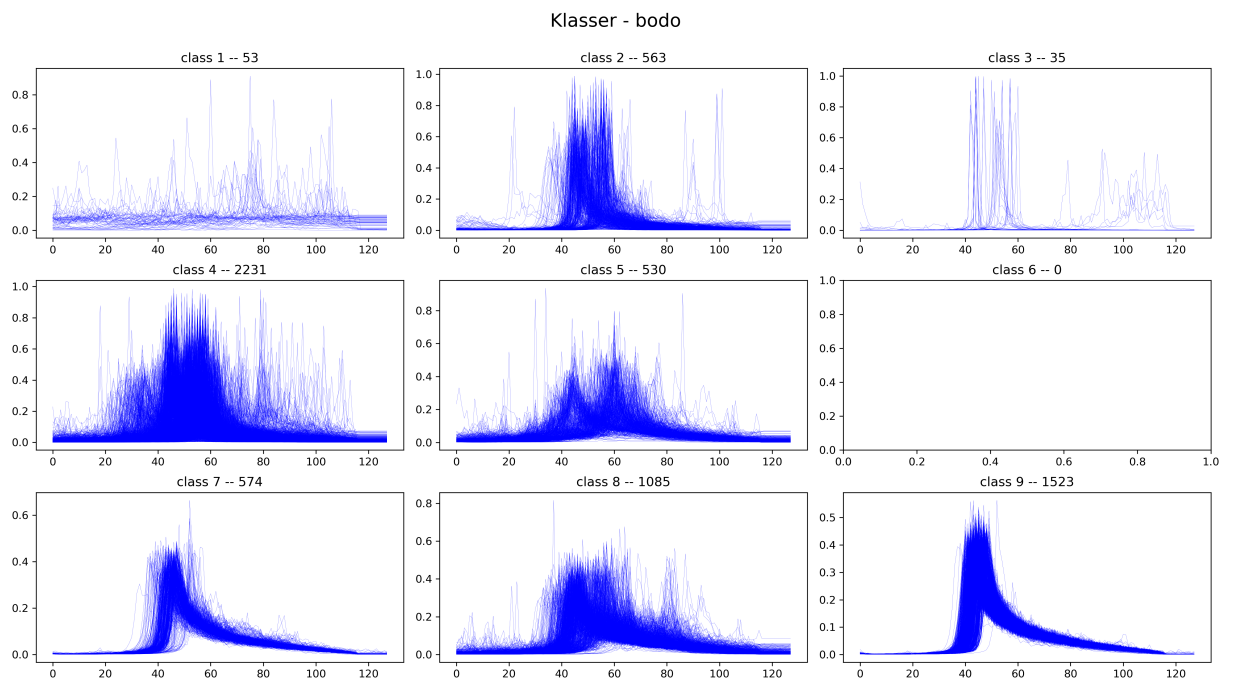
Figur B.12: Predikerte pulser fra valideringsområde rundt Heimsjø.



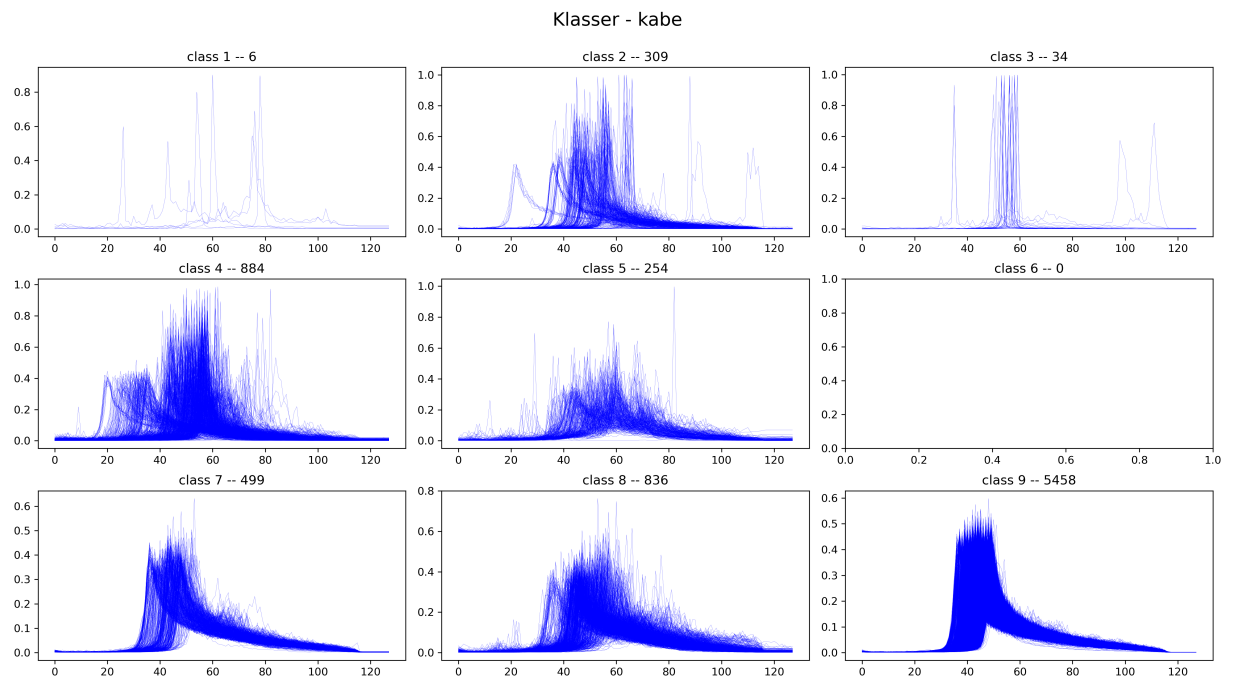
Figur B.13: Predikerte pulser fra valideringsområde rundt Mausund.



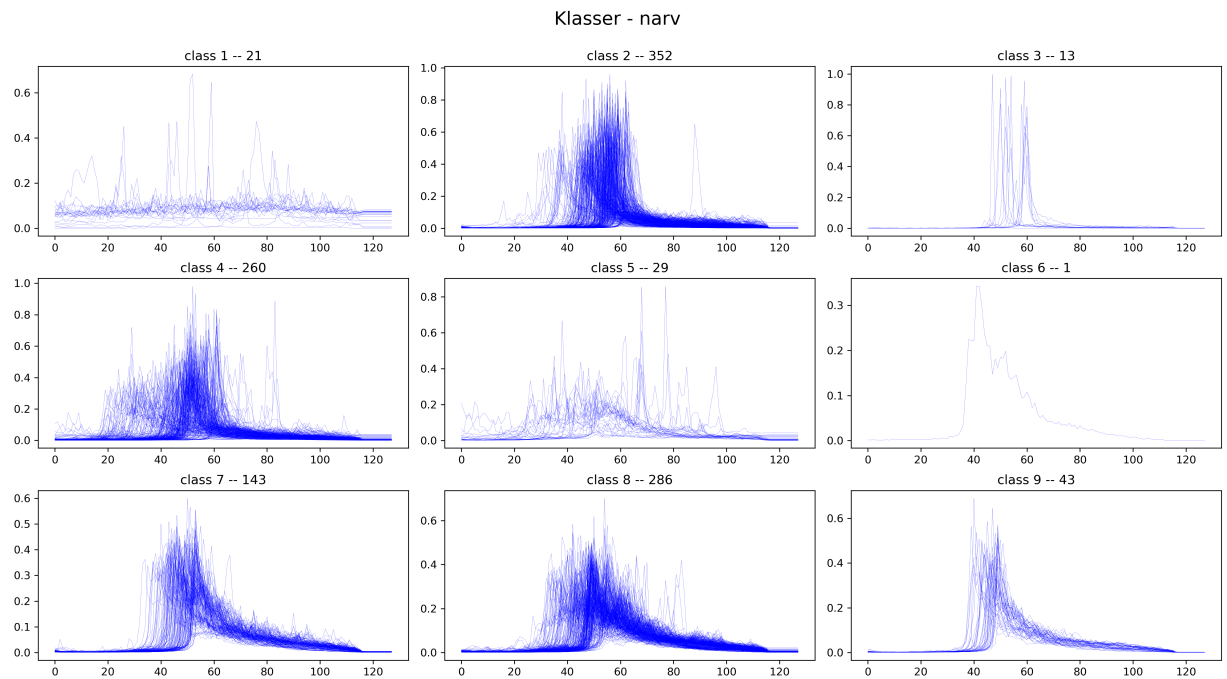
Figur B.14: Predikerte pulser fra valideringsområde rundt Rørvik.



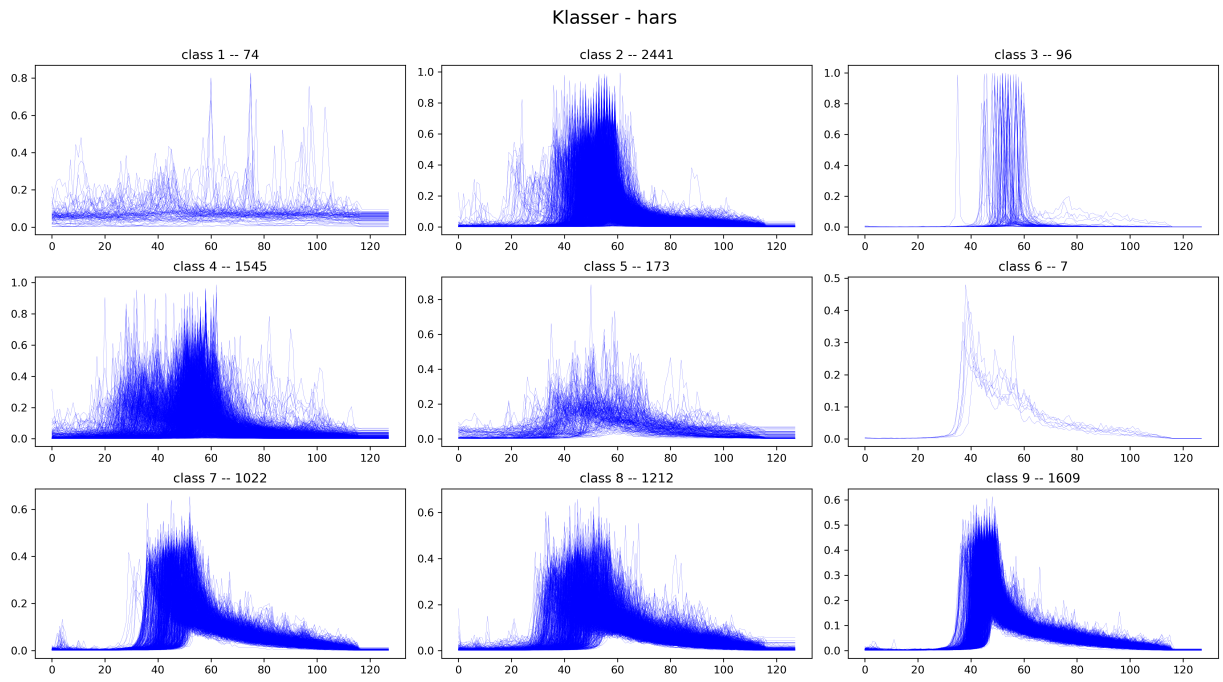
Figur B.15: Predikerte pulser fra valideringsområde rundt Bodø.



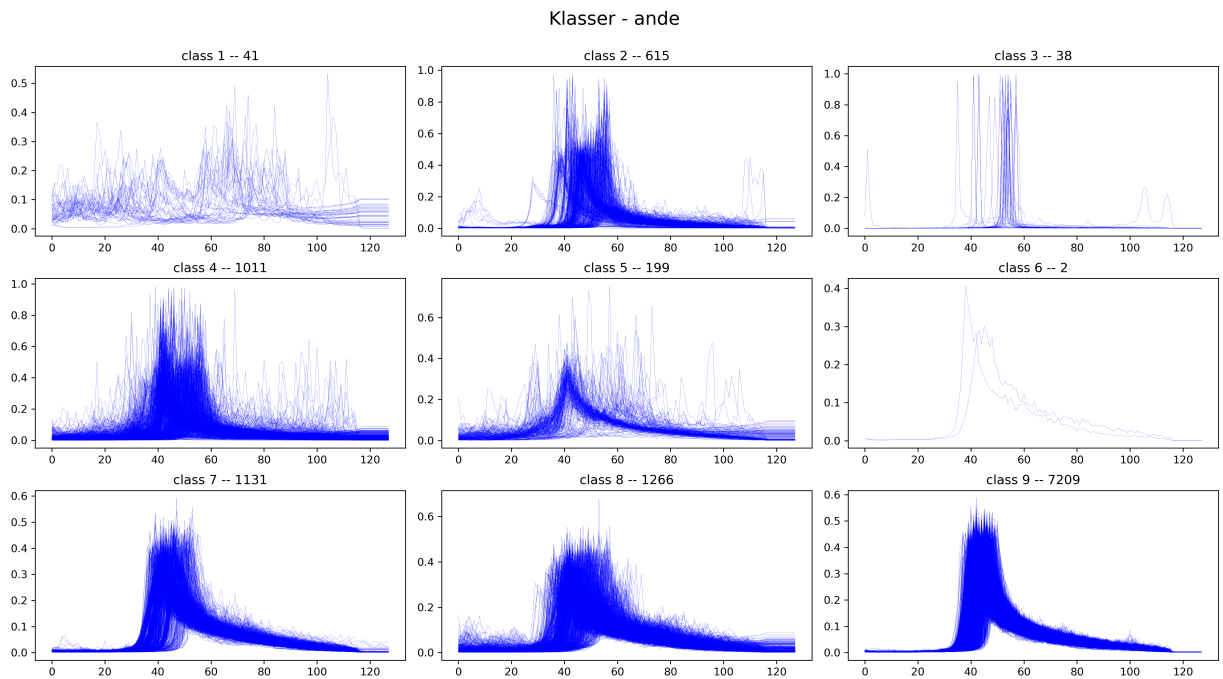
Figur B.16: Predikerte pulser fra valideringsområde rundt Kabelvåg.



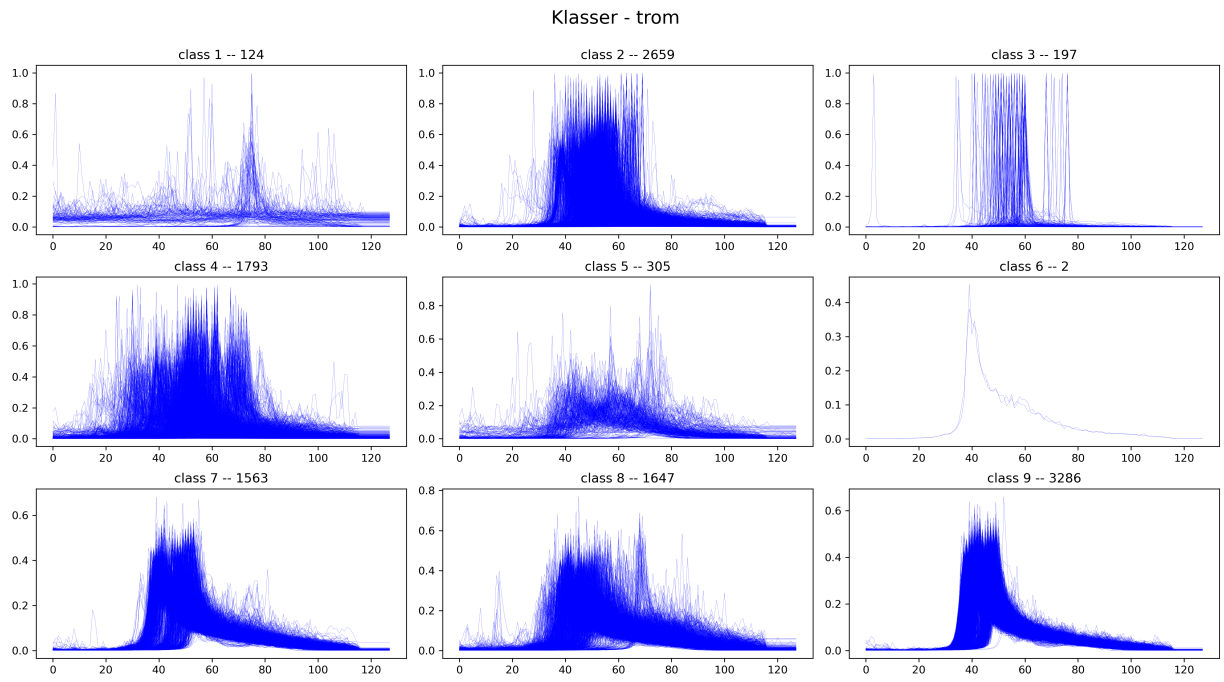
Figur B.17: Predikerte pulser fra valideringsområde rundt Narvik.



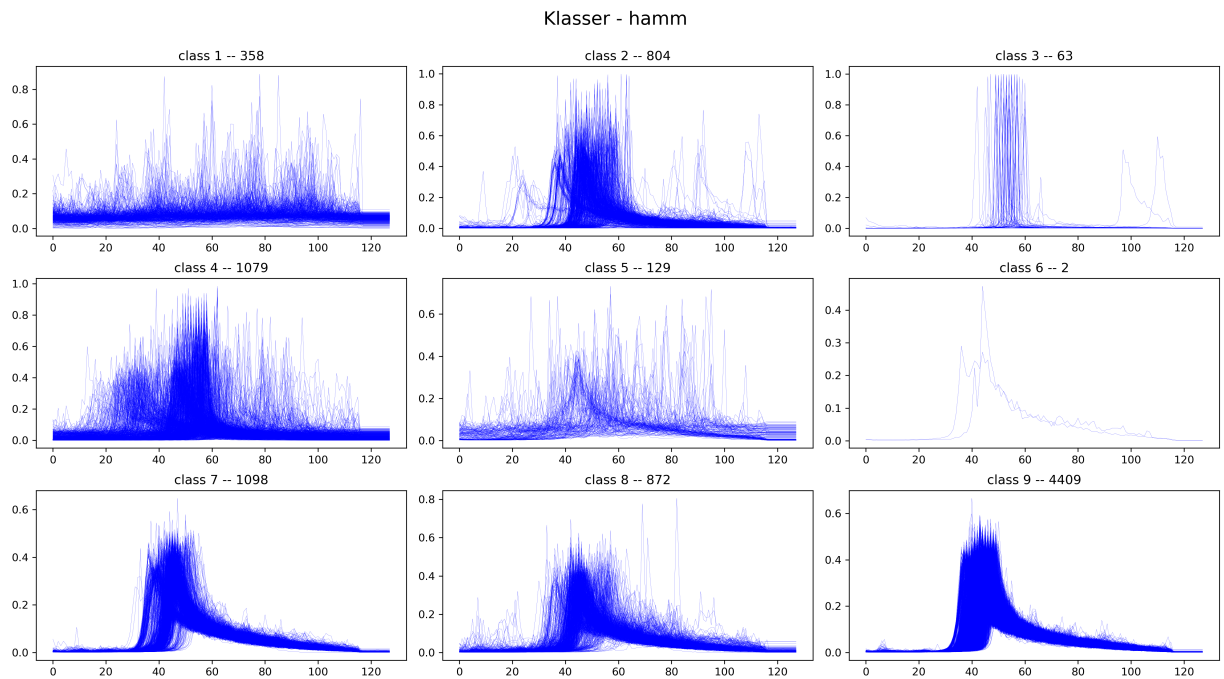
Figur B.18: Predikerte pulser fra valideringsområde rundt Harstad.



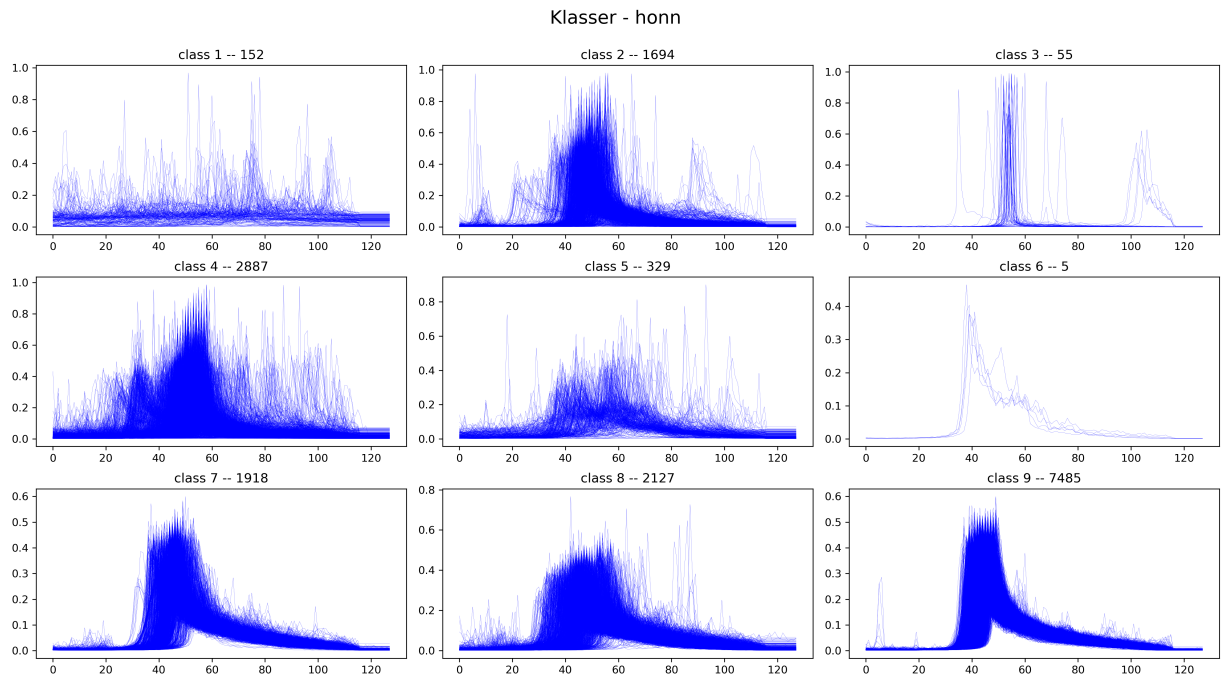
Figur B.19: Predikerte pulser fra valideringsområde rundt Andenes.



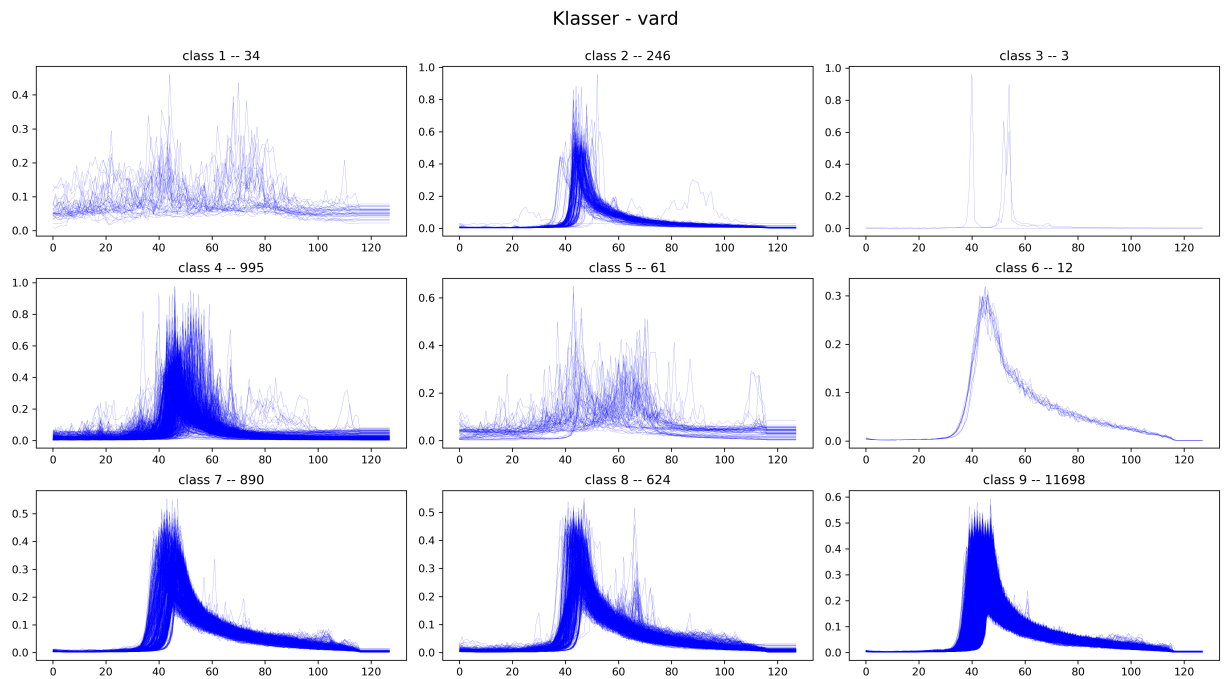
Figur B.20: Predikerte pulser fra valideringsområde rundt Tromsø.



Figur B.21: Predikerte pulser fra valideringsområde rundt Hammerfest.



Figur B.22: Predikerte pulser fra valideringsområde rundt Honningsvåg.



Figur B.23: Predikerte pulser fra valideringsområde rundt Vardø.



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway