



Ruteprediksjon av maritim trafikk med nevrale nettverk

Marius Bye Hagen

Master i datateknologi

Innlevert: juni 2018

Hovedveileder: Trond Aalberg, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknologi og informatikk

Sammendrag

Langs norskekysten forekommer det i snitt rundt 70 grunnstøtinger årlig. Disse kan medføre tap av liv, miljøutslipp, og store økonomiske kostnader. En analyse av ulykkesstatistikken fra norskekysten viser at 70% av alle grunnstøtinger skyldes menneskelige feil. Det er derfor sett et behov for å modernisere overvåkingen av skipstrafikken.

Nyere forskning har vist lovende resultater for prediksjon av posisjonsdata og tidsseriedata gjennom bruk av nevrale nettverk. For videre forskning i dette området har dette prosjektet fått tilgang på tre år med historiske AIS-data for skipstrafikken i Norge. Disse er benyttet for å prøve å predikere et fartøys rute på et geografisk avgrenset område. Målet er å se om man med nøyaktighet kan predikere framtidige verdier til et fartøy på kysten. Det er forsøkt både ettstegsprediksjon og prediksjon lengre fram i tid. Dette er gjort for flere arkitekturer av forskjellige typer nevrale nettverk.

Alle arkitekturer som er utprøvd viser gode resultater på ettstegsprediksjon, mens det på prediksjon av flere posisjoner fremover i tid sees varierende resultater blant de forskjellige arkitekturer. Noen av de utprøvd arkitekturer klarer dette godt i en sving i ruten, andre fikk en del problemer i denne. Det gjennomgås også en dataanalyse av den brukte AIS-dataen som viser en del utfordringer ved å bruke AIS-data til slike formål.

Abstract

Along the Norwegian coastline it is estimated that there are approximately 70 groundings on average every year. These can lead to loss of human life, environmental emissions, and big financial cost. An analysis of accidents statistics from the Norwegian coastline shows that 70% of all groundings are due to human error. These numbers show a need for modernizing the surveillance of the coastal traffic in Norway.

Recent studies in this field have presented promising results using neural networks for prediction on geospatial data and time series data. To further research in this area, this master thesis project has received access to three years worth of historical AIS data of the coastal traffic in Norway, which has been implemented in trials to predict a vessel's trajectory within a geographically defined area. To determine if it is possible to accurately predict future positions of a vessel on the coast, both one step prediction and prediction of multiple steps ahead in time are used. This is explored on multiple architectures for different types of neural networks.

All tested architectures demonstrate effective results on "one step ahead"-prediction. Whereas prediction of "multiple steps ahead in time" yielded varying results for different architectures. For example, a turn in the route was managed successfully by some of the architectures, while others exhibited difficulty in predicting that turn. A data analysis of the used AIS data is indicative of some of the challenges that can occur when using AIS data for these purposes.

Forord

Denne masteroppgaven ble skrevet våren 2018 på Norges teknisk-naturvitenskapelige universitet (NTNU), ved fakultet for informasjonsteknologi og elektronikk, avdeling for data og kunstig intelligens. Oppgaven ble gitt av Norconsult Informasjonssystemer AS avdeling Fundator.

Jeg vil spesielt takke mine veiledere Trond Aalberg og Sindre Sundvall Fjermestad for god veiledning og faglige diskusjoner gjennom arbeidet med denne oppgaven. Jeg vil også takke Norconsult Informasjonssystemer AS avdeling Fundator for å stille med kontorplass, datautstyr og data for gjennomføringen av eksperimentene. Jeg vil takke alle ansatte hos avdeling Fundator som har gitt et trivelig og inkluderende miljø å jobbe i. Av disse vil jeg spesielt takke Thomas Hjelde Thoresen som har hjulpet til med dataauthenting, Andreas Ravnestad som har gitt god informasjon om bakgrunnen for oppgaven, og Jørgen Longva som har introdusert meg og veileder fra NTNU om Fundators visjon med oppgaven. Tilslutt vil jeg takke min familie og venner som har gitt god støtte gjennom arbeidet med oppgaven, spesielt min samboer og gode venner som har lest gjennom oppgaven og gitt gode og konstruktive tilbakemeldinger.

Trondheim, Juni 2018

Marius Hagen

Innhold

SAMMENDRAG	I
ABSTRACT	III
FORORD	V
FIGURER	IX
TABELLER	X
AKRONYMER	XI
1 INNLEDNING	1
1.1 FORMÅL MED OPPGAVEN	1
1.2 AVGRENSNING	2
1.3 PROBLEMSTILLING.....	2
1.4 BIDRAG.....	3
2 BAKGRUNN	5
2.1 ULYKKESTATISTIKKER PÅ NORSKE FARVANN	5
2.2 TIDLIGERE ARBEID VED FUNDATOR	9
2.2.1 <i>Fundators bakgrunn for prosjektet</i>	9
2.2.2 <i>Fundators løsning</i>	10
3 TEORI	13
3.1 AIS OG SSNN.....	14
3.2 DBSCAN.....	16
3.3 NEVRALE NETTVERK.....	17
3.3.1 <i>Standard nevralt nettverk</i>	17
3.3.2 <i>RNN</i>	17
4 RESULTATER VED TIDLIGERE FORSKNING	19
4.1 DBSCAN.....	19
4.2 AVVIKSDETEKSJON MED STATISTIKK OG SANNSYNLIGHET KOMBINERT MED DBSCAN.....	20
4.3 NEVRALE NETTVERK.....	20
5 METODE	25
5.1 OVERORDNET METODE	25
5.1.1 <i>Datapreparering</i>	25
5.1.2 <i>Vurdering av modellene</i>	25

5.2	DATA	26
5.2.1	<i>Avgrensing og forklaring</i>	26
5.2.2	<i>Visualisering av valgte område</i>	27
5.2.3	<i>Forberedelse av data</i>	28
5.3	MODELLER	30
5.3.1	<i>Standard nevralt nettverk</i>	30
5.3.2	<i>RNN og LSTM</i>	30
5.4	EKSPERIMENTER	31
5.4.1	<i>Prediksjon av ett punkt fram i tid</i>	32
5.4.2	<i>Prediksjon opp til ti punkter fram i tid i sving</i>	32
6	RESULTATER	33
6.1	STANDARD NEVRALT NETTVERK.....	35
6.2	RNN.....	39
6.3	LSTM	42
7	DISKUSJON	45
7.1	MODELLVALG	56
7.2	INPUT-PARAMETERVALG.....	57
8	KONKLUSJON OG VIDERE ARBEID	59
8.1	KONKLUSJON	59
8.2	VIDERE ARBEID	60
9	BIBLIOGRAFI	63

Figurer

Figur 2.1 Prosentvis fordeling av ulykkestyper.	5
Figur 2.2 Direkte årsaker til grunnstøtinger og kollisjoner i perioden 1984-2013, fordelt i to tidsepoker.	6
Figur 2.3 Direkte årsak i kategorien person for grunnstøtinger i perioden 1984 – 2013 delt i to tidsepoker.	7
Figur 2.4 Petroleums- og offshorehendelser i tidsperioden 2012 – 2016, kategorisert i hendelsestype.	8
Figur 3.1 DBSCAN, forskjellige punkttyper.	16
Figur 4.1 En modell av det nevrale nettverket, viser struktur, inputverdier og outputverdier.	21
Figur 4.2 Strukturen på ruteprediksjonen, viser hvordan klyngene og de nevrale nettverkene benyttes til prediksjonen.....	22
Figur 4.3 Resultat av ruteprediksjon av en rute på 24 timer	22
Figur 5.1 Plot av historiske data på ruten fra Mo i Rana til Bodø hvor den røde firkanten markerer området det skal testes i.	27
Figur 5.2 Data fra en seilas som skal benyttes til testingen	28
Figur 5.3 Utvalgt rute for testing av modellene, ruten går fra høyre til venstre.	32
Figur 6.1 Ettstegsprediksjon for standard nevrale nettverk.....	35
Figur 6.2 Ti-stegsprediksjon for standard nevrale nettverk	37
Figur 6.3 Ettstegsprediksjon for RNN.....	39
Figur 6.4 Ti-stegsprediksjon for RNN	40
Figur 6.5 Ettstegsprediksjon for LSTM	42
Figur 6.6 Ti-stegsprediksjon for LSTM	43
Figur 7.1 Testdatasett, utfordrende del markert	48
Figur 7.2 Seilas som svingte innom land på slutten av ruten, og har svak sving i område med ti-stegsprediksjon.	49
Figur 7.3 All treningsdata til modellene.....	50
Figur 7.4 Seilaser med tette data og svak sving i område hvor ti-stegsprediksjon er utført	51
Figur 7.5 Seilas med tette data og brå sving i predikert område	52
Figur 7.6 Seilaser med spredde data og hull i predikert område	53
Figur 7.7 Seilaser med kun data på slutten.....	55

Tabeller

Tabell 3.1 Innrapporteringsintervall for AIS-meldinger	15
Tabell 5.1 Utdrag fra kommadelt Excel-fil	26
Tabell 5.2 Forklaring av de forskjellige dataene i kommadelt Excel- fil.....	27
Tabell 6.1 Modeller for standard nevralt nettverk	33
Tabell 6.2 Modeller for RNN.....	34
Tabell 6.3 Modeller for LSTM.....	34

Akronymer

AIS	Automatisk Identifikasjonssystem
BP	Backpropagation
COG	Course Over Ground
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DBSCANSD	Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction
DNV GL	Det Norske Veritas Germanischer Lloyd
GV	Gravity Vectors
GRU	Gated Recurrent Unit
IMO	International Maritime Organization
Lat	Latitude
Lon	Longitude
LSTM	Long Short-Term Memory
MMSI	Maritime Mobile Service Identity
MSE	Mean Squared Error
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SDU	Sjøfartsdirektorates Ulykkesdatabase
SOG	Speed Over Ground
SSN	SafeSeaNet

SSNN

SafeSeaNet Norge

TB

Terabyte

1 Innledning

Denne oppgaven tar utgangspunkt i et behov for å modernisere overvåkningen av skipstrafikk. Langs norskekysten forekommer det mange ulykker hvert år, hvor de fleste skyldes menneskelig feil. Grunnstøting er den vanligste formen for ulykke langs norskekysten, og kan medføre tap av liv, miljøutslipp og store økonomiske kostnader [1].

Ideen er at man ved å bruke data fra et automatisk identifikasjonssystem (AIS) og data fra SafeSeaNet Norge (SSNN) kan utføre prediksjon på et fartøys framtidige posisjon, samt detektere avvik på et fartøys rute i sanntid. AIS inneholder informasjon om et fartøys posisjon, fart og retning på et gitt tidspunkt og SSNN inneholder et fartøys til- og fra-havn. Begge inneholder identifikasjonen til et fartøy noe som gjør det mulig å koble disse dataene sammen.

Oppgaven er gitt av Norconsult informasjonssystemer AS avdeling Fundator heretter omtalt som Fundator. Fundator har blitt gitt tilgang på tre år med historiske data, som muliggjør bruk av maskinlæring for prediksjon.

I denne oppgaven sees det på hvordan tidligere forskning har løst lignende problemer og forsøkt å predikere et fartøys posisjon fram i tid. Det er ønskelig å bestemme om man kan predikere et fartøys posisjon fram i tid basert på ruten fartøyet hittil har valgt, og eventuelt hvor mange punkter fram i tid man kan predikere fartøyets posisjon.

1.1 Formål med oppgaven

I denne oppgaven er det av interesse å sammenligne standard nevrale nettverk opp mot rekurrente nevrale nettverk. I samarbeid med oppgavegiver ble det valgt å se på nettverkstypene: standard nevrale nettverk, standard RNN (Recurrent Neural Network) og LSTM (Long Short – Term Memory). Dette for å se om man kan benytte nevrale nettverk til ruteprediksjon i sanntid og hvilke av disse typene som ser ut til å være best egnet til å predikere på AIS-data. Dette er et viktig steg for å videre kunne benytte denne prediksjonen for å detektere avvik.

Målet med oppgaven er ikke å finne den optimale arkitekturen for en type nevrale nettverk, men undersøke om nevrale nettverk kan være aktuelle for å predikere ruten til et fartøy i sanntid basert på AIS-data. På sikt er det ønskelig med en avviksdeteksjonsmekanikk som inkluderer nevrale nettverk som predikerer ruter fram i tid for å kunne avverge grunnstøtinger.

En av grunnene til at det er ønskelig å se om man kan benytte nevrale nettverk til å predikere på AIS-data er at det eksisterer mye treningsdata og derfor er det viktig at algoritmene for prediksjon skal kunne re-trenes med kun nye data.

I kapittel 2 presenteres data som viser hvorfor det er viktig å kunne ha et system som kan predikere avvik på kysten.

1.2 Avgrensning

Det er i denne studien ikke valgt å utprøve modeller for avviksdeteksjon, men å utelukkende se på framtidsprediksjon på AIS-data. Framtidsprediksjon er et nødvendig steg for å kunne predikere om et fartøy avviker fra forventet rute. Det at det ikke eksisterer noen data som er klassifisert som ulykker eller avvik, medfører at det må gjøres en grundigere undersøkelse av hvordan dette skal gjøres.

Det er også gjort en avgrensning på det geografiske området denne forskningen skal forsøke å predikere data på. Denne avgrensningen er gjort fordi det å forsøke å trene nettverk på hele norskekysten vil bli et for stort omfang. De tre årene med historisk data fra hele norskekysten utgjør rundt 3TB. Ved å benytte data fra én til- og fra-havn i et mindre geografisk område kan man få testet prediksjonsevnen til de nevrale nettverkene på AIS-data, samt muliggjøre oppskalering om det skulle være ønskelig.

1.3 Problemstilling

Oppgaven skal undersøke om nevrale nettverk kan være aktuelle for å predikere ruter i sanntid basert på AIS-data. For å kunne finne ut om nevrale nettverk er en god løsning på utfordringene oppgaven ønsker å løse, må man se hvor nøyaktig et nevralt nettverk kan predikere posisjonen og ruten til et fartøy fram i tid. Det er også viktig å se om nettverket klarer å predikere ruten til et fartøy langt nok fram i tid, til at tiltak som forhindrer ulykken kan iverksettes. Det skal også undersøkes om man ved bruk av rekurrente nevrale nettverk kan oppnå en større nøyaktighet enn ved standard nevrale nettverk. Dette fører til problemstillingen:

- Kan man predikere ruten til et fartøy ved bruk av nevrale nettverk på AIS-data?
 - Kan man predikere én AIS-melding framover i tid ved bruk av nevrale nettverk?

- Er det mulig å predikere ruten lengre fram i tid?
- Kan man ved å bruke rekurrente nevrale nettverk oppnå større nøyaktighet enn ved bruk av standard nevrale nettverk?

1.4 Bidrag

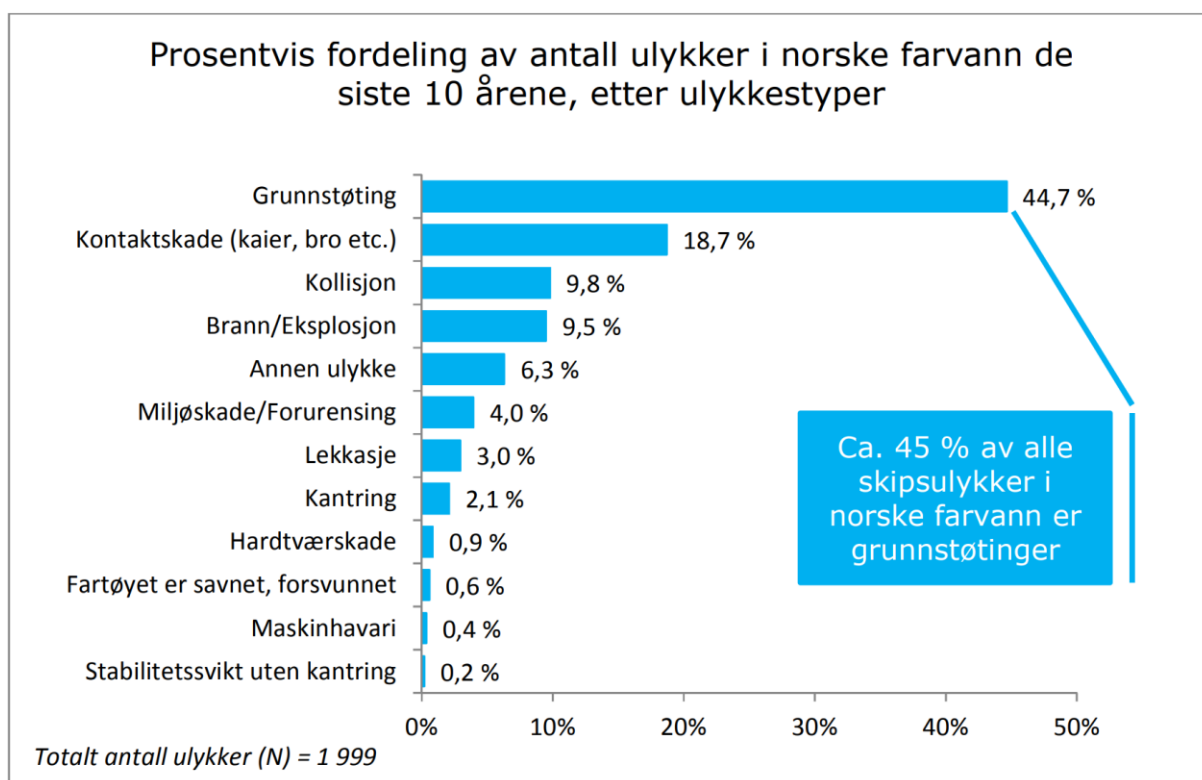
- Det har blitt hentet data fra ulykkesstatistikker og framtidsprognoser på norskekysten som viser viktigheten av å forbedre overvåkingen av kysttrafikken.
- Det er utført et litteraturstudium på anvendelsen av klyngingsalgoritmen DBSCAN på AIS-data og annen posisjonsdata, både for anvendelse på ruteprediksjon og muligheten til å kunne utføre en statistikkanalyse på posisjonsdataen.
- Et studium er også gjennomført på bruken av nevrale nettverk på kystdata og hvilke typer nettverk som kan egne seg på posisjonsdata og tidsseriedata, slik som AIS-data.
- Det er utført eksperimenter med bruk av forskjellige typer nevrale nettverk på et geografisk avgrenset område for å se om man nøyaktig kan predikere fartøys posisjon ved bruk av AIS-data. Resultatene fra disse eksperimentene er dokumentert og diskutert.
- Det er utført en analyse av AIS-dataen tilgjengelig for det geografiske området og diskutert feilkilder i disse dataene. Det er også diskutert hvilke parametere man kunne ha inkludert for å oppnå bedre resultater.

2 Bakgrunn

2.1 Ulykkesstatistikker på norske farvann

Kystverket har sammen med DNV GL (Det Norske Veritas Germanischer Lloyd) i 2014 utarbeidet en helhetlig sjøsikkerhetsanalyse for norske farvann. Denne består av flere rapporter som blant annet omhandler ulykkesanalyser og framtidsprognoser [2].

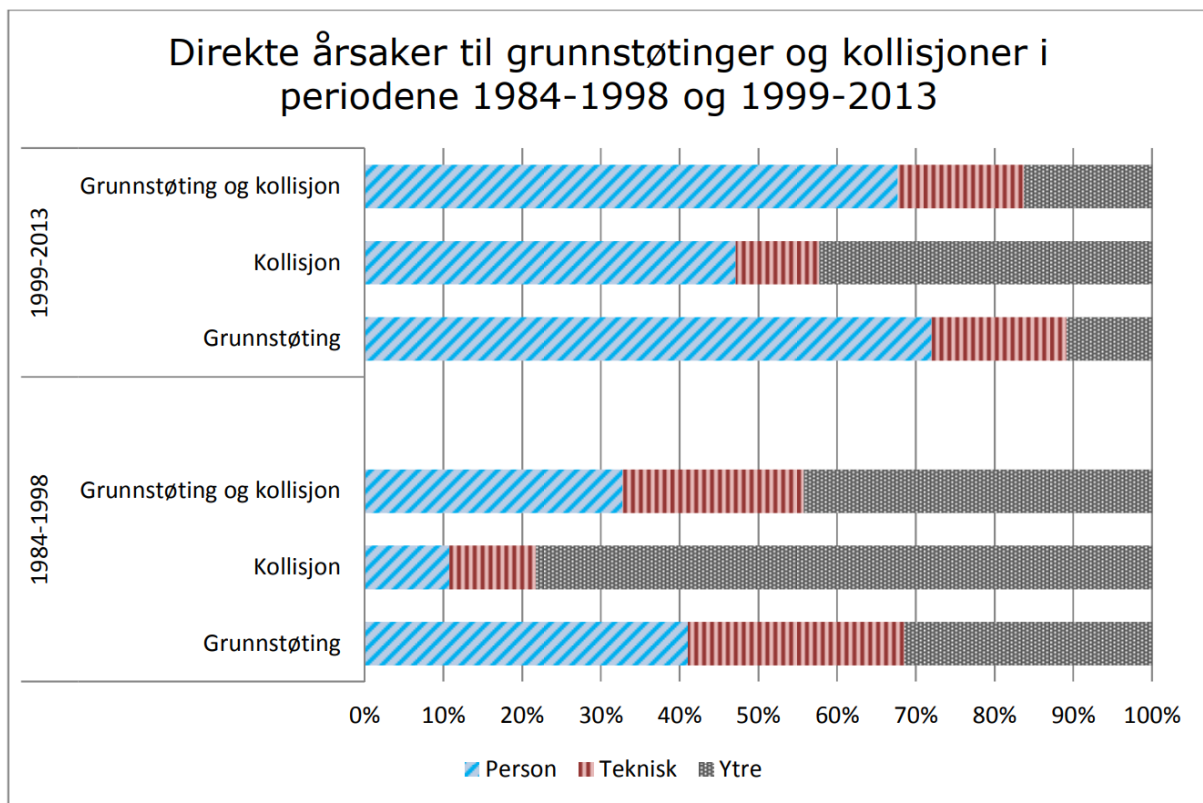
Figur 2.1 [3] viser at grunnstøtinger utgjør 44,7% av alle innrapporterte ulykker og dominerer ulykkesbildet i norske farvann, da neste type som er kontaktskade utgjør 18,7%. Dette er en analyse basert på tall fra alle ulykker i tidsperioden 2003 – 2013 hentet fra Sjøfartsdirektorates ulykkesdatabase (SDU) [3].



Figur 2.1 Prosentvis fordeling av ulykkestyper.

Norge er ikke det eneste landet hvor grunnstøtinger dominerer ulykkesbildet. Andre land med lange utfordrende kystlinjer som Sverige, Danmark og Storbritannia har grunnstøtinger som den mest fremtredende ulykkestypen [3].

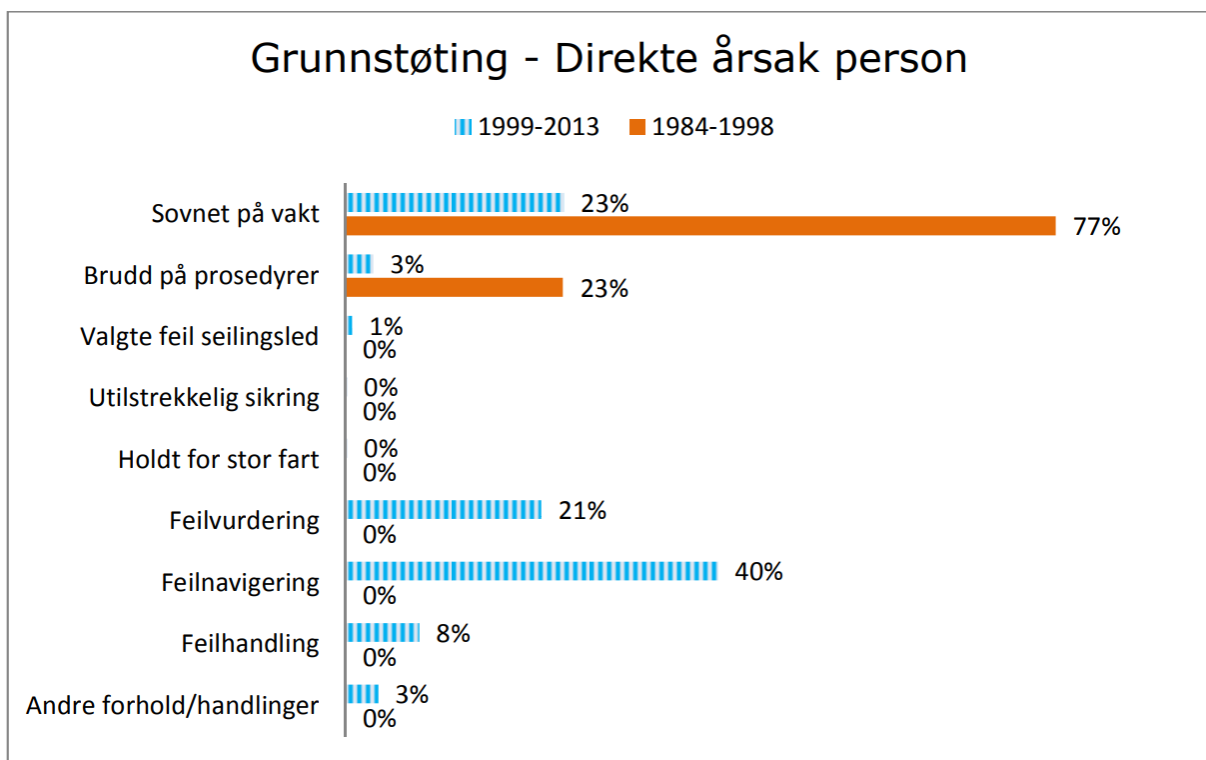
Sjøsikkerhetsanalysen 2014 inneholdt også en årsaksanalyse. Statistikken viser at i tidsperioden 1999 – 2013 skyldes over 70% av alle grunnstøtinger menneskelig feil. Dette er en økning på 30% fra perioden 1984 – 1998. Økningen skyldes ikke nødvendigvis at antall menneskelige feil har gått opp, men at det er blitt mer akseptert at ulykker kan skyldes menneskelig feil [4]. Figur 2.2 [4] viser en grafisk framstilling av dette. Her kan man også se at over 15% av grunnstøtinger skyldes tekniske feil i perioden 1999 – 2013, og kun litt over 10% skyldes ytre påvirkning. Årsaken teknisk feil for grunnstøtinger minket i underkant av 5%, mens ytre feil minket med rundt 20%. Dette kan tyde på at de ulykkene i tidsperioden 1984 – 1998 som skyldes menneskelig feil, ble kategorisert som ytre feil. Minkingen av tekniske feil kan være på grunn av at teknologien har blitt bedre i løpet av de siste 15 årene.



Figur 2.2 Direkte årsaker til grunnstøtinger og kollisjoner i perioden 1984-2013, fordelt i to tidsepoker.

Når over 70% av alle innrapporterte grunnstøtinger skyldes menneskelig feil, er det interessant å se på de underliggende feilene. Figur 2.3 [4] viser en analyse av data fra SDU på direkte årsak for grunnstøtinger i de tilfellene der menneskelig feil er grunnen til ulykkene.

Dataene fra 1993 – 2013 er spesielt interessante, da disse er nyere og det har blitt rapportert i mer detaljerte kategorier i denne perioden enn tidligere. Perioden 1984 – 1998 er inndelt i to kategorier, mens det i tidsperioden 1999 – 2013 ble innrapportert i syv kategorier. Her kan man se at i denne perioden utgjorde *feilnavigering* 40% av alle innrapporterte grunnstøtinger som skyldes menneskelig feil. Denne dominerer blant de direkte årsakene, da neste direkte årsak *sovnet på vakt* er skyld i 23%. Denne er igjen fulgt av kategorien *feilvurdering* som har skyld i 21% av grunnstøtinger.

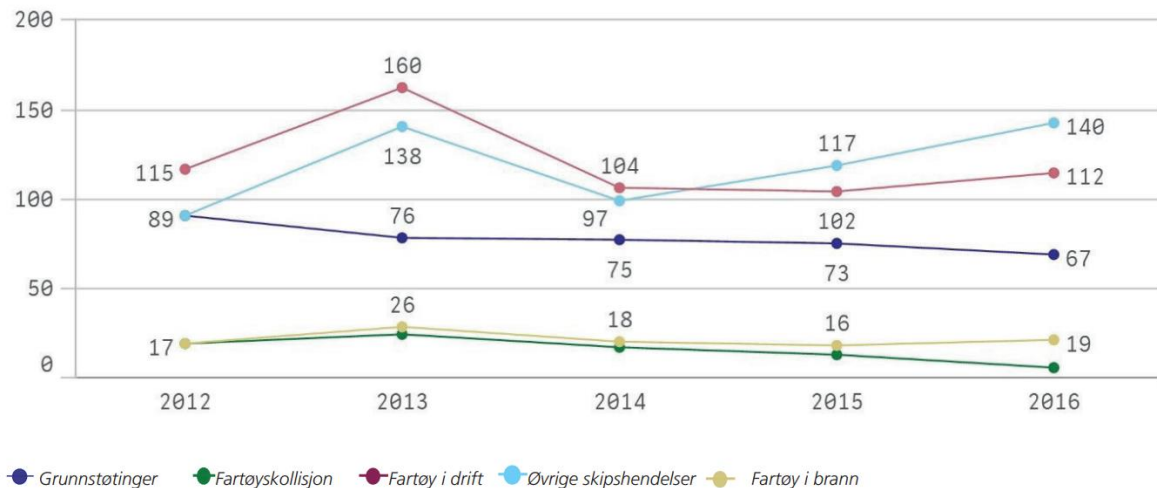


Figur 2.3 Direkte årsak i kategorien person for grunnstøtinger i perioden 1984 – 2013 delt i to tidsepoker.

DNV GL har også laget en framtidsprognose, hvor det er gjort en analyse av hvordan skipstrafikken og ulykkesbildet vil se ut i 2040. Størst fokus er lagt på ulykker som vil innebære tap av menneskeliv. Det forventes at det i 2040 vil forekomme 1,2 grunnstøtinger

med tap av menneskeliv, 0,3 flere enn hva som var forventet verdi i 2013, altså en økning på 32%. Dette er tall basert på ingen endring i dagens systemer [5].

Figur 2.4 [6] viser at det ble innrapportert 89 grunnstøtinger i 2012. Man kan se at det lå jevnt rundt 75 i 2013, 2014 og 2015. I 2016 var det en liten reduksjon fra dette ned til 67 grunnstøtinger.



Figur 2.4 Petroleums- og offshorehendelser i tidsperioden 2012 – 2016, kategorisert i hendelsestype.

Det at det de siste fire årene har vært et snitt på 75,5 grunnstøtinger i året er for høyt ifølge beredskapsdirektør i Kystverket Johan Marius Ly [7]. Grunnstøtinger flest fører ikke til akutt forurensning, men de største forurensningsulykkene langs Norskekysten har skjedd som en følge av grunnstøtinger [6].

Kystverket har selv publisert en sluttrapport på sjøsikkerhetsanalysen 2014. I denne rapporten var ett av de prioriterte tiltakene at man bør satse på utvikling og det å ta i bruk ny teknologi som kan forutse og avverge ulykker [8].

Det at over 70% av alle grunnstøtinger som skjer langs norskekysten skyldes menneskelig feil, hvorav 40% på grunn av feilnavigering, tyder på at et system som kan predikere ruter til fartøy og detekttere avvik på disse vil kunne bidra til å forhindre flere av disse ulykkene.

2.2 Tidligere arbeid ved Fundator

2.2.1 Fundators bakgrunn for prosjektet

Kystverket har vært kunde hos Fundator siden 2005. Samarbeidet startet med et prosjekt for å utvikle en database som dannet grunnlaget for SafeSeaNet Norge (SSNN), noe som Fundator fikk rammeavtaler på å utvikle. Dette gjelder blant annet integrasjoner mot det norske forsvaret, politiet, tollvesenet og norske havner¹.

SSNN er integrert med det sentrale SafeSeaNet (SSN) systemet hvor Norge er et av flere medlemsland. SSN er en europeisk standard forvaltet av European Maritime Safety Agency (EMSA). Mange skip er meldepliktige, som medfører at de må rapportere inn til SSNN når de skal fra et sted til et annet på kysten. Denne ordningen ble innført for å forenkle rutiner som tidligere ble gjort med fysiske skjemaer.

SSNN er en av to viktige datakilder i skipstrafikken for det prosjektet Fundator nå har startet opp i samarbeid med Kystverket. Den andre datakilden er automatisk identifikasjonssystem (AIS) data. Dette systemet har eksistert lenge, og ble utviklet for å unngå kollisjoner mellom båter. AIS-data fra et fartøy blir sendt ved at transpondere på båter sender et signal som går over radio (VHF). Dataene er ikke kryptert, og krever heller ikke noen form for autentisering for sender og mottaker. Den inneholder posisjonen, farten, og retningen til båten.

Det eksisterer flere kilder til støy i AIS dataene. De kan være unøyaktige i tilfeller der GPS ikke er kalibrert. Retningen til fartøyet kan være avhengig av strøm og vind. Transponderen krever en manuell prosess for å sette inn identifikasjonsnummer til båten. Dersom den manuelle prosessen ikke blir gjort riktig kan det medføre at en båt utgir seg for å være en annen. Med andre ord eksisterer det mange kilder til støy i dataen, noe som er en typisk utfordring med AIS. Fordelen er at det eksisterer mye data; Fundator har fått tilgang på tre år med AIS-data på skip som har ferdet i norske farvann fra Kystverket. Dette utgjør rundt 3TB med rå AIS-data, noe som gjør det statistisk mulig å kunne eliminere mye av støyen.

Fundators utvalgte metode for dette er beskrevet i kapittel 0.

Kystverkets oppgave i Norge er å skape trygghet på norske farvann for å unngå tap av liv, berge miljøet og beredskap. De forvalter den norske skipsnæringen og miljøet den påvirker. Trafikken på kysten har vært belemret med dumping av avfall, brenning av diesel og annen

¹ Andreas Ravnstad, jobbet for Fundator siden 2008 og har i perioden 2009 – 2016 vært engasjert hos Kystverket.

forurensning. Slik atferd på kysten har bidratt til at havet er veldig forøplet. Det eksisterer den dag i dag ingen god overvåkning for å detektere slik adferd. Et annet stort problem på kysten er grunnstøtinger, en båt som går på grunn kan begynne å lekke olje, noe medfører veldig store konsekvenser både miljømessig og økonomisk. Kystverket må da rykke ut for å rydde opp etter grunnstøtingen. En grunnstøting skyldes stort sett menneskelige eller tekniske feil [4].

Fundators visjon for dette prosjektet er på sikt å kunne detektere avvik i sanntid på skipstrafikken. Ideen er at dette kan nås med å kombinere AIS-data, SSNN data og maskinlæring. Når man kombinerer AIS-data og SSNN vet man hvilken fra- og til-havn et fartøy har. Dette er forskjellen på hva Fundator har gjort sammenlignet med hva Fundator fant av relevant forskning. Dette har gitt tilsynelatende gode resultater med klyngingsalgoritmen som er brukt.

Det som så langt eksisterer er et system som baserer seg på klynging og statistikk. Modellen tar ikke høyde for at det klyngingsalgoritmen betrakter som støy, kan være en reell rute. Alternative ruter som brukes relativt få ganger i forhold til hovedrutene og kan feilaktig bli betraktet som støy. Det er ønskelig å få et mer raffinert system som gjenkjenner de alternative rutene.

I neste delkapittel vil løsningen og utfordringene med Fundators implementasjon av klynging og avviksdeteksjon bli presentert.

2.2.2 Fundators løsning

Fundator har utnyttet det faktum at både AIS-data og SSNN-data inneholder et fartøys identifikasjon. Dette gjør at man kan avgrense klyngingskjøringen fra hele Norge ned til seilas på til- og fra-havn. Dette utgjør rundt 14 000 seilas, som ble puttet i egne filer. Disse inneholder tre år med historiske AIS data for de seilasene som lot seg knytte opp mot SSNN. AIS dataene til disse filene er logaritmisk fordelt, noe som gjør at man ikke kan ha standard parametere til klyngingsalgoritmen. Det er urealistisk å manuelt analysere dataen i hvert av de 14 000 forskjellige datasettene for så å sette disse parameterne. Løsningen til Fundator på dette ble at klyngingen ble det siste steget av en større algoritme, hvor de første stegene gikk

på å analysere hvert av datasettene den skulle utføre klyngingen på. Dette for at algoritmen selv skulle kunne sette best mulig parametere for utførelsen av klyngingen ².

Det første som måtte gjøres når man analyserte en seilasgruppe er å få delt den opp i mindre biter. Innenfor en seilasgruppe følger fartøyer ofte ruter basert på størrelse og type noe som fører til for store variasjoner til at man kan få gode parametere for den. En annen grunn er at det ble for mye data å håndtere i en og samme analyse, både når det gjaldt tidsbruk og den tilgjengelige maskinvarens kraft og minne.

Klyngingsalgoritmen som ble benyttet er DBSCAN (Density-Based Spatial Clustering of Applications with Noise). DBSCAN er presentert i neste kapittel.

Etter klyngingen ble utført, ble hver klynge splittet opp i egne segmenter langs gjennomsnittsretningen. For alle segmentene ble gjennomsnittsposisjonen regnet ut i henhold til GV (Gravity Vectors) strategien til Liu, B., et al. [9]. I tillegg ble det utført utregning av både standardavvik og maksimumsavvik for posisjon, fart og retning på høyre og venstre side. Disse utregningene ble lagret i GV.

Avviksdeteksjonen benytter en interpolasjon mellom de to nærmeste GV for å detektere om fartøyet er innenfor normalen eller ikke.

En utfordring med denne modellen er at den baserer seg på at klyngingsalgoritmen detekterer alle mulige ruter et fartøy kan ta. Hvis ikke vil den varsle om avvik selv om et fartøy befinner seg i en alternativ rute som ble klassifisert som støy av klyngingsalgoritmen.

² Sindre Sundvall Fjermestad, var med i utviklingen av Fundators løsning. Har en mastergrad fra NTNU i Datateknikk med kunstig intelligens som spesialisering.

3 Teori

Ved å benytte posisjonsdata fra skipstrafikken eksisterer det flere framgangsmåter og tilnærminger for å kunne predikere en rute. En rute i denne sammenhengen er hvor posisjonsdataen har blitt gruppert på til- og fra-havn. Innad de grupperte dataene eksisterer det ofte flere ruter. Det er gjerne noen hovedruter som de aller fleste fartøyene følger, i tillegg kan man også se alternative ruter som bare et fåtall fartøy følger.

En utfordring med algoritmer som skal predikere ruten til et fartøy, det vil si finne ut hvor fartøyet trolig vil befinne seg lengre fram i tid, er å da kunne skille på disse alternative rutene et fartøy kan ta mellom en til- og fra-havn. En annen utfordring er at posisjonsdataene benyttet ofte inneholder støy [10].

Det kan være vanskelig å definere avvik på en rute fordi den grupperte posisjonsdataen ikke inneholder noe informasjon om hvorvidt dataen avviker fra normalen eller ikke. En mulig måte å bestemme om en rute ser ut til avviker fra normalen eller ikke, kan være å benytte statistikk og sannsynlighet som da bestemmer ut fra de lovlige rutene funnet hvor mye et fartøy kan bevege seg utenfor disse uten at det skal regnes som et avvik [9, 11].

Tidligere forskning har sett på bruk av DBSCAN (Density-Based Spatial Clustering of Applications with Noise) i forbindelse med opphoping av veitrafikk. Resultatene fra denne forskningen var gode, men det ble utført med spesifikke parametere på en konstruert situasjon, hvor blant annet effekten av lyskryss ikke var tatt høyde for [12]. Det er også sett på bruken av DBSCAN til å lage rammeverk for å detektere baner i et datasett. Disse hadde sine utfordringer hvor en var veldig sensitiv for parametersettingen i algoritmen, noe som krever stor domeneforståelse for hvert område hvor algoritmen skal anvendes [13]. En annen algoritme var ikke like sensitiv for parametersetting, men fungerte best på rette baner [14]. En ulempe hos begge disse var at de ikke tok høyde for fart og retning. Tidligere forskning har også sett på bruken av DBSCAN på AIS-data [9, 11, 15]. Disse vil bli introdusert i kapittel 4.

Det er også blitt sett på bruk av standard nevrale nettverk for prediksjon på maritim data. Tidligere forskning har sett på å bruke et ett-lags nevral nettverk for å predikere posisjonen indirekte ett steg fram i tid ved å predikere differansen i longitude og latitude. Nettverket hadde da fart, retning og differansen i longitude og latitude som input, og kun differansen til longitude og latitude som output. Dette ble benyttet for å predikere ett steg fram i tid.

Resultatene var tilsynelatende gode, men testomfanget var ikke spesielt stort, da det kun inneholdt 30 verdier med et tidsintervall på ett minutt [16].

En annen løsning var å trene og predikere nevralt nettverk med kun posisjonen til fartøyene. Her ble det også benyttet data som var hentet ut med ett minuttts intervall, og dermed argumenter for at man ikke trenger fart og retning da nettverket hadde flere tidligere posisjoner for å bestemme retningen og farten kunne bestemmes med avstanden mellom punktene. Dette ga gode resultater for prediksjon av tre punkter fram i tid [17]. Dette fungerer godt i et tilfelle hvor dataen kommer med et jevnt tidsintervall, problemet er at det ikke fungerer slik for AIS-data, fordi disse blir sendt med forskjellig tidsintervall basert på retningsendring og fart [18]. En tredje løsning gikk på å benytte fire tidligere longitude og latitude verdier som input og én longitude og latitude verdi som output, for å predikere ett punkt 15 minutter fram i tid. Dette ga varierende resultater hvor det enkelte steder ble feil på andre desimal for longitude og latitude [19].

Tidligere forskning har også sett på bruken av DBSCAN kombinert med standard nevralt nettverk for å predikere ruten til et skip ved bruk av AIS-data [20]. Denne framgangsmåten vil bli presentert i kapittel 4.

Innenfor familien RNN (Recurrent Neural Network) som er en familie nevralt nettverk for prosessering av sekvensielle data [21] har LSTM (Long Short – Term Memory) fått veldig gode resultater i de siste årene på flere applikasjonsområder som stemmegjenkjenning og språkoversetting [22, 23]. Det som kjennetegner disse applikasjonsområdene er at de krever at nettverket husker hva det tidligere har observert, altså dataen er på tidsserieformat.

Det er i tillegg sett på bruken av LSTM kombinert med DBSCAN for å predikere på posisjonen til fotgjengere innenfor et avgrenset område [24]. Det beste resultatet ble oppnådd ved å benytte et LSTM nettverk for hver klynge funnet med DBSCAN, hvor én klynge representerte én bane personer kan gå i. De konkluderte også med at et slikt rammeverk kunne benyttes på annen type trafikk, hvor maritim trafikk var spesifikt nevnt.

3.1 AIS og SSNN

AIS er et system introdusert av IMO (International Maritime Organization). AIS danner sammen med SSNN informasjonen de nevralt nettverkene vil få fra et fartøy i sanntid. SSNN står for informasjonen om fartøyets til- og fra-havn. Resten av informasjonen om fartøyet vil komme fra AIS.

Alle skip med bruttotonn på 300 og oppover som ferdes på internasjonale farvann, og kargoskip med 500 bruttotonn og oppover som ikke ferdes på internasjonale farvann, samt passasjerskip uansett størrelse er påkrevd å ha AIS installert [18].

AIS-data er delt inn i tre forskjellige typer data: statisk informasjon, dynamisk informasjon og seilasrelatert informasjon [18]:

- Statisk informasjon er informasjon som blir manuelt satt inn for et fartøy ved installasjon av AIS. Dette er da data som MMSI-nummer, lengde og type.
- Dynamisk informasjon er informasjon som blir automatisk oppdatert fra fartøyets sensorer som er koblet til AIS. Dette er da data som COG (Course Over Ground), SOG (Speed Over Ground) og skipets posisjon som har en nøyaktighet på omtrent 10 meter.
- Seilasrelatert informasjon er informasjon som blir manuelt satt og oppdatert under seilasen. Dette kan være informasjon om farlig last om bord, destinasjon og estimert tid til ankomst og ruteplan.

Tabell 3.1 [18] viser tidsintervallet på AIS-meldinger og hvordan dette endrer seg med skipets tilstand. Det vil si om det er fortøyd eller ankret opp, samt hastighet og om det endrer retning eller ikke.

Tabell 3.1 Innrapporteringsintervall for AIS-meldinger

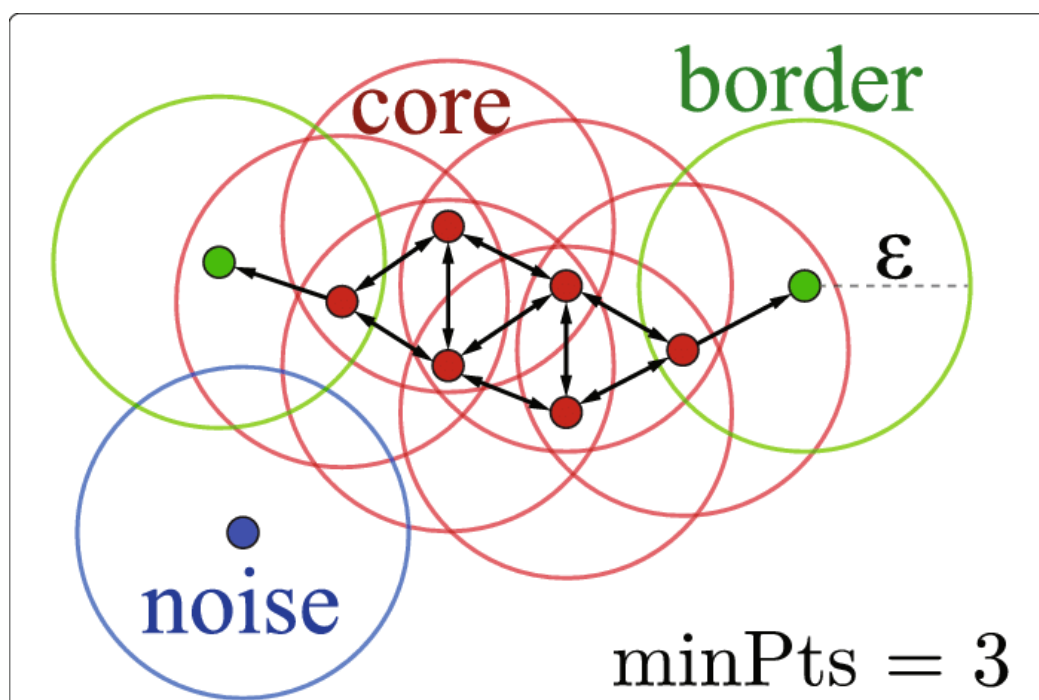
Skipstilstand	Innrapporteringsintervall
Ankret opp eller fortøyd og hastighet på under 3 knop	3 minutt
Ankret opp eller fortøyd og hastighet over 3 knop	10 sekund
Hastighet på 0 -14 knop	10 sekund
Hastighet på 0-14 knop og endrer retning	3 1/3 sekund
Hastighet på 14-23 knop	6 sekund
Hastighet på 14-23 knop og endrer retning	2 sekund
Hastighet på over 23 knop	2 sekund
Hastighet på over 23 knop og endrer retning	2 sekund

3.2 DBSCAN

DBSCAN er en tetthetsbasert klyngingsalgoritme introdusert av Ester, M, et.al. i 1996 [25]. Før man gjør en analyse av et datasett definerer man et minimum antall punkter (minPts) og en maksimal distanse (Eps). Det eksisterer tre typer punkter: kjernepunkt, grensepunkt og støypunkt.

Et punkt er et kjernepunkt hvis antall punkter i et nabolag med størrelse Eps rundt punktet er større eller lik minPts. Et punkt er et grensepunkt hvis det er i nabolaget til et kjernepunkt, men selv ikke kvalifiserer som kjernepunkt. Et punkt er et støypunkt hvis det verken er et kjernepunkt eller et grensepunkt.

Figur 3.1 [26] viser hvordan en typisk klynge funnet ved DBSCAN kan se ut, her er de røde punktene kjernepunktene. Som modellen viser, har alle kjernepunkt minimum tre punkter inkludert seg selv innenfor sitt nabolag siden minPts er satt til tre. Sirklene rundt hvert punkt illustrerer punktets nabolag med størrelse Eps. De grønne punktene er grensepunkter, da de har mindre enn tre punkter innenfor sitt nabolag, men minst ett av disse er et kjernepunkt. Kjernepunktene er ikke alle i samme Eps-nabolag, men er tetthetstilkoblet hverandre. Dette fører til at alle punktene som ikke er støy, altså alle sett bort fra det blå punktet er i en og samme klynge.



Figur 3.1 DBSCAN, forskjellige punkttyper.

3.3 Nevrale nettverk

Innenfor hovedgrenen nevrale nettverk, eksisterer det forskjellige typer som alle har sine ulike styrker. I dette kapitlet vil det bli presentert de nevrale nettverkene som denne forskningen skal benytte.

3.3.1 Standard nevral nettverk

Standard nevral nettverk er nevrale nettverk som benytter enveis tilkobling mellom node-lagene i nettverket. Noe som betyr at informasjonen i nettverket flyter kun en vei i nettverket fra input til output. Nettverket bruker BP (back-propagation), som vil si at det under trening sammenligner sin predikerte verdi med det verdien faktisk var, og deretter endrer vektene som sendes til nodene i nettverket basert på dette [16, 20, 27].

3.3.2 RNN

RNN er en familie nevrale nettverk for prosessering av sekvensielle data [21]. Et standard RNN har noen viktige kvaliteter som skiller dem fra standard nevral nettverk, som gjør dem egnet for sekvensielle data. Der et standard nevral nettverk anser alle tilfeller som uavhengige, tar et standard RNN høyde for tidligere tilfeller når det beregner outputen til det nåværende tilfellet. Det har muligheter for å lagre indre forhold i nettverket over tid, noe som gir et dynamisk minne i nettverket [21].

Det eksisterer mange forskjellige implementasjoner av RNN [21, 28-36]. LSTM [36] er et av disse som det fokuseres i større grad på dette prosjektet. Standard RNN heretter omtalt som RNN er også en modelltype det vil fokuseres på i dette prosjektet.

3.3.2.1 LSTM

LSTM ble introdusert av Hochreiter, S. og J. Schmidhuber [36] i 1997 og ble forbedret i 1999 av Gers, F.A., J. Schmidhuber, og F. Cummins [37] ved å introdusere en port for å glemme.

Et LSTM består av et sett rekurrente koblede blokker. Innad i en slik blokk er det tre porter, inputport, outputport og en port for å glemme. Et LSTM har mulighet til å huske aktiveringer i tid som et RNN, men kan lære når det skal glemme eller huske data, dette gjøres ved hjelp av de introduserte portene. Nettverket kan kun kommunisere med blokkene via de introduserte portene [38].

Det eksisterer mange varianter av LSTM. LSTM med kikkhulltilkoblinger [39], GRU (Gated Recurrent Unit) [40], og Toveis LSTM [22, 38] er noen av disse. I denne forskningen er det valgt å se på LSTM som algoritmetype og ikke se på de forskjellige variantene.

4 Resultater ved tidligere forskning

4.1 DBSCAN

Tidligere forskning har sett på bruken av DBSCAN på AIS-data. Pallotta et al. foreslo i 2013 en metodikk kalt TREAD (Traffic Route Extraction and Anomaly Detection) [11]. Som så på AIS-data fra forskjellige områder i verden og forsøkte å kjøre en klyngingsalgoritme på disse. Deretter predikerte de hvilken rute et fartøy ville kjøre og detekterte avvik i denne.

Forskningsgruppen benyttet DBSCAN som klyngingsalgoritme, og fant at dette ga veldig gode resultater i områder med høy trafikk tetthet og rigide rutesystemer hvor de fikk klynget 95% av trafikken. Når de kjørte klyngingen på områder der rutesystemene ikke var like rigide, eller der det var forskjell i trafikk tettheten fungerte ikke DBSCAN spesielt godt. Det dårligste resultatet de fikk var på trafikken i det indiske hav hvor kun 40% av trafikken ble klynget. De brukte 50 dager med historiske data i den kjøringen hvor de fikk klynget 95% av trafikken.

Dobrkovic et al. hadde i 2015 hadde også utfordringer med DBSCAN [15]. De forsøkte å utføre klynging på AIS-data mellom Nederland og Storbritannia, hvor det er geografisk stor varians i hvor mange ruter som eksisterer innenfor et område. Noen av rutene ble detektert av DBSCAN som forventet, mens andre ble kun til en stor klynge da tettheten på AIS-data i dette området ble for stor i forhold til parameterne satt i DBSCAN. Det ble foreslått å forsøke å dele opp disse geografiske områdene som en mulig løsning på dette, men det ble ikke gjennomført. Det konkluderes med at algoritmene deres burde testes med større datasett, helst flere måneder med data noe som tyder på at det settet det ble eksperimentert med var betraktelig mindre.

Liu et al. [9] publiserte i 2014 deres resultater med bruk av en implementasjon av DBSCAN kalt DBSCANSD (Density-Based Spatial Clustering of Applications with Noise considering Speed and Direction) som hadde gode resultater på deres to utvalgte områder. En annen styrke er at de tok med fart og retning som parametere i klyngingsalgoritmen. En begrensning de hadde var at den er sensitiv på parameterne som på forhånd måtte settes, og krevde god domeneforståelse på området som algoritmen skulle kjøre klynging på. Det ble benyttet to måneder med historiske AIS-data på begge områdene.

4.2 Avviksdeteksjon med statistikk og sannsynlighet kombinert med DBSCAN

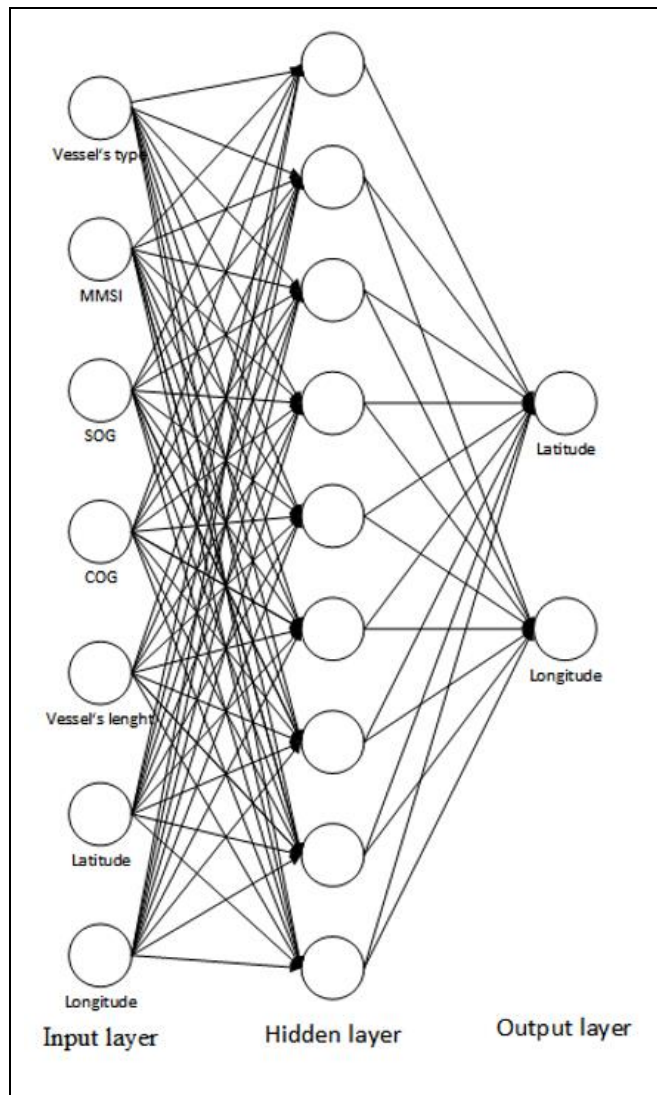
Pallotta et al. [11] så også på avviksdeteksjon. For å detektere avvik benyttes en statistikkmodell som ser på sannsynligheten for at et fartøy har den posisjonsvektoren i det observerte punktet. Det vil bli klassifisert som et avvik dersom denne er lav. Dette fungerer bra i et maritimt område der rutene er rigide og hvor det ikke er mye trafikk til vanlig. En statistikkmodell vil klassifisere for eksempel en unnamanøver som en anormalitet, noe som også er korrekt. I et område med høy trafikk vil derimot slike avvik forekomme ofte. Derfor er det ønsket at en avviksdeteksjon skal kunne skille på forventede avvik og avvik som potensielt kan lede til en farlig situasjon. Et annet problem er at det kan eksistere alternative ruter som kun benyttes i en gitt værtype eller situasjon, denne ruten kan ha blitt klassifisert som støy. Dette vil føre til at når fartøy benytter denne ruten vil sannsynlighetsmodellen alltid klassifisere den som anormalitet. Det betyr at selv i de områdene DBSCAN fungerte best på, hvor den fikk klynget opp til 95% av trafikken, vil en modell som kun benytter statistikk og sannsynlighet rapportere falske avvik.

Liu et al. [9] benyttet i deres klyngingsalgoritme både fart og retning, i tillegg til posisjon. Etter utførelsen av klyngen analyserte de klyngene og genererte GV (Gravity Vectors) for deres klynger. Disse inneholdt gjennomsnittet av fart, retning og posisjon, samt mediandistansen. De implementerte ingen avviksdeteksjon, men konkluderte med at GV kunne brukes til dette.

4.3 Nevrale nettverk

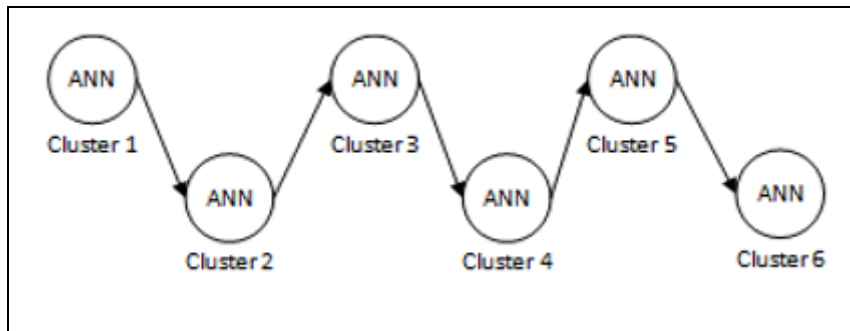
Ut fra kapittel 4.2 ser det ut til at det å bruke klassiske modeller for å finne avvik og predikere ruter på posisjonsdata i et miljø hvor det kan forekomme mange naturlige avvik, ikke er det beste alternativet. Dette vil kunne føre til mange falske rapporteringer om avvik. Derfor har denne forskningen valgt å se på nevrale nettverk som et alternativ for å kunne predikere ruter og senere detektere avvik.

Daranda [20] benyttet et datagrunnlag bestående av AIS-data. Modellen presentert hadde to hoveddeler. Den første delen besto av klyning av vendepunkt i ruten ved bruk av DBSCAN. Den andre delen av modellen besto av standard nevrale nettverk som trente på å predikere vendepunktet til fartøyet innad i de identifiserte klyngene.



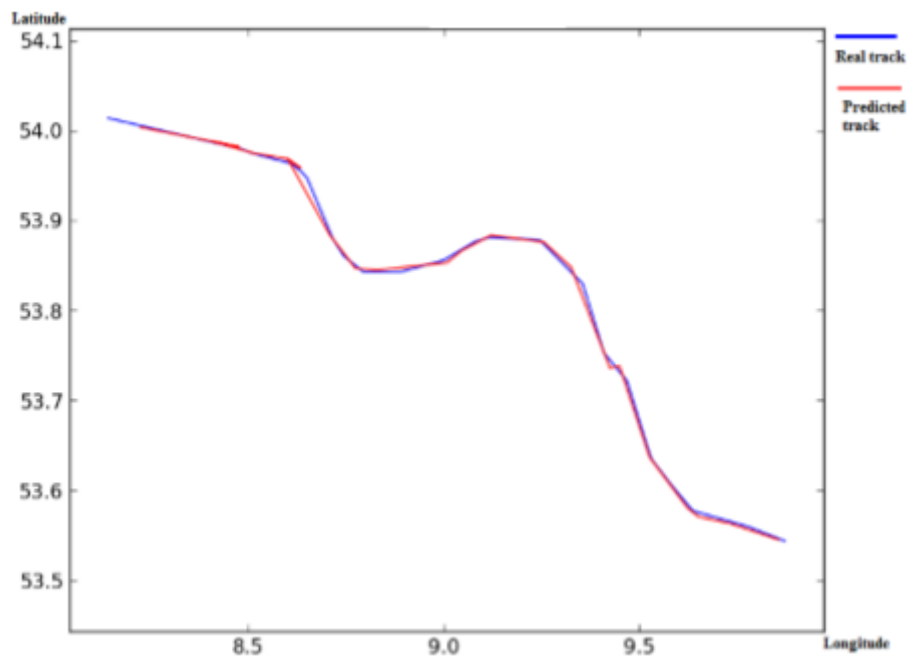
Figur 4.1 En modell av det nevrale nettverket, viser struktur, inputverdier og outputverdier

Fra Figur 4.1 [20] ser man at det nevrale nettverket besto av kun ett skjult lag. Inputen til algoritmen besto av data fra AIS: latitude, longitude, COG, SOG, MMSI, fartøytype og båtlengde. Outputen til nettverket er en prediksjon på Longitude og Latitude, altså fartøyets forventede posisjon.



Figur 4.2 Strukturen på ruteprediksjonen, viser hvordan klyngene og de nevrale nettverkene benyttes til prediksjonen

Figur 4.2 [20] viser hvordan Daranda predikerte rutene. Hver klynge fikk sitt eget nevrale nettverk. Nettverket blir da matet med input fra AIS-data for å predikere det neste vendepunkt (longitude og latitude) til fartøyet. Denne oppdelingen av nevrale nettverk fører til at hvert nettverk blir trent til å finne vendepunkt innenfor en liten del av kysten (innad i en klynge).



Figur 4.3 Resultat av ruteprediksjon av en rute på 24 timer

Figur 4.3 [20] viser at Daranda hadde gode resultater på prediksjonen. Her er visert den røde linjen den predikerte ruten og den blå linjen den faktiske ruten, som man kan se har den truffet godt på mange av punktene hvor det endres retning. Det må tas i betraktning at hver

klynge hadde sitt eget trenete nevralt nettverk, som eneste oppgave var å predikere longitude og latitude for svingpunktet. Som man ser fra Figur 4.3 [20] så har den predikerte ruten markante linjer hvor den faktiske ruten har mer avrundede svinger. Dette er en måte å predikere på som ser ut til å fungere godt i mer åpne områder hvor det er få svinger og fartøyene følger mest mulig rett kurs. I et trangt sund hvor et fartøy vil utføre mange svinger for å unngå å kjøre på land vil denne framgangsmåten trenge mange identifiserte svingpunkter.

Det å ha et nevralt nettverk for hvert svingpunkt, noe som skalerer til å bli veldig mange nevralt nettverk, er ikke nødvendigvis den optimale løsningen for denne forskningen. Ambisjonen med denne forskningen er å finne en løsning som kan skaleres opp til å bli benyttet på hele norskekysten. Det er derfor valgt å trenge nevralt nettverk på ett geografisk område. Dette krever mer trening av hvert nettverk og trolig må hvert nettverk være av en mer avansert struktur enn et standard nevralt nettverk med kun ett skjult lag, men kan føre til at nettverket bedre lærer å finne de alternative rutene for seilasene og lærer seg å predikere selve ruten og ikke kun et svingpunkt.

5 Metode

5.1 Overordnet metode

5.1.1 Datapreparering

For å anvende nevralt nettverk på et datasett må man først få dataen over på et format et nevralt nettverk kan forstå. Formatet kommer an på hvilken type nevralt nettverk man benytter, men det er en del fellestrekk for alle modellene.

Det vanlige er å dele opp datasettet i de tre kategoriene testdata, valideringsdata og treningsdata. Det er ønskelig å få mest mulig data inn i treningen og det er derfor valgt å ikke benytte valideringsdata. En annen grunn er at valideringen av modellene gjøres ved å se på resultatene fra treningskjøringen. Det er også forsøkt å predikere mer enn ett steg fram i tid, som er det nettverket trenes på. Dette kan føre til at modeller som er best på ettstegsprediksjon, ikke nødvendigvis er den beste modellen på flere steg fram i tid.

Splittingen av dataen er ofte gjort ved å ta en prosentandel av dataen som treningsdata og en annen prosentandel til validering og testing. For denne oppgaven var det ønskelig å ha en fullstendig rute som testdata og ikke ta ut en bestemt prosentandel av det totale datasettet. Derfor ble det valgt å ta ut én rute fra datasettet som treningsdata.

Videre må dataen prepareres i to lister, en som nettverket skal ta inn som input og en annen som nettverket skal ta å sjekke sine predikerte verdier mot, for å bestemme sin egen nøyaktighet. I disse listene er det viktig at innslagene inneholder like mange verdier og at disse verdiene representerer det samme på samme plass i hvert innslag i listen.

Noe som også ofte gjøres og vil bli gjort i disse eksperimentene er å normalisere dataene, slik at de inneholder kun verdier mellom 0 og 1. Dette gjør det lettere for nettverkene å trene [41].

5.1.2 Vurdering av modellene

I denne forskningen er det fokus på å se om det er mulig å anvende nevralt nettverk på datagrunnlaget. Dette gjør at eksperimentene og vurderingen av disse vil bli utført med en kvalitativ dataanalyse. For vurdering av resultatene for modellene vil det utføres en visuell

analyse. Der man ser på de predikerte resultatene opp mot de faktiske resultatene og vurderer nøyaktigheten på denne måten.

5.2 Data

5.2.1 Avgrensing og forklaring

For denne forskningen er det hentet ut data fra en database som inneholder tre år med historiske AIS-data som Fundator har fått tilgang til fra kystverket. Disse dataene er gruppert på til- og fra-havn og lagret i en kommadelt Excel-fil. I denne databasen er det lagret AIS-data fra kysttrafikk for alle havner i hele Norge. Det er valgt å se på data som går fra Mo i Rana til Bodø. Denne ruten er valgt fordi denne ruten inneholder en hel del øyer og trange sund noe som medfører at fartøyene må navigere seg gjennom disse og det blir da naturlig en del svinger i denne ruten. Grunnen til at Mo i Rana til Bodø ble valgt kontra Bodø til Mo i Rana er at det eksisterer omtrent dobbelt så mye historiske data for den retningen.

Tabell 5.1 viser et utdrag av innslagene i den kommadelte filen på tabellform. I tillegg til disse dataene er det også lagret avgang og ankomst, disse er det valgt å fjerne, da de vil være like for all dataen fordi dataen hentet ut til denne forskningen er kun fra en til- og fra-havn.

Tabell 5.2 viser forklaring på hva dataene i hver kolonne er.

Tabell 5.1 Utdrag fra kommadelt Excel-fil

mmsi	date_time_utc	Lon	Lat	Sog	Cog	v_id
231758000	2016-03-07 15:05:50	14.0707	66.3061	12.7	238.9	865739
231758000	2016-03-07 15:06:01	14.0694	66.3057	12.7	239.1	865739
231758000	2016-03-07 14:55:11	14.0682	66.3055	12.7	239.1	865739
231758000	2016-03-07 14:54:20	14.0670	66.3052	12.7	239.0	865739
231758000	2016-03-07 14:54:31	14.0656	66.3048	12.7	239.0	865739

Tabell 5.2 Forklaring av de forskjellige dataene i kommadelt Excel- fil

mmsi	date_time_utc	Lon	Lat	Sog	Cog	v_id
Unik id til hvert fartøy	Tid og dato for den gitte dataen	Longitude til fartøyet	Latitude til fartøyet	Fart over land til fartøyet	Kurs over land til fartøyet	Unikt nummer for hver seilas.

5.2.2 Visualisering av valgte område

Figur 5.1 viser rutene etter at datasettet ble rensket for data som hadde satt feil til- og fra-havn, medførte at de dukket opp i dette datasettet selv om det var helt klart at de ikke fulgte den gitte ruten. Her har hvert seilas fått en egen farge slik at man kan lettere skille på hver rute.

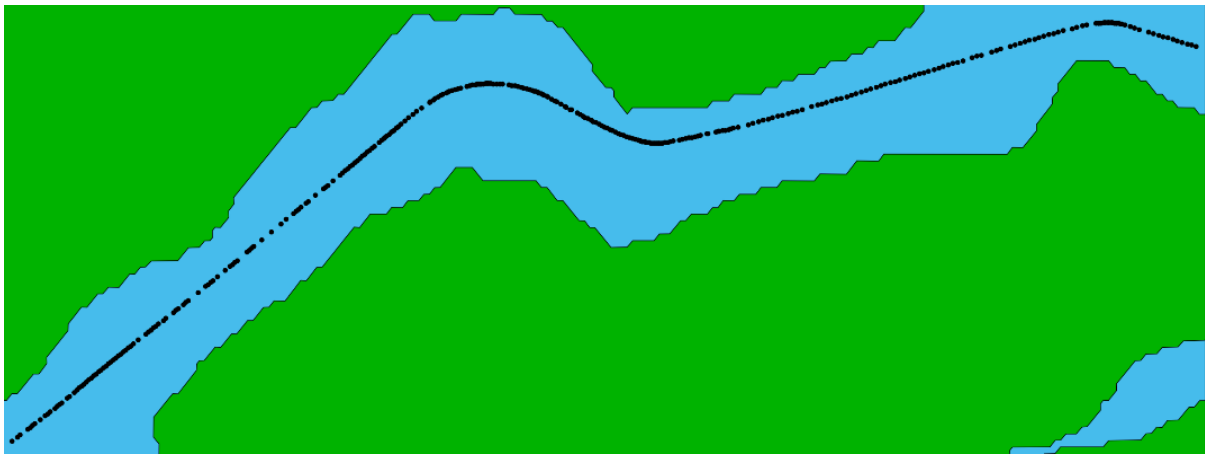


Figur 5.1 Plot av historiske data på ruten fra Mo i Rana til Bodø hvor den røde firkanten markerer området det skal testes i.

Det som er ønskelig er å se hvor godt modellene greier å predikere posisjonen til fartøyene. Dette kan testes ut på en mindre del av ruten. Det man ser fra Figur 5.1 er at alle seilaser begynner langs den samme ruten. I tillegg til at starten inneholder mest treningsdata er det

også en del svinger, noe det er ønskelig å finne ut om man kan predikere. Den røde firkanten viser valgt område for testing.

Figur 5.2 viser hvordan en seilas typisk er på dette området. Ruten begynner i høyre hjørne og slutter i venstre. Det geografiske området er avgrenset på en longitude fra 13.19 til 13.56 og latitude fra 66.1885 til 66.25. Dette tilsvarer ca. 41 155 meter i lengde og 6 650 meter i bredde.



Figur 5.2 Data fra en seilas som skal benyttes til testingen

5.2.3 Forberedelse av data

I den kommaseparerte Excel- filen har man de tilgjengelige dataene for alle seilaser utført i tre år på den gitte ruten. For alle modeller utnyttes det at dato og tidspunkt for AIS-meldingen i tillegg til identifikasjonen til fartøyet er tilgjengelig i dataen. Dette gjør at man kan gruppere dataene på seilaser og sortere dem på dato. Dette medfører at dataen blir på tidsserieformat, fordi innad i en seilas er første dato når seilassen startet og siste dato når seilassen slutter.

Etter at dataene er gruppert og sortert blir all data utenom longitude, latitude, COG og SOG rensket bort fra datasettet, dette fordi det kun er disse som er ønsket at nettverket skal ta i betraktning når det skal predikere framtidig posisjon for fartøyene. Etter dette blir dataene normalisert.

5.2.3.1 Datasett splitt

Ved å ha dataene gruppert på seilaser, kan man enkelt trekke ut en eller flere seilaser til å inngå i testsettet av modellene, mens resten av dataene inngår i treningssettet.

Det er valgt å trekke ut én av seilasene som testsett. Dette gjør at man kan teste alle modellene med likt testsett og dermed enklere kunne ta en avgjørelse på hvilke modeller som ser ut til å gjøre det best på disse dataene. Dette gjøres for at modellene ikke skal ha sett dataen på forhånd noe den heller ikke ville ha gjort i en reel situasjon, da seilasen den predikerer enda ikke har kjørt ruten.

5.2.3.2 Standard nevral nettverk

For standard nevral nettverk trenger man ikke å gjøre noen ytterlige endringer på dataen. Her holder det å lage to lister, hvor den ene er det som skal sendes inn til nettverket og den andre er det nettverket skal måle sine resultater opp mot. Dermed inneholder den første lista all data utenom siste punkt, og den andre all data utenom det første punktet. Dette fører til at nettverket blir trent på å predikere neste punkt basert på det punktet det får inn. Altså får nettverket som input $x[t]$ og sammenligner sin output med $x[t + 1]$.

5.2.3.3 RNN og LSTM

RNN og LSTM krever en tidsserie med data som input. Lengden på denne kan man selv konfigurere i nettverket. Som output har disse enten sekvenser, eller enkeltverdier. Det vil si at man selv kan konfigurere om nettverket skal returnere en tidsserie med all predikerte data fram til outputen, eller kun outputen. Dette betyr at dataen måtte forberedes på en mer avansert måte enn for standard nevrale nettverk. Man må fremdeles ha to lister, men disse må inneholde en liste med tidsserier på lengden av tidsserien antall verdier. Dataen konverteres til en liste over matriser med tidsserielengde gange fire verdier som er COG, SOG, longitude og latitude, hvor hver matrise inneholder alle punktene fram til det punktet den skal forsøke å predikere. Om man setter opp at nettverket skal returnere hele sekvensen fram til det predikerte punktet, må datalistene som nettverket skal sammenligne sin egen output med, være på samme format som inputen. Forskjellen er at den skal inneholde alle punktene i tidsserien til og med det punktet som skal predikeres.

5.3 Modeller

Det er valgt å se på standard nevrale nettverk, LSTM og RNN for å se om man kan predikere posisjonen til et fartøy fram i tid ved hjelp av AIS-data. Målet med denne sammenligningen er å se om man ser noen signifikant forskjell i prediksjonsevne for framtiden på disse typene.

Det som er felles for utprøvingen av alle modellene er at det er ønskelig å teste forskjellige variasjoner av antall lag i nettverket, antall noder, antall re-treninger (epoker) av nettverket, og forskjellige typer aktiveringsfunksjoner i outputlag. Alle modeller utprøvd får muligheten til å trene opp til 20 000 epoker. For en modellarkitektur vil det kun framvises resultater fra antall epoker som ga de beste resultatene.

For å begrense antall modeller å teste er det gjort noen avgjørelser som er felles for alle modellene. Disse avgjørelsene ble tatt ved en initial testperiode. Ut fra denne ble det kommet fram å sette batchstørrelsen til 32, optimizer til adam og loss-funksjon til MSE (Mean Squared Error). Det er også mer spesifikke endringer man har muligheten til å gjøre på de forskjellige typene nevrale nettverk. Disse vil bli presentert i de neste delkapitlene.

5.3.1 Standard nevralt nettverk

For standard nevrale nettverk har man muligheten til å endre hvilken aktiveringsfunksjon man benytter i de skjulte lagene. Her ble det etter en initial testperiode valgt å benytte ReLU (Rectified Linear Unit) i alle skjulte lag.

5.3.2 RNN og LSTM

De to neste nettverkstypene som er valgt å se på er RNN, og en avansert form av samme kategori: LSTM. Disse to har gjerne de samme parameterne man vil gjøre endringer på. Det som er viktig er at man i de skjulte lagene, altså lagene mellom input og output returnerer sekvenser. Det må også tas en avgjørelse på om man skal returnere sekvenser i siste lag. Dette gjør framtidsprediksjon enklere, da man får alle punktene i hele sekvensen for en prediksjon av neste AIS-data innslag. Dette gjør at man for framtidsprediksjon kan gi disse dataene til modellen for å predikere punktet etter, basert på forrige prediksjon. Dette kan gjøres så mange ganger man ønsker, alt etter hvor mange punkter fram i tid det er ønskelig å predikere. Det ble valgt for alle modellene utprøvd å benytte seg av at outputlaget returnerte sekvenser, da det ikke påvirket prediksjonsevnen og gjorde prediksjonen av flere punkt fram i tid enklere å implementere.

Det man også må se på er hvilken type nettverkslag man ønsker å ha i siste lag i modellen. Her kan det være at å benytte et standard nevral nettverkslag er bedre enn å bruke et lag av den type modell man tester. Da er det også viktig å teste hvilken aktiveringsfunksjon som ser ut til å være best for dette. En annen viktig innstilling for disse modellene er hvor store tidsseriene skal være. Det å ha lengre tidsserier fører til at man må ha mye mer data i minnet, det samme gjelder for hvor mange noder og hvor mange lag man har. Man kan møte en begrensning der, fordi om maskinen ikke har nok minne til å kunne ha all data for kjøringen i minnet samtidig, vil tiden det tar å trene nettverkene fort eskalere. Dette er en begrensning det må tas høyde for når man tester ut nye strukturer for nettverket, fordi man fort ikke har muligheten til å få trent nettverket nok ganger for å kunne avgjøre om den strukturen passer eller ikke for dataen.

For å redusere antall modeller som måtte utprøves ble det valgt å benytte en tidsserie på ti verdier for alle modellene. Dette var nok til å se at modellene fikk nok historisk data til å utføre prediksjonene, samtidig som at treningen av modellene ikke tok for lang tid, slik at man fikk utprøvd flest mulig varianter med tanke på antall lag og noder.

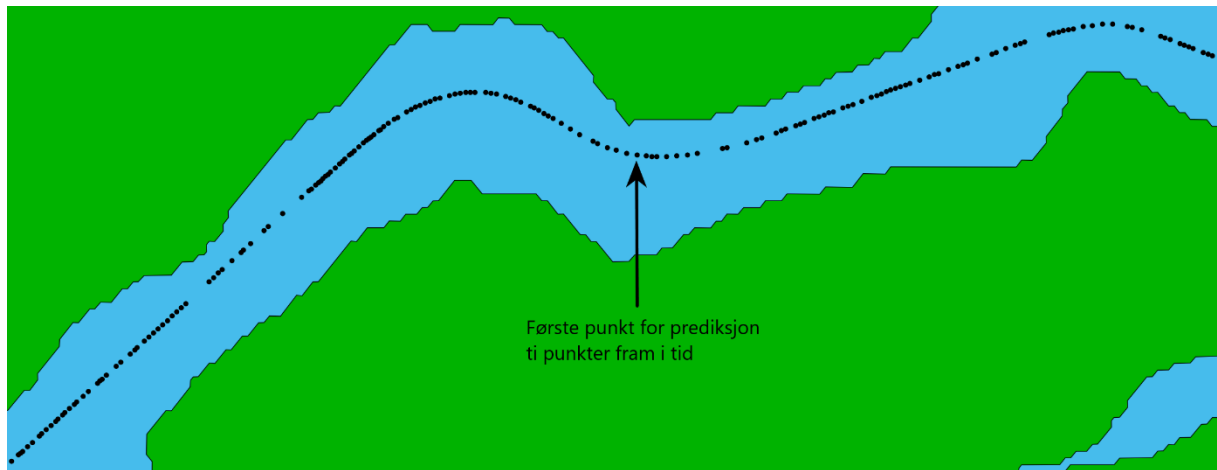
5.4 Eksperimenter

Eksperimentene skal utføres på det presenterte datasettet med de presenterte modelltypene. Som man ser fra Figur 5.3 har den valgte ruten to svinger etter hverandre i motsatt retning. I tillegg til at den har to rette partier. Den har også en sving helt i starten, men den kan det bli vanskelig for RNN og LSTM å teste på, da denne fort blir å komme før den første tidsserien er fylt opp. Dette er da avhengig av tidsseriestørrelsen.

Det er valgt å gjøre alle eksperimentene på en utvalgt seilas. Det er viktig at modellen ikke har trent på dataen det testes på, for å simulere sanntidsprediksjon. En annen grunn til at det er valgt å kun teste på en seilas, er at dette vil gjøre presentasjonen av predikert data opp mot faktiske data enklere å sammenligne. Da vil alle modellene det testes på, ha likt sammenligningsgrunnlag.

LSTM og RNN bruker veldig mye lengre tid å trene enn standard nevrale nettverk. Dette medfører at standard nevrale nettverk kan kjøre trening av større modeller over flere epoker på kortere tid. Dermed er det valgt at de standard nevrale nettverkene skal få muligheten til å trene på omtrentlig lik tid som LSTM og RNN, noe som gjør at disse kan ha større modeller

og flere epoker. Figur 5.3 viser en representasjon av dataene som skal benyttes som testsett for modellene.



Figur 5.3 Utvalgt rute for testing av modellene, ruten går fra høyre til venstre.

5.4.1 Prediksjon av ett punkt fram i tid

For å bedømme prediksjonsevnen til nettverkene er et første eksperiment på modellene å kjøre prediksjon ett punkt fram i tid langs hele ruten. Dette vil gi en indikasjon på prediksjonsevnen til modellen. Dersom man ser at disse prediksjonene avtar veldig fra ruten seilassen følger kan man anta at det å predikere lengre fram i tid vil bli vanskelig med dette grunnlaget. I tillegg kan man også se om deler av ruten er enklere eller vanskeligere å predikere enn andre deler av ruten, noe som kan gi en indikasjon på hvilke områder modellen er god på å predikere og hvilke områder modellen har vanskeligheter med å predikere.

5.4.2 Prediksjon opp til ti punkter fram i tid i sving

For å bedømme prediksjonsevne lengre fram i tid er det valgt å forsøke å predikere opp til ti punkter fram i tid i starten av første sving midt i ruten. For å godt kunne si om modellen kan predikere lengre fram i tid er det viktig å få prøvd dette ut i en sving, da dette er noe som ofte forekommer på ruter langs norskekysten. Det er valgt å ta samme punkt som skal være første punkt for alle modellene å predikere, dette for å kunne sammenligne modellene på en mest mulig rettferdig måte. Det første punktet er framhevet i Figur 5.3.

6 Resultater

Det vil bli framvist to resultater for hver arkitektur. Det er da henholdsvis ettstegsprediksjon langs hele ruten og ti-stegsprediksjon fra ett punkt på ruten. I figurene framvist vil de røde punktene representere de predikerte dataene og de sorte punktene representerer de faktiske dataene.

Alle eksperimentene er utført på samme rute, ti-stegsprediksjonene er i tillegg utført på samme punkter i denne ruten. Dette er vist fram i Figur 5.3.

Variasjonene av modellene og antall re-treninger (epoker) er framvist i Tabell 6.1, Tabell 6.2 og Tabell 6.3. Tabell 6.1 inneholder en oversikt over standard nevrale nettverk, Tabell 6.2 over RNN og Tabell 6.3 over LSTM. Index viser hvilken delfigur av resultatene modellen tilhører. Alle modeller framvist har fått mulighet til å trene på 20 000 epoker, men kun beste resultat fra modellene blir framvist.

Tabell 6.1 Modeller for standard nevrale nettverk

Index	Antall skjulte lag	Noder i hvert lag	Outputlag	Antall epoker
a	2	32	Sigmoid	500
b	2	32	Sigmoid	10 000
c	3	32	Sigmoid	10 000
d	3	50	Sigmoid	5 000
e	4	50	Sigmoid	5 000
f	5	50	Sigmoid	5 000
g	6	50	Sigmoid	5 000

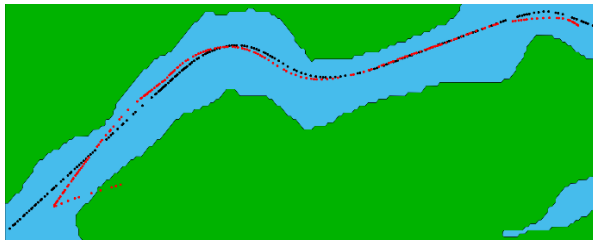
Tabell 6.2 Modeller for RNN

Index	Antall skjulte lag	Noder i hvert lag	Outputlag	Antall epoker
a	2	32	RNN	500
b	2	32	Sigmoid	500
c	2	32	Linear	500
d	2	32	Sigmoid	10 000
e	3	32	Sigmoid	15 000
d	3	50	Sigmoid	10 000

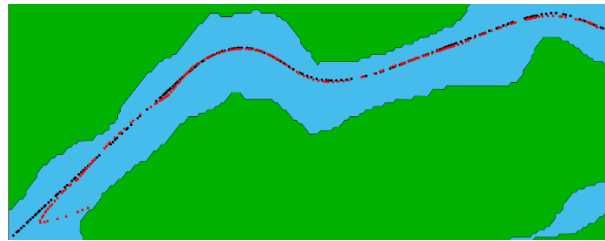
Tabell 6.3 Modeller for LSTM

Index	Antall skjulte lag	Noder i hvert lag	Outputlag	Antall epoker
a	2	32	LSTM	500
b	2	32	Sigmoid	500
c	2	32	Sigmoid	5 000
d	3	32	Sigmoid	20 000
e	2	50	Sigmoid	5 000
f	3	50	Sigmoid	15 000

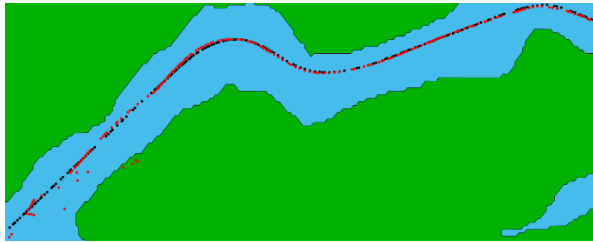
6.1 Standard nevralt nettverk



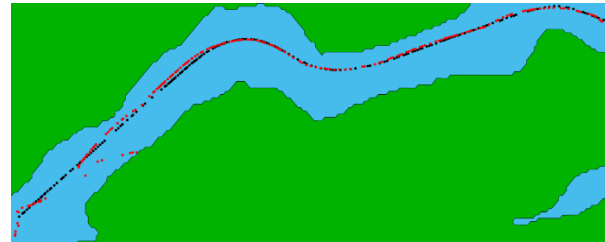
(a) to skjulte lag, 32 noder, 500 epoker



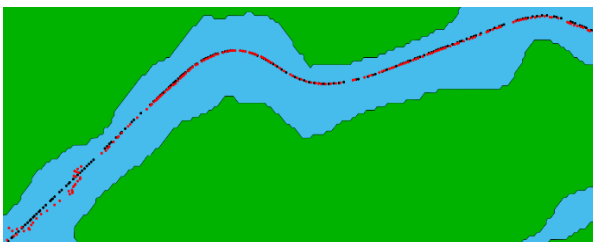
(b) to skjulte lag, 32 noder, 10000 epoker



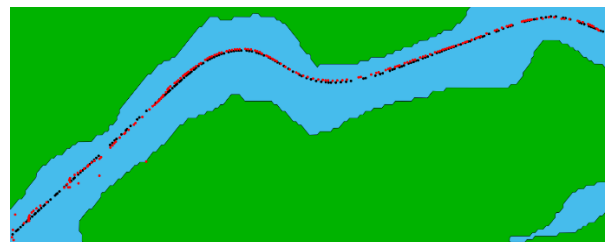
(c) tre skjulte lag, 32 noder, 10000 epoker



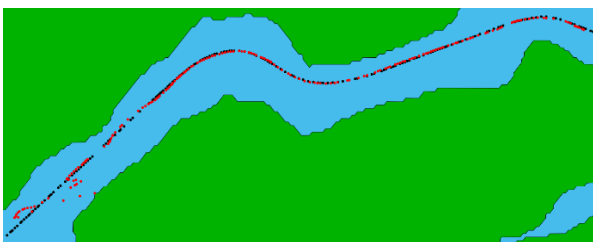
(d) tre skjulte lag, 50 noder, 5000 epoker



(e) fire skjulte lag, 50 noder, 5000 epoker



(f) fem skjulte lag, 50 noder, 5000 epoker



(g) seks skjulte lag, 50 noder, 5000 epoker

Figur 6.1 Ettstegsprediksjon for standard nevrale nettverk

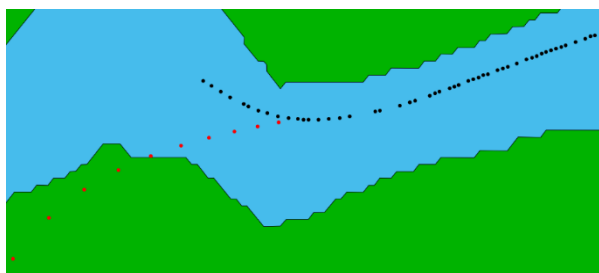
Figur 6.1 Viser ettstegsprediksjon på alle modeller utprøvd med standard nevrale nettverk.

Alle modeller følger ruten greit, men på begge nettverkene med to skjulte lag ser man at predikeres en sving på slutten som ikke er i den faktiske ruten. Begge modellene i Figur 6.1 a og b har begge to skjulte lag på 32 noder, men modellen i Figur 6.1.b er trent 10 000 epoker, mens modellen i Figur 6.1.a er trent 500 epoker. Som forventet er prediksjonen bedre på det nettverket som har trent 10 000 epoker, der det sees at de predikerte dataene er nærmere den

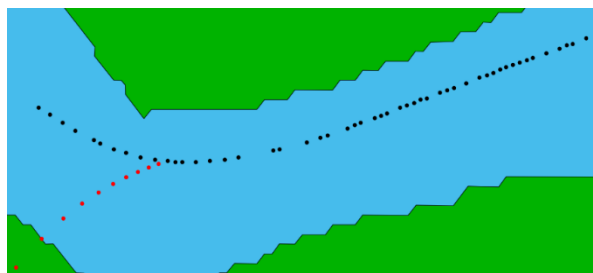
faktiske ruten, spesielt da i alle tre svingene. Selv om den som er trent 10 000 epoker treffer bedre på ruten, har begge det samme problemet på slutten av ruten, hvor det predikeres en skarp sving som ikke er med i den faktiske ruten.

Ved å øke mengden lag til tre skjulte lag først med fremdeles 32 noder i hvert lag som vist i Figur 6.1.c og 50 noder i hvert lag vist Figur 6.1.d, blir denne skarpe svingen fjernet fra de predikerte dataene. Begge har fremdeles problemer i det samme område, men nå er den skarpe svingen blitt redusert til støy og begge prediksjonene ender opp med å predikere punkter helt ned til slutten av ruten. Det ser ut til at ved å ha 50 noder i hvert lag får man redusert omfanget av denne støyen litt, da punktene holder seg noe nærmere den faktiske ruten enn ved 32 noder.

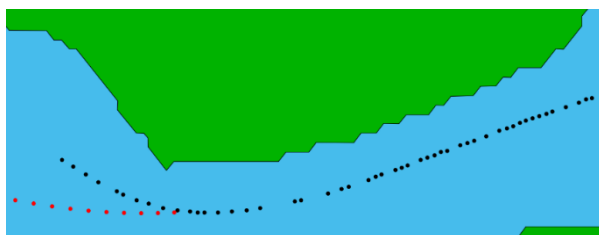
Figur 6.1.e viser resultatene for fire skjulte lag på 50 noder. Her blir støyen i prediksjonen på slutten av ruten redusert betraktelig og det observeres at den følger ruten på slutten ganske bra, men at det er noe feil i prediksjonene her når man sammenligner det med den faktiske ruten. Fra Figur 6.1.f ser man resultatene for en modell med fem skjulte lag på 50 noder. Prediksjonene følger her ruten på slutten bedre enn for fire skjulte lag, men har predikert enkelte punkter med mer støy enn det som ble observert for fire skjulte lag. Tilslutt er det utprøvd en modell med seks skjulte lag på 50 noder vist fram i Figur 6.1.g. Det observeres at prediksjonene på slutten får en hel del støy, mer enn det som ble observert for modellene med fire og fem skjulte lag.



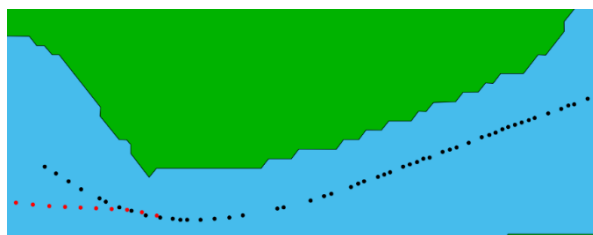
(a) to skjulte lag, 32 noder, 500 epoker



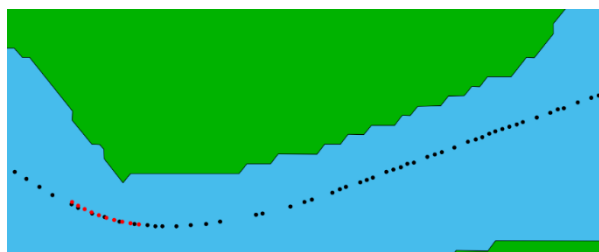
(b) to skjulte lag, 32 noder, 10000 epoker



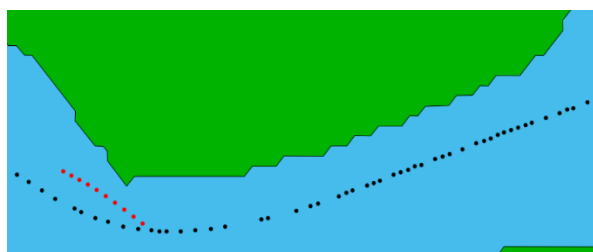
(c) tre skjulte lag, 32 noder, 10000 epoker



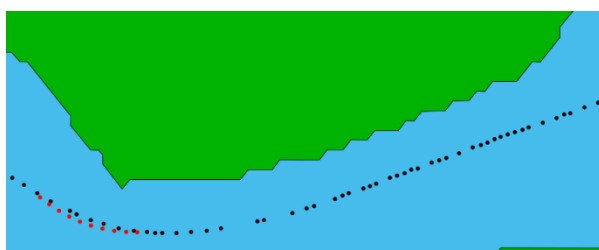
(d) tre skjulte lag, 50 noder, 5000 epoker



(e) fire skjulte lag, 50 noder, 5000 epoker



(f) fem skjulte lag, 50 noder, 5000 epoker



(g) seks skjulte lag, 50 noder, 5000 epoker

Figur 6.2 Ti-stegsprediksjon for standard nevrale nettverk

Figur 6.2 viser ti-stegsprediksjon på alle modeller utprøvd for standard nevrale nettverk. Her observeres det store forskjeller i nøyaktigheten til de forskjellige modellene. Fra Figur 6.2.a kan det observeres at prediksjonene til en modell med to skjulte lag på 32 noder trent 500 epoker bommer veldig på prediksjonene. Dette sees ved at svingen blir predikert til å være i feil retning og avstanden mellom hvert punkt er for stor, så store at den predikerte ruten ender opp på land langt unna forventet slutt på ruten. Det samme observeres i Figur 6.2.b som er et

nettverk av samme arkitektur, bare at det i stedet er trent 10 000 epoker. Mindre avstander mellom hvert predikert punkt i denne i forhold til den som er trent 500 epoker, men fremdeles ender prediksjonen på land i feil retning.

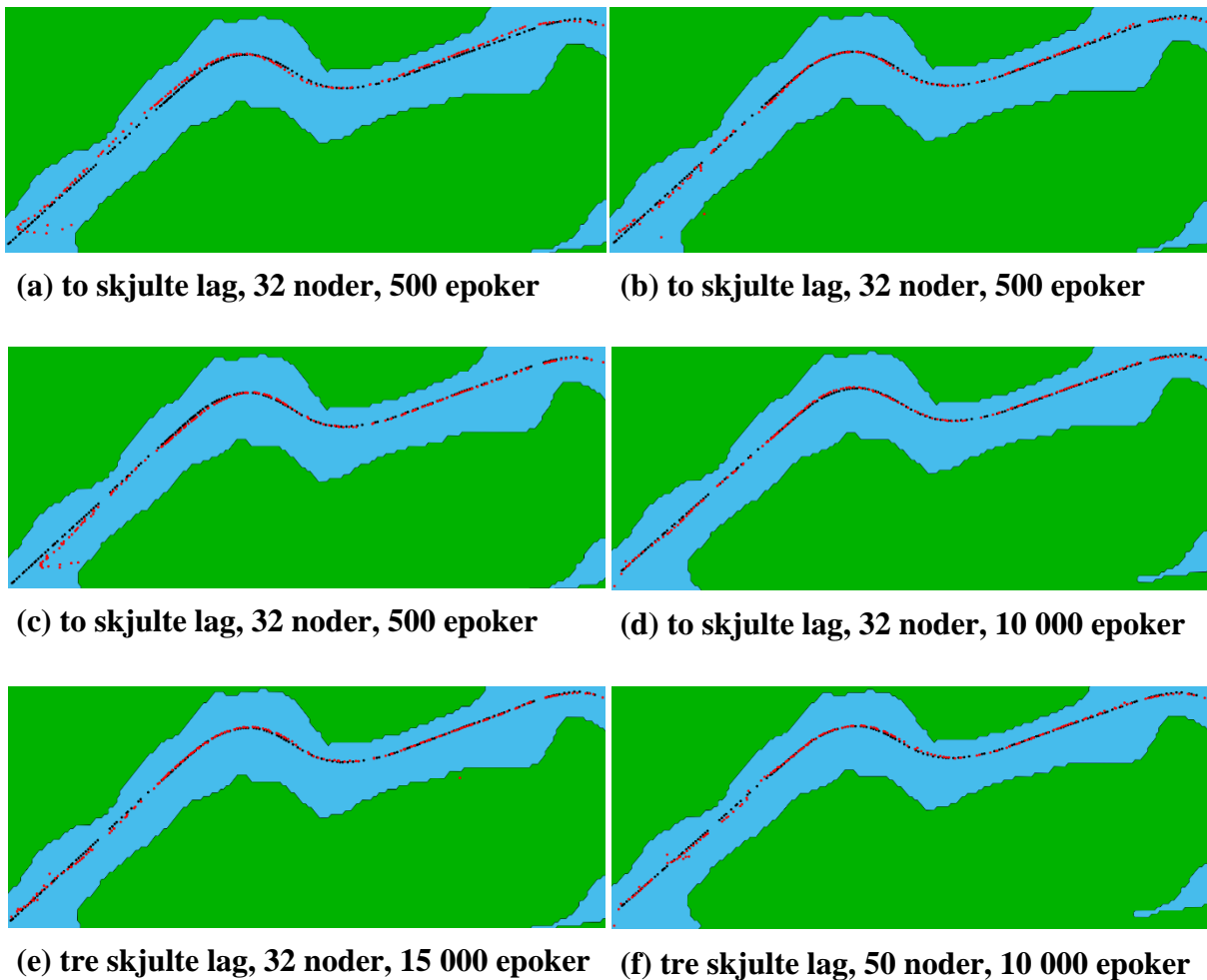
Ved å få inn enda et skjult lag i det nevrale nettverket, altså en modell med tre skjulte lag på 32 noder ser man fra Figur 6.2.c at prediksjonen ble mye bedre. Man kan se at banen på svingen fremdeles ikke blir predikert korrekt, men nå i stedet for en sving i feil retning, ender svingen med å bli predikert litt for slakt. Avstanden mellom hvert punkt er også predikert mye bedre, da det sees at i avstand er punktet så si predikert fram til hvor den faktiske ruten går.

Figur 6.2.d viser resultatet fra en modell på 50 noder med tre skjulte lag. Det observeres at modellen følger banen til den faktiske ruten på de tre første punktene så si perfekt, men etter det blir det predikert en liten knekk i svingen som fører til at de resterende punktene blir predikert litt feil. Modellen predikerte også avstanden mellom punktene veldig godt, og gjorde prediksjonen bedre enn modellen med 32 noder og likt antall skjulte lag.

Resultatene for en modell med fire skjulte lag på 50 noder er framvist i Figur 6.2.e. Denne modellen ser man predikerer banen til ruten helt perfekt, men de predikerte punktene er alt for tett. Dette medfører at den predikerte ruten ender opp med å kun komme fram til litt over halvveis i den faktiske ruten. Ved fem skjulte lag på 50 noder ser man fra Figur 6.2.f at avstandene blir bedre enn det var ved fire skjulte lag, men banen blir predikert feil og man ser at modellen med tre skjulte lag predikerer bedre enn hva denne gjorde.

Figur 6.2.g viser resultatet for seks skjulte lag på 50 noder, her observeres det at banen er predikert veldig bra litt dypere sving enn i den faktiske ruten. Dermed ikke like godt predikert bane som ved fire skjulte lag, men avstandene er predikert mye bedre. Dette fører til at den predikerte banen ender opp til det åttende punktet i den faktiske ruten.

6.2 RNN

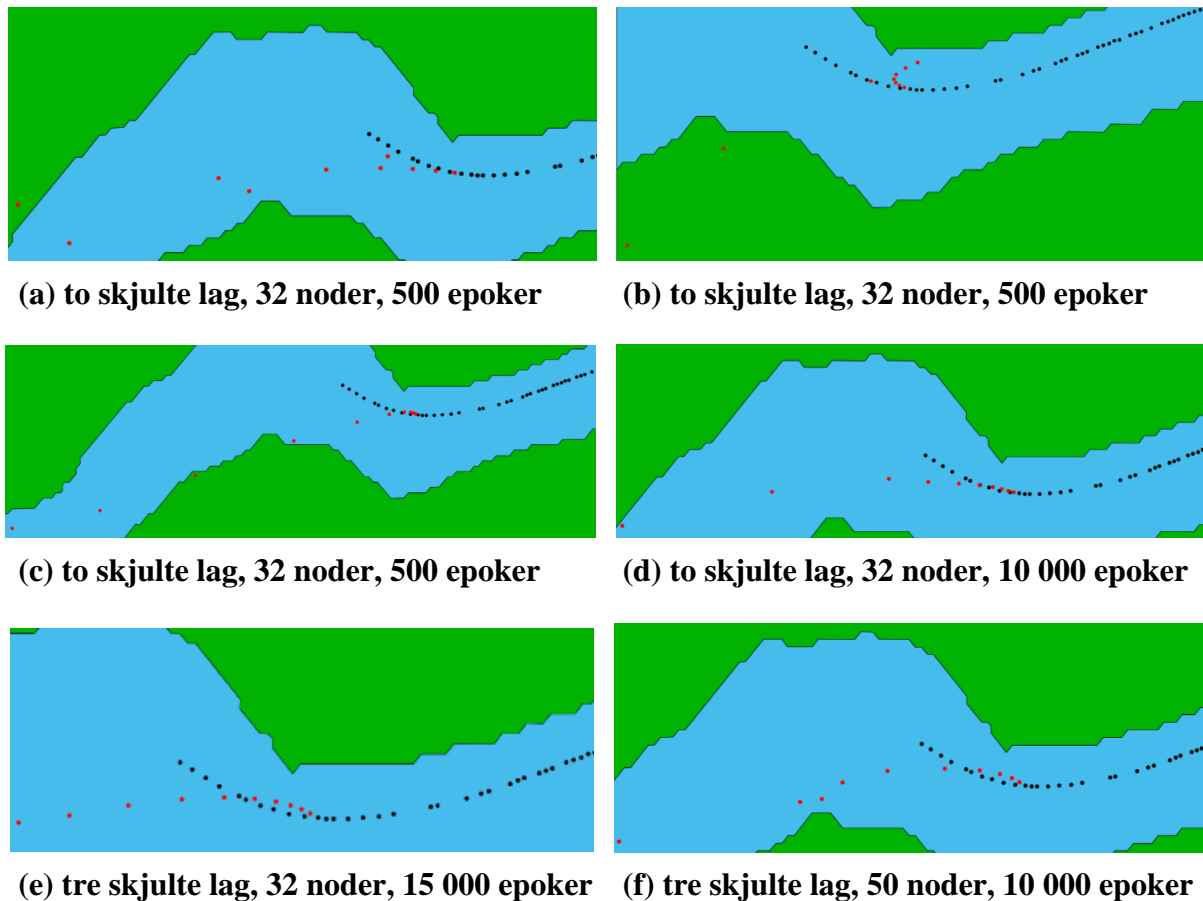


Figur 6.3 Ettstegsprediksjon for RNN

Figur 6.3 viser ettstegsprediksjoner for alle utprøvde RNN modeller. Figur 6.3 a, b og c har alle to skjulte lag på 32 noder trent 500 epoker, forskjellen er outputfunksjonen. Figur 6.3.a benytter RNN i outputlag, Figur 6.3.b benytter sigmoid i outputlag og Figur 6.3.c benytter linear i outputlag. Det observeres at modellene med både linear og RNN i outputlag ender med å predikere en sving som ikke er tilstede i de faktiske dataene, mens modellen som bruker sigmoid i outputlag predikerer mer riktig i dette område. Denne predikerer to punkter langt utenfor ruten på slutten, men ellers er punktene predikert nærmere den faktiske ruten. Det ble dermed avgjort å benytte sigmoid i de resterende modellene.

Modellen som predikerer dette området best er den med to skjulte lag på 32 noder som er trent 10 000 epoker framvist i Figur 6.3.d. Denne følger ruten så si perfekt, med da et unntak på slutten hvor den predikerer det siste punktet for langt fram i forhold til den faktiske ruten.

Det kan observeres fra Figur 6.3.e at modellen med tre skjulte lag på 32 noder også predikerer ruten godt, men at det i starten av ruten blir predikert ett punkt langt fra ruten som ender opp på land. Modellen med tre skjulte lag på 50 noder kan man se fra Figur 6.3.f ikke får dette støypunktet, men predikerer mer støy på slutten av ruten enn hva de to foregående modellene gjorde.

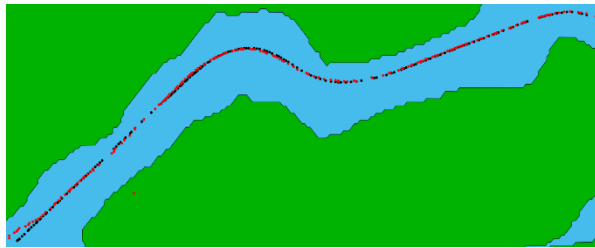


Figur 6.4 Ti-stegsprediksjon for RNN

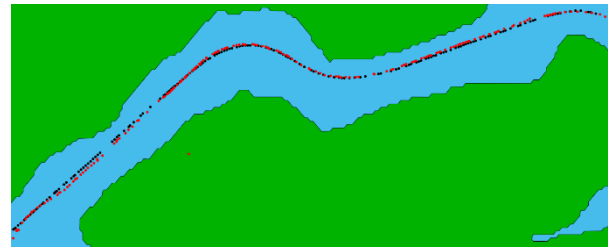
Figur 6.4 viser ti-stegsprediksjon for alle utprøvde RNN. Rekkefølgen på disse modellene er den samme som for Figur 6.3. Noe som betyr at Figur 6.4 a, b og c henholdsvis viser resultater for RNN i outputlag, sigmoid i outputlag og linear i outputlag. Det man kan observere er at hvor ettstegsprediksjonen gjorde det generelt bedre og mer stabilt for RNN enn standard nevral nettverk kan det motsatte sies om ti-stegsprediksjonen. Alle modellene endte opp med å predikere de siste punktene langt unna den faktiske ruten. Modellen med to skjulte lag på 32 noder og sigmoid i outputlag som ble trent 500 epoker ser man fra Figur

6.4.b predikerte rett for første punkt, deretter ble prediksjonene for tette og snudde retning. Tilsvarende modell med RNN i outputlag vist fram i Figur 6.4.a predikerte de to første punktene korrekt, deretter ble tredje punkt predikert med litt for liten sving i forhold til faktisk rute. Etter det tredje punktet ser man at de predikerte punktene får for stor avstand mellom seg i forhold til faktisk rute noe som eskalerer for hvert predikert punkt. Modellen med linear i outputlag vist fram i Figur 6.4.c predikerte de fire første punktene for tett, deretter blir avstandene som predikeres for lange og det tiende punktet ender opp på slutten av testområde. I Figur 6.4 d, e og f har alle modellene benyttet sigmoid i outputlag. Alle disse modellene har predikert de fire første punktene nær den faktiske ruten, men det er litt variasjoner i hvordan disse har predikert banen og de seks resterende punktene. Modellen som predikerte banen fram til det fjerde punktet best var modellen med to skjulte lag på 32 noder trent 10 000 epoker som er framvist i Figur 6.4.d. Denne predikerer litt for tett i starten noe som medfører at femte predikerte punkt ender på det fjerde punktet i den faktiske ruten. Deretter kan man se at det sjette predikerte punktet ender opp ganske nært punktet det skal predikere, men at banen er predikert med litt for lite svingning. Videre blir de fire siste punktene predikert for langt fram, og denne avstanden eskalerer med hvert predikert punkt. Modellen som predikerte avstanden til det tiende punktet best var modellen med tre skjulte lag på 32 noder vist i Figur 6.4.e, fremdeles predikerte denne for langt fram og som de andre modellene for langt ned i forhold til faktisk rute. Figur 6.4.f viser resultatene for modellen med tre skjulte lag på 50 noder. Det kan observeres at denne gjorde det dårligere enn de to foregående modellene, da den hverken predikerte banen like godt som modellen i Figur 6.4.d, eller predikerte avstanden like godt som modellen Figur 6.4.e.

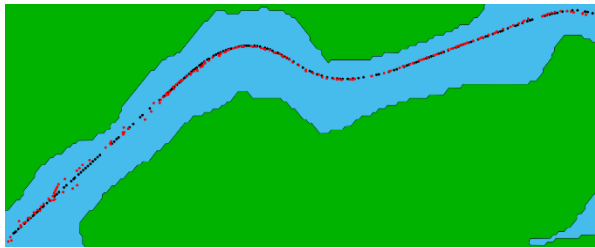
6.3 LSTM



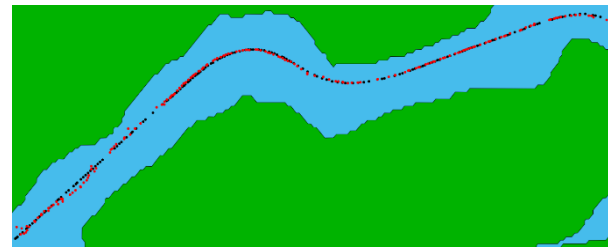
(a) to skjulte lag, 32 noder, LSTM



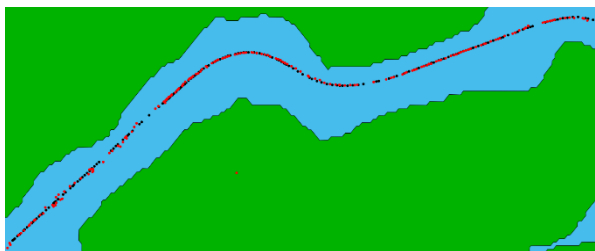
(b) to skjulte lag, 32 noder, sigmoid



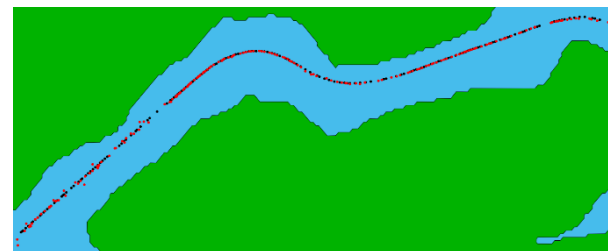
(c) to skjulte lag, 32 noder, 5 000 epoker



(d) tre skjulte lag, 32 noder, 20 000 epoker



(e) to skjulte lag, 50 noder, 5 000 epoker



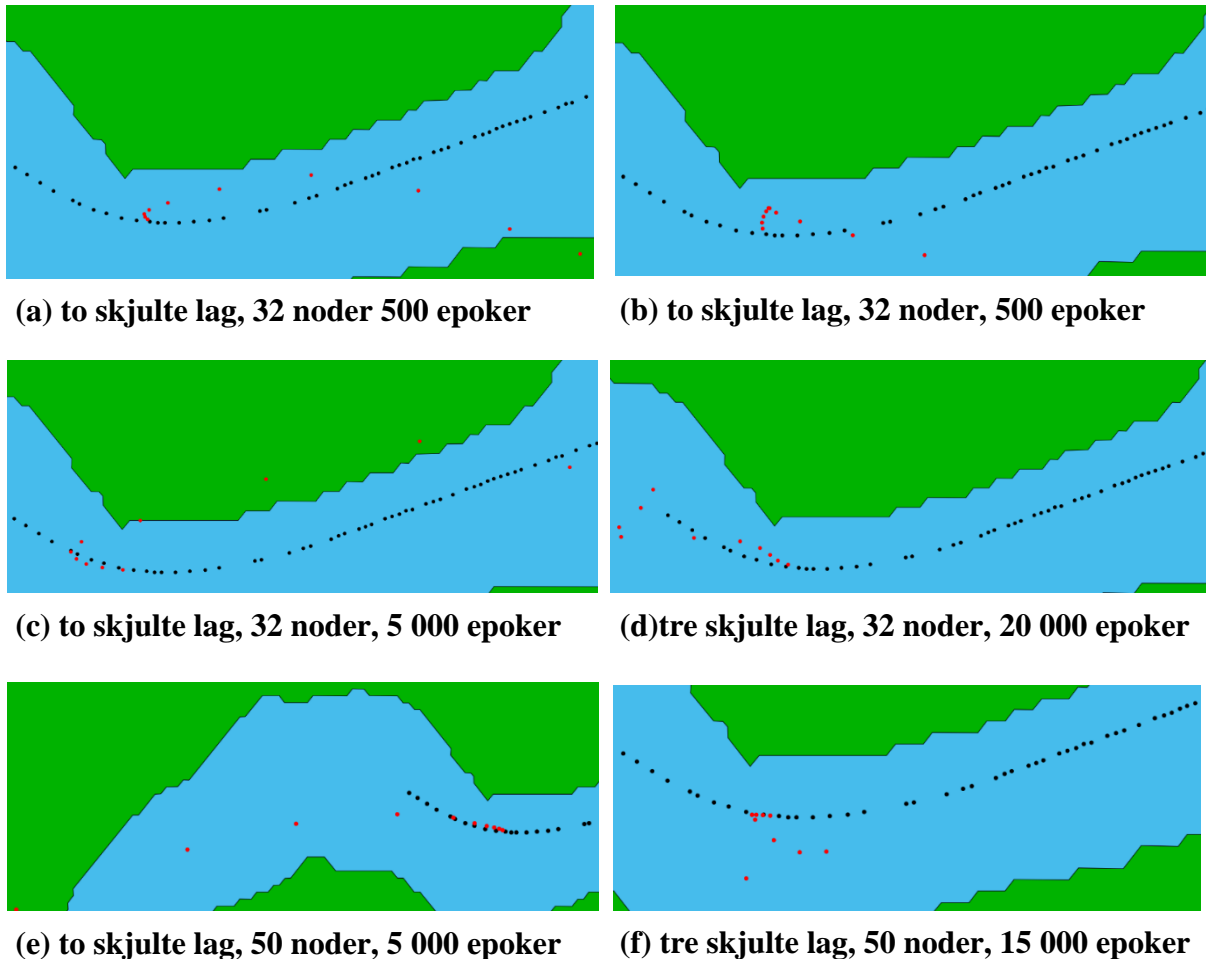
(f) tre skjulte lag, 50 noder, 15 000 epoker

Figur 6.5 Ettstegsprediksjon for LSTM

Figur 6.5 viser ettstegsprediksjon for alle utprøvde LSTM- modeller. Figur 6.5 a og b viser begge resultater for et LSTM med to skjulte lag på 32 noder trent 500 epoker. Forskjellen er at modellen for Figur 6.5.a har LSTM i outputlag mens modellen for Figur 6.5.b har sigmoid i outputlag. Fra disse to ser man at begge modellene ender opp med å predikere ett støypunkt på land mot slutten av ruten, men at modellen med sigmoid i outputlag predikerer punktene på slutten nærmere den faktiske ruten enn den med LSTM i outputlag.

Fra Figur 6.5 kan man observere at alle modellene predikerer ruten nær den faktiske ruten, og at disse resultatene er mer stabilt gode enn både for RNN og standard nevrale nettverk. Alle modeller utenom modellen for Figur 6.5.a har benyttet sigmoid i outputlag. Modellen med to skjulte lag på 32 noder trent 5 000 epoker kan man se at predikerte mer støy rundt den faktiske ruten enn hva resten av rutene gjorde. Fra Figur 6.5.e kan det observeres at det er

predikert ett punkt på land mot slutten av ruten, som også var tilfellet for modellene i Figur 6.5 a og b. Modellen som predikerte ruten best var den med tre skjulte lag på 50 noder som er vist fram i Figur 6.5.f. Denne hadde også litt støy rundt den faktiske ruten på slutten, men predikerte ingen punkter på land og hadde minst støy på slutten.



Figur 6.6 Ti-stegsprediksjon for LSTM

Figur 6.6 viser ti-stegsprediksjon for alle utprøvde LSTM, modellen i Figur 6.6.a har LSTM i outputlag, resten har sigmoid. Som for RNN kan man observere at resultatene for ti-stegsprediksjon er mer varierende enn de var for de standard nevrale nettverkene. Det kan observeres at alle modellene sett bort fra modellen med tre skjulte lag på 32 noder framvist i Figur 6.6.d endte opp med å predikere det tiende punktet langt unna den faktiske ruten. Denne modellen predikerte banen fram til det femte punktet ganske godt, men etter dette ble banen predikert feil. Tilslutt endte det tiende punktet opp nærme det tiende punktet for den faktiske ruten.

Modellene framvist i Figur 6.6 a, b, c og f endte opp med å snu retning i løpet av prediksjonen. Både den i Figur 6.6.a og b snudde retning etter de første to punktene, den i Figur 6.6.f predikerte de fire første punktene mellom første og andre punkt i den faktiske ruten for så å gå i feil retning. Modellen i Figur 6.6.c derimot predikerte de seks første punktene korrekt, før den predikerte en for brå sving og snudde retning. Modellen i Figur 6.6.e fulgte banen til den faktiske ruten best, men begynte prediksjonen for tett noe som medførte at det sjette predikerte punktet endte mellom det fjerde og femte faktiske punktet. Videre eskalerte avstanden mellom de predikerte punktene. Noe som medførte at det tiende punktet endte langt unna det faktiske tiende punktet og på land.

7 Diskusjon

I kapittel 6 er det prøvd ut tre forskjellige hovedtyper av nevrale nettverk for å teste ut hvor godt man kan predikere posisjonen til et fartøy med AIS-data. Disse hovedtypene er standard nevrale nettverk, RNN og LSTM.

Da denne avhandlingen ble startet var tanken at det å predikere gjennom en sving skulle gå lett for de fleste modeller som hadde litt trening, og at man kunne gå over på større utfordringer som to svinger og se på et sted hvor rutene splittet og samlet seg igjen. Dette ble ikke utfallet da det å predikere fram i tid på AIS-data viste seg å være en større utfordring enn først antatt. Selv om dette ble en større utfordring enn først antatt, har mange av modellene vist lovende resultater på denne svingen.

Det første eksperimentet som ble utført for modellene var å se prediksjonsevnen når man forsøker å predikere ett punkt fram i tid. De fleste modellene klarte dette godt. Et unntak her for mange av modellene, og da spesielt for de standard nevrale nettverkene, var at på slutten av ruten endte noen av modellene opp med å predikere en del feil. Mange av modellene predikerte en sving som ikke var der, mens andre predikerte en hel del støy i dette området. Sett bort fra dette området har ettstegsprediksjonene vært gode. Det registreres noen avvik, men tilsynelatende veldig gode prediksjoner.

Det andre eksperimentet som ble utført for modellene var prediksjon av ti punkter fram i tid, her var det mer varierende resultater. Alle modellene ble testet på samme sted på samme rute for å best mulig kunne sammenlignes. En usikkerhet som følge av dette er at modellen kanskje predikerte akkurat dette punktet dårlig, men om man hadde valgt punktet bak eller foran, hadde prediksjonen blitt bedre. En grundigere utprøving kunne man ha utført, og da kanskje på flere steder i ruten for å se at det ikke bare var tilfeldig at akkurat dette stedet nettverket klarte eller ikke klarte å predikere. Gitt tilgjengelig tid for denne forskningen ville denne framgangsmåten ført til at man ikke hadde fått testet like mange variasjoner av modellene.

Fra Figur 6.5 ser man at alle ettstegsprediksjonene til LSTM-modellene utfører prediksjonen med lite støy og alle modellene unngår å predikere en sving på slutten som ikke er der. RNN-modellene predikerer også ettstegsprediksjonene stabilt godt sett fra Figur 6.3 hvor to av modellene endte opp med å predikere en sving som ikke var der, men dette var da de modellene som hadde kun trent 500 epoker. Sett bort fra disse to ble det predikert lite støy for RNN-modellene. Fra Figur 6.1 observeres det at de standard nevrale nettverkene også hadde

to modeller med en sving på slutten hvor en av disse hadde trent 10 000 epoker. I tillegg kan det observeres at de standard nevrale nettverkene med tre skjulte lag predikerte mye støy i det samme område. Denne støyen ble redusert ved fire og fem skjulte lag, men kom tilbake igjen ved seks skjulte lag.

Med resultatene fra ettstegsprediksjonen var det forventet at det samme gjaldt for ti-stegsprediksjonen, at man her også så at LSTM og RNN predikerte bedre enn hva de standard nevrale nettverkene gjorde. Dette ble ikke tilfellet, der man ser fra Figur 6.4 at alle RNN-modellene endte opp med å predikere siste punkt i ti-stegsprediksjonen langt fra det tiende punktet i den faktiske ruten. Det kan observeres at tre av modellene predikerte de fire første punktene korrekt og den beste av disse var modellen med to skjulte lag på 32 noder som predikerte opp til det sjette punktet korrekt. Deretter predikerte den de resterende punktene med for stor avstand.

Fra Figur 6.6 kan det observeres at LSTM-modellene også hadde varierende resultater for ti-stegsprediksjonen. Hvor man ser at alle modellene sett bort fra den med tre skjulte lag på 32 noder endte opp med å predikere det tiende punktet langt unna det faktiske tiende punktet. Modellen med tre skjulte lag på 32 noder traff det tiende punktet godt, men banen fram til det tiende punktet var ikke korrekt. Fire av modellene endte opp på et tidspunkt i prediksjonen å snu og gå motsatt vei av hva ruten gjorde, mens den siste endte opp med å predikere for langt fram i tid. To av modellene predikerte banen og avstanden fram til det sjette punktet godt.

Modellene for standard nevrale nettverk, som gjorde ettstegsprediksjonen dårligst av de utprøvde typene nevrale nettverk kan man se fra Figur 6.2 utførte ti-stegsprediksjonen mer stabilt godt og hadde modeller som gjorde det bedre enn de beste RNN-modellene og LSTM-modellene. Alle modellene sett bort fra den med to skjulte lag på 32 noder, som både med 500 epoker trening og 10 000 epoker trening predikerte svingen i feil retning, predikerte banen og avstanden til det tiende punktet ganske godt. Modellen med fire skjulte lag på 50 noder traff best på banen, men predikerte ruten for kort og endte da med å predikere det tiende punktet på det sjette punktet i den faktiske ruten. Den som predikerte best med tanke på avstanden var modellen med seks skjulte lag på 50 noder, denne predikerte det tiende punktet på det åttende punktet i den faktiske ruten og i tillegg predikerte den banen godt hvor den hadde en marginalt dypere svingning enn den faktiske ruten.

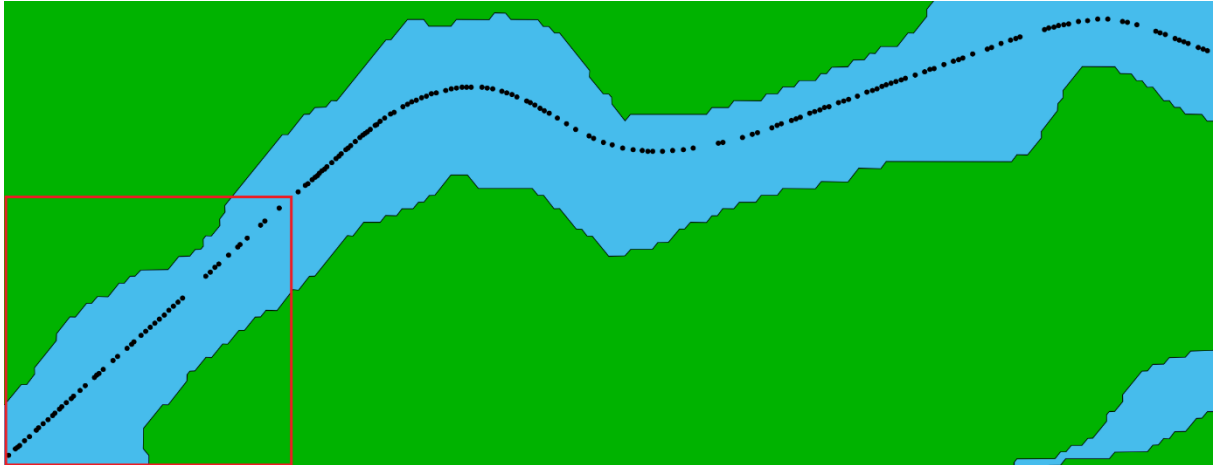
På ti-stegsprediksjon ser man at det beste resultatet var fra et standard nevralt nettverk. Både RNN modellene og LSTM modellene hadde flere modeller som traff ganske godt på de første fire til seks punktene, men det fikk alvorlige konsekvenser for prediksjonen når

svingningskurven begynte å bli feil i forhold til den faktiske ruten. Disse konsekvensene var at mange av modellene ofte fikk en brå sving etter en feil prediksjon, dette ofte i kombinasjon med eskalerende avstand mellom punktene.

LSTM og RNN er nettverkstyper som er spesialisert på å håndtere tidsseriedata, da de har muligheten for å huske aktiveringer og tar høyde for tidligere case når de beregner outputen til nåværende case. AIS-dataen i denne forskningen er konvertert til tidsseriedata ved at den er gruppert på seilaser og sortert på dato, noe som skulle betydd at LSTM og RNN ville gjøre det godt på denne strukturen. Samtidig kan dette føre til at om modellen predikerer en feil på et punkt så kan det se ut for modellen at det skal utføres en sving, eller at avstanden mellom punktene skal øke. De standard nevrale nettverkene derimot som ser på hvert punkt som uavhengige. Dermed vil det i det samme tilfelle kun se på det nyeste punktet når det skal predikere neste punkt.

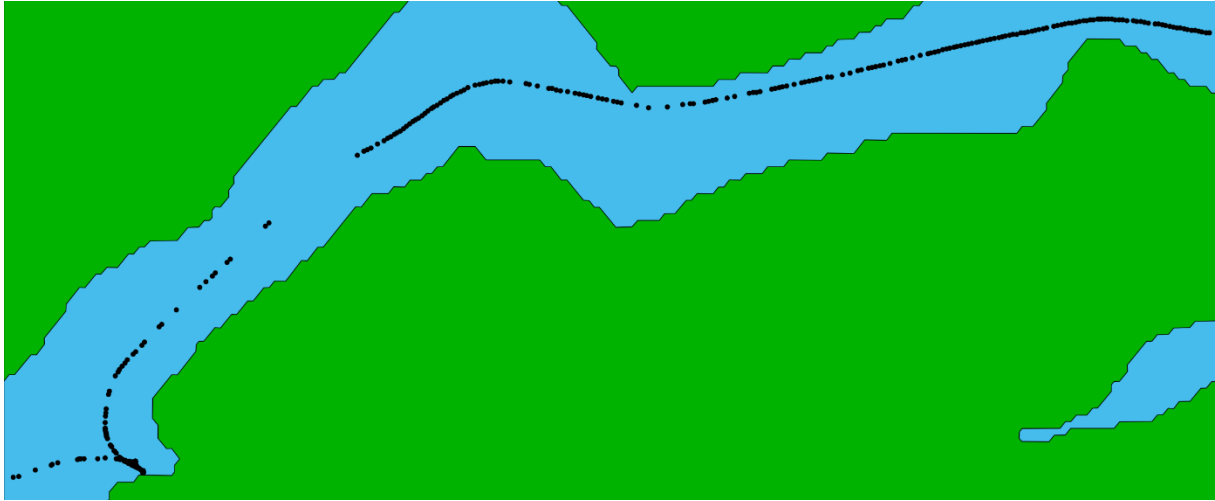
Det må også tas i betraktning at standard nevrale nettverk er veldig mye raskere å trene, derfor var det mulig å trene dypere modeller. Dette betyr at dersom man hadde gitt de rekurrente nevrale nettverkene mer tid til å trene dypere modeller, eller til å trene de utprøvde modellene over flere epoker så hadde kanskje disse gjort det bedre. Samtidig har man allerede tatt et veldig avgrenset geografisk område, og om dette skulle blitt implementert for større deler av norskekysten er det viktig at nettverkene har en realistisk treningstid for å kunne anvendes på både større og flere geografiske områder.

Ettstegsprediksjonene så ut til å gjøre det godt på hele området, men mange av modellene gjorde en del feil på siste del av ruta, hvor det ikke var noen svinger eller andre tilsynelatende problematiske data å predikere. Figur 7.1 viser den biten av ruten hvor det var en del modeller som hadde problemer med å predikere den rette linjen. Både standard nevrale nettverk og RNN hadde modeller som predikerte en sving som ikke var der i dette området. I tillegg predikerte de standard nevrale nettverkene en del støy i dette området.



Figur 7.1 Testdatasett, utfordrende del markert

Figur 7.2 viser en av rutene som ble brukt til trening av det nevrale nettverket. Som man ser så valgte akkurat denne seilasen å kjøre innom land på slutten av testområdet. Dette medfører to utfordringer for de nevrale nettverkene. Det første er at det blir en splitt i mulige baner et fartøy kan ta, enten fortsette rett fram eller ta en liten sving. Det andre er at når båten svinger tilbake til ruten igjen ender den opp med å være lengre ned og fram enn den ville vært om den fulgt ruten til testdatasettet. Dette kan forklare hvorfor mange modeller endte opp med å predikere dette tilsynelatende enkle området mye dårligere enn på resten av datasettet på ettstegsprediksjonene. Dette kan også forklare hvorfor LSTM og RNN ser ut til å ha gjort det bedre enn standard nevrale nettverk på denne biten, fordi disse tar hensyn til flere punkter tilbake i tid når den predikerer neste punkt. Dette kan medføre at nettverkene enklere ser om den skal gå fortsette videre, eller svinge ned. Splitter og sammenknyttinger av ruter er noe som ofte forekommer på norske farvann og er viktig at prediksjonsmodellene håndterer godt.

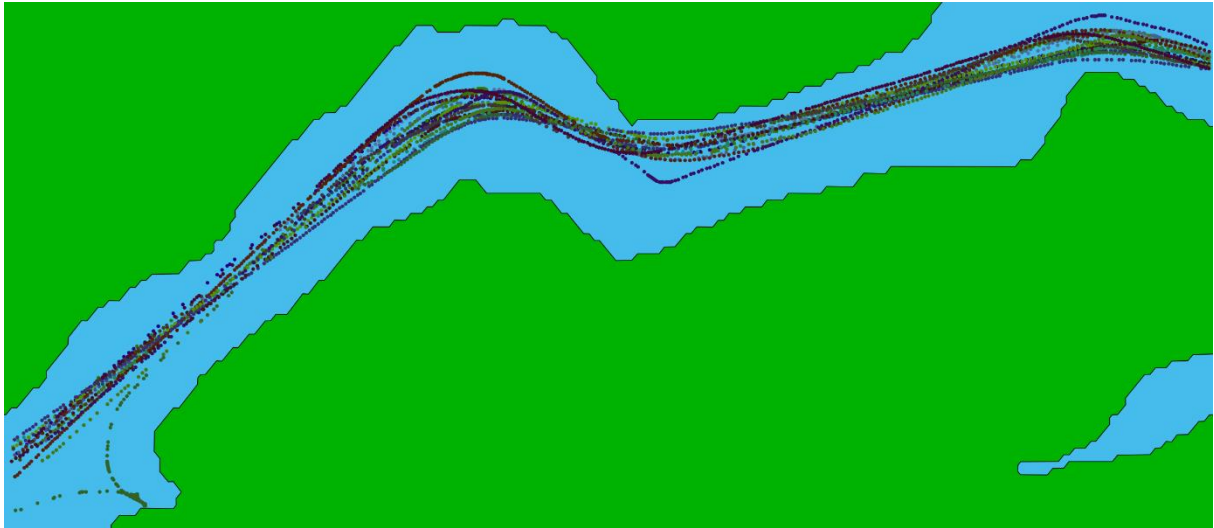


Figur 7.2 Seilas som svingte innom land på slutten av ruten, og har svak sving i område med ti-stegsprediksjon.

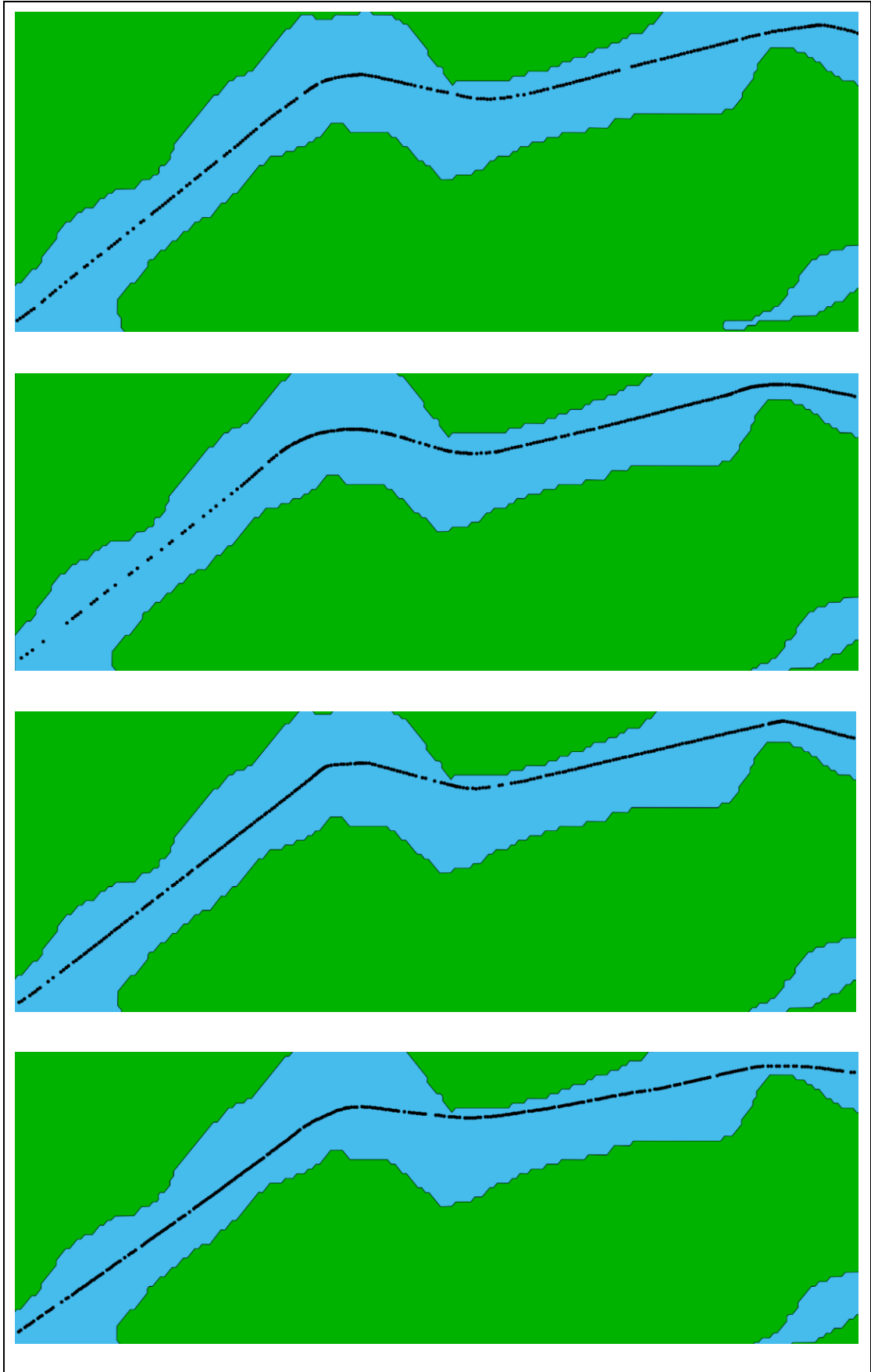
Figur 7.3 viser alle treningsdataene som er brukt på modellene. Fra Figur 7.3 ser man at fartøyene varierer veldig hvordan de går inn i den svingen som er forsøkt predikert og hvordan dem tar den. Noen ruter ser man går veldig mye lengre ned, noe som fører til en skarpere sving, mens andre går veldig høyt opp som fører til en slakere sving. For å bedre kunne visualisere dette må man se på en og en rute. Det er valgt å vise fram de som trolig har gjort utførelsen av prediksjonen mest vanskelig.

Fra Figur 7.3 ser man i tillegg at det er veldig stor spredning på hvor seilasene går i testdataen som de nevrале nettverkene har trent på. Dette i seg selv er med på å forklare at det har vært mye problemer med å predikere i området. De nevrале nettverkene forsøker å representere én

rute, men når treningsdataen er så spreidd blir det vanskelig å lære hvor fartøyet kommer til å være fram i tid.

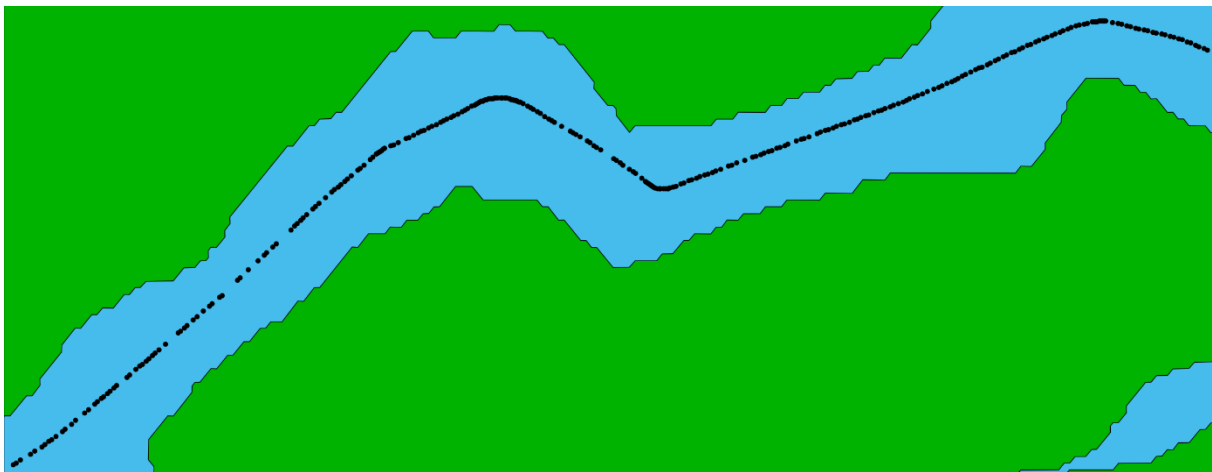


Figur 7.3 All treningsdata til modellene

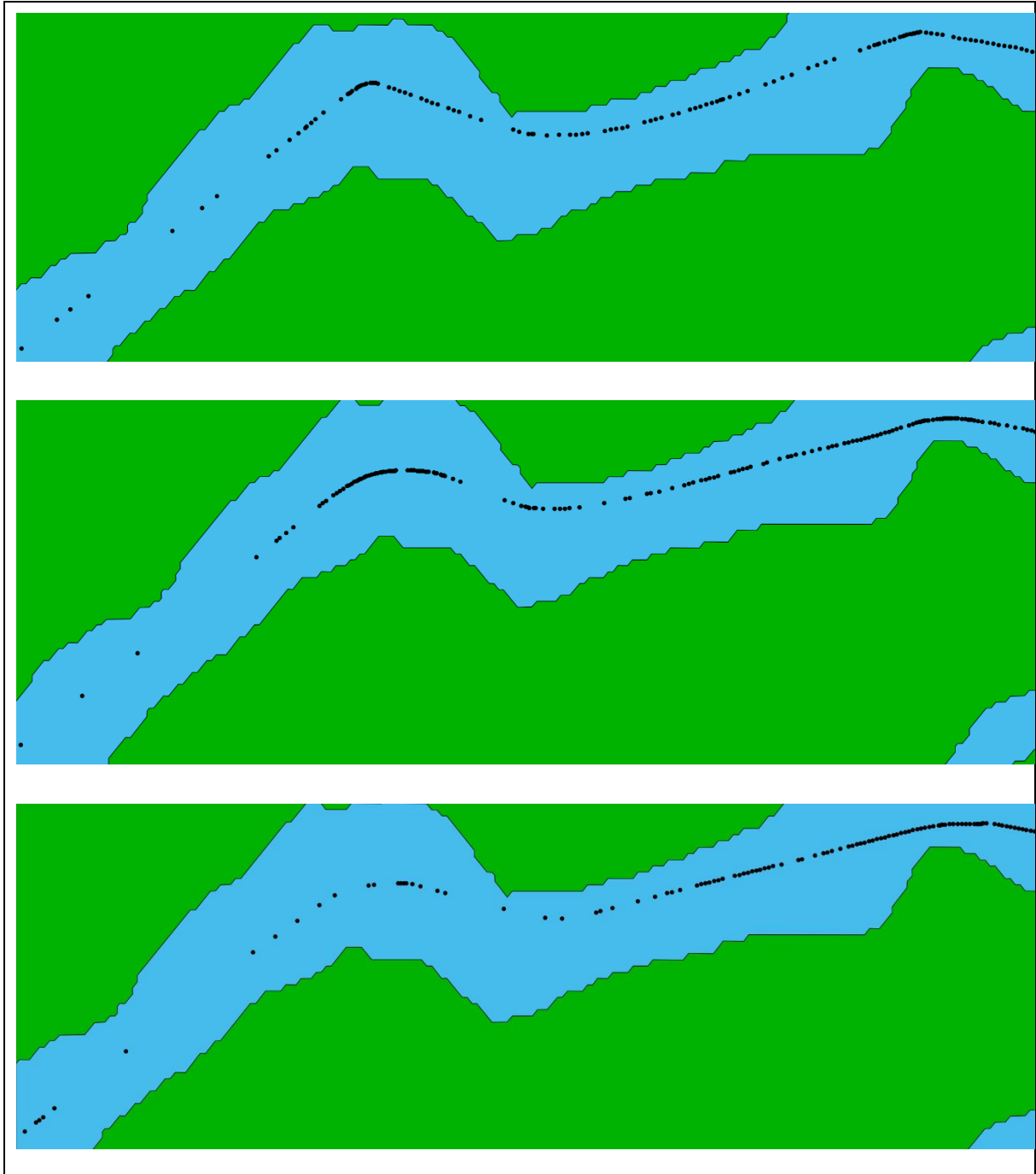


Figur 7.4 Seilaser med tette data og svak sving i område hvor ti-stegsprediksjon er utført

Fra Figur 7.4 ser man fire seilaser som har svak sving i det predikerte området, det samme gjelder seilassen i Figur 7.2. Figur 7.5 viser en seilas hvor svingen er langt nede og veldig brå. Dette viser hvor forskjellig seilasene utfører denne svingen, både med svingningsgrad og hvor de ligger i ruten. Dette forklarer også hvorfor modellene hadde problemer med å predikere denne svingen. Da nettverkene har sett så mange forskjellige måter å gjennomføre denne svingen på vil det bli vanskelig å kunne predikere hvordan akkurat den ene seilassen skal gå gjennom den. Det man også kan observere er at seilasene i Figur 7.2, Figur 7.4 og Figur 7.5 har mye tettere data i seilassen enn hva testruten hadde. Dette kan forklare hvorfor mange modeller har predikert for tett i ti-stegsprediksjonen.



Figur 7.5 Seilas med tette data og brå sving i predikert område



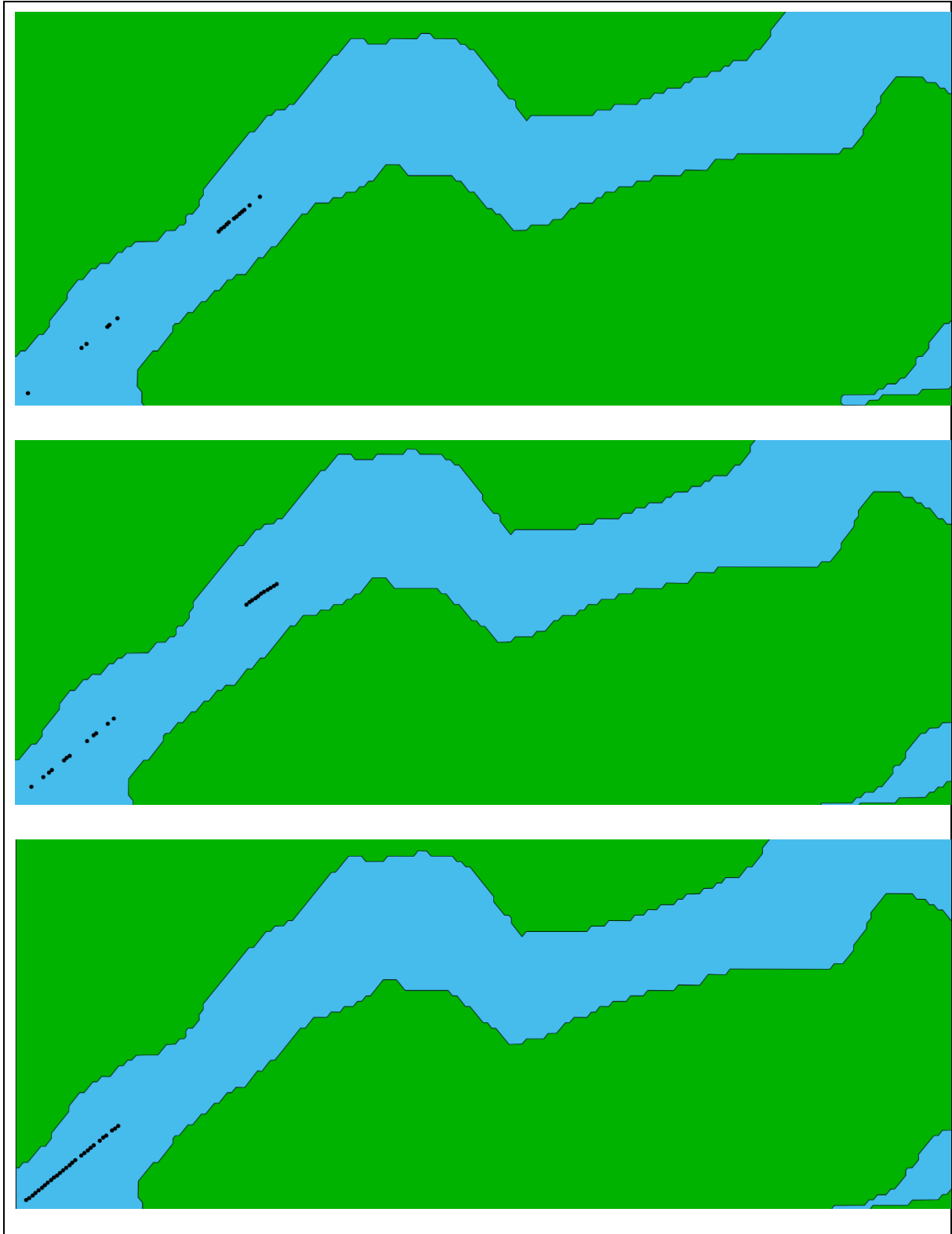
Figur 7.6 Seilaser med spredde data og hull i predikert område

Figur 7.6 viser seilaser fra treningsdataen som inneholder få punkter og har hull i svingen som det ble forsøkt å predikere på. I tillegg ser man at nederste seilas kun har ett punkt i det område det ble forsøkt predikere på.

Figur 7.7 viser tre seilaser i treningsdataen som ikke inneholder noe data før på slutten av ruten. Man kan også se at de to første seilasene inneholder store hull i ruten. Det at seilassen

ikke inneholder noe data gir ikke mer problemer enn det at det er mindre treningsdata for det manglende område.

Slike hull og spredte data kan komme av at AIS-data har blitt tapt under sending, eller at transponderne om bord i fartøyene ikke har sendt data hyppig nok. Uansett grunn kan dette føre til at nettverkene ikke lærer område godt nok og lærer at fartøyene varierer veldig i fart i området med at neste punkt plutselig er langt framfor forrige. Dette er et stort problem for modellene. I alle seilasene presentert ser man at det av og til forekommer slike hull, men av varierende størrelse. Dette er størst problem for RNN og LSTM og forklarer mye av problemene disse hadde med å predikere lengre fram i tid. RNN og LSTM i motsetning til standard nevralt nettverk ser på tidsserier av data. Med en tidsserie på ti historiske data vil dette bety at modellen vil observere et slikt hopp i dataen ti ganger, én gang som output og ni ganger som en del av den historiske dataen. I tillegg husker RNN og LSTM aktiveringer over tid. Hullene i ruten vil da potensielt ødelegge for treningen av nettverkene da dette vil bli med i beregningen av outputen for senere tilfeller. Et standard nevralt nettverk derimot vil observere dette hoppet kun en gang og etterpå så vil det være glemt når nye outputverdier skal beregnes. Dette kan også forklare hvorfor LSTM-modellene og RNN-modellene ofte predikerte for store avstander mellom punktene.



Figur 7.7 Seilaser med kun data på slutten

Man kan argumentere for at før man startet treningen og prediksjonen av de nevrane nettverkene så hadde man grundig gått gjennom all dataen tilgjengelig og tatt bort og rettet på

dataen som så ut til å ikke stemme. For eksempel interpolert inn nye punkter i mellom steder med gap i ruten, eventuelt fjernet disse seilasene fra datasettet. Siden denne forskningen ønsket å se på muligheten for å benytte nevralt nettverk på norskekysten for å kunne predikere fartøys posisjon basert på AIS-data ble disse tatt med. Det er urealistisk å på forhånd skulle manuelt analysere data fra hele norskekysten hvor tre år med historiske data utgjør rundt 3TB som er gruppert på 14 000 forskjellige til- og fra-havner.

En annen kilde til feilprediksjoner er hvor høy nøyaktighet det kreves for at man skal kunne predikere godt på longitude og latitude. Normaliseringen av dataen før den trener og predikerer har sørget for at verdiene som kan bli predikert ligger innenfor det geografiske avgrensede området, ved at normaliseringen er gjort med hensyn på minimum og maksimum verdier innad i datasettet. Det geografiske området er avgrenset på en longitude fra 13.19 til 13.56 og latitude fra 66.1885 til 66.25. Siden at dataen blir skalert ned til verdier mellom null og én, samt at nettverkene ikke kan predikere longitude og latitude verdier over eller under dette, klarer man å predikere ganske nøyaktig neste punkt. Fremdeles kreves det veldig nøyaktige prediksjoner fordi longitude spekteret representerer omtrent 41 155 meter og latitude spekteret representerer omtrent 6 650 meter. Dette vil si at hver prosent feil på longitude vil føre til en feil på omtrent 410 meter og hver prosent feil på latitude føre til en feil på omtrent 66 meter. Om det geografiske område nettverket skal trene på blir utvidet, vil det føre til at nettverket må bli enda mer nøyaktig på å predikere de normaliserte dataene, fordi de da representerer et større spekter av longitude- og latitude- verdier.

7.1 Modellvalg

Denne forskningen så på bruk av standard nevralt nettverk, RNN og LSTM for å predikere ruter på maritim trafikk. Fremdeles måtte noen valg bli tatt for hvordan man skulle prøve ut modeller av disse typene. Valgene av strukturen og parameterne på de modellene som eksperimentene ble utført på ble tatt ved at etter en initial testperiode. I denne perioden ble det utprøvd forskjellige parametere og strukturer på modelltypene som ble testet. Dette førte til følgende valg for modellene eksperimentene ble testet på. På de standard nevralt nettverkene ble det valgt å ikke benytte annet enn ReLU som aktiveringsfunksjon på de skjulte lagene. For LSTM og RNN ble det valgt å benytte ti historiske data for framtidsprediksjonen og én måte å utføre framtidsprediksjon på, ved å rekursivt gi modellen predikerte data fram til den hadde predikert ønsket antall. For alle modeller ble det valgt å ha en batchstørrelse på 32 og adams som optimizer. For standard nevralt nettverk kom det fram i utprøvingen at sigmoid var best

egnet som outputfunksjon. For LSTM og RNN ble dette gjort ved en initial test med en modell på to skjult lag med 32 noder som trente over 500 epoker. Den outputfunksjonen som gjorde det best i ettstegsprediksjonen ble benyttet i de videre eksperimentene for denne modelltypen.

Alle modellene ble trent på kun prediksjon av ett punkt fram i tid. Det kunne vært at det å trene modellene på å predikere flere punkter fram i tid hadde vært en god løsning. Dette er ikke nødvendigvis tilfelle da dette både vil kreve lengre treningstid per epoke og antageligvis flere epoker med trening enn trening på ett steg fram i tid.

Selv om standard nevrale nettverk ikke er designet spesifikt for tidsseriedata, kunne det å forsøke å trene dem med å ta inn mer enn ett punkt som input gjort de bedre på å predikere det utfordrende område vist i Figur 7.1. Dette kunne samtidig ha gjort dem dårligere på å predikere flere punkter fram i tid, basert på de utfordringene man så at LSTM og RNN hadde med dette. Spesielt på grunn av hullene observert i treningsdataen. Dette fordi de da ville fått inn oftere slike hull som input.

Denne forskningen så i hovedsak på arkitekturen til modellene, altså antall noder og antall lag. Det ble også sett på omfanget av å øke antall epoker nettverkene trente over, hvor alle modeller fikk trent opp til 20 000 epoker. Ved å gjøre det på denne måten sørget man for at alle modelltypene hadde samme utgangspunkt når man skulle sammenligne dem.

Det er mange parametere og veldig mange kombinasjoner av disse man kan ha i de forskjellige typene nevrale nettverkene, samt mange forskjellige aktiveringsfunksjoner man kan benytte. Denne forskningen har sett kun på et fåtall av disse og sannsynligheten for at det eksisterer modeller som kan være mer optimale for problemområdet er stor. Målet med denne forskningen var ikke å finne den mest optimale modellen innad en type nevrale nettverk for prediksjon av AIS-data, men å se på forskjellige typer nevrale nettverk og se om man kunne predikere framtidige data på AIS-data for et fartøy.

7.2 Input-parametervalg

For valg av input-parametere til modellene ble det valgt å se på de dataene som var lett tilgjengelig. Det ble benyttet at det på forhånd var koblet sammen data fra AIS-meldinger fra norskekysten og data fra SSNN. De inneholdt tidspunkt for AIS-meldingen og hvilken seilas den tilhørte. Samtidig gjorde dette det lett å få hentet ut data fra en til- og fra-havn. Dette

gjorde det mulig å kunne få dataene over på tidsserieformat ved å gruppere på seilas og sortere på tidspunkt.

Det man sto igjen med data å predikere på var da COG, SOG, longitude og latitude. Disse verdiene ble brukt for å trene nettverket på å predikere de samme parameterne ett steg fram i tid. Grunnen til at COG og SOG også ble predikert var for å kunne benytte de predikerte verdiene når man skulle predikere neste verdi basert på forrige predikerte verdi. Dette er et minimum av parametere man bør ha for å kunne predikere posisjonen. Man kan argumentere for at kun å predikere longitude og latitude ville vært nok og kunne kanskje ha gitt bedre resultater i det området som ble valgt for testing. Siden denne forskningen var for å se om det er mulig å benytte nevralt nettverk på områder på norskekysten må man trene på å også predikere disse verdiene. Hvordan skal nettverkene kunne vite i et mer komplekst område hvorvidt det skal utføre en sving, eller kjøre rett fram, eller kanskje snu om det bare ser på longitude og latitude og ikke har noen formening om fartøyets fart og retning. Et nevralt nettverk kan ikke benytte avstanden og svingningsgraden mellom tidligere punktene for dette, da AIS-meldinger sendes oftere om farten er høy eller retningen endres. I tillegg om nettverket skal benyttes til å predikere lengre fram i tid enn hva som ble utprøvd i denne forskningen er trolig ikke retningen på historiske data nok til å forutse framtidig retning.

Det eksisterer mange flere parametere som kunne ha gjort predikeringen av nettverkene bedre. Denne forskningen har kun tatt i bruk de høyst nødvendige parametere for å kunne predikere rutene. Grunnen til dette var at det var det som var tilgjengelig av data i databasen som dataene ble hentet fra. I en AIS-melding har man for eksempel tilgang til type fartøy og lengden på fartøyet. Dette er verdier som trolig er med på å avgjøre valg av kjørebane og rute. Andre verdier som kan påvirke et fartøys rute kan være: tyngde på last, type last, flo eller fjære og været.

8 Konklusjon og videre arbeid

8.1 Konklusjon

Det er utviklet og testet tre forskjellige typer nevrale nettverk for å predikere posisjonen til fartøy på kysten ved bruk av AIS-data koblet sammen med SSNN-data. Det er utført to eksperimenttyper på mange arkitekturer av disse tre nettverkstypene for å se om dette er mulig. Først ble det utført en test for å se om prediksjon ett steg fram i tid er mulig med modellen. Deretter ble det utført prediksjon ti steg fram i tid i en sving for å se om det er mulig å predikere lengre fram i tid med modellene. Dette for å kunne svare på om man kan anvende nevrale nettverk til å nøyaktig predikere ruten til et fartøy. Både det å predikere én AIS-melding fram i tid og det å predikere den lengre fram i tid. Det siste målet med forskningen var å undersøke om man kunne oppnå større nøyaktighet gjennom å anvende rekurrente nevrale nettverk.

Det er utført to geografiske avgrensninger. Først er det hentet ut kun data som tilhører en til- og fra-havn. Deretter er det kun tatt data fra et utfordrende område innenfor dette som inneholder to svinger etter hverandre.

Alle modeller utprøvd i denne forskningen klarte å predikere ett steg fram i tid. Den største utfordringen her var på slutten av ruten hvor noen av modellene hadde problemer med å predikere ruten. Så man kan konkludere med at man kan predikere ruten til et fartøy én AIS-melding fram i tid med bruk av nevrale nettverk. Det man også ser er at man ved å bruke rekurrente nevrale nettverk oppnår en større nøyaktighet på dette. Det gjelder både for RNN-modellene og LSTM-modellene.

For å se om man kan predikere ruten lengre fram i tid ble det forsøkt å predikere opp til ti steg fram i tid på modellene. Her var det en del varierende resultater blant modellene. Man ser at de standard nevrale nettverkene gjorde denne prediksjonen best og hadde mest stabile resultater. De fleste modeller klarte å predikere noen punkter fram i tid korrekt, og noen av modellene klarte å predikere alle ti punktene fram i tid med ganske god nøyaktighet hvor én predikerte det tiende punktet til å være på punkt nummer åtte. Det konkluderes derfor med at det er mulig å predikere ruten lengre fram i tid ved bruk av nevrale nettverk. Det kan også konkluderes med at for prediksjon lengre fram i tid ikke oppnådde større nøyaktighet med bruk av rekurrente nevrale nettverk.

Standard nevrale nettverk oppnådde en større nøyaktighet enn rekurrente nevrale nettverk på ti-stegsprediksjon. På en annen side håndterte de rekurrente nevrale nettverkene RNN og LSTM bedre det utfordrende området på slutten av ruten hvor det var en splitt i hvor treningsdataen gikk. RNN og LSTM gjorde det også generelt bedre på ettstegsprediksjonen. Splitting og sammenknytting av ruter er noe man ofte ser langs norskekysten og det er derfor viktig at modellene håndterer dette på en god måte. Basert på disse funnene er det ikke mulig å kunne konkludere hvilken type nevral nettverk som egner seg best for prediksjon langs norskekysten. Dette kommer an på hvilke kvaliteter man verdsetter størst, både framtidsprediksjon og håndtering av splitter og sammenknyttinger av ruter er viktige kvaliteter for prediksjon på kysten i Norge. Samtidig så man at RNN og LSTM klarte å predikere opp til seks punkter korrekt, og standard nevrale nettverk klarte å predikere hele ruten til fartøyet i ettstegsprediksjonen. Dermed kan det konkluderes med at man kan benytte nevrale nettverk for å predikere ruten til et fartøy.

8.2 Videre arbeid

Det som anbefales av videre arbeid er å teste ut å benytte flere parametere inn til modellen for å se om nøyaktigheten vil bli bedre. Fartøy følger ofte litt forskjellige ruter, dette kan ha en sammenheng med hvilken type fartøy dette er. Andre parametere som det kan tenkes påvirker ruten til et fartøy kan være hvor tungt lastet det er, eller hvilken type last den har om bord. Andre faktorer kan være lengden på fartøyet, om det er flo eller fjære, eller hvordan været er. I denne forskningen ble det benyttet modeller som utelukkende hadde trent på å predikere ett steg fram i tid. Det kan være at man hadde oppnådd en større nøyaktighet om man trente modeller på å predikere flere punkter fram i tid. En anbefaling til videre arbeid er derfor å undersøke om man da oppnår bedre resultater. En annen anbefaling er å forsøke å trene standard nevrale nettverk med flere punkter enn ett som input, dette kan være noe som gjør at de predikerer områder hvor data splittes bedre.

Videre så burde nettverkene prøves ut på mer utfordrende områder. Dette kan være ruter som inneholder splitt i dataen, altså at en hovedrute deler seg opp i to eller flere ruter. Områder hvor flere ruter samler seg til en stor rute, og da også teste nettverkene i områder hvor ruter både deler seg og samler seg. En annen ting som hadde vært av interesse var å se om man kan predikere to svinger etter hverandre.

Det anbefales å dele opp i mindre geografiske områder dersom dette skal implementeres for hele norskekysten, det anbefales også at dette er delt opp på til- og fra-havn. Dette for å ha flere spesialiserte nevrale nettverk i stedet for å ha ett nettverk som skal kunne håndtere hele norskekysten. Ved å dele opp i til- og fra-havn sørger man for at rutene nettverket skal predikere på går i en retning. Hvor store geografiske områder man skal dele opp norskekysten i er noe som må videre eksperimenteres på. Større geografisk område fører til at nettverket må være enda mer nøyaktig i prediksjonen, da normaliserte longitude- og latitude-verdier representerer et større spekter av verdier. Samtidig vil det at man deler opp i mindre geografiske områder krevere flere nevrale nettverk for å dekke hele norskekysten. Et alternativ til å dekke hele norskekysten er å dekke kritiske områder hvor flest farlige situasjoner oppstår.

9 Bibliografi

- [1] Johnsrud, H., et al., *Analyse av sannsynligheten for ulykker med tap av menneskeliv og akutt forurensning fra skipstrafikk i norske farvann*. 2015.
- [2] *Peker ut videre kurs for arbeid med sjøsikkerhet*. 2015 01.11.2016 [sitert 2017 03 Nov]; Tilgjengelig fra: <http://kystverket.no/Nyheter/2015/November/Peker-ut-videre-kurs-for-arbeid-med-sjosikkerhet/>.
- [3] Johnsrud, H., et al., *Analyse av ulykkesstatistikken for norske farvann de siste 30 årene (1984-2013)*. 2015.
- [4] Merwe, F., et al., *Årsaksanalyse av grunnstøtinger og kollisjoner i norske farvann*. 2015.
- [5] Kystverket, *Analyse av sannsynligheten for ulykker med tap av menneskeliv og akutt forurensning fra skipstrafikk i norske farvann i 2040*. 2016.
- [6] Ly, J.M., *Hendelser håndtert i 2016*. 2017.
- [7] Kystverket. *Flere akutte forurensningshendelser i 2016*. 2017 03.05.2017 [sitert 2017 08. Nov]; Tilgjengelig fra: <http://kystverket.no/Nyheter/2017/april/akutt-forurensning-2016/>.
- [8] Dimmen, A. og T. Langemyr, *Kystverkets oppsummering med vurderinger og anbefalinger*.
- [9] Liu, B., et al., *Knowledge-based Clustering of Ship Trajectories Using Density-based Approach*. 2014: s. 603-608.
- [10] Perera, L.P., P. Oliveira, og C.G. Soares, *Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction*. IEEE Transactions on Intelligent Transportation Systems, 2012. **13**(3): s. 1188-1200.
- [11] Pallotta, G., M. Vespe, og K. Bryan, *Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction*. Entropy, 2013. **15**(6): s. 2218-2245.
- [12] Diker, A.C. og E. Nasibov *Estimation of traffic congestion level via FN-DBSCAN algorithm by using GPS data*. 2012. 1-4 DOI: 10.1109/ICPCI.2012.6486279.
- [13] Idrissov, A. og M.A. Nascimento *A Trajectory Cleaning Framework for Trajectory Clustering*. 2012.
- [14] Lee, J.-G., J. Han, og K.-Y. Whang, *Trajectory clustering: a partition-and-group framework*, in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. 2007, ACM: Beijing, China. s. 593-604.
- [15] Dobrkovic, A., M.-E. Iacob, og J.v. Hillegersberg, *Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data*, in *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*. 2015, ACM: Graz, Austria. s. 1-8.
- [16] Xu, T., X. Liu, og X. Yang, *Ship Trajectory Online Prediction Based on BP Neural Network Algorithm*, in *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*. 2011. s. 103-106.
- [17] Simsir, U. og S. Ertugrul, *Prediction of manually controlled vessels' position and course navigating in narrow waterways using Artificial Neural Networks*. Applied Soft Computing, 2009. **9**(4): s. 1217-1224.
- [18] IMO, *Revised guidelines for onboard operational use of ship Automatic Identification Systems (AIS)* 2016.
- [19] Zissis, D., E.K. Xidias, og D. Lekkas, *Real-time vessel behavior prediction*. Evolving Systems, 2016. **7**(1): s. 29-40.

- [20] Daranda, A., *Neural network approach to predict marine traffic*. Baltic Journal of Modern Computing, 2016. **4**(3): s. 483-495.
- [21] Goodfellow, I., Y. Bengio, og A. Courville, *Sequence Modelling: Recurrent and Recursive Nets*, in *Deep Learning*. 2016, MIT Press. s. 363-405.
- [22] Graves, A. og N. Jaitly. *Towards end-to-end speech recognition with recurrent neural network*. presenter på *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.
- [23] Sutskever, I., O. Vinyals, og Q.V. Le. *Sequence to sequence learning with recurrent neural network*. presenter på *Advances in Neural Information Processing Systems (NIPS 2014)*. 2014.
- [24] Fernando, T., et al., *Soft + Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection*. CoRR, 2017. **abs/1702.05552**.
- [25] Ester, M., et al., *A density-based algorithm for discovering clusters in large spatial databases with noise*. 1996: s. 226-231.
- [26] Mehle, A., B. Likar, og D. Tomaževič, *In-line recognition of agglomerated pharmaceutical pellets with density-based clustering and convolutional neural network*. IPSJ Transactions on Computer Vision and Applications, 2017. **9**(1): s. 7.
- [27] Khashei, M. og M. Bijari, *An artificial neural network (p,d,q) model for timeseries forecasting*. Expert Systems with Applications, 2010. **37**(1): s. 479-489.
- [28] Juang, C.F., *A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design*. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 2004. **34**(2): s. 997-1006.
- [29] Liu, S., et al. *A Recursive Recurrent Neural Network for Statistical Machine Translation*. presenter på *Proceedings of the Association for Computational Linguistics*. 2014. ACL.
- [30] Ku, C. og K. Lee, *Diagonal recurrent neural networks for dynamic systems control*. IEEE Transactions on Neural Networks, 1995. **6**(1): s. 144-156.
- [31] Yi, Z. og K.K. Tan, *Hopfield Recurrent Neural Networks*, in *Convergence Analysis of Recurrent Neural Networks*. 2004, Springer US: Boston, MA. s. 15-32.
- [32] Kosko, B., *Bidirectional Associative Memories*. IEEE Transactions on Systems, Man, and Cybernetics, 1988. **18**(1): s. 49 - 60.
- [33] Elman, J.L., *Finding Structure in Time*. Cognitive Science, 1990. **14**(2): s. 179-211.
- [34] Jordan, M.I., *Chapter 25 - Serial Order: A Parallel Distributed Processing Approach*, in *Advances in Psychology*, J.W. Donahoe and V. Packard Dorsel, Editors. 1997, North-Holland. s. 471-495.
- [35] Jaeger, H. og H. Haas, *Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication*. Science, 2004. **304**(5667): s. 78.
- [36] Hochreiter, S. og J. Schmidhuber, *Long Short-Term Memory*. Neural Computation, 1997. **9**(8): s. 1735-1780.
- [37] Gers, F.A., J. Schmidhuber, og F. Cummins *Learning to forget: continual prediction with LSTM*. IET Conference Proceedings, 1999. 850-855.
- [38] Graves, A. og J. Schmidhuber, *Framewise phoneme classification with bidirectional LSTM and other neural network architectures*. Neural Networks, 2005. **18**(5): s. 602-610.
- [39] Gers, F.A. og J. Schmidhuber. *Recurrent nets that time and count*. presenter på *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. 2000.

- [40] Cho, K., et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. s. 1724-1734.
- [41] Chollet, F., *Fundamentals of machine learning*, in *Deep Learning with Python*. 2017, Manning Publications. s. 93-116.