

Jemila Daush

Passiv akustiske statusindikatorer - en mulighetsstudie

Masteroppgave i Elektronisk systemdesign og innovasjon

Veileder: Lise Lyngsnes Randeberg

Medveileder: Eirik Svendsen

Juni 2022

Jemila Daush

Passiv akustiske statusindikatorer - en mulighetsstudie

Masteroppgave i Elektronisk systemdesign og innovasjon
Veileder: Lise Lyngsnes Randeberg
Medveileder: Eirik Svendsen
Juni 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for elektroniske systemer



Kunnskap for en bedre verden

Forord

Denne masteroppgaven er siste del av studiet Elektronisk systemdesign og innovasjon med spesialisering i signalbehandling og kommunikasjon ved NTNU. Oppgaven er gitt av SINTEF Ocean, med Eirik Svendsen som bedriftsveileder og Lise Lyngsnes Randeberg som hovedveileder. Svendsen, Kristbjörg Jónsdóttir og John Reidar Mathiassen fra SINTEF Ocean har stått for faglig veiledning innen maskinlæring, signalbehandling og akvakultutr. Hovedveileder Randeberg har gitt veiledning i rapportskriving.

Jeg ønsker å takke Svendsen, Jónsdóttir og Mathiassen for god og jenvlig oppfølging i løpet av hele semesteret. Likedeles takker jeg Randeberg for veiledning det siste året. En stor takk til familie og venner som har støttet og heiet på meg i fem år.

Sammendrag

Et lakseoppdrettsanlegg har ofte flere merder der hver merd kan ha opp til 200 000 laks om gangen. Høy biomassetetthet og endret miljø under operasjoner kan påvirke fiskenes velferd blant annet grunnet stress eller dødelighet. Masteroppgaven er gitt av SINTEF Ocean hvor formålet er å utforske mulige statusindikatorer med et søkelys på å forbedre fiskevelferden i norsk lakseoppdrett. I denne oppgaven er laksens aktivitet tatt for seg som en mulig adferdsindikator, satt opp mot en lydmiljøindikator som detekterer støy, natt og fôring. Oppgaven er formet som en mulighetsstudie der det forskes på om det i det hele tatt er mulig å detektere aktivitet, fôring, natt og støy blant laks i merder ved bruk av akustisk signalbehandling og maskinlæring. Et slikt sensorsystem skal kunne varsle om unormal aktivitet, i tillegg til å si om det for eksempel har vært en unormal lyd dag med mye støy. Det ble produsert én lydmiljøindikator og 24 aktivitetsindikatorer, basert på ulike kombinasjoner av digitale signalbehandlingsmetoder og maskinlæringsalgoritmer.

Abstract

There is often several cages in a salmon farm, where each cage can have up to 200,000 salmon at a time. High biomass density and changed environment during operations can affect the fish's welfare due to stress or mortality, among other things. This master's thesis is given by SINTEF Ocean where the purpose is to explore possible status indicators with a focus on improving fish welfare in Norwegian salmon farming. In this task, the salmon's activity is considered as a possible behavior indicator, set up against a sound environment indicator that detects noise, night and feeding. The thesis is shaped as a feasibility study where it is researched whether it is at all possible to detect activity, feeding, night and noise among salmon in cages using acoustic signal processing and machine learning. Such a sensor system should be able to warn of abnormal activity, in addition to saying whether it has, for example, been an abnormal sound day with a lot of noise. One sound environment indicator and 24 activity indicators were produced, based on different combinations of digital signal processing methods and machine learning algorithms.

Nomenclature

ANN eng. Artificial Neural Networks, kunstig nevralt nettverk

BP eng. Backpropagation, tilbakeforplantning

CNN eng. Convolutional Neural Networks, konvolusjonelle nevralt nettverk

DFT eng. Discrete Fourier Transform, diskret Fourier transformasjon

FHF Fiskeri- og havbruksnæringens forskningsfinansiering

FN falsk negativ

FP falsk positiv

FT eng. Fourier Transform, Fourier transformasjon

ISTFT eng. Inverse Short time Fourier Transform, invers kort tid Fourier transformasjon

MFCC eng. Mel-frequency cepstral coefficients, Mel-frekvens cepstrale koeffisienter

MLP eng. Multi-layer Perceptron, Flerlags Perceptron

PC eng. Principal Component, Prinsippal komponent

PCA eng. Principal Component Analysis, Prinsippal komponent analyse

SN sann negativ

SP sann positiv

STF eng. Short Time Fourier, kort tid Fourier

STFT eng. Short time Fourier Transform, kort tid Fourier transformasjon

SVM Support Vector Machine, støttevektormaskin

VLF eng. Very-Low-Frequency, svært lavfrekvent

Innhold

Forord

Abstract i

Sammendrag i

Nomenklatur ii

1 Introduksjon 1

2 Teori 3

2.1 Akustikk og akustisk signalbehandling 3

2.1.1 Sampling og samplingsteoremet (Nyquist-regelen) 3

2.1.2 Bølgeligning 4

2.1.3 Fourier transformasjon 5

2.1.4 Kort tid Fourier transformasjon 6

2.1.5 Digital filtrering 7

2.1.6 Mel-frekvens cepstrale koeffisienter 8

2.1.7 Normalisering og standardisering 9

2.1.8 Prinsippal komponent analyse 10

2.2 Maskinl ring 11

2.2.1 Dyp l ring 12

2.2.2 St ttevektormaskin 16

2.3 Ytelsesberegninger 19

2.4 k -fold kryssvalidering 20

3 Metode 21

3.1 Metode for innsamlet data 21

3.2 Overordnet metode 24

3.3 Analysere data 26

3.3.1 Segmentering og skalering 26

3.4 Filtrering og resampling 27

3.5	Representasjon av lyd	28
3.6	Valg av maskinlæringsmodell og arkitektur	29
3.6.1	2D CNN og 1D CNN	29
3.6.2	MLP	32
3.7	Trening og test	32
4	Resultater	34
4.1	Lydmiljøindikator	34
4.1.1	Datasekk	34
4.1.2	Trening	35
4.1.3	Test	36
4.2	Aktivitetsindikator	41
4.2.1	Båndpassfilter	41
4.2.2	Datasekk	41
4.2.3	Trening	50
4.2.4	Test	61
4.2.5	Generaliserte modeller	63
5	Diskusjon	67
5.1	Datasekk	68
5.1.1	Treningssett	68
5.1.2	Testsett	69
5.2	Metode	70
5.2.1	Signalbehandling for lydmiljøindikator	70
5.2.2	Signalbehandling for aktivitetsindikator	70
5.2.3	Maskinlæring	71
5.3	Trening	72
5.3.1	Tapsfunksjoner	72
5.3.2	Kryssvalidering	73
5.4	Test	74
5.4.1	Lydmiljøindikator	74
5.4.2	Aktivitetsindikator	75

5.4.3 Troverdighet av prediksjonsplott	76
5.5 Fremtidig arbeid	77
6 Konklusjon	78
Referanser	79

1 Introduksjon

Akvakultur er dyrking av organismer i vann, og omfatter deriblant oppdrettsnæringen [1]. Det er en teknologisk avansert og eksportrettet næring som er i rask endring. Fisk er et av Norges største eksportvarer med en salgsverdi på 120,8 milliarder kroner i 2021, hvorav 67% av salget var laks [2]. Fiskeindustrien slo nye rekorder det samme året, blant annet passerte eksportvolumet 3 millioner tonn med en verdiøkning på 14,3% sammenlignet med året før.

Norsk fiskeoppdrett ble sterkt utbredt på 70-tallet, med årene vokste næringen og nye oppfinnelser i anleggene ble til. For eksempel bruk av notposer fylt med fisk i vann, også kalt for *merd* [1]. De er i bruk for oppbevaring, fôring og stell av fisk og er plassert ulikt basert på oppdrettsarten og anleggsfasilitetene. Norsk atlantisk laks er som oftest oppbevart i tanker på land i deres ferskvannsfase, og deretter blir de sluppet ut i havet, da plassert i slike merder. Et lakseoppdrettsanlegg har ofte flere merder der hver merd kan ha opp mot 200 000 laks om gangen. Den enorme mengden fisk fører til utfordringer med overvåking av fiskevelferden. Høy biomassetetthet og endret miljø under operasjoner kan påvirke fiskenes velferd blant annet grunnet stress eller dødelighet.

SoundWell II er et prosjekt som tar for seg dette problemet, de forsker på mulighetene for å utvikle en ikke-invasiv akustisk statusindikator for oppdrettslaks. Målet er å bruke indikatoren for overvåking og forbedring av fiskevelferd [3]. *Teknologiske verktøy for objektiv dokumentasjon av velferd hos laksefisk ved håndteringsoperasjoner* (OWITOOLS) er et lignende prosjekt [4]. De har blant annet et ønske om å bruke teknologi og effektive verktøy for å øke menneskets evne til å ha kontroll over akvakulturoperasjoner på anlegget, med et mål om å kunne ivareta god velferd [4].

Masteroppgaven er gitt av *SINTEF Ocean* som er ansvarlig organisasjon for OWITOOLS prosjektet og bidragsytere i *Soundwell II* [4] [3]. Data brukt i denne oppgaven er samlet inn av *SINTEF Ocean* for prosjektene nevnt ovenfor. Oppgavens mål er å utforske mulige statusindikatorer med et fokus på å forbedre fiskevelferden i norsk lakseoppdrett. *Havforskningsinstituttet* har foreslått gode velferdsindikatorer, for eksempel fiskens adferd, helse og fysiologisk funksjon [5]. I denne oppgaven er laksens aktivitet tatt for seg som en mulig adferds indikator, satt opp mot en lyd miljøindikator. Oppgaven er formet som en mulighetsstudie der det forskes på om det er mulig å detektere aktivitet, fôring, natt og støy blant laks i merder ved bruk av akustisk

signalbehandling og maskinl ring. Et slikt sensorsystem skal kunne varsle om unormal aktivitet, i tillegg til   si om det for eksempel har v rt en unormal lyddag med mye st y.

Relevant teori vil f rst bli introdusert. Akustisk signalbehandling og maskinl ring er hovedsakelig fagomr dene som er av interesse for oppgaven. Siden er materialet gitt av SINTEF Ocean blitt presentert. Derav lydopptak fra merder i Rataren fiskeoppdrettsanlegg og skriftlige logger fra opptaksdagene. Videre er metoder valgt for oppgaven blitt beskrevet, b de for lydmilj indikatoren og aktivitetsindikatoren. Evalueringsm linger av trente maskinl ringsmodeller og test av indikatorene blir s  vist i resultat seksjonen. Til slutt diskuteres resultatene opp mot hverandre og teorien. Mulighetsstudiet vil konkludere om oppgaven er en god teknisk l sning p  problemet, om det er mulig   bruke maskinl ring gitt signalbehandlingen valgt i metoden, og om maskinl ring er n dvendig   bruke i det hele tatt.

2 Teori

2.1 Akustikk og akustisk signalbehandling

Lyd kan produseres på en rekke ulike måter, inkludert ved vibrasjon og endring av luftstrøm [6]. Lyd er et sanseintrykk som kan oppfattes av skapninger med hørsel [7]. Mennesket oppfatter et bredt spenn av frekvenser, ofte i frekvensområdet 20-20 000 Hz. *Infralyd* og *ultralyd* er frekvensområdene utenfor menneskets oppfatning av lyd, og er henholdsvis under og over vårt frekvensområde. Støy defineres som uønsket lyd i miljøet, det er lyder man kunne vært foruten og kan i visse tilfeller føre til skade på miljøet og skapninger med hørsel [6]. Fisker bruker hørsel og lyd for blant annet kommunikasjon og orientering. Enhver lyd som forhindrer fiskenes hørsel, kan påvirke deres overlevelse og velferd. I oppdrettsanlegg er det mange menneskeskapte lyder som kan oppfattes som støy for fiskene. Ifølge Popper et al. oppfatter fisker lave frekvenser under 1k Hz [8]. Svært lavfrekvent (eng. Very-Low-Frequency, VLF) båndet i havet er i intervallet 1 til 100 Hz. Lyder av værforhold, jordskjelv, sjøpattedyr og menneskeskapte lyder påvirker dette båndet [9].

Lydbølger kan være rene sinusbølger med én frekvens, eller en sum av flere sinusbølger med ulike frekvenser. Fourier transformasjon (eng. Fourier Transform, FT) er et verktøy som dekomponerer en bølgeform i flere alternative representasjoner basert på sinus og cosinus funksjoner. FT viser at enhver bølgeform er en sum av sinusoder.

Et av de viktigste stegene ved maskinlæring er signalbehandling og preprosessering. Signalbehandling er et bredt begrep som innebærer manipulasjon av data for å sørge for at brukt data er av interesse. For at maskinlæringsmodeller ikke skal trene på trivielt støy, må dette kompenseres for ved signalbehandling og preprosessering.

2.1.1 Sampling og samplingsteoremet (Nyquist-regelen)

For å representere et analogt lydsignal i digital form, samler man det originale lydsignalet med en spesifikk samplings rate. Samplings rate er definert av variabelen samplingsfrekvens, f_s , og definerer hvor ofte et signal skal samples per sekund. Tar man for seg et kontinuerlig tidssignal $x(t)$, defineres det samlede diskrete signalet som $x[n] = x(nT_s)$, hvor n er et heltall og T_s er samplings perioden. $x[n]$, i motsetning til $x(t)$, er en sekvens av tall der hver variabel i sekvensen

har en distinkt verdi. Hvorav $x(t)$ kan illustreres som en sammenhengende graf.

For å rekonstruere et analogt signal, $x(t)$, som inneholder frekvenser opp til f_{max} , sier samplings teoremet at det må samples med en $f_s > 2f_{max}$ [10]. Hvis dette kravet ikke er oppfylt vil det originale signalet bli forvrengt under samplingen. Den høyeste frekvensen en samplingsfrekvens kan sample uten forvrenging er $f_n = \frac{1}{2}f_s$, hvor f_n er kalt Nyquist frekvensen eller Nyquist grensen. Å nedsample betyr å sample et signal på nytt med en lavere samplingsfrekvens for å minimere data samples per signal.

2.1.2 Bølgeligning

Denne del seksjonen er basert på og inspirert av *Acoustic Signal Processing* i *Springer Handbook of Acoustics* av Thomas Rossing og William M. Hartmann[10].

Det enkleste signalet er en sinusbølge med én enkelt frekvens. En sinusfunksjon $x(t)$, avhengig av tid t , er vist i ligning 1,

$$x(t) = C \sin(\omega t + \phi) \quad (1)$$

hvor C er amplitude, $\omega = 2\pi f$ er vinkelfrekvens avhengig av frekvens f og ϕ er startsfase. $f = 1/T$, der T er tiden for én bølgeperiode. En kompleks bølge er summen av to sinusbølger, som vist i ligning 2,

$$x(t) = C_1 \sin(\omega_1 t + \phi_1) + C_2 \sin(\omega_2 t + \phi_2) \quad (2)$$

Hver fikset komponent, C , ω og ϕ , kan være ulik for sinusbølgene og er dermed merket med 1 eller 2 for de to ulike sinusbølgene den komplekse bølgen $x(t)$ består av. Dette kan videreføres til en generell bølgeform som kan sees i ligning 3.

$$x(t) = \sum_{n=1}^N C_n \sin(\omega_n t + \phi_n) \quad (3)$$

Ligningen er beskrevet som en sum av N komponenter, der n er nummeret på hver enkelt sinusbølge. Ved bruk av den trigonometriske identiteten i ligning 4,

$$\sin(\theta_1 + \theta_2) = \sin\theta_1\cos\theta_2 + \sin\theta_2\cos\theta_1 \quad (4)$$

kan man omskrive ligning 3 til ligning 5,

$$x(t) = \sum_{n=1}^N A_n \cos(\omega_n t) + B_n \sin(\omega_n t) \quad (5)$$

hvor $A_n = C_n \sin\phi_n$ og $B_n = C_n \cos\phi_n$.

2.1.3 Fourier transformasjon

En frekvensavhengig representasjon av et signal kan fremstilles ved å Fourier transformere det tidsavhengige signalet. Generelt er FT en kompleks funksjon bestående av både reelle og imaginære deler. FTen er en kontinuerlig funksjon av frekvens, transformasjonen krever konseptet om negative frekvenser og pleier å være symmetrisk med hensyn til forekomsten av både positive og negative frekvenser. FTen av signalet $x(t)$ er gitt av integralet vist i ligning 6,

$$X(\omega) = \mathcal{F}[x(t)] = \int_{-\infty}^{\infty} e^{-i\omega t} x(t) dt \quad (6)$$

For et diskret signal $x[n]$ byttes integralet med en sum, som vist i ligning 13,

$$X(\omega_k) = \sum_{n=0}^{N-1} x[n] e^{-i\omega_k n} \quad (7)$$

hvor $\omega_k = \frac{2\pi}{N}k$, der N er antall samples og $k = 0, 1, \dots, N-1$. Dette er også kalt diskret FT (eng. Discrete Fourier Transform, DFT).

Den inverse FT uttrykker signalet som en funksjon av tid i form av transformasjonen, som vist i ligning 8.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega t} X(\omega) d\omega \quad (8)$$

Invers av DFT er vist i ligning 9.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(\omega_k) e^{i\omega_k n} \quad (9)$$

FTen er som oftest kompleks, som nevnt tidligere, selv for tilfeller med reelle funksjoner av tid som $x(t)$. Dette medfører et krav om at $X(-\omega)$ må være kompleks konjugant av $X(\omega)$, som vist i ligning 10.

$$X(-\omega) = X^*(\omega) \quad (10)$$

Som igjen sier at $\text{Re } X(-\omega) = \text{Re } X(\omega)$ og $\text{Im } X(-\omega) = -\text{Im } X(\omega)$. Magnituden av det komplekse signalet kan regnes ut med ligning 11,

$$|X(\omega)| = \sqrt{[\text{Re } X(\omega)]^2 + [\text{Im } X(\omega)]^2} \quad (11)$$

2.1.4 Kort tid Fourier transformasjon

Kort tid Fourier transformasjon (eng. Short Time Fourier Transform, STFT) er et vanlig brukt verktøy for analyse, modifikasjon, og syntese av tale og lydsignaler. STFT kan bli sett på som et glidende vindu transformasjon. Gitt en input $x(t)$ av tilfeldig varighet, blir datasegmenter utvunnet ved vanlige intervaller ved bruk av et tidsbegrenset vindu $w[n]$. Disse segmenterte signalene eller rammene kan bli uttrykket som vist i ligning 12,

$$x_l[n] = w[n]x[n + lL], \quad 0 \leq n \leq N - 1 \quad (12)$$

hvor N er vinduslengden, l er ramme indeksen, og L er hopp størrelsen, med andre ord mellomrommet i samples mellom påfølgende applikasjoner av det glidende utvinningsvinduet. Indeks n er en lokal tidsindeks, altså en indeks relativ til starten av det glidende vinduet. For hver signalramme er en DFT utført, som gir likning 13,

$$\begin{aligned} X[k, l] &= \sum_{n=0}^{N-1} x_l[n] e^{-i2\pi nk/K} \\ &= \sum_{n=0}^{N-1} w[n] x[n + lL] e^{-i2\pi nk/K} \end{aligned} \quad (13)$$

hvor K er DFT størrelsen og k er en frekvensindeks. STFTen $X[k, l]$ preger da det lokale tids-frekvens oppførsel av signalet rundt tid lL og frekvensindeks k for en samplingshastighet f_s , disse

diskrete indeksene tilsvarer kontinuerlig tid lL/f_s og frekvens kf_s/K . For å forenkle notasjonen, er radiell frekvens,

$$\omega_k = \frac{2\pi k}{K}$$

ofte inkorporert, slik at STFT-uttrykket blir som vist i ligning 14.

$$X[k, l] = \sum_{n=0}^{N-1} w[n]x[n + lL]e^{-i\omega_k n} \quad (14)$$

Noter at for en hopplengde på $L = 1$, kan STFT kan bli skrevet direkte som en funksjon av den originale tidsvariabelen t , vist i ligning 15.

$$X[k, l] = \sum_{n=0}^{N-1} w[n]x[n + t]e^{-i\omega_k n} \quad (15)$$

Invers STFT (ISTFT) er kalkulert ved å ta invers av hver DFT vektor av STFT, hvor overlappen i STFT kalkulasjonen er tatt hensyn til.

2.1.5 Digital filtrering

Digital filtrering i denne sammenhengen er en prosess som opptrer i frekvensdomenet. Med $x(t)$ som signalet som skal filtreres og $y(t)$ det filtrerte signalet, der deres FTer er relatert til hverandre som vist i ligning 16,

$$Y(\omega) = H(\omega)X(\omega) \quad (16)$$

hvor $H(\omega)$ er frekvensresponsen av filteret. Et eksempel på et slikt filter er et Butterworth-lavpass filter hvor frekvensresponsen er som vist i ligning 17,

$$|H(\omega)_{lp}|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \quad (17)$$

Frekvensresponsen for et Butterworth høypass filter er vist i ligning 18.

$$|H(\omega)_{hp}|^2 = \frac{\omega^{2N}}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \quad (18)$$

Et båndpassfilter kan lages ved å ha et lavpass og et høypass filter i kaskade, som vist i ligning 19,

$$x[n] \longrightarrow h_{lp}[n] * h_{hp}[n] \longrightarrow y[n] \quad (19)$$

hvor $x[n]$ er det diskrete signalet av $x(t)$, $y[n]$, er det diskrete filtrerte signalet og $h_{lp}[n]$ og $h_{hp}[n]$ er henholdsvis overføringsfunksjonene til lavpass og høypass filtrene.

Et annet type filter er Savitzky-Golay, det er minste kvadraters digitale polynomfiltre brukt for å glatte (eng. smooth) signaler. Idéen bak filteret er å ta for seg en gruppe av $2M + 1$ samples sentrert rundt $n = 0$ av en sekvens med samples $x[n]$ av et diskret signal, og finne koeffisientene av et polynom som minimerer gjennomsnittlig kvadratisk tilnærmingsfeil. Ligning 20 beskriver utregningen av polynomene og ligning 21 er gjennomsnittlig kvadratisk tilnærmingsfeil.

$$p(n) = \sum_{k=0}^N a_k n^k \quad (20)$$

$$\epsilon_N = \sum_{n=-M}^M \left(\sum_{k=0}^N a_k n^k - x[n] \right) \quad (21)$$

Hvor polynomet er av N te orden og a_k er polynomkoeffisientene for hver orden k opp til $k = N$.

Artifakter kan oppstå ved bruk av glatting filtre som Savitzky-Golay. En typisk artefakt er veldig glatte topper som kommer av sterk ulineært i det ufiltrerte signalet. Dette kan unngås ved å øke antall samples i hver gruppe (vindu), eller minke orden N .

For mer informasjon om Savitzky-Golay filtre, se Savitzky et.al [11].

2.1.6 Mel-frekvens cepstrale koeffisienter

Kommende del seksjon er basert på og inspirert av *Homomorphic Systems and Cepstrum Analysis of Speech* i *Springer Handbook of Speech Processing* av Ronald W. Schafer [12].

Motivasjonen bak formuleringen av Mel-frekvens cepstrale koeffisienter (eng. Mel-frequency cepstral coefficients, MFCC) er å finne en ny cepstrum representasjon som kan imitere menneskelig oppfatning av lyd. I de fleste implementasjonene er en kort tid Fourier (eng. Short Time Fourier, STF) analyse gjort først, resulterende i en DFT $X_m[k]$ for den m -te rammen. Da er DFT verdiene gruppert sammen i kritiske bånd og vektet med triangulære vektfunksjoner. Mel-spektrum av den m -te rammen definert for $r = 1, 2, \dots, R$ som vist i ligning 22,

$$MF_m[r] = \frac{1}{A_r} \sum_{k=L_r}^{U_r} |V_r[k]X_m[k]|^2 \quad (22)$$

hvor $V_r[k]$ er vektfunksjonen for det r -te filteret i intervallet fra DFT indeks L_r til U_r . A_r er definert som vist i ligning 23,

$$A_r = \frac{1}{A_r} \sum_{k=L_r}^{U_r} |V_r[k]|^2 \quad (23)$$

og er en normaliseringsfaktor innebygd i vektfunksjonen. For hver ramme, er en diskre cosinus transformasjon (eng. Discrete Cosine Transform, DCT) av \log til magnituden av mel-filter utgangen kalkulert for å forme funksjonen $mfcc[n]$ som vist i 24.

$$mfcc[n] = \frac{1}{R} \sum_{r=1}^R \log(MF_m[r]) \cos \left[\frac{2\pi}{R} \left(r + \frac{1}{2} \right) n \right] \quad (24)$$

Typisk, $mfcc[n]$ er evaluert for et antall av koeffisientene N_{mfcc} som er mindre enn antall mel-filtre.

2.1.7 Normalisering og standardisering

Visse maskinlæringsmetoder er sensitive for store variasjoner i et datasett, for eksempel algoritmer basert på Gradient Descent, slik som nevrale nettverk og regresjon. Distanse baserte algoritmer som K-means og støtvektormaskin (Support Vector Machine, SVM) blir særlig påvirket av dataens spann. Det kommer av at algoritmene bruker distanse mellom data punktene for å sjekke og lære likheter [13].

Normalisering er en skaleringssteknikk hvor verdier skiftes og reskaleres slik at de er innenfor et intervall som har et mindre spann enn utgangspunktet. Formel 25 skalerer data til et sett av

verdier i området $[a, b]$,

$$x' = a + \frac{(x - \min(x))(b - a)}{\max(x) - \min(x)} \quad (25)$$

hvor x' er den oppdaterte vektoren, x er den originale vektoren, og a og b henholdsvis er minimum og maksimum verdiene valgt.

Standardisering av et datasett vil fjerne gjennomsnittet og oppnå enhetsavvik, dvs. standardavvik lik 1. Dette gjøres ved å regne ut gjennomsnittet og standardavviket av hvert trekk (eng. feature) for hele datasettet, deretter oppdatere hver verdi i datasettet med likning 26,

$$x' = \frac{x - \hat{x}}{\sigma} \quad (26)$$

hvor \hat{x} er gjennomsnittet for den originale vektoren og σ er standardavviket.

Standardisering er ofte brukt i algoritmer som SVM og kunstig nevralt nettverk (eng. Artificial Neural Network, ANN).

2.1.8 Prinsippal komponent analyse

Teksten under er inspirert av del seksjonen *Principal Component Analysis* fra teoridelen i prosjektoppgaven [14].

Prinsippal komponent analyse, (eng. Principal Component Analysis, PCA) transformerer et stort datasett til et mindre, samtidig som viktig informasjon blir bevart [15]. Med andre ord er en PCA transformasjon en metode brukt for dimensjonalitetsreduksjon. Det er en statistisk basert metode hvor informasjonen er verdsatt med hensyn til variasjon.

Hver variabel i det nye datasettet er en lineær kombinasjon av variablene i det originale settet, også visst som prinsipiell komponent (eng. Principal Component, PC). PCene er ortogonale til hverandre, som betyr det er ingen overflødig informasjon. Man kan velge hvor mange komponenten man vil ha, dette bestemmer dimensjonen av det nye datasettet.

Første steget av PCA transformasjonen er å kalkulere gjennomsnittet for hver dimensjon (kolonne) d av en gitt matrise, \mathbf{A} , med dimensjon [16]. Matrisen $\bar{\mathbf{A}}$ er så en vektor som inneholder gjennomsnittene for hver dimensjon i matrise \mathbf{A} , som sett i ligning 27,

$$\bar{\mathbf{A}} = [g\text{jennomsnitt}(\mathbf{A}_1), g\text{jennomsnitt}(\mathbf{A}_2), \dots, g\text{jennomsnitt}(\mathbf{A}_d)] \quad (27)$$

Neste steg er å regne ut kovariansen, som da blir kalkulert med ligningen 28,

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{x})(Y_i - \bar{y}) \quad (28)$$

hvor *cov* står for kovarians (covariance), variablene *X* og *Y* representerer kolonnene, *i* er antall av verdier inneholdt i kolonnene og \bar{x} og \bar{y} er gjennomsnittene av kolonnene kalkulert over. De ulike kombinasjonene av kovarians former en matrise, \mathbf{A}_{cov} , av *dxd* dimensjon. Fulgt av kalkulerings av egenvektorene, \mathbf{v} , og deres egenverdier, λ . Som er funnet når $\mathbf{A}_{cov}\mathbf{v} = \lambda\mathbf{v}$ er tilfredsstillt. Eigenvektorene er så sortert etter minkende egenverdier, som fører til steget hvor en kan velge *k* antall egenvektorer med de største egenverdiene. Dette resulterer til en *dxd* dimensjons matrise, \mathbf{W} .

2.2 Maskinlæring

Maskinlæring er en gren av kunstig intelligens. Konseptet for maskinlæring er å bruke digital data for å lære maskiner å gjenkjenne mønster. Ved tradisjonell programmering defineres regler på forhånd, i motsetning til maskinlæringsalgoritmer som skal kunne finne regler basert på inndata. Datasettet brukt i maskinlæring er ofte delt inn i et treningssett som blir brukt til å trene maskinlæringsmodeller, og et testsett for å teste modellen. Fordelingen på datasettet er ofte 80% trening og 20% test

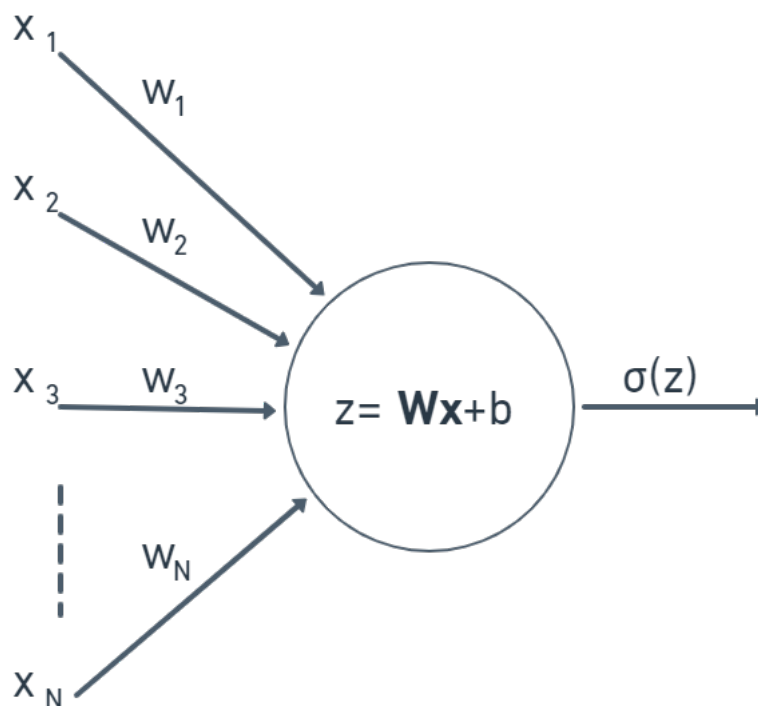
Maskinlæring kan videre deles inn i kategoriene veiledet trening (eng. supervised learning), ikke-veiledet trening (eng. unsupervised learning) og forsterket læring (eng. reinforcement learning). Denne oppgaven tar kun for seg veiledet trening. Slik trening er avhengig av å ha et merket (eng. labeled) treningssett, som vil si at enhver datasample i settet har en kjent etikett. Disse etikettene kalles som oftest klasser. Tilfellet der modellen skal lære seg å skille mellom klasser, kalles det klassifisering. Prediksjon er når man for eksempel bruker en klassifiseringsmodell for å forutsi et utfall.

2.2.1 Dyp læring

Teksten under er inspirert av del seksjonen *Artificial Neural Networks* fra teorien i prosjektoppgaven [14].

Et kunstig nevralt nettverk (eng. Artificial Neural Networks, ANN) er enkelt definert som en simulasjon av menneskets biologiske nevralt prosess. På lik linje som vårt sinn, klarer man med ANN å trene maskinen til å gjenkjenne mønstre og symboler i blant annet bilder.

Begrepet dyp læring kommer av at de kunstige nettverkene, på lik linje som hjernen, bearbeider informasjon i flere lag. Hvert lag inneholder noder som representerer biologiske nevroner, koblingen mellom de kunstige nodene representerer aksoner og dendritter i hjernen. Disse nodene mottar og sender signaler til hverandre, slik nervecellene i hjernen opererer med dendritter og aksoner. Koblingsvektene i ANN representerer synapser og aktivitet i cellekroppen er tilnærmet med aktiverings funksjoner i modellen [17]. Figur 1 illustrerer en nodes funksjon i en ANN, den bruker en vektor av input $\mathbf{x} = [x_1, x_2, \dots, x_N]$, deres vekt matrise \mathbf{W} som inneholder vektene w_1, w_2, \dots, w_N av en skjevhet (eng. bias) b for å kalkulere deres input vekt $z = \mathbf{W}\mathbf{x} + b$.



Figur (1) Illustrasjon av et kunstig nevron og dens funksjon. Figur fra [14] s. 9.

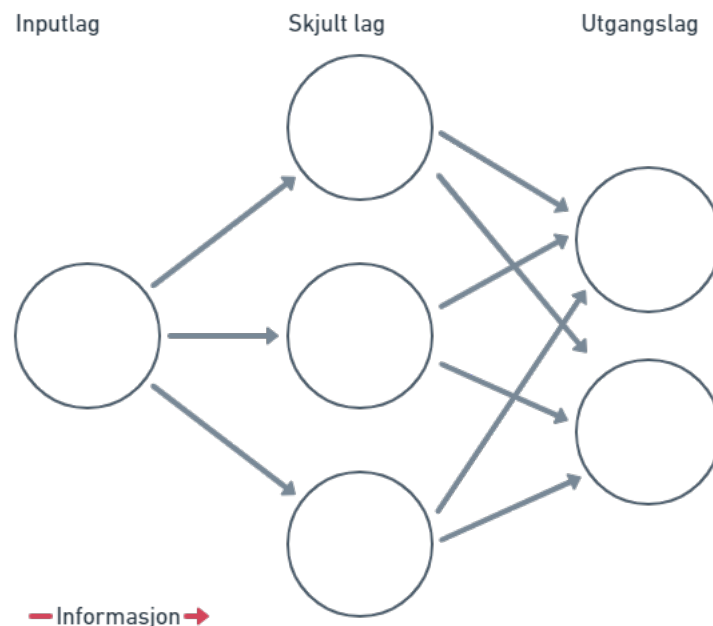
Nodens utgang (eng. output) er kalkulert med en aktiveringsfunksjon σ . Ulike aktiverings-

funksjoner serverer ulike outputs, for eksempel en Sigmoid og Leaky Linear Unil (ReLU) vist nedenfor,

- Sigmoid: gir ut en verdi fra 0 til 1 med funksjonen $f(x) = \frac{1}{e^{-x}}$
- Leaky ReLU: gir ut en verdi med $f(x) = \begin{cases} \alpha x, & \text{for } x < 0, \text{ hvor } \alpha \text{ er en verdi n\ae} 0 \\ x, & \text{for } x \geq 0 \end{cases}$

En flerlags perceptron (eng. Multi-layer Perceptron, MLP) er en type ANN. Nettverket har et input lag, et utgangslag, og kan inneholde ett eller flere skjulte lag mellom. Lagene består av nevroner fungerende på samme måte som beskrevet ovenfor. Utgangslaget og hvert skjulte lag kan ha deres egen aktiverings funksjon, og antall nevroner per lag er valgfritt.

Et *Feed Forward* nettverk har informasjon gående én retning, fra input til utgang. Det er ingen tilbakemeldings (eng. feedback) koblinger i nettverket. Begrepet *Fully Connected* beskriver at koblingene fra hvert nevron i ett lag, er koblet til alle nevroner i neste lag. En sammensetning av disse nettverksfunksjonene er kalt et *Feed Forward Fully Connected* nettverk, et eksempel på en arkitektur for et slikt MLP nettverk er vist i figur 2.



Figur (2) Illustrasjon av et enkelt skjult ettlags Feed Forward Fully Connected MLP med et innptlag og et utgangslag. Figur fra [14] s. 10.

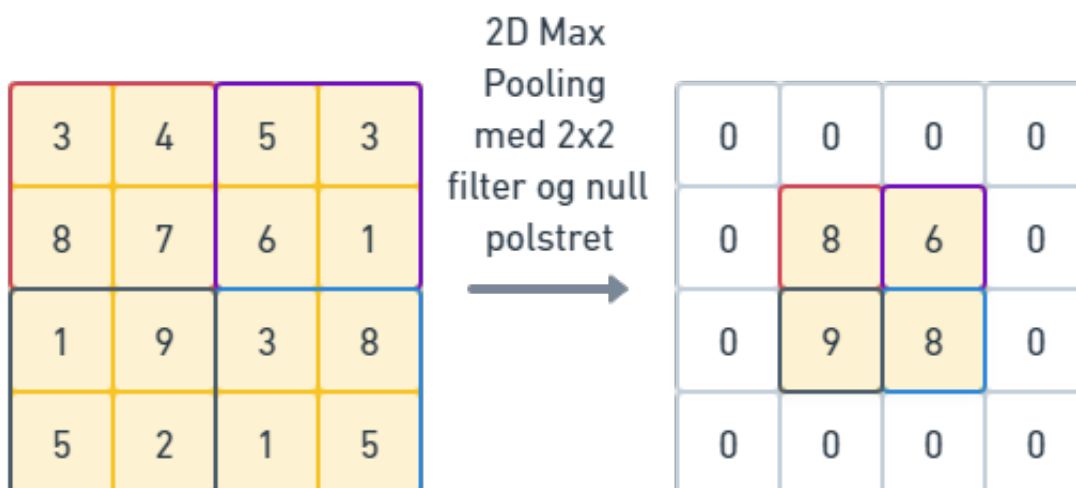
Et ANN er et spesifikt type Feed Forward MLP, og er ofte brukt for bildeanalyse. Et CNN

er i stand til å velge ut eller detektere mønster, hvor basisen av et CNN er konvolusjonelle lag. Transformasjonen av informasjon mellom lagene er en konvolusjonell operasjon, i motsetning til en vanlig MLP.

For hvert konvolusjonelle lag må antall filtre bli spesifisert, hvor filteret er verktøyet brukt for å detektere mønster i bildet. Mønster som blir detektert i starten er noe enkelt som en kant, en linje eller et hjørne. Jo dypere man går i nettverket, vil filtrene kunne detektere spesifikke symboler eller objekter som fugler, hunder eller katter. Et filter glir over input bildet, hvor glidingen er prosessen som refereres til som konvolverende.

Videre vil vi ta utgangspunkt i et 2D CNN. La oss anta S antall filtre (eng. kernels), hver av størrelsen $N \times M$ med tilfeldige tall, og et input bilde. Filtrene vil gli over hver $N \times M$ piksler på bildet, deretter lagres dot produktet av piksel og filter verdiene som én verdi i et nytt bilde. Denne prosessen produserer S antall utganger. Disse utgangene vil så være inputs i neste lag, og sånn vil den samme prosedyren gjentas like mange ganger som nettverket har konvolusjonelle lag.

CNNer inkluderer ofte en prosess kalt Max Pooling, og brukes etter et konvolusjonelt lag. Man trenger å definere antall ønskede Max Pooling filtre og deres størrelse. For eksempel vil S antall $N \times M$ filtre gi ut S bilder, hvor hver pikselverdi er maksverdien av $N \times M$ pikslene filteret tar for seg når den glir over input bildet. Et eksempel på Max Pooling med et 2×2 filter med null polstring (eng. zero padding), kan sees i figur 3.



Figur (3) Max Pooling med et 2×2 filter, inkludert null polstring på utgangen. Figur fra [14] s.12.

Polstring kan bli brukt for hvert lag, konvolusjon eller Pooling, for å unngå endring i romslige dimensjoner av input bildet.

Tilbakeforplantning

Tilbakeforplantning (eng. Backpropagation, BP) algoritme oppdaterer trenbare parametere av ANNen ved å kalkulere gradienter av en gitt tapsfunksjon (eng. Loss function), L , med hensyn til parameterne. De trenbare parametrene er vektene og skjevhetene nevnt tidligere. Tapsfunksjonen kan for eksempel være en *Binary Crossentropy* typisk brukt for binære klassifiseringsproblemer, eller *Categorical Crossentropy* for et klassifiseringsproblem med flere enn to klasser. Gradient nedstigning (eng. Gradient descent) er når parameterne er justert i negativ retning, for å videre minimere tapsfunksjonen. En generell ligning for gradient nedstigning til vektene w er vist i likning 29, og for skjevheten b i 30. α er lærings raten som kontrollerer trinnstørrelsen når gradient nedstigningen går mot et ideelt minimum punkt.

$$w_{oppdatert} = w - \alpha \frac{\delta L}{\delta w} \quad (29)$$

$$b_{oppdatert} = b - \alpha \frac{\delta L}{\delta b} \quad (30)$$

Vektene og skjevhetene er kalkulert for hvert lag i nettverket, gående fremover. Ved enden av nettverket blir tapsfunksjonen kalkulert. Formålet av BP er å minimere denne funksjonen ved å kalkulere gradienter og bruke de for å oppdatere hver trenbar parameter, gradient nedstigning, gående bakover i nettverket.

Overtrening

Det er ingen siste avgjørelse på hvordan et nettverks arkitektur av en ANN-modell burde være, alt avhenger av tilgjengelig data og bruken av modellen. Det er modell spesifikasjoner kalt hyperparametere som kan justeres underveis for å finne den beste modellen for den gitte oppgaven. Noen eksempler av ulike hyperparametere med en enkel forklaring vises i tabell 1.

Epoke (eng. epoch)	Antall ganger modellen jobber gjennom treningssettet
Gruppestørrelse (eng. batch size)	Antall samples å jobbe gjennom før justering av de trenbare parameterne
Skjulte lag	Antall skjulte lag i nettverket
Nevroner per lag	Antall nevroner per lag

Tabell (1) Hyperparametere i ANN, og en kort forklaring.

ANNer er utsatt for overtrening under treningsprosessen. Det forklarer situasjonen når modellen lærer treningsdataen for godt, og mister generalisering. Modellen kan begynne å trene på støyete data, og vil prestere dårlig ved prediksjon av usett data. Dette kan oppstå når nettverket består av for mange skjulte lag, de flere lagene gjør det enklere for nettverket å huske treningssettet. En annen begrensing som kan lede til overtrening er å ha et lite datasett sammenlignet med antall lag i nettverket.

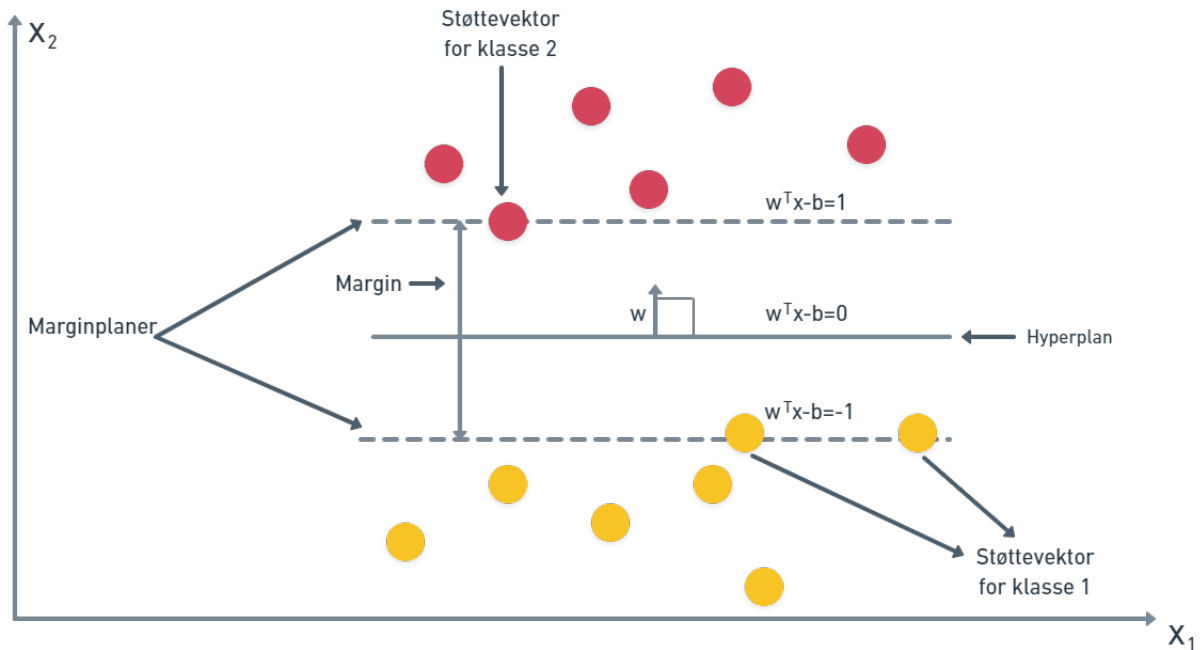
Overtrening kan bli detektert med bruken av læringskurver under trening av en ANN. Læringskurvene er oppnådd ved bruk av et valideringssett i tillegg til treningssettet. Ved å måle trenings og validerings tapet under trening, og analysering av tapet over tid (epoker), vil det være klart om overtrening oppstår. Valideringstapet vil øke mens treningstapet minker. Ved å stoppe treningen før valideringstapet begynner å øke, kan modellen forhindre fra overtrening [18].

2.2.2 Støttevektormaskin

Teksten under er inspirert av del seksjonen *Support Vector Machine* fra teorien i prosjektoppgaven [14].

SVM er en veiledet læringsalgoritme brukt for klassifisering og regresjon [19]. Hvert objekt utsatt for klassifisering er representert som et punkt i et N dimensjonalt rom hvor koordinatene av punktet er dens egenskaper (eng. features). Klassifisering oppgaven er utført ved å opprette et hyperplan som separerer klassene. Hyperplanet er en linje i 2D og et plan i 3D, med andre ord har hyperplanet $N - 1$ dimensjoner. SVM prøver å finne hyperplanet som separerer klassene på best mulig vis. Den maksimerer distansen fra punktene i hver kategori. Denne distansen kalles for en margin, og punktene som ligger på marginplanet er støttevektorene. For å finne dette hyperplanet trengs et datasett hvor hvert objekt og dens features er merket med dens kjente klasse.

Eksempelet i figur 4 illustrerer et 2D lineært separabelt binært klassifiseringsproblem. Dette er videre brukt for å forklare SVM matematisk.



Figur (4) 2D lineær separabel binær klassifikasjon med SVM. Figur fra [14] s. 7.

Som sett i figur 4, er hyperplanet definert med likning 31,

$$\mathbf{w}^T \cdot \mathbf{x} - b = 0 \quad (31)$$

hvor \mathbf{x} er en støttevektor som et punkt i vektorrommet, \mathbf{w} er en vektor og b er en konstant, alle er i vektorrommet R^N , hvor $N = 2$ i dette tilfellet. Støttevektorene på hver side av hyperplanet tilfredsstiller hver av deres likning. Støttevektorer på marginen for klasse 1 tilfredsstiller likning 32, og støttevektorene på motsatt margin tilfredsstiller likning 33,

$$\mathbf{w}^T \cdot \mathbf{x} - b = -1 \quad (32)$$

$$\mathbf{w}^T \cdot \mathbf{x} - b = 1 \quad (33)$$

Ved å kombinere likning 33 og 32, er beslutningsregelen som vist i likning 34 oppnådd. som nevnt tidligere, er det et ønske om å maksimere margin distansen, likning 35, mens den oppfyller

beslutningsregelen i likning 34,

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) - 1 \geq 0, \quad i = 1, 2, \dots, N \quad (34)$$

$$M = \frac{2}{\|\mathbf{w}\|^2} \quad (35)$$

hvor y_i representerer kategorien for hvert punkt \mathbf{x}_i og M er marginen. Dette leder til likning 36,

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2}, \quad \text{slik at, } y_i(\mathbf{w}^T \cdot \mathbf{x}_i - b) - 1 \geq 0 \quad (36)$$

Denne løsningen er basert på en hard margin SVM, hvor modellen ikke tillater feilklassifisering. En myk margin (eng. soft margin) introduserer slake (eng. slack) variabler ξ_i , $i = 1, 2, 3 \dots N$, som gir muligheten til å løse et lineært ikke separabelt problem ved å tillate noen feilklassifiseringer. ξ_i er relatert til et datapunkt \mathbf{x}_i , hvor ξ_i representerer datapunktets distanse til dens marginplan. Hvis datapunktet er plassert på rett side av hyperplanet, er ingen distanse målt, $\xi_i=0$. Optimaliseringsproblemet blir så endret til likningen vist i 37,

$$\min_{\mathbf{w}} \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^N \xi_i, \quad \text{slik at, } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i \quad (37)$$

hvor C er kostparameteren.

Til nå har vi kun tatt hensyn til forekomst av en lineær separasjon, noe som ikke alltid er den beste løsningen. I tilfeller der det ikke passer med en slik separasjon, kan blant annet Kernels som ikke lineær, polynom eller radiell basisfunksjon (eng. Radial Basis function, RBF) bli brukt.

2.3 Ytelsesberegninger

Teksten under er sterkt inspirert av del seksjonen *Performance Metrics* fra teorien i prosjektoppgaven [14].

Det er fire mulige utfall ved en binær klassifisering, vist i listen under,

- Sann positiv (SP): positiv sample klassifisert rett
- Sann negativ (SN): negativ sample klassifisert riktig
- Falsk negativ (FN): positiv sample klassifisert feil
- Falsk positiv (FP): negativ sample klassifisert feil

hvor positive samples er av den ene klassen og negative samples er av den andre klassen.

Presisjon (eng. Precision) representerer raten av sanne prediksjoner i relasjon til total antall prediksjoner og er kalkulert med ligning 38. Tilbakekall (eng. Recall) representerer raten av sanne prediksjoner i forhold til totalt antall positive samples og er kalkulert med ligning 39.

$$\text{Presisjon} = \frac{SP}{SP + FP} \quad (38)$$

$$\text{Tilbakekall} = \frac{SP}{SP + FN} \quad (39)$$

Ved å kalkulere harmonisk gjennomsnitt av presisjon og tilbakekall får man målingen *F1-score*. Ligning 40 viser utregningen av *F1-score*.

$$F1 - score = \frac{2 \cdot \text{Presisjon} \cdot \text{Tilbakekall}}{\text{Presisjon} + \text{Tilbakekall}} \quad (40)$$

Total nøyaktighet (eng. Accuracy) av modellen er kalkulert med ligning 41.

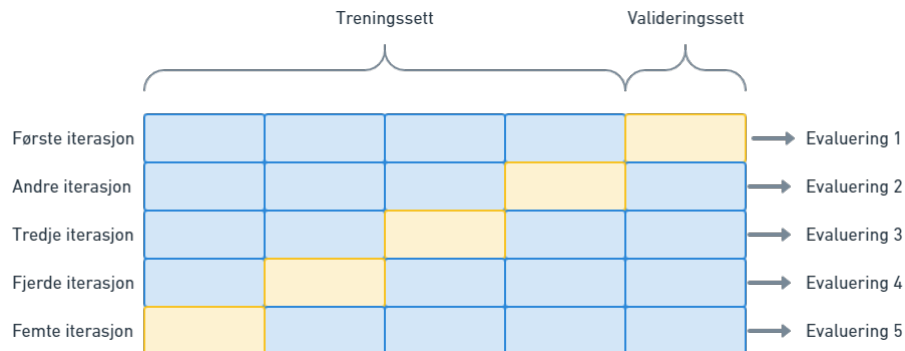
$$\text{Nøyaktighet} = \frac{SP + SN}{SP + FP + SN + FN} \quad (41)$$

Nøyaktighetsmålingene er mest gyldige for et balansert datasett.

2.4 k -fold kryssvalidering

Teksten under er sterkt inspirert av del seksjonen *k-Fold Cross-Validation* fra teorien i prosjekt-oppgaven [14].

k -fold kryssvalidering er en metode brukt for å evaluere en klassifiseringsmodell. Metoden deler treningssettet inn i k folder og bruker $k - 1$ folder for trening og én fold for validering. Et eksempel med $k = 5$ folder er illustrert i figur 5.



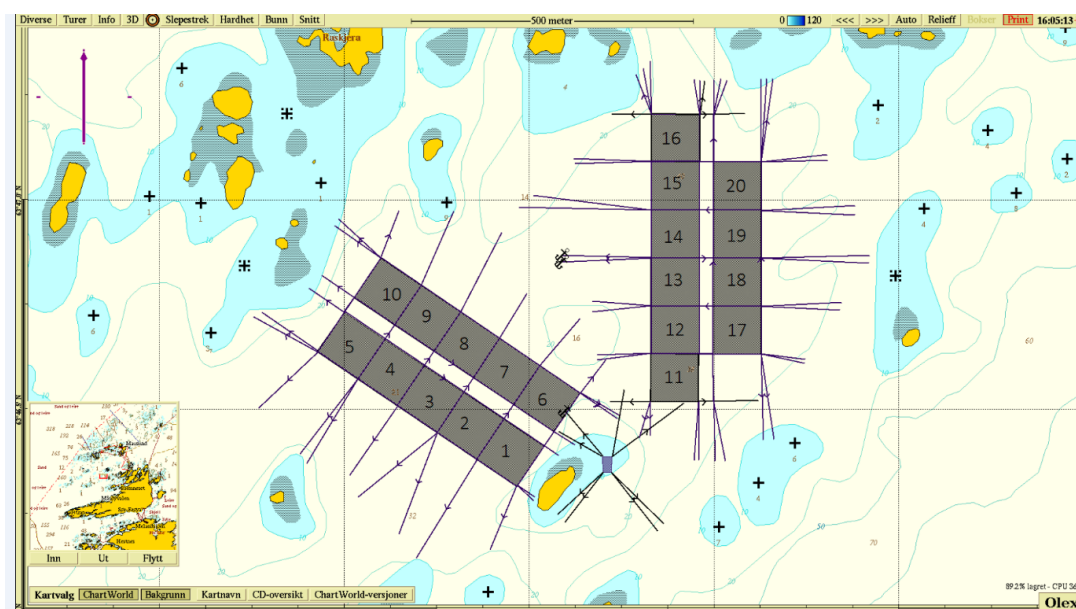
Figur (5) 5-folds kryssvalidering. Fire folder for trening og én fold for validering for hver iterasjon. Sender ut ytelsesberegninger for hver iterasjon. Figur fra [14] s. 15.

Hele treningssettet er inndelt i fem folder, hvor fire av de er brukt for trening og én fold for validering for hver iterasjon. For hver iterasjon endres trenings og validerings delene og modellen evalueres med ytelsesberegninger. Gjennomsnittet av ytelsesberegningene er kalkulert til slutt og gir en generell vurdering av modellen. Denne metoden reduserer sjansen for overtrening og gir en mer nøyaktig evaluering av treningen.

3 Metode

3.1 Metode for innsamlet data

Dataen brukt i oppgaven er gitt av SINTEF Ocean. De har utført opptak av ulike merder på fiskeanleggene Rataren I og Rataren II på Frøya, Trøndelag. Til sammen er det 20 merder fordelt på begge anleggene. Hver merd er merket med et nummer, der merd 1-10 er i Rataren I og merd 11-20 er i Rataren II. Oppgaven tar kun for seg data fra merd 3 og 7 fra Rataren I. Merd 3 hadde ca. 196 000 fisk etter avlusing med en snittvekt på 3,6 kg. Merd 7 hadde ca. 177 000 fisk med snittvekt på 3,0 kg. Dette er data fra 1.12.2021, snittvekten har nok økt siden. Figur 6 viser anleggene og fordeling av merdene og deres merking med tall fra 1 til 20.



Figur (6) Rataren I, merd 1-10 og Rataren II, merd 11-20.

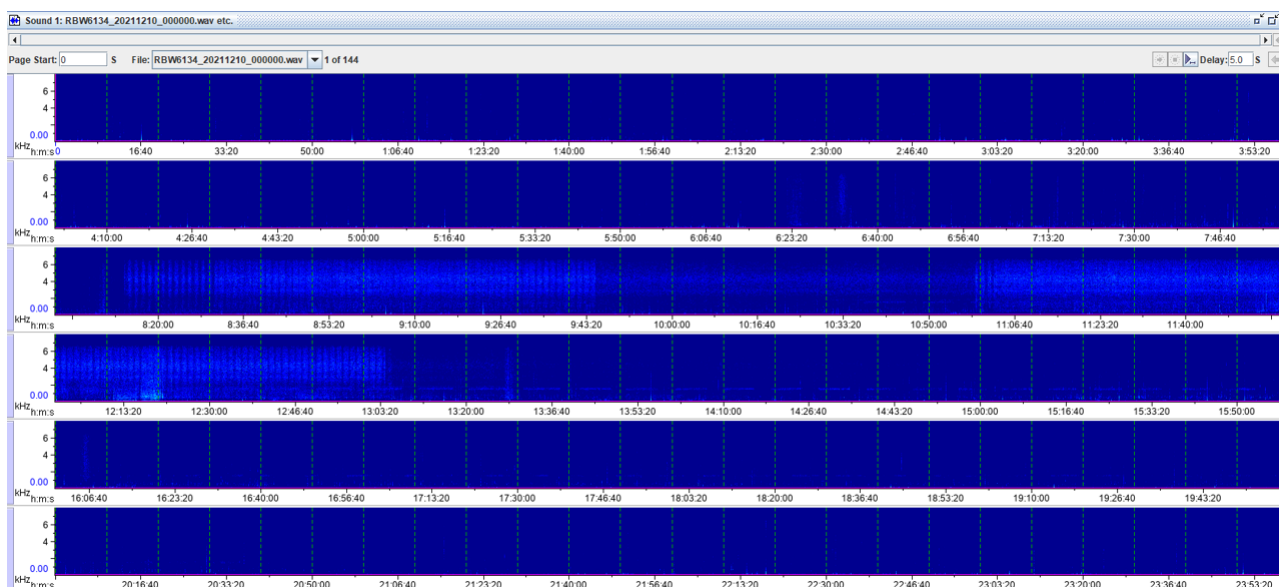
Dataen er i utgangspunktet innsamlet for to ulike prosjektet, SoundWell II finansiert av Forskningsrådet og OWITOOLS finansiert av Fiskeri- og havbruksnæringens forskningsfinansiering (FHF) [3] [4]. SoundWell II fokuserer på overvåking av lyd fra fisk på et generelt plan, hvor OWITOOLS er rettet direkte mot håndtering og avlusing.

Hydrofonen brukt for opptak er icListen HF fra OceanSonic, RB9-ETH 25-bit Smart Hydrophone 200kHz for begge merder. Det er kontinuerlig opptak i merdene, hvor hver lydfil er lagret som en .wav fil og er på 10 min med samplingsfrekvens på 16kHz. Med lydopptakene fra hydrofonene følger det skriftlige logger fra dagene. De kan brukes for å merke data. SINTEF Ocean har i

tillegg presentert lyden i form av et spektrogram. Det kan være nyttig å kunne se lyden som et lydbilde for analyse. Det er tidssparende å se på et spektrogram og gjenkjenne mønster fra en lyd dag, enn å høre på 24 timers lydopptak.

Formålet med en lydmiljøindikator er å ha oversikt over lyd for enhver aktivitet som oppstår i anlegget. Valget falt på de tre klassene, fôring, natt og støy. Fôring, fordi det var en operasjon som opptrådte hver dag og er lett gjenkjennelig på spektrogrammene. Det var ønsket av SINTEF Ocean å se om det var mulig å skille mellom natt og dag, og valget falt på kun å ha en klasse for natt. Fôringen varte løpet av hele dagen, noe som kan føre til at modellene trener på lydene fra operasjonen og ikke klassen dag. For å unngå å kun lage en lydmåler, ble det valgt å ha en støy klasse som er all annen lyd enn den kjente prosessen fôring. Videre var det ønsket å overvåke laksens aktivitet løpet av dagen for å se sammenheng mellom aktivitet og lydmiljøet.

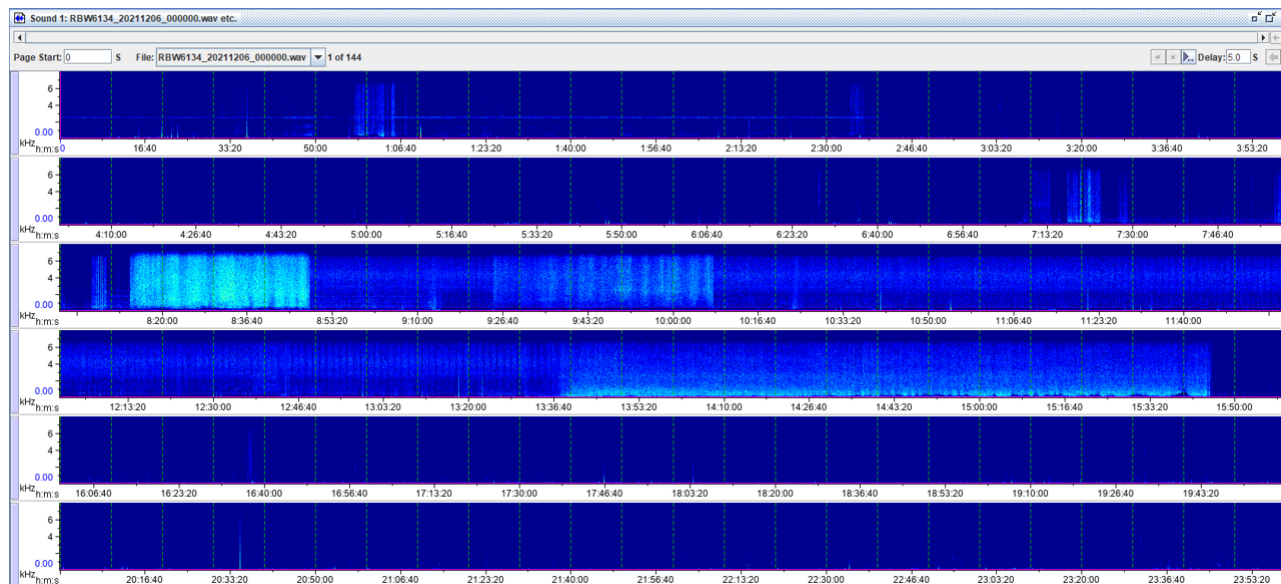
Figur 7 er et eksempel på et spektrogram av en relativt lite støyet dag fra merd 3 den 10. desember 2021. Fra ca. 08:20-13:03 er det en lyd som opptrer i frekvensområdet 0-6k Hz. Ut ifra loggen fra oppdrettsanlegget og andre lite støyet dager, viser det seg at dette er typisk lydbilde for fôring. Etter 10:33 opptrer det en gjentakende støy frem til 15:50, ellers er dagen relativt stille.



Figur (7) Spektrogram av lydopptak fra 10.12.2021, merd 3, Rataren I. Lyddag med lite støy.

Utifra observasjoner fra anlegget ifølge SINTEF Ocean, er laksen mer aktiv rett før fôring. Figur 7 viser tegn på lyder ved lave frekvenser under 1k Hz, rett før fôring. Det kan dermed antas at lyden som opptrer på spektrogrammet er lyd av aktiv laks. Ved lytting til lydfiler ved denne tiden viser det seg at det er hovedsakelig plaskelyder.

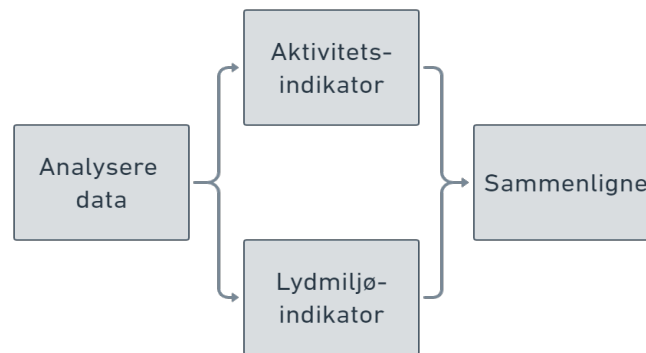
Figur 8 er et eksempel på en støyet dag. Fra midnatt til ca. 02:40 er det en lyd på ca. 2,5k Hz. Ifølge loggen er dette en lyd fra en generator. Rundt 08:20 er det støy, og føringen er ikke tydelig i spektrogrammet, men det er antatt at det varer fra ca. 08:30 til ca. 13:30. Det er støy frem til 15:50, ellers er det ganske stille.



Figur (8) Spektrogram av lydopptak fra 06.12.2021, merd 3, Rataren I. Lyddag med mye støy.

3.2 Overordnet metode

Figur 9 visualiserer den overordnede metoden brukt i oppgaven. Oppgaven er todelt etter *Analyse data*, den ene delen tar for seg hele lydbildet med et formål om å lage en *Lydmiljøindikator*, den andre delen er en passiv akustisk *Aktivitetsindikator*. Formålet med todelingen er å kunne se om det er en korrelasjon mellom detektert lydmiljø og prediktert aktivitet.



Figur (9) Overordnet metode.

For begge indikatorene er det en felles grunnleggende metode som er brukt:

1. Segmentere data
2. Skalering
3. Filtrering og resampling
4. Valg av lydrepresentasjon
5. Valg av modell
6. Trene
7. Teste

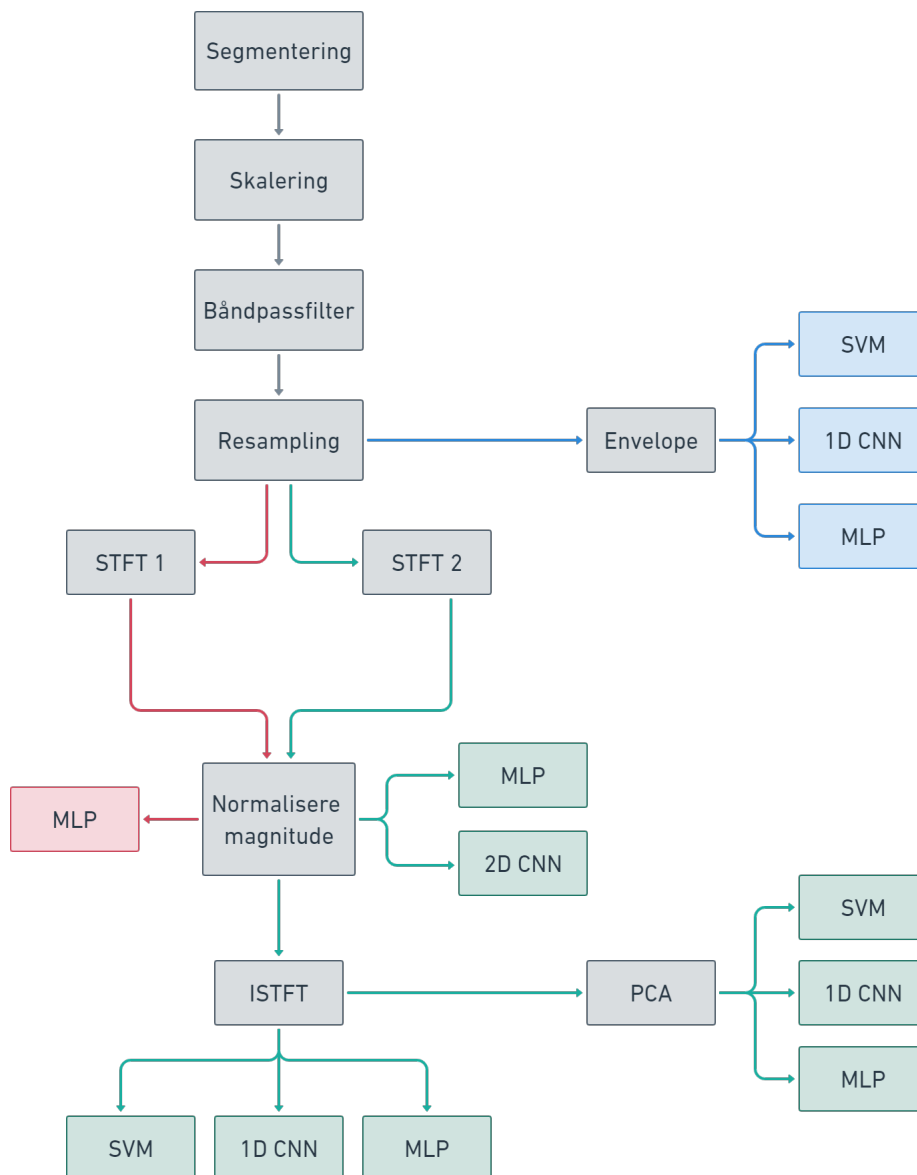
Den grunnleggende metoden vist i listen ovenfor kan programmeres i Python, Matlab o.l. I dette tilfellet er det brukt Python (versjon 3.8.8) fremfor andre programmeringsspråk grunnet tilgjengelige preprosesserings og maskinlærings biblioteker som scikit-learn og keras [20] [21] [22]. JupyterLab (versjon 3.2.1) ble brukt som grensesnitt [23]. Datamaskinen brukt er bærbar hp elitebook 840 g7-Core i7-16 GB RAM-475 GB SSD, lånt av SINTEF Ocean. Ved prosessering oppgaver der den bærbare datamaskinen ikke hadde nok kapasitet til å prosessere, ble en datamaskin satt opp med Ubuntu 18 og hadde GeForce 1080 brukt.

Et blokkdiagram av metoden for lydmiljøindikatoren kan sees i figur 10.



Figur (10) Metode for lydmiljøindikatoren.

Et blokkdiagram av metoden for aktivitetsindikatoren kan sees i figur 11



Figur (11) Metode for aktivitetsindikatoren.

Videre vil begge metoder og valg av blokker begrunnes og forklares nærmere.

3.3 Analysere data

Som sett i figur 9 er starten på metoden å inspisere og analysere data. Det innebærer å sammenligne spektrogram av ulike lyddager med logg fra samme dag, for å gjenkjenne mønster med et formål om å merke dataen. Dataen merkes med fôring, natt, dag og støy, i tillegg til å dele dagene inn i støyfrie og støyfulle dager. Der støy er definert som all annen klar lyd sett på spektrogrammet, som ikke er fôring. Denne delen av metoden er uavhengig av oppgaveinndelingen og gjøres kun en gang.

Ved å opprette Excel ark for hver merd holder man oversikt over merkingen av data for hver dag. For forståelse og egen verifisering av dataen i forhold til loggen, ble det lyttet til et par lydfiler som inneholdt fôringslyder, støy og støyløse lyder. For lydmiljøindikatoren ble dagene delt inn i trening og test for å unngå data lekkasje på tvers av datasettene. Nettsiden *Time and date* ble brukt for å definere dag og natt i forhold til nautisk tussmørke på Frøya, Trøndelag [24]. Det ble valgt å ikke definere dagen i forhold til soloppgang og solnedgang fordi dagene er for korte i desember og at arbeidsdagen starter før soloppgangen som var rundt klokken 10:00. Videre er tidspunkt for fôring og støy definert ut ifra spektrogrammene, logger gitt av anleggene og lytting til et par av filene.

Med en antagelse om at laksen er mer aktiv rett før fôring enn rundt midnatt, kan dataen for aktivitetsindikatoren deles inn i to, høy aktivitet og lite aktivitet. I dette tilfellet er man kun interessert i ren lyd som enten inneholder høy aktivitet eller ikke, det er dermed valgt å ikke trene på støyete lyd. Analysen av spektrogrammene er da annerledes enn den er for lydmiljøindikatoren. Det er nå kun to tidspunkt av interesse, rundt 08:00 rett før fôring og ved midnatt. Treningssettet for aktivitetsindikatoren er annerledes enn for lydmiljøindikatoren, fordi de tar for seg ulike klasser.

Treningssettene er så og si balansert med ca. like mange segmenter fra hver klasse. Fulle dager valgt ut til testsettet, både støyfrie og støyfulle lyddager. Hver lyddag blir et eget testsett som preprosesserer separat.

3.3.1 Segmentering og skalering

Valg av tid per segment er avhengig av bruk. For lydmiljøindikatoren kan lengre lydsegmenter være av interesse. Da er det sikkert at fullstendige lyder er inkludert, og at lydene ikke er kuttet for

kort. I motsetning til aktivitetsindikatoren, der kortere segmenter kan være mer gunstig. Ved større lydfiler kan mønster forsvinne eller repeteres for ofte i en og samme fil. For lyd miljøindikatoren ble hver lydfil på 10 min segmentert til 10 filer på 1 min hver fil, for aktivitetsindikatoren var hvert segment på 10 s hver.

Samme normalisering og standardiserings metode kan brukes for begge indikatorene, men må gjøres separat fordi segmenteringen er ulik. En normalisering endrer alle verdier for å passe inn i et gitt intervall, for eksempel $[-1, 1]$ som er valgt i denne oppgaven. Normaliseringen skjer automatisk når man leser lydfilene med *librosa.load* funksjonen [25].

Standardisering gjøres ved å regne ut gjennomsnittsverdien og standardavvik av treningssettet. Deretter skalarers trenings og test-settet med disse verdiene ved å bruke ligning 26 fra teorien. Beregningen gjøres med funksjonen *sklearn.preprocessing.StandardScaler*, og datasettene blir transformert med funksjonen *.fit* [26].

3.4 Filtrering og resampling

Filtrering av frekvenser er et nyttig preprosesseringsverktøy hvis det er et gitt frekvensintervall man vil analysere. Alle frekvenser i lyd miljøet er av interesse, det er da ikke nødvendig med filtrering og resampling for den delen av oppgaven.

For aktivitetsindikatoren er det usikkert hvor den interessante informasjonen ligger. I det tilfellet kan det være greit å ta for seg ulike frekvensområder. Det er ønsket å isolere aktivitetslyder gitt av laksen, eksempler på dette kan være plasking og klikkelyder. Som nevnt i teorien er det forsket på at laks oppfatter lyder av lave frekvenser, det kan da være interessant å se på frekvenser under 1k Hz. For å unngå menneskeskapt lyder og lyder fra værforhold, er ikke frekvenser under 100Hz av interesse. For å ikke miste for mye informasjon kan det være gunstig å se på høyere frekvenser i tillegg, for eksempel et sted mellom 1k-8k Hz. Ut ifra spektrogrammene er det tydelig at det ligger informasjon i 500-4k Hz området. Det er dermed valgt å ta for seg to filtreringer [100-1k] Hz og [500-4k] Hz. Fordi det er valgt to filtreringer, er behandlingen av data fra båndpassfiltrering i figur 11 og ut, gjort to ganger.

Det er lagd et Butterworth båndpassfilter med funksjonen *signal.butter* fra biblioteket SciPy med tredje orden [27]. Signalet er så filtrert frem og tilbake med funksjonen *signal.sosfiltfilt* fra Scipy [28]. Deretter blir signalene nedsamlet med funksjonen *resample* fra Librosa biblioteket

[29].

3.5 Representasjon av lyd

Valgt representasjon av lyd er ulik for lydmiljø og aktivitetsindikatoren. For lydmiljøet er det inkludert lyder mennesker kan gjenkjenne, som båtlyder, menneskelig aktivitet og stillhet. Da kan lyden for eksempel representeres i form av MFCCer. Den originale samplingsraten på 16k Hz er brukt. Vinduslengde på kalkulering av MFCC er satt til 50 ms, hopplengden er på 25 ms og antall koeffisienter er 13. MFCC er kalkulert med funksjonen *feature.mfcc* fra biblioteket Librosa [30]. Det gir en dimensjon på (13, 241) for MFCC cepstrumet, som vil si at det er 241 samples per koeffisient.

Når det kommer til aktivitetsindikatoren, er det ikke like intuitivt for mennesker å gjenkjenne lyden. Det er i tillegg usikkert hvor den beste informasjonen ligger. I det tilfellet kan det være greit å teste ut flere datarepresentasjoner. Som sett i figur 11 er lyden representert på fem ulike måter, envelope, to ulike STFT spektrogrammer med normalisert magnitudo til 1, ISTFT av STFT 1 og PCA av ISTFT.

Det kalkuleres 1D DFT av reell data med *fft.rfft* funksjonen fra NumPy biblioteket, deretter kalkuleres envelope til hvert segment [31]. En envelope er en kurve som skisserer ekstremalene til et signal. Det tas STFT av signalene med funksjonen *signal.stft* fra Scipy [32]. Hamming vindu med vinduslengde på 100 ms og en hopplengde på 50 ms er brukt for STFT 2. Vinduslengde på 10 ms og 5 ms på hopplengde er valgt for STFT 1. Videre vil vi skille mellom representasjonene ved å referere til de som STFT 1 og STFT 2. Neste blokk på figur 11 er *Normalisere magnitudo til 1* som blir ufrørt på både STFT 1 og 2, for å gjøre dataen magnitudoavhengig. STFT 1 blir invers med *signal.istf* funksjonen fra Scipy [33]. Igjen, med hammingvindu og samme størrelse på vinduslengde og hopplengde som tidligere. En PCA transformasjon blir gjort på ISTFT med funksjonen *PCA* fra Sklearn og *fit_transform* [34]. Det er valgt å gå for 3 PCer per segment.

3.6 Valg av maskinlæringsmodell og arkitektur

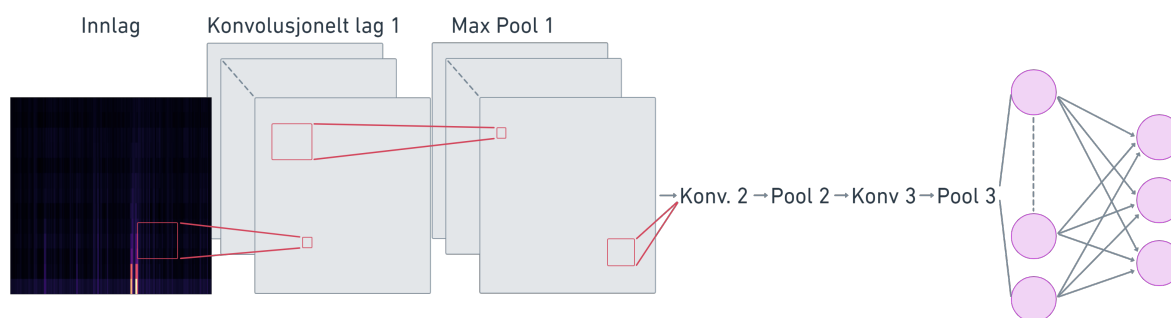
Ved valg av maskinlæringsmodell er det viktig å ta hensyn til valgt lydrepresentasjon. MFCCer og spektrogram kan analyseres som bilder, en 2D CNN modell kan da være gunstig å bruke. For envelopes, PCA og ISTFT kan for eksempel SVM, MLP og 1D CNN brukes. Oppgavens metode på valg av arkitektur er å gå for standard modellarkitekturer og kun fokusere på å endre hyperparametere som antall epoker og gruppestørrelse for ANN modeller. Valget om å gå for faste modellarkitekturer er gjort for å spare tid, på bekostning av at mer spesialiserte modeller kan prestere bedre enn den fastsatte arkitekturen. Totalt vil 26 ulike modeller trenes, seks SVM modeller, ti MLP-modeller, seks 1D CNN modeller og fire 2D CNN modeller.

3.6.1 2D CNN og 1D CNN

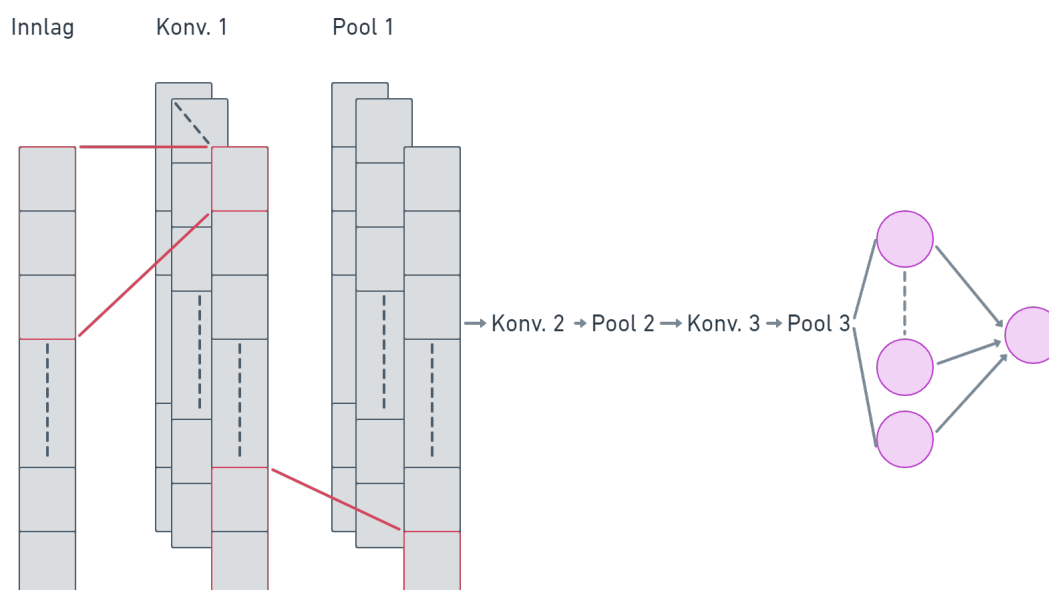
Modellarkitekturer for 2D og 1D CNN algoritmene er vist i figur 12 og 13. Arkitekturen er inspirert av prosjektoppgaven [14] og The Keras Blog [35]. Oversikt over detaljer for lagene i 1D CNN modellen er vist i 2, og tabell 3 for 2D CNN.

2D CNN modellen har tre 2D konvolusjonelle lag, og 1D CNN har tre 1D lag, hvor alle lagene er aktivert med en Leaky ReLu aktiveringsfunksjon. Null polstring er brukt på alle lag for å beholde lik dimensjon som inngangen.

For både 1D CNN og 2D CNN er første lag vektet med 32 filtre, andre og tredje lag med 64 filtre hver. Størrelsen på filtrene for 1D CNN er (3) på alle lag, og (3, 3) for alle lag i 2D CNN modellen. Max-Pooling er brukt etter hvert konvolusjonelle lag, med en størrelse på (2, 2) for 2D CNN og (2) for 1D CNN.



Figur (12) Arkitektur for 2D CNN modell med klassifisering av tre klasser.



Figur (13) Arkitektur for 1D CNN modell med binær klassifisering.

Etter siste Max-pooling operasjon blir spekteret flatet ut i en 1D array og er så fullt koblet til ett lag med 128 nevroner for begge modellene. Dette laget er aktivert med en Leaky Relu funksjon, som igjen er koblet til utgangen. Modellen har et utgangslag aktivert med en Sigmoid funksjon, og gir ut én sannsynlighet per klasse. Hver utgang gir med andre ord en verdi mellom 0 og 1, uavhengig av hva de andre sannsynlighetene er for modeller med flere enn to klasser. Dette gir modellen muligheten til å si at det kan være like sannsynlig at det er natt som at det er støyete, for lydmiljøindikatoren. Utgangslaget til modellen brukt for lydmiljøanalysen har tre nevroner i utgangslaget sitt. Alle andre modeller er binære klassifikasjoner, og har dermed én nevron på utgangen.

	Max-Pool	Konv. lag 1	Konv. lag 2	Konv. lag 3	Lag 4	Output
Aktivering	-	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Sigmoid
Antall filter/ nevroner	-	32	64	128	128 nevroner	1 nevron
Størrelse på filter	(2)	(3)	(3)	(3)	-	-
Polstring	Lik	Lik	Lik	Lik	-	-

Tabell (2) Oversikt over detaljer for lagene i 1D CNN modellen.

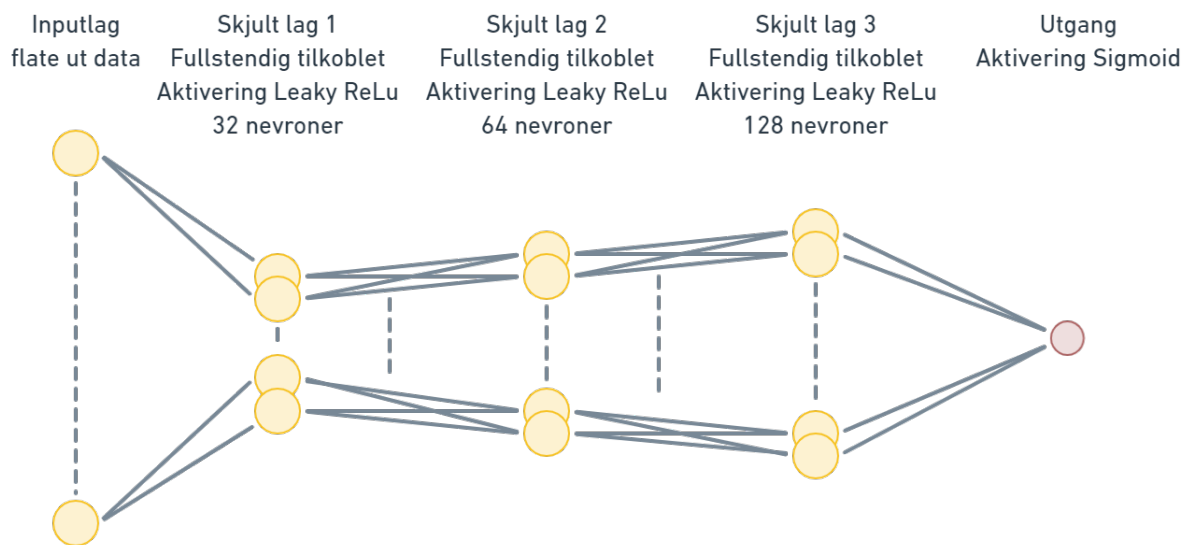
	Max-Pool	Konv. lag 1	Konv. lag 2	Konv. lag 3	Lag 4	Output
Aktivering	-	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Leaky ReLu, $\alpha = 0,1$	Sigmoid
Antall filter/ nevroner	-	32	64	128	128 nevroner	1(3) nevroner
Størrelse på filter	(2, 2)	(3, 3)	(3, 3)	(3, 3)	-	-
Polstring	Lik	Lik	Lik	Lik	-	-

Tabell (3) Oversikt over detaljer for lagene i 2D CNN modellen.

Fordi lyd miljøindikatoren tar for seg tre klasser, ble modellen trent med en categorical crossentropy og Adam optimizer fra Keras [36]. Modellene til aktivitetsindikatoren ble trent med Binary Crossentropy og Adam optimizer, også fra Keras. Leaky ReLu ble valgt i stedet for vanlig ReLu for å unngå at lave verdier ikke er differensierbare ved null.

3.6.2 MLP

MLP modellen har tre fullt tilkoblede skjulte lag, ett inngangslag og ett utgangslag, som sett i figur 14. Alle skjulte lag er aktivert med en Leaky Relu funksjon. Det første skjulte laget inneholder 32 nevroner, det andre laget 64 og det tredje 128. Antall nevroner for hvert lag var valgt med inspirasjon fra arkitekturen til CNN modellene. Utgangen er lik som CNN modellen, med en Sigmoid funksjon. Modellen ble trent med en Binary Crossentropy og Adam optimizer fra Keras, fordi modellen kun er brukt til aktivitetsindikatoren.



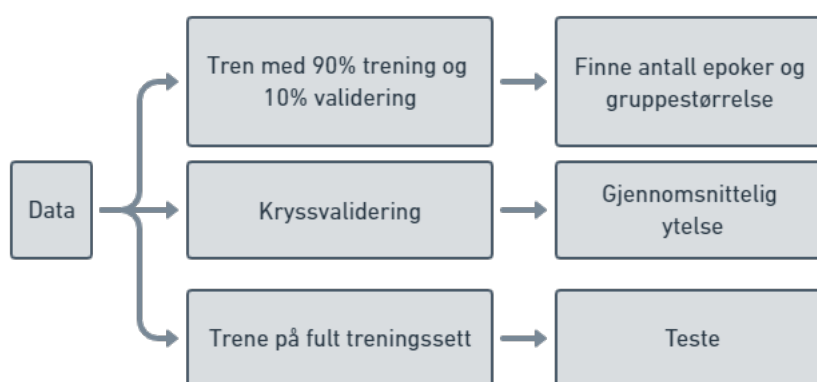
Figur (14) Arkitektur av MLP modellen.

3.7 Trening og test

Blokkdiagram i figur 15 viser de tre ulike stegene for trening av ANN modellene. Hyperparameter justering på antall epoker og gruppestørrelse er valgt etter å ha trent modellen på 90% av treningssettet og validert på resterende 10% med en funksjon kalt *callbacks.EarlyStopping* fra Keras biblioteket [37]. funksjonen stopper treningen når valideringsstapet øker, som kan være en indikasjon på overtrening. Man velger vilkårlige verdier på antall epoker i første omgang og ulike gruppestørrelse, i dette tilfellet 100 epoker og gruppestørrelse på [10, 20, 500, 1000]. Det er greit å ha en høy verdi på antall epoker for å sørge for at tapsfunksjonen har muligheten til å minimeres, før treningen kanskje stoppes av funksjonen som detekterer overtrening. Tapsfunksjonene for trenings og valideringssettene blir plottet for å analysere hvilke hyperparametere som gir best resultat.

Kryssvalidering med 10-folder er valgt basert på innlegget [38], hvor det nevnes at 10 folder generelt resulterer i et modellferdighetsestimert med lav skjevhet og liten varians. *Presisjon, tilbakekall, f1-score* og *nøyaktighet* er valgt evalueringsmål som kalkuleres av kryssvalideringen for å kunne si noe om kvaliteten på modellene. For kryssvalideringen kan det være gunstig å trene på få epoker for å minimere prosesseringstid. Hvis validering og treningstapet er stabilt ut treningen, kan man gå for minst antall epoker når tapet stabiliseres nær null.

Deretter blir modellen trent på fullt treningssett og testet på fulle lyd dager. For å glatte ut prediksjonsplottene gitt av modellene, brukes et Savitzky-Golay digitalt filter med funksjonen *signal.savgol_filter* fra Scipy biblioteket [39].



Figur (15) Treningssettene for CNN og MLP modellene trenes tre separate ganger for tre ulike formål.

Hyperparameter justering for SVM modellene gjøres ved bruk av et rutenettsøk (eng. grid search) med funksjonen *model_selection.GridSearchCV* fra Scikit-learn biblioteket [40]. Valgte hyperparametere å justere på er kostparameteren C med verdiene [100, 200], kernelen var fast og valgt til å være lineær. Kryssvalideringen av SVM modellene ble utført på datamaskinen satt opp med Ubuntu 18 og hadde GeForce 1080.

Det er ønsket å lage en baselinemodell for en typisk normal stille dag ved å bruke lyd miljøindikatoren, og teste én støyet og én lite støyet dag til for sammenligning med baselinen. Det er i tillegg ønsket å sammenligne prediksjonsplott fra begge statusindikatorerne ved å ha et felles testsett.

4 Resultater

4.1 Lydmiljøindikator

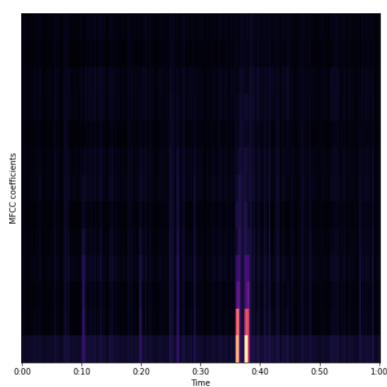
4.1.1 Datasett

Treningssettet består av 2560 segmenter, 840 natt samples, 910 samples fra fôring og 810 samples av støy fra (hvilke dager). Én full lyddag tilsvarer 1440 segmenter på 1 min hver, hvor *.wav* filene har en størrelse på 5,14 GB til sammen. Datamaskinen brukt hadde en begrensning på 15,8 GB brukbar RAM, som igjen førte til en begrensning på størrelsen på treningssettet.

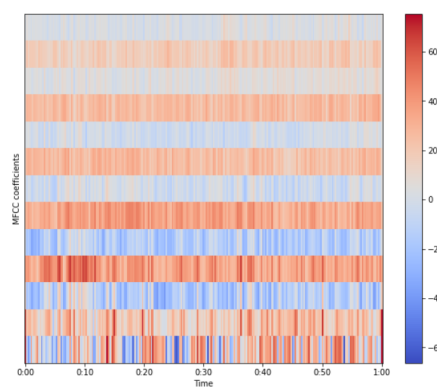
Det totale testsettet bestod av åtte fulle lyddager, der lyddagene ble prosessert hver for seg, og lagret på datamaskinen. Tre fulle dager som inneholdt lite miljøstøy ble brukt for å lage baseline plottet for klassene, to dager fra merd 3 og én fra merd 7. Én dag med mye støy fra merd 3 ble testet på for sammenligning med baselinen. Fire lyddager, to fra hver merd, ble testet på for sammenligning med senere testresultater fra aktivitetsindikatorene. Fordi det kun testes på én dag om gangen, tar vi for oss et helt datasett som treningssettet og én lyddag for test. Treningssettet er dermed 64% av datasettet.

MFCC

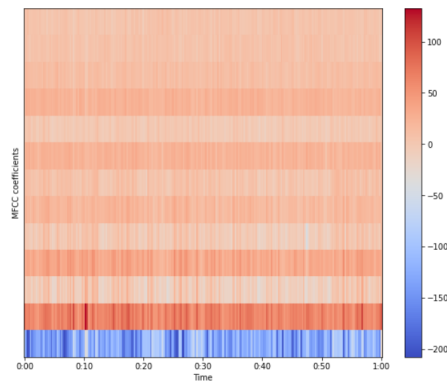
Figur 16 er et eksempel på et MFCC cepstrum av et segment fra klassen natt, figur 17 er fra fôring klassen og 18 er fra støy klassen. Hver MFCC har størrelse (13, 241), med andre ord 241 verdier per koeffisient.



Figur (16) MFCC av et lydsegment fra natt.



Figur (17) MFCC av et lydsegment fra fôring.

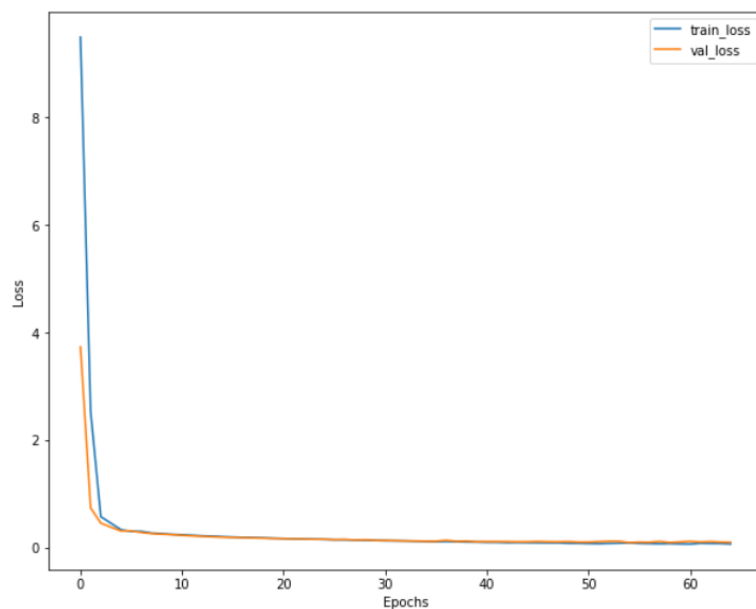


Figur (18) MFCC sprektrogram av et lydsegment med støy.

4.1.2 Trening

Hyperparametere og kryssvalidering

Figur 19 viser tapsfunksjonen til modellen. Treningen stoppet etter 65 epoker. Treningen var stabil etter 10 epoker, dermed er det brukt 15 epoker videre i kryssvalideringen og siste trening. Tabell 4 viser evalueringverdiene fra kryssvalideringen. Gjennomsnittlig F1-mål, tilbakekall, presisjon og nøyaktighet er på 89% hver. Som sagt er valgt epoke 15, men treningen stoppet etter 65 epoker og gruppestørrelsen er på 1000. Tabell 5 tar for seg forvirringsmatrisen for modellen.



Figur (19) Tapsfunksjonene fra trening og validering av 2D CNN modellen til lyd miljøindikator.

Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse
2D CNN	0,89	0,89	0,89	0,89	15(65)	1000

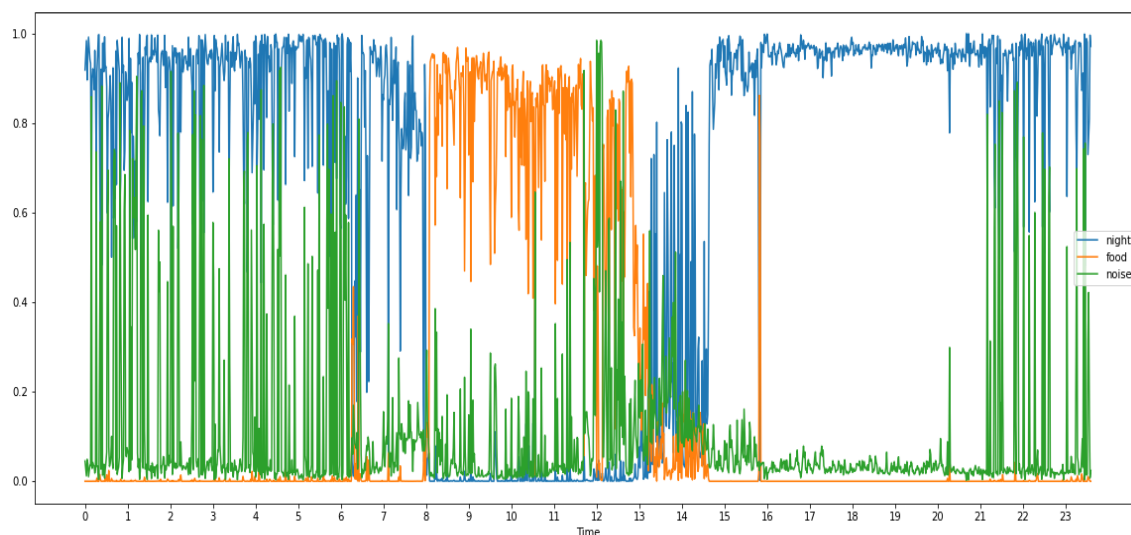
Tabell (4) Ytelsesberegninger fra kryssvalideringen til 2D modellen til lydmiljøindikatoren.

2D CNN			
	Pred. natt	Pred. fôring	Pred. støy
Sann natt	760	0	80
Sann fôring	10	884	16
Sann støy	76	100	634

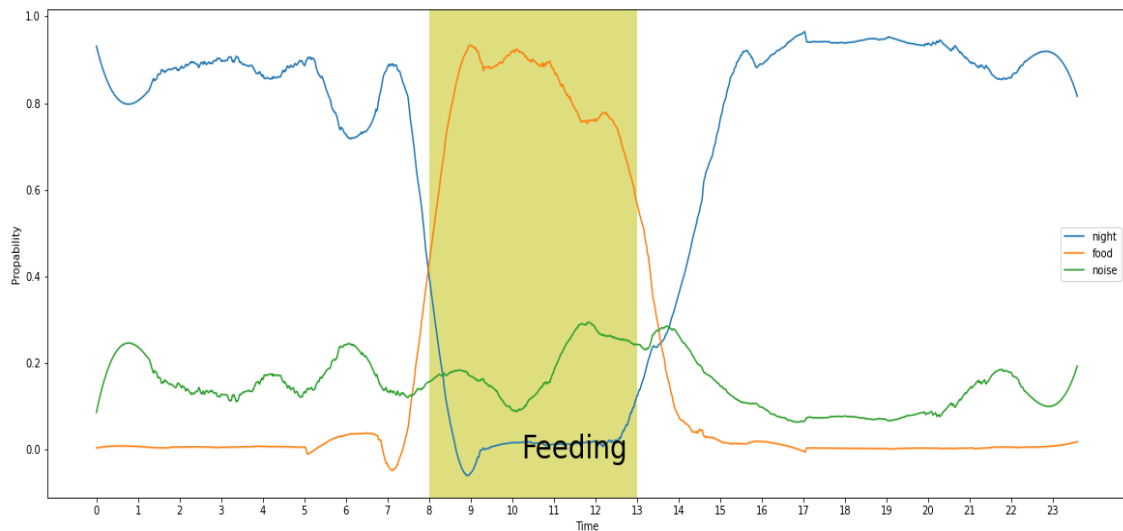
Tabell (5) Forvirringsmatrise for 2D CNN modellen til lydmiljøindikatoren.

4.1.3 Test

Figur 20 er plottet av ufiltrert prediksjon av samme dag som spektrogrammet vist i figur 7. Det er så brukt et Savgol filter med vindusstørrelse på 151 samples (minutter) og 3. orden. Figur 21 er filtrert versjon av testdagen. Det er et eksempel på en relativt støyfri dag. Den blå grafen viser fordelingen av klassen natt løpet av en dag, den oransje er fordelingen av fôring og den grønne viser fordelingen av støy for alle prediksjonsplott vist i figur 20-27.

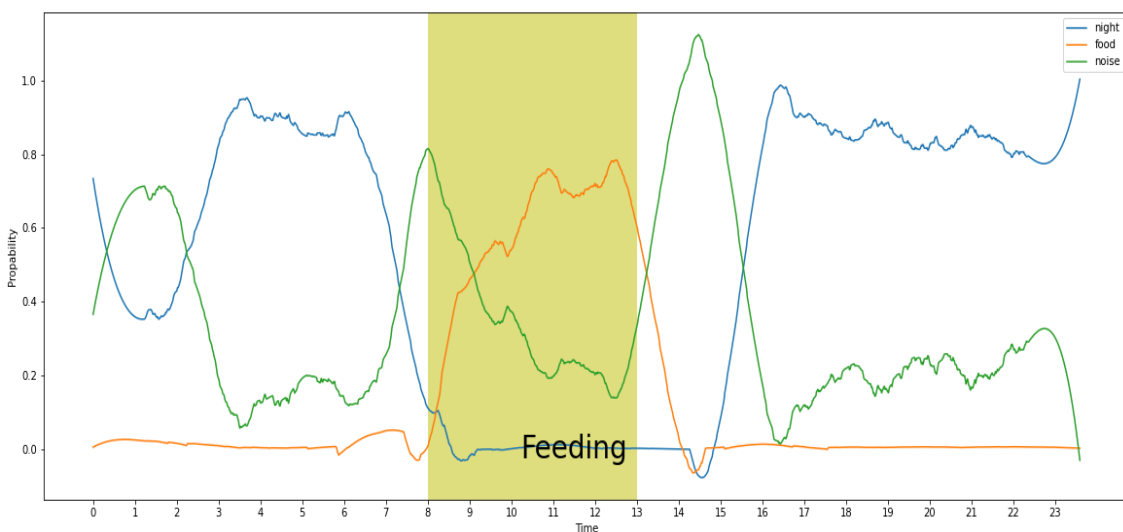


Figur (20) Ufiltrert prediksjon av en dag med lite støy. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.



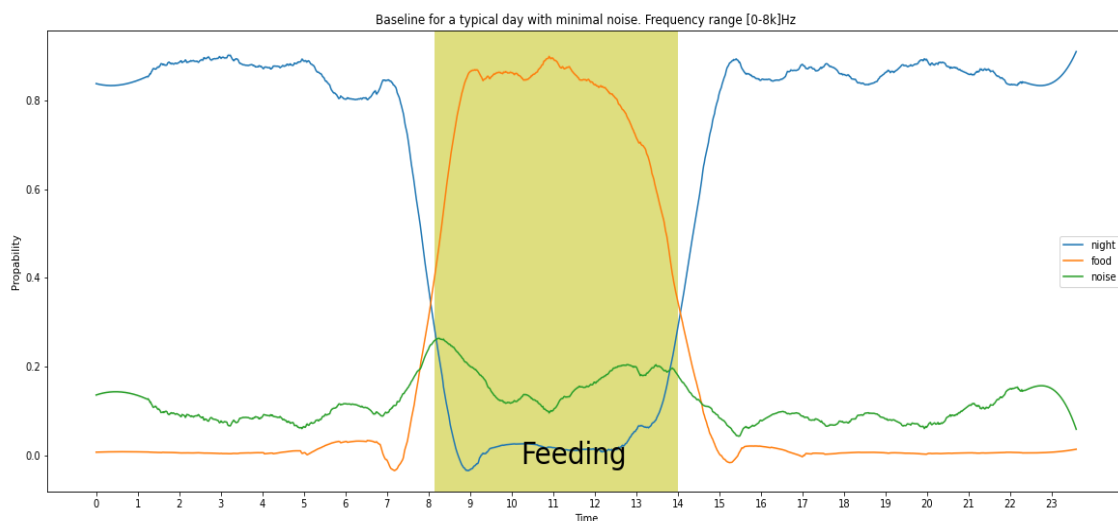
Figur (21) Filtret prediksjon av en dag med lite støy. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.

Figur 22 er filtret fordeling av klassene for en relativt støyete dag, spektrogrammet til lyd dagen er vist i figur 8.



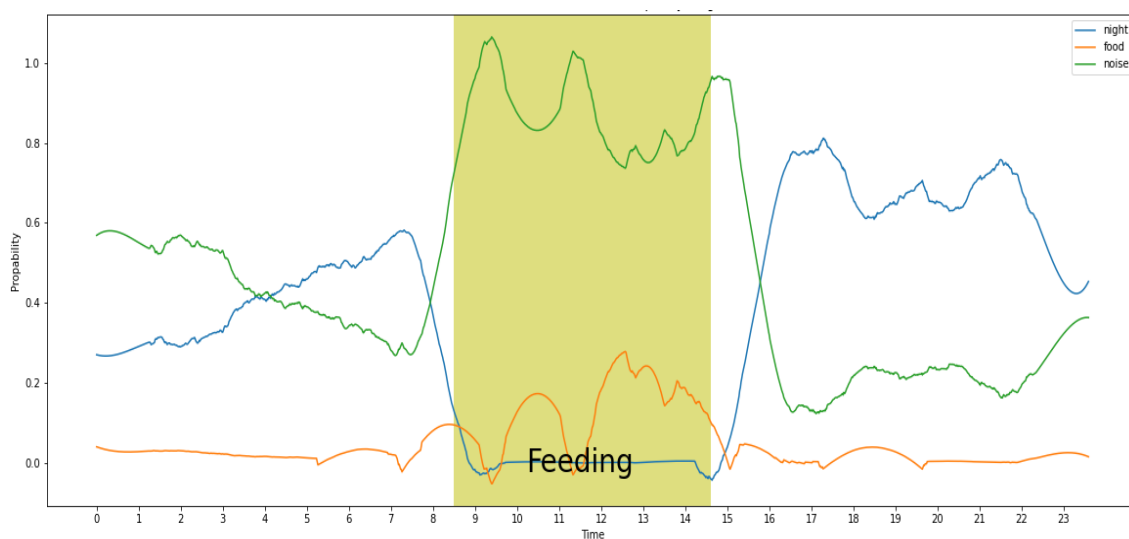
Figur (22) Filtret prediksjon av en dag med mye støy. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.

Baseline plottet er vist i figur 23 og er gjennomsnittet av hver klassefordeling av tre ulike relativt lite støyete dager.

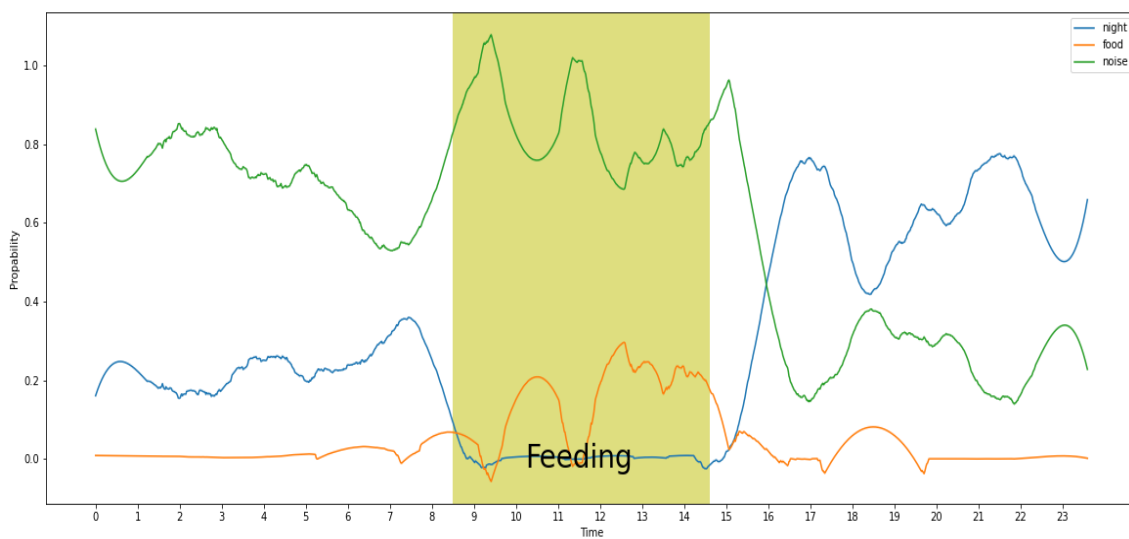


Figur (23) Baseline for en typisk støyfri dag. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.

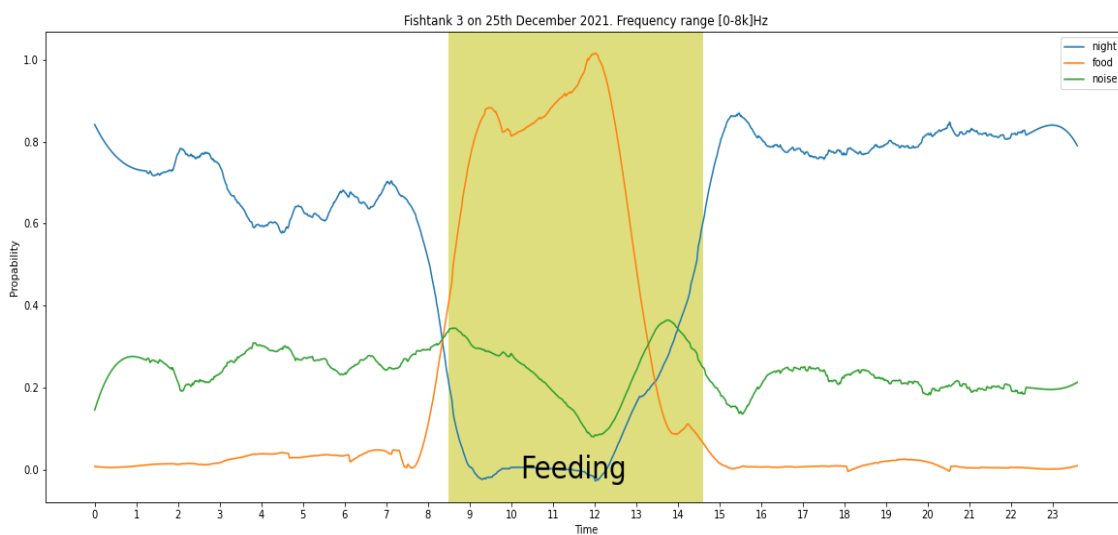
Det er som nevnt tidligere valgt ut fire tilfeldige lyddager som et felles testsett for lyd miljø- og aktivitetsindikatoren. Det er to dager, 19.12.2021 og 25.12.2021, fra to ulike merder 3 og 7. Figur 24 er fra merd 3 19.12.2021, figur 25 er samme dag fra merd 7. Figur 26 er fra merd 3 25.12.2021, figur 27 er samme dag fra merd 7.



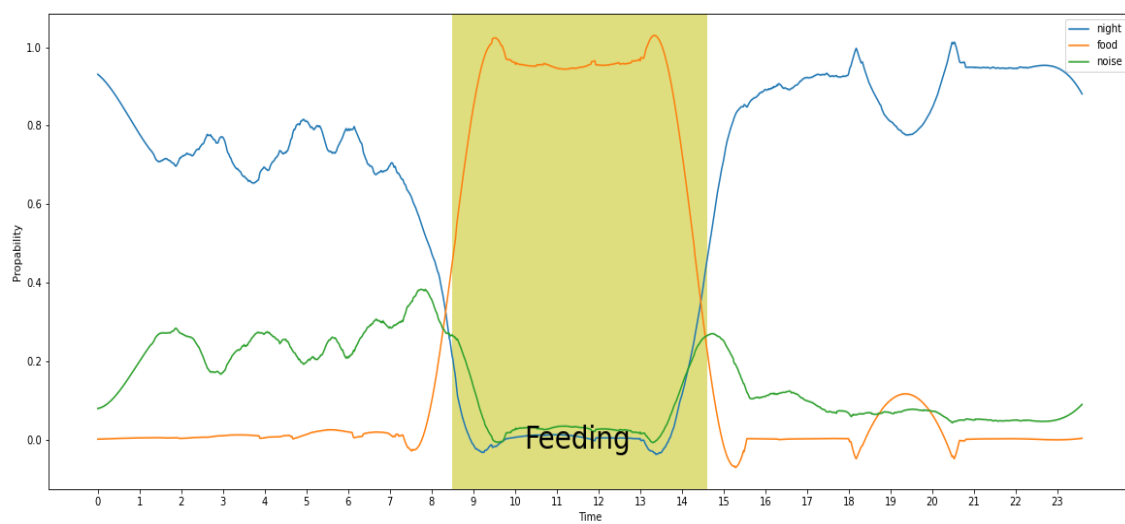
Figur (24) Filtrert prediksjon av dag 19.12.2021 merd 3. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.



Figur (25) Filtret prediksjon av dag 19.12.2021 merd 7. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.



Figur (26) Filtret prediksjon av dag 25.12.2021 merd 3. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.

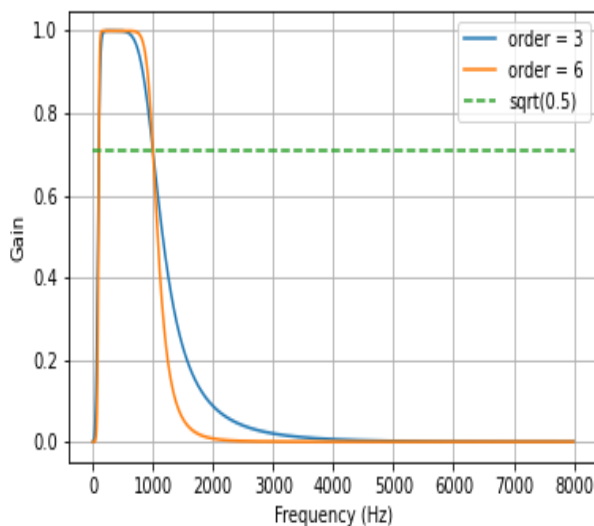


Figur (27) Filtret prediksjon av dag 25.12.2021 merd 7. Blå graf representerer prediksjon av natt, oransje er fôring og grønn er støy.

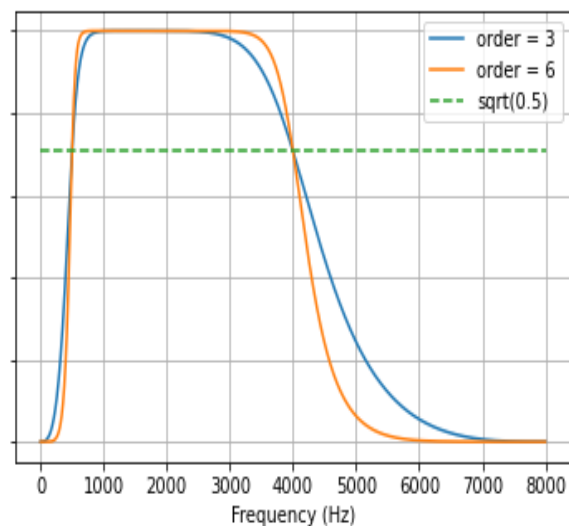
4.2 Aktivitetsindikator

4.2.1 Båndpassfilter

Figur 28 viser [100-1k] Hz Butterworth båndpassfilteret. Grafen er plottet for å sjekke hvilken orden som skal velges. Orden 6, den oransje grafen, filtrerer best. Det er dermed valgt å gå for orden lik 3 fordi signalene filtreres to ganger, som da tilsvarer at signalet filtreres med orden 6. Det samme går for båndpassfilteret for frekvensområdet [500-4k] Hz vist i figur 29. Fordi filtrene kan slippe gjennom uønskede frekvenser over den høyeste frekvensgrensen, velges det å resample med 2,5 ganger den høyeste valgte frekvensen. Det tilsvarer $f_{s_ny} = 2,5 \cdot 1\text{kHz} = 2,5\text{kHz}$ for [100-1k] Hz båndpassfilteret, den høyeste frekvensen som opptrer i det filtrerte signalet er da $f_{max_ny} = f_{s_ny}/2 = 1,25\text{kHz}$. For det andre filteret blir den nye samplingsfrekvensen $f_{s_ny} = 2,5 \cdot 4\text{k} = 10\text{kHz}$, og $f_{max_ny} = 5\text{kHz}$. Frekvensområdene av interesse er dermed [100-1,25k] Hz og [500-5k] Hz.



Figur (28) Butterworth båndpassfilter [100-1k] Hz for 3. orden, oransje graf og 6. orden, blå graf. Den stipla grønne linjen viser 3dB nedgang på gain.



Figur (29) Butterworth båndpassfilter [500-4k] Hz for 3. orden, oransje graf og 6. orden, blå graf. Den stipla grønne linjen viser 3dB nedgang på gain.

4.2.2 Datasett

Treningssettet til aktivitetsindikatoren inneholder 1800 samples på 10 s hver, 900 merket som lav aktivitet og 900 merket med høy aktivitet. Testsettet er fire fulle lyddager, der hver dag består

av 8640 segmenter. Dataen er representert på fem ulike måter, PCA, ISTFT, STFT 1, STFT 2 og envelope. I tilfellet der datasettet består av treningssettet og én lyd dag, er treningssettet 17% av datasettet. Det kommer av begrensninger med RAM størrelse og få antall dokumenterte støyfrige dager. I tillegg var det begrensning på når på dagen segmentene for høy aktivitet ble utvalgt, kun 30 min før føring. For å bevare et balansert treningssett, ble like mange segmenter fra lite aktivitet klassen utvalgt, til tross for muligheten for mer data for denne klassen. Tabell 6 viser størrelse på hvert segment i hver av lydrepresentasjonene. PCA, ISTFT og Envelope dataen er 1D arrays, mens STFT er 2D matriser. Størrelsen på hver sample definerer formen på dataen modellene trener og predikerer på.

	$f_{max} = 1,25kHz$	$f_{max} = 5kHz$
Representasjon	Størrelse	Størrelse
PCA	(3)	(3)
ISTFT	(10 000)	(25 000)
STFT 1	(12, 1924)	(12, 1924)
STFT 2	(4501, 21)	(1250, 21)
Envelope	(151)	(40)

Tabell (6) Oversikt over størrelse på lydrepresentasjonene PCA, ISTFT, STFT log 2 og envelope for frekvensområdene [100-1,25k] Hz og [500-5k] Hz.

Videre vil to segmenter fra hver klasse, lite aktivitet og høy aktivitet, bli presentert i de ulike representasjonene av lyd valgt.

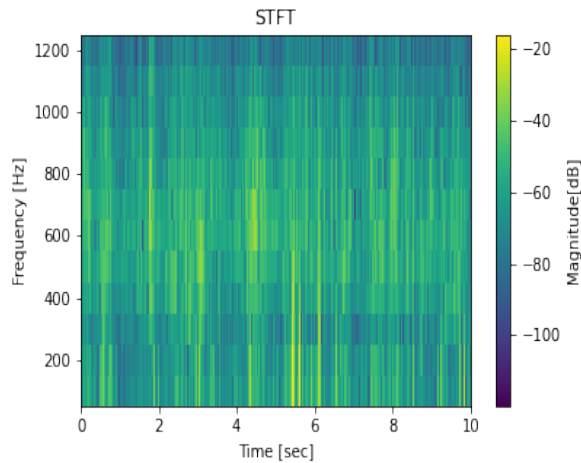
STFT 1 OG STFT 2

Figur 30 viser magnitudo spektrogrammet i dB i frekvensområdet [100-1,25k] Hz av et tilfeldig sample fra klassen med høy aktivitet, vinduslengden er på 10 ms og overlapp på 5 ms. Figur 31 er fase spektrogrammet i radianer av samme sample, men i dette tilfellet er magnituden normalisert til 1. Figur 32 og 33 viser magnitudo og fase spektrogrammet av samme sample som figurene over, vinduslengden er på 1 s og overlapp på 50 ms.

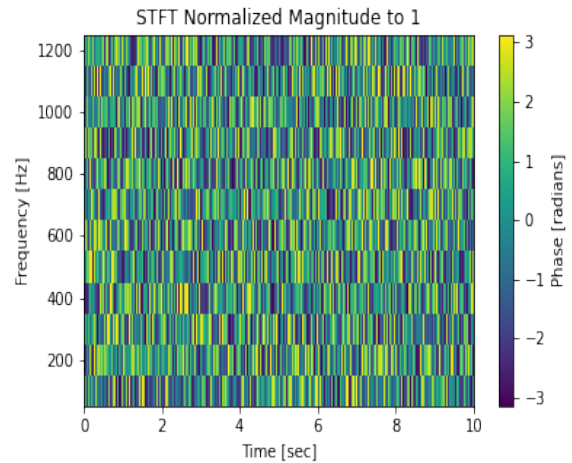
Figur 34 og 35 viser magnitudo og fase spektrogrammet av et tilfeldig sample fra klassen med lav aktivitet, vinduslengden er på 10 ms og overlapp på 5 ms. Figur 36 og 37 viser magnitudo spektrogrammet og fase spektrogrammet samme sample med lav aktivitet vinduslengden er på 1

s og overlapp på 50 ms.

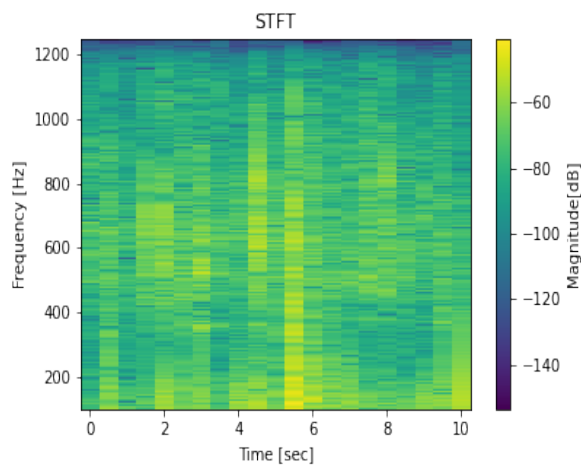
Figur 38 - 45 viser de samme plottene med samme rekkefølge som er nevnt ovenfor, men for frekvensområdet [500-5k] Hz.



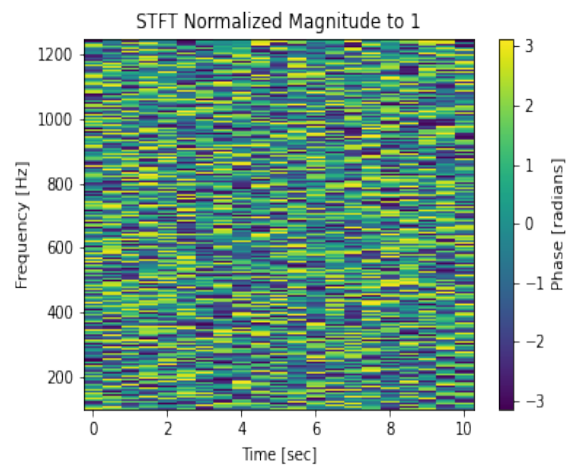
Figur (30) Magnitude [dB] spektrum av STFT 1 i frekvensområdet [100-1,25k] Hz, av et sample med høy aktivitet.



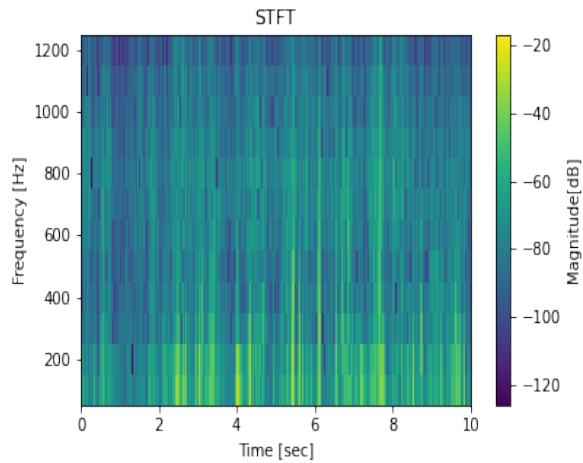
Figur (31) Fase [radian] spektrum av STFT 1 med normalisert magnitudo i frekvensområdet [100-1,25k] Hz, av et sample med høy aktivitet.



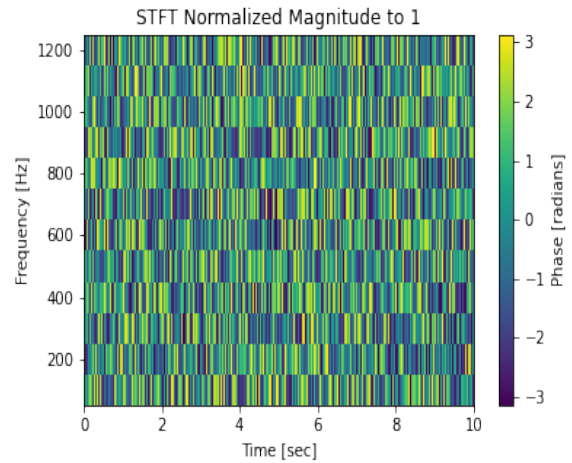
Figur (32) Magnitude [dB] spektrum av STFT 2 i frekvensområdet [100-1,25k] Hz, av et sample med høy aktivitet.



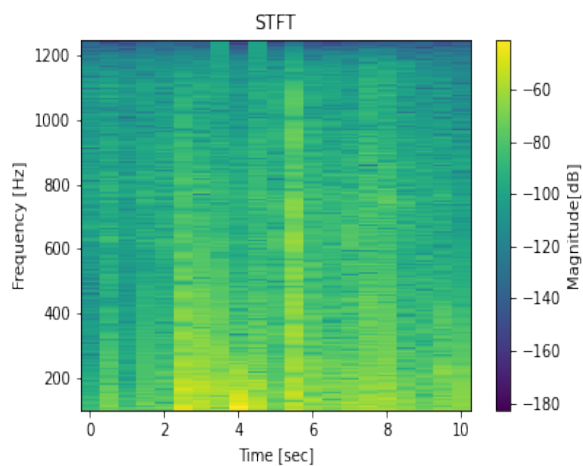
Figur (33) Fase [radian] spektrum av STFT 2 med normalisert magnitudo i frekvensområdet [100-1,25k] Hz, av et sample med høy aktivitet.



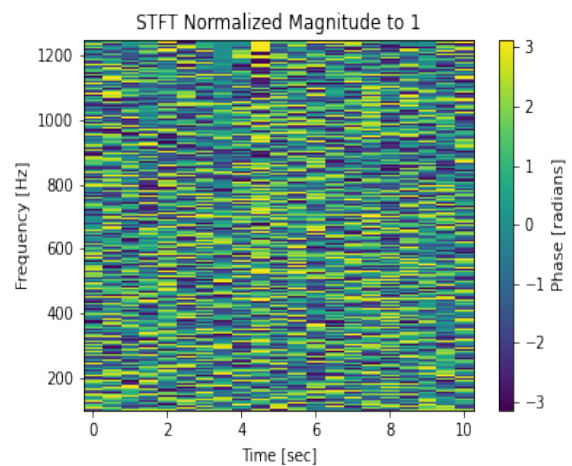
Figur (34) Magnitude [dB] spektrum av STFT 1 i frekvensområdet [100-1,25k] Hz, av et sample med lav aktivitet.



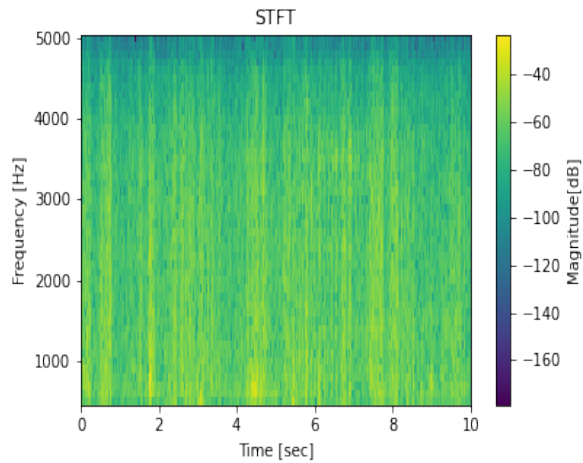
Figur (35) Fase [radian] spektrum av STFT 1 med normalisert magnitudo i frekvensområdet [100-1,25k] Hz, av et sample med lav aktivitet.



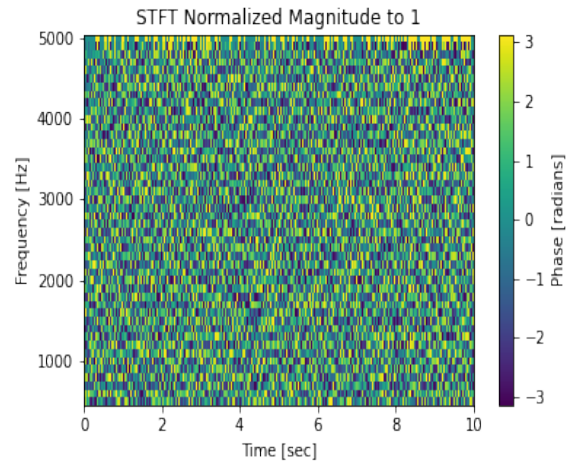
Figur (36) Magnitude [dB] spektrum av STFT 2 i frekvensområdet [100-1,25k] Hz, av et sample med lav aktivitet.



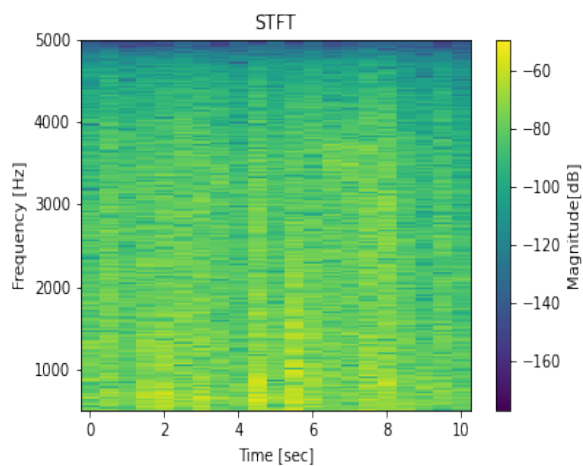
Figur (37) Fase [radian] spektrum av STFT 2 med normalisert magnitudo i frekvensområdet [100-1,25k] Hz, av et sample med lav aktivitet.



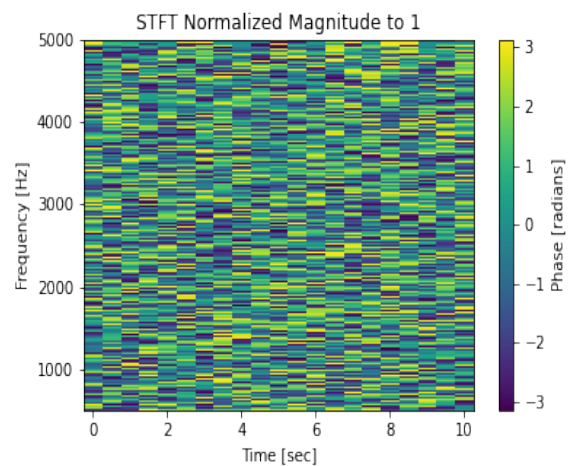
Figur (38) Magnitude [dB] spektrum av STFT 1 i frekvensområdet [500-5k] Hz, av et sample med høy aktivitet.



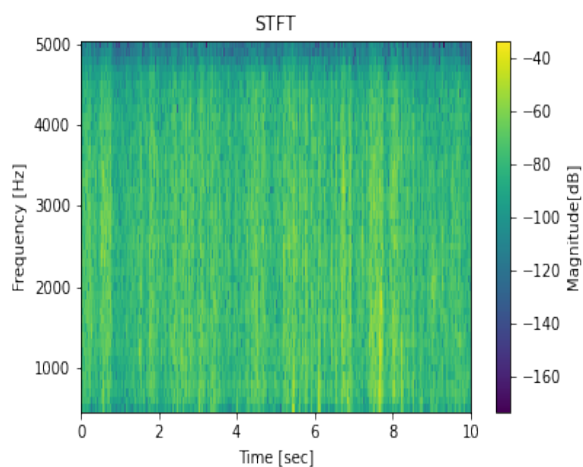
Figur (39) Fase [radian] spektrum av STFT 1 med normalisert magnitudo i frekvensområdet [500-5k] Hz, av et sample med høy aktivitet.



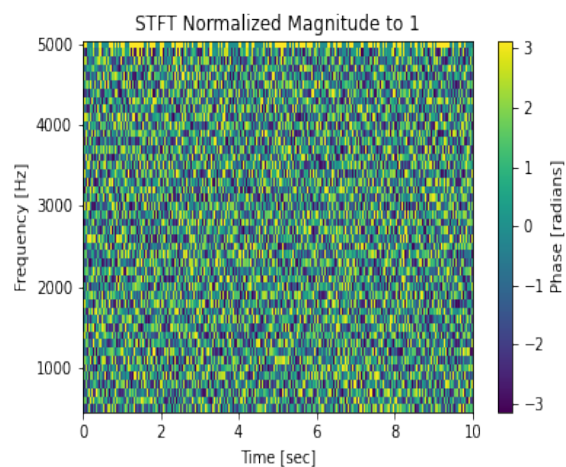
Figur (40) Magnitude [dB] spektrum av STFT 2 i frekvensområdet [500-5k] Hz, av et sample med høy aktivitet.



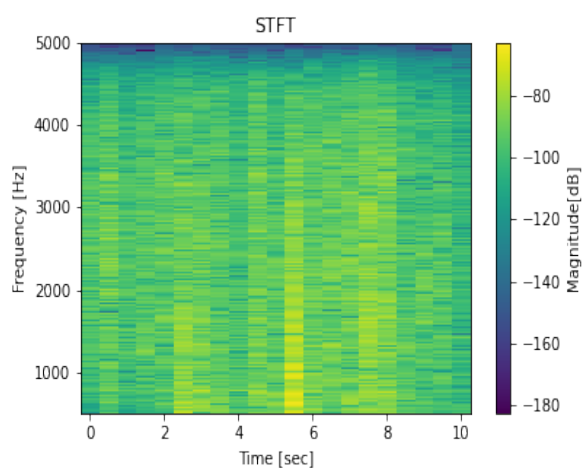
Figur (41) Fase [radian] spektrum av STFT 2 med normalisert magnitudo i frekvensområdet [500-5k] Hz, av et sample med høy aktivitet.



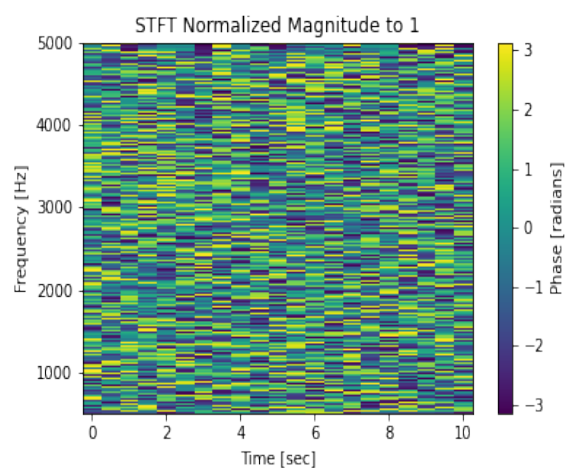
Figur (42) Magnitude [dB] spektrum av STFT 2 i frekvensområdet [500-5k] Hz, av et sample med lav aktivitet.



Figur (43) Fase [radian] spektrum av STFT 2 med normalisert magnitudo i frekvensområdet [500-5k] Hz, av et sample med lav aktivitet.



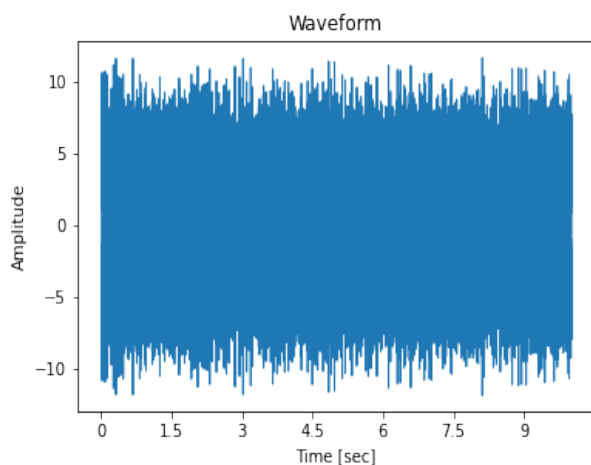
Figur (44) Magnitude [dB] spektrum av STFT 2 i frekvensområdet [500-5k] Hz, av et sample med lav aktivitet.



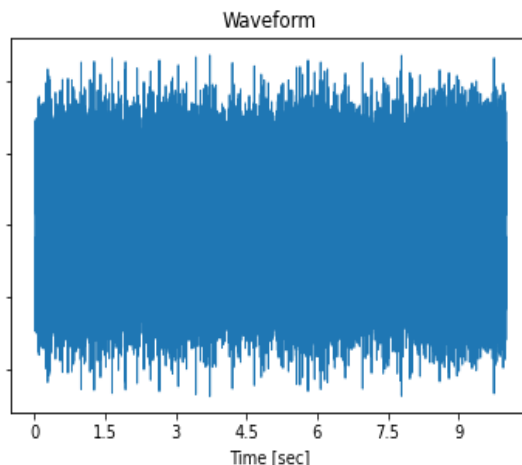
Figur (45) Fase [radian] spektrum av STFT 2 med normalisert magnitudo i frekvensområdet [500-5k] Hz, av et sample med lav aktivitet.

ISTFT

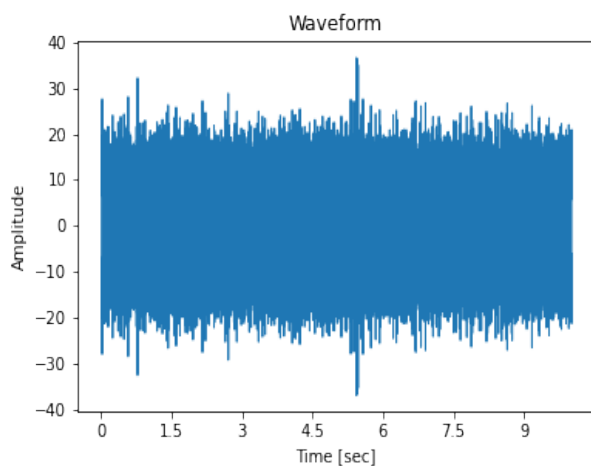
Figur 46 er inverse av STFT av lydsegmentet fra høy aktivitetsklassen i frekvensområdet [100-1,25k] Hz som er vist i figur 30 og figur 47 er inverse av den prosesserte samplen vist i figur 34. Figur 48 og 49 viser plottet av ISTFT av figurene 38 og 42 i frekvensområdet [500-5k] Hz.



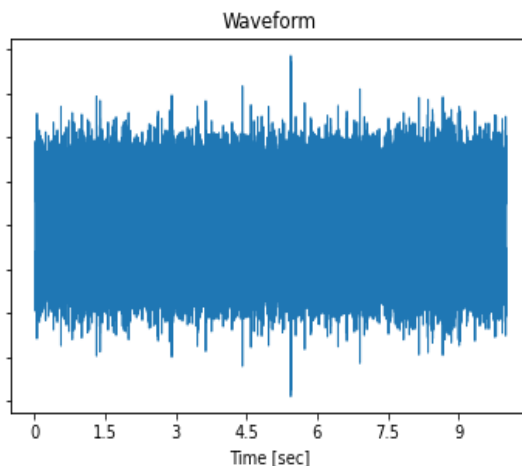
Figur (46) Bølgeformen til ISTFT av et lydsegment med høy aktivitet i frekvensområdet [100-1,25k] Hz. Amplitude over tid i sekunder.



Figur (47) Bølgeformen til ISTFT av et lydsegment med lite aktivitet i frekvensområdet [100-1,25k] Hz. Amplitude over tid i sekunder.



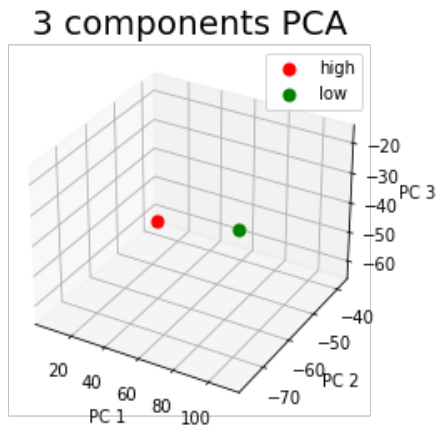
Figur (48) Bølgeformen til ISTFT av et lydsegment med høy aktivitet i frekvensområdet [500-5k] Hz. Amplitude over tid i sekunder.



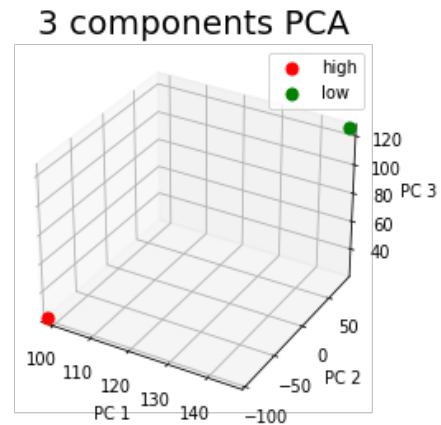
Figur (49) Bølgeformen til ISTFT av et lydsegment med lav aktivitet i frekvensområdet [500-5k] Hz. Amplitude over tid i sekunder.

PCA

Figur 50 viser tre komponents PCA transformasjon av ISTFT bølgeformene fra figur 46 og 47, hvor det røde punktet er sampelet med høy aktivitet og den grønne er lav aktivitet. Figur 51 er plottet til PCA transformasjonene til de samme segmentene, men i [500-5k] Hz frekvensområdet.



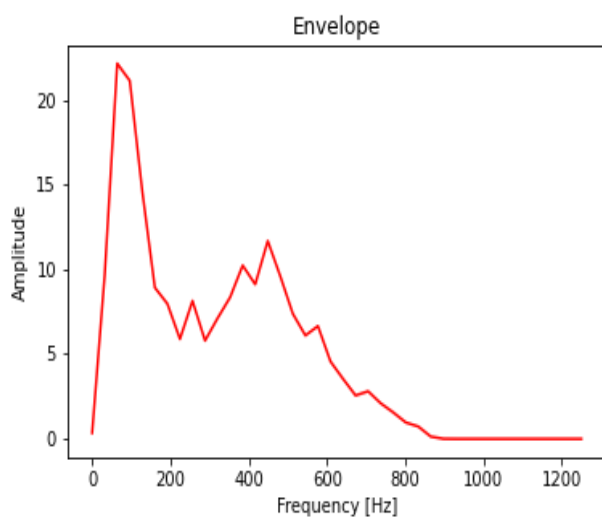
Figur (50) Trekomponent PCA transformerte lydsegmenter med lav (grønn) og høy (rød) aktivitet i frekvensområdet [100-1,25k] Hz.



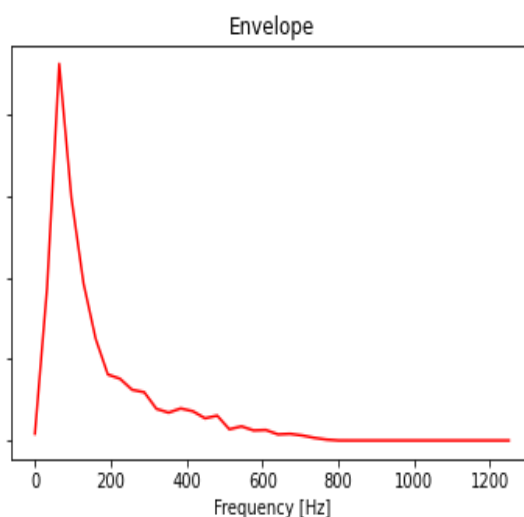
Figur (51) Trekomponent PCA transformerte lydsegmenter med lav (grønn) og høy (rød) aktivitet i frekvensområdet [500-5k] Hz.

Envelope

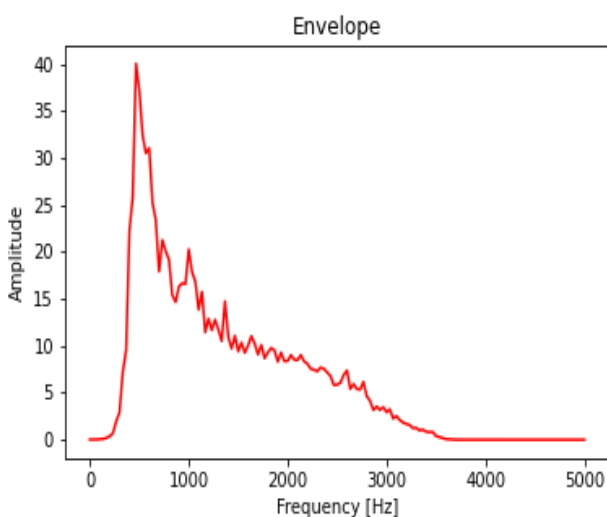
Figur 52 er envelopen i frekvensområdet [0-1,25k] Hz til DFTen av lydsegmentet fra høyaktivitetsklassen etter filtrering med [100-1k]Hz og figur 53 er det samme for lydsegmentet fra lavaktivitetsklassen. Figur 54 og 55 viser envelopene av de samme segmentene som ovenfor, men i frekvensområdet [500-5k] Hz.



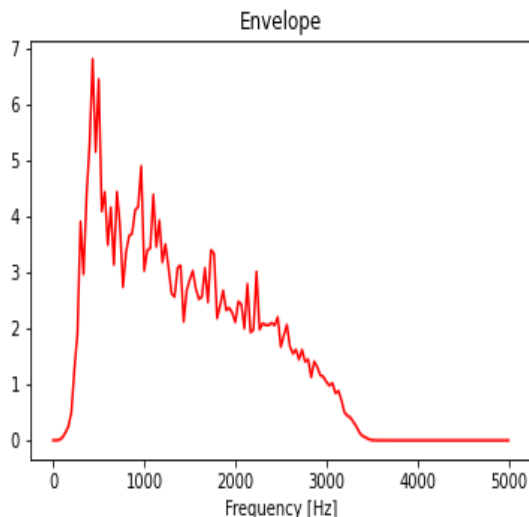
Figur (52) Envelope av et lydsegment med høy aktivitet i frekvensområdet [0-1,25k] Hz.



Figur (53) Envelope av et lydsegment med lav aktivitet i frekvensområdet [0-1,25k] Hz.



Figur (54) Envelope av et lydsegment med høy aktivitet i frekvensområdet [0-5k] Hz.



Figur (55) Envelope av et lydsegment med lav aktivitet i frekvensområdet [0-5k] Hz.

4.2.3 Trening

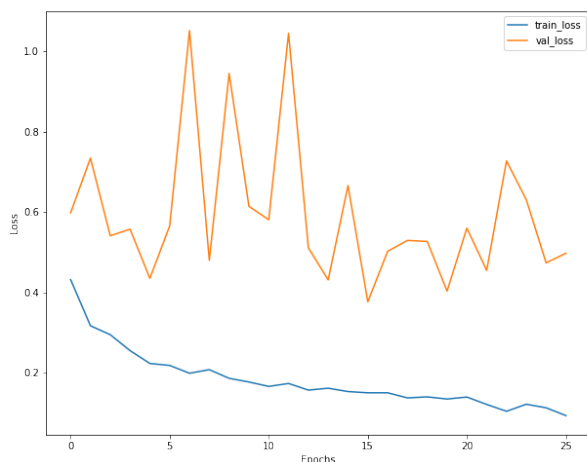
Rutenettsøk og kryssvalidering av SVM modellene ble gjort på en datamaskin som står på kontoret til SINTEF Ocean og ble koblet til via TeamViewer. Den bærbare datamaskinen som i utgangspunktet skulle bli brukt hadde verken nok minne eller kapasitet til å kjøre rutenettsøk og kryssvalidering på SVM. Datamaskinen som dermed ble brukt var satt opp med Ubuntu 18 og hadde GeForce 1080.

Hyperparametere og kryssvalidering for $f_{max} = 1,25k$ Hz

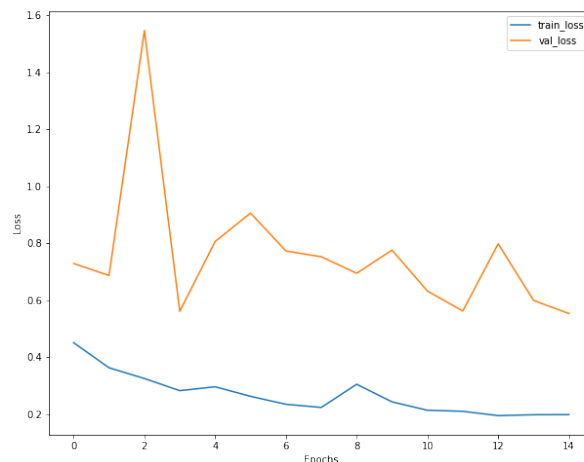
Tapsfunksjonene, ytelsesberegninger og forvirringsmatrisene til modellene brukt for lydrepresentasjonene valgt for aktivitetsindikatoren hvor dataen er filtrert med [100-1k] Hz båndpassfilteret vil bli presentert.

Figur 56 til 64 viser tapsfunksjonene til modellene for hver modell. Annenhver tabell 7 til 16 viser ytelsesberegninger og hyperparametere, og forvirringsmatrise for hver modell.

Envelope



Figur (56) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på envelope i frekvensområdet [0-1,25k] Hz.



Figur (57) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på envelope i frekvensområdet [0-1,25k] Hz.

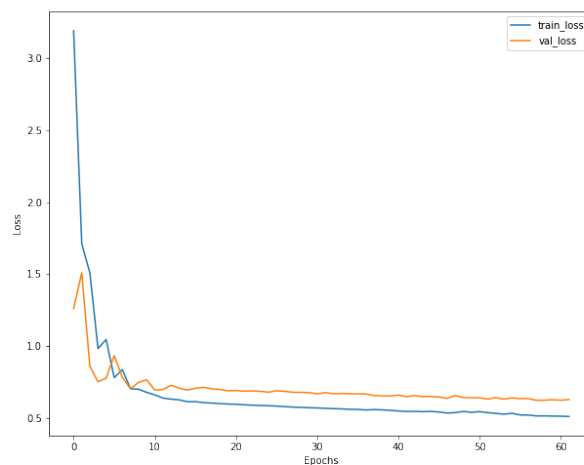
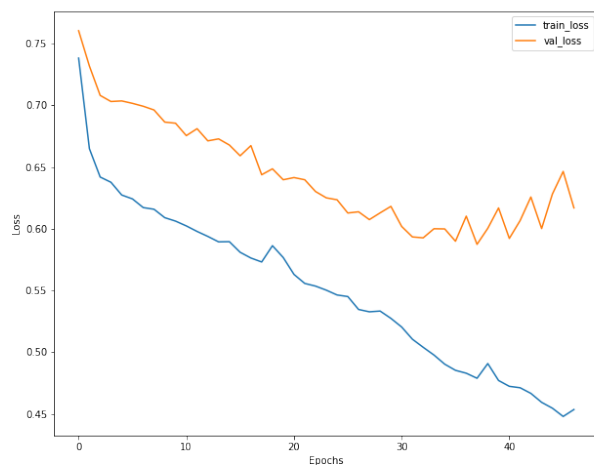
Envelope								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0,79	0,79	0,79	0,79	-	-	100	lineær
1D CNN	0,93	0,93	0,93	0,93	25	20	-	-
MLP	0,91	0,91	0,91	0,91	14	10	-	-

Tabell (7) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på envelope data i frekvensområdet [0-1,25k] Hz.

	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	844	56	834	66	705	195
Sann lav	62	838	64	836	175	725

Tabell (8) Forvirringsmatrise for modellene trent på envelope data i frekvensområdet [0-1,25k] Hz.

PCA



Figur (58) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på PCA i frekvensområdet [100-1,25k] Hz.

Figur (59) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på PCA i frekvensområdet [100-1,25k] Hz.

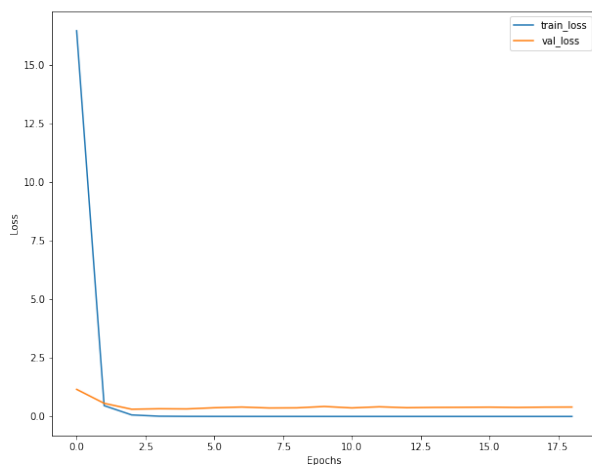
PCA								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0,52	0,57	0,60	0,57	-	-	100	lineær
1D CNN	0,93	0,93	0,93	0,93	47	200	-	-
MLP	0,72	0,72	0,72	0,72	62	500	-	-

Tabell (9) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på PCA data i frekvensområdet [100-1,25k] Hz.

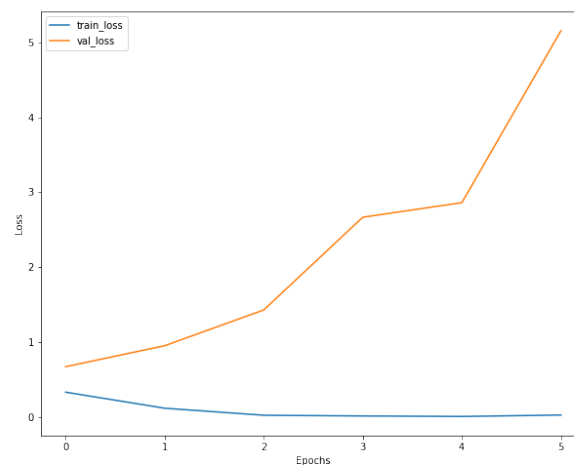
	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	844	56	834	66	705	195
Sann lav	62	838	64	836	175	725

Tabell (10) Forvirringsmatrise for modellene trent på PCA data i frekvensområdet [100-1,25k] Hz.

ISTFT



Figur (60) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på ISTFT i frekvensområdet [100-1,25k] Hz.



Figur (61) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på ISTFT i frekvensområdet [100-1,25k] Hz.

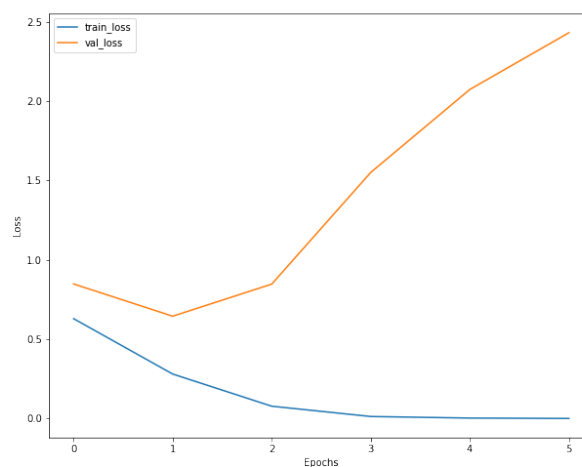
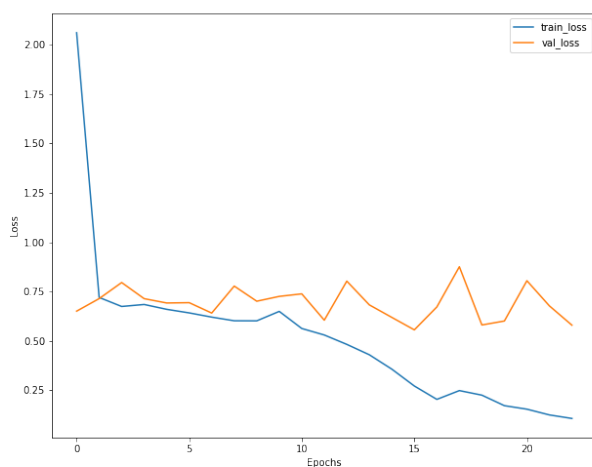
ISTFT								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0.93	0.93	0.93	0.93	-	-	100	lineær
1D CNN	0.91	0.91	0.92	0.91	8(18)	100	-	-
MLP	0.87	0.87	0.87	0.87	5	10	-	-

Tabell (11) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på ISTFT data i frekvensområdet [100-1,25k] Hz.

	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	844	56	853	47	725	175
Sann lav	62	838	107	793	59	841

Tabell (12) Forvirringsmatrise for modellene trent på ISTFT data i frekvensområdet [100-1,25k] Hz.

STFT 1



Figur (62) Tapsfunksjoner for trening (blå) og validering (oransje) for 2D CNN modellen trent og validert på STFT 1 i frekvensområdet [100-1,25k] Hz.

Figur (63) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på STFT 1 i frekvensområdet [100-1,25k] Hz.

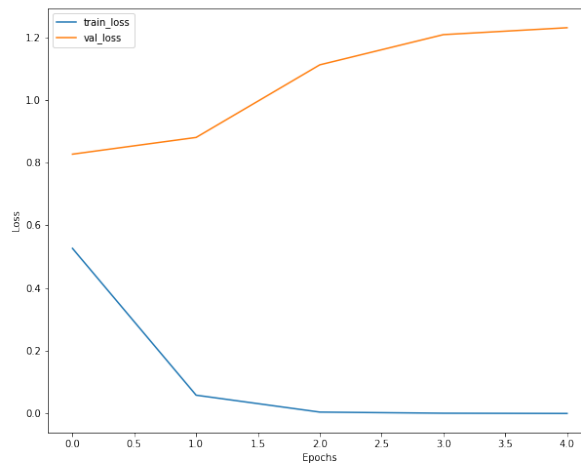
STFT 1								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
2D CNN	0,94	0,94	0,94	0,94	8(18)	100	-	-
MLP	0,87	0,87	0,87	0,87	5	10	-	-

Tabell (13) Ytelsesberegninger for 2D CNN og MLP modellene trent på STFT 1 data i frekvensområdet [100-1,25k] Hz.

	2D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	400	500	702	198
Sann lav	314	586	31	869

Tabell (14) Forvirringsmatrise for modellene trent på STFT 1 data i frekvensområdet [100-1,25k] Hz.

STFT 2



Figur (64) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på STFT 2 i frekvensområdet [100-1,25k] Hz.

STFT 2								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
MLP	0,85	0,85	0,85	0,86	5	10	-	-

Tabell (15) Ytelsesberegninger for MLP modellen trent på STFT 2 data i frekvensområdet [100-1,25k] Hz.

	MLP	
	Pred høy	Pred lav
Sann høy	709	191
Sann lav	70	830

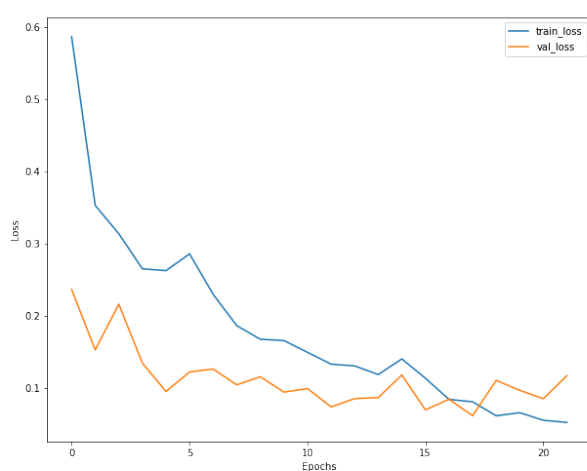
Tabell (16) Forvirringsmatrise for modellen trent på STFT 2 data i frekvensområdet [100-1,25k] Hz.

Hyperparametere og kryssvalidering for $f_{max} = 5k$ Hz

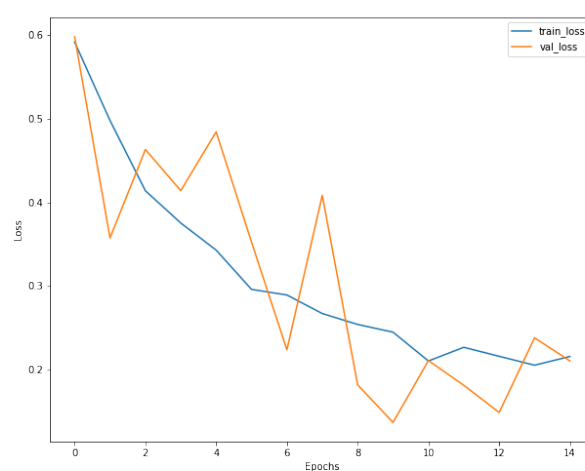
Tapsfunksjonene, ytelsesberegninger og forvirringsmatrisene til modellene brukt for lydrepresentasjonene valgt for aktivitetsindikatoren hvor dataen er filtrert med [500-4k] Hz båndpassfilteret.

Figur 65 til 73 viser tapsfunksjonene til modellene for hver modell. Annenhver tabell 17 til 26 viser ytelsesberegninger og hyperparametere, og forvirringsmatrise for hver modell.

Envelope



Figur (65) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på envelope data i frekvensområdet [0-5k] Hz.



Figur (66) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på envelope data i frekvensområdet [0-5k] Hz.

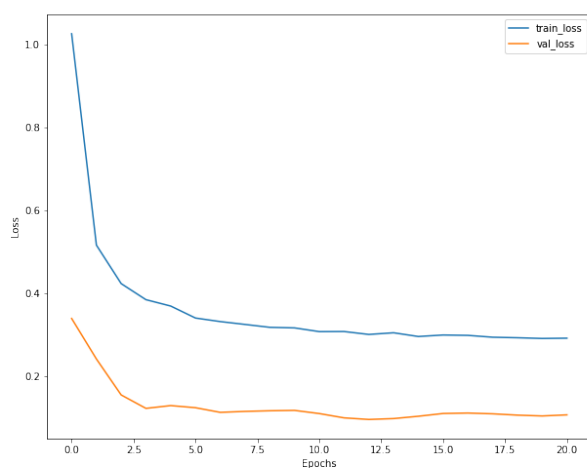
Envelope								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0,90	0,90	0,90	0,90	-	-	100	lineær
1D CNN	0,93	0,93	0,93	0,93	25	20	-	-
MLP	0,90	0,90	0,90	0,90	14	10	-	-

Tabell (17) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på envelope data i frekvensområdet [0-5k] Hz.

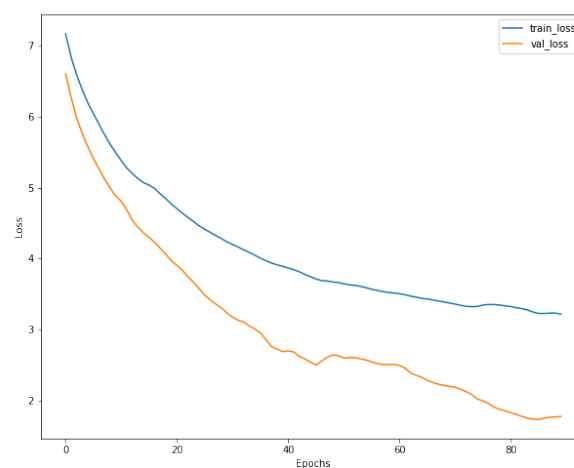
	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	803	97	839	61	812	88
Sann lav	89	811	64	836	98	802

Tabell (18) Forvirringsmatrise for modellene trent på envelope data i frekvensområdet [0-5k] Hz.

PCA



Figur (67) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på PCA data i frekvensområdet [500-5k] Hz.



Figur (68) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på PCA data i frekvensområdet [500-5k] Hz.

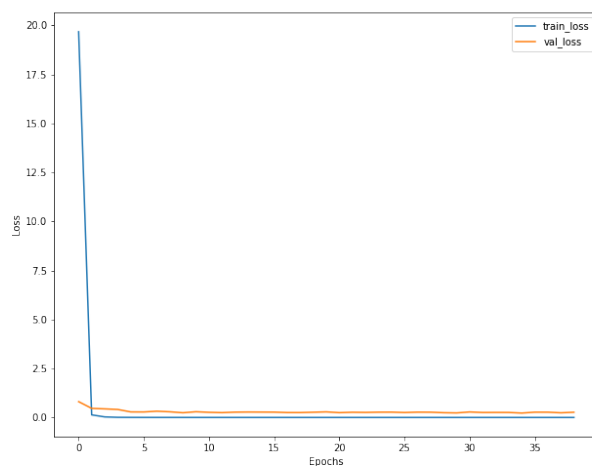
PCA								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0,86	0,86	0,86	0,86	-	-	200	lineær
1D CNN	0,88	0,88	0,88	0,88	47	200	-	-
MLP	0,88	0,88	0,88	0,88	62	500	-	-

Tabell (19) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på PCA data i frekvensområdet [500-5k] Hz.

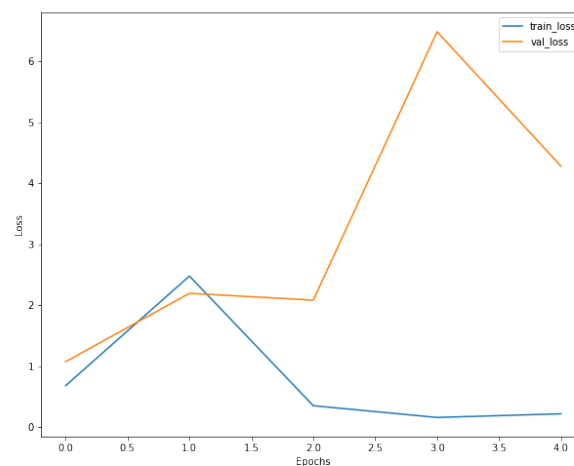
	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	749	151	718	182	738	162
Sann lav	96	804	41	859	58	842

Tabell (20) Forvirringsmatrise for modellene trent på PCA data i frekvensområdet [500-5k] Hz.

ISTFT



Figur (69) Tapsfunksjoner for trening (blå) og validering (oransje) for 1D CNN modellen trent og validert på ISTFT data i frekvensområdet [500-5k] Hz.



Figur (70) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på ISTFT data i frekvensområdet [500-5k] Hz.

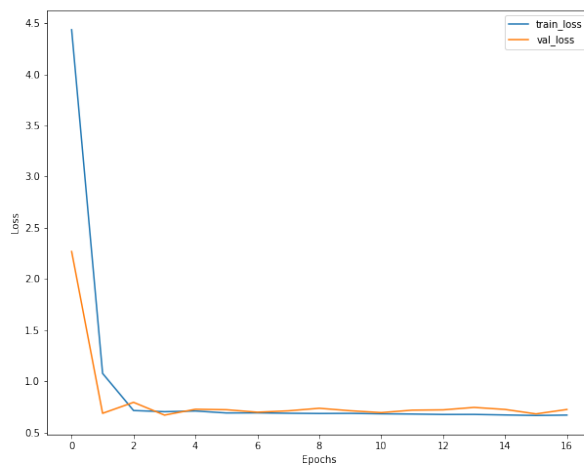
ISTFT								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
SVM	0,85	0,85	0,85	0,85	-	-	200	lineær
1D CNN	0,88	0,88	0,88	0,88	5(39)	100	-	-
MLP	0,60	0,64	0,71	0,64	4	20	-	-

Tabell (21) Ytelsesberegninger for SVM, 1D CNN og MLP modellene trent på ISTFT data i frekvensområdet [500-5k] Hz.

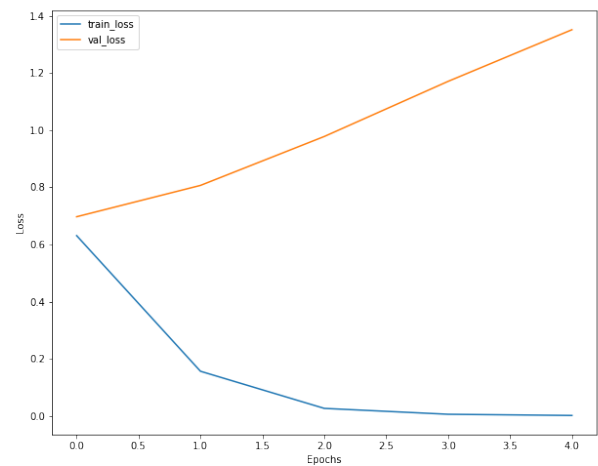
	SVM		1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	751	149	792	108	305	595
Sann lav	118	781	109	791	57	843

Tabell (22) Forvirringsmatrise for modellene trent på ISTFT data i frekvensområdet [500-5k] Hz.

STFT 1



Figur (71) Tapsfunksjoner for trening (blå) og validering (oransje) for 2D CNN modellen trent og validert på STFT 1 data i frekvensområdet [500-5k] Hz.



Figur (72) Tapsfunksjoner for trening (blå) og validering (oransje) for MLP modellen trent og validert på STFT 1 data i frekvensområdet [500-5k] Hz.

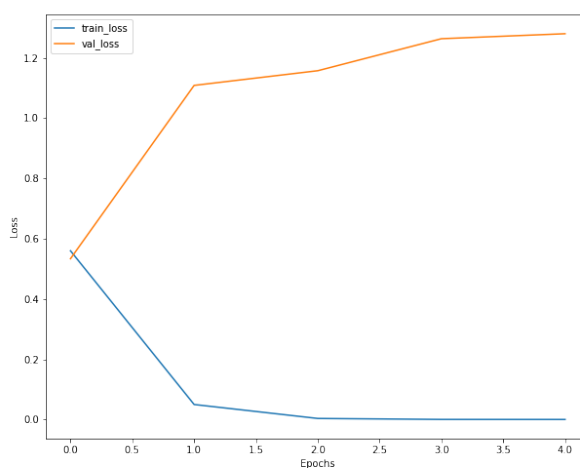
STFT 1								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
2D CNN	0,54	0,55	0,55	0,55	5(16)	500	-	-
MLP	0,69	0,71	0,78	0,71	5	10	-	-

Tabell (23) Ytelsesberegninger for 2D CNN og MLP modellene trent på ISTFT data i frekvensområdet [500-5k] Hz.

	2D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	271	629	410	490
Sann lav	269	631	37	863

Tabell (24) Forvirringsmatrise for modellene trent på STFT 1 data i frekvensområdet [500-5k] Hz.

STFT 2



Figur (73) Tapsfunksjoner for trening (blå) og validering (oransje) for 2D CNN modellen trent og validert på STFT 2 data i frekvensområdet [500-5k] Hz.

STFT 2								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
MLP	0,68	0,71	0,78	0,71	5	10	-	-

Tabell (25) Ytelsesberegninger for MLP modellen trent på STFT 2 data i frekvensområdet [500-5k] Hz.

	MLP	
	Pred høy	Pred lav
Sann høy	398	502
Sann lav	28	872

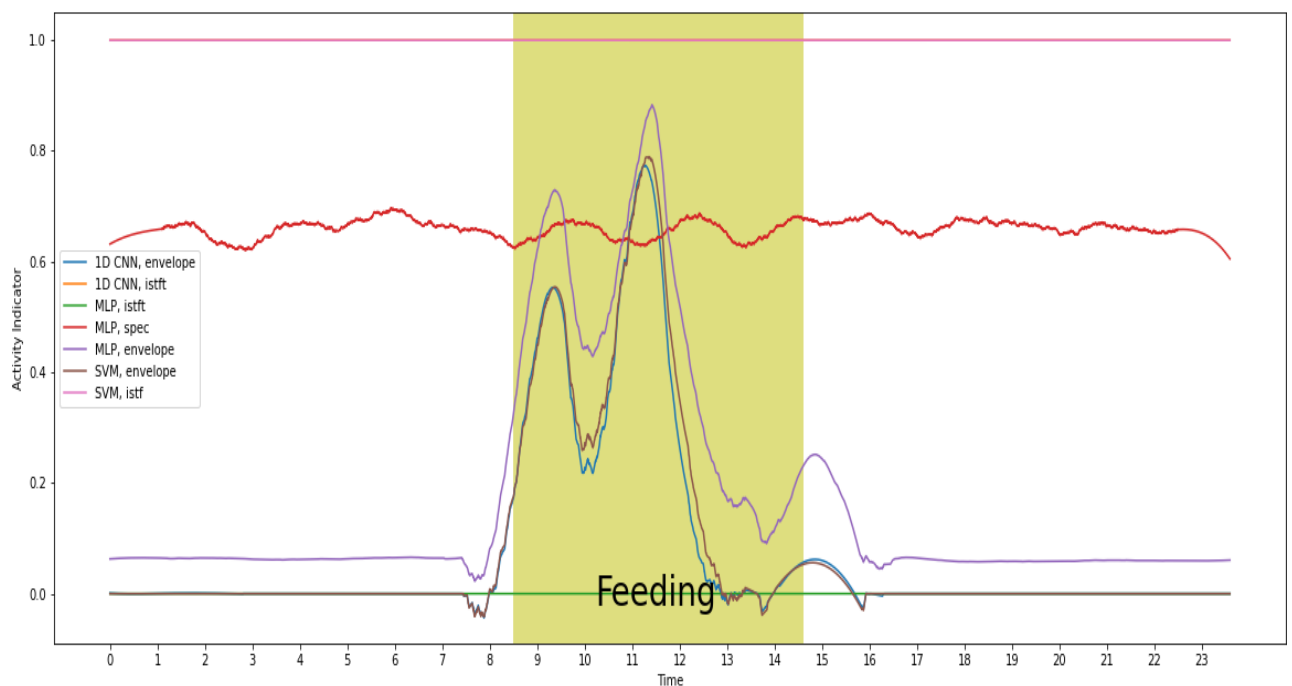
Tabell (26) Forvirringsmatrise for modellen trent på STFT 2 data i frekvensområdet [500-5k] Hz.

4.2.4 Test

Kun modeller med nøyaktighet over 85% fra kryssvalideringen ble brukt for test. For datasettene filtrert med [100-1k] Hz båndpassfilteret er det 1D CNN trent på envelope og ISTFT, MLP trent på envelope, ISTFT og STFT 1 og SVM trent på envelope og ISTFT. For datasettene filtrert med [500-4k] Hz er det 1D CNN trent på envelope, ISTFT og PCA, MLP trent på envelope og PCA og SVM trent på envelope, ISTFT og PCA.

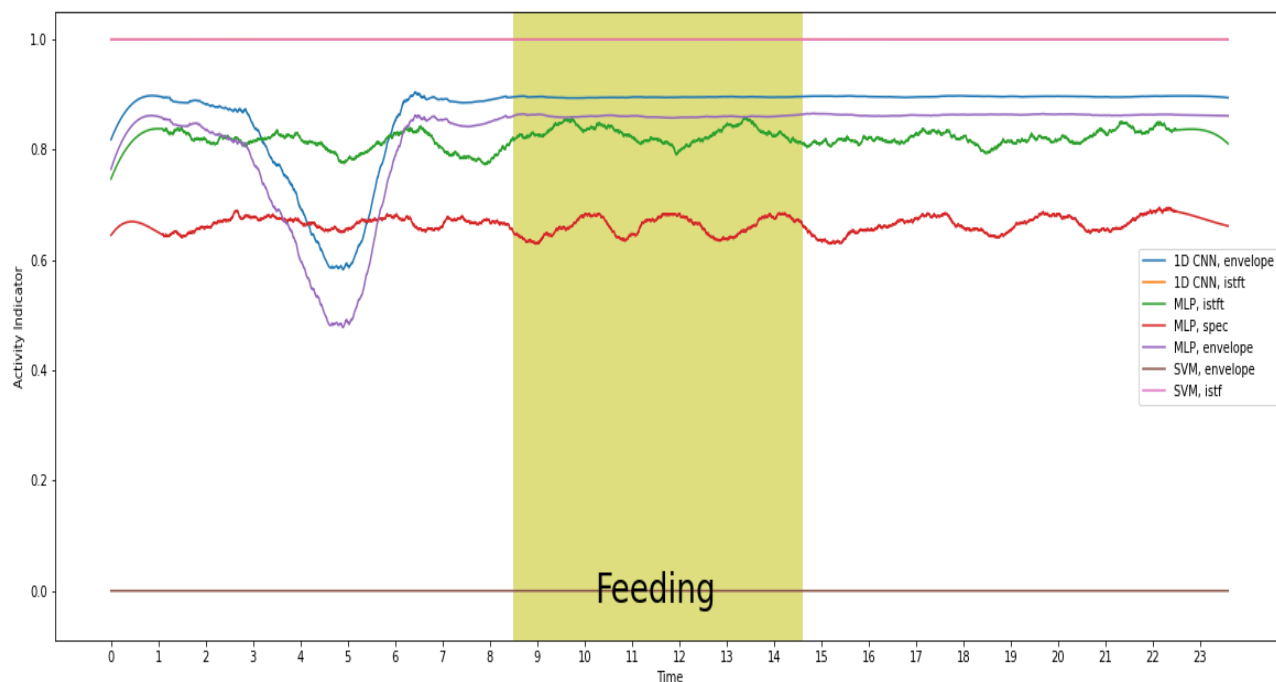
Blant aktivitetsindikatorene var testsettet splittet, lyddagene 19.12.2021 fra merd 3 og 25.12.2021 fra merd 7 ble brukt for test av modellene trent på [100-1k] Hz filtrert data. Lyddagene 19.12.2021 fra merd 7 og 25.12.2021 fra merd 3 var testdagene for modellene trent på [500-4k] Hz filtrert data.

Figur 74 er sannsynlighetsplottet for prediksjon av aktivitet den 19. desember 2021 merd 3 med de syv modellene trent på [100-1k] Hz filtrert data med nøyaktighet over 85%.



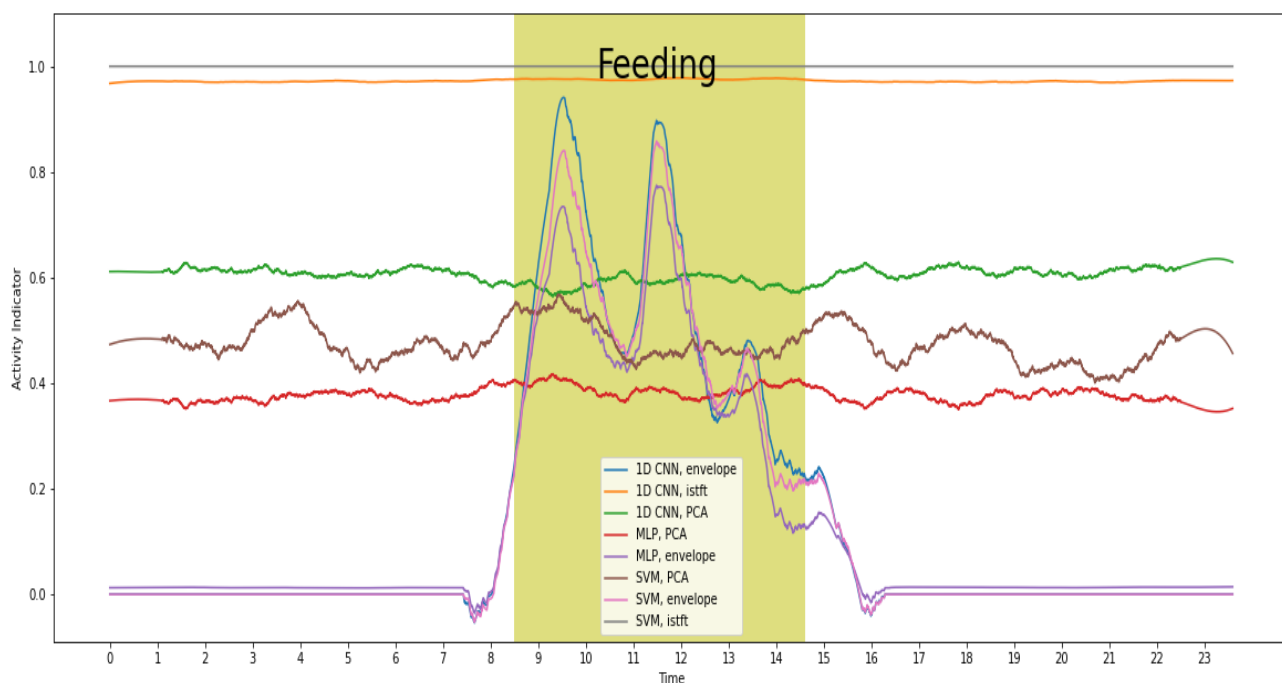
Figur (74) Filtrert prediksjon av aktivitet for lyddag 19.12.2021 merd 3, med modeller trent på [100-1k] Hz filtrert data med nøyaktighet over 85%.

Samme modeller er brukt for å teste på lyddagen 25. desember merd 7, sannsynlighetsplottet er vist i figur 75



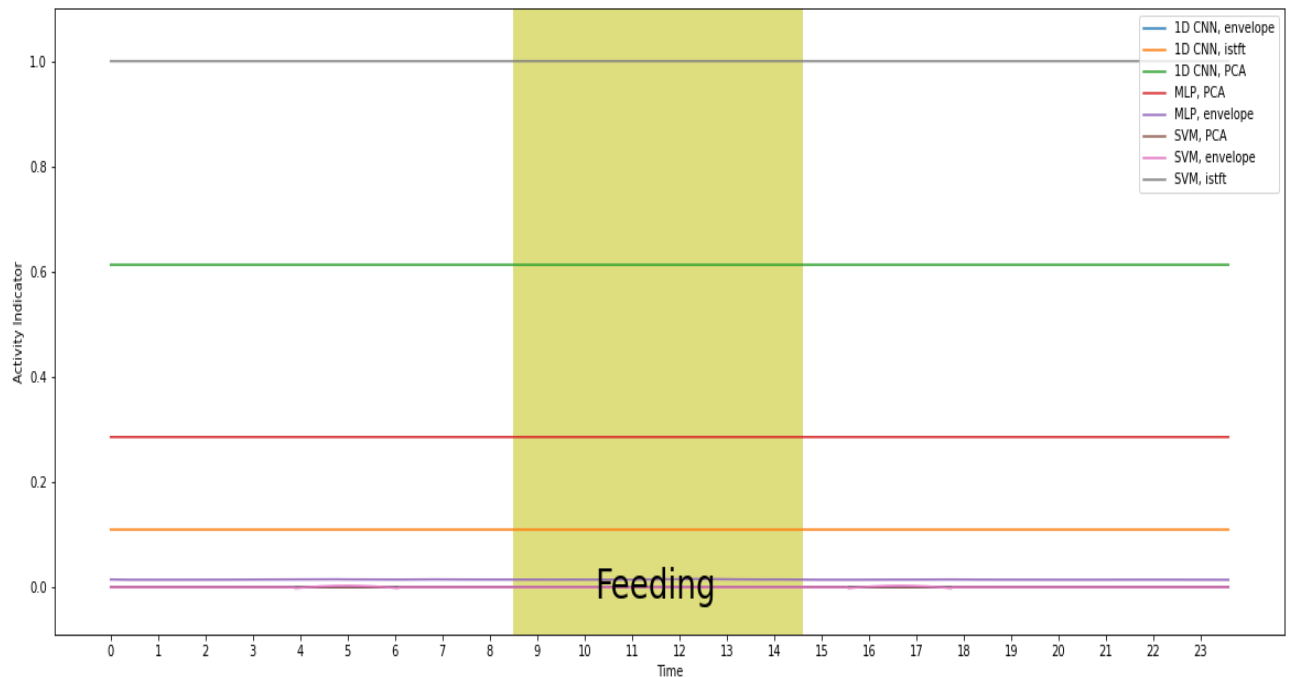
Figur (75) Filtret prediksjon av aktivitet for lyddag 25.12.2021 merd 7, med modeller trent på [100-1k] Hz filtret data med nøyaktighet over 85%.

Figur 76 er sannsynlighetsplottet for prediksjon av aktivitet den 19. desember 2021 merd 7 med de åtte modellene trent på [500-4k] Hz filtret data med nøyaktighet over 85%.



Figur (76) Filtret prediksjon av aktivitet for lyddag 19.12.2021 merd 7, med modeller trent på [500-4k] Hz filtret data med nøyaktighet over 85%.

Samme modeller er brukt for å teste på lyddagen 25. desember merd 3, sannsynlighetsplottet er vist i figur 77



Figur (77) Filtrert prediksjon av aktivitet for lyddag 25.12.2021 merd 3, med modeller trent på [500-4k] Hz filtrert data med nøyaktighet over 85%.

4.2.5 Generaliserte modeller

Med et formål om å generalisere modellene MLP og 1D CNN modellene trent på envelope data, ble modellene trent med mer data. 1440 ekstra samples ble lagt til i treningssettet fra dager rundt dager i testsettet. Datasettet ble filtrert med begge filtre fra tidligere, separat. Samme hyperparametere fra tidligere ble brukt.

Tabell 28 er ytelsesberegningene og 27 er forvirringsmatrisene for modellene trent på data filtrert med [100-1k] Hz båndpassfilteret.

Envelope								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
1D CNN	0,89	0,89	0,89	0,89	26	2	-	-
MLP	0,87	0,87	0,87	0,87	15	10	-	-

Tabell (27) Ytelsesberegninger for mer generaliserte 1D CNN og MLP modeller trent på Envelope data i frekvensområdet [0-1,25k] Hz.

	1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	1435	185	1454	166
Sann lav	182	1438	259	1361

Tabell (28) Forvirringsmatrise for mer generaliserte 1D CNN og MLP modeller trent på Envelope data i frekvensområdet [0-1,25k] Hz.

Tabell 30 er ytelsesberegningene og 29 er forvirringsmatrisene for modellene trent på data filtrert med [500-4k] Hz båndpassfilteret.

Envelope								
Modell	F1	Tilbakekall	Presisjon	Nøyaktighet	Epok	Gruppestørrelse	C	Kernel
1D CNN	0,91	0,91	0,91	0,91	22	20	-	-
MLP	0,87	0,87	0,87	0,87	15	10	-	-

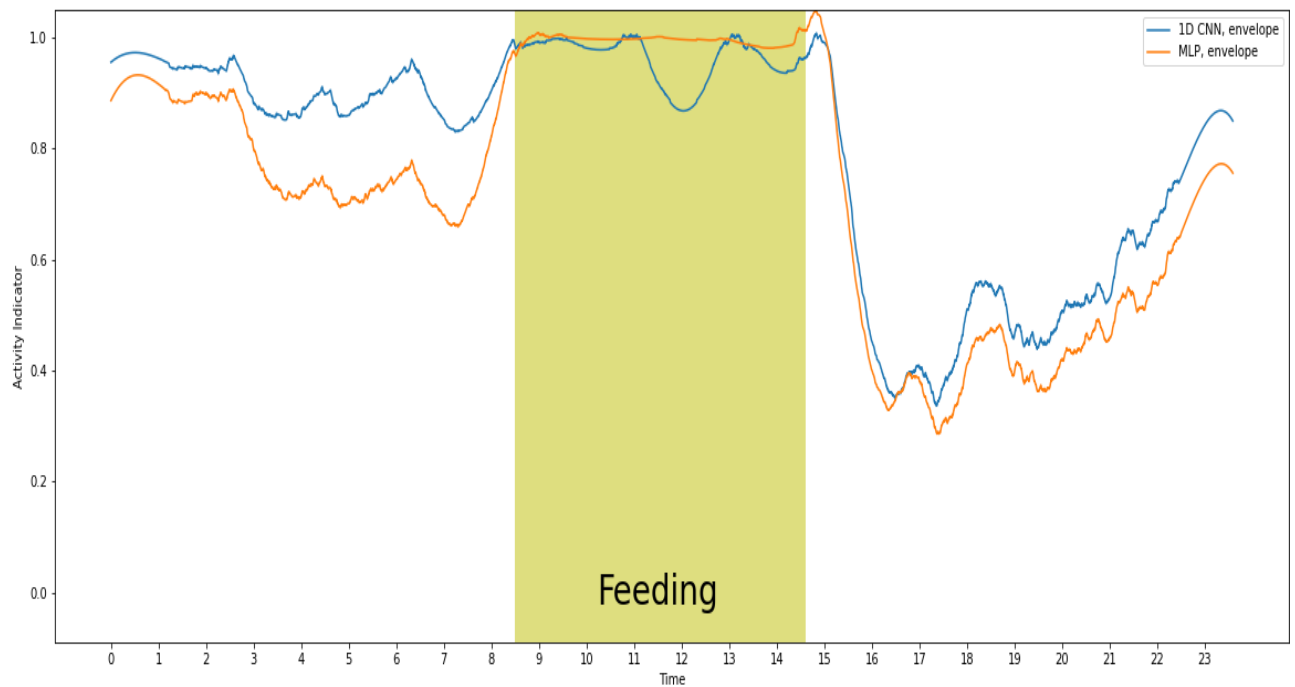
Tabell (29) Ytelsesberegninger for mer generaliserte 1D CNN og MLP modeller trent på envelope data i frekvensområdet [0-5k] Hz.

	1D CNN		MLP	
	Pred høy	Pred lav	Pred høy	Pred lav
Sann høy	1451	169	1386	234
Sann lav	136	1484	189	1431

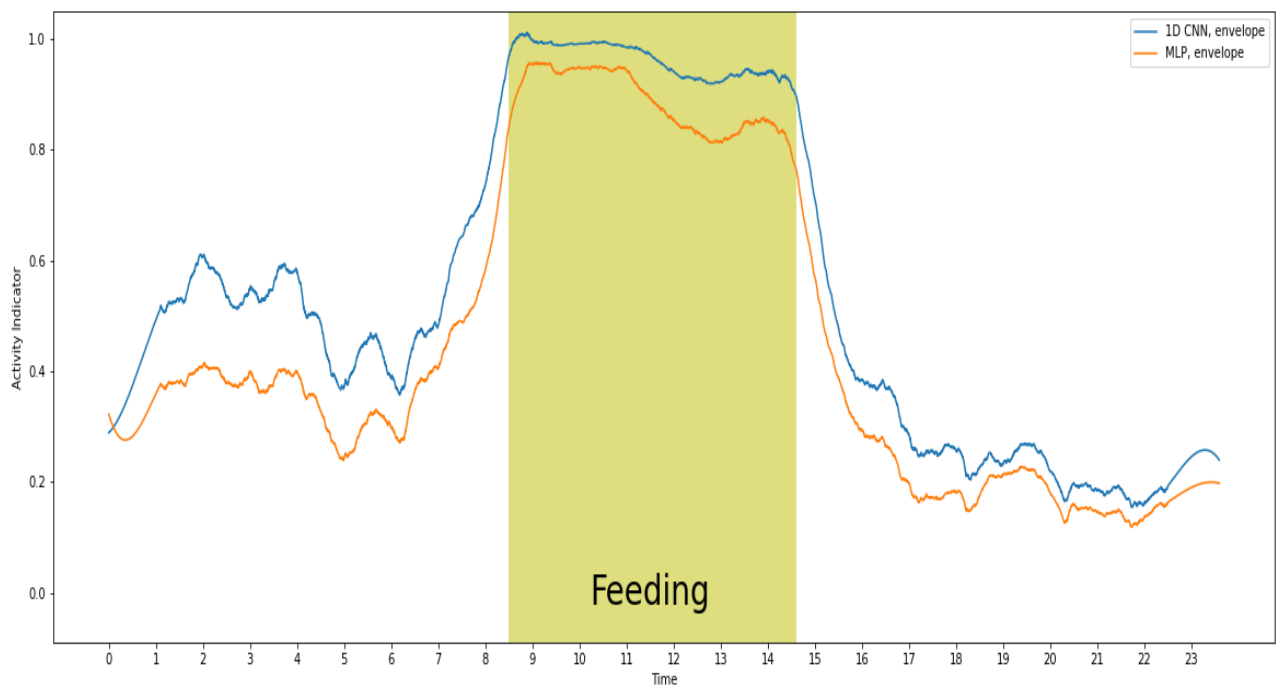
Tabell (30) Forvirringsmatrise for mer generaliserte 1D CNN og MLP modeller trent på envelope data i frekvensområdet [0-5k] Hz.

Det er testet på de fire felles lyddagene som er i testsettet for både lydmiljøindikatoren og aktivitetsindikatoren. Prediksjonsplott for lyddag 19.12.2021 fra merd 3 vist i figur 78 og

25.12.2021 fra merd 7 vist i figur 79, fra modellene trent på envelope data i frekvensområdet [0-1k] Hz.

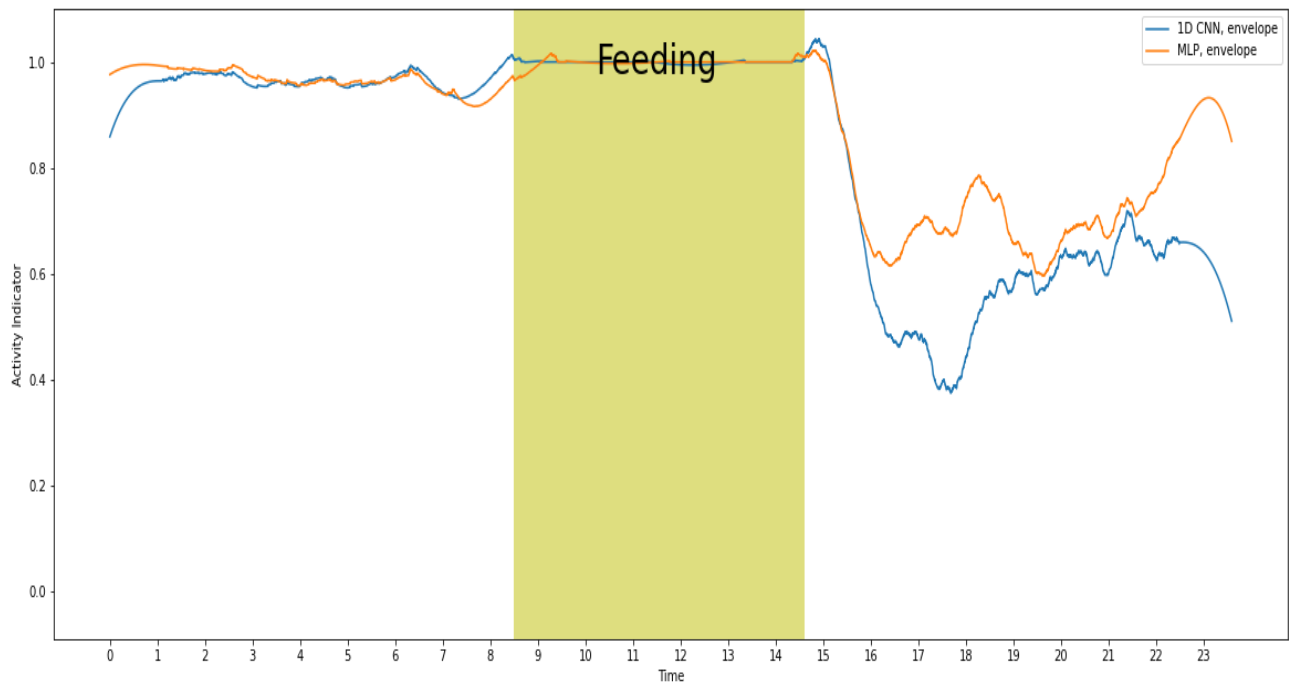


Figur (78) Filtret prediksjon av aktivitet for lyddag 19.12.2021 merd 3, med modeller trent på [100-1k] Hz filtrert generalisert data med nøyaktighet over 85%.

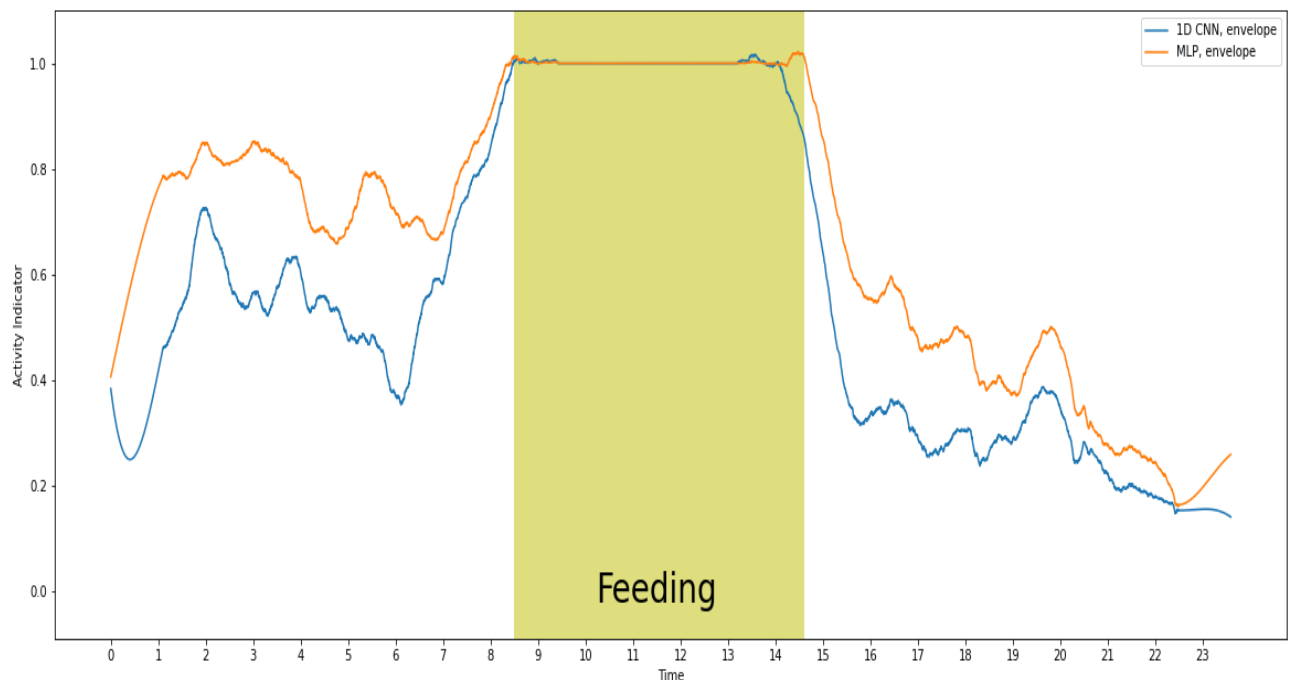


Figur (79) Filtret prediksjon av aktivitet for lyddag 25.12.2021 merd 7, med modeller trent på [100-1k] Hz filtrert generalisert data med nøyaktighet over 85%.

Prediksjonsplott for lyddag 19.12.2021 fra merd 7 vist i figur 80 og 25.12.2021 fra merd 3 vist i figur 81, fra modellene trent på envelope data i frekvensområdet [0-5k] Hz.



Figur (80) Filtret prediksjon av aktivitet for lyddag 19.12.2021 merd 7, med modeller trent på [500-4k] Hz filtrert generalisert data med nøyaktighet over 85%.



Figur (81) Filtret prediksjon av aktivitet for lyddag 25.12.2021 merd 3, med modeller trent på [500-4k] Hz filtrert generalisert data med nøyaktighet over 85%.

5 Diskusjon

Sensorsystemer med kontinuerlig målinger er til hjelp for overvåking av laks i oppdrettsanlegg, og maskinlæring kan effektivisere analyseprosessen til slike systemer. Men er maskinlæring mulig å bruke i denne sammenhengen?

Maskinlæringsalgoritmer er ofte basert på at det er et stasjonært miljø [41]. Det vil si at systemet ikke kan reagere på nåværende prediksjoner om sin fremtidige tilstand. I havet kan alt fra bølger til regn kan påvirke lydmiljøet, som igjen kan påvirke modellens prediksjoner. Til tross for det, ble det vist i evaluering av modellene at alle klassene, støy, natt, fôring og aktivitet, skal være mulig å gjenkjenne ved bruk av maskinlæring. Det var i tillegg mulig å prediktere fôring på test av lyddager. Det kan tyde på at prosesser med særegne lydbilder skal kunne være mulig å ha en indikator på, selv om de opptrer i et lite stasjonært lydmiljø.

Før man i det hele tatt skal ta for seg et klassifiseringsproblem, er det nødvendig å vite hva det er man vil finne. I dette tilfellet ville man for eksempel finne aktivitetslyder av laks i merder, men det var uvisst hvordan lydbildet så ut eller lyden hørtes ut. Dermed var det valgt å utforske flere mulige representasjoner av lyd. Svakheten til dette mulighetsstudiet var at flere valg tatt i metoden var basert på antagelser uten kvalitetssikring av valgene, eller resultatet av valgene. For eksempel valget om å merke lydsegmenter rett før fôring som høy aktivitet, basert på antagelsen om at da var fisken aktiv. Hvert segment plukket ut burde blitt verifisert før det ble merket. Uten verifisering av samples, kan man ikke med sikkerhet si hva maskinen har lært seg på. Slike svake valg tatt tidlig i arbeidet ga mulige feil underveis.

Videre vil datasettet brukt for indikatorene og fordeling av trening og testsettene bli diskutert. Deretter følger diskusjon av valgte metoder for signalbehandlingen og maskinlæringen. Til slutt vil trening og test bli diskutert.

5.1 Datasettet

5.1.1 Treningssett

Treningssettet for lydmiljøindikatoren er ulik treningssettet for aktivitetsindikatoren fordi modellene trener på ulike klasser. Lydmiljøindikatoren trente på 14 t med stille natt, 15,17 t fôring og 13,5 t støy. Aktivitetsindikatorene trente på 2,5 t liten aktivitet og 2,5 t høy aktivitet. Fordi alle modellene var basert på veiledet læring, måtte treningssamplene merkes med en klasse.

På grunn av lite detaljerte logger i forhold til når ulike prosesser opptrådte, var spektrogrammene ekstra viktige. Spektrogrammene som var tilgjengelige var fra tidlig desember 2021, dermed var det kun disse dagene som kunne merkes og bli brukt til trening. Dette førte til et lite generalisert treningssett i første omgang. Lyddagene fra samme dag, men ulik merd, og lyddager etter hverandre kan minne om hverandre. Det kan blant annet komme av at værforholdet var likt og at like operasjoner ble gjort samme dag på ulike merder eller løpet av flere dager. Det ble dermed valgt å trene fire modeller på et mer generalisert treningssett for aktivitetsindikatorene. Treningssettet ble ekspandert med 1440 nye samples fordelt likt på begge klassene. Til sammen var det 1620 samples for hver klasse, som tilsvarer 4,5 t med lyd av lav aktivitet og 4,5 t av lyd med høy aktivitet.

For aktivitetsindikatoren er det antatt at fisken er aktiv rett før fôring og lite aktiv på kveldstid. Antagelsen var introdusert av SINTEF Ocean og er basert på ekte observasjoner. Men kredibiliteten til metoden valgt for merking på grunnlag av antagelsen, kan diskuteres. I dette tilfellet er alle samples for treningssettet valgt fra samme tidspunkt, rett før fôring og ved midnatt. Uten å dobbeltsjekke at fisken faktisk har vært mer aktiv før fôring på de utvalgte dagene. Det er kun sjekket at lyddagene er relativt stille på de tidspunktene ved bruk av spektrogrammene.

Merkingen er i tillegg gjort av egen analyse av spektrogrammene og er ikke verifisert av profesjonelle. Det er lyttet til noen segmenter ved merking av data, men ikke alle segmenter er blitt verifisert ved lytting. Det er dermed stor sannsynlighet for feil merking, noe som kan føre til feil trening i forhold til klassen. Riktig trening av maskinlæringsmodeller er såpass viktig at det ikke egner seg med feil merket data for veiledet læring. Alt modellen lærer seg er basert på det den får tilgang til, og om det på forhånd er stor usikkerhet på merkingen av samples, er det forventet å få usikkert resultat.

5.1.2 Testsett

Testsettet til lydmiljøindikatoren inneholdt flere lyddager enn for aktivitetsindikatoren. Dette var for å blant annet lage en baseline, og teste på en relativt støyet dag med tilgjengelig spektrogram for å sjekke om indikatoren fungerer best mulig. Totalt var testsettet for lydmiljøindikatoren på ni lyddager. Fire av disse lyddagene var testsettet som var til felles for begge indikatorene. Formålet med et felles testsett var å kunne se om det var en sammenheng mellom fiskens aktivitet og de tre klassene fra lydmiljøet.

Fordi det var valgt å ikke merke testsettene, var det kun en observasjonsevaluering som ble gjennomgått. Konsekvensen av at testdataen ikke er merket er at det ikke er mulig å regne på ytelse, slik det er gjort for treningssettet. Valget om å ikke merke testdata var blant annet fordi det var tidskrevende. I motsetning til treningsdataen, skulle ikke testsegmenter velges, hver test skulle gjennomføres på en hel lyddag. Det å merke 24 t med enten 1440 samples per dag for lydmiljøindikatoren eller 8640 samples per dag for aktivitetsindikatoren, ble nedprioritert. Det ble heller prioritert å ha et større testsett for å kunne sammenligne testresultatene fra begge indikatorene mot hverandre.

Nedprioriteringen gir en indikasjon på at fremgangsmåten for oppgaven gikk i strid mot at det var en mulighetsstudie. Valget forklart ovenfor var basert på at indikatorene skulle fungere, noe som ikke er gitt i en mulighetsstudie. Uten ytelsesberegninger kan ikke testresultatene evalueres med fastsatte tall på ytelse. Muligheten for subjektive tolkninger på observasjon av testresultatet kan dermed komme frem.

Det felles testsettet av de fire lyddagenene var splittet mellom aktivitetsindikatorene for de to ulike frekvensområdene av interesse. Det ideelle hadde vært at det ble testet på alle fire lyddager, for alle aktivitetsindikatorer. Valget falt igjen på tidsforbruket og lagringsplass. Det tok lang tid å preprosessere en hel lyddag, særlig for de høyere frekvensene. Datamaskinen krasjet når koder ble kjørt parallelt, på grunn av lav RAM. Konsekvensen av at testsettet ble splittet er at det ikke er mulig å få sammenlignet prediksjonsplottene av de samme modellene trent på lik type data, men ulikt frekvensområde. Det hadde vært interessant å se om det var tydelige forskjeller, eller om de ga samme resultat.

Vanligvis er fordelingen på trening og test data på 80/20 av hele datasettet. Det er gitt at en modell er bedre trent jo mer data den trener på. Dette er igjen en indikasjon på at det er antatt at

modellene er gunstige, fordi den typiske fordelingen på 80/20 ikke var oppfylt. Dette blir videre diskutert i del seksjon 5.4.

5.2 Metode

Metoden for lydmiljøindikatoren hadde få steg. Det var kun ett utfall i metoden, det ga resultater av én maskinlæringsmodell som kun trente på lyd representert som MFCC. For aktivitetsindikatoren var det et søkelys på å se om det var mulig å kunne skille ut informasjon om laksens aktivitet ved bruk av signalbehandling. Maskinlæringsmodellene for begge statusindikatorerne hadde allerede ferdigvalgt arkitektur, hvor kun hyperparameterne ble endret på.

5.2.1 Signalbehandling for lydmiljøindikator

Figurene 16-18 er tilfeldige samples fra klassene natt, fôring og støy. Det er tydelig for øyet at de er ulike, som indikerer at modellen skal kunne klare å skille klassene. En MFCC representasjon approksimerer hørsel responsen til mennesker. Dette er til vår fordel fordi det blant annet skal skilles mellom fôring og støy lyder, som kan gjenkjennes av mennesker.

Tar man for seg oppgaven som en mulighetsstudie, er valget om å kun teste én lydrepresentasjon svak. Det var antatt at MFCC skal prestere best i denne sammenhengen basert på dens egenskaper, uten å teste om antagelsen stemmer. Valg av variablene vinduslengde, overlapp og antall koeffisienter var også fastsatt. Alternativt kunne det testes ulike variasjoner av MFCC, som for eksempel ulike valg av variablene nevnt ovenfor eller tillegg av dynamiske trekk som *delta* og *delta-delta* koeffisienter.

5.2.2 Signalbehandling for aktivitetsindikator

Signalbehandlingsmetoden er sterkt basert på lydbildet gitt i frekvensdomenet. Selv representasjonen av lyd i tidsdomenet (ISTFT), har gjennomgått en transformasjon til frekvensdomenet, for og så bli invers transformert. Det er mye nyttig informasjon i frekvensdomenet, men det er det ikke gitt at det er der den mest relevante informasjonen ligger. Særlig i en mulighetsstudie som dette. Til sammen produserte metoden fem ulike representasjoner av lyd, i to ulike frekvensområder.

Det var prøvd å unngå å lage en lydmåler ved å ikke forholde seg til amplitude eller magnitude

av signalene, ved å normalisere magnituden til 1 for STFT dataen. Med et formål om å se om det er endring i fasevariasjonene til lyd fra aktiv fisk i motsetning lite aktiv fisk. Ifølge figurene 3135 er det mulig å se at det ikke er et tydelig mønster i fasevariasjonene for hver av klassene. Dette gjaldt for både STFT 1 og 2 i begge frekvensområdene. Det var tydeligere forskjeller mellom klassene i magnitudespekteret for både STFT 1 og 2 som var filtrert med [100-1k] Hz båndpassfilteret, enn det var for spektrene for de høyere frekvensene. Dette gir et tegn på at høy aktivitet syns i magnitudespekteret, og at det kan være større skille mellom klassene blant lave frekvenser.

ISTFT av STFT 1 i begge frekvensområdene viste lite ulikheter for øyet, men det kan fremdeles være et mønster som kan detekteres av modellene. Det var utført en PCA transformasjonen av ISTFT dataen for å gjøre dette tydeligere, noe som viste seg var til nytte. Det kan sees at det er mulig å skille mellom klassene for de utvalgte segmentene, i begge frekvensområder. Det samme gjelder for envelopene, i det lave frekvensområdet hadde segmentet med høy aktivitet en amplitudetopp rundt 400 Hz, som ikke var synlig for segmentet med lav aktivitet. Det var videre antatt at segmentet med høy aktivitet skulle ha høyere amplitude for enn segmentet med lav aktivitet i det andre frekvensområdet, noe som stemte. Segmentet med lav aktivitet nådde ikke en amplitude over 7, hvor segmentet med høy aktivitet var oppe på 40.

5.2.3 Maskinlæring

Konsekvensen av at det var valgt arkitektur for modellene på forhånd, var at modellene ikke ble skreddersydd til de ulike datasettene.

For ANNene var treningsmetoden tredelt, som vist i figur 15. Første steget var å finne antall epoker og gruppestørrelse, steg to for evaluering og steg tre var selve treningen av modellen. Styrken av tredelingen er at epoke og gruppestørrelsene var justert for hver modell trent på hvert datasett. Evalueringen av modellene var basert på gjennomsnittet av 10 ytelsesberegninger, noe som ga et mer generelt bilde av hvordan modellen kunne prestere i prediksjon. Dette gir da muligheten til å kunne bruke hele treningssettet for å trene modellen, fordi evalueringen og justeringen allerede er gjort. En svakhet ved metoden er at det kan være store variasjoner mellom ytelsesberegningene for hver iterasjon i kryssvalideringen. Stor variasjon i ytelse kan vise til ustabilitet i treningen. I denne metoden er ikke beregningene for hver iterasjon lagret, og det kan dermed ikke kommenteres på variasjonen.

Bruk av rutenettsøk for hyperparametere for SVM krevde mye datamaskin kapasitet og lang tid. Dette var ikke tatt hensyn til før lenger ut i arbeidet, noe som førte til valg tatt på feil grunnlag. Evaluering av modellene ble utsatt og nedprioritert. Rutenettet utforsket kun én kernel, lineær, og $C = \{100, 200\}$. Det ga lite spillerom for tilrettelegging av modellarkitekturen for hvert datasett. Det tok mellom to til fire dager for rutenettsøk og kryssvalidering av hver SVM modell presentert i oppgaven. Hadde utfordringen blitt tatt hensyn til fra start, kunne det vært mulig å justere flere parametere for et bedre resultat.

5.3 Trening

Til tross for usikkerheten med merkingen av samples, både for aktivitetsindikatoren og lyd-miljøindikatoren, presterte modellene ganske greit i kryssvalideringen. Det viste seg at modellene generelt klarte å finne og lære mønstre i dataen, og at det skal være mulig å skille klassene fra hverandre. Ytelsesberegningene ga en indikasjon på at testen skulle gå bra, selv om flere av tapsfunksjonene ga en annen tolkning.

5.3.1 Tapsfunksjoner

Vi tar for oss modellene trent og validert på Envelope data filtrert med [100-1k] Hz båndpassfilter som et eksempel på dårlig trening. Figur 56 viser tapsfunksjonene for trening (blå graf) og validering (oransje graf) for 1D CNN modellen, og figur 57 viser tapsfunksjonene for MLP modellen. Valideringen er ganske ustabil for begge modellene, som gir en indikasjon på at treningen ikke er gunstig. Selv om tapsfunksjonen til treningssettet stabiliseres og går nærmere null, rekker ikke treningen å fortsette fordi valideringstapet har en drastisk variasjon fra en epoke til neste. Valideringstapet øker ikke med tiden, men veksler sterkt mellom 0.6 og over 1. Ustabiliteten i valideringstapet kan være et tegn på at klassene ikke er ulike nok for å skilles til enhver epoke. På grunn av variasjonen kan det forventes et dårlig resultat på evalueringen.

Et godt eksempel på overtrening er tapsfunksjonene til MLP modellen trent og validert på STFT 2 filtrert med [500-5k] Hz, vist i figur 73. Valideringstapet vokser når treningstapet minker. I motsetning, trener 2D CNN modellen brukt for lyd miljøindikatoren optimalt. Tapsfunksjonene vist i figur 19 gir ingen tegn på overtrening, men heller at treningen er ideell. Det samme gjelder for 1D CNN modellen trent på ISTFT vist i figur 60.

5.3.2 Kryssvalidering

Til tross for flere tegn på dårlig trening, presterte de fleste modellene bra under kryssvalideringen. Formålet med kryssvalidering var ikke å trene modellene, men heller få en oversikt over ytelsesberegninger.

2D CNN modellen trent på MFCC data for lydmiljøindikatoren hadde en nøyaktighet på 89% etter kryssvalideringen. Dette er et bra resultat, noe som var forventet ut ifra tapsfunksjonene. Presisjon, tilbakekall og F1-score for støy-klassen er under 89%, som er gjennomsnittet for alle ytelsesberegningene på tvers av klassene. Forvirringsmatrisen i tabell 5 viser at modellen presterer verst på støy. Det kan komme av at modellen er trent på ulik type støy. Definisjonen av støy er for bred og vag. Lyder produsert av en båt og et menneske skal i følge denne oppgaven være under lik klasse, noe som kan være vanskelig å lære på fordi lydene er ulike. Klassen tar i tillegg for seg støy under fôring og natt, noe som gir modellen muligheten til å for eksempel lære på mønstre typisk for fôring når den skal trene på støy.

Med 94% på alle ytelsesberegninger etter kryssvalidering, presterte 1D CNN modellen trent på ISTFT data med [100-1k] Hz filtrering, best av alle aktivitetsindikatorene. 14 av 24 modeller hadde en nøyaktighet over 85%. Ifølge tabell 31 ser man at alle modellene trent på envelope data, for begge filtreringene, hadde en nøyaktighet over 85%.

STFT 1 er magnitudeuavhengig, og dens invers er i tidsdomenet. Det vil dermed bety at ISTFT representerer lyden i form av amplitude over tid. Envelope data representerer amplitude over frekvens. PCA var en dimensjonalitetsreduksjon av ISTFT, som vil si at dataen er amplitudeuavhengig. Tabell 31 viser at amplitudeuavhengig data presterer best, både i tids og frekvensdomenet. Det vises også at en tre komponents PCA transformasjon fungerte bedre for data filtrert med [500-4k] Hz båndpassfilteret.

Modell	Representasjon av lyd	Filtrering
SVM	Envelope	[100-1k] Hz
MLP	Envelope	[100-1k] Hz
1D CNN	Envelope	[100-1k] Hz
SVM	Envelope	[500-4k] Hz
MLP	Envelope	[500-4k] Hz
1D CNN	Envelope	[500-4k] Hz
SVM	ISTFT	[100-1k] Hz
MLP	ISTFT	[100-1k] Hz
1D CNN	ISTFT	[100-1k] Hz
1D CNN	ISTFT	[500-4k] Hz
SVM	PCA	[500-4k] Hz
MLP	PCA	[500-4k] Hz
1D CNN	PCA	[500-4k] Hz
MLP	STFT 1	[100-1k] Hz

Tabell (31) Modeller for aktivitetsindikatoren med nøyaktighet over 85%.

5.4 Test

5.4.1 Lydmiljøindikator

Kryssvalideringen av lydmiljøindikatoren ga en indikasjon på at den klarer å skille godt mellom fôring, natt og støy, noe som også var tydelig ved sammenligning av prediksjonsplottene og deres spektrogram og logg fra dagen. Ifølge spektrogrammet for eksempelet av en relativt støyfri dag, vist i figur 7, er det tydelig at fôringen varer fra ca. 08:20-13:03. Prediksjonsplottet, i figur 21, for den samme dagen viser at sannsynligheten for fôring øker etter kl. 07:00, når en topp rundt kl.09:00 og minker mellom kl. 12:00-14:00. På spektrogrammet er det en lyd som opptrer mellom 06:23 og 06:40, dette er mest sannsynlig støy toppen på figur 21 rundt kl. 06:00.

I denne oppgaven er det ikke kun interessant å vite hva som er mest sannsynlig til enhver tid, men om modellen for eksempel kan detektere støy til tross for at fôring dominerer. Dette kan sees i figur 21. Fra kl. 11:00-14:00 er det et par støytopper over 20%, disse kan komme av den periodiske lyden på 2k Hz som opptrer rundt samme tid på spektrogrammet. Lignende

sammenhenger kan sees mellom spektrogrammet for dagen som er presentert som er mer støyet dag i figur 8, og tilhørende prediksjonsplott.

Det virker som om at modellen antar at stillhet er natt. I desember 2021 varierte daglengdene mellom 4t og 21 min til 5t og 11 min. Sammenlignes plottene med lengde på dagene i desember, kan man tro at modellen klarer å gjenkjenne stille natt. Vi tar for oss prediksjonsplottet av 25.12.2021 fra merd 3 vist i figur 77. Solnedgangen var kl. 14:37, som kan stemme med plottet. Men som nevnt i metoden, var natt i denne oppgaven definert av nautisk tussemørke, som var kl. 17:00. Det forsterker teorien om at modellen lærer stillhet og ikke stille natt.

5.4.2 Aktivitetsindikator

Det er tydelig at aktivitetsindikatorne oppfører seg som lydmålere og ikke aktivitetsindikatorer. Den første tanken var at for en stille dag skulle det være aktivitetstopper før og kanskje under fôring, og at laksen ellers var lite aktiv. Tanken var basert på observasjoner gjort under lytting til ulike lydfiler. Det var mer plasking før og under fôring enn på kvelden.

25.12.2021 var en relativt rolig dag. For nå tar vi for oss at høy aktivitet er utgangsverdier over 0,5 på plottet, og lav aktivitet er under 0,5. Alle modellene brukt for å teste denne dagen, både fra merd 3 og 7, predikerte enten kun høy eller lav aktivitet. Selv variasjonene som fant sted i figur 75, var minimale og har ingen klar forklaring. Figur 77 viser at modellene hadde likt utgang for alle testsamples. Dette ga en mistanke om at lyddagen 25.12.2021 fra merd 3 ikke inneholdt lyd. Det ble undersøkt ved lytting til filene, og stemte ikke. 19.12.2021 var det notvask i de ulike merdene på Rataren. Notvasken opptrådte under fôringen, og er synlige i figur 74 som toppene mellom kl. 09:00 og 12:00, og fra 09:00 til 15:00 i figur 76.

Det er verdt å merke seg at modellene ikke reagerer på lyden av fôring, men kun notvask. Og at det kun er modellene som er amplitude avhengige, altså envelope data, som reagerer på notvask lydene. De ble dermed valgt for videre test av modellene som er trent på et større og mer generalisert treningssett. Det var ikke valgt å trene SVM modellen på nytt fordi det var for tidskrevende.

Ved videre test av modellene som ble trent på et mer generalisert treningssett viser det seg at modellene oppførte seg annerledes. Modellene reagerte på fôring lyder, og så ganske like ut på tvers av lyddagene. Antar vi at lyddag med lik dato, men fra ulik merd er like, vil man se at

det er lite forskjeller mellom plottene. Noe som kan si noe om at det ikke er store forskjeller på prestasjon på prediksjon for modellene trent på data med ulike frekvensområder. Her kan vi i tillegg argumentere for at det ikke var nok å generalisere modellene for å få et mer nøyaktig prediksjonsplott.

5.4.3 Troverdighet av prediksjonsplott

Prediksjonsplottene er filtrert med Savgol filteret og det er klare artifakter som oppstår. Som nevnt i teorien er filteret sensitivt mot sterk ulineæritet, noe som kan sees i de ufiltrerte plottene. Som sett i figur 20, oppstår det tilfeller der den ene klassen har høy sannsynlighet for et par segmenter etter hverandre, og så går plottet rett ned. Når dette oppstår mange ganger etter hverandre kan artifakten sett på figur 25 rundt kl. 00:00 og kl. 23:00 for natt og støy grafene, oppstå. Det blir plottet en lokalt topp eller bunnpunkt som egentlig er en artifakt av filteret. Dette kan unngås ved å minimere orden eller øke vinduslengde. Vinduslengdene valgt var rundt 10% av antall samples, som i utgangspunktet skal være bra nok. Det er krevende å velge filterparametere for å unngå artifakter i lange tidsserier med stor dynamikk. Om større vinduslengde velges vil sannsynlighetsplottet miste mer informasjon om variasjoner som oppstår i det ufiltrerte plottet.

Sannsynlighetsplottet er i tillegg usikkert. Til tross for at det er valgt en sigmoid aktiveringsfunksjon som utgang, er det ofte at summen av sannsynlighetene for klassene ved et tidspunkt er lik 1. Dette fører til at modellen antar at det enten er det ene eller det andre, enn at flere ting kan oppstå samtidig. For eksempel prediksjonsplottet på figur 24, vi vet at fôring varer fra ca. 08:30-14:30 ut ifra logger fra den dagen. Men fordi støyet fra notvasken dominerer rundt den tiden, antar modellen at fôring og støy ikke kan oppstå samtidig med lik sannsynlighet.

5.5 Fremtidig arbeid

Det er mye som kan og burde forskes videre på i dette feltet. Slike statusindikatorer kan fremme både friskevelferd og økonomisk vekst i oppdrettsnæringen. Forslag til fremtidig arbeid kan være:

- Forske på lakselyder i merder for å vite hva det er man skal detektere.
- Verifisering av samples av eksperter, eller produsere en blåkopi for hver klasse.
- Ta utgangspunkt i metodene brukt for aktivitetsindikatorerne med annen opptaksmetode, for eksempel beamforming av flere hydrofoner for å unngå uønskede lyder.
- Trene på et mye større treningssett, for eksempel opptak fra et helt år. Dette kan gjøre maskinlæringsmodellene robuste mot lydvariasjoner i værforhold.
- Produsere maskinlæringsmodeller som kan gjenkjenne andre kjente operasjoner på oppdrettsanlegget, som for eksempel notvask.

6 Konklusjon

Statusindikatorene utforsket i denne oppgaven kan være til god hjelp for oppdrettsnæringen. Metodene brukt i oppgaven for å produsere indikatorene var et godt utgangspunkt, men er ikke en ferdigstilt teknisk løsning på problemet. Evalueringen av modellene i kryssvalideringen tyder på at det skal være mulig å gjenkjenne støy, fôring, natt og aktivitet med maskinlæringsmodeller. 15 av 25 modeller i oppgaven hadde en nøyaktighet over 85% i kryssvalideringen. Ved test av lydmiljøindikatoren viser det seg at det er mulig å prediktere den kjente operasjonen fôring i oppdrettsanlegg. Det er usikkert om lydmiljøindikatoren predikterer natt slik det er ønsket, det er en mistanke om at indikatoren antar at stillhet er lik natt. Til tross for gode resultater på kryssvalideringen av aktivitetsindikatorene, tyder det på at de amplitudeavhengige modellene i frekvensdomenet fungerte som lydmålere ved prediksjon. Arbeidet utført i denne masteroppgaven burde bli videreført og brukt som inspirasjon til utvikling av lignende statusindikatorer. Ved fremtidig arbeid burde det fokuseres på reduksjon av uønskede lyder under opptak, og å produsere modeller som er robuste mot lydvariasjoner basert på blant annet endring i værforhold.

Referanser

- [1] Terje Bolstad og Svein-johan Hansen Bernt Bjerkestrand. *Akvakultur - vg2 - havbruk i Norge*. Vett og Viten, 2013.
- [2] Norsk Fiskeoppdrett AS. *Tidenes beste år for norsk sjømateksport: - Laks var i særklasse*. <https://www.kyst.no/article/sjoemateksporten-passerte-120-milliarder-kroner-i-fjor/>. Hentet: 02.04.2022.
- [3] Norges forskningsråd. *Development of an acoustic status indicator for soundscape monitoring in aquaculture: active and passive sound making in Atlantic salmon*. <https://prosjektbanken.forskningsradet.no/project/FORISS/280512?Kilde=FORISS&distribution=Ar&chart=bar&calcType=funding&Sprak=no&sortBy=date&sortOrder=desc&resultCount=30&offset=0&TemaEmne.2=Havbruks-+og+fiskeriteknologi>. Hentet: 02.04.2022.
- [4] Fiskeri og havbruksnæringens forskningsfinansiering. *Teknologiske verktøy for objektiv dokumentasjon av velferd hos laksefisk ved håndteringsoperasjoner (OWITOLS)*. <https://www.fhf.no/prosjekter/prosjektbasen/901594/>. Hentet: 02.04.2022.
- [5] Tore Kristiansen. *Tema: Fiskevelferd*. <https://www.hi.no/hi/temasider/akvakultur/fiskevelferd>. Hentet: 02.04.2022.
- [6] Thomas Rossing. *Introduction to Acoustics. I Rossing T. Springer Handbook of Acoustics*. Springer, New York, NY, 2007.
- [7] Truls Gjestland. *lyd*. <https://snl.no/lyd>. Hentet: 04.04.2022.
- [8] Arthur Popper, Anthony Hawkins, Richard Fay, David Mann, Soraya Bartol, Thomas Carlson, Sheryl Coombs, William Ellison, Roger Gentry, Michele Halvorsen, Svein Løkkeborg, Peter Rogers, Brandon Southall, David Zeddies, and William Tavolga. *Sound Exposure Guidelines*, pages 33–51. 05 2014.
- [9] William Wilcock, Kathleen Stafford, Rex Andrew, and Robert Odom. Sounds in the ocean at 1-100 Hz. *Annual review of marine science*, 6, 07 2013.
- [10] Thomas Rossing og William M. Hartmann. *Acoustic Signal Processing. I Rossing T. Springer Handbook of Acoustics*. Springer, New York, NY, 2007.

-
- [11] Abraham. Savitzky og M. J. E. Golay. *Smoothing and Differentiation of Data by Simplified Least Squares Procedures*. pages 1627–1639, 1964.
- [12] Ronald W. Schafer. *Homomorphic Systems and Cepstrum Analysis of Speech. I: Benesty, J., Sondhi, M.M., Huang, Y.A. Springer Handbook of Speech Processing*. Springer, Berlin, Heidelberg, 2007.
- [13] Aniruddha Bhandari. *Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization*. <https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>. Hentet: 04.04.2022.
- [14] Jemila Daush. *Detection of Smolt from Hyperspectral Images of Farmed Atlantic Salmon using Machine Learning*. Prosjektrapport i TFE4580, Institutt for elektroniske systemer, NTNU – Norges teknisk-naturvitenskapelige universitet, Desember 2021.
- [15] Zakaria Jaadi. *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. <https://builtin.com/data-science/step-step-explanation-principal-component-analysis>, 2021. Hentet: 24.03.2022.
- [16] Akash Dubey. *The Mathematics Behind Principal Component Analysis*. <https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b64>. Hentet: 24.03.2022.
- [17] Imad Basheer og M. N. Hajmeer. *Artificial Neural Network: Fundamentals, Computing, Design, and Application*. 2001.
- [18] Sidharta Misra, Hao Li, and Jibao He. *Machine Learning for Subsurface Characterization*. Elsevier, 2020. p. 114.
- [19] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, LLC, 2006. p. 325-336.
- [20] Python. *python*. <https://www.python.org/>. Hentet: 10.06.2022.
- [21] scikit learn. *scikit-learn Machine Learning in Python*. <https://scikit-learn.org/stable/>. Hentet: 10.06.2022.
- [22] Keras. *About Keras*. <https://keras.io/about/>. Hentet: 10.06.2022.

-
- [23] Jupyter. *JupyterLab Documentation*. <https://jupyterlab.readthedocs.io/en/stable/>. Hentet: 10.06.2022.
- [24] Python. *Frøya, Trøndelag, Norge — Soloppgang, solnedgang og dagens lengde, desember 2021*. <https://www.timeanddate.no/astronomi/sol/@3156429?maaned=12&year=2021>. Hentet: 10.06.2022.
- [25] Librosa. *librosa.load*. <https://librosa.org/doc/main/generated/librosa.load.html>. Hentet: 10.06.2022.
- [26] Scikit-learn. *sklearn.preprocessing.StandardScaler*. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>. Hentet: 10.06.2022.
- [27] SciPy. *scipy.signal.butter*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html>. Hentet: 10.06.2022.
- [28] SciPy. *scipy.signal.sosfiltfilt*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.sosfiltfilt.html>. Hentet: 10.06.2022.
- [29]
- [30] Librosa. *librosa.feature.mfcc*. <https://librosa.org/doc/main/generated/librosa.feature.mfcc.html>. Hentet: 10.06.2022.
- [31] NumPy. *numpy.fft.rfft*. <https://numpy.org/doc/stable/reference/generated/numpy.fft.rfft.html>. Hentet: 10.06.2022.
- [32] SciPy. *scipy.signal.stft*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.stft.html>. Hentet: 10.06.2022.
- [33] SciPy. *scipy.signal.istft*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.istft.html>. Hentet: 10.06.2022.
- [34] Scikit-learn. *sklearn.decomposition.PCA*. <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>. Hentet: 10.06.2022.
- [35] Francois Chollet. *Building powerful image classification models using very little data*. <https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>, 2016. Hentet: 10.06.2022.

- [36] Keras. *Optimizers*. <https://keras.io/api/optimizers/>. Hentet: 10.06.2022.
- [37] Jason Brownlee. *EarlyStopping*. <https://machinelearningmastery.com/k-fold-cross-validation/>. Hentet: 10.06.2022.
- [38] Keras. *A Gentle Introduction to k-fold Cross-Validation*. https://keras.io/api/callbacks/early_stopping/. Hentet: 10.06.2022.
- [39] SciPy. *scipy.signal.savgol_filter*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html. Hentet: 10.06.2022.
- [40] Scikit-learn. *sklearn.model_selection.GridSearchCV*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. Hentet: 10.06.2022.
- [41] Masashi Sugiyama og Motoaki Kawanabe. *Machine Learning in Non-Stationary Environments Introduction to Covariate Shift Adaptation*. 2012.

