



RAPPORT

Rapport aktivitet 2.2

Beskrivelse av tilgang til datakilder

7. okt 2024 Jon Åge Stakvik-Løvås

Kapittel 1 INTRODUKSJON

Denne delrapporten omhandler Realisering av potensialet i sensordata for forbedret effektivitet og fiskevelferd i landbasert oppdrett. Prosjektet er et samarbeid mellom Salmon Evolution og datainnsamlingsplattformen Clarify.

Rapporten gir en detaljert oversikt over hvilke datakilder Salmon Evolution har fått tilgang til og koblet seg på, og hvilke verktøy som er brukt for å oppnå dette.

Kapittel 2 RAMMEVERK FOR DATAHENTING

Dette kapittelet gir en oversikt over hvilke software og rammeverk Salmon Evolution har valgt å bruke for å muliggjøre det å hente og lagre data fra ulike systemer.

I oppdrettsnæringen er det en realitet at det finnes svært mange leverandører og mye spesiallagede løsninger. En konsekvens av dette er at for å hente og lagre data så må man bygge et fleksibelt rammeverk som kan hente data fra ulike systemer med ulik struktur.

2.1. To hovedtyper av data

Salmon Evolution definerer data på to hovedmåter; tidsseriedata og relasjonelle data.

Tidsseriedata er data som består av en numerisk verdi og et tidspunkt. Dette kan være f.eks. sensorikdata. Tidsseriedata blir raskt svært store mengder og det er utfordrende å lage rammeverk som håndterer dette på en robust og god måte.

Relasjonelle data består av flere kolonner som har en eller flere relasjoner til hverandre. Et eksempel på dette er vaksinasjonsdata hvor mye av dataen ikke er tid eller en numerisk verdi, men hvilken type vaksine det var, hvilken dato var vaksinen produsert, hvem var produsenten. Lagring av relasjonelle data krever et mindre komplisert rammeverk og oppetid på løsningen, men heller mer arbeid i selve struktureringen av dataene.

Salmon Evolution har valgt å samarbeide med Clarify for håndtering av tidsseriedata. Clarify har utviklet et rammeverk hvor de på en robust og god måte kan ta imot store mengder data uten at kunden må bruke tid på dette. Dataen som kommer inn til Clarify kan raskt visualiseres av alle brukere og det går derfor raskt fra innsamling av data til bruk av data.

For håndtering av relasjonelle data bruker Salmon Evolution en løsning bygget på PostgreSQL. Noen av dataene fra de relasjonelle databasene kan videre strømmes til Clarify der hvor man ønsker å se sammenhenger med andre tidsseriedata.

2.2. Valg av software for datahenting

Et viktig prinsipp for Salmon Evolution er å velge flest mulig open source løsninger for å hente ut data. Open Source løsninger gir stort eierskap til løsningene som bygges

og gjør det også enklere å bytte løsninger senere om man finner bedre eller andre metoder for å gjøre noe.

Følgende software og rammeverk har Salmon Evolution benyttet seg for datahentning og lagring:

- **Python:** Python er et høynivå, allsidig programmeringsspråk som er kjent for sin lesbarhet og enkle syntaks. Det brukes bredt innen utvikling av programvare, dataanalyse, maskinlæring og webutvikling på grunn av sitt omfattende økosystem av biblioteker og rammeverk. Hjemmeside: <https://www.python.org/>
- **Mosquitto:** Mosquitto er en lettvekts, åpen kildekode MQTT-meldingsmegler (message broker) som muliggjør kommunikasjon mellom enheter i tingenes internett (IoT). Den implementerer MQTT-protokollen, som er designet for å være effektiv og ressursbesparende, ideell for applikasjoner med begrenset båndbredde og strømforbruk. Hjemmeside: <https://mosquitto.org/>
- **Linode:** Linode er en skybasert infrastrukturleverandør som tilbyr virtuelle private servere (VPS) og andre skybaserte løsninger for utviklere og bedrifter. Det er kjent for sin pålitelighet, ytelse og brukervennlighet, og brukes ofte til webhotell, dataanalyse, applikasjonsutvikling og mer. Hjemmeside: <https://www.linode.com/>
- **Windmill:** Windmill er en åpen kildekode utviklerplattform og arbeidsflytmotor som forenkler opprettelsen av interne verktøy, arbeidsflyter og datadrevne applikasjoner. Den støtter flere programmeringsspråk som Python, TypeScript, Go, PHP, Bash, SQL og Rust, og tilbyr både lavkode- og kodebaserte utviklingsalternativer. Hjemmeside: <https://www.windmill.dev/>
- **Postgres:** PostgreSQL, ofte referert til som Postgres, er et kraftig, åpen kildekode-relasjonsdatabasesystem som bruker og utvider SQL-standard. Det er kjent for sin pålitelighet, skalerbarhet og støtte for avanserte funksjoner som komplekse spørringer, datastrukturer og utvidelser. Hjemmeside: <https://www.postgresql.org/>
- **GitLab:** GitLab er en plattform for DevOps og kildekodeadministrasjon som tilbyr funksjoner for versjonskontroll, kontinuerlig integrasjon og kontinuerlig levering (CI/CD). Den gir utviklingsteam mulighet til å samarbeide effektivt gjennom hele programvareutviklingssyklusen, fra kodeutvikling til testing og distribusjon.
- **Ignition:** Ignition, utviklet av Inductive Automation, er en universell plattform for industriell automasjon som brukes til SCADA (Supervisory Control and Data Acquisition), IIoT (Industrial Internet of Things), MES (Manufacturing Execution Systems) og mer. Det tilbyr verktøy for å koble til og integrere industrielle systemer, lage visuelle dashbord, og automatisere prosesser på en skalerbar måte, alt gjennom en fleksibel serverbasert modell som støtter ubegrenset antall klienter, enheter og tilkoblinger. Ignition er kjent for å være kryssplattform, bygget på åpne standarder som SQL, Python, OPC UA og MQTT, og kan brukes på både lokale servere og i skyen.

2.3. Erfaringer fra datainnsamlingsarbeidet

En av de store utfordringene Salmon Evolution hadde ved oppstart av prosjektet Salmosense var å håndtere de store mengdene sensordata som finnes. Dette gjelder spesielt dataene fra Ignition hvor det strømmes inn over 100 000 ulike datakilder, og mange av dem oppdateres flere ganger i minuttet. Denne utfordringen har langt på

vei blitt løst ved å ta i bruk Ignition sin mott modul 'mott transmission' og strøms data direkte til Clarify. Et pågående arbeid er å strukturere dataene som kommer inn til Clarify på en bedre måte slik at det blir enklere for personer som ikke kjenner anlegget i detalj å finne korrekt måling.

Det er også brukt betydelig med tid på å hente ut biomassedata og strøms data til Clarify. Her har noe av utfordringen vært at Fishtalk/AkvaGroup sitt API har vært under utvikling. Men i løpet av 2024 har dette blitt nærmere ferdig og klargjort som et eget produkt.

En annen erfaring vi har gjort oss er viktigheten av å ha en god relasjonell databaseløsning for å håndtere datatyper som ikke er tidsserier. Dette har vi utviklet for å håndtere blant annet screeningdata fra Pharmaq.

Kapittel 3 DATAKILDER

I delkapitlene nedenfor er datakildene Salmon Evolution foreløpig har koblet seg opp til oppsummert.

Hver datakilde oppsummeres basert på følgende hovedpunkter:

1. Datakildebeskrivelse og hvilken informasjon som hentes.
2. Hvordan er tilgang til datakilden oppnådd.
3. Konkret hvordan dataen er tilgjengeliggjort for oss og hvordan den ligger lagret.
4. Softwarerammeverk for henting av data fra originalsystem til samlet database.

3.1. Prosessdata – Ignition

3.1.1. Datakildebeskrivelse og hvilken informasjon som hentes
Ignition er prosesskontrollsystemet som brukes til overvåking og styring av nært hele produksjonsanlegget på Salmon Evolutions matfiskanlegg ved Indre Harøy. Dette inkluderer pumping av vann, oksygentilsetning av vann, oppvarming av vann og rensing av vann. I tillegg overvåkes og driftes mange andre undersystemer via Ignition som f.eks. slamanlegg. Dette inkluderer data om ventilåpninger, pumping, energi, vannnivå, oksygenforbruk, CO2 og andre vannkvalitetsmålinger osv.

Ignition er derfor en hub for svært mye driftsdata og mesteparten av dette samples flere ganger i minuttet og skaper svært store datamengder.

3.1.2. Hvordan er tilgang til datakilden oppnådd
Ignition er valgt av Salmon Evolution som vårt kjernesystem for drift av produksjonsanlegget. Dette er et system som Salmon Evolution eier selv og har tilgang til selv. Tilgangen til systemet har derfor kommet direkte fra innkjøpsstrategien.

Dataene som vi får tilgang til i Ignition kommer fra svært mange undersystemer. Ved oppbygning av anlegget ble alle kritiske prosesser knyttet opp til Ignition, og dermed ble også dataen gjort tilgjengelig. I ettertid har flere og flere undersystemer blitt knyttet opp til Ignition og dataen blitt gjort tilgjengelig.

I hovedsak brukes protokollen OPC-UA for kommunikasjon mellom undersystemer (PLSer) og Ignition, men i noen tilfeller brukes også MQTT.

3.1.3. Hvordan er data tilgjengeliggjort for oss og hvordan ligger den lagret

Dataen i Ignition blir håndtert og lagret på to måter.

- 1) Data som kommer inn til Ignition lagres på lokale databaser basert på en konfigurasjon (hvor ofte og hvilke data som skal lagres). Historiske data kan derfor lastes ut fra Ignition.
- 2) All data vi ønsker å sende strømmes direkte fra Ignition til Clarify ved bruk av mqtt.

3.1.4. Softwarerammeverk for henting og lagring av data

For å strøme data fra Ignition til Clarify brukes Ignition sin MQTT transmission modul: <https://docs.chariot.io/display/CLD80/MQTT+Transmission>. I Clarify er det satt opp en integrasjon til Clarify sin mqtt broker: <https://docs.clarify.io/developers/tools/mqtt-broker/>.

3.2. Føringssystemer – Laksesystemer

3.2.1. Datakildebeskrivelse og hvilken informasjon som hentes

Laksesystemer leverer føringssystem og brukes for all føring av fisk på matfiskanlegget til Salmon Evolution. Systemet inneholder all informasjon om utført mengde i hvert fiskekar, appetittfaktor, silonivå og så videre. I tillegg overvåkes driften av førtransport fra silo til føringssystem i fiskekar via Laksesystemer.

3.2.2. Hvordan er tilgang til datakilden oppnådd

Datatilgang er oppnådd via avtale med leverandør. Leverandør måtte sette opp en strømming av data fra deres server til en mottakende part administrert av Salmon Evolution.

3.2.3. Hvordan er data tilgjengeliggjort for oss og hvordan ligger den lagret

Laksesystemer kommuniserer mellom sine komponenter med protokollen MQTT. Denne dataen blir strømmet videre til Salmon Evolution og må håndteres i sanntid for å lagres. I tillegg blir historisk data lagret i interne databaser i Laksesystemer sin server på anlegget. Men den historiske dataen er kun tilgjengelig for SE å bruke via det leverte toppsystemet.

3.2.4. Softwarerammeverk for henting og lagring av data

Salmon Evolution lagrer data fra føringssystemet i Clarify. MQTT data strømmes fra Laksesystemer til en intern MQTT broker Salmon Evolution har satt opp. MQTT dataene fra Laksesystemer inkluderer alt som er mulig å lese av. Salmon Evolution gjør en mapping av et utvalg av data i vår broker og sender utvalget videre til Clarify via MQTT. Dokumentasjon på Clarify sin MQTT modul kan sees her: <https://docs.clarify.io/developers/tools/mqtt-broker/>

3.3. Pelletspilldata – ReelAppetite

3.3.1. Datakildebeskrivelse og hvilken informasjon som hentes
ReelAppetite er et system som gir en sanntidsmåling av fôrspill og appetitt i hvert fiskekar ved å telle en delmengde av pelleten som ikke blir spist. Dette er en unik teknologi tilpasset landbasert oppdrett.

Salmon Evolution henter ut informasjon om alle identifiserte pellet med nøyaktig tidspunkt for når det var detektert.

3.3.2. Hvordan er tilgang til datakilden oppnådd
ReelAppetite er et separat system som kjører på leverandøren ReelData sitt softwareinfrastruktur. Produktet kommer med et eget brukergrensesnitt hvor data fra fôringssystemet hentes, og fôringen styres i sanntid.

Salmon Evolution har gjort en avtale med ReelData for tilgang til rådata ved lesetilgang direkte fra databasene til ReelData.

3.3.3. Hvordan er data tilgjengeliggjort for oss og hvordan ligger den lagret

ReelData benytter seg av SQL basert lagring. Fôrspillinformasjonen ligger da lagret i separate SQL tabeller på ReelData sin softwareinfrastruktur. Salmon Evolution leser data fra disse tabellene og videresender til Clarify. Grunndataen som er lagret er hver eneste pellet som er talt med nøyaktig tidssample for når den ble detektert.

Dokumentasjon av databasen og databasestruktur fås direkte fra ReelData.

3.3.4. Softwarerammeverk for henting og lagring av data
Data leses fra ReelData sin database ved hjelp av et Python script. Det gjøres en spørring etter nye fôrspilldata og informasjonen lagres i et Pandas DataFrame. Denne spørringen gjøres jevnlig. Deretter lastes data direkte inn til Clarify ved å bruke Clarify sin modul pylarify som muliggjør direkte opplasting av data fra et Pandas DataFrame til Clarify.

3.4. Fiskevelferdsdata – Pharmaq og Patogen

3.4.1. Datakildebeskrivelse og hvilken informasjon som hentes
Det gjøres jevnlig screeninger av agens på fisk. Salmon Evolution har benyttet både Pharmaq og Patogen som analyseleverandører. Typisk informasjon som lagres og tilgjengeliggjøres er informasjon om hvilket agens som er undersøkt, hvilket vev prøven er tatt fra, dato, enhet (fiskekar) og prøveresultat som positiv/negativ og evt Ct verdi som sier grad av positivitet.

Per i dag benyttes Pharmaq som hovedleverandør og prøveresultat og informasjon hentes automatisk via Pharmaq sitt API gjennom analyseløsningen iWISE.

3.4.2. Hvordan er tilgang til datakilden oppnådd
Pharmaq har utviklet et API for å kunne gi datatilgang til kundene sine. Dette er en tjeneste man kan velge å betale for.

3.4.3. Hvordan er data tilgjengeliggjort for oss og hvordan ligger den lagret

Pharmaq har valgt å tilgjengeliggjøre dataen sin i et JSON format som består av flere «siden» av JSON dokumenter. For å få tilgang spør Salmon Evolution etter hver side av JSON dokumentene og deretter slås disse sidene sammen til et større dokument.

API dokumentasjon fås ved direktekontakt med Pharmaq.

3.4.4. Softwarerammeverk for henting og lagring av data

Data fra Pharmaq leses ved hjelp av et Pythonscript som kjøres regelmessig. I dette Pythonscriptet slås sidene i JSON dokumentet sammen og det totale JSON dokumentet lagres deretter i en PostgreSQL database internt i Salmon Evolution sine systemer. Deretter transformeres JSON dokumentet til en utvalgt tabellstruktur for alle prøveresultatene i en ny tabell. Denne tabellen slås videre sammen med historisk data fra Patogen for å få en totaltabell som oppsummerer all fiskehelseinformasjon.

Sluttresultatet er en SQL tabell som inneholder all informasjon fra alle screeninger og kan lastes inn i Excel eller tas videre til f.eks. Clarify.

3.5. Biologisk data – Fishtalk

3.5.1. Datakildebeskrivelse og hvilken informasjon som hentes

Fishtalk er Salmon Evolutions valgte biomassehåndteringssystem. I dette systemet håndteres biomassen og all data relatert til antall, vekt, dødelighet, biomasse, fôring, behandlinger, prøveuttak og så videre ligger lagret her. I tillegg beregnes det svært mange KPIer for oppfølging av vekst her.

Disse dataene blir hentet ut og tilgjengeliggjort sammen med driftsdata for temperatur, vannkvalitet og lignende fra Ignition.

3.5.2. Hvordan er tilgang til datakilden oppnådd

Fishtalk er utviklet av selskapet AkvaGroup. AkvaGroup har utviklet et API som muliggjør tilgang til Fishtalkdata. Dette APIet er en tjeneste som kan kjøpes fra AkvaGroup.

3.5.3. Hvordan er data tilgjengeliggjort for oss og hvordan ligger den lagret

Data fra Fishtalk er tilgjengeliggjort via det overnevnte APIet. AkvaGroup kan etter at API tjenesten er kjøpt opprette og gi tilgang til å gjøre spørringer. For å hente data gjøres det spørringer til flere ulike scope. Eksempel på scope er

- 'Feed' – Gir info om fôringsdata
- 'Batch' – Gir info om biomasse
- 'Samples' – Gir info om prøveuttak

Informasjonen fra spørringene oppsummeres i et JSON dokument.

Dokumentasjonen på APIet til Fishtalk ligger tilgjengelig på denne nettsiden:

<https://api.akvagroup.com/docs/index.html>

3.5.4. Softwarerammeverk for henting og lagring av data

For tilgang til Fishtalkdata har Salmon Evolution utviklet et Pythonbibliotek for å hente, transformere og sende data til Clarify. Spøringer fra dette biblioteket kjøres jevnlig og data fra Fishtalk tilgjengeliggjøres i nær sanntid til Clarify.

For å begrense data inn til Clarify gjøres det en sammenligning av dataen som skal lastes opp med dataen som allerede ligger lagret på Clarify. Hvis et datapunkt ligger lagret med samme verdi og samme tidssample i Clarify som det som skal lastes opp så droppes datapunktet som skal lastes opp.