

WHITEPAPER

Online monitorering og prædiktiv vedligeholdelse

Asger Nordlund, Senior specialist

IoT, Data & Services Innovation, FORCE Technology (asno@forcetechnology.com)

Mads Johansen, Specialist

IoT, Data & Services Innovation, FORCE Technology (majh@forcetechnology.com)



Indholdsfortegnelse

Introduktion	2
Motivering	2
Vedligeholdelsespolitikker	2
Korrigerende vedligeholdelse	3
Forebyggende vedligeholdelse	3
Internet of Things	4
Maskinlæring	5
Processen	6
Behovs- og systemanalyse	7
Behovsanalyse	7
FMECA	7
Systemanalyse	8
Interessentanalyse	8
Dataanalyse	8
Tilstandsindikatoranalyse	9
Sensor- og datatilgængelighedsanalyse	9
SCADA	10
Regler og forskrifter	10
Online monitorering	10
Design og udvikling	10
Implementering	11
Prædiktiv vedligeholdelse	11
Dataindsamling	11
Udvikling af ML-model	12
Implementering af ML-model	12
Konklusion	13

Introduktion

Motivering

For at få det optimale ud af sin anlægsinfrastruktur kræver det online overvågning/monitorering. Online overvågning er vigtig for at man på et oplyst grundlag kan opdage og løse driftsproblemer i tide eller identificere mulige kapacitetsbegrænsninger inden de får betydning for driften.

Historisk har indførelsen af online trådløs monitorering været omkostningstungt, ustabil og administrativt nærmest umuligt i stor skala. Det betyder, at beslutninger der har betydning for driften tages ud fra alt for få datapunkter og at det ofte har gjort det tidskrævende og dyrt at identificere flaskehalse og de bagvedliggende årsagssammenhænge.

I modsætning til tidligere, har IoT sensor teknologien og stadig bedre trådløse kommunikationsformer gjort det muligt at ændre fokus, således at indsatsen kan fokuseres på hvor dataindsamlingen kan gøre bedst gavn og hvordan data skal tolkes for at få det optimale ud af sin anlægsinfrastruktur.

Udover at få indblik i den øjeblikkelige drift er IoT teknologien anvendelig til at forudse hvornår infrastruktur er modent til udskiftning eller trænger til eftersyn. Det kan være motorer, pumper og andet maskineri, men også kloakledninger og alt andet infrastruktur.

Vedligeholdelse er en vigtig aktivitet i de fleste organisationer. Den kan have stor betydning for omkostninger og pålidelighed. Det er begge parametre, der vil påvirke en virksomheds driftseffektivitet og konkurrenceevne målt på parametre som pris, kvalitet og leverancekraft. Uplanlagt nedetid af udstyr kan potentielt også resultere i omdømme-tab og også bøder for en virksomhed hvis vedligeholdelsen ikke udføres rettidigt eller korrekt. I sidste ende kan det ende med at blive et kritisk sikkerhedsproblem.

Der er flere forskellige måder at planlægge og udføre vedligeholdelse på (en virksomheds vedligeholdelsespolitikker). Afhængigt af typen af udstyr, kan det f.eks. være en fordel at overvåge udstyrets tilstandsindikatorer for at hjælpe med at bestemme, hvornår vedligeholdelse er nødvendig. En måde at udnytte moderne teknikker/teknologier på er at bruge IoT sensorer til at overvåge tilstande. Brug af IoT teknologi gør det muligt at indføre automatisk og online overvågning, hvor sensorer kan bruges til at indsamle data om tilstanden af bestemt udstyr, som derefter kan lagres centralt og gøres tilgængelig online til gennemsyn uden at man er fysisk i nærheden af det overvågede udstyr. Dette er stærkt ønskeligt i f.eks. geografisk spredt spildevands infrastruktur.

Hvis tilstandsbaseret vedligeholdelse ikke er tilstrækkelig, kan man også bruge prædiktiv vedligeholdelse. Prædiktiv vedligeholdelse involverer typisk tilstandsovervågning, fejl diagnose, fejlprognose og vedligeholdelsesplaner, som hver især giver visse udfordringer hvis ikke man er opmærksom. Prædiktiv vedligeholdelse kan være nyttig i mange tilfælde, men nogle gange står investeringen ikke mål med udbyttet. Hvis dette er tilfældet, kan onlineovervågning være nok til at forbedre driften.

Denne hvidbog vil beskrive forskellige begreber relateret til online overvågning og prædiktiv vedligeholdelse, såsom vedligeholdelsespolitikker, IoT, Internet of Things og maskinlæring. Desuden vil den an vise en vej til, hvordan man udvikler og begynder at bruge online overvågning eller prædiktiv vedligeholdelse, dvs. hvilke trin der bør involveres og hvad man skal overveje undervejs i udviklingsprocessen.

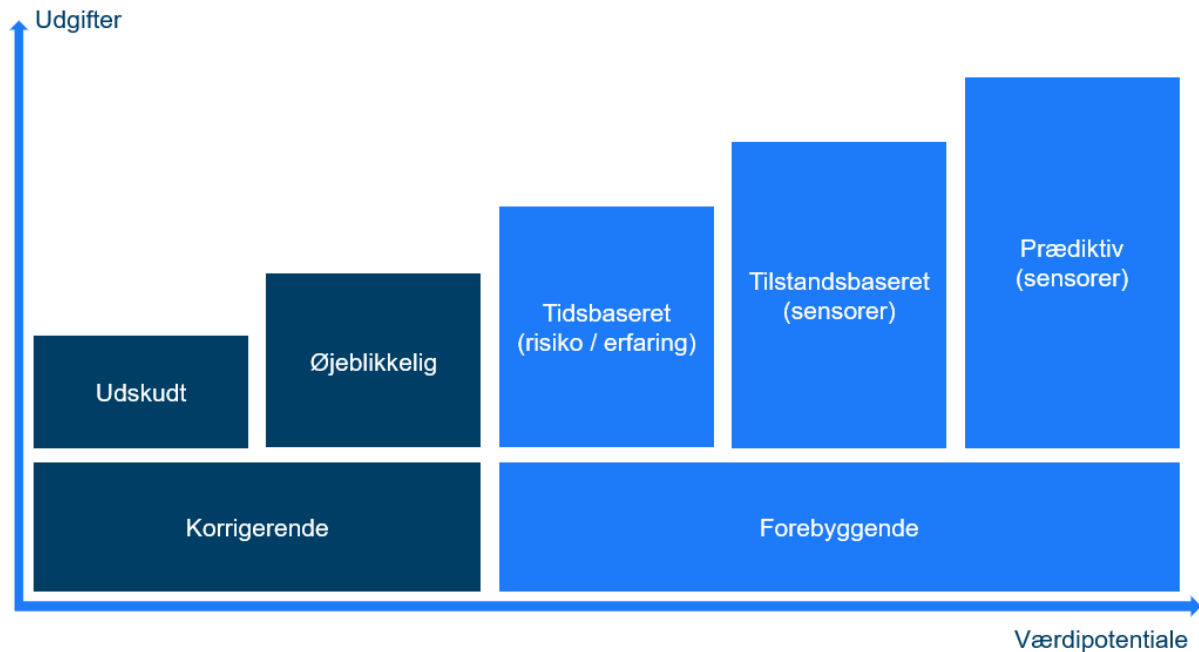
Vedligeholdelsespolitikker

Vedligeholdelse er en væsentlig udgift for enhver produktion og udgør mellem 15 % og 40 % af omkostningerne for de fleste produktioner - i nogle tilfælde endda op til 60 %.¹ I disse estimater er det ikke kun vedligeholdelse, der er dyr, men også de effekter og konsekvenser, som dårlig vedligeholdelse har, og derved vil fokus på effektiv vedlige-

¹ Al-Najjar, 1997 og Mobley et al., 2002

holdelse i de fleste tilfælde reducere omkostningsniveauet og potentielt forbedre indtægten. Udover at reducere omkostningsniveauet vil en optimal vedligeholdelsespolitik også være med til at sikre at ens produkter opnår en højere kvalitet.

De to overordnede indsats typer for vedligeholdelsespolitikker er "Korrigerende" og "Forebyggende". **Error! Reference source not found.** viser hovedtyperne af vedligeholdelsespolitikker med stigende kompleksitet.



Figur 1: Vedligeholdelsespolitikker, inddelt i kategorier og arrangeret efter kompleksitet. Ref: Lee et al., 2006

Korrigerende vedligeholdelse

Korrigerende vedligeholdelse er vedligeholdelse, der udføres efter en fejl er erkendt i systemet. Dette fører til nedetid, da driften stoppes mens problemet løses. Korrigerende vedligeholdelse kan opdeles i to underkategorier: "Udskudt" og "Øjeblikkelig".

Udskudt er, når der ikke er nogen handlingsplan for når en fejl opstår, dvs. problemet undersøges og rettes efter fejlen er opstået. På den anden side betyder øjeblikkelig, at en handlingsplan er på plads for forskellige fejl, før de opstår. Begge har styrker og svagheder og kan afhænge af typen af udstyr. I den øjeblikkelige politik er der mindre nedetid, fordi en handlingsplan er klar, men det kræver mere tid til planlægning for når forskellige typer fejl opstår.

Forebyggende vedligeholdelse

Forebyggende vedligeholdelse er, når vedligeholdelsen udføres på forhånd. Dvs. før nedbrud, på en planlagt måde. I modsætning til korrigerende vedligeholdelse, hvor fejl udbedres efter de er opstået, er målet med forebyggende vedligeholdelse altid at udføre vedligeholdelse før nedbruddet finder sted. Forebyggende vedligeholdelse kan opdeles i tre underkategorier: "Tidsbaseret", "Tilstandsbaseret" og "Prædiktiv".

Den tidsbaserede vedligeholdelse er afhængig af erfaring eller datablade på delene i systemerne. Det er service, eller vedligeholdelsesplanlægning, der typisk er baseret på producentens viden om cirka hvor længe noget er bygget til at holde før nedbrud.

Tilstandsbaseret, er hvor de tilstandskritiske dele vælges, og nogle parametre, der vil afsløre delens tilstand, overvåges. Når parametrene viser, at delene bør vedligeholdes, planlægges vedligeholdelsen, og der sættes dermed ind før fejlen opstår.

Prædiktiv vedligeholdelse er, når maskinlæring bruges til at genkende mønstre i systemets adfærd før en fejl. På denne måde vil det være muligt at forudsige mulige fejl, og planlægge vedligeholdelsen til lige før fejlen opstår.

Forebyggende vedligeholdelsespolitikker har, ligesom korrigerende vedligeholdelsespolitikker, fordele og ulemper.

Tidsbaseret vedligeholdelse er enkel og kræver meget lidt indsats at implementere, men det har den ulempe, at det ikke nødvendigvis vil reducere omkostningerne eller forhindre nedetider. Dele kan blive skiftet for ofte, mens de stadig er i god stand, hvilket medfører unødvendige reservedelsomkostninger. På den anden side kan en del blive slidt hurtigere end normalt, hvilket forårsager et uventet nedbrud, der resulterer i omkostningstung nedetid.

Det samme kan siges om tilstandsbaseret vedligeholdelse. En parameter, der overvåges, kan have en aflæsning, der aflæses som en indikation på fejl, men i virkeligheden er der bare tale om en læsefejl i systemet. Eller hvis aflæsning sker manuelt og/eller med jævne mellemrum kan dette betyde nedetid hvis noget går i stykker i mellemtiden. Dette antages dog mindre forekommende, og tilstandsbaserede politikker er en god løsning til simple systemer, hvor fejlindikatorparameteren er kendt og let kan overvåges. En tilstandsbaseret politik er lidt mere kompleks end den tidsbaserede politik, men vil gøre planlægningen af vedligeholdelse lettere.

Prædiktiv vedligeholdelse kan udelukke muligheden for, at dele udskiftes unødigt, og kan afsløre mønstre, der indikerer svigt eller reduktion i virke, som kan være ukendt og umulig at opdage. Det har en større kompleksitet og vil derved være mere omkostningstungt at implementere.

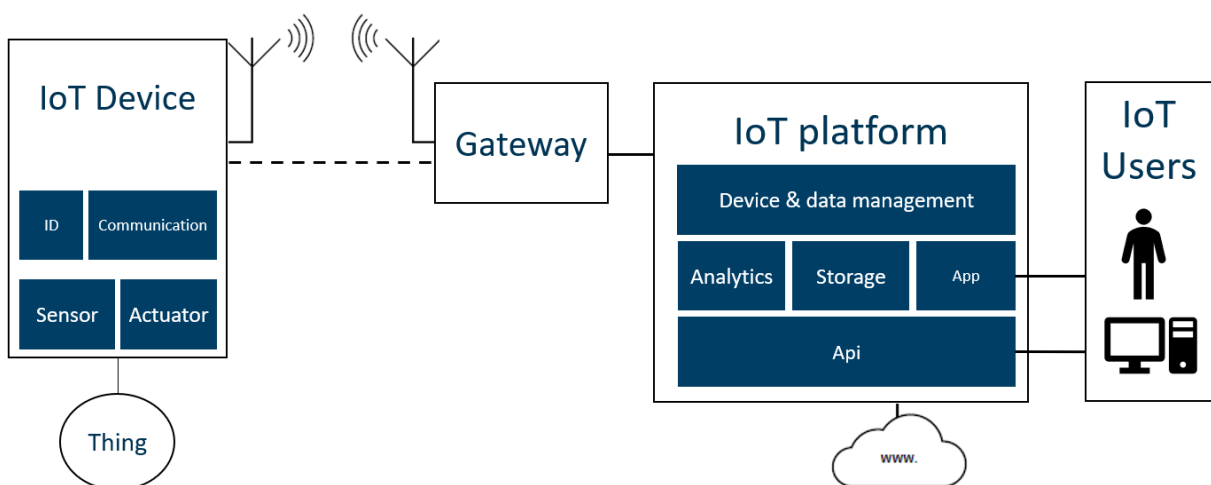
Den type vedligeholdelsespolitik, der passer bedst til et system, afhænger af mange ting. Hovedsageligt af at omkostningerne ved at implementere dem skal være billigere end de besparelser, der kan opnås. Hvis det ender med at blive dyrere eller tidskrævende at lave en forebyggende vedligeholdelse, så kan en korrigerende vedligeholdelsespolitik være mere fornuftig.

Internet of Things

Internet of Things (IoT) forbinder produkter og infrastruktur til internettet, hvor de kan aflevere data og information fra sensorer, som der efterfølgende kan handles på.

De fleste kender allerede IoT rigtig godt, da IoT-systemer er implementeret en hel del i det moderne hjem. F.eks. højttalere, der er tilsluttet nettet, samt køkkenmaskiner eller lys, der kan styres ved hjælp af en smartphone.

IoT er desuden ved at blive et stort aktiv for industrien, lige fra temperatursensorer til overvågning af kølekæder og vægtsensorer, der afgør, om en butikshylde er ved at løbe tør, til mere komplekse løsninger som prædiktiv vedligeholdelse.



Figur 2: Overblik over komponenter i et IoT system.

Et eksempel på en IoT-systemarkitektur ses i Figur 2. Et IoT-system vil bestå af en eller flere sensorer og potentielt en komponent, der skifter tilstand, f.eks. tænde/slukke, bevæge sig langs en akse, dreje en ventil osv., der er forbundet til en "ting". Enheden har et unikt digitalt ID og en form for kommunikationsteknologi til at overføre data til og fra enheden. Der kan muligvis være et netværk af enheder, der arbejder sammen. I nogle tilfælde sendes data via en "gateway" inden det ender i det centrale system hvor data som regel lagres og analyseres. Brugeren af et IoT-system kan styre eller interagere med enheder og bruge data fra enheder via IoT-plattformen. Denne platform har et lager (en database) til opbevaring af data og mulighed for at analysere og behandle data. Den kan desuden være forbundet til andre systemer og hente data via en såkaldt API, som bruges til kommunikation mellem to systemer.

Maskinlæring

Kort sagt er kunstig intelligens (Artificial Intelligence - AI) en teknologi, der gør det muligt for en maskine at simulere menneskelig adfærd ift. at tage beslutninger. Maskinlæring (Machine learning - ML) er en delmængde af kunstig intelligens, hvor en maskine automatisk lærer af at læse historiske data uden at være eksplicit programmeret.

I ML er det ikke nødvendigt at give detaljerede instruktioner, som computeren skal følge. I stedet forsynes en algoritme med data, som den behandler for at lære mønstre i disse data. Efterhånden som flere og flere data behandles af algoritmen, lærer den gradvist disse mønstre ved at opdatere sin interne forståelse af dataene. Outputtet af en sådan algoritme er en ML-model, som repræsenterer, hvad der blev lært af algoritmen. Afhængigt af algoritmen kan modellen bruges til, for eksempel, at forudsige fremtidige tilstande for en maskine.

Selvom ML-algoritmer automatisk kan lære mønstre i data, kan ML-algoritmer stadig være svære at udvikle, da der er mange aspekter der skal overvejes. Data er en stor del af ML, da algoritmerne lærer af de leverede træningsdata. Træningsdata er opsamlede data som man giver som input til ML-algoritmen og det output man får er en model der repræsenterer det som algoritmen lærte af de givne træningsdata. Derfor er forbehandling af data vigtig. Data skal have den rigtige struktur og være af god kvalitet, så maskinlæringsalgoritmer kan bruge dem.

Når en ML-model skal trænes, efter at have forberedt træningsdata, kan forskellige indlæringsretninger benyttes hvoraf tre kan fremhæves her:

Overvåget læring (supervised learning)

Uovervåget læring (unsupervised learning)

Halv-overvåget læring (semi-supervised learning)

Overvåget læring betyder, at data er mærket, og ML-algoritmen kan bruge dette ved læring. Dette kunne for eksempel være et datasæt med billeder af katte og hunde, hvor hvert billede er mærket med enten "kat" eller "hund". På den anden side har uovervåget læring ikke mærker på træningsdata.

Halv-overvåget læring er en blanding af de to, hvor kun en del af data er mærket. Her kan man bruge en algoritme på den mærkede del af dataene for efterfølgende mærke resten og derefter træne en ML-model på det fulde mærkede datasæt.

Inden for uovervåget læring findes begrebet anomali detektion (anomaly detection). Anomali detektion handler om at finde de datapunkter (ofte en lille procentdel), som er forskellige fra resten af dataene. Her er der to forskellige tilgange, hvor den ene bruger data forurenede af afvigelser, kaldet outlier detektion (outlier detection), og en anden bruger en halv-overvåget tilgang, hvor der kun trænes på data uden afvigelser. Så senere, når der testes med nye data, der indeholder afvigelser, bør afvigelse være anderledes end data fra træningssættet. Dette kaldes nyhedsdetektion (novelty detection).

Afhængigt af de tilgængelige data, er der forskellige måder at forudsige udstyrets aktuelle tilstand og udstyrets resterende levetid. Når der bruges uovervåget læring, kunne en tilgang være at kigge efter antallet af afvigelser, da der kan være nogle få afvigelser hist og her, når der indsamles data, men hvis de pludselig bliver hyppige, kan maskinen være ved at nå en fejl. Brug af uovervåget læring har sine fordele og ulemper. Det er nogle gange lettere at begynde at arbejde med uovervågede algoritmer, da dataene ikke skal mærkes, og det kan være en omfattende opgave at mærke data. På den anden side betyder arbejdet med umærkede data også, at der måske ikke er en klar definition af, hvornår en maskine er ved at nå en fejl, og der kan derfor være en vis fortolkning involveret. Hvis der er forskellige typer af fejl, og der arbejdes med umærkede data, betyder det desuden, at fejltypene ikke kan skelnes.

Brug af overvåget læring eliminerer flere af udfordringerne fra uovervåget læring, da data nu indeholder information om udstyrets tilstand eller hvilke typer fejl der er opstået. Med overvågede indlæringsalgoritmer og nok data fra forskellige fejl kan det være muligt at forudsige typen af en kommende fejl, og ikke blot at der er ved at opstå en fejl. Dette kan selvsagt hjælpe i planlægningen af vedligeholdelsen. En af de større udfordringer ved overvåget læring er, hvordan man får de mærkede data.

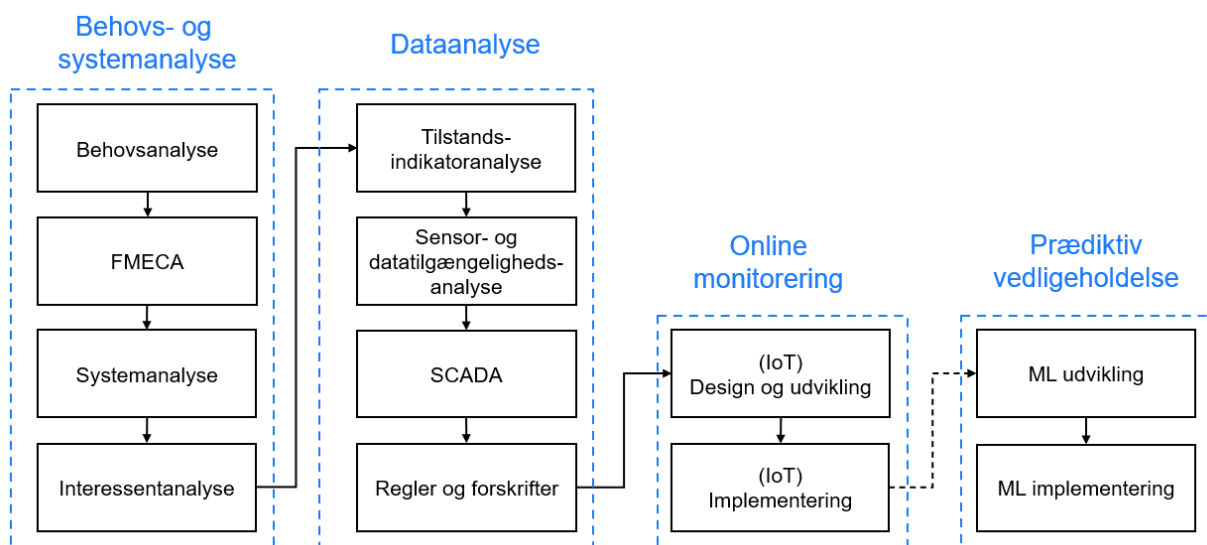
ML kan bruges i den prædiktive vedligeholdelsespolitik, for eksempel når der er store mængder data, mange forskellige input, eller når dataene har en kompleksitet, som er umulig for et menneske at analysere. Selvom ML kan være gavnligt, er det måske ikke altid nødvendigt. I tilfælde, hvor der er behov for prædiktiv vedligeholdelse, kan der være scenarier, hvor inputdataene er enkle nok til, at trends eller andre simple statistiske metoder kan bruges til at forudsige fremskridt i en maskines livscyklus. Fordi ML-udvikling kan være en meget omfattende opgave, er det derfor værd at overveje og forstå systemet og dataene, før det fastslås, at ML er vejen frem.

Processen

Denne proces vil guide brugeren gennem tre hovedtrin, som systematisk vil afklare, hvor/hvordan man fokuserer opmærksomheden, og implementere en online overvågningsløsning, og om nødvendigt et fjerde trin til at forbedre vedligeholdelsespolitikken til en prædiktiv vedligeholdelsesløsning.

Trinene er:

1. Behovs- og systemanalyse
2. Dataanalyse
3. Online monitorering
4. (Forudsigende vedligeholdelse)



Figur 3: Procesdiagram.

Hvert trin kan opdeles yderligere i underopgaver, som de følgende afsnit vil forklare. Det første trin, se Figur 3, består af en analyse af behov, eksisterende system(er), og eksisterende data. Først udføres behovsanalysen for at forstå problemuniverset og herefter systemanalysen for at forstå systemet både som en helhed og dets delelementer. Dataanalyse kan udføres efterfølgende, for at man bedre kan forstå sammenhænge i systemer og løsninger.

Onlinemonitoreringen laver man ved at kombinere den indsigt behovs-, data-, og system-analysen har tilført af viden. Indsigten gør én i stand til enten at designe et unikt IoT monitoreringssystem eller til at man kan indkøbe en færdig overvågnings løsning fra en leverandør.

Man kan overveje om trin 1-3 er tilstrækkelig og derved nøjes med onlinemonitorering, der kræver menneskelig overvågning og evt. indgreb ved fejl. Man kan også gå videre og indbygge procesreglerne i monitoreringssystemet, således at den menneskelige overvågning mindskes og regler lægges ind i IoT systemet, så drift ikke kræver lige så høje kompetencer.

Den prædiktive vedligeholdelse er for de systemer, hvor en online monitoreringsløsning ikke vil være tilstrækkelig, eller hvor systemet forventes at trække på fordelene ved at indføre en overvåget vedligeholdelsespolitik, hvad angår omkostningsoptimering, højere kvalitet i produktionen eller lignende, i overensstemmelse med afsnittet om vedligeholdelsespolitikker.

Processen er iterativ således at man fra hvert trin i processen kan gå tilbage til et tidligere trin og reevaluere eller tilpasse det efterhånden som man opbygger viden gennem processen.

Behovs- og systemanalyse

Behovsanalyse

Behovsanalysen er en forundersøgelse, der er tæt knyttet til afsnittet om vedligeholdelsespolitikker, da analysen vil afsløre, hvilken type politik der aktuelt anvendes, og afgøre, om en forebyggende vedligeholdelsespolitik er fordelagtig.

Resultatet skal være en analyse af den nuværende politik og omkostningerne ved nævnte politik samt et groft skøn over omkostningsreduktion, hvis en tilstandsbaseret eller prædiktiv vedligeholdelsespolitik skal implementeres. Hvis det viser sig at være værdifuldt at implementere en af disse, ville næste skridt være at udføre en FMECA.

FMECA

Failure Mode, Effects & Criticality Analysis (FMECA) er et risikoanalyseværktøj til at bruge, når man analyserer, hvilken form for vedligeholdelse, der skal bruges, da det er en systematisk tilgang til at finde, hvor man skal fokusere opmærksomheden. Når der udføres en FMECA, laves der en liste, der indeholder alle problemer, der forårsager fejl, eller fejl som kunne forbedres med hensyn til omkostningsreduktioner. Når alle problemer er blevet listet, bør de evalueres på tre parametre:

1. **Hyppighed**
Hvor ofte dette problem eller denne type fejl forventes eller opleves at opstå. (Meget hyppigt forekommende fejl scores højt.)
2. **Alvorlighed**
Hvor meget systemet påvirkes, inklusive konsekvenser, hvis fejlen opstår. (Meget alvorlige problemer scores højt.)
3. **Detekterbarhed**
Ved fejl, ville den nuværende vedligeholdelsesinspektion opdage fejlen eller årsagen til fejlen. (Scores højt hvis fejlen ikke ville opdages under inspektion.)

Hver parameter får et punkt på en valgt skala (kan være 1 til 10), og værdierne af parametrene multipliceres for at afsløre risikoprioritetsnummer (Risk Priority Number - RPN)². Problemstillingerne kan rangeres i faldende rækkefølge fra højeste RPN til laveste, og derved bruges til at vurdere de nødvendige handlinger.

$$RPN = \text{Hyppighed} * \text{Alvorlighed} * \text{Detekterbarhed}$$

² https://www.navsea.navy.mil/Portals/103/Documents/NSWC_Crane/SD-18/Test%20Methods/MILHDBK338B.pdf

Issue	Hypighed	Alvorlighed	Detekterbarhed	RPN
Overophedning	3	4	4	48
...

Tabel 1: Eksempel på FMECA

Når FMECA er blevet udført, evalueres et eller flere problemer med de højeste risikoprioritetsnumre, og det bestemmes, om en prædiktiv vedligeholdelsespolitik eller et online monitoreringssystem kan eliminere eller reducere virkningerne af problemerne.

Systemanalyse

Det er god praksis at lave en analyse af hele systemet, herunder også grove forklaringer af de dele af systemet, som måske ikke direkte påvirker prioritetsspørgsmålene fra FMECA. Dette vil give fælles grundlag og en fælles problemforståelse for det videre arbejde.

Figur 4 er et simpelt skema over hvordan en systemanalyse kan laves, ved at opdele systemet i delsystemer, og muligvis i enkelt komponenter.

Når man gør det, kan hvert delsystem beskrives i detaljer, men som nævnt ovenfor afhænger detaljeringsgraden af relevansen for den valgte problemstilling. Jo mere relevans delsystemet har for den valgte fejl, jo mere detaljeret skal det beskrives.

Hvis systemet er stort og komplekst, bør der laves et flowdiagram eller driftskoncept for at beskrive hele systemet.

Interessentanalyse

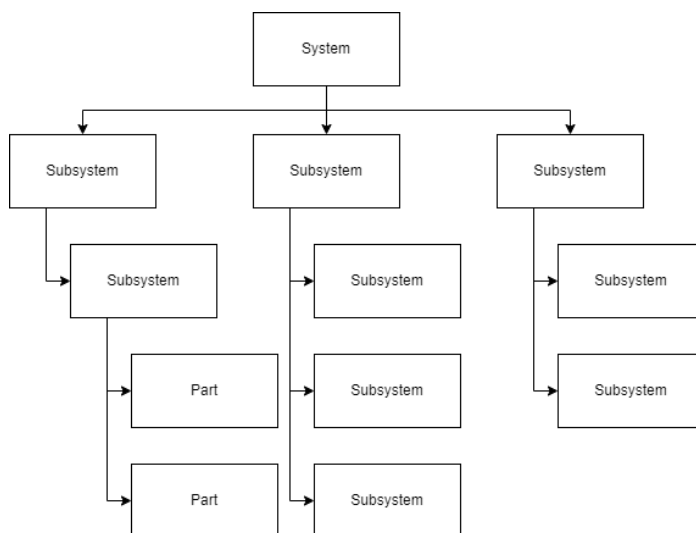
En interessentanalyse er tæt knyttet til systemanalysen, og eftersom en masse eksempler på metoder til interessentanalyser kan findes online, er det ikke nærmere beskrevet i denne hvidbog, hvordan man laver en sådan analyse.

De vigtigste interessenter, er normalt dem der ejer og driver systemet, såvel som enhver instans, enten offentlig, statslig eller privat/intern, som kan have nogen betydning med hensyn til lovgivning eller overholdelse af systemet.

Dataanalyse

Dette afsnit vil se nærmere på de data, der er nødvendige for at implementere en online monitoreringsløsning. Som resultat af behovsanalysen, er systemet blevet analyseret og problemstillingerne blevet prioriteret. Der kan derfor tages stilling til hvilke data er nødvendige og hvordan de opsamles, hvis ikke de allerede er tilgængelige.

Analysen er blevet opdelt i tre trin, der undersøger de fysiske fænomener systemet udfører og som kan kvantificeres for at afsløre mønstre, der fører til fejl, samt hvordan man måler disse fænomener.



Figur 4: Delsystem diagram.

Tilstandsindikatoranalyse

For de problemer og fejl, der er anført i FMECA, vil der være nogle tilstandsindikatorer, der viser sig gennem fysiske fænomener. Disse kan f.eks. enten være en temperaturstigning eller vibration, der indikerer, at noget et eller andet sted ikke fungerer korrekt. Identifikation af de forskellige tilstandsindikatorer for de forskellige fejl der kan opstå, er vigtig. Det er disse indikatorer der skal samles data omkring og som især vil være nødvendige i en prædiktiv vedligeholdelsesløsning. Efter at have gennemgået behovsanalysen, især FMECA, og tilstandsindikatoranalysen, herunder de ovennævnte overvejelser, bør det besluttes, hvad der skal overvåges, og der skal udarbejdes en kravliste til dataene.

Kravlisten bør være grundig nok til at inkludere must-haves til online overvågning af systemet, eventuelt nice-to-haves, og bør specificeres på et generisk sprog for at undgå misforståelser. Eksempler på krav, der skal tages i betragtning, kunne omfatte hvilke fænomener dataene skal repræsentere, format, tidsperspektivet (hvor ofte der skal samles data), evt. lagring osv. Dette kunne for eksempel være en IoT-enhed der måler på temperatur for indeklima. Her ville det være relevant at specificere hvor ofte der skal måles ift. hvor ofte og hvor hurtigt et eventuelt automatisk varme anlæg kan reagere. Altså hvor ofte anlægget skal modtage beskeder om at skrue op for varmen eller ned for varmen og hvad der kan betale sig ift. hvor hurtigt det varmer et rum op. Måske skal enheden lave flere målinger før den sender data for at mindske usikkerhed i målinger. Der kan desuden være krav til usikkerheden. Formatet på data og hvor præcist målingerne skal repræsenteres er også relevant. Man kan desuden overveje om der skal måles temperatur eller i stedet bevægelse i rummet, luftfugtighed, lys eller andet.

Sensor- og datatilgængelighedsanalyse

Efter at have fundet tilstandsindikatorerne, skal de tilgængelige sensorer og de data, der leveres gennem disse, beskrives. Det er selvfølgelig nødvendigt at finde sensorer der kan måle på tilstandsindikatorerne, men det skal også vurderes hvor præcise disse skal være og hvordan de passer ind i et budget, altså skal man vælge de dyre, men præcise sensorer, de billige eller noget imellem.

Det skal desuden vurderes hvilket data der allerede er tilgængeligt. Det kan for eksempel være man allerede har et SCADA-system (se næste afsnit) der samler relevant data og som man kan hente data fra. Her skal man dog også have regler og forskrifter in mente eftersom man ikke altid kan få adgang til data på tværs af systemer.

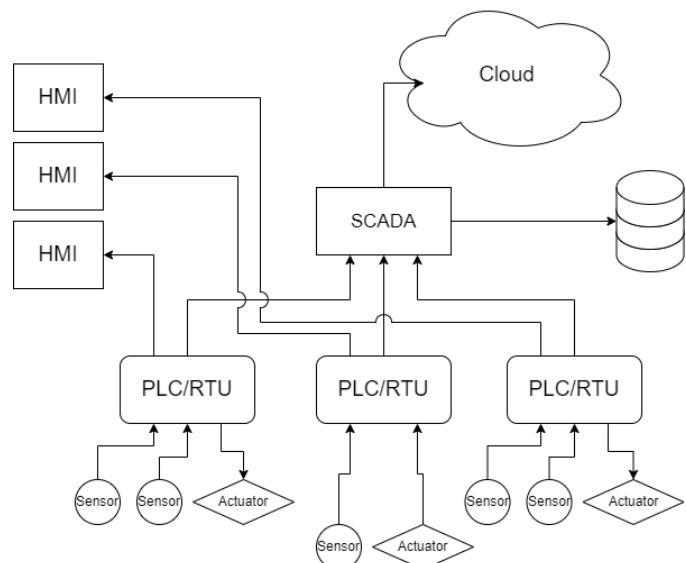
Når man overvejer dataene, bør de betragtes fra to vinkler; hvilke data leveres i øjeblikket fra sensorer eller andre kilder? Er de tilgængelige data tilstrækkelige med hensyn til de fænomener, der ønskes overvåget? Systemanalysen skal afsløre, om systemet omfatter sensorer, der vil levere data, og det bør overvejes, om de leverede data er realtid eller historiske.

Tilgængeligheden og prisen på sensorer, samt kompleksiteten af måleudstyret kan også være med til at bestemme om man skal fortsætte med de valgte tilstandsindikatorer eller vælge nogle andre som måske ikke ville være lige så gode, men som ville være meget billigere at udvikle et system omkring. Dette kan også betyde noget ift. hvilke fejl/problemer fra FMECA der fokuseres på.

SCADA

Et eksempel på, hvornår sensorer allerede er tilgængelige, er når et system bruger SCADA. SCADA (Supervisory Control And Data Acquisition) er et kontrolsystem, der bruges til at overvåge og kontrollere komplekse industrielle systemer. Den generelle opbygning af et SCADA-system kan ses i Figur 5.

Hvis et system bruger SCADA, vil det omfatte sensorer og aktuatorer forbundet til en PLC (Programmable Logic Controller) eller en RTU (Remote Terminal Unit). PLC'en/RTU'en vil modtage signaler fra sensorerne og levere styringer til aktuatorerne, og de vil blive overvåget lokalt gennem et HMI (Human Machine Interface). Hver PLC/RTU vil have et enkelt HMI, men alle produktionens PLC'er/RTU'er vil blive forbundet til et kontrolrum, som vil overvåge alle sensorer og være i stand til at styre alle aktuatorer for hele systemet på et, for systemet, globalt plan. SCADA-systemet vil hente alle data og uploade dem til skyen eller gemme dem lokalt i en database.



Figur 5: Eksempel på et SCADA-system.

Regler og forskrifter

Ligesom interessentanalysen vil beskrive nogle eksterne lovgivningsmæssige regler, kan der være andre overvejelser i forhold til compliance eller interne regler eller forskrifter, som vil påvirke datatilgængeligheden. Disse regler bør beskrives, da de vil være en del af designbeslutningerne.

Som for eksempel, hvis systemet er stort og komplekst, kan det bruge noget som SCADA. SCADA medfører at der er mange sensorer og dermed data tilgængelige. Man skal være opmærksom på at der kan være interne regler eller standarder, der ikke tillader, at dataene kan deles eksternt. Er der begrænsninger for datatilgængeligheden eller dataindsamlingsmuligheden, vil det begrænse mulighederne i onlineovervågningsløsningen.

Online monitorering

Behovs- og dataanalyseprocesserne vil give et fælles forståelsesgrundlag for systemet, som en online monitoreringsløsning skal implementeres på. Som det fremgår af afsnittet om vedligeholdelsespolitikker, betyder valget af den bedst egnede politik ikke, at prædiktiv vedligeholdelse vil være den bedste for alle systemer, og derfor bør det overvejes, om en online monitoreringsløsning vil være tilstrækkelig.

Online monitorering har en lighed med tilstandsbaseret overvågning i den forstand, at nogle tilstande i systemet bliver målt og overvåget, med den lille forskel, at det vil blive overvåget online, hvilket ikke nødvendigvis gælder for tilstandsbaseret vedligehold. Derfor bør det først fastlægges, hvilke data der er tilgængelige, og hvilke yderligere data der vil være nødvendige, dvs. hvilke fænomener der skal måles, som ikke allerede måles, hvilket sker gennem de delprocesser, der er anført i forrige afsnit. I dette afsnit fortsættes der med udvikling og implementering af online monitoreringsløsningen.

Design og udvikling

Efter at have fastlagt de data, der er nødvendige for online monitoreringsløsningen, og de data, der allerede er tilgængelige, kan man identificere hvilke data der mangler. Der skal designes en IoT-enhed til at opsamle dette data. At designe IoT-enheden er en kompleks og mangfoldig opgave, som er for kompleks til at beskrive i detaljer i denne hvidbog. Generelt afhænger designet af en IoT enhed af, hvilke fænomener den skal overvåge, hvilke sensortyper

der skal bruges, det miljø som enheden skal installeres i, hvor ofte der skal samles data, om enheden kører på batteri, opsamler energi selv eller er koblet til strøm permanent, om størrelsen og formen på enheden er vigtig, osv.

Afhængig af det ønskede design og kravene kan det være muligt for eksempel at udvikle sensorenheder, men det er også en mulighed at købe sig til en løsning. Der er flere standardsensorenheder og dataplatforme at overveje. Det er en god idé at have en overordnet idé om, hvordan systemet skal se ud og fungere, inden man køber løsninger. Derfor er de forudgående analyser stadig vigtige selvom man beslutter sig for at købe en løsning gennem en leverandør. Analyserne kan bl.a. hjælpe med at fastlægge de krav som skal kunne opfyldes af den rigtige løsning, hvis der er nogen, eller finde den løsning der inkluderer nok af den ønskede funktionalitet. Det er desuden en god idé at udvikle en proof-of-concept løsning til at starte med og teste på udvalgte maskiner inden løsningen skaleres.

Implementering

Når IoT-systemet er udviklet efter designet og kravene fra tidligere trin, skal det implementeres. Dette implementeringstrin er det første af to implementeringstrin, hvor det andet kommer senere, efter at have udviklet en maskinlæringsmodel. Det andet implementeringstrin består i at implementere ML-modellen i det eksisterende IoT-system, som implementeres i dette trin. Dette trin er for det meste fysisk, da det handler om at implementere det udviklede system og sætte det i produktion.

Sensorenheder, gateways, server/platform og brugergrænseflade skal installeres, konfigureres og gøres tilgængelige for dem, der vil bruge det i drift. Et vigtigt aspekt af implementeringstrinnet er at lave en sidste test af systemet for at sikre, at det fungerer som forventet. Når systemet kører, og testen er færdig, er det klar til brug.

Det er som nævnt også muligt at købe sig til en løsning. I så fald vil implementeringen højst sandsynligt ske i samarbejde med eller af leverandøren af systemet.

Prædiktiv vedligeholdelse

Den sidste del af processen handler om at tilføje prædiktiv vedligeholdelse til online monitoreringssystemet. I de foregående afsnit er et IoT-system blevet designet og udviklet til at overvåge udstyrs tilstandsindikatorer. Nogle gange kan det være en omfattende opgave eller umuligt at udføre overvågningen af maskinfejl manuelt. Det er her, ML kan spille en vigtig rolle, da maskiner i mange tilfælde kan lære mere komplekse mønstre i de indsamlede data sammenlignet med mennesker eller fremskynde opdagelsen af kommende fejl. Denne del af processen handler om at indsamle nok træningsdata, udvikle en ML-model og til sidst implementere den.

Dataindsamling

Dataindsamling handler mest om at sikre, at den implementerede online monitoreringsløsning fra forrige trin, kører og fungerer korrekt og at få det system til at samle data fra sensorerne og gemme det til senere brug. For eksempel til udvikling af en maskinlæringsmodel. Opgaven i dette trin er således begrænset til at holde øje med systemet og lade det samle data. Systemet kan naturligvis allerede nu bruges til at overvåge og manuelt kontrollere, om udstyret fungerer, som det skal, men den automatiske forudsigelse kommer senere, når en model er udviklet og implementeret.

Hvis opgaven ikke er at lave prædiktiv vedligeholdelse lige nu, men i stedet at blive ved online monitorering, så er systemet nu mere eller mindre klar til brug. Dette afhænger naturligvis af den ønskede funktionalitet vedrørende typer af visualiseringer og advarsler som systemet skal give og om disse allerede er udviklet i forrige del af processen. Hvis ikke kan systemet videreudvikles undervejs når man får mere erfaring med systemet.

Afhængigt af typen af ML-model, der skal implementeres, kan i det mindste variere i længde. Hvis for eksempel, hvis en fysisk model eller en regelbaseret model er bestemt til at være bedst egnet til et givent problem, i stedet for en ML-model, så kunne de udvikles, før der indsamles (flere) data. Dette skyldes, at udvikling af en regelbaseret eller fysisk model ikke nødvendigvis kræver historiske data i samme omfang som ML, men i stedet kræver viden om problemet og hvordan det kan modelleres. Denne udvikling kunne baseres på input fra fageksperter, for eksempel ved at definere visse tærskler, som sensormålingerne ikke bør overskride.

På den anden side kræver implementering af en ML-model en masse data til træning af modellen og generelt, jo mere data jo bedre. Det betyder dog også at jo mere data modellen skal bruge til træning, jo længere tid kan det tage at

indsamle, forberede og træne. Derfor bør mængden af data passe til kompleksiteten af problemet og modellens kompleksitet. Dette kan være meget svært at vurdere.

Udvikling af ML-model

Dette trin handler om at udvikle og træne en ML-model. Bemærk, at ML-modeller ikke altid er nødvendige for prædiktiv vedligeholdelse, for eksempel i tilfælde, hvor problemet er mindre komplekst, kan enkle regler og tendenser være nok til at forudsige, hvornår udstyr skal vedligeholdes.

Hvis den påkrævede model ikke er en ML-model, men i stedet regelbaseret eller en fysisk model, skal dette trin ikke nødvendigvis udføres efter indsamling af data. I stedet kan det gøres lidt parallelt med at udvikle IoT-systemet, da det primært bygger på fagekspertise. Udvikling af ML-modeller kræver også forståelse af problemet i forhold til at designe modellen eller valg af algoritme, men i sidste ende kræver det en masse data for modellen at blive trænet på.

Når man udvikler en ML-model, er der flere faktorer at overveje. Én ting er typen af model, som i nogle tilfælde kan afhænge af problemtypen og problemets kompleksitet. At have en god forståelse af disse to kan hjælpe med at vælge, hvilken type model der skal bruges, i den forstand at en eller anden type problem kan have en bestemt type ML-model, som passer meget godt. Desuden hvis man ved, at et givent problem er meget komplekst giver det måske ikke mening at bruge en meget simpel model, hvor forventningen fra begyndelsen er, at den sandsynligvis ikke vil fungere.

Sammen med problemstillingen og problemkompleksiteten kan de tilgængelige data også styre valget af model i bestemte retninger. Når man udvikler IoT-systemet til indsamling af data, skal det naturligvis overvejes, hvilke tilstandsindikatorer der er og få målinger fra dem, men alligevel kan der være tilfælde, hvor det ikke er muligt at hente data fra visse indikatorer.

I forhold til de tilgængelige data skal man også overveje kvaliteten af data og evt. manglende data. At sikre datakvaliteten af et IoT-system er et helt emne i sig selv, da der er mange faktorer, der kan påvirke de indsamlede data, såsom udfald af trådløs forbindelse, målenøjagtighed, kalibreringer, miljøfaktorer som placering af sensor og det miljø, hvori den er sat op, for at nævne et par stykker. Som nævnt i det foregående afsnit skal man også overveje mængden af data, mulig klasseubalance og hvor ofte målinger foretages.

Et andet vigtigt aspekt ved udvikling af en maskinlæringsmodel er dataforbehandling. I nogle tilfælde kan data ikke bruges direkte i maskinlæring og skal behandles, dvs. forberedes til brug. Der er flere forskellige måder at gøre dette på. Det kan være, at det er nødvendigt at udføre signalbehandling på dataene. Et eksempel kunne være, hvis et system indsamler vibrationsdata. Afhængigt af vibrationen kan hver indsamlet prøve bestå af flere tusinde værdier indsamlet over en kort periode. Afhængigt af algoritmen vil det sandsynligvis være umuligt at bruge de rå data som input til ML-modellen, når man har så stor en stikprøvestørrelse. I stedet kan man lave forbehandling i form af signalbehandling, hvor det rå signal analyseres, og parametre uddrages fra denne analyse. En anden potentiel forbehandlingsmetode er reduktion af dimensionaliteten i data. Dette handler om at reducere antallet af inputparametre til modellen, enten ved manuelt at vælge parametre blandt de indsamlede data eller ved at bruge en algoritme, som tager alle parametre, analyserer strukturen af data og transformerer dem til et reduceret antal parametre.

Implementering af ML-model

Denne anden implementeringsfase handler om at tilføje ML-modellen til det tidligere implementerede IoT-system. Indtil videre vil systemet primært blive brugt til at indsamle data om og overvåge udstyrets helbred, uden at forudsige, om det er ved at fejle, medmindre en sagkyndig manuelt inspicerer de indsamlede data. I det følgende vil forskellige måder at implementere ML på blive beskrevet og dette step handler kun om at implementere ML modellen. Det betyder at overvejelser om hvor den skal implementeres og hvordan bør gøres i design fasen.

Der er flere forskellige måder at implementere ML-modeller på, og implementeringen afhænger af typen af system og krav. En mulighed er at implementere modellen i skyen, hvilket selvfølgelig betyder at mange ressourcer er til rådighed, men betyder også, at forsinkelsen fra måling til forudsigelse er lang sammenlignet med at implementere den på "edge", hvilket er en anden mulighed. At have ML på edge betyder at modellen er implementeret på IoT-enheden hvilket resulterer i kortere forsinkelse. Specielt hvis enheden direkte kan påvirke miljøet uden at sende data til skyen. Enhederne har dog også meget mindre regnekraft end skyen og det begrænser derfor kompleksiteten af modeller der kan implementeres på edge. Desuden kan en ML-model implementeret på edge mindske data overførslen hvis kun resultatet sendes til skyen i stedet for det rå data.

Implementeringsmetoderne er ofte opdelt i offline-, batch-, realtids- og streambehandling. Offline behandling er, når modellen udvikles og bruges på for eksempel en lokal pc. Dette kan for eksempel bruges til engangsbegivenheder eller på ad hoc basis, når der er et datasæt tilgængeligt for en specifik hændelse, der kræver forudsigelser, og en ny hændelse vil kræve en ny model. Batchbehandling er, når forudsigelser foretages på regelmæssig basis og på en batch af data. Dette kan for eksempel være med den samme model, der bruges på forskellige batches over tid, for eksempel i en ugentlig cyklus, eller modellen kan opdateres, hver gang nye træningsdata bliver tilgængelige. Til realtidsbehandling trænes modellen i et separat miljø, men gøres tilgængelig online. Den kan derefter lave forudsigelser on-demand, når den bliver bedt om det, og på live-data med kun en lille forsinkelse. Igen kan modellen trænes én gang og derefter bruges, eller den kan opdateres, når nye træningsdata bliver tilgængelige. Endelig er streambehandling tæt forbundet med realtidsbehandling, da denne også er live, men den største forskel er, at den bruger en kø til forudsigelsesansøgningerne, fordi arbejdsbyrden er variabel.

Disse implementeringsmuligheder, bortset fra offline-behandlingen, kunne også være relevante for modeller, der er implementeret helt ude på IoT-enheden (edge), men da der er meget mindre beregningskraft, er edge sandsynligvis mere egnet til noget som realtidsbehandling.

Et andet aspekt at overveje er opdatering af modellen, når nye træningsdata er tilgængelige. En model kunne udvikles på et datasæt og derefter implementeres på en eller anden måde og slet ikke opdateres. Men når man kører et prædiktiv vedligeholdelsesprogram, vil data blive indsamlet hele tiden, så de nye data kan være brugbare til træning af modellen og kan derfor bruges til at opdatere den med flere data og nyere data.

Til sidst er det sidste trin at teste det implementerede system og model(ler). Test af systemet over en periode vil vise, om det er i stand til at opdage fejl i god tid. I takt med at systemet kører og testes, vil der opnås forskellige indsigter i forhold til systemets og modellens ydeevne, som kan bruges til at optimere og forbedre systemet over tid, for eksempel hvis det er nødvendigt senere at opdatere modellen eller udvikle en ny.

Konklusion

Online monitorering og prædiktiv vedligeholdelse er effektive koncepter der kan give en masse værdi for en virksomheds driftsudgifter. Det er vigtigt at have sig for øje at det ikke altid er nødvendigt i første omgang at investere i en prædiktiv vedligeholdelsesløsning. En sådan investering skal reelt kunne mindske vedligeholdelsesomkostninger eller forbedre produktionen og nok til at retfærdiggøre de relativt høje omkostninger ifm. udvikling og implementering af et sådant system. Det kan også være lidt af en udfordring at implementere, samt svært at få og fastholde et overblik over alle de mange aspekter der skal overvejes ved udviklingen af sådan en løsning.

Målet med denne hvidbog er at skitsere en overordnet proces for udviklingen af en online monitoreringsløsning eller hvis nødvendigt hele vejen til en prædiktiv vedligeholdelsesløsning. Startende med at analysere det eksisterende system for at finde kritisk udstyr der leder til nedetid samt hvordan udstyret fejler. Når fejlene, og hvordan de påvirker systemet, analyseres er det muligt at prioritere fokusområder for udviklingen sådan at en online monitorerings- eller prædiktiv vedligeholdelsesløsning giver mest muligt værdi. Efter systemanalysen er det nødvendigt at evaluere den data der allerede forefindes, hvis der er noget. Denne data kan ofte bruges i en IoT-løsning. I nogle tilfælde kan det også være muligt at bygge ovenpå et allerede eksisterende system i stedet for at udvikle en separat løsning. Som nævnt afhænger dette selvfølgelig af flere forskellige aspekter.

Når fundamentet er lagt i form af definerede krav og analyserede muligheder, kan udviklingen igangsættes og efterfølgende kan systemet implementeres og sættes i produktion. Ofte vil de første versioner have karakter af pilotprojekter og være et proof-of-concept for at sikre at systemet er brugbart inden en fuld skala løsning udvikles.

Efter en online monitoreringsløsning er udviklet og implementeret, kan denne fungere som fundamentet for en udvikelse hvor en ML-model implementeres for at gøre det til en prædiktiv vedligeholdelsesløsning. Dette kræver meget historisk data hvilket kan samles gennem online monitoreringsløsningen som en start og når nok data er samlet kan videreudviklingen udføres og fejl kan begynde at blive forudset før de opstår.