

Projektresultat  
Styrkeområde 5:  
**Cirkulära  
produktions-  
system och  
underhåll**

En bred bild över Artificiell Intelligens (AI) och de områden där AI-tekniker kan göra nytta, byggd på diskussioner med aktörer i **projektet Simon inom Produktion2030**

# Artificiell intelligens inom drift och underhåll

**Denna broschyr** *Artificiell intelligens inom drift och underhåll* är gratis och får användas fritt i oförvanskat skick, både i privat och kommersiellt syfte. Den får inte säljas eller på något sätt förvanskas.

## Projekt "Simon" inom Produktion2030s styrkeområde Cirkulära produktionssystem och underhåll

Syftet med denna guide är att ge en bred bild över Artificiell Intelligens (AI) och de områden där vi tror att AI-tekniker kan komma till nytta. AI-tekniker kan vara särskilt användbara i

tillämpningar inom drift och underhåll, liksom inom hållbarhetsområdet, för att öka svenska företags konkurrenskraft och effektivitet.

### Om Patrick Doherty, Olov Andersson och Yang Liu

Patrick Doherty är professor vid institutionen för datavetenskap (IDA), Linköpings universitet, Sverige. Han leder avdelningen Artificiell intelligens och integrerade datorsystem vid IDA. Han är även chefredaktör för tidskriften Artificial Intelligence, och sitter i styrelsen för KR Inc. Han är också en EURAI(ECCAI) Fellow. Hans huvudsakliga forskningsområden är kunskapsrepresentation, automatisk planering, intelligenta autonoma system och multi-agent system.

Olov Andersson är doktorand inom datavetenskap på Linköpings universitet. Han forskar på tillämpad artificiell intelligens

och maskininlärning, i synnerhet skalbara och robusta metoder för både beslutsstöd och autonomt beslutsfattande. Olov har samarbetat med både små och stora företag, och har ett näringslivsflutet inom forskning och utveckling av algoritmer för internetannonsering.

Yang Liu tog sin civilingenjörsexamen i Telekommunikationsteknik och TeknD i Industriellt Management vid University of Vaasa, Finland, år 2005 respektive 2010. Yang har en fast tjänst som biträdande professor vid Institutionen för ekonomisk och industriell utveckling (IEI) vid Linköpings universitet och är gästfakultet vid Department of Production vid University of Vaasa.

Han är ämnesföreträdande professor vid Jinan University, Kina och adjungerad/gästprofessor vid flera andra universitet. Yang har publicerat över 90 refereegranskade vetenskapliga artiklar. Yangs huvudsakliga forskningsinriktningar omfattar smart tillverkning, produkttjänstinnovation, beslutsstödssystem, hållbara konkurrensfördelar,

styrssystem, autonoma robotar, signalbehandling och mönsterigenkänning.

Mer information finns på  
<https://liu.se/medarbetare/patdo22>  
<https://liu.se/medarbetare/oloan10>  
<https://liu.se/medarbetare/yanli35>

Denna broschyr är en del av forskningsprojektet Simon – New Application of AI for Services in Maintenance towards a Circular Economy – som finansierats av Produktion2030 och företagen Attentec AB, Saab Group, Toyota MH.

# Artificiell intelligens inom drift och underhåll

En bred bild över Artificiell Intelligens (AI) och de områden där AI-tekniker kan göra nytta, byggd på diskussioner med aktörer i projektet Simon inom Produktion2030

## Innehåll

- Uppbyggnad och innehåll
- 1 Artificiell Intelligens – en översikt
  - 1.1 Problemlösning
  - 1.2 Kunskapsrepresentation och resonemang
  - 1.3 Automatisk planering
  - 1.4 Osäkerhet i kunskap och resonemang
  - 1.5 Beslutsteori
  - 1.6 Maskininlärning
- 2 Tillämpning av AI inom underhåll och hållbarhet
  - 2.1 Klassificering av AI-problem vid underhåll
  - 2.2 Modellering av driftavbrott
  - 2.3 Automatiserat beslutsfattande
- 3 Referenslista



# Uppbyggnad och innehåll

Denna översikt bygger på diskussioner med de industriella aktörerna i det Vinnovafinansierade FoU-projektet Simon inom Produktion2030, samt litteraturstudier. Guiden består av två delar där den första ger en generell översikt av AI som område, medan den andra specifikt tar upp dess tillämpningar inom drift och underhåll.



# 1

## Artificiell Intelligens – en översikt

Artificiell Intelligens (AI) omfattar en bred samling tvärvetenskapliga ämnen och har en rik historia som går tillbaka till slutet av 1950-talet. Det är ett ständigt växande område där både teori och verktyg har utvecklats för användning i många olika tillämpningar [30, 27].

Ett sätt att få en god förståelse för AI är att studera moderna definitioner samt hur det undervisas på universitetsnivå. Låt oss först börja med en definition som används av AAAI, American Association for Artificial Intelligence. AAAI grundades 1979 och är ett ideellt vetenskapligt sällskap som främjar framsteg för AI (<https://www.aaai.org/>). En av deras definitioner av AI är följande:

*”The scientific understanding of the mechanism’s underlying thought and intelligent behavior and their embodiment in machines”.*

En av pionjärerna inom AI och robotik, Hans Moravec från Carnegie Mellon University,

karaktäriserar fältets stora mål som ”en fritt rörlig maskin med en människas intellektuella förmåga” [78].

En modernare tolkning är att AI syftar till att konstruera ”intelligenta agenter”, vare sig det är robotsystem eller rent mjukvarubaserade system. Ett primärt mål är att studera och förstå samspelet mellan resonemang (Reasoning, tankeprocess bakom slutledning) och beteende eller handling. Denna mer neutrala, icke-robotiska karaktärisering möjliggör användandet av olika verktyg inom AI i många olika tillämpningsområden, såsom hållbarhet och underhåll. Därför skulle en mer lämplig beskrivning kunna vara ”ett system med funktionaliteter som skulle anses vara intelligenta av en extern observatör”. På senare tid har termen intelligensförstärkning (efter engelskans Intelligence Augmentation eller Intelligence Amplification) använts för att karakterisera system som integrerar AI-teknik för att öka effektivitet och värde för användaren.

Eftersom AI är ett tvärvetenskapligt område som omfattar många vetenskapliga discipliner och integration av olika teorier och verktyg, kan det för att förstå området vara bra att ta en titt på en av de mest populära läroböckerna inom området. Boken är skriven av Stuart Russell och Peter Norvig, den sistnämnde forskningschef på Google, och heter ”Artificial Intelligence: A Modern Approach” [32]. Den används som lärobok vid över 200 universitet och även som referensskälla av ingenjörer och forskare inom området. Boken bygger på idén om att bygga intelligenta agenter ur ett evolutionsprocessperspektiv. Det börjar med enkla reaktiva agenter eller modeller med liten intern kunskap men som kan uppfatta och agera. Den går sedan vidare mot mer sofistikerade målbaserade agenter som omfattar modeller som införlivar eller internaliserar olika aspekter av sin miljö och använder dessa modeller för att resonera och fatta beslut. Beslutsprocessen kan använda klassiska tekniker baserade på användning av logik eller tekniker från beslutsteori som bygger på sannolikhetslära. De mest sofistikerade agenterna innehåller även inlärningsförmågor för att dynamiskt förbättra utförandet genom att ta tillvara på erfarenheter.

Vid utvecklingen av sådana intelligenta agenter tar boken upp flera verktyg och tekniker som även kan användas fristående inom tillämpningsområden som drift och underhåll av maskiner. Boken omfattar fem centrala delar:

1. Problemlösning (Problem Solving)
2. Kunskap, resonemang och planering (Knowledge, Reasoning and Planning)
3. Osäkerhet i kunskap och resonemang (Uncertain Knowledge and Reasoning)
4. Inlärning (Learning)
5. Kommunikation, uppfattande och agerande (Communication, Perceiving and Acting)

Kommande avsnitt ger en översikt över de fyra första delarna.

### 1.1 Problemlösning

Problemlösning är som nämnts en central del inom AI, där sökningar används för att lösa ett problem [28]. Målet med sökning är oftast att hitta en lösning som uppfyller i förväg ställda villkor, men det kan även vara att hitta den bästa lösningen enligt någon given rangordning. Sökning har således viss överlappning med klassisk matematisk optimering. Sökning inom AI kan delas in i oinformerad sökning, informerad sökning och lokalsökning. Sökning i den här bemärkelsen kan syfta på problem med struktur, ofta sekventiella beslut, som kan modelleras i ett ”sökträd” bestående av noder som representerar problemets tillstånd, sammanknutna av grenar som representerar alla beslutsalternativ.

#### 1.1.1 Oinformerad sökning

Oinformerad sökning är domänoberoende, det vill säga utan någon semantisk förståelse för problemområdet mer än i domänens ursprungliga problemformulering. Sökningen täcker tekniker såsom djupet-först- och bredden-först-sökning, samt iterativ fördjupning av djupet-först-sökning. Djupet-först-sökning med återspårning (Backtracking) är en av de sökmotorer som används i många algoritmer och tillämpningar, exempelvis används den som grund för algoritmer relaterade till slutledning (inferens), automatisk planering samt bivillkorslösare (Constraint Solvers).

#### 1.1.2 Informerad sökning

Informerad sökning är en form av intelligent sökning som nyttjar existerande information om en problemdomän. Detta översatts matematiskt till en ”heuristikfunktion” som ger en fingervisning



om ett alternativs (nods) lämplighet under sökningen [12]. Heuristiker är en teknik utformad för att snabbare lösa ett problem när klassiska metoder är för långsamma, eller för att hitta en tillräckligt bra lösning när klassiska metoder inte lyckas hitta någon exakt lösning. Idén här är att sökandet efter lösningar i ett sökträd blir effektivare och mer riktad mot målet med hjälp av en heuristik.

En av de viktigaste algoritmerna som upptäckts är A\*-sökning [18]. Givet att en heuristikfunktion uppfyller kriteriet att inte överskatta den verkliga kostnaden, så kan man visa att algoritmen är optimal och optimalt effektiv för specifika problemområden. Idag har A\*-sökning en utbredd användning i många tillämpningsområden, i synnerhet inom robotik.

Ett bra industriellt exempel på detta är den robotbaserade logistiklösningen för lager som utvecklats av KIVA Systems. KIVA köptes nyligen av Amazon och lösningen kallas numera för Amazon Robotics. Amazon använder för närvarande cirka 100 000 robotar; för ett medelstort lager uppskattas antalet till runt 50-100 mindre robotar. Dessa robotar transporterar varor till mänskliga medarbetare och kan nyttja A\*-sökning algoritmer upp till 100 000 gånger om dagen för att hitta giltiga och kollisionfria vägar i lagret för leveransen.

### 1.1.3 Lokal sökning

Lokal sökning [19] är ett alternativ till den stegvisa generering av sökträd som används i de oinformerade och informerade metoderna för sökning. Dessa brukar benämnas "giriga algoritmer" eftersom de för varje beslut bara tittar på sina närmaste nodgrannar och väljer de tillåtna noder som förefaller närmast en problemlösning. Ett exempel är "Hill Climbing"-sökning. Dessa algoritmer är snabba och använder

lite minne. Å andra sidan kan dessa algoritmer lätt fastna i icke-optimala lösningar och icke-lösningar, så kallade lokala minima eller maxima. Det finns även varianter som försöker motverka problem med lokala minima, såsom simulerad härdning (Simulated Annealing), strålsökning (Beam Search) samt genetiska algoritmer. Dessa tekniker kan vara framgångsrika inom vissa specifika domäner.

### 1.1.4 Adversarial sökning

Ett annat intressant tillämpningsområde för sökning är utvecklingen av algoritmer för tvåmanna- och n-mannaspel såsom backgammon, schack [8] och Go [34]. Det intressanta med dessa spel är relationen mellan spelaren och dess motståndare och de potentiellt stora sökrymder som måste sökas igenom för att hitta effektiva drag. Till exempel har schack i genomsnitt över 35 möjliga handlingar för varje position, 100 möjliga drag (50 per spelare) vilket gör att det finns  $10^{150}$  noder i ett sökträd. Go är ännu värre med i genomsnitt 200 möjliga handlingar med cirka 300 drag, vilket leder till  $10^{700}$  noder i ett sökträd. Några av de tekniker som används här är min-max-sökning, alfa-beta-beskränning och Monte Carlo-trädsökning [6]. På senare tid har dessa tekniker kombinerats med maskininlärningstekniker vilket har resulterat i världsklassspelare i både schack och Go.

## 1.2 Kunskapsrepresentation och resonemang

Kunskapsrepresentation och resonemang är ett av de stora forskningsområdena inom AI [38, 5, 11]. Grundidén är att ju mer kunskap en agent har om sig själv och sin miljö, desto mer kompetent kommer agenten att vara i sina beslutsprocesser. Målet är att hitta olika sätt att representera kunskap som sedan uttrycks som datastrukturer i ett intelligent system. Kunskapen kan därefter

resoneras kring med hjälp olika resonemangs- och inferenstekniker. Resultatet är ett intelligent system som kan göra förutsägelser om framtida skeenden och fatta beslut i nuet för att optimera utförandet i de miljöer den befinner sig i.

En av de grundläggande metoderna för representation är användandet av logik och inferensteknik kopplad till logik, där ordet logik bör tolkas i en vidare bemärkelse. Det inkluderar klassisk logik och bevisteori men sträcker sig långt utöver detta. I synnerhet har kombinationen av temporallogik och modellprövning för korrekthetsverifiering haft en bred användning inom industrin för modellering och felsökning av komponenter i storskaliga komplexa system, såsom bilar.

I anpassningen av World Wide Web till den semantiska webben, en viktig komponent inom Internet-of-Things (IoT), används beskrivningslogik [3]. Beskrivningslogiker är välstrukturerade fragment av första ordningens logik som innehåller specialiserade inferensalgoritmer som används för att resonera kring begreppshierarkier. Dessa börjar användas av många företag för att specificera och representera interna kunskaper om företaget i sig och de produkter de erbjuder.

Internet-of-Things kan tolkas som ett stort nätverk av agenter som kommunicerar med varandra. Varje "sak" kan ha en tillhörande agent som gör saken mer intelligent i den meningen att den kan utföra olika former av resonemang om sig själv och andra agenter som den interagerar med, inklusive människor. Dessa multiagentsystem relaterade till IoT är även av stort intresse för forskningen inom AI [33, 40]. Det handlar om modellering av system med flera agenter där verktygen för logik, spelteori, sannolighetsteori och beslutsteori används för att specificera och förstå multiagentsystem.

Vi har nu gått igenom grundläggande tekniker och verktyg som har utvecklats för kunskapsrepresentation och resonemang. Många av dess tekniker har förenats till kunskapsbaserade system (tidigare kallade expertsystem) [36]. Dessa system består av en domänberoende inferensmotor optimerad för resonemang med regler och, för specifika domäner, införlivade domänberoende regler för resonemang för just dessa domäner. Systemen används i många olika tillämpningar inom många olika branscher, från finans och logistik till medicinska diagnoser. Dessa branscher är än idag viktiga områden för tillämpad AI.

## 1.3 Automatisk planering

Automatisk planering [14, 15, 23] syftar till att skapa teori och verktyg för generell problemlösning, och är ett aktivt forskningsområde inom AI. En enklare reaktiv agent kan i viss utsträckning agera i den verkliga världen genom att helt enkelt reagera på direkta intryck och sensoriska indata. Dock kan de, förutom i de allra enklaste scenarierna, inte ta hänsyn till de långsiktiga konsekvenserna av varje handling. Vissa handlingar kan då väljas bort på grund av kortsiktigt negativa konsekvenser trots att de är positiva på lång sikt. Andra handlingar som faktiskt väljs och utförs kan vara skadliga eller till och med förhindra att agentens mål uppnås.

De långsiktiga konsekvenserna av en handling är inte omedelbart uppenbara och kan därmed inte analyseras isolerat. Om en handling exekveras kan det hindra andra handlingar från att utföras eller ändra på andra handlingars resultat. Detta kan i sin tur ha spridningseffekter som bara kan förstås till fullo genom att överväga alla potentiella följder av en viss handling. På samma sätt kan exekvering av en handling möjliggöra andra handlingar eller ändra resultat på ett



positivt sätt; om denna kombination exekveras kan det istället leda till ytterligare möjligheter.

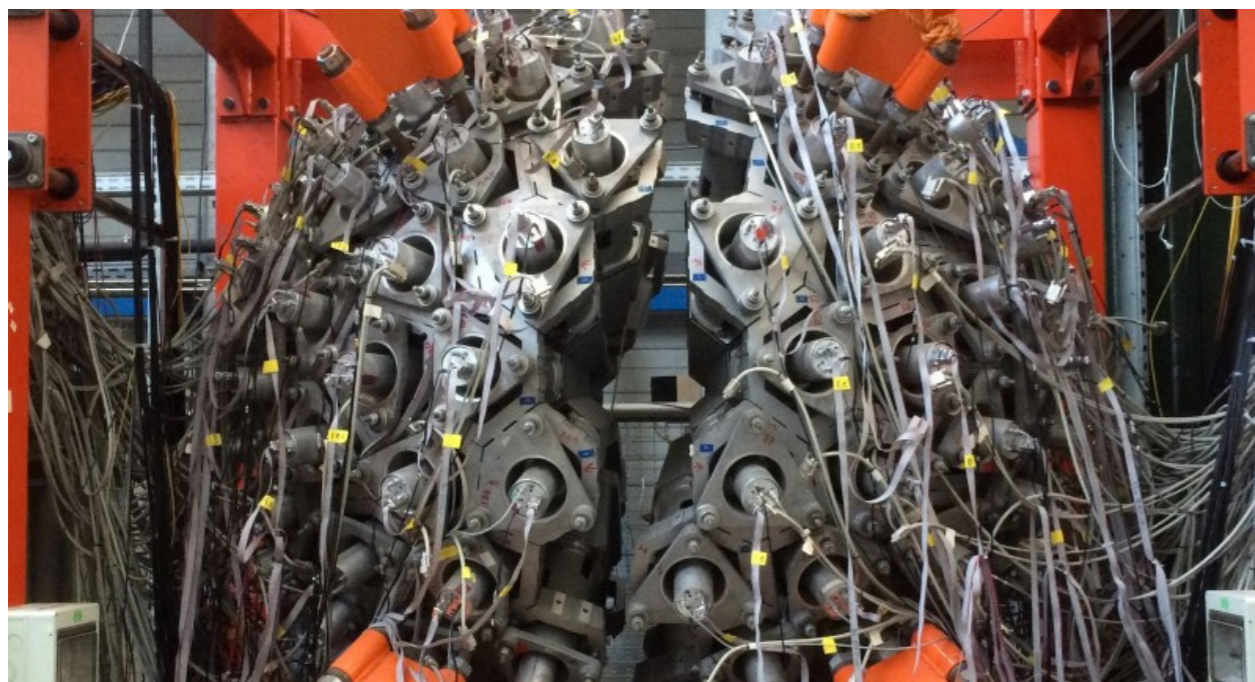
För att på ett korrekt sätt ta hänsyn till de fullständiga långsiktiga konsekvenserna av olika handlingar så krävs det ett resonemang kring planen för de handlingar som är avsedda för att uppnå ett specifikt mål under särskilda omständigheter. När målen är absoluta – exempelvis att en robottruck måste placera alla pallar på rätt sätt, inte bara i de flesta fall, och måste hitta ett sätt att göra det på inom en satt tidsfrist – kan det vara nödvändigt att i förväg använda resonemang om planer för att undvika återvändsgränder (exekvering som inte kan slutföras) eller undvika att bryta mot restriktioner. Även i de fall där ett reaktivt tillvägagångssätt skulle ha uppnått målen kan resonemang kring planer och ”tänka långsiktigt” leda till planer med högre kvalitet eller lägre kostnad.

Klassisk planering bygger ofta på tidigare nämnda algoritmer för trädsökning, men

implementationsmässiga detaljer kan abstraheras bort genom att först beskriva problemet med ett domänberoende språk, så kallad domänberoende planering. På så sätt ämnar automatisk planering tillhandahålla generella verktyg för att lösa planeringsproblem. Planering under osäkerhet är ett annat stort område, och i avsnitt 1.6.3 kommer vi att avhandla sådan planering i kontexten av självlärande system.

### 1.4 Osäkerhet i kunskap och resonemang

Miljöer där intelligenta agenter tillämpas kan vara mycket komplexa och ofta kan agenten bara observera en del av sin omgivning åt gången. Dessa observationer kan i sin tur innehålla mätfel och således vara osäkra. På grund av detta är våra modeller av omgivningens olika aspekter, och den interaktion som agenten kan ha med den, ofullständiga. Här kan icke-monotoniska resonemangstekniker erbjuda ett sätt att hantera vissa typer av ofullständighet.



Ett annat spår i AI-forskning är att använda probabilistisk inferens som ett sätt att adressera den osäkerhet som en agent har kring delar av sin miljö. Användningen av probabilistisk resonemangsteknik har använts länge inom AI och sträcker sig tillbaka till 60-talet. Två tidiga intresseområden var tillämpningen av probabilistiskt resonemang för medicinsk diagnos och för mönsterigenkänning.

#### 1.4.1 Bayesianskt nätverk

I slutet av 70- och 80-talet användes DAG (Directed Acyclic Graph) för att åskådliggöra osäkra variabler och deras interaktioner, vilket blev populärt i form av beslutsnätverk och inflytelsediagram (Influence Diagram). En banbrytande bok skriven av Pearl [29] introducerade användningen av probabilistisk resonemangsteknik till AI-samfundet på ett jordnära och pragmatiskt sätt genom probabilistisk inferens baserad på bayesianska nätverk. Strukturen i ett bayesianskt nätverk kodar implicit sannolikhetsfördelningen över alla osäkra variabler (så kallade slumpvariabler) i ett problem genom att nyttja oberoendeförhållanden bland dem på ett effektivt sätt. Bayesiansk probabilistisk resonemangsteknik blev således en viktig grund för inferens i olika typer av intelligenta system tack vare det arbete som utfördes av Pearl och hans kollegor, vilket beskrivs mer detaljerat i boken.

Grundidén bakom detta genombrott var att utnyttja beroenden/oberoenden bland slumpvariabler i modelleringsprocessen och att åskådliggöra dessa genom att använda en DAG. Följande ikoniska problem beskrivs i Pearls bok:

- En person installerar ett nytt inbrottslarm hemma. Det reagerar på inbrott men kan ibland även reagera på jordbävningar.
- Personen har två grannar, John och Mary, som

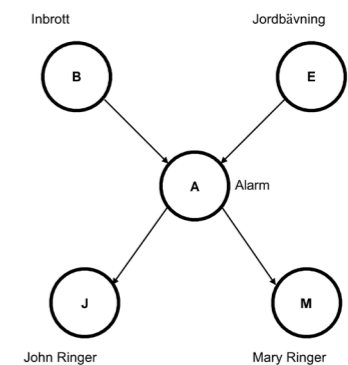
- lovar att ringa till dennes jobb när larmet går.
  - o John ringer alltid när han hör larmet men ibland blandar han ihop telefonsignalen med larmljudet.
  - o Mary, som gillar högljudd musik, missar larmet helt och hållet.

Följande DAG visar problemet där slumpvariabler som direkt påverkar andra variabler pekar på dessa. En ”barnnod” är beroende av sin modernod. Varje nod har en sannolikhetstabell kopplad till sig som beskriver det probabilistisk beroendet mellan respektive nod och dess modernoder, inom sannolikhetslära kallad en ”betingad” sannolikhet.

B	E	P(A)	A	P(J)	A	P(M)
T	T	0.95	T	0.90	T	0.70
T	F	0.94	F	0.05	F	0.01
F	T	0.29				
F	F	0.001				

Med hjälp av kedjeregeln och de oberoende-antaganden som strukturen i grafen medför, så kan sannolikhetsfördelningen över alla variabler karakteriseras som en produkt av enklare betingade fördelningar.

$$P(B, E, A, J, M) = P(B)P(E)P(A | B, E)P(J | A)P(M | A)$$



Figur 1: Exempel på bayesiskt nätverk.

Där:

$$P(x^1, \dots, x^n) = \prod_{i=1}^n P(x^i | parents(X_i))$$

Anta att vi skulle vilja beräkna fördelningen för B betingat på att vi har observerat att j, m är sant:

$$P(B | j, m)$$

Per definition:

$$P(B | j, m) = \frac{P(B, j, m)}{P(j, m)} = \alpha P(B, j, m)$$

För att beräkna detta måste vi summera över utfallen av de osäkra variablerna:

$$\alpha \sum_e \sum_a P(B, e, a, j, m) = \alpha \sum_e \sum_a P(B) P(e) P(a | B, e) P(j | a) P(m | a)$$

P(B) ligger inte inom summeringsområdet så vi kan flytta ut det:

$$\alpha P(B) \sum_e \sum_a P(e) P(a | B, e) P(j | a) P(m | a)$$

P(e) ligger inte inom summeringsområdet för a, så vi kan flytta ut det också:

$$\alpha P(B) \sum_e P(e) \sum_a P(a | B, e) P(j | a) P(m | a)$$

Denna ekvation representerar kortfattat ett "och/eller"-träd, där "och" är produkten och "eller" är summeringen. Därefter går det att slutföra den probabilistiska inferensen genom att läsa av de betingade sannolikhetsvärdena i tabellerna kopplade till nätverket. En fördel med probabilistiska metoder är att det ger ett konsekvent ramverk för att beräkna sannolikhetsfördelningen över osäkra variabler av intresse baserat på de variabler vi redan observerat. Denna teknik har tyvärr beräkningsmässiga begränsningar vid stora problem, och i allmänhet används ofta approximativa tekniker snarare än exakt inferens.

#### 1.4.2 Kommentarer

Aktuell forskning inom bayesianska nätverk och relaterade områden går under den bredare beteckningen "grafisk modellering" och är ett

viktigt område inom AI. Några av de senaste böckerna inom området, skrivna av Darwche [10] och Koeller [22], ger en omfattande och aktuell översikt av forskningsområdet. Ett exempel på detta är utvecklingen och användningen av Hidden Markov Models (HMMs) som har fått en stor betydelse inom taligenkänning och bildbehandling. Generaliseringar till probabilistiskt resonemang över tid har också lett till genombrott inom robotik, med ett underområde som kallas probabilistisk robotik. Här har den banbrytande boken av Thrun et al [37] lett utvecklingen framåt.

På senare tid har det förekommit en hel del aktiviteter inom AI kring utvecklingen av relationella och första ordningens sannolikhetsmodeller. I stället för att vara begränsad till ovanstående representationer är ett önskemål att öka representationsspråket som används i probabilistisk modellering till brottstycken av första ordningens logik. Nya tekniker under utveckling kombinerar sannolikhets teori med flexibiliteten i första ordningens logik. Resultatet kommer att ge en dramatisk ökning av antalet problem som kan hanteras med hjälp av probabilistiska metoder. Detta är ett hett forskningsområde med både djupa och kvarstående problem som behöver lösas både teoretiskt och praktiskt.

### 1.5 Beslutsteori

Studien av beslutsteoretiska agenter är en viktig del av AI. I det enklaste fallet får en agent en uppsättning potentiella handlingar som den kan utföra och väljer bland dem baserat på hur önskvärt deras omedelbara utfall är. För att kunna göra detta måste en agent ha någon form av uppfattning om vad som anses vara ett bra eller dåligt utfall. Låt oss anta att utfallen är tillstånd. Agentens inställning till att vara i ett tillstånd definieras av en nyttofunktion U(s) som tilldelar ett nummer som mäter hur önskvärt ett tillstånd s är.

I detta tillvägagångssätt är handlingar stokastiska i den meningen att exekvering av en handling kan resultera i ett av ett flertal olika tillstånd med tillhörande sannolikhet. För en specifik handling bör såldes dessa sannolikheter summera till 1. Följaktligen antas en sannolikhetsfördelning för varje handling över sina utfallstillstånd där, givet eventuellt ytterligare observationer i form av evidens e, sannolikheten att få utfall s<sub>i</sub> när man utför en handling a är:

$$P(\text{result}(a) = s' | a, e)$$

Om vi utgår från dessa sannolikhetsfördelningar och nyttofunktionen U(s) så är den förväntade nyttan av handlingen, givet bevisningen e (betecknad EU(a | e)), följande:

$$EU(a | e) = \sum_{s'} P(\text{result}(a) = s' | a, e) U(s')$$

Detta är den genomsnittliga nyttan av resultaten av en handling viktad mot sannolikheten för att det utfallet inträffar givet handlingen a.

En beslutsteoretisk agent, med ett urval av handlingar i ett visst sammanhang e, skulle då välja den handlingen som maximerar det förväntade nyttovärdet (MEU):

$$\text{best action} = \text{argmax}_a EU(a | e)$$

Observera att i detta enkla fall tar agenten ingen hänsyn till framtida situationer mer än vad som implicit uttrycks i de värden som ges av utfallstillstånden via nyttofunktionen.

I boken "Artificial Intelligence: A Modern Approach" [32] menar författarna att i huvudsak kan MEU-principen ses som att den definierar alla delar av AI. Med hänsyn till vad en agent uppfattar utöver dess förkunskap, e, så beräknar den olika kvantiteter, maximerar nyttan av utförbara handlingar i ett tillstånd, utför handlingen och upprepar processen

från det nya tillståndet. Om nyttofunktionen korrekt kan kvantifiera en prestandamätning för en agent i den omgivning som agenten är inbäddad så skulle denna strategi också maximera dess utförande. Då gör agenten alltid det rätta! Det motsatta gäller om nyttofunktionen inte är korrekt, då kan det innebära att agenten ofta gör fel.

Längre fram i denna rapport kommer nyttomaximering över en sekvens av framtida beslut att beaktas under rubriken reinforcement learning. Dessa metoder låter en agent lära sig både en nyttofunktion, och en strategi som maximerar den, baserad på dess erfarenhet av exekvering av handlingar i sin miljö och observation av resultaten. Observationer är i det här fallet både evidens i form av utfallstillstånd efter en handling, men också en belöning som relaterar huruvida utfallstillståndet är bra eller dåligt. Därefter kan agenten generera optimala planer som bestämmer korrekt handling i alla tillstånd.

### 1.6 Maskininlärning

Målsättningen med maskininlärning är att förse agenter med en viss förmåga att lära sig från observationer (data, evidens) istället för att uttryckligen bli programmerade hur varje situation ska hanteras [4]. Fältet är traditionellt indelat i tre typer av inlärning; övervakad inlärning (Supervised Learning), oövervakad inlärning (Unsupervised Learning) samt belöningsbaserad inlärning (Reinforcement Learning). Även om teknikerna som används för att möta utmaningarna delar teoretiska grunder med klassisk statistik och optimering, så skiljer de sig i inlärning genom att de är mer fokuserade på datadrivna tillvägagångssätt. Datadrivna tillvägagångssätt försöker härleda mönster och lösa problem med endast lite förkunskap och betraktar problemet mer som en "svart låda". Graden av förkunskaper om problemet som behövs, eller ens kan användas, varierar från metod till metod. Enklare



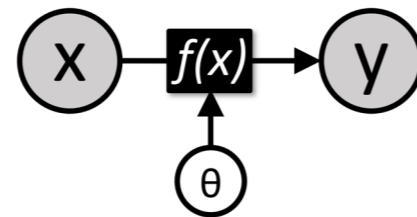
parameteriska modeller från klassisk statistik förutsätter ofta mer om problemets struktur, medan maskininlärningsmodeller som neuronät och beslutsträd är i grundutförande svarta lådor.

För att kunna modellera ett godtyckligt komplext problem behöver vi en tillräckligt kraftfull modellfamilj. Ett populärt alternativ är neuronät (rent formellt artificiella neuronät) som under milda antaganden är "universell approximatorer" och således kan approximera alla modeller givet tillräckligt mycket data. Dessutom kan ett neuronät med flera lager, under lämpliga förhållanden, lära sig att dra nytta av flera lager av abstraktion i problemet, så kallad djupinläring [16]. Detta har på senare tid gjort det möjligt att hantera större och mer komplexa problem än tidigare.

### 1.6.1 Övervakad inläring

Övervakad inläring är det mest etablerade området inom maskininläring och kan ses som ett sätt att lära sig en godtycklig uppgift utifrån exempel på olika problemställningar samt deras korrekta svar. Detta kan sedan användas för att exempelvis förutspå vad som ska hända för en ny problemställning genom att generalisera utifrån givna exempel. Begränsad förkunskap är ett typiskt antagande inför uppgiften, övervakad inläring betraktas ofta som en svart låda som omvandlar indata till utdata. Detta kan matematiskt ses som att lära sig en okänd matematisk funktion  $y = f(x)$ , där vi har par med exempel på både indata  $x$  liksom utdata  $y$ . Den okända funktionen  $f$  omsätts i praktiken till en approximation definierad via parametrar  $\theta$ , till exempel "vikter" som representerar kopplingsstyrkor i ett neuronät. Detta kan modelleras som det bayesianska nätverket i Figur 2. Eftersom vi i förväg inte känner till den korrekta modellen  $f$ , eller de parametrar  $\theta$  som bäst approximerar den, kan man även se detta som en form av osäkerhetsresonemang (se avsnitt 1.4).

Detta ger även en formell metod för att lära sig den okända funktionen, genom att tänka sig den som exempelvis ett neuronät med parametrar  $\theta$  och göra probabilistisk inferens på dessa parametrar utifrån givna data,  $P(\theta|X, Y)$  [13]. Som tidigare nämnts är probabilistisk inferens tyvärr ett beräkningsmässigt svårt problem när man har allt för många parametrar. Inlärningsalgoritmer är fortfarande ett aktivt forskningsområde där nuvarande bästa praxis bygger på ytterligare approximationer och heuristiker [4, 16].



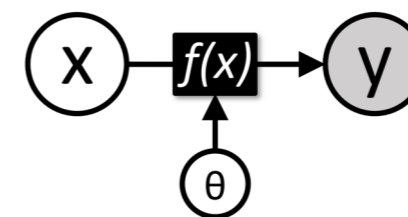
Figur 2: Övervakad inläring lär sig en okänd matematisk funktion  $f$  från indata  $x$  till utdata  $y$ , givet exempel på båda. Här illustreras det som ett bayesianskt nätverk med både  $x$  och  $y$  som skuggade observerade "bevisnoder". Den okända funktionen  $f$  är definierad av parametrar  $\theta$ , till exempel vikterna i ett neuronät, vilka måste läras från exemplen på  $x$  och  $y$ .

Övervakad inläring kan te sig som en rätt snäv matematisk konstruktion, men kan tillämpas på en förvånansvärt bred mängd av tillämpningar. Allt där indata och utdata kan kodas till siffror, till exempel text, bilder, video, tal med mera. Inom bildbehandling kan exempelvis indata vara pixlar och utsignalen kan vara svaret på frågan om bilden innehåller en bil. Inom affärsanalys kan indata vara kundinformation och utdata sannolikheten att kunden nappar på ett erbjudande. Begränsningen i denna teknik är att mer avancerade modeller som neuronät kräver en större mängd exempel på både indata och utdata, vilket behöver samlas in om det saknas.

Övervakad inläring är en teknik som i stor utsträckning används inom näringslivet. På senare tid har det i kombination med djupa neuronät bl.a. blivit möjligt att få spjutspetsresultat i bild-, video-, text- och talsammanhang. Idén bakom djupinläring är att vissa uppgifter kan brytas ner naturligt i flera abstraktionsnivåer, där varje lager bara behöver lösa en enklare uppgift istället för att ta sig an hela problemet direkt [16]. Det stigande intresset för dessa tekniker har även lett till ett antal någorlunda mogna mjukvaruramverk som Google Tensorflow [1]. Dessa möjliggör snabb utveckling av prototyper, liksom experimenterande med och spridning av, lämpliga typer av djupa neuronätmodeller. Djup övervakad inläring driver många av de senaste AI-tjänsterna som textöversättning, bildbehandling och taligenkänning i stora teknikföretag som Google och Facebook.

### 1.6.2 Oövervakad inläring

Oövervakad inläring försöker eliminera behovet av att tillhandahålla korrekta par med både in- och utdata,  $x$  och  $y$ , istället försöker den hitta mönster direkt från tillgängliga data. Till exempel att sammanfatta data genom att hitta kluster



Figur 3: Oövervakad inläring försöker hitta mönster i data. Precis som i övervakad inläring handlar det om ett okänt samband mellan indata och utdata, men utan givna exempel på indatan. Presenterat som ett bayesianskt nätverk visualiseras det genom att endast de observerade utsignalerna är skuggade.

av liknande data, eller genom att identifiera en mindre mängd variabler som är tillräckliga för att förklara variationen i alla ursprungliga variabler.

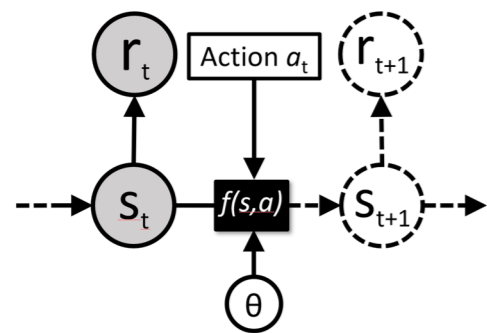
I många fall kan detta matematiskt ses som en härledning av både den okända svarta låd-funktionen  $f(x)$  och den underliggande datagenererande processen  $x$  (nu icke-observerad) från den observerade datan  $y$ . Detta visualiseras i Figur 3. Till exempel går det att observera filmbetygen ( $y$ ) för användare på Netflix och försöka hitta underliggande teman i filmer som användare gillar ( $x$ ), även om dessa endast indirekt observeras genom individuella filmbetyg. Ett annat exempel är att observera textdata och försöka automatiskt sortera liknande texter i grupper, eller tilldela dem en begränsad uppsättning av ämnen. Det kan även vara så enkelt som att hitta korrelationer mellan produkter i till exempel en kundvagn, så kallad "Association Rule Mining".

### 1.6.3 Belöningsbaserad inläring

Belöningsbaserad inläring (Reinforcement Learning, RL) utvidgar definition av maskininläring från modellering och prediktion till att inkludera beslutsfattande (se avsnitt 1.5). Algoritmen är här en rationell agent som lär sig hur den ska agera i en osäker omgivning genom att prova sig fram (Trial and Error). Agenten behöver bara problemställningen vid den aktuella tidpunkten, här ett tillstånd  $s_t$ , och den omedelbara "belöningen",  $r_t$ , för det tillståndet. Belöningen kan representera vinsten av ett affärsbeslut, eller omvänt kostnad som material och miljö. Sådana vinster och kostnader representeras av en siffra, belöningen, som agenten använder för återkoppling kring hur den kan förbättras. Som Figur 4 visar måste agenten för varje steg välja en handling,  $a_t$ , vilket kommer att leda till ett nytt tillstånd, samt en ny belöning. Detta är ett sekventiellt beslutsproblem då handlingarnas



ordning avgör den slutgiltiga ordningen av tillstånd och belöning. Agenten måste planera ett antal steg i förväg för att hitta rätt ordning av handlingar som maximerar den totala belöningen över tiden. I klassisk belöningsbaserad inlärning antas inga förkunskaper om vare sig belöningen eller hur tillståndet förändras ( $s_{t+1} = f(s_t, a)$ , där  $f$  är okänd) som svar på en handling.



Figur 4: Belöningsbaserad inlärning utvidgar definitionen av inlärning till att simultant lösa ett beslutsproblem. Givet ett problemtillstånd,  $s_t$ , och en belöning för det tillståndet,  $r_t$ , kommer agenten försöka lära sig att minimera kostnaderna över tiden. Vi kan visualisera detta som ett bayesianskt nätverk utökat med handlingsnoder (ett så kallat Influence Diagram) där den vid varje tidpunkt,  $t$ , måste fatta ett beslut baserat på observerade tillstånd och belöningar.

Belöningsbaserad inlärning erbjuder ytterligare ett inlärningssätt som inte är beroende av korrekta exempel. I många fall i den verkliga världen är belöningarna och kostnaderna för problemtillstånden kända och lätta att definiera matematiskt, vilket gör att sådana algoritmer själva kan lära sig, antingen via simulering eller i den aktuella miljön.

Hittills har de mest iögonfallande framstegen varit i simulerade miljöer och spel, till exempel att lära sig att spela dataspel direkt från videoströmmar [26]. Nyligen använde Google Deepmind

belöningsbaserad inlärning i kombination med trädsökningsteknik för att bygga en agent för spelet Go som slog den mänskliga mästaren [34]. Agenten startade med övervakad inlärning baserad på exempel från matcher som spelades av stormästare. Därefter kunde den förbättra sig automatiskt genom att spela mot olika versioner av sig själv och belönas vid vinst. Eftersom de omfattar såväl inlärning som planering anses sådana algoritmer även naturligt lämpade för användning inom till exempel robotik och autonomi. Vissa problem kvarstår dock, särskilt när det gäller säkerhet, data- och beräkningskrav, vilket vi avhandlat tidigare (se exempelvis [2]).

Eftersom belöningsbaserad inlärning är generellt tillämpbar på beslutsuppgifter med osäkerhet, finns det en mängd tillämpningar inom produktion och underhåll. Allt från lagerstyrning och schemaläggning till prediktivt underhåll kan involvera osäkra variabler. Approximativa metoder som belöningsbaserad inlärning kan behövas då exakta lösningar ofta är omöjliga att beräkna. Eftersom många affärsproblem i den verkliga världen faller inom denna kategori kan belöningsbaserad inlärning ses som ett lovande tillvägagångssätt att lösa komplexa sekventiella affärsbeslut i allmänhet.

En vanlig begränsning av dessa metoder är dataeffektivitet, eftersom det ofta kan krävas miljoner interaktioner med omgivningen. Detta problem kan dock förminska genom att träna agenten i en simulationsmodell som är en tillräckligt bra approximation av dess målmiljö. Om en sådan modell saknas kan man först nyttja övervakad inlärning för att lära sig en problemmodell  $f$ , för att sedan använda den inlärda modellen som en simulator för att lära sig en lösning på det sekventiella beslutsproblemet. Det vill säga, vilka handlingar  $a_t$  som ger minimal kostnad. Detta ger en generell två-delad strategi för problem där resultatet av en handling är

i förväg okänt, först med hjälp av ett lämpligt övervakat inlärningsangreppssätt för att lära sig en problemmodell och sedan automatiserat beslutsfattande via belöningsbaserad inlärning på denna modell. Dessa steg kan även upprepas för att förbättra resultatet.

#### 1.6.4 Allmänna råd för tillämpningar

Även om maskininlärning kan framstå som en universallösning finns det begränsningar för inlärningsmetoderna, speciellt med avseende på tillgänglighet av stora och användbara datamängder för uppgiften. Detta kan vara en flaskhals i traditionella industri- och tillverkningssektorer där vinstmarginalerna är små och datainsamling kan kräva investeringar i sensorer och samordning med kunder. Sekundära flaskhalsar för maskininlärning inkluderar tillgänglig beräkningskraft, där det främst handlar om att träna dessa modeller utifrån data, men delvis även när man använder dem för att göra nya prediktioner.

Allmänt tillgänglig mjukvara för maskininlärning fungerar bäst när det finns gott om användbara data, och helst ett mindre antal relevanta variabler i indata. Om det finns ett stort antal relevanta variabler kan generiska verktyg för svartlådemodeller ge sämre resultat. Däremot för problemområden med inbyggd hierarkisk struktur som text, bild och tal kan djupinlärningstekniker utnyttja stora datamängder och effektiv träning på grafik kort för att överkomma denna begränsning. Inom exempelvis bildbehandling, där inmatningsbilder kan innehålla en miljon pixelvärden, har djupinlärningstekniker kommit att dominera.

Problem med en liten datamängd i förhållande till antalet variabler förblir ett problem. Det är inte ovanligt att ett problem med till synes stor datamängd i praktiken är ett flertal delvis

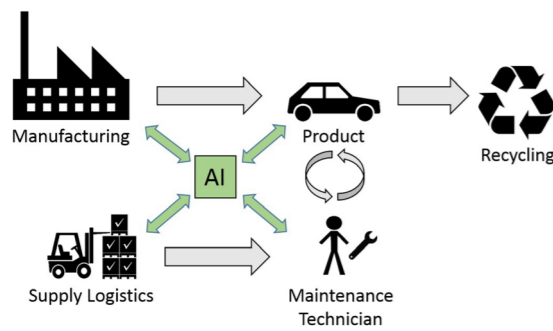
relaterade problem, som var för sig får liten mängd data. Vid sådana tillfällen kan det vara en fördel att inkludera de förkunskaper och struktur som finns för den uppgift som ska läras in. Probabilistiska tillvägagångssätt är det teoretiska ramverk som är lämpligast för att föra in förkunskaper [4, 13], där de intuitivt kan visualiseras som inferens i ett bayesianskt nätverk strukturerat efter den aktuella uppgiften. En tydlig fördel med probabilistisk inlärning är att den också rapporterar konfidensnivån för resultaten och möjliggör enklare tolkning. Sådana bayesianska inlärningstekniker kan emellertid vara alltför beräkningsintensiva varför probabilistiska modeller ofta kräver mer expertis att tillämpa. Ett enklare men mindre robust alternativ kan vara att nyttja neuronät med ramverk som Tensorflow för att strukturera nätet efter problemet. Till exempel kan ett flertal relaterade problem med fördel modelleras tillsammans som ett delat neuronät där varje delproblem har sin egen utnod.

Hittills har vi fokuserat främst på modellering och prediktion, att förutsäga någon aspekt av verksamheten bättre. Sådana modeller som tränats via maskininlärning kan sedan användas som beslutsstöd för manuellt beslutsfattande, men som förklarades i avsnitt 1.6.3 kan dessa också automatiseras via beslutsteori och belöningsbaserat lärande. Ett annat vanligt tillämpningsområde är anomalitetsdetektering (avsnitt 2.2.2), det vill säga att automatiskt hitta och flagga för saker utanför det ”normala”, datapunkter som eventuellt visar på ett avvikande beteende. När dessa har identifierats kan de granskas av domänexperter för att ta reda på vad felet är och om det innebär ett problem. I en tillverkningsprocess kan exempelvis en identifierad avvikelse i en komponents livslängd upptäckas så att det går att undersöka den underliggande orsaken. Detta kan göras både via övervakad och oövervakad inlärning.

# 2

## Tillämpning av AI inom underhåll och hållbarhet

I detta avsnitt beskrivs hur AI kan tillämpas på produktunderhåll och hållbarhetsproblem. Vi kommer att använda ett urval av de AI-tekniker som presenterades i den första delen av dokumentet, med fokus på maskininlärning. Diskussionen kommer att kretsa kring det förenklade produktlivscykel diagrammet som visas i Figur 5, vilket växte fram ur diskussionerna med de deltagande företagen.



Figur 5: En förenklad produktlivscykel från ett underhållsperspektiv.

Produktlivscykeln börjar med de val som gjorts under tillverkningen, exempelvis vilka material och komponenter som valts för en produkt. Utan att begränsa oss, kommer vi för enkelhetens skull fortsättningsvis anta att produkten är någon slags maskin. I det här projektet har vi främst tittat på underhåll av fordon såsom gaffeltruckar eller flygplan. Under maskinens livscykel erbjuder tillverkaren både regelbundet underhåll och akuta reparationstjänster via underhållstekniker, vanligtvis enligt ett överenskommet serviceavtal. Detta avtal kan även innehålla avtal om servicenivån, där godtagbara avbrott för användandet av maskinen anges, och kompensation för eventuella avvikelser från

detta beskrivs i detalj. För tydlighetens skull utgår vi från att användningen av maskinen säljs som en tjänst. Detta innebär tillverkaren och kunden delar intresset för att minska den totala livscykelkostnaden och att tillverkaren till stor del har kontroll över maskinens livscykel.

Syftet här är att undersöka hur AI kan förbättra tillverknings- och underhållsprocessen så att antalet fel minimeras, livslängden ökas och i slutändan att finansiella och miljömässiga kostnader reduceras. AI-tekniker har stor potential inom automatisering och förbättring av flera aspekter i denna process, till exempel genom att i förväg förutsäga fel och planera underhåll efter behov.

Även om vi valde att illustrera AI som en central låda i Figur 5, kommer det sannolikt i praktiken att börja som ett antal separata system för specifika delproblem. Det är viktigt att förtydliga att en förutsättning för att djupt integrera AI i en traditionell produktlivscykel kommer att kräva en adekvat informationsteknologisk infrastruktur för insamling, lagring och bearbetning av relevant data. Om AI-funktionaliteten måste byggas in i till exempel ett fordon eller vara centralt placerad i en molntjänst beror på realtidskraven för uppgiften, tillgänglig bandbredd, samt processorkraften ombord på fordonet. Centralisering kan i teorin leda till bättre beslutsfattande genom kontinuerlig aggregering av information från alla maskiner samt från alla steg i deras livscykel. Att exempelvis känna till det beräknade underhållsbehovet hos en fordonsflotta hjälper inte bara den enskilda underhållsteknikern utan är också ett stöd för logistiken kring reservdelslager och produktdesign. De flesta uppgifterna inom underhåll har inte något realtidskrav, vilket gör att en till största delen centraliserad AI är tekniskt möjlig. I detta avseende kan andra trender som IoT, Big Data-infrastruktur samt molntjänster (Cloud Computing) vara viktiga drivkrafter för AI.

I följande avsnitt presenterar vi en taxanomi av olika typer av problem som kan uppstå vid underhåll. Denna är baserad på diskussioner med de deltagande företagen i projektet och hur de kan anpassa de AI-tekniker som vi tagit upp i tidigare avsnitt. Eftersom datadrivna AI-tekniker visade sig vara de mest intressanta ligger fokus på dessa.

### 2.1 Klassificering av AI-problem vid underhåll

Baserat på diskussioner med de deltagande företagen och litteraturstudier har vi tagit fram följande klassificering av lovande AI-tillämpningar:

- Modellering av driftavbrott
  - Diagnostik (vad gick fel?)
  - Anomalitetsdetektering (har något gått fel?)
  - Prognostik (när kommer det att gå fel?)
- Automatiserat beslutsfattande
  - Planering av förebyggande underhåll (vad och när ska underhållas?)
  - Beslutsstöd och planering för reparation (hur vi ska reparera?)
  - Reservdelslagring (stora reservdelslager eller riskera längre driftstopp?)
  - Automatiserad maskinjustering (kan användningen effektiviseras?)

### 2.2 Modellering av driftavbrott

I de flesta fall kan modellering och härledning från data reduceras till olika kombinationer av övervakad och oövervakad inlärning som beskrivet i avsnitt 1.6.1 and 1.6.2. Även om terminologin är annorlunda så utgör underhållssektorn inget undantag.

Vissa domänspecifika övervägande behövs när det kommer till vilka modeller som ska läras in och vilken inlärningsalgoritm som ska användas.

Helt datadrivna svartlådemodeller som neuronät kräver normalt den minsta mängden arbete. I synnerhet är programvaror för djupa neuronät som Tensorflow [1] relativt mogna. Å andra sidan är dessa datakrävande och data från fordon kan vara kostsamma att samla in. Behov att inkorporera förkunskaper kan därför uppstå, exempelvis i form av fysiska modeller av fordonsbeteende eller åtminstone av problemets struktur. Det tyder på att det krävs mer skraddarsydda modeller och kanske till och med probabilistiska inferensmetoder vid stor databrist. Även om relativt mogna verktyg finns för probabilistisk modellering [9] kräver sådana domänspecifika modeller normalt mer arbete att implementera. Att bestämma vilket angreppssätt som är mest lämpligt måste göras från fall till fall, men svartlådemodeller kan vara en bra utgångspunkt.

Vi lämnar dock implementationsfrågan och går istället vidare till att beskriva vilka typer av problem som kan lösas med dessa tekniker. Vi använder terminologin från Jardine et al [20] för att klassificera dessa problem i de breda kategorierna nedan.

#### 2.2.1 Diagnostik

Diagnostik syftar till att diagnostisera misstänkta eller faktiska fel som uppstår i en produkt, exempelvis som ett beslutsstöd för underhållstekniker på plats eller som information direkt till kunden. Med tillräckligt många exempel på tidigare problem och dess lösningar är det enkelt att föra över detta till ett ramverk för maskininlärning som ett övervakat inlärningsproblem (se avsnitt 1.6.1). Indata  $x$  kan till exempel bestå av systemmätningar, fordonets ålder, användning, fordonstyp och tidigare servicehistorik, och som utdata den problemorsak som teknikern har identifierat. Det är sedan enkelt att lära sig modellen  $y = f(x)$  för att klassificera nya observerade indata efter sannolika orsaker.

Detta är den enklaste tillämpningen av maskin-inlärningstekniker, men också den minsta beträffande omfattning. Genom att utnyttja framsteg inom djupinlärning kan även högdimensionell och semistrukturerad data som text, ljud eller bilder användas som indata. Beroende på modellens komplexitet kan det emellertid också krävas ett stort antal exempel för att lära sig en användbar diagnostisk modell. Sådan djupinlärning är redan vanlig inom medicinsk diagnos baserad på bilder [24], och det finns rapporter om att General Electrics använder sådan AI för att diagnostisera problem på jetmotorer utifrån det ljud de genererar [41].

### 2.2.2 Anomalitetsdetektering och övervakning

Anomalitetsdetektering försöker upptäcka onormala händelser vilket kan användas för att besluta om inspektion eller förebyggande underhåll innan ett fel inträffar. En diagnosmodell kan också användas förebyggande för detta vilket ibland sorteras in under diagnostik i litteratur om underhåll.

Anomalitetsdetektering kan emellertid även dra nytta av en annan metod för maskininlärning än bara förlita sig på manuellt markerade exempel på fel. Eftersom målet är att upptäcka avvikelser från vissa normala systemtillstånd som varierar över tiden är ett självövervakat eller oövervakat tillvägagångssätt också tillämpligt. Enligt det tidigare exemplet i avsnitt 1.6.1 kan en *självövervakad* modell  $x_{t+1} = f(x_t)$  utvecklas till att förutsäga ett nytt tillstånd baserat på föregående värden. Prediktionen kan då jämföras med utdata över tid och flagga för stora avvikelser som behöver kontrolleras. En sådan självövervakad inlärning har fördelen att det inte behövs manuell märkning av fel, och stora mängder data kan då samlas in via exempelvis IoT-lösningar.

Om misstanken finns om ett viktigt icke-observerat tillstånd i problemet går det även att använda sig

av oövervakade inlärningstekniker för att härleda detta. Sedan görs en detektering på ett sådant ”dolt” tillstånd istället för att göra det direkt på mätningarna.

Sådan självövervakad och oövervakad anomalitetsdetekteringsteknik kan inte erbjuda en diagnos eller ens användas som bevis på att något är fel. De kan dock fungera som en varningsindikator att systembeteendet är ovanligt, vilket kan motivera ytterligare inspektion.

### 2.2.3 Prognostik

Prognostik (Prognostics) försöker bygga modeller baserade på när ett fordon eller komponent kommer att falla, det vill säga att förutsäga livslängden, överlevnadsanalys. Ibland kallas detta den återstående användbara livslängden (Remaining Useful Life, RUL) eller tid till fallering (Time-To-Failure, TTF). Den går därmed längre än diagnostikmodeller som i första hand försöker svara på frågan om vad som är fel just nu. En tillräckligt detaljerad prognosmodell kan faktiskt omfatta diagnostik, då den kan beräkna det för tillfället mest troliga felet av de förväntade fel som förutsagts för varje maskindel.

Även om sådana prognosmodeller är kraftfullare än diagnosmodeller är de också något svårare än ett typiskt övervakat inlärningsproblem. Vi observerar vanligtvis bara livslängden indirekt, när de går sönder. Om inte medellivslängden är kort i förhållande till datainsamlingsperioden måste sådana modeller avspegla denna typ av ”censurerade” data [42]. En elegant lösning är att använda en hierarkisk probabilistisk modell där livslängden kan behandlas som ett internt tillstånd som härleds från observationer. Denna metod har också nyligen kombinerats med djupa neuronät [21, 31]. Ett enklare tillvägagångssätt är att dela upp livslängden i fixerade tidsintervall. Eftersom både antal maskiner och antal fel är känt i varje

tidsintervall, går det att beräkna sannolikheten för fel uppstår per intervall. Detta kringgår problemet på bekostnad av att modellen blir mindre precis över tid. Speciellt bör man vara försiktig med att avvikelser kan ackumulera om man prognosticerar flera tidssteg framåt, men om detta är av mindre intresse indikerar en av våra fallstudier att det kan fungera tillfredställande. Att anpassa djupinlärning för överlevnadsanalys är fortfarande ett område under utveckling. En variant som fokuserar på tidsseriesdata, vilket förefaller lovande för till exempel sensordata från maskiner, föreslogs nyligen i en magisteruppsats [25]. Det finns också flera variationer på detta tema med öppen källod, till exempel [35].

## 2.3 Automatiserat beslutsfattande

Beslutsfattande är ett viktigt ämne inom underhåll, i slutändan vill man antingen använda AI-tekniker som beslutsstöd, eller för att automatisera hela processen. Om det till exempel finns en modell över sannolikheten för maskinfel, eller att denna har lärts in med AI-teknik enligt avsnitt 2.2, kan den användas för att optimera underhållsprocessen. Dessa beslutsproblem kan lösas algoritmiskt och automatiseras via antingen klassisk beslutsteknik eller mer generella AI-tekniker såsom belöningsbaserad inlärning (se avsnitt 1.6.3). För enkelhetens skull antar vi att både miljökostnader och finansiella kostnader är sammanslagna till en kostnad, så att beslutsproblemet endast har ett mål. Det finns även sätt att göra avvägningar mellan dessa när de ligger i konflikt, så kallad multi-objektiv optimering. Här listar vi några idéer kring användningen av AI för att automatisera och förbättra beslutsfattandet vid underhållsproblem.

### 2.3.1 Planering av förebyggande underhåll

Inlärda modeller av tid till fel, per del eller per maskin, kan användas för ett antal planeringsproblem inom underhåll. Prognosmodeller som

ger en sannolikhetsfördelning över fel är ideala för planering av underhåll och förebyggande ersättning av maskindelar. En modell som kan göra informativa förutsägelser för varje maskin eller fordon skulle också kunna anpassa underhållet för varje enskild maskin. En felmodell över enkla fixerade tidsintervall kan användas för att avgöra vilka delar som i förebyggande syfte ska ersättas vid ett underhållstillfälle, medan en tidskontinuerlig felmodell också skulle möjliggöra en optimering av själva underhållsintervallen.

Det finns betydande forskning inom underhållsplanering [7, 17]. Dock brukar de bygga på enklare prediktiva modeller än vad som föreslås här, liksom ganska specialiserade lösningar på problemets struktur. Vad belöningsbaserad inlärning kan tillföra här är en *generell* metod för att hitta approximativa lösningar även för stora beslutsproblem med osäkerhet. Just osäkerhet är en viktig aspekt om man vill dra nytta av approximativa datadrivna prognostikmodeller (avsnitt 2.2).

Som tidigare nämnts kan diagnos- och anomalitetsdetektering också användas förebyggande som en indikator på när det är dags att involvera en underhållstekniker. Med hjälp av utbyggnaden av IoT-lösningar kan detta också automatiseras för att kontakta en underhållstekniker automatiskt för en närmare titt. När en sådan inspektions- eller reparationsåtgärd ska ske är ett beslutsproblem och en optimal beslutspolicy kan tas fram via till exempel belöningsbaserad inlärning. Eftersom anomalitetsdetektering i allmänhet kräver mindre data än prognostik, kan det vara rimligt att använda en tudelad metod med både anomalitetsdetektering för tidig varning och en mer detaljerad tid-till-felmodell för förebyggande underhåll eller utbyte av enskilda maskindelar.

Till sist kan maskiner eller maskindelar bli dyrare att underhålla över tiden eller hamna på



efterkälken vad gäller energieffektivitet eller andra miljökostnader jämfört med nyare teknik. Ett exempel är fordon med förbränningsmotorer. Vid någon tidpunkt kan det vara optimalt att helt enkelt återvinna maskinen (eller delen) och ersätta den med en ny modell (illustrerat som den sista pilen i Figur 5). Detta är också ett beslutsproblem, och eftersom anskaffningsvärdet måste vägas mot diskonterade framtida underhållskostnader är det ett sekventiellt beslutsproblem som är lämpligt för belöningsbaserad inlärning.

### 2.3.2 Beslutsstöd och planering för reparationer

En diagnostisk modell som direkt förutsäger problemorsaken kan användas som ett beslutsstöd på plats för vilken maskindel som ska ersättas av en underhållstekniker. En mer avancerad användning av beslutsstöd är om en skräddarsydd diagnostisk modell av fordonet finns som ett bayesianskt nätverk, men orsaken till felet är ännu inte känd. Detta är en form av probabilistisk inferens. Sådana bayesianska nätverk kan användas för att automatisera felsökningsprocessen genom att automatiskt skapa planer som ger mest information om problemets orsak i så få steg som möjligt [39].

### 2.3.3 Reservdelslogistik

Modern logistik minskar behovet av lagerytor; när och var lagerföringen görs är avgörande för

en effektiv lagerhantering. En lovande idé är att använda de inlärd prognosmodellerna för maskinfel även för att effektivisera lagerhållningen av maskindelar. Det skulle även vara möjligt att göra prognoser för varje fordon för att driva sådan logistik till det yttersta. Beslutsteori och belöningsbaserad inlärning erbjuder ett sätt att väga in de risker som följer av driftstopp och servicenivåavtal, samt väga detta mot kostnaden för potentiellt överflödigt lagerhållning.

### 2.3.4 Automatiserad maskinjustering

Slutligen, med tanke på IoT-tillväxten, så är det enklare än någonsin att samla in data från maskiner som fordonsflottor. Som tidigare nämnts kan sådan data användas för att kontinuerligt lära upp prognostiska modeller för maskinfel. Sådana modeller skulle kunna vara betingade på olika driftparametrar för maskinen i fråga. Via IoT skulle dessa driftparametrar sedan kunna optimeras automatiskt så att finansiella och miljömässiga kostnader minimeras. Vissa biltillverkare skickar till exempel redan rutinmässigt ut mjukvaruuppdateringar. Dessa kan anpassa parametrar som styr hur ett fordon fungerar, som motor- eller batteriparametrar, som potentiellt påverkar livslängd, energieffektivitet och miljökostnader. Genom att nyttja övervakad inlärning kombinerat med beslutsteori eller belöningsbaserad inlärning skulle maskinparametrar automatiskt kunna optimeras i drift utefter maskinens användningsprofil.





# 3

## Referenslista

1. Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In OSDI, volume 16, pages 265–283, 2016.
2. Olov Andersson, Methods for Scalable and Safe Robot Learning. Licentiate Thesis No 1780. Linköping Studies in Science and Technology, Linköping University, 2017.
3. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.
4. C. M. Bishop. Pattern recognition and machine learning. Springer, 2006.
5. R. J. Brachman and H. J. Levesque. Knowledge Representation and Reasoning. Morgan Kaufman, 2004.
6. C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothakris, and S. Colton. A survey of monte carlo tree search. IEEE Transactions on Computational Intelligence and AI in Games, 2012.
7. Jing Cai, Xin Li, and Xi Chen. Optimization of aeroengine shop visit decisions based on remaining useful life and stochastic repair time. Mathematical Problems in Engineering, 2016.
8. M. Campbell, A. J. Hoane, and F. H. Hsu. Deep blue. Artificial Intelligence, 134, 2002.
9. Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. Journal of Statistical Software, Articles, 76(1):1–32, 2017.
10. Darwiche. Modeling and Reasoning with Bayesian Networks. Cambridge University Press, 2009.
11. E. Davis. Representations of Commonsense Reasoning. Morgan Kaufman, 1990.
12. R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A\*. Journal of the ACM, 32(3):505–536, 1985.
13. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. Bayesian Data Analysis, Third Edition. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.
14. M. Ghallab, D. Nau, and P. Traverso. Automated Planning and Acting. Cambridge University Press, 2016.
15. Malik Ghallab, Dana Nau, and Paolo Traverso. Automated Planning: Theory and Practice. Morgan Kaufmann Publishers, San Francisco, California, USA, 2004.
16. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.
17. Emil Gustavsson, Michael Patriksson, Ann-Brith Strömberg, Adam Wojciechowski, and Magnus Önnheim. Preventive maintenance scheduling of multi-component systems with interval costs. Computers & Industrial Engineering, 76:390 – 400, 2014.
18. P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 1968.
19. H. Hoos and T. Stützle. Stochastic Local Search: Foundations and Applications. Morgan Kaufman, 2005.
20. Andrew KS Jardine, Daming Lin, and Dragan Banjevic. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical systems and signal processing, 20(7):1483–1510, 2006.
21. Jared L Katzman, Uri Shaham, Alexander Clonger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC medical research methodology, 18(1):24, 2018.
22. D. Koeller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009.
23. S. LaValle. Planning Algorithms. Cambridge University Press, 2006.
24. Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. Medical image analysis, 42:60–88, 2017.
25. Egil Martinsson. Wtte-rnn: Weibull time to event recurrent neural network. Master’s thesis, University of Gothenburg, Sweden, 2016
26. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, and S. Petersen. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.
27. N. J. Nilsson. Artificial Intelligence: A New Synthesis. Morgan Kauffman, 1998.
28. Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Artificial Intelligence. Addison-Wesley, 1984.
29. Pearl. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman, 1988.
30. D. Poole and A. K. Mackworth. Artificial Intelligence: Foundations of Computational Agents. Cambridge University Press, 2017.
31. Rajesh Ranganath, Adler Perotte, Nöemie Elhadad, and David Blei. Deep survival analysis. In Machine Learning for Healthcare Conference, pages 101–114, 2016.
32. S. Russell and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition edition, 2010.
33. Y. Shoham and K. Leyton-Brown. Multi Agent Systems: Algorithmic, Game- Theoretic, and Logical Foundations. Cambridge University Press, 2009.
34. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, and S. Dieleman et al. Mastering the game of go with deep neural networks and tree search. Nature, 529(7587):484–489, 2016.
35. Gianmario Spacagna. Deep Time-to-Failure. Github, 15 January 2018. Available: <https://github.com/gm-spacagna/deep-ttf>, 2018. [Online; accessed 19-July-2018].
36. R. D. Sriram. Intelligent Systems for Engineering. Springer, 1997.
37. S. Thrun, W. Burgard, D. Fox, and R. C. Arkin. Probabilistic Robotics. MIT Press, 2005.
38. F. van Harmelen, V. Lifschitz, and B. Porter, editors. Handbook of Knowledge Representation. Elsevier, 2008.
39. Håkan Warnquist. Troubleshooting Trucks: Automated Planning and Diagnosis. PhD thesis, Linköping University, Sweden, 2015.
40. Wooldridge. An Introduction to Multi Agent Systems. Wiley, 2009.
41. Elizabeth Woyke. General Electric Builds an AI Workforce. MIT Technology Review, 41. 27 June 2017 Available: <https://www.technologyreview.com/s/607962/general-electric-builds-an-ai-workforce/>, 2017. [Online; accessed 19-July-2018].
42. Ishwaran, U.B. Kogalur, E.H. Blackstone, M.S. Lauer. Random survival forests. Annals of Applied Statistics. 2(3):841-60, 2008.

---

## Sammanfattning

Syftet med denna guide är att ge en bred bild över Artificiell Intelligens (AI) och de områden där vi tror att AI-tekniker kan komma till nytta. AI-tekniker kan vara särskilt användbara i tillämpningar inom drift och underhåll, liksom inom hållbarhetsområdet, för att öka svenska företags konkurrenskraft och effektivitet.

Denna översikt bygger på diskussioner med de industriella aktörerna i det Vinnova-finansierade FoI-projektet Simon inom Produktion2030, samt litteraturstudier. Guiden består av två delar där den första ger en generell översikt av AI som område, medan den andra specifikt tar upp dess tillämpningar inom drift och underhåll.

---

Med stöd från:



FORMAS



STRATEGISKA  
INNOVATIONS-  
PROGRAM

