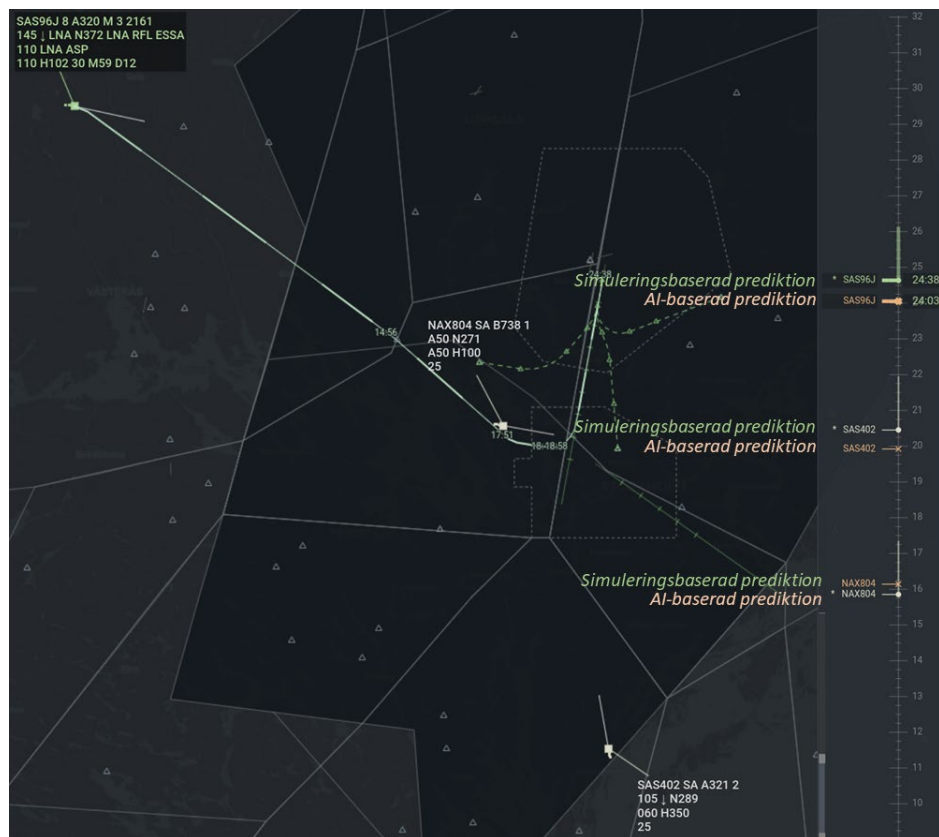


AI-baserad landningsprediktion

– Slutrapport projektet AI-baserad
landningsprediktion inom IRIS-
programmet

2022-01-17



Martin Castor, Martin Insulander, Jonathan Borgvall

Innehåll

1. Sammanfattning	4
2. Executive summary	5
3. Projektets deltagare	6
4. Introduktion	7
4.1. Läsanvisning	7
4.2. Projektets bakgrund	7
4.3. Teoribakgrund	10
5. Metod	14
5.1. Data	14
5.2. Förståelse för data och dess spridning	17
5.3. Modelleringsprocess	19
6. Resultat	20
6.1. Utvärderade modeller	20
6.2. Föreslagen modell	21
6.3. Implementation i demonstrator	21
6.4. Användartest	25
7. Diskussion	28
8. Rekommendationer	29
9. Referenser	30

Figurer

Figur 1. Visuell beskrivning av projektets grundidé.	9
Figur 2. Skillnaden mellan klassisk programmering och maskininlärning	11
Figur 3. Schematisk beskrivning av klassiskt neuralt nätverk	11
Figur 4. Schematisk beskrivning av ett klassiskt feed-forward neuralt nätverk med fler lager.	12
Figur 5. Utdrag ur MNIST databasen.....	13
Figur 6. Projektets faser.	14
Figur 7. En visualisering av särdragen i träningsdatabasen.	17
Figur 8. Spridningsdiagram mellan "Time to go" och "Direct Distance to Destination".	18
Figur 9. Spridningsdiagram mellan "Direct distance to Destination" och den i efterhand beräknade "Track Distance to Destination".	18
Figur 10. Föreslagen modells arkitektur	21
Figur 11. Exempel på relativt precisa prediktioner med förklaringar av viktiga informationselement.....	22
Figur 12. Exempel på relativt precis prediktion.	23
Figur 13. Exempel på mindre precis prediktion	24
Figur 14. Exempel på luftläget från användartestet.	25
Figur 15. Jämförelse rutt enligt färdplan och vad ANN troligen utgått i sin prediktion.....	26

Tabeller

Tabell 1. Jämförelse datatillgänglighet Flightradar24 och GAIA systemet. ..	14
Tabell 2. Transformerad träningsdata.....	16
Tabell 3. Alternativa modeller.	19
Tabell 4. Prediktionsförmåga hos utvärderade modeller.	20

1. Sammanfattning

Syftet med forsknings och innovationsprojektet ”AI-baserad landningsprediktion” inom IRIS-programmet var att studera om artificiella neurala nätverk (ANN), vilket är en typ av mycket användbara algoritmer från området Artificiell Intelligens (AI), kan användas för att prediktera landningstider och därigenom landningssekvenser för flygplan på väg in till en flygplats. Inom IRIS-programmet studeras olika åtgärder som kan hjälpa flygledare att tilldela fler kurvade RNP AR-inflygningar. Användning av kurvade RNP AR-inflygningar medför flera effektivitets- och miljövinster och ökad användning av dessa landningsprocedurer är en viktig del för att Arlanda ska kunna möta kraven i sitt miljötillstånd. Kurvade RNP AR-inflygningar är dock mer kognitivt krävande för flygledarna och ett beslutsstöd som hjälper flygledarna att se den troliga landningssekvensen är eventuellt ett verktyg som avlastar flygledarna.

I projektet användes luftlägesdata för flygtrafiken under 2019 för att konstruera en träningsdatabas på vilket flera ANN tränades. Flera olika konfigurationer av ANN prövades för att se vilken ANN-modell som kunde prediktera landningstider och landningssekvensen mest precist. Den ANN-modell som predikterade mest precist integrerades därefter med en flygledningssimulator för att demonstrera hur prediktionerna skulle kunna användas operativt. Ett användartest med en mycket erfaren operativ flygledare genomfördes.

Den ANN-modell som användes i demonstratorn lyckades prediktera landningstiden med ett medelfel strax under ± 1 minut, vilket var tillräckligt för att kunna presentera en användbar landningssekvensindikation i de ”etiketter” som används för att presentera information om flygplanen i det operativa flygledningssystemet.

En demonstration gällande skillnaden i prediktionsförmåga mellan AI-baserad prediktion enligt den typ som studerats i projektet och simuleringsbaserad prediktion för finns tillgänglig på <https://geistt.com/projects/aibaland>. Lösenordet finns tillgängligt via kontakt med martin.castor@geistt.com, martin.insulander@geistt.com eller anette.nas@swedavia.se.

Notera att den lösning som projektet studerade inte kräver några större infrastruktursatsningar, utan använder sig av öppet tillgänglig data och kan se ”hur det brukar bli” innan trafiksituationer uppstått och flygledarna börjat hantera situationen. Detta kan vara ett kostnadseffektivt sätt att komplettera andra sätt att prediktera landningstider och landningssekvens. Ett exempel på ett annat sätt att göra denna prediktion är att ladda ner information från flygplanens färdatorer, som dock bara tar hänsyn till de enskilda flygplanens planering innan flygledarna börjar ge klareringar och som kräver ny teknik på många flygplan. Rapporten avslutas med ett antal rekommendationer för fortsatt forskning och utvecklingsverksamhet.

2. Executive summary

The purpose of the research and innovation project “AI-based landing predictions” within the IRIS-programme was to study whether artificial neural networks (ANN), which are a class of very useful algorithms from the Artificial Intelligence (AI) field, can be used to predict landing times and thus landing sequences for aircraft inbound to an airport. The IRIS-programme studies various actions that can be taken in order to increase the number of curved RNP AR-approaches that air traffic controllers assign. Use of curved RNP AR-approaches leads to several positive efficiency and environmental effects and their increased frequency of use is an important requirement in the environmental permit for Arlanda Airport. However, curved RNP AR-approaches are more cognitively demanding for the air traffic controllers and a decision support tool that helps the air traffic controllers to see the probable landing sequence can potentially reduce the cognitive load on the air traffic controllers.

The project used air traffic data for 2019 to construct a training database on which several different ANN’s were trained. Several different configurations of the ANN’s were tested in order to see which that could predict landing times and the landing sequence with the highest accuracy. The ANN-model that predicted most accurately was then integrated into an air traffic control simulator to demonstrate how the predictions could be used operatively. A user test with an operative very senior air traffic controller was conducted.

The ANN-model that was used in the demonstrator was able to predict the landing times with a mean error of ± 1 minute, which was sufficient to be able to present a useful indication of landing sequence in the labels that are used to present information about the aircraft in the operative air traffic control system.

A demonstration of the difference in prediction capability of AI-based prediction of the type that was investigated in the project and a simulation-based prediction approach is available at <https://geistt.com/projects/aibaland>. The password is available through contact with martin.castor@geistt.com, martin.insulander@geistt.com or anette.nas@swedavia.se.

Note that the solution that the project investigated not requires any larger infrastructure investments. Rather, it uses publicly available data and can see how traffic situations generally play out and make predictions before air traffic control officers have started to manage any traffic situation. This can be a cost-effective way to complement other ways to predict landing times and landing sequences. An example of another way to do the predictions is to download information from the aircrafts flight management system. However, the flight management system only considers the individuals aircrafts planning before air traffic controllers have started to make traffic clearances and it also requires new technical systems on many aircraft. The report concludes with a number of recommendations for future research and development activities.

3. Projektets deltagare

Martin Castor, Martin Insulander & Jonathan Borgvall, GEISTT AB (Group for Effectiveness, Interaction, Simulation, Technology, and Training), Stockholm.

Frågor kring rapportens innehåll ställs lämpligen till Martin Castor, GEISTT AB, martin.castor@geistt.com och till projektansvarig vid Swedavia, Anette Näs, anette.nas@swedavia.se.

4. Introduktion

Denna rapport är slutrapport för projektet AI-baserad landningsprediktion inom IRIS-programmet.

4.1. Läsanvisning

I avsnitt 4.2 Projektets bakgrund ges en beskrivning av projektets bakgrund samt mål och syfte.

I avsnitt 4.3 Teoribakgrund ges en mycket kortfattad introduktion till det stora och diversifierade AI-området (Artificiell Intelligens). Stycket kan hoppas över om läsaren upplever sig bekant med AI-området eller bara bryr sig om hur väl projektet lyckades prediktera landningssekvensen.

I avsnitt 5 Metod beskrivs framtagningen av de mönsterigenkännande artificiella neurala nätverken (ANN) som projektet använde.

I avsnitt 6 Resultat beskrivs hur väl de använda ANN lyckades prediktera landningssekvensen. I avsnitt 6.3. beskrivs den konceptdemonstrator som projektet tog fram för att visa hur prediktionen av landningssekvensen skulle kunna användas som ett beslutsstöd för flygledare. I avsnitt 6.4. Användartest beskrivs några observationer från det användartest som genomfördes.

I avsnitt 7 Diskussion diskuteras erfarenheter från projektet.

I avsnitt 8 Rekommendationer beskrivs kortfattat rekommendationerna för fortsatt verksamhet.

4.2. Projektets bakgrund

Rapporten är slutrapporten från projektet AI-baserad landningsprediktion med Swedavia projektnummer 710321.30.06. Projektet finansierades av Trafikverkets Forsknings och Innovationsportfölj för Luftfartsområdet och genomfördes inom ramen för IRIS-programmet (Icke-raka inflygningar till Stockholm Arlanda Airport) mellan juni 2020 och december 2021.

Projektets verksamhet kopplar mot målområdena ”Tillgängliga och effektiva flygplatser i ett hållbart samhälle” och ”Tillgängliga och effektiva flygvägar i ett hållbart samhälle” i Trafikverkets Forskning och Innovations plan 2021-2026 (Trafikverket, 2021), och projektets resultat är riktade mot uppfyllandet av Arlandas nuvarande miljötillstånd.

Projektets mål var att ta fram en demonstrator för en landningsrelaterad beslutsstödskomponent för IRIS-programrelaterade behov. IRIS-programmets syfte är att höja frekvensen med vilken kurvade RNP AR-inflygningar till Arlanda används. Genom ökad användning av kurvade inflygningar, även vid hög trafikintensitet, finns avsevärda miljö- och kapacitetsvinster.

En central del av arbetet för de flygledare som leder flygplanen under inflygningen är att sekvensera dem, dvs. bestämma turordningen för landningen och sedan ge lämpliga klareringar, dvs. olika typer av styrningar och klartecken, in mot landningen. Målet med demonstratorutvecklingen var att undersöka om och hur det går att stödja sekvenseringsprocessen med hjälp av algoritmer från AI-området (Artificiell Intelligens) och deras förmåga till mönsterigenkänning. Prediktioner av denna typ kan då potentiellt bli en komponent i ett beslutsstöd för flygledare.

Notera att varje flygplan i sin färdplan (*flight plan*) har en planerad landningstid redan vid start och den uppdateras kontinuerligt av flygplanets FMS¹ (Flight Management System) för det egna flygplanet. Tanken i detta projekt är att kontinuerligt och dynamiskt bedöma den förväntade landningstiden utifrån helheten i den aktuella situationen, sökandes efter mönster gällande alla relevanta flygplan som identifierats i historiska data. Inom Stockholm TMA² (Terminal Monitoring Area) så sker mycket av ledningen från flygledare genom så kallad radarvektorer och de tider som finns i den färdplanen kan därför förändras, särskilt vid hög belastning på flygplatsen. Ett verktyg som kan prediktera

¹ FMS, <https://skybrary.aero/articles/flight-management-system>

² TMA, https://www.icao.int/MID/Documents/2014/PBN%20Workshop-Tunis/13%20%20EUR%20PBN%20Airspace%20Workshop_Designing%20Volumes%20+%20Sectors-vJUL2013%20.pdf

landningstiden, och därigenom landningssekvensen, utifrån det aktuella luftläget kan därför bli ett användbart verktyg för flygledarna, eller åtminstone ge dem en automatiserad bedömning att jämföra sina beslut med.

En flygledare ansvarig för inflygningarna vid ATCC³ Arlanda hanterar uppskattningsvis 80 landningar per dag, vilket resulterar i ca 17 000 landningar per år sittandes i position. Genom detta arbete lär flygledaren sig mer eller mindre implicit vad som är viktigt för att bedöma en situation och hur antalet landningar kan optimeras. Flygledaren kan oftast snabbt uppfatta om något avviker från det normala mönstret, även om det ibland kan vara svårt att verbalisera. Flygledaren tar för sina beslut hänsyn till en multidimensionell rymd av faktorer, där olika faktorer uppdateras med varierande frekvens. En delmängd av dessa beslut och underliggande mönster bör även ett ANN kunna identifiera, och på så sätt användas för att stödja flygledarens arbete och fungera som ett extra säkerhetsnät⁴. Alla verktyg som underlättar flygledarnas arbete och som ger dem större tidsmarginaler för beslut ökar chanserna att de är komfortabla med att välja kurvade inflygningar.

System och tillämpningar baserade på Artificiell Intelligens (AI) och maskininlärning (ML) har haft stora genombrott de senaste sju åren. AI-området är stort och diversifierat och inrymmer en mängd olika ambitioner, ansatser, och algoritmväl. Genombrotten inom AI de senaste åren bygger på den teoribildning som återfinns i hänvisad litteratur i avsnitt 4.3, men systemen har tack vare ökad beräkningskapacitet, algoritmutveckling, och tillgänglighet till träningsdata, lett till resultat som fått stor uppmärksamhet, både för forskningsområdet och för praktiska tillämpningar.

Allra störst uppmärksamhet har lösningar baserade på så kallade artificiella neurala nätverk (ANN) fått. ANN har goda mönsterigenkännande egenskaper och behöver inte initialt känna till explicita regler för vilka mönster de ska känna igen. Istället kan nätverket, givet att det finns mönster att hitta, utifrån en tillräcklig mängd träningsdata, anpassa de interna vikterna mellan alla noderna i nätverket så att systemet därefter kan användas för att klassificera eller prediktera värden för ”ny data”, baserat på hur mönster i tidigare träningsdata sett ut. En väl beprövad variant av denna AI-ansats, och den som användes i detta projekt, är mönsterigenkänning med ANN tränade genom så kallad övervakad inlärning, dvs. för varje stimuli ett ANN exponeras för under träningsfasen så får det också information om vad som är rätt svar för detta stimuli, se vidare i avsnitt 4.3.

Grundidén i detta projekt var därför att pröva hur användbara prediktioner, baserade på ANN mönsterigenkänningsförmåga, som går att fram. Grundidén presenteras visuellt i Figur 1. Prediktionerna skulle också vara användbara som indata till beslutsstöd, och relatera till IRIS-programmets mål. Genom att använda öppet publicerad ADS-B⁵ data från exempelvis OpenSky Network⁶, Flightradar24⁷ eller Planefinder⁸, kan en databas med träningsdata byggas upp för en större mängd landningar som sedan kan användas för att träna ett ANN.

³ Air Traffic Control Centre

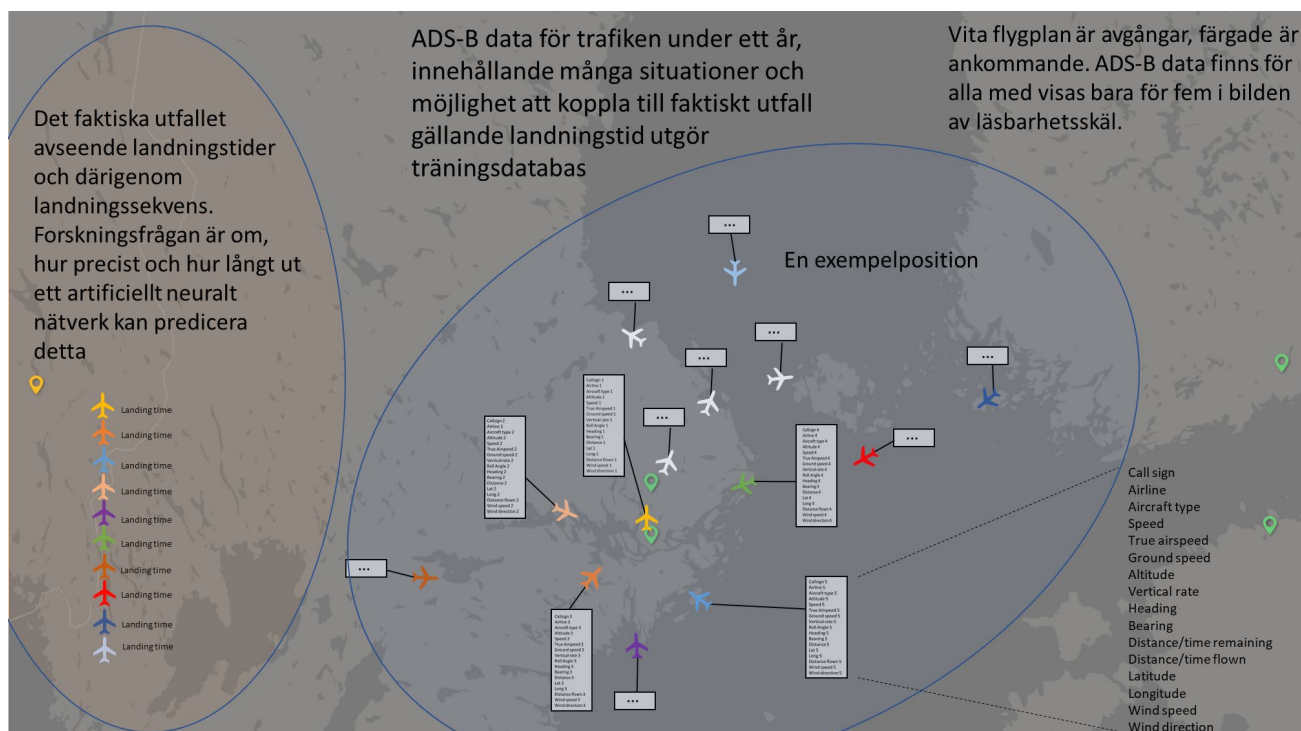
⁴ Safety nets på Skybrary, https://www.skybrary.aero/index.php/Safety_Nets

⁵ ADS-B, Automatic Dependence Surveillance Broadcast, [https://www.skybrary.aero/index.php/Automatic_Dependent_Surveillance_Broadcast_\(ADS-B\)](https://www.skybrary.aero/index.php/Automatic_Dependent_Surveillance_Broadcast_(ADS-B))

⁶ Open Sky Network, <https://opensky-network.org/>

⁷ Flightradar24, <https://www.flightradar24.com/>

⁸ Planefinder, <https://planefinder.net/>



Figur 1. Visuellt beskrivning av projektets grundidé.

Projektets mål var att kunna göra "tillräckligt bra" prediktioner om sekvensen från ett godtyckligt tidsfruset ögonblick, "a snapshot in time", utan kunskap om vad som hänt tidigare eller någon kunskap rörande flygplanets färdplan, *flight plan*. SESAR⁹ genomför nu försök med EPP¹⁰, Extended Projected Profile, genom vilket den färdplan och prediktion som finns i flygplanets FMS kan länkas ner till marken. Summan av all EPP information bör ge en mycket bra prediktion om allt flyter på smidigt och kommer in i TMA med redan tillräcklig separation, men när det finns flera flygplan behöver flygledare oftast vidta åtgärder för att separera dem och denna information finns inte i FMS innan den skett. Det neurala nätverket kan dock genom att det är tränat att se mönster bland de historiska situationerna eventuellt se hur trafiken brukar hanteras.

Observera att rapportförfattarna bedömer att flygledare kommer att behövas för mycket lång tid framöver för att se farliga och avvikande situationer samt fatta de kritiska besluten. Ett ANN:s mönsterigenkänningsförmåga baserar sig på historiska data i den datamängd det tränats på. Om riskabla mönster och avvikelser uppstår, som inte finns i träningsdata, så kommer ett ANN ha begränsad eller ingen förmåga att upptäcka dem. En mänsklig flygledare har också tillgång till mer data, exempelvis gällande avvikande förutsättningar, som inte nödvändigtvis finns som information i de tekniska systemen.

En central poäng är att valet av träningsdata är avgörande, och om det finns historiska orsaker till att fördelningar i data ser ut som de gör så kommer detta finnas kvar i den klassificeringsalgoritm som systemet lär sig och använder framöver. Om ett mönsterigenkänningssystem av den typ som använts i projektet tränas på en datamängd som innehåller en viss typ av situationer och avvikelser så är det mönster i denna typ av situationer som systemet kommer att känna igen. För ett skarpt framtida system är det därför centralt att förstå vem som har valt ut den datamängd systemet tränats med, vilka kriterier som användes för att välja data, ev. bortfallsanalyser om viss typ av data utelämnats, och styra upp en process som gör att systemet över tid tränas på nya data.

Grundregler från statistiken, som exempelvis att korrelation inte nödvändigtvis implicerar kausalitet får inte glömmas bort, men ett neuralt nätverk har inte förmågan att förstå detta. Nätverket kan också klassificera på andra särdrag än de som utvecklarna sett framför sig.

⁹ SESAR JU, <https://www.sesarju.eu/>

¹⁰ EPP, Extended Projected Profile, <https://www.sesarju.eu/sesar-solutions/extended-projected-profile-epp-availability-ground>

4.3. Teoribakgrund

Artificiell intelligens är ett i nuläget mycket snabbt expanderande område, både som forskningsområde och som tillämpad teknologi i allt fler produkter. Som forskningsområde är det dock inte nytt utan brukar sägas ha sin start 1956 vid en nu berömd forskningskonferens på Dartmouth College i New Hampshire, vid vilken uttrycket sägs ha myntats första gången.

Det är för rapportens syfte inte meningsfullt att reproducera tidigare historiska översikter gällande AI-områdets framväxt. För den mer detaljerade historien bakom AI-områdets framväxt hänvisas till referensverken *Artificial Intelligence – A Modern Approach* (Russel & Norvig, 2010)¹¹ och *Deep Learning* (Goodfellow, Bengio, & Courville, 2016)¹². Båda dessa böcker används i hög utsträckning som grundlitteratur vid AI-kurser världen över.

AI bör ses som ett samlingsnamn för en större familj av delvis likartade, men också delvis relativt olika tekniker för att utveckla datorprogram med problemlösningsförmåga. Som Russel och Norvig (2010) påpekar skulle troligen begreppet beräkningsbar rationalitet (*computational rationality*) varit ett lämpligare begrepp för att förstå innebörden. Det handlar alltså om att utveckla datorprogram som kan komma fram till slutsatser som uppfattas som rationella, ofta genom att kunna urskilja mönster i någon datamängd och sedan agera på ett relevant sätt utifrån detta.

Någon etablerad och ensad definition av vad AI är finns inte idag men det finns naturligtvis en uppsjö förslag. EU High level expert group on AI (EU HLEG AI, 2018), som samlat ett större antal AI-expert, publicerade nyligen denna definition:

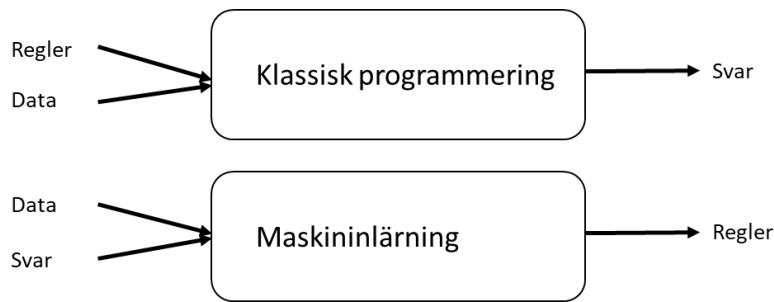
Artificial intelligence (AI) refers to systems designed by humans that, given a complex goal, act in the physical or digital world by perceiving their environment, interpreting the collected structured or unstructured data, reasoning on the knowledge derived from this data and deciding the best action(s) to take (according to pre-defined parameters) to achieve the given goal. AI systems can also be designed to learn to adapt their behaviour by analysing how the environment is affected by their previous actions. As a scientific discipline, AI includes several approaches and techniques, such as machine learning (of which deep learning and reinforcement learning are specific examples), machine reasoning (which includes planning, scheduling, knowledge representation and reasoning, search, and optimization), and robotics (which includes control, perception, sensors and actuators, as well as the integration of all other techniques into cyber-physical systems). (EU HLEG AI, 2018).

AI bör alltså betraktas som ett paraplybegrepp som spänner över diverse beräkningsansatser som används i många olika tillämpningar. AI med delområdet maskininlärning är vetenskapliga områden som bör betraktas som delar av datavetenskapen, även om det finns starka relationer till flera andra vetenskapliga områden som exempelvis statistisk. AI kan i många fall sägas vara tillämpad statistik. AI är alltså en samling koncept, problem/uppgifter man vill lösa, och metoder för att lösa problemen/uppgifterna. Olika AI-ansatser kan i princip vara applicerbara på "vad som helst", givet att det innehåller ett passade beräkningsproblem. För problem som klustring, prediktion, regression, klassificering, rankning, summering, anomali-detektion, rekommendationer, och beslutsfattande finns många olika algoritmer från AI-området som är användbara.

Ett viktigt begrepp i sammanhanget är om systemet har förmåga till maskininlärning (*machine learning*). Med maskininlärning avses att beteenden eller förmågan att exempelvis klassificera med hjälp av algoritmerna som används inom aktuell AI-ansats kan utvecklas över tiden, utan att utvecklaren explicit behöver beskriva hur de ska förändras. Begreppet maskininlärning användes redan 1959 av Artur Samuel som var en tidig pionjär. Han beskrev det som ett forskningsområde som ger datorer förmågan att lära sig utan att explicit programmeras. Maskininlärning representerar således ett annat paradigm än den klassiska programmeringen, se exempelvis Chollets (2017) beskrivning i Figur 2. I klassisk programmering så skriver utvecklaren kod som beskriver de regler som behövs för att hantera indata för att få önskat svar som utdata. I maskininlärning (åtminstone inom övervakad inlärning) så får systemet data och svar, och systemet identifierar utifrån detta de regler som behövs. Dessa regler kan sedan appliceras på nya indata för att hantera en ny datamängd och ge korrekta svar. En maskininlärningstillämpning programmeras alltså inte explicit avseende sitt beteende, utan tränas genom exponering av träningsdata.

¹¹ <http://aima.cs.berkeley.edu>

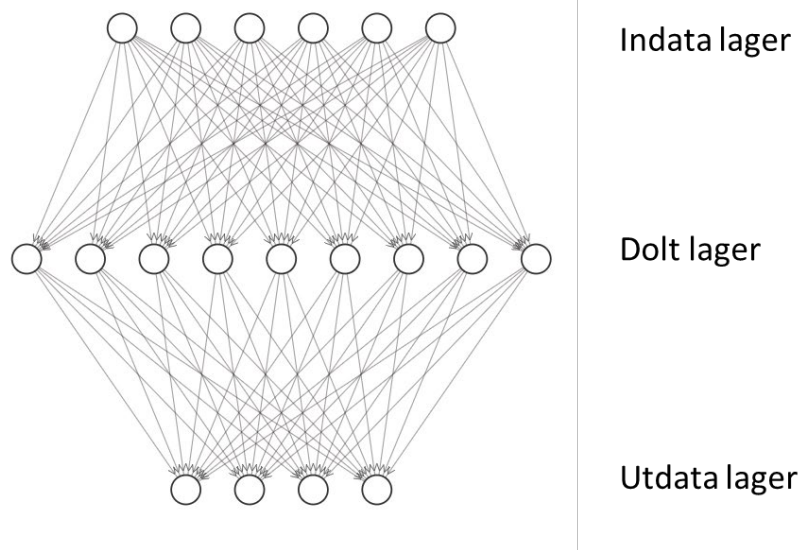
¹² <http://www.deeplearningbook.org>



Figur 2. Skillnaden mellan klassisk programmering och maskininlärning enligt Chollet (2017).

Nuförtiden avser maskininlärning ofta användningen av någon form av artificiella neurala nätverk (ANN) som tränas på stora datamängder, där nätverkets vikter gradvis anpassas under träningen för att optimera systemets prestation, exempelvis för en klassificeringsuppgift. Artificiella neurala nätverk är framför allt mönsterigenkännare som kvantifierar relationerna mellan olika särdrag i den datamängd som nätverket tränas på. Förklaringen av hur systemet klassificerar den indata systemet hanterar är ofta mycket svårt att tillgodogöra sig för en människa. Ett artificiellt neuralt nätverk av idag kan innehålla miljontals parametrar som uppdateras, och efter träning kanske resulterar i en lösning som fungerar för den aktuella klassificeringsuppgiften, men som inte nödvändigtvis är den enklaste eller mest effektiva lösningen.

Ett ANN består i sin enklaste form av tre lager, ett indata-lager som tar emot data, ett dolt lager och ett utdata-lager som ger resultaten, se Figur 3. Varje nod i indatalagret söker efter ett visst särdrag (*feature*). Om denna nod registrerar sitt specifika särdrag i datamängden så aktiveras noden och skickar sin aktivering vidare till alla noder den är kopplad mot.

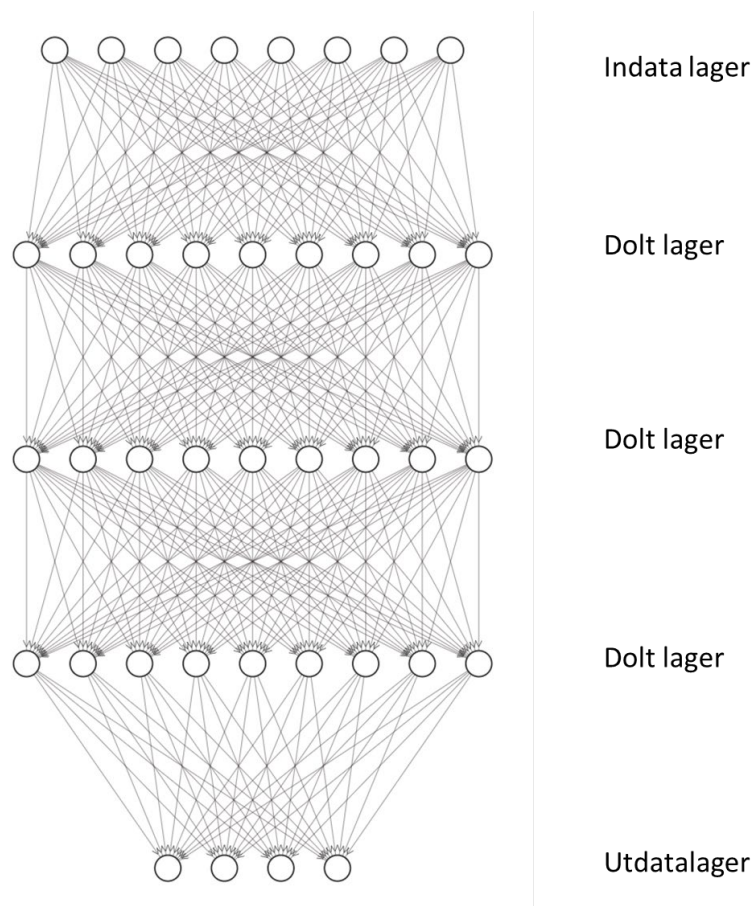


Figur 3. Schematisk beskrivning av klassiskt neuralt nätverk (feed-forward).

Några grundläggande parametrar som kan variera för ett ANN är:

- Antal noder i inlagret
- Antal noder i utlagret
- Antal dolda lager och antalet noder i dem
- Aktiveringsvikter och aktiveringströskeln mellan länkade noder i nätverket
- Topologin/nätverksstrukturen, dvs. vilka noder som länkar till vilka
- Om noder kan återmata information till sig själv, s.k. *recurrent networks*
- Om information rörande felklassificering återmatas till nätverket, s.k. *backpropagation*

Figur 4 beskriver ett ANN med fler lager och de senaste årens genombrott har ofta relaterat till så kallad djupinlärning (*deep learning*) där en viktig komponent är att nätverket har avsevärt många fler lager än i figuren.



Figur 4. Schematisk beskrivning av ett klassiskt *feed-forward* neuralt nätverk med fler lager. Observera att alla noder enbart kopplar framåt (dvs. åt nedåt i figuren) och kopplar till alla noder i nästa lager.

Komplexiteten, de relativt abstrakta koncepten, och terminologin rörande ANN gör det snabbt överskådligt för den oinvidde. Förståelse för hur lager, vikter, uppdatering av vikter och motsvarande fungerar förklaras mycket pedagogiskt och animerat på 3Blue1Brown¹³ vilken rekommenderas läsaren för att snabbt förstå grunderna. Filmen förklarar den mest klassiska versionen av *feed-forward* nätverk, men förståelse för deras uppbyggnad och funktion utgör basen för att förstå grunderna.

En central skiljelinje mellan olika typer av ANN-ansatser är hur nätverken tränas, dvs. hur vikterna mellan olika noder i nätverket uppdateras efter att nätverket exponerats för indata. En ofta använd uppdelning är att skilja mellan följande tre huvudtyper av träning:

- Övervakad inlärning (*supervised learning*)
- Oövervakad inlärning (*unsupervised learning*)
- Förstärkningsinlärning (*reinforcement learning*)

Dessa olika huvudtyper av träning leder till relativt olika tillvägagångssätt vid modelleringen. I detta projekt användes övervakad inlärning är den historiskt sett äldsta och bedömt fortfarande den mest vanligt använda träningsformen. Nätverket lär sig här rätt svar för varje stimuli från någon annan, oftast ett av människor annoterat rätt svar. För varje stimuli, exempelvis en bild, har alltså en människa angett vad som finns på bilden. Ett klassiskt exempel visas i Figur 5. I MNIST databasen ingår 70 000 handskrivna siffror med rätt tolkning angiven, exempelvis så visar den översta vänstra cellen en handskriven åtta och rätt svar är angivet till en åtta.

¹³ <https://www.youtube.com/watch?v=aircArvvnKk>



Figur 5. Utdrag ur MNIST databasen.

Ett exempel på övervakad inlärning hos människor kan vara ett barn som ska lära sig läsa. Oftast sker det genom att en vuxen person pekar på en bokstav och uttalar den. Barnet lär sig då med hjälp av att koppla den visuella bilden av bokstaven till ett ljud. På samma sätt behöver algoritmen tränas upp, under träningen ger man den data som är kopplad till en etikett (*label*) som avspeglar korrekt eller önskat värde för just den datapunkten.

För detta projekt är indata de särdrag (*features*) som beskrivs i avsnitt 5.1 och det korrekta svaret (*target*) är den faktiska landningstiden.

Processen kring maskininlärning för ANN med övervakad inlärning följer typiskt följande punkter:

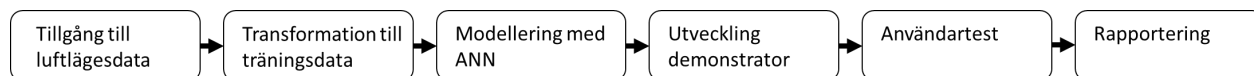
- Samla in data
- Städa och organisera datamängden
- Förstå datamängden
- Bygg en modell (dvs. en nätverksarkitektur)
- Låt modellen tränas på data
- Justera modellen
- Validera modellen på en tidigare oanvänd delmängd av data
- Använd det tränade nätverket/modellen för den tilltänka uppgiften

Att samla in, städa, och förstå datamängden kan ofta uppta 70% av tiden i ett projekt och denna aktivitet behöver ofta stöd av domänexperter som kan beskriva vad olika variabler representerar och hur/varför det som verkar vara avvikelser i data kan vara enkelt förklarat om man förstår domänen.

När en träningsdatabas finns tillgänglig behöver strukturen för det artificiella neurala nätverket bestämmas och detta kallas modellering. För ett typiskt ANN-projekt är detta en relativt explorativ verksamhet. Det finns tumregler för hur nätverken lämpligen utformas, men exakt vilken modell som lyckas prediktera bäst i den specifika tillämpningen måste oftast prövas fram.

5. Metod

Figur 6 beskriver de faser projektet gick igenom gällande hanteringen av träningsdata och modellering.



Figur 6. Projektets faser.

5.1. Data

Projektets första åtgärd var att söka efter data på luftläget kring Arlanda flygplats som skulle kunna vara användbar som träningsdata till artificiella neurala nätverk. En stor mängd data behövs. Denna typ av data finns tillgänglig via öppna ADS-B baserade tjänster som Flightradar24, OpenSky och Planefinder. Dessutom registrerar LFV denna typ av data i det så kallade GAIA-systemet. Givet Covid 19 situationen och bristen på representativ flygtrafik under projektet var data från GAIA-systemet det enklaste sättet att få tillgång till grunddata. GAIA-systemets data är dessutom mycket strukturerad och av god kvalitet. Efter säkerhetsgenomgång och utfiltrering av militär trafik fick projektet tillgång till 18 månaders trafikdata, med fem sekunders upplösning, för luftrummet upp till 100 nautiska mil från Arlanda med både starter och landningar till och från flygplatserna ESSA (Stockholm Arlanda) och ESSB (Stockholm Bromma) inkluderade.

Tabell 1 visar en jämförelse mellan data i Flightradar24 och GAIA-systemet. Båda datakällorna ger också tillgång till mer data än vad som presenteras i tabellen.

Tabell 1. Jämförelse av datatillgänglighet Flightradar24 och GAIA systemet.

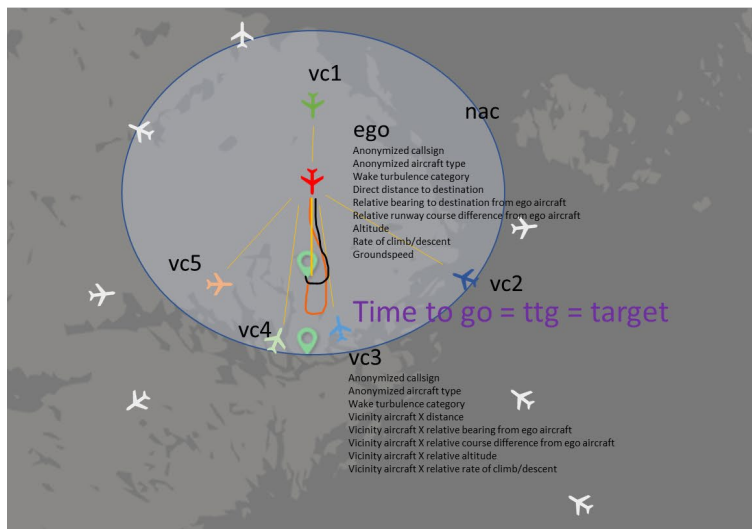
ADS-B data från Flightradar24	GAIA data och särdragsnamn
Call sign	flt_callsign = flight callsign
Airline	
Aircraft type	flt_actype = flight aircrafttype
Wake category type	flt_wct = flight wake category
Speed	
True airspeed	
Ground speed	gs = ground speed
Altitude	alt = altitude
Vertical rate	rocd = rate of climb/descent
Heading	
Bearing	
Distance/time remaining	dirdist_des = direct distance destination
Distance/time flown	dirdist_dep = direct distance departure

ADS-B data från Flightradar24	GAIA data och särdragsnamn
Latitude	lat = latitude
Longitude	lon = longitude
Wind speed	
Wind direction	

Utifrån den grunddata som fanns i GAIA-data gjorde projektet ett antal beräkningar och transformationer för att skapa projektets träningsdatabas som beskrivs i Tabell 2 och Figur 7 nedan. Notera att det alltså inte är GAIA-data rakt av som användes i den modellering som projektet genomförde. Varje rad i GAIA-databasen behöver transformeras till en helhetssituation för alla flygplan för varje tidssteg. Jämför med Figur 7 som är en visualisering av de särdrag som projektet tog fram. För varje tidssteg, som här valdes till 20 sekunder, beräknas särdragen för varje rad i GAIA-databasen (som representerar ett flygplan på en viss punkt), och resultatet blir den träningsdatabas som användes i projektet. Detta gjordes för att transformera helheten i luftlägen till data som kan uttryckas i tabellform.

Tabell 2. Transformerad träningsdata

Ego flygplansrelaterade särdrag	Kommentar
Anonymized aircraft type, anon_actype	
Wake turbulence category, wtc	
Direct distance to destination, dirdest_des	
Relative bearing to destination from ego aircraft, rel_bearing_des_n	
Relative runway course difference from ego aircraft, rel_rwy_course_abs_n	
Altitude, alt	
Rate of climb/descent, rocd	
Groundspeed, gs	
Globala TMA/områdesrelaterade särdrag	
Number of aircraft within 50 nautical miles from ego aircraft, nac	
Flygplanspars närhetsrelaterade särdrag (för de fem flygplanen närmast ego flygplanet, dvs. fem uppsättningar av dessa särdrag VC1-VC5, kallas Vicinity särdrag i fortsatt text)	
Anonymized callsign, vcX_anon_callsign	Användes inte under modelleringen
Anonymized aircraft type, vcX_anon_actype	
Wake turbulence category, vcX_wtc_n	
Vicinity aircraft x distance, vcX_dist_n	
Vicinity aircraft x relative bearing from ego aircraft, vcX_rel_bearing_n	
Vicinity aircraft x relative course difference from ego aircraft, vcX_rel_course_abs_n	
Vicinity aircraft x relative altitude, vcX_rel_alt_n	
Vicinity aircraft x relative rate of climb/descent, vcX_rel_rocd_n	
Kriterievariabel, facit för landningstid	
Time to go = ttg	Target särdrag, beräknades i efterhand. Används för träningen, men används inte för prediktionen när nätverket väl tränats



Figur 7. En visualisering av särdragen i träningsdatabasen.

Totalt ingår alltså 51 särdrag (9 ego flygplansrelaterade, 1 områdesrelaterad (nac) och 8*5 Vicinity-relaterade särdrag samt en kriterie/facit-variabel på vad landningstiden blev (dvs. target särdraget) i träningsdatabasen. I Figur 7 visas Direct Distance to Destination, dvs. fågelvägen till flygplatsen, med ett gult streck framför det röda ego-flygplanet. Det orangea strecket visar hur en klassisk ILS-inflygning skulle kunna se ut, medan det svarta strecket grovt visar en inflygning enligt RNP Z-proceduren. De callsign relaterade särdragen användes inte under projektets modellering, men skulle kunna vara användbara för andra typer av analyser, baserade på delmängder av databasen, exempelvis prediktioner för ett visst flygbolag. Time to go som är det så kallade target-särdraget vid träningen anges i lila och visas inte i figuren men är en i efterhand framräknad tid kvar till landning för varje rad i träningsdatabasen.

Projektet tog fram några varianter av träningsdatabaser, primärt en databas med ca 550 000 rader i databasen, där varje rad beskriver luftläget utifrån ett flygplans perspektiv med de variabler som anges i Figur 7, med tjugo sekunders uppdatering och för tre månaders trafik. En enskild flygning finns då representerad i databasen ca 200 gånger.

I en annan träningsdatabas som togs fram i projektet gjordes en sortering så att varje enskild flygning bara förekom några få gånger i databasen och filtrerat för landningar som gick till landningsbana 01R. Denna träningsdatabas innehåller 38793 rader med 15 minuters intervall i datapunkterna, baserad på all trafik under 2019. Det är resultaten för denna databas som redovisas i rapporten, men prediktionsförmågan för den större databasen är relativt likvärdig.

5.2. Förståelse för data och dess spridning

Att förstå ett tränat ANN:s klassificering är mycket svårt för en människa. Nätverket tar hänsyn till variansen i alla tillgängliga variabler och försöker matcha dem mot rätt svar (*target*). Svaret till varför nätverket klassificerar som det gör står att finna i alla interna vikter mellan nätverkets noder, vilka är väldigt många och därför oöverblickbara. För att ge viss inblick i hur data ser ut så återges i detta avsnitt några figurer som beskriver datamängden i träningsdatabasen.

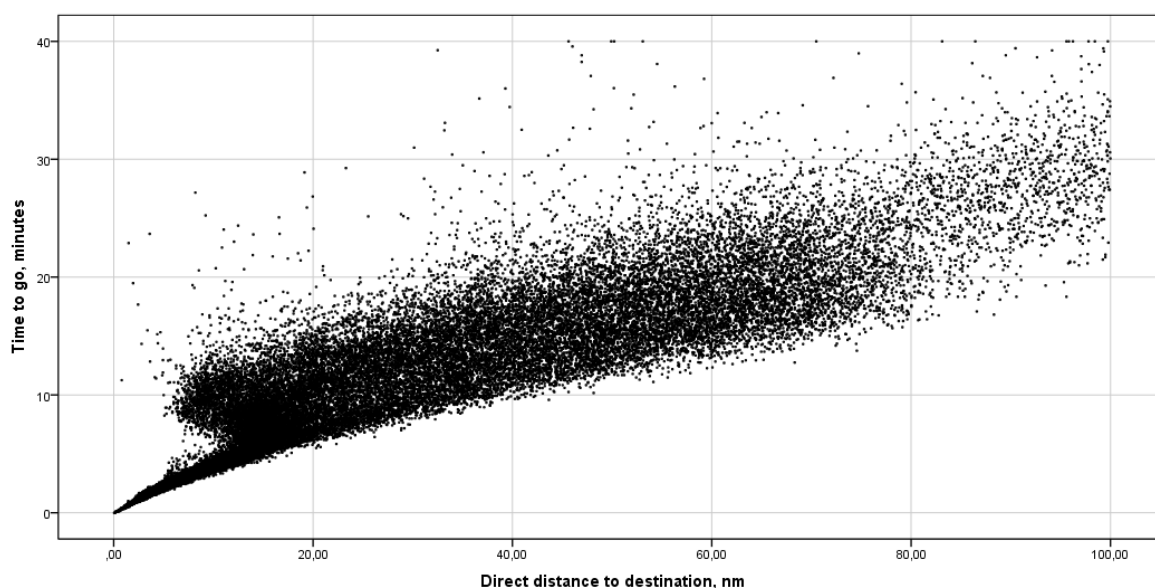
ego

- Anonymized callsign
- Anonymized aircraft type
- Wake turbulence category
- Direct distance to destination
- Relative bearing to destination from ego aircraft
- Relative runway course difference from ego aircraft
- Altitude
- Rate of climb/descent
- Groundspeed

- Number of aircraft

vcx

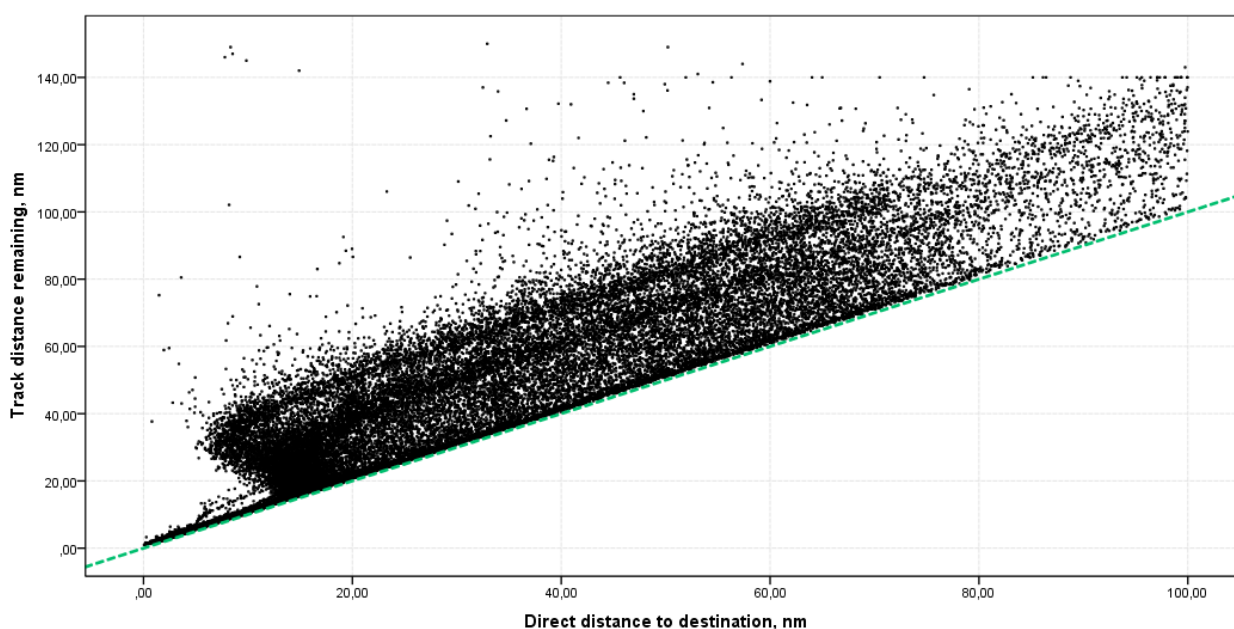
- Anonymized callsign
- Anonymized aircraft type
- Wake turbulence category
- Vicinity aircraft X distance
- Vicinity aircraft X relative bearing from ego aircraft
- Vicinity aircraft X relative course difference from ego aircraft
- Vicinity aircraft X relative altitude
- Vicinity aircraft X relative rate of climb/descent



Figur 8. Spridningsdiagram (scatterplot) mellan "Time to go" och "Direct Distance to Destination".

Korrelationen mellan variablerna i Figur 8 är $r=0,895$. Beroende på vilket fenomen som studeras kan detta anses vara en mycket stark korrelation, men som framgår av figuren så finns det ändå relativt många punkter som avviker från en perfekt korrelation.

I efterhand går det också att beräkna kvarvarande färdväg (track distance) i nautiska mil, dvs. hur långt flygplanet faktiskt flög för varje rad i träningsdatabasen. För att ge ytterligare förståelse för data visas i Figur 9 relationen mellan "Direct distance to Destination" och den i efterhand beräknade "Track Distance to Destination". Direct distance to Destination, som används i ANN-prediktionen, har en hög korrelation med Track Distance Remaining, $r=0,927$, men det finns ändå många punkter, dvs. alla som inte ligger strax över den streckade gröna linjen, där sambandet anger att det inte är kortaste vägen, motsvarande en "direct to"-klarering som gäller.



Figur 9. Spridningsdiagram mellan "Direct distance to Destination" och den i efterhand beräknade "Track Distance to Destination".

5.3. Modelleringsprocess

För att beskriva det rättframt ska modelleringen ses som olika ställningstaganden avseende vilken information nätverket har tillgång till när det gör sin prediktion av landningstiden. Hur bra kan nätverket gissa landningstiden om det vet om värdet på ego-variablerna (dvs. flygplanets kvarvarande avstånd fågelvägen till flygplatsen, fart, höjd, flygplanstyp osv. enligt Figur 7)? Blir gissningen bättre om nätverket också har information om de närmaste fem flygplanen? I vilken utsträckning påverkas förmåga att gissa rätt av nätverkets interna utformning (antal dolda lager, aktiveringsfunktioner osv.)

Under den iterativa modelleringsprocessen i projektet utvärderades ett relativt stort antal alternativa modeller där både olika konfigurationer av ingående särdrag, dvs. särdragen i Figur 7 och olika antal dolda lager användes.

Några huvudsakliga varianter av de modeller som utvärderades för att se skillnaderna i prediktionsförmåga anges nedan i Tabell 3. Alla utvärderade konfigurationer beskrivs inte i rapporten. Tabell över deras prediktionsförmåga redovisas i avsnitt 6.1.

Tabell 3. Alternativa modeller.

Modell	Idé bakom modellen
Modeller med alla särdrag med 1, 2 eller 3 dense layers	Alla särdrag kan eventuellt bidra till modellernas förståelse av situationen och dess prediktionsförmåga. Olika antal lager gör att modellen ev. i högre utsträckning kan använda informationen i särdragen
Modell med enbart ego flygplanets särdrag	Ett försök för att se hur bra prediktionen blir bara utifrån ego-särdragen
Modell med enbart Direct distance to destination (dirdest) och höjd (alt)	Ett försök till en minimal modell med två särdrag som eventuellt skulle räcka för tillräckligt bra prediktion
Modell med ego särdragen och olika konfigurationer av Vicinity-särdragen	Ett försök att reducera från att använda alla särdrag men ändå inkludera Vicinity-särdragen
Modell med särdrag valda utifrån en fristående korrelationsanalys (dirdist, rel rwy course, alt, gs, vc 1 dist)	Ett försök för att se hur hög prediktionsförmågan blir om bara "de bästa" variablerna, utifrån en korrelationsanalys, inkluderas
Modell med särdrag valda utifrån en fristående regressionsanalys (dirdest, rel rwy course, alt, nac)	Ett försök för att se hur hög prediktionsförmågan blir om bara "de bästa" variablerna, utifrån en regressionsanalys, inkluderas

Olika varianter av så kallade aktiveringsfunktioner har också utvärderats i projektet men redovisas inte av läsbarhetskäl, då skillnaderna varit minimala. I alla de resultat som redovisas i Tabell 4 har ReLu-funktionen använts för aktiveringen.

6. Resultat

6.1. Utvärderade modeller

I Tabell 4 presenteras resultaten avseende prediktionsförmåga för ett representativt urval bland de modeller som utvärderats.

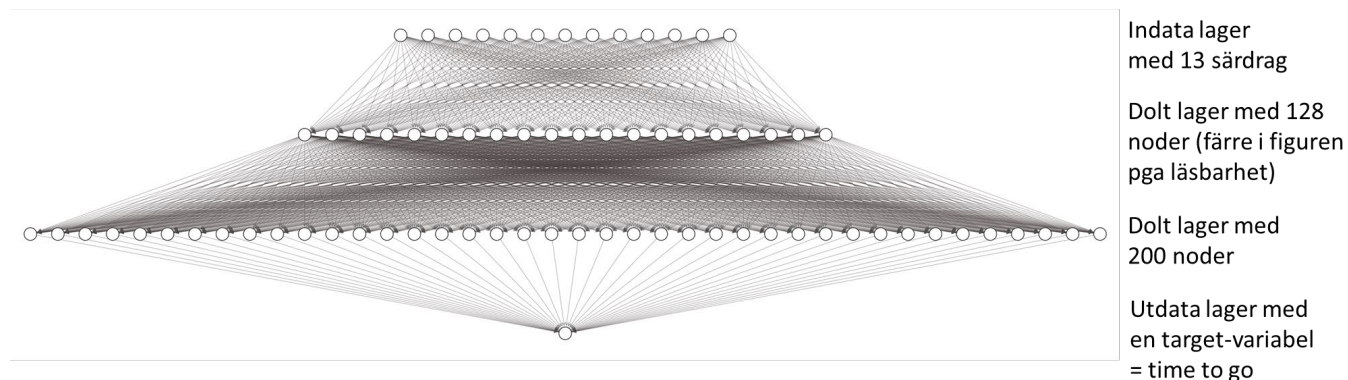
Tabell 4. Prediktionsförmåga hos utvärderade modeller.

Variant av modell	Medelfel i minuter	Antal sekunder sämre än bästa modellen
Modell med alla särdrag 1 dense layer	1,206	16,5
Modell med alla särdrag 2 dense layers	1,237	18,36
Modell med alla särdrag 3 dense layers	1,222	17,46
Modell med enbart ego flygplanets särdrag	0,948	1
Modell med enbart Direct distance to destination (dir_dist) och altitude (alt)	1,502	34,26
Modell med ego särdragen och olika konfigurationer av Vicinity-särdragen (här bara de avståndsrelaterade Vicinity-särdragen)	0,931	-
Modell med särdrag valda utifrån en fristående korrelationsanalys (dir_dist, rel rwy course, alt, gs, vc 1 dist)	1,038	6,42
Modell med särdrag valda utifrån en fristående regressionsanalys (dir_dist, rel rwy course, alt, nac)	1,094	9,78

Vilket framgår av tabellen ovan är den modell som bäst predikterar landningstiden en modell som består av ego-variablerna och den avståndsrelaterade vicinity-variabeln för de fem närmaste flygplanen. Skillnaden mot nästa modell som bara består av ego-variablerna är mycket liten och skillnaden i medelfelet för prediktionen är ungefär en sekund. En minimal modell med bara direkt avstånd till destinationen och höjden predikterar som medelfel cirka 35 sekunder sämre.

6.2. Föreslagen modell

Baserad på jämförelsen avseende medelfel, som här bör tolkas som kvalitén i prediktionsförmågan, som återfinns i Tabell 4 så valdes en modell med alla de ego-relaterade särdragen samt de avståndsrelaterade vicinity-särdragen för de närmaste fem flygplanen. Modellens arkitektur visas i Figur 10 och använder sig två lager av fullt kopplade dolda lager, *fully connected/dense layers*, dvs. alla noder kopplar mot alla i nästa lager.



Figur 10. Föreslagen modells arkitektur (notera att färre noder i dolda lagren visas pga läsbarhet).

6.3. Implementation i demonstrator

I detta avsnitt beskrivs den demonstrator som togs fram för att visa hur prediktionerna skulle kunna användas operativt.

Den prediktion av landningstid och därigenom prediktion av landningssekvens som den framtagna ANN-modellen gör integrerades för detta med simulatormiljön SkyLARC. SkyLARC är en flygledningssimulator speciellt anpassad för konceptutveckling i tidiga faser. Rapportens författare har tidigare deltagit i BKI-projektet (Beslutsstöd Kurvad Inflygning) som tog fram koncept för olika typer av beslutsstöd för flygledare, dock utan att använda maskininlärningsalgoritmer i dessa. I BKI-projektet utvecklades simulatormiljön SkyLARC, se Castor, Insulander, Strid, Hellman, Borgvall och Rappich (2021) för vidare beskrivning.

För implementationen av demonstratorn användes AI-plattformen Peltarions¹⁴ förmåga att ladda ner en fristående ANN-modell, som man definierat med hjälp av Peltarions effektiva visuella gränssnitt, och sedan anropas denna fristående modell för att göra prediktionerna.

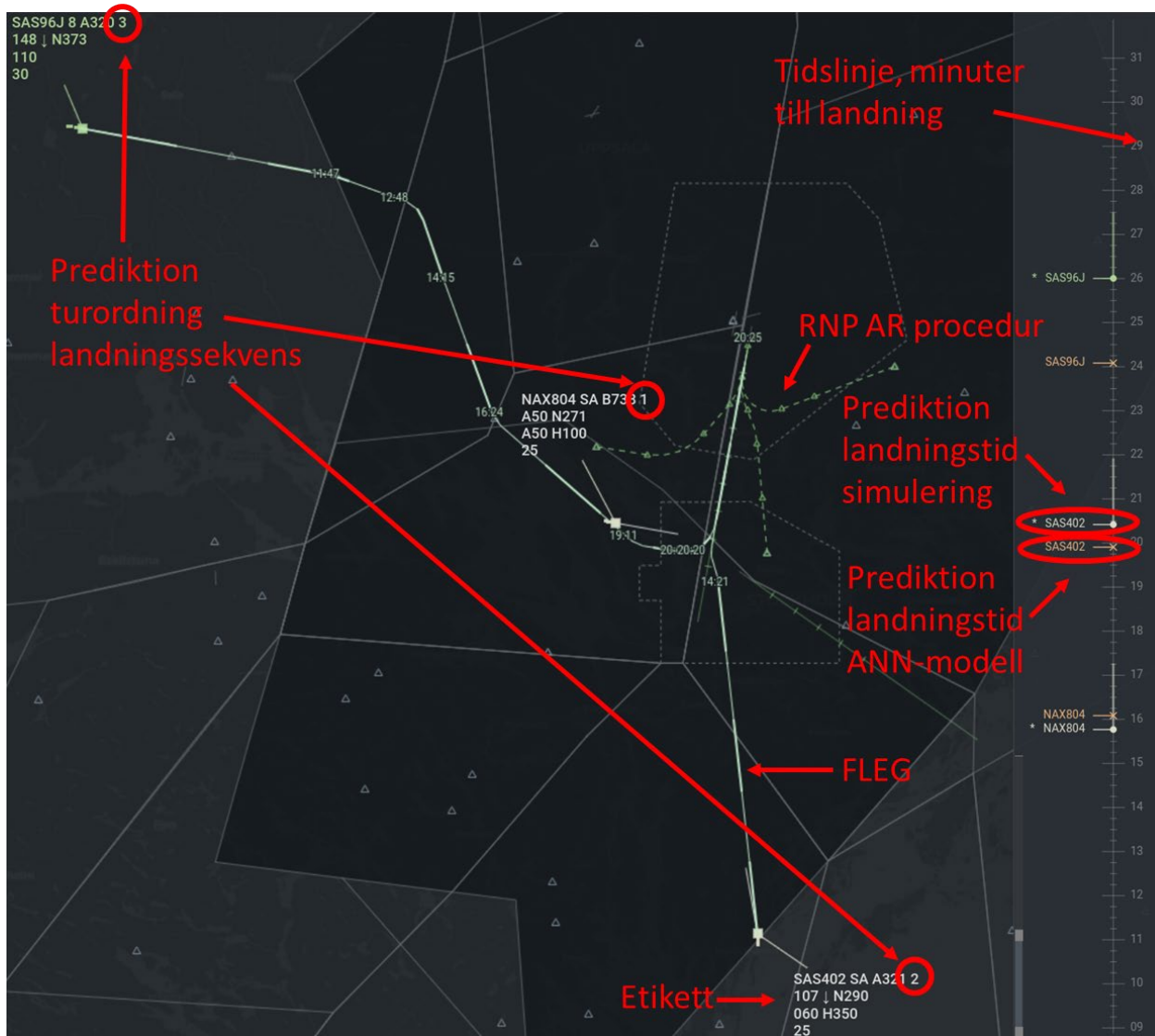
Några exempel på prediktionen som åstadkoms genom ANN-modellen presenteras i Figur 11-Figur 13 nedan. I figurerna visas en enkel trafiksituation med två till tre flygplan, med den s.k. FLEG-presentationen (Flight Leg, som visar planerad färdväg) påkopplad. På tidslinjen (som inte finns i det skarpa TopSky systemet) till höger i figuren visas prediktionen för landningstid med simuleringsbaserad prediktion i grönt eller vitt¹⁵, medan prediktionen avseende landningstid för ANN-modellen anges i beige-rosa.

I etiketterna för respektive flygplan anger den sista siffran i rad 1 av etiketten vilken turordning i landningssekvensen som ANN-modellen predikterar, baserat på den predikterade landningstiden. Exempelvis så anges turordning 3 för SAS96J i Figur 11. För dessa enkla situationer är det relativt lätt att se den troliga sekvensen även utan stöd, men när det är mer än tio flygplan på väg in mot flygplatsen kan denna automatiska sekvensprediktion troligen bli mer användbar.

Figur 11 visar ett exempel när prediktionsförmågan med förklarande texter i rött.

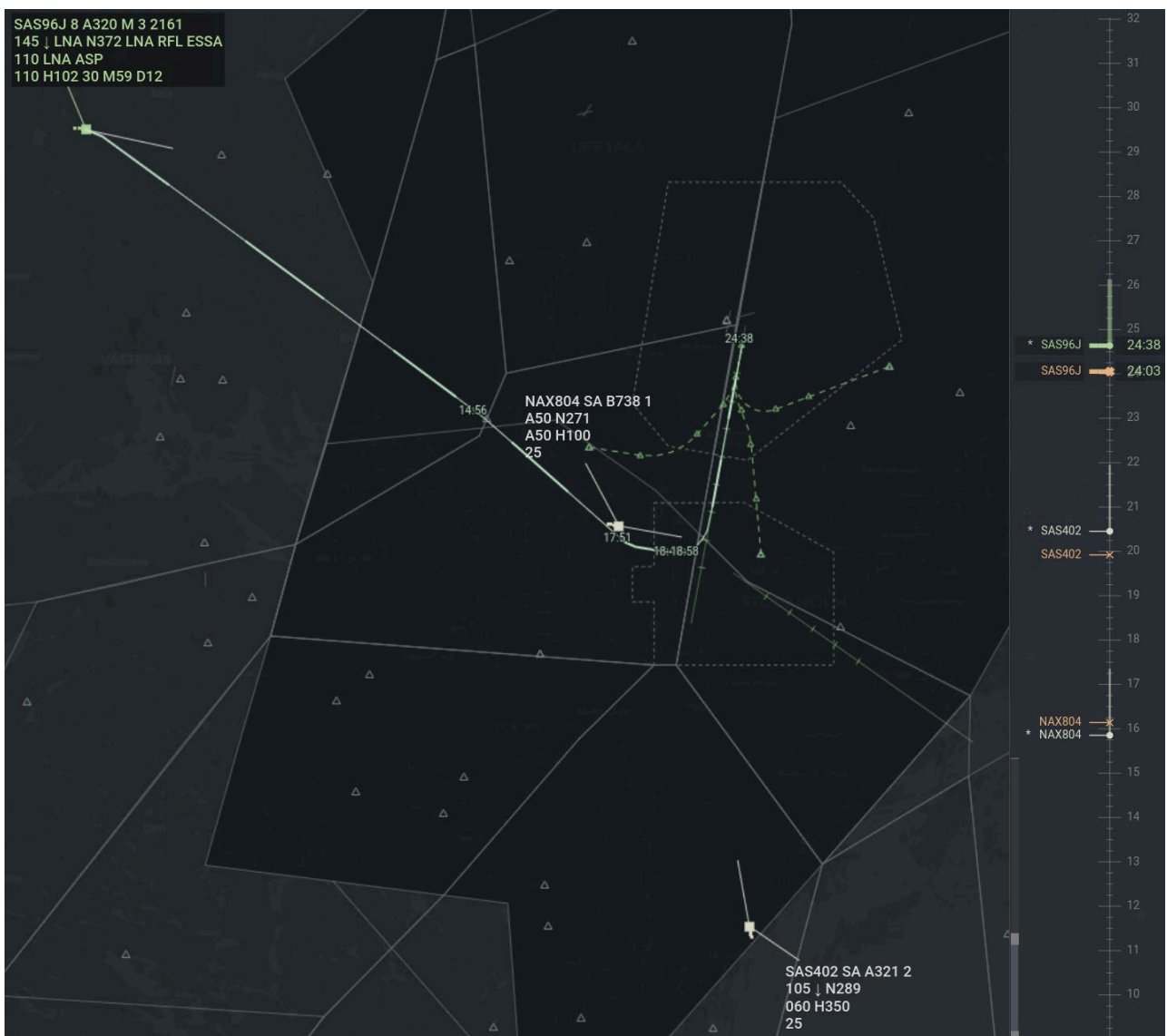
¹⁴ <https://peltarion.com/>

¹⁵ Färgen avgörs av huruvida aktuell flygledare gjort "assume" för detta flygplan, dvs. angett i systemet är kontrollen över flygplanet ligger hos honom eller henne.

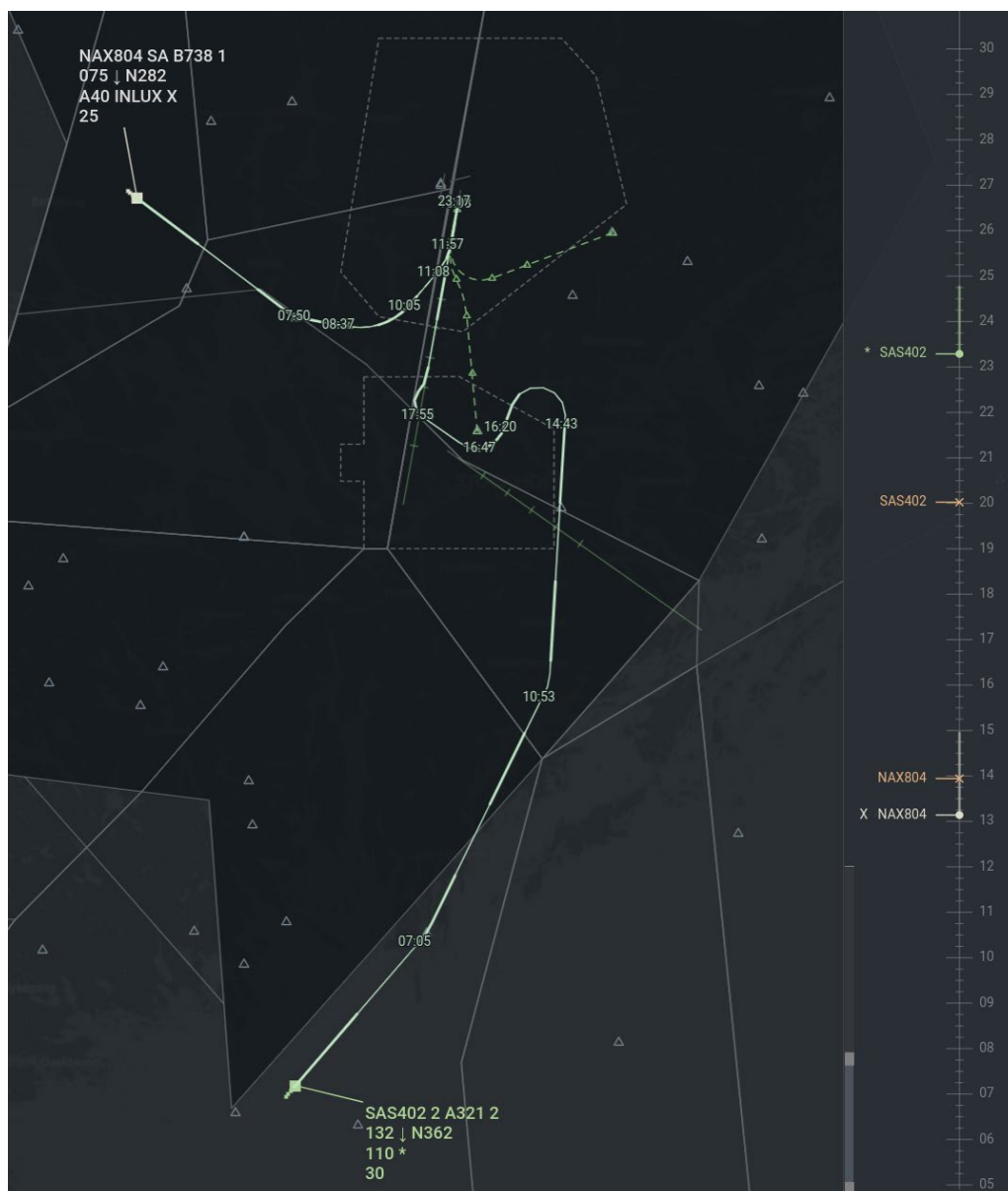


Figur 11. Exempel på relativt precisa prediktioner med förklaringar av viktiga informationselement.

Figur 12 visar ett exempel på en relativt precis prediktion.



Figur 12. Exempel på relativt precis prediktion.



Figur 13. Exempel på mindre precis prediktion med STAR via TEB och ILS-landning.

I Figur 13 är skillnaden för den simuleringsbaserade prediktionen och den ANN-baserade prediktionen större. Den simuleringsbaserade prediktionen räknar på den färdplan som flygplanet följer, inklusive en STAR (Standard Terminal Arrival Route) som går via punkten TEB. ANN-prediktionen baserar sig i stället på hur situationen i praktiken brukar hanteras, dvs. om det inte finns något hinder så får SAS402 i denna situation ofta radarledning, vektorering, till finalen och kan kapa några minuters flygtid.

6.4. Användartest

Ett enklare användartest av den implementerade demonstratorn genomfördes 2021-11-10 med en mycket erfaren approach-flygledare från ATTC Stockholm. Vid detta användartest introducerades den ANN-baserade prediktionsförmågan och flygledaren fick därefter leda trafik under två pass på cirka 20 minuter var. Testdeltagaren ledde in åtta flygplan per scenario och väder och vindförhållanden var enkla att hantera. Scenariot var delvis förenklat utan starter och Bromma-trafik, men också lite försvårat av att flygledaren hade ansvar för hela TMA:t trots den aktuella trafikintensiteten. I verkligheten hade en flygledare i samma position inte varit ansvarig för så många flygplan samtidigt utan fler positioner hade varit öppna. Deltagaren tvingades därigenom agera aningen mycket i rollen som Final Director och behövde ägna relativt mycket uppmärksamhet åt att vektorera flygplanen. Han hann därför inte riktigt använda prediktionen i tidslinjen eller sekvensindikationen i etiketten. Information om sekvensen skulle vara mer användbar om han haft tid att agera i rollen som APP-C, Approach Coordinator. Värde att få ett förslag på vilken turordning i landningssekvensen ett flygplan har bör öka allt eftersom det är fler flygplan på ingång, i komplexa scenarier och längre utanför TMA-gränsen. Deltagaren vid användartestet är också mycket erfaren och kunde snabbt även utan stöd se den troliga sekvensen i dessa scenarier.



Figur 14. Exempel på luftläget från användartestet.

I Figur 15 nedan ges ett exempel på situation från användartestet där ANN-prediktionen utgår från hur trafiksituationen brukar spela ut och jämförs med hur informationen i färdplanen anger att situationen kommer spela ut. I den vänstra delen av figuren visas hur den planerade rutten för NAX74F med inpassering i TMA via punkten ELTOK ser ut, utpekad med FLEG-funktionen (Flight Leg) och med ett för rapporten förstärkande grönt streck norr om FLEG:en. Detta motsvarar den prediktion som en så kallad performance-modell i flygplanets FMS predikterar kommer hända, eftersom det är aktuell färdplan som anges och tiden för landning är 35 minuter och sju sekunder över hel timme, vilket framgår av tidsangivelserna längs FLEG. I tidslinjen visas prediktionen enligt ANN-modellen som anger 30 minuter och 48 sekunder över hel timme för landning.

I den högra delen av figuren visas snarare den händelseutveckling som ANN-modellen gör sin prediktion ifrån. Observera att författarna här visar vad som sker om NAX74F skulle få en flygvägsförkortning, *shortcut*, och gå via punkten LNA, för rapporten förstärkt med ett gult streck söder om FLEG. Detta är en möjlighet som flygledarna kan utnyttja om förutsättningarna tillåter det, och den är relativt vanlig men finns inte som en publicerad procedur. Eftersom luftläget ser ut som det gör i figuren så predikterar ANN att landningstid är 30 minuter och 48 sekunder över hel timme (vilket syns i tidslinjen, 30:53 är tiden enligt performance-modellen). ANN ser alltså vad som troligen kommer ske gällande landningstid med givande förutsättningar, snarare än vad som är planerat och klarerat. Notera att den FLEG som visas i den högra delen av figuren inte är något som ANN enkelt skulle kunna förklara och visas av pedagogiska skäl för att förklara att det är denna vanliga flygvägsförkortning som ligger bakom prediktionen 30:48.



Figur 15. Jämförelse rutt enligt färdplan (vänster) och vad ANN troligen utgått i sin prediktion (höger).

Några diskussionspunkter och observationer från detta användartest återges i punktlistan nedan:

- Detta användartest var första gången deltagaren introducerades till den tänkta funktionen men full klarhet om att det är användbart att få information om det troliga sekvensnumret från ANN-prediktionen för en flygledare i position uppnåddes inte. ANN-prediktionen dock bättre än att utgå från direkt avstånd mellan flygplan och flygplats, men för en approach-flygledare i position är nog simuleringsbaserad prediktion/FMS information nedlänkat via EPP mer användbart som stöd, se lösningsförslag i BKI-projektet (Castor, m.fl, 2021). Vid mycket trafik och hög kognitiv belastning, och stort behov av separation med vektorering kan sekvensprediktionen troligen vara användbar, men ska eventuellt vara något flygledaren då kan ta fram då, snarare än att det visas hela tiden. Notera dock att det nu pågående SESAR-projektet FAST också tänker sig indikation av sekvensnumret i etiketten, även om det inte tas fram genom någon maskininlärningsalgoritm. Även vid överlämning mellan flygledare vid skiftbyte bör ANN-prediktionen kunna vara ett stöd. Eventuellt gör ANN-prediktionens störst nytta som en ytterligare datakälla för input till A-CDM¹⁶ processen (*Airport Collaborative Decision Making*). Brandbilar, personal i turn-around processen och väntande taxibilar skulle troligen ha nytta av den här typen av prediktionerna. Dessa typer av aktörer kommer knappast få tillgång till de skarpa flygledningssystemen, medan ANN-prediktionerna kan göras utifrån öppen AD-B data.
- Flygledarnas inmatningar i det skarpa flygledningssystemet TopSky innehåller mycket värdefull information som tillsammans med luftlägesinformation av den typ som användes i detta projekt troligen skulle kunna användas för att förbättra prediktionerna avsevärt.
- Utöver beräkning av närmaste flygplan skulle motsvarande variabler för åskmoln (CB, Cumulonimbus) också kunna ingå. Förekomst av CB är en av de faktorer som i hög grad skapar svårhanterade trafiksituationer.
- I BKI-projektet diskuterades möjligheten att flygledarna ska kunna styra sekvensen genom att bara ange sekvensen i en tidslinje och att stödsystem sedan gör om det till detaljerade klareringar. ANN-prediktion som genomförts i detta projekt skulle då kunna användas för att sätta initialvärden/förslag för detta, som flygledaren därefter kan modifiera om sekvensen behöver ändras.
- Om Arlanda flygplats skulle gå mot ökad användning av så kallade STAR:s¹⁷ (*Standard Terminal Arrival Route*) och så kallade slutna procedurer i kombination med RNP AR-procedurer skulle man kunna träna en modell för varje konfiguration av dessa och troligen kunna förbättra prediktionen. Genom fasta procedurer minskar osäkerheterna och värdet av informationen i flygledarnas inmatningar minskar då.

¹⁶ A-CDM, <https://www.eurocontrol.int/concept/airport-collaborative-decision-making>

¹⁷ STAR, <https://skybrary.aero/articles/sids-and-stars>

- Simuleringsbaserad prediktionen och ANN-baserade prediktion är två olika typer av prediktion med olika beteende och användningsråden. Den simuleringsbaserade prediktionen visar vad planen är, och vad som kommer hända, givet nuvarande klareringar. Den ANN-baserade prediktionen visar hur utfallet, landningstid och därigenom landningssekvens, brukar bli givet nuvarande förutsättningar. Det tar längre tid innan konsekvensen av olika klarering syns med den ANN-baserade prediktionen, men den kan göra kvalificerade antaganden om vad som kommer hända, beroende på hur liknande situationer brukar spela ut.
- Projektet genomförde på IRIS-programmöte 2021-11-12 en enklare demonstration av projektets resultat. Deltagare var representanter från IRIS-programmets parter, dvs. personal från Swedavia, LFV och Novair. Sju personer deltog. En inspelning som återger vad som demonstrerades gällande skillnaden i prediktionsförmåga mellan AI-baserad prediktion enligt den typ som studerats i projektet och simuleringsbaserad prediktion för finns tillgänglig på <https://geistt.com/projects/aibaland>. Lösenordet finns tillgängligt via kontakt med martin.castor@geistt.com, martin.insulander@geistt.com eller anette.nas@swedavia.se.

7. Diskussion

De av projektet framtagna ANN-modellerna lyckas med tillgängliga träningsdata prediktera landningstiden med cirka en minut i medelfel. Detta är tillräcklig precision för att prediktionen av sekvensen inte ska "hoppa" i demonstratorn. Den erfarna flygledare som testat systemet bedömde det som intressant som en del i ett beslutsstöd, även om hans erfarenhet gjorde att han kunde se/förutsäga sekvensen väldigt väl även utan beslutsstöd.

Informationen i ett flygplans FMS har högre precision än ANN-prediktionen, men visar prediktionen för det enskilda flygplanet och vad som är planerat, och är inte en prediktion av helheten och "hur det brukar bli". De ANN-modeller som projektet använde lyckas utifrån "ett tidsfruset ögonblicksläge", prediktera landningstiden tillräckligt bra för att vara användbart som en komponent i ett beslutsstöd. Notera att varje sådant ögonblicksläge grundar sig på hur erfarna flygledare har hanterat liknande luftlägen under 2019. Genom träningen av det neurala nätverket kan man populärt uttryckt därför säga att nätverket får tillgång till erfarenheten från alla ATCC:s flygledare och använder denna erfarenhet för att göra sina prediktioner.

Ökad precision i prediktionen är givetvis alltid intressant och det går ofta att göra åtgärder för att förbättra prediktionsförmågan. Projektet gjorde en noga genomtänkt transformation av luftlägesdatan med positioner (GAIA-data) som LFV delgav projektet, till den träningsdata och särdrag som användes i projektet för att träna de ANN-modeller som användes. Att vidareutveckla de särdrag som användes (*feature engineering*) kan eventuellt ge något mer exakta prediktioner. Ett eventuellt uppföljande projekt bör beakta detta. ANN kan också använda bilder som särdrag, för att komplettera tabelldata, eller enbart bilder. Detta gjordes inte i projektet efter rekommendationer att det kan bli alltför beräkningsintensivt, men skulle kunna utforskas vidare i ett uppföljande projekt.

En diskussionspunkt som i princip alltid finns kring AI-projekt som använder ANN ansatser är frågan om "Explainable AI", dvs. hur kan användaren förstå hur system där ANN används kommer fram till de svar det gör. Denna förståelse är en central komponent för att en användare ska våga lita på systemet och använda det operativt. Projektet har inte haft utrymme att göra några insatser på detta område, utöver det faktum att ingående särdrag är enkla att var för sig förklara för flygledarna. Hur de vägs samman är dock mycket svårare att förklara och detta är en av ANN-ansatsens svagheter.

En annan ständigt återkommande fråga rör etiska implikationer av användningen av olika AI-baserade system. En mängd frågor att bota i finns här. Projektet har tagit del av riktlinjer kring detta från EU Joint Research Centre (EU JRC, 2017) och EU High level group on AI (EU HLEG AI, 2018), men i övrigt inte gjort några insatser gällande detta viktiga område. Virginia Dignum, professor vid Umeå universitet och medlem i EU HLEG AI, använde följande ord vid Vetenskapsrådets seminarie kring "Ansvarsfull teknisk utveckling – AI, robotik, etik", den 8 mars 2019:

- AI påverkar och är påverkad av våra sociala system
- Design är aldrig värdeneutral
- Öppenhet och tydlighet är centralt – ansvarsskyldighet (*accountability*), ansvarsfullhet (*responsibility*), och transparens (*transparency*) är ledord för utvecklingen inom området
- Optimal AI är förklaringsbar AI
- AI-system är artefakter som utvecklas av oss för våra egna syften
- Vi sätter begränsningar

I princip handlar det alltså inte om vad AI-system kan göra, utan om vad vi vill att de ska göra.

Förhoppningsvis har rapporten lyckats förmedla den centrala insikten att prediktionen som nätverken lyckas göra bygger på historiska data och hur flygledare brukar hantera motsvarande trafiksituationer. Om förutsättningarna ändras, exempelvis genom att ett nytt luftrum införs eller om trafiken förändras på något mer radikalt sätt så kommer nätverkets förmåga att prediktera landningstiden att minska. Om man skulle använda ANN i operativa system är det alltså centralt att det finns en strategi för när och på vilken datamängd som nätverket tränas igen. En annan central insikt är att ANN ser mönster i den historiska datamängden, men för att kunna hantera situationer för vilka det inte finns mönster, som exempelvis nödsituationer och andra avvikelser, måste det alltid finnas en mänsklig flygledare med som bedömer allvarlighet och ser lösningar. En slutsats är därför att olika AI-baserade ansatser som ANN kan bidra med olika typer av flygsäkerhets- och optimeringsrelaterade funktioner, men att kompetensen hos mänskliga flygledare är mycket svår att ersätta.

8. Rekommendationer

- **Skapa ett uppföljande projekt.** Analys av trafiksituationen med hjälp av artificiella neurala nätverk har genom projektet visat sig genomförbart. Prediktionerna som gjordes bedömdes vara användbara som en komponent i ett beslutsstöd som stöder flygledarnas operativa arbete, och skapar därmed i förlängningen utrymme för fler kurvade RNP AR-inflygningar. En fördel med ANN-prediktionerna är att de utifrån redan tillgänglig ADS-B data kan göra prediktioner på hur olika situationer brukar spela ut. Det behövs ingen ny infrastruktur eller dataflöden och ANN-prediktionen kan göras utifrån en statisk luftlägesbild. Troligen kan maskininlärningsbaserade prediktioner som de ANN-prediktioner som testats i detta projekt användas för att komplettera andra typer av prediktioner, exempelvis simuleringsbaserad prediktion baserat på de performance-modeller som finns i flygplans FMS. Forskningsfinansiärer och systemägare inom luftfartsområdet bör därför skapa förutsättningar för ett uppföljande projekt.
- **Använd fler och vidareutvecklade luftlägesrelaterade särdrag.** Mängden och typen av särdrag kan eventuellt leda till bättre prediktionsförmåga. Ett uppföljande projekt bör göra motsvarande analyser med ytterligare utvecklade särdrag, så kallad *feature engineering*. ANN har förmåga att använda bilder som särdrag, som skulle kunna rymma mycket information, även om det kan leda till beräkningsmässiga prestandautmaningar. Eventuellt kan prediktionsförmågan också förbättras genom att träna modeller på olika delar av databasen, exempelvis att inkludera information om vilken landningsbanan som använd i anropet till modellen.
- **Inkludera särdrag relaterade till flygledarnas systeminmatningar.** LFV registrerar alla inmatningar flygledarna gör i sitt flygledningssystem TopSky. Denna data bedöms vara en värdefull källa att också träna ANN på, separat eller som komplement till trafikdata som användes i detta projekt. Denna data på flygledarnas inmatningar har dock högre sekretessgrad än öppen ADS-B trafikdata. Ett projekt där denna typ av data analyseras skulle behöva mycket långa ledtider om någon analys skulle göras utanför LFV, pga. de säkerhetsgodkännande som kommer behövas.
- **Gör tester och valideringar mot skarp trafik i realtid.** ANN-modeller av den typ som tagits fram av projektet kan relativt enkelt stimuleras av skarp trafik för validering och analys av hur bra prediktionerna blir. Detta kan göras fristående från dagens flygledarsystem och helt baserat på öppen ADS-B data. Om informationen ska presenteras på samma sätt som i demonstratorn som beskrivs i avsnitt 6.3 krävs dock naturligtvis integration med TopSky-systemet.

9. Referenser

Castor, M., Insulander, M., Strid, E., Hellman, E., Borgvall, J. & Rappich, L. (2021). Beslutsstöd Kurvad Inflygning. Slutrapport BKI-projektet inom IRIS-programmet.

Chollet, F. (2017). Deep Learning with Python. Manning.
<https://www.manning.com/books/deep-learning-with-python>

EU HLEG AI (2018). The European Commission's high-level expert group on Artificial Intelligence. A definition of AI – Main capabilities and scientific disciplines.
https://ec.europa.eu/futurium/en/system/files/ged/ai_hleg_definition_of_ai_18_december.pdf

EU JRC (2018). Artificial intelligence – European Perspective. JRC113826 EUR 29425. <https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/artificial-intelligence-european-perspective>

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
<http://www.deeplearningbook.org>

Russel, S., & Norvig, P. (2021). Artificial Intelligence: A Modern Approach 4th Ed.
<http://aima.cs.berkeley.edu>