

Jouka Matias Ahponen

KONEOPPIMISEN HYÖDYNTÄMINEN LENTOKONEIDEN JÄRJESTELMÄ- SUUNNITTELUSSA

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Ohjaaja: Heikki Saha
Toukokuu 2022

TIIVISTELMÄ

Jouka Ahponen: Koneoppimisen hyödyntäminen lentokoneiden järjestelmäsunnittelussa
Kandidaatintyö
Tampereen yliopisto
Konetekniikan tutkinto-ohjelma
Toukokuu 2022

Tekoälyn ja koneoppimisen merkitys yhteiskunnassa on korostunut merkittävästi viimeisten vuosikymmenten aikana. Tässä tutkielmassa on tarkoitus tutkia koneoppivien menetelmien hyödyntämisen tilaa ja tulevaisuutta lentokoneiden järjestelmäsunnittelussa. Tavoitteena on löytää vastaukset koneoppivien menetelmien tutkimuksen tilasta, sekä löytää sopivia järjestelmäsoveluskohteita, joissa voidaan hyödyntää koneoppimista. Työ on suoritettu kirjallisuustutkimuksena.

Työ on jaettu kahteen osaan. Ensin tutustutaan yleisesti koneoppiviin menetelmiin, erityisesti geneettisiin algoritmeihin ja neuroverkkoihin. Lisäksi tässä osassa käsitellään menetelmiä tiedon keruuseen ja käsittelyyn liittyen koneoppimisen näkökulmasta. Toisessa osassa tutustutaan tarkemmin koneoppimista hyödyntäviin suunnittelu- ja järjestelmäsoveluksiin. Erityinen tarkastelu tehdään aerodynaamiseen suunnitteluun ja optimointiin, sekä lentokoneen moottoreiden suunnitteluun ja optimointiin. Lisäksi tarkastellaan koneoppimisen hyödyntämistä avioniikkajärjestelmissä ilma-aluksen törmäyksenestojärjestelmän kautta.

Tutkielmassa havaittiin, että tutkimusta koneoppimisen hyödyntämisestä lentotekniikan parissa on jo tehty jonkin verran ja tulokset ovat pääasiassa olleet lupaavia. Hyviä tuloksia oli saatu erityisesti suunnittelusoveluksissa korvaamalla numeeriset laskentamenetelmät koneoppivilla korvikemalleilla. Koneoppivien menetelmien tulokset vastasivat hyvin numeeristen menetelmien tuloksia, ja erityisesti niiden laskenta-aika oli huomattavasti lyhyempi. Suunnittelun lisäksi lupaavia tuloksia saatiin myös koneoppimisen käytöstä osana lentokoneen järjestelmiä.

Tutkielman perusteella koneoppimisella tulee olemaan hyvin suuri merkitys tulevaisuudessa lentokonetekniikassa. Erilaisia käyttökohteita on useita, eivätkä ne rajoitu pelkästään järjestelmäsunnitteluun. Tällä hetkellä on tärkeää tunnistaa sopivimmat käyttökohteet ja menetelmät, sekä varmistaa koneoppivien menetelmien turvallisuus ja luotettavuus.

Avainsanat: Koneoppiminen, Lentokonejärjestelmät, Järjestelmäsunnittelu, Optimointi, Korvikemalli

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. KONEOPPIMINEN JA SEN TAUSTA	3
2.1 Tekoäly ja koneoppiminen	3
2.1.1 Koneoppivat järjestelmät	3
2.1.2 Geneettiset algoritmit	4
2.1.3 Neuroverkot	6
2.1.4 Haasteet ja riskit	8
2.2 Tiedon käsittely ja hyödyntäminen	9
2.2.1 Tiedon keruu ja tiedonlouhinta	10
2.2.2 Olemassa olevan tiedon siirtäminen ja yhdistäminen	11
3. KONEOPPIVAT SUUNNITTELU- JA JÄRJESTELMÄSOVELLUKSET	13
3.1 Aerodynaaminen mallintaminen ja optimointi	13
3.2 Lentokoneen moottorijärjestelmien mallintaminen ja optimointi ...	16
3.3 Ilma-aluksen törmäyksenestojärjestelmän parantaminen	17
3.3.1 Ilma-aluksen törmäyksenestojärjestelmä yleisesti	17
3.3.2 Koneoppivien menetelmien hyödyntäminen liikenteen törmäyksenestojärjestelmässä	19
4. KONEOPPIMISEN TULEVAISUUS LENTOKONETEKNIKASSA	22
5. YHTEENVETO	24
LÄHTEET	26

1. JOHDANTO

Tekoälyn ja koneoppimisen merkitys uusina tieteenaloina on korostunut merkittävästi viimeisten vuosikymmenien aikana. Tähän mennessä tekoälyä ja koneoppimista on tutkittu erityisesti informaatio- ja tietojenkäsittelytieteissä. Yhä enemmän algoritmit ohjaavat maailmaamme laskentatehon ja teknisten sovellusten kehittyessä. Tämä voidaan nähdä jo nyt erityisesti suurten teknologianyhtiöiden käyttäessä koneoppivia algoritmeja sisältönsä optimointiin. (Jordan & Mitchell 2015) Erityisesti viimeisen vuosikymmenen aikana tekoälyn tuomia mahdollisuuksia on alettu tosissaan tutkimaan myös muilla tieteenaloilla, esimerkiksi teknisessä suunnittelu- ja kehitystyössä (Wuest et al. 2016, s. 23; Brunton et al. 2021, s. 1–3).

Koneoppimisella tarkoitetaan itsenäisesti toimivaa järjestelmää, jonka tarkoitus on kehittää itse itseään pohjatiedon ja käyttäjän toiminnan perusteella. Laajasta tekoälystä poiketen sen käyttötarkoitus tai tehtävä on rajattu käyttöympäristöön (Jordan & Mitchell 2015). Lisäksi koneoppimisen hyödyntämisen kanssa on tärkeää muistaa edelleen ihmisen rooli osana järjestelmää suunniteltaessa uusia järjestelmiä (Hall et al. 2020, luku 2). Ihmisen täytyy pystyä ymmärtämään ja arvioimaan koneoppivan järjestelmän toimintaa, jotta saadut tulokset voidaan vahvistaa järkevinä (Alpaydin 2021, s. 186–189).

Tekoälyn ja koneoppimisen odotetaan tuovan suuria muutoksia myös ilmailualalla. Euroopan lentoturvallisuusvirasto (engl. *European Union Aviation Safety Agency, EASA*) (2020) tuo esiin mahdollisuuksia koneoppiville sovelluksille erityisesti lentokoneiden suunnittelussa ja operoinnissa, kunnossapidossa, liikenteenohjauksessa ja useassa muussa osa-alueessa. Lentoliikenteen lisääntyminen ja ilmastonmuutos luovat erityisiä haasteita ja painetta uusille ratkaisuille lentokonetekniikassa. Koneoppivilla sovelluksilla voi olla merkittävä rooli näiden ongelmien ratkaisussa tulevaisuudessa. (EASA 2020)

Tässä kandidaatintyössä on tarkoitus selvittää koneoppimisen roolia lentokonetekniikassa. Työn tutkimuskysymykset ovat:

- Millaisia sovelluksia tutkitaan tai on jo käytössä?
- Mitkä järjestelmät tai sovellukset ovat sopivimpia?

Aihe on rajattu erityisesti koneoppimisen tutkimiseen järjestelmäsuunnittelun näkökulmasta. Aiheen rajausta on ollut aiheellinen, sillä lentotekniikka ja koneoppimisen yhdis-

tämistä voidaan tarkastella useasta eri näkökulmasta ja osa-alueita on monia, kuten edellä kävi ilmi. Esimerkiksi koneoppivat kunnossapito- tai valmistusmenetelmäsovellukset on jätetty työn rajauksen ulkopuolelle.

Lisäksi työssä keskitytään tarkastelemaan koneoppimisen hyödyntämistä ihmiskeskeisestä näkökulmasta. Tällä tarkoitetaan sitä, että koneoppiminen on helpottamassa ihmisen työtä osana järjestelmää, mutta ei korvaa ihmistä kokonaan. Eli suunnittelutehtävissä ihminen on yhä päätöksenteon keskiössä lopullista valintaa tehdessä. Esimerkiksi autonomisen ilmailun tavoittelu tekoälyn tai koneoppimisen avulla jätetään kokonaan tutkimuksen ulkopuolelle.

Työ suoritettiin kirjallisuustutkimuksena käyttäen lähdeaineistona pääasiallisesti alan tieteellistä kirjallisuutta koneoppimista hyödyntävistä sovelluksista lentokonetekniikassa. Lisäksi hyödynnettiin myös yleisesti lentokonetekniikan sekä tekoälyn ja koneoppimisen teoriakirjallisuutta sekä tutkimuksesta syntynyttä kirjallisuutta. Lähdemateriaalia löytyi kattavasti erityisesti kansainvälisestä kirjallisuudesta englannin kielellä. Lähdemateriaali rajoittuu pääasiallisesti vuosille 2010–2022 muutamaa poikkeusta lukuun ottamatta.

Työn rakenne koostuu viidestä pääluvusta. Toisessa luvussa käsitellään koneoppimista yleisesti teoreettisesta näkökulmasta. Luvussa esitellään koneoppimisen peruseriaatteen sekä perehdytään tarkemmin kahteen koneoppimisen toteutusmenetelmään: geneettisiin algoritmeihin ja neuroverkkoihin. Lisäksi luvussa käsitellään koneoppimiselle olennaisia tiedonkäsittelyn menetelmiä kuten tiedonlouhinta ja olemassa olevan tiedon yhdistäminen. Kolmannessa luvussa käsitellään varsinaisia suunnittelu- ja järjestelmäsovelluksia sekä esitellään aiheesta tehtyjä tutkimuksia sekä niiden tuloksia.

Neljännessä luvussa siirrytään tutkimustulosten analyysiin ja johtopäätöksiin. Lisäksi luvussa tuodaan esiin kirjoittajan omia päätelmiä ja pohdintaa, sekä tarkastellaan työn rajausta tarkemmin. Viidennessä ja viimeisessä luvussa käydään vielä merkittävimmät löydökset ja tulokset läpi, sekä kootaan tutkielman tärkein anti yhteen.

2. KONEOPPIMINEN JA SEN TAUSTA

Tässä luvussa käydään läpi koneoppimisen teoreettista taustaa. Koneoppiminen on lopulta hyvin laaja kokonaisuus, joka käsittää useita erilaisia menetelmiä, algoritmeja ja laskentatapoja. Tarkoituksena on esitellä erikseen muutamia yleisimpiä koneoppimisen menetelmiä kuten geneettiset algoritmit ja neuroverkot, joihin perustuvia sovelluksia esitellään luvussa 3. Lisäksi luvussa esitellään myös koneoppimisen kannalta tärkeitä tiedonkäsittelymenetelmiä.

2.1 Tekoäly ja koneoppiminen

Tekoälylle on useita eri määritelmiä, ja sen yksiselitteinen määrittely on hankalaa *älykkyys*-käsitteen haasteellisuuden vuoksi (Wang 2019). Ailisto et al. (2018, s. 7) ovat määritelleet tekoälyn seuraavasti: ”Tekoälyn avulla koneet, laitteet, ohjelmat, järjestelmät ja palvelut voivat toimia tehtävän ja tilanteen mukaisesti järkevällä tavalla”. Tekoäly on yleisesti luokiteltu yleisen ja sovelletun tekoälyn haaroihin. Yleisellä tekoälyllä tarkoitetaan tekoälyä, jolla on jonkinlainen tietoisuudeksi määriteltävä ominaisuus. Tällä hetkellä tämänkaltaista tekoälyä ei ole vielä pystytty luomaan. Koneoppiminen sijoittuu sovelletun tekoälyn haaraan edustaen datapohjaisia menetelmiä. Se on yksi suosituimmista tekoälyn teknisistä toteutustavoista. (Kääriäinen et al. 2018, s. 21)

2.1.1 Koneoppivat järjestelmät

Koneoppivalla järjestelmällä tarkoitetaan ohjelmistokokonaisuuksia, jotka pystyvät oppimaan sille annettua dataa ja tunnistamaan sieltä tiettyjä toistuvia kaavoja, joiden avulla se pystyy tekemään valistuneen arvauksen tulevaisuudesta (Dangeti 2017, s. 9). Lisäksi järjestelmän kyky oppia eli kyky kehittää itse itseään on keskeistä. (Kääriäinen et al. 2018, s. 21).

Koneoppivaan järjestelmään sisältyy algoritmit olennaisena osana järjestelmää. Kieli-toimiston sanakirja (2022) määrittelee algoritmin sarjana täsmällisesti määriteltäviä käskyjä tai toimenpiteitä jonkin tehtävän suorittamiseksi. Algoritmit voidaan siis käsittää eräänlaisena ohjekirjana sille, mitä sisään tuleville syönteille tulee tehdä, jotta päästään haluttuun lopputulokseen. Koneoppivissa järjestelmissä oppiminen tapahtuu järjestelmälle annettujen algoritmien rajoissa (Shalev-Shwartz & Ben-David 2014, s. 25).

Koneoppivan järjestelmän oppiminen tapahtuu pääsääntöisesti kolmella tavalla: ohjautusti, ohjaamattomasti ja vahvistavasti. Ohjatussa oppimisessa järjestelmää pyritään opettamaan esimerkkien avulla ja antamalla palautetta siitä, miten järjestelmä suoriutui tehtävästä. Kuvantunnistusjärjestelmät ovat yksi esimerkki tällaisesta oppimismallista. Ohjaamaton oppiminen toimii muuten samalla tavalla kuin ohjattu oppiminen, mutta järjestelmälle ei anneta palautetta sen suoriutumisesta. Erilaiset järjestelytehtävät hyödyntävät usein ohjaamatonta oppimista. Vahvistetussa oppimisessa järjestelmä saa palautetta toiminnastaan ja järjestelmä pyrkii löytämään tien haluttuun tulokseen itsenäisesti käyttäen palautetta hyödyksi. Palaute ei kuitenkaan välttämättä ole välitöntä päätöksen jälkeen, kuten ohjatussa oppimisessa, ja järjestelmä käyttää useampaa palautetta kerralla oppimiseen. Otetaan esimerkiksi tietokoneen pelaama shakki, jossa tulee ottaa huomioon jokaisen siirron positiiviset ja negatiiviset vaikutukset, mutta näitä vaikutuksia ei voida tietää heti siirron jälkeen vaan vasta kun peli on edennyt tarpeeksi tai päättynyt. (Dangeti 2017, s. 9, 50, 360)

2.1.2 Geneettiset algoritmit

Geneettiset algoritmit ovat ongelmanratkaisussa käytettyjä algoritmeja, joita käytetään erityisesti optimointi- ja hakuongelmien ratkaisemiseen. Niiden idea pohjautuu osittain Charles Darwinin esittämään evoluutioteoriaan. Tavoitteena on hyödyntää luonnonvalinnan periaatteita koneoppivassa järjestelmässä parhaimman ratkaisun eli globaalin optimin löytämiseksi paikallisten minimien joukosta. Algoritmin toiminta etenee sukupolvittain. Jokaisen sukupolven parhaimpia yksittäisratkaisujen ominaisuuksia yhdistellään keskenään uuden sukupolven luomiseksi niin kauan kunnes lopetuksen ehdot täyttyvät. (Sivanandam 2008, s. 15, 29–30; Mishra & Sahoo 2017)

Geneettiset algoritmit ovat lopulta hyvin laaja koneoppimistutkimuksen osa-alue ja erilaisia algoritmisovelluksia on useita kymmeniä (Sivanandam 2008). Tässä työssä ole mahdollista käydä niitä kaikkia läpi, vaan tarkoituksena on esittää yleinen toimintaperiaate. Algoritmista 1 on esitelty yksinkertainen geneettinen algoritmi. Käytännössä suuri osa geneettisen algoritmin sovelluksista soveltaa yksinkertaisen algoritmin kaavaa tavalla tai toisella (Sivanandam 2008, s. 20).

```

function GeneettinenAlgoritmi():
2   alusta alkupopulaatio;
   while (lopetuksen ehdot eivät täyty)
4     for i in populaatio: //arvioidaan jokaisen populaation jäsentä
       arvioi sopivuus sopivuusfunktiolla;
6       valitse parhaat yksilöt;
       risteytä parhaat yksilöt;
8       suorita mutaatio; //muokataan yksilöitä hakuavaruuden laajenta-
miseksi
10      luo uusi populaatio; //parhaiden yksilöiden risteytykset ja mutaa-
tiot
12  return lopputulos;

```

Algoritmi 1. Yksinkertaisen geneettisen algoritmin pseudokoodi (muokattu lähteestä: Sari & Tuna 2018, s. 5)

Alkupopulaatio (engl. *initial population*) on satunnaisesti luotu ratkaisujen joukko yksittäisratkaisuja eli kromosomeja. Jokaisen kromosomin sopivuus arvioidaan sopivuusfunktiolla (engl. *fitness function*). Sopivuusfunktio vertaa jokaista kromosomia keskenään ja sitä, mitkä ratkaisut ovat parempia kuin toiset. Valintavaiheessa (engl. *selection*) parhaimmat ratkaisut varastoidaan. Kun kaikki kromosomit on käyty läpi, valitaan tietty määrä parhaiten suoriutuneita kromosomeja, joiden ominaisuuksia yhdistellään seuraavan sukupolven luomiseksi (engl. *crossover*). Yhdistely voi tapahtua esimerkiksi yhden pisteen risteytyksellä (engl. *single point crossover*), jolloin kaksi yksilöä leikataan yhdestä pisteestä ja ominaisuudet yhdistetään ristiin. Esimerkiksi yksilö A voidaan kuvata binäärilukuna 0111 ja yksilö B binäärilukuna 1101. Näiden yksilöiden risteytys puolestavälistä antaa uudet yksilöt 0101 ja 1111, kun molempien alku- ja loppuosa yhdistetään ristiin. Käytännössä risteytystapoja on useita erilaisia ja parhaimman risteytystavan valinta riippuu käyttökohteesta. (Sari & Tuna 2018, s. 4-5; Sivanandam 2008, s. 39-56)

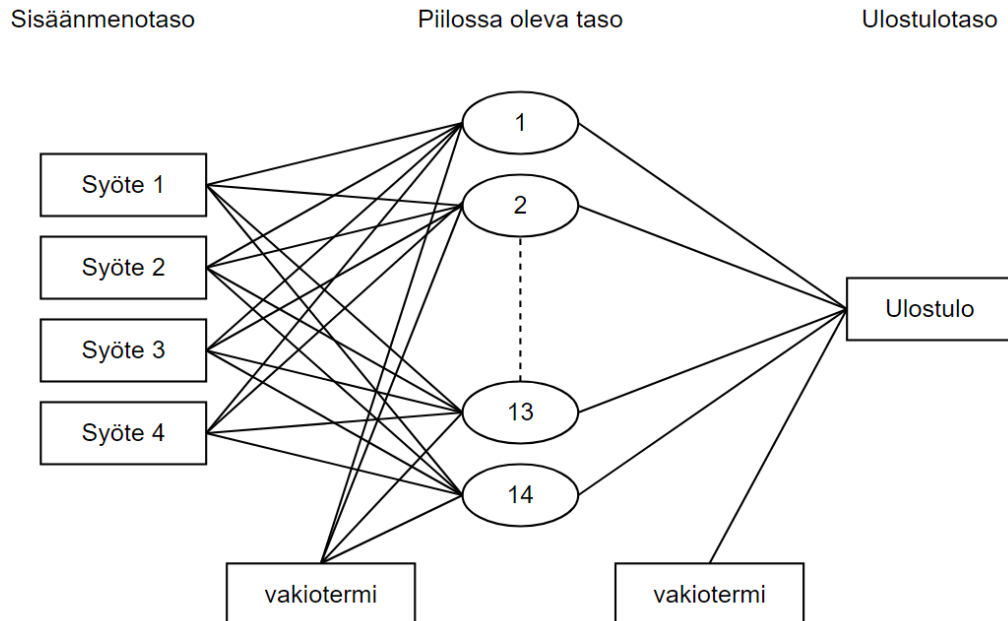
Risteytyksen jälkeen suoritetaan vielä mutaatio ennen uuden populaation luomista. Mutaatio on tärkeä osa geneettisen algoritmin toimintaa, jotta hakuavaruus (engl. *search space*) voidaan käydä läpi kokonaisvaltaisesti parhaimman ratkaisun löytämiseksi. Mutaatiolla tarkoitetaan yksilöiden sattumanvaraista muokkausta hienovaraisesti. Yksinkertainen mutaatio voi esimerkiksi kääntää yhden yksilön biteistä pienellä todennäköisyydellä. Mutaatioiden avulla voidaan estää, että algoritmi jäisi jumiin paikalliseen minimikohtaan. Näin voi käydä, mikäli yksittäinen bitti, joka johtaa lokaaliin minimiin jumittuu tiettyyn arvoon koko populaatiossa, eikä siten voi enää vaihtua risteytyksen keinoin. (Sari & Tuna 2018, s. 4; Sivanandam 2008, s. 56) Mutaatiota ei kuitenkaan halua lii- kaa, sillä mitä enemmän algoritmi sisältää mutaatiota, sen satunnaisempaa sen toiminta on (Sivanandam 2008, s. 57).

Lopulta kun mutaatio on suoritettu, luodaan uusi populaatio edellisen sukupolven parhaista ratkaisuista, niiden risteytyksistä ja mutatoituneista ratkaisuista. Koe suoritetaan uudestaan tälle populaatiolle. Algoritmi arvioi, valitsee, risteyttää ja suorittaa mutaation tälle uudelle populaatiolle. Tämä kaava pyörii niin kauan, kunnes lopetuksen ehdot täyttyvät. Esimerkkejä lopetuksen ehdoista on, kun sopivuudessa ei havaita enää muutosta usean sukupolven ajalta, sukupolvien määrän rajaaminen tai algoritmin toimintajan rajaaminen. Lopullista ratkaisua tulee vielä arvioida kriittisesti, miten hyvin se täyttää algoritmille asetetut tavoitteet. (Sivanandam 2008 s, 57–59)

Geneettiset algoritmit eivät välttämättä aina anna sopivaa tulosta. Syitä voi olla monia, kuten algoritmin liiallinen sattumanvaraisuus tai hakuavaruuden liiallinen laajuus, jolloin algoritmi ei ehdi löytää sopivaa tulosta tarpeeksi nopeasti (Sivanandam 2008, s. 34–36). Myös huonosti suunniteltu sopivuusfunktio voi tehdä geneettisestä algoritmista huonon, jolloin se valitsee huonoja yksilöitä luomaan uutta sukupolvea. Erityisesti monimutkaisemmissa sovelluksissa sopivuus voi olla monien osa-alueiden summa, jolloin tulee myös määrittää mitä ominaisuuksia sopivuusfunktion tulisi painottaa. (Sivanandam 2008, s. 41) Lisäksi on myös hyvä tunnistaa tilanteet, joissa perinteiset menetelmät ovat geneettisiä algoritmeja tehokkaampia ja tarkempia (Sivanandam 2008, s. 34–36). Yleisesti ottaen geneettiset algoritmit ovat kuitenkin erittäin tehokkaita erityisesti optimointitehtävissä ja niitä käytetään paljon useilla eri lentotekniikan osa-alueilla (Anderson 2003). Erilaisia lentotekniikan sovelluksia, joissa hyödynnetään geneettisiä algoritmeja, esitellään tarkemmin luvussa 3.

2.1.3 Neuroverkot

Neuroverkko (engl. *neural network*) on yksi tärkeimmistä koneoppimista toteuttavista menetelmistä erityisesti, kun kone pyrkii tunnistamaan tiettyjä kuvioita ja samankaltaisuuksia. Se perustuu dataohjattuun tiedonkäsittelyyn (Pietikäinen & Silvén 2019, s. 21). Neuroverkkoteorian inspiraationa on toiminut biologisten neuroniverkkojen toiminta ongelmanratkaisussa. Neuroverkko voidaan jakaa kolmeen eri tasoon: sisäänmenotasoon (engl. *input layer*), piilossa olevaan tasoon (engl. *hidden layer*) ja ulostulotasoon (engl. *output layer*). Käytännössä piilossa olevia tasoja voi olla useampia kuin yksi. Jokainen taso käsittää tietyn määrän neuroneita, jotka on yhdistetty viereisten tasojen neuroneihin synapseilla eli kahden neuronin välisellä kytkennällä. (Kwon 2011, s. 3–5) Neuroverkon rakennetta on havainnollistettu kuvassa 1, jossa piilossa oleva taso sisältää 14 neuronia.



Kuva 1. Kuvaus tyypillisestä neuroverkosta, jossa piilossa oleva taso sisältää 14 neuronia. (muokattu lähteestä Kwon 2011, s. 5)

Toiminta perustuu sisääntulossa olevien neuronien aktivoitumiseen riippuen sisään tulevasta datasta, joka aktivoi sisäänmeno tasossa tietyt neuronit. Sisäänmenotasossa aktivoituneet neuronit käsittelevät datan ja lähettävät siitä tiedon seuraavalle tasolle synapseja pitkin. Seuraavan tason neuroneihin tulevista signaaleista lasketaan painotettu summa, johon lisätään vielä vakiotermi (engl. *bias*) joka vaikuttaa neuronin aktivoitumisherkkyteen. (Kwon 2011, s. 3–5)

Neuronit aktivoituvat aktivaatiofunktion perusteella. Yleisimmät aktivointifunktiot ovat porrasfunktio, lineaarinen funktio ja S-käyrä. Uudet aktivoituneet neuronit lähettävät tiedon jälleen eteenpäin synapseja pitkin. Lopulta tiettyjen neuronien aktivoituessa ulostulotasolla järjestelmä tulkitsee neuronien tilan ja ilmoittaa tuloksen. (Kwon 2011, s. 3–5)

Tarkastellaan seuraavaksi neuroverkkojen oppimista tarkemmin. Kuten jo edellä mainittiin, saavat neuronit arvonsa signaaleista lasketun painotetun summan kautta. Toisaalta jopa yksinkertaisimmissa kuvantunnistus neuroverkoissa voi neuroneita olla useita satoja. Painokertoimien laskeminen käsin jokaiselle neuronille ei siten ole suositeltavaa. Enemmän tavoitteena on järjestelmän kyky määrittellä painokertoimet itse opettamisen kautta. Tätä varten järjestelmälle luodaan hukka-funktio (engl. *loss function*), jonka tavoitteena on kuvata kuinka hyvin neuroverkon ulostulo vastaa haluttua dataa. Yksi yleisimmistä hukka-funktioista on keskineliövirhe (engl. *Mean squared error, MSE*):

$$MSE = \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}, \quad (1)$$

jossa y_i on i -alkion haluttu arvo perustuen esimerkiksi kokeellisiin testeihin, \hat{y}_i saman alkion arvioitu arvo ja N alkoiden lukumäärä. Lopullisena tavoitteena on hukkafunktion minimoiminen, sillä mitä pienempi hukkafunktion arvo on, sitä paremmin neuroverkon arvot täsmäävät havaittua dataa. Tähän käytetään gradienttimenetelmää, jossa hukkafunktiolle lasketaan gradientti painokertoimien suhteen, ja painokertoimia muutetaan asteittain gradientin negatiiviseen suuntaan, jonne funktio vähenee nopeimmin. (Kwon 2011, s. 3–5, 9, 12–15)

Yksi neuroverkon oppimisen keskeisistä termeistä on vastavirta-algoritmi (engl. *backpropagation*). Vastavirta-algoritmilla tarkoitetaan neuroverkon opettamista viimeisestä kerroksesta eli ulostulokerroksesta takaisinpäin. Jos tiedetään valmiiksi mahdollisia ulostuloarvoja parametreille, joita neuroverkolla halutaan ratkaista, nämä arvot voidaan syöttää neuroverkolle ja antaa verkon käydä läpi tämä data gradienttimenetelmällä. Tällöin neuroverkko määrittelee itse sille annetun datan perusteella parhaimmat painokertoimet. Mitä enemmän laadukasta dataa on käytössä, sen tarkemmin neuroverkko pystyy painokertoimet määrittämään. (Kwon 2011, s. 9, 226)

2.1.4 Haasteet ja riskit

Kuten monen muun uuden teknologian kanssa, myös koneoppimiseen liittyy tiettyjä haasteita ja riskejä, jotka tulee ottaa huomioon. Tämänlaisia haasteita ja riskejä ovat esimerkiksi aineiston vinoutuminen tai mallin luominen liian pienellä aineistolla, joka ei anna luotettavaa tulosta. Toinen merkittävä haaste on mallin tulkittavuus ja ennustettavuus. Miten voidaan olla varmoja, että malli antaa halutunlaisia tuloksia? Yksi tärkeä keino on ohjelmistomallien vahvistaminen ja varmistaminen. Tällöin mallia testataan aineistolla, jota ei käytetty opetuksessa ja tutkitaan mallin antamat tulokset. Tämä ei kuitenkaan välttämättä riitä erityisesti opetusjoukon ja testausjoukon ollessa pieniä. (Alpaydin 2021, luku 7)

Eryteisesti mallin luominen liian pienellä aineistolla sekä tuloksen varmistaminen ovat yleisiä ongelmia koneoppimisen hyödyntämisessä erityisesti uusien järjestelmien suunnittelussa ja optimoinnissa. Jotta uusi laite tai osa voitaisiin optimoida, tarvitaan useita, joskus jopa tuhansia, numeerisen virtausdynamiikan eli CFD (engl. *Computational Fluid Dynamics*) -malleja tai fyysisiä kokeita, mutta jopa yksittäisen mallin tai kokeen suorittaminen voi viedä parhaimmillaankin useamman päivän (Min et al. 2017, s.

51). Min et al. (2017) ovat esitelleet tavan, jossa tietoa jo vanhoista olemassa olevista järjestelmistä ja laitteista yhdistellään ja siirretään käytettäväksi uuden järjestelmän tai laitteen suunnittelussa. Kuitenkin myös tässä mallissa tulee ottaa huomioon mitä dataa siirretään ja yhdistellään. Huonosti valittu data voi johtaa jopa huonompaan tulokseen kuin ilman siirtämistä tehty malli, mikäli data ei vastaa optimoitavan laitteen toimialuetta. (Min et al. 2017, s. 6) Tiedon siirtämiseen perehdytään vielä tarkemmin luvussa 2.2.2.

Lisäksi jokaisen järjestelmän käyttäjän pitäisi pystyä ymmärtämään ja arvioimaan mallin toimintaa. Ei riitä vain, että ulostulo on ymmärrettävässä muodossa, vaan järjestelmän täytyy myös pystyä selittämään, miksi se päätyi juuri tähän tulokseen. Tämänkaltaista tekoälyä kutsutaan myös selitettävissä olevaksi tekoälyksi (engl. *explainable artificial intelligence, XAI*). (Alpaydin 2021, s. 186–189)

Eriyksen tärkeää järjestelmän sisäisen toiminnan ymmärtäminen on turvallisuuskriittisissä toiminnoissa. Jo perinteisen automaation lisääntymisen on huomattu johtavan onnettomuuksiin, koska laitteen käyttäjä ei ole täysin sisäistänyt miten automaatio vaikuttaa järjestelmän toimintaan eri tilanteissa. (Ogle et al. 2008) Esimerkkejä on useita myös ilmailualalta, viimeisimpänä tunnettuna tapauksena Boeing 737 MAX onnettomuudet (Billings 1996, Gawron 2019 mukaan, s. 10; The Federal Democratic Republic of Ethiopia Ministry of Transport 2020).

Järjestelmiä ei voida jättää yksinomaan automaation tai koneoppivien algoritmien vaaraan, vaan myös järjestelmän käyttäjä tulee tunnistaa osana järjestelmää. Paras hyöty järjestelmistä saadaan, kun yhdistetään järjestelmän kyky käsitellä ja yhdistää suuria määriä dataa nopeasti sekä järjestelmän käyttäjän kyky tulkita järjestelmän toimintaa ja oikeanlaisen väliintulon mahdollisuus, mikäli järjestelmä ei toimi halutulla tavalla. (Ogle et al. 2008)

2.2 Tiedon käsittely ja hyödyntäminen

Luvussa 2.1 esiteltiin koneoppimisen teoriaa pääpiirteittäin. Jotta koneoppivasta mallista tai järjestelmästä saataisi mahdollisimman tehokas ja tarkka, täytyy sille syötetyn datan olla toiminnan kannalta merkityksellistä. Tämä vaatii usein tiedon käsittelyä ja yhdistelyä ennen sen syöttämistä koneoppivaan järjestelmään. Seuraavassa tutustutaan kahteen koneoppimisen kannalta merkitykselliseen käsittelymenetelmään: tiedonlouhintaan ja olemassa olevan tiedon siirtämiseen ja yhdistämiseen.

2.2.1 Tiedon keruu ja tiedonlouhinta

Kuten edellä luvussa 2.1 mainittiin, on koneoppiminen vahvasti riippuvainen tiedosta ja sille syötetystä aineistosta eli datasta. Tietotekniikan käytön lisääntyessä on myös datan kerääminen lisääntynyt eksponentiaalisesti (Hilbert & Lopez 2011). Nykyään dataa kerätään kaikesta mahdollisesta useilla tekniikan osa-alueilla. Tällaista kerättyä dataa kutsutaan massadatakksi (engl. *big data*). Toisaalta kaikki mahdollinen saatavissa oleva data ole välttämättä tarpeellista koneoppivan järjestelmän toiminnan kannalta, vaan ylimääräinen data saattaa jopa heikentää sen toimintaa (Gudivada et al. 2017, s. 5). Tiedonlouhinnan (engl. *data mining*) tarkoituksena on löytää merkityksellisiä riippuvuussuhteita laajoista aineistoista ja tehdä datasta siten hyödyllistä eri käyttötarkoituksiin (Chen et al. 1996, s. 1).

Lentokonetekniikassa tilanne massadatan keräämisen suhteen on erityisen hyvä. Dataa kerätään eri järjestelmien toiminnasta huomattava määrä jokaiselta lennolta lähes reaaliajassa. Tiedon keruuta on ollut vauhdittamassa erityisesti kunnossapidon optimointi ja järjestelmien tilanseurantaan perustuvan kunterusteisen kunnossapidon käyttöönotto. (Gerdes et al. 2016, s. 399–400)

Massadatan suhteen yksi ongelma on massadatan pirstaloituminen, sillä eri tahot keräävät dataa pääsääntöisesti omalta ydinalueeltaan; moottorivalmistajat moottoreista, lentoyhtiöt lentoreiteistä ja aikatauluista, lennonjohto ylilennoista ja niin edelleen. Dataa voi siis olla hankalasti saatavissa yhdestä paikasta, eri lähteiden ja formaattien yhdistäminen voi olla hankalaa, eikä datan laadusta välttämättä ole takeita. (Burmester et al. 2018, s. 56–60)

Nykyisessä massadatan tulvassa hyödyllisten toistuvuuksien, riippuvuussuhtien ja mallien löytäminen suoraan on hyvin hankalaa. Tätä varten massadataa käsitellään usein tiedonlouhinnan avulla edellä mainittujen ominaisuuksien hahmottamiseksi. Tiedonlouhinta itsessään käsittää laajan kirjon erilaisia menetelmiä käsitellä massadataa, ja menetelmä usein riippuukin lopullisesta käyttötarkoituksesta ja tavoitteesta. Yhteistä kaikille menetelmille kuitenkin on pyrkimys tiedon lisäämiseen datasta tehtävien löydösten avulla. (Ge et al. 2017) Tiedonlouhinta on usein keskeisessä asemassa erityisesti koneoppivan mallin luomisessa. (Bertoni et al. 2020, 2–3) Myös useat luvussa 3 käsiteltävät sovellukset sisältävät malleja, joissa on hyödynnetty tiedonlouhinta. Lisäksi myös luonteensa takia tiedonlouhinta itse hyödyntää usein koneoppivia menetelmiä osana datan analysointia (Ge et al. 2017).

2.2.2 Olemassa olevan tiedon siirtäminen ja yhdistäminen

Erityisesti dataohjatussa tekoälyssä, kuten neuroverkoissa, luotettavuuteen pyritään syöttämällä mahdollisimman paljon hyödyllistä dataa kaikkien vaihtoehtojen kattamiseksi (Pietikäinen & Silvén 2019, s. 50). Kuitenkin uusien järjestelmien suunnittelussa saatetaan törmätä niin kutsuttuun alustusongelmaan (engl. *cold start problem*), jossa suunnittelun alla olevasta järjestelmästä ei ole saatavilla yhtään dataa, ja uuden datan rakentaminen esimerkiksi CFD-malleilla olisi erittäin työlästä. Tällöin tietoa täytyy pystyä keräämään muilla tavoin esimerkiksi muista toimivista järjestelmistä tai simuloimalla järjestelmiä optimoidusti, ja yhdistelemällä näillä menetelmillä saatu tieto uuden mallin luomiseksi. (Min et al. 2017, s. 51–52)

Min et al. (2017) esittävät kolme menetelmää olemassa olevan tiedon hyödyntämiseen: Siirto-oppiminen (engl. *transfer learning*), moniajo-oppiminen (engl. *multi-task learning*) ja usean näkökulman oppiminen (engl. *multi-view learning*). Siirto-oppimisessa malli perustuu voimakkaasti muista malleista siirrettyyn tietoon ja näiden mallien vertailuun, josta uusi malli valitsee aiemmin parhaiten suoriutuneen mallin ja kokeilee sen soveltuvuutta uudessa järjestelmässä. Moniajo-oppivassa mallissa uutta tietoa pyritään saavuttamaan suorittamalla useita toisiinsa liittyviä tehtäviä samanaikaisesti. Moniajo-oppiminen voi olla hyödyllistä erityisesti monimutkaisissa järjestelmäkokonaisuuksissa, jotka koostuvat useista eri osajärjestelmistä. Yksi esimerkki tällaisesta järjestelmäkokonaisuudesta, on lentokoneen moottori, joita käsitellään vielä tarkemmin luvussa 3.2. Viimeisenä oppimisen mallina on ilmiön tarkastelu useasta eri näkökulmasta, ja lopulta eri näkökulmista saadun tiedon yhdistäminen. Monimutkaisen järjestelmän kaikkien parametrien samanaikainen tarkastelu esimerkiksi numeerisen simuloinnin kautta, voi tehdä mallista hitaan ja kalliin käyttää. Tarkastelua esimerkiksi tietyn komponentin osalta voidaan rajata eri parametreihin ja tehdä tarkastelu näiden parametrien näkökulmasta. Myöhemmin eri parametrien näkökulmista tehdyt simulaatiot voidaan yhdistää lopullisen optimoidun tuloksen aikaansaamiseksi. Toinen esimerkki usean näkökulman oppimisesta on myös erilaisen lähdedatan käyttö eri simuloinneissa ja näiden simulointien tulosten yhdistäminen. (Min et al. 2017, s. 52–54)

Tiedon siirtämisessä tulisi ottaa huomioon myös negatiivinen siirtäminen, joka heikentää mallin tai järjestelmän toimintaa. Mikäli siirtäminen tehdään huolimattomasti, voidaan vahingossa siirtää tietoa, joka ei ole mallin toiminnan kannalta oleellista, tai pahimmassa tapauksessa jopa ohjaa mallia väärään suuntaan. Erityisesti massadatasta käyttökohteen kannalta merkityksellisen tiedon valitseminen voi olla hankalaa, joten tiedon merkityksellisyyttä pitäisi pystyä mittaamaan. Toisaalta tiedon merkityksellisyyden mittaaminen on usein hankalaa. Siirron kohteesta on usein saatavilla hyvin vähän

tietoa alustusongelma takia, jolloin tiedon lähteen ja kohteen välistä suhdetta on vaikea arvioida kohteen kannalta merkityksellisen tiedon määrittelemiseksi. (Weiss et al. 2016, s. 29–32)

Olemassa olevan tiedon yhdistämisen hyödyntäminen on vielä melko tuntematon alue lentotekniikassa (Min et al. 2017, s. 52). Yleisesti menetelmää on kuitenkin käytetty onnistuneesti useissa ei koneoppimisen sovelluksissa, ja menetelmän hyödyntämisen eteen tehty tutkimus on ollut nousussa (Weiss et al. 2016). Todennäköisesti koneoppivien sovellusten lisääntyessä lentotekniikassa, myös olemassa olevan tiedon siirtäminen ja yhdistäminen nousee enemmän esiin erilaisissa sovelluksissa.

3. KONEOPPIVAT SUUNNITTELU- JA JÄRJESTELMÄSOVELLUKSET

Perehdytään seuraavaksi varsinaisiin koneoppimisen käyttökohteisiin lentokonetekniikassa. Tässä työssä tarkastelu jakaantuu kahteen osaan: suunnitteluovelluksiin ja järjestelmäsovelluksiin. Erityisesti suunnitteluovelluksia tarkastellaan luvuissa 3.1 ja 3.2 aerodynamiikan ja lentokoneen moottoreiden mallintamisen ja optimoinnin kautta. Luvussa 3.3 tehdään syvempi katsaus varsinaiseen järjestelmäsovellukseen tarkastelemalla ilma-aluksen törmäyksenestojärjestelmän toimintaa ja kehitystä.

3.1 Aerodynaaminen mallintaminen ja optimointi

Lentokoneen suunnittelu, erityisesti aerodynaaminen suunnittelu, on kallista ja aikaa vievää. Tietokoneiden laskentatehon lisääntyttyä, erilaiset numeeriset tietokonepohjaiset menetelmät, erityisesti numeerinen virtausdynamiikka ja elementtimenetelmä (engl. *finite element method, FEM*), ovat mahdollistaneet nopeamman tavan analysoida lentokoneiden järjestelmiä ja rakenteita. Toisaalta kyseiset menetelmät ovat yhä verrattain tehottomia useiden vaihtoehtojen samanaikaiseen analysointiin ja valintaan, tai edes yksittäisen rakenteen tai järjestelmän optimointiin. Optimointi vaatisi esimerkiksi useiden numeeristen simulointien suorittamista ja parametrien hienovaraista muuttamista. Samalla suunnittelijalta vaaditaan hyvää ymmärrystä aerodynamiikasta ja menetelmien toiminnasta. (Li et al. 2019, s. 2; Bertoni et al. 2020, s. 9–10, 15; Dong et al. 2021, s. 8) Viime aikoina optimoinnin automatisointia onkin tutkittu paljon ja tulokset ovat olleet lupaavia (Anderson & Aftosmis 2015).

Lopullisia menetelmiä aerodynaamisten rakenteiden ja järjestelmien optimointiin on tutkimuksissa esitelty useita erilaisia, eikä tässä työssä ole mahdollista tarkastella niitä kaikkia. Otetaan tarkastelun alle kuitenkin luvuissa 2.1.4 ja 2.1.5 esitellyt geneettiset algoritmit ja neuroverkot.

Skinner & Zare-Behtash (2018, s. 942–946) käsittelevät laajassa optimointimenetelmiin liittyvässä artikkelissa kattavasti useita eri menetelmiä mukaan lukien geneettisiä algoritmeja. Heidän mukaansa geneettiset algoritmit tuovat selviä hyötyjä aerodynaamiseen optimointiin seuraavasti:

- Kyky löytää useita optimaalisia ratkaisuja ja suorittaa valintaa suunnitelmien välillä
- Kyky prosessoida informaatiota samanaikaisesti optimoiden useita pisteitä suunnitteluavaruudessa
- CFD-koodi voidaan mukauttaa geneettisiin algoritmeihin ilman suuria muutoksia
- Geneettiset algoritmit pystyvät toimimaan myös tilanteissa, joissa laskentaan saattaa sisältyä numeerista kohinaa

Ongelmat algoritmin käytössä liittyvät lähinnä algoritmin optimointiin ja parametrien kuten sopivuusfunktion valintaan tai hakuavaruuden laajuuden rajaamiseen. Hakuavaruutta voidaan rajata esimerkiksi suorittamalla algoritmi useassa eri tasossa, jossa ensin koko hakuavaruudelle suoritetaan alustava karsinta tiettyjen kriteerien mukaan. Vain kriteerit täyttävät ratkaisut etenevät lopulliseen yksityiskohtaiseen tarkasteluun. (Skinner & Zare-Behtash 2018, s. 942–946)

Myös syväoppimista ja neuroverkkojen käyttöä on tutkittu jo jonkin verran. Dong et al. (2021) ovat tuoneet artikkelissaan esille erilaisten neuroverkkojen käyttöä useissa erilaisissa lentoteknisissä sovelluksissa kuten aerodynaamisessa optimoinnissa ja lentodynamiikan mallintamisessa. Heidän mukaansa neuroverkoilla on mahdollista lisätä CFD mallinnuksiin datapohjaisia malleja esimerkiksi turbulenssin mallintamiseen, tai korvata CFD-prosessi kokonaan neuroverkkopohjaisella korvikemallilla (engl. *surrogate model*). Lisäksi he mainitsevat artikkelissaan myös rakenteellisen mallinnuksen, jossa neuroverkoilla voidaan mallintaa lentokoneen rakenteisiin kohdistuvia monimutkaisia dynaamisia voimia ja momentteja. (Dong et al. 2021) Tämä mahdollistaisi esimerkiksi lentokoneen rakenteen väsymisen arviointia reaaliajassa perustuen suoraan lentodataan ja kyseisen lentokoneyksilön tekemiin lentoliikkeisiin ilman tarvetta suoralle kuormitusmittaukselle esimerkiksi voima-antureilla (Halle et al. 2014).

Edellä mainittuja menetelmiä on myös sovellettu jo varsinaisissa suunnitteluprojekteissa. Boutmedjet et al. (2019) suorittivat alustavan suunnittelun miehittämättömän ilma-aluksen (engl. *Unmanned Aerial Vehicle, UAV*) siipiprofiilille käyttämällä geneettisen algoritmin ja neuroverkon yhdistelmää. Siipiprofiilin optimointi tapahtui käyttämällä geneettistä algoritmia, johon oli sisällytetty neuroverkko korvaamaan perinteinen aerodynaaminen virtauslaskuri. Neuroverkon sisäänmenotasoon syötettiin siiven geometriset ominaisuudet kuten tyvijänne C_r (engl. *root chord*), kärkijänne C_t (engl. *tip chord*), kär-

kiväli b (engl. *wing span*) ja pinta-ala S , sekä lisäksi myös näistä johdettuja parametreja kuten sivusuhte A (engl. *aspect ratio*), kartiokkuus λ (engl. *taper ratio*) ja siipikuormitus K_S . Geometrisista ominaisuuksista johdettujen parametrien kaavat on esitelty taulukossa 2. Siipikuormituksen kaavassa esiintyvä F on koko siiven pinta-alaan kohdistuva voima. Neuroverkon ulostulotasolta haettiin arvoja nostokertoimelle C_L (engl. *lift coefficient*) ja vastuskertoimelle C_D (engl. *drag coefficient*), joiden avulla siiven suorituskykyä arvioitiin. Neuroverkon opetus suoritettiin vastavirta-algoritmilla syöttäen 256 simulatiota, jotka toteutettiin aerodynaamisella virtauslaskurilla. (Boutemedjet et al. 2019, s. 469, 471–472)

Taulukko 1. Siiven parametrien kaavat
(muokattu lähteestä Boutemedjet et al. 2019, s. 469)

Siiven pinta-ala	$S = \frac{(C_r - C_t)b}{2}$	(2)
Sivusuhte	$A = \frac{b^2}{S}$	(3)
Siipikuormitus	$K_S = \frac{F}{S}$	(4)
Siiven kartiokkuus	$\lambda = \frac{C_t}{C_r}$	(5)

Lopullisena tavoitteena itse geneettiselle algoritmilta oli profiilin optimointi ilmassaoloajan maksimoimiseksi ja vastuksen minimoimiseksi kohtauskulman ollessa nolla sekä näiden erilaisille painotetuille yhdistelmille. Tuloksena saatiin 13 erilaista siipiprofiilia, joista yksi valittiin sovitettavaksi lopulliseen ilma-alukseen. Siipiprofiilin soveltuvuus varmistettiin vielä paneelimenetelmällä sekä CFD-simulaatioilla. Tulokset näistä menetelmistä vastasivat hyvin lähelle geneettisen algoritmin ja neuroverkon arvioita siiven suorituskyvystä, ja täten geneettistä algoritmia ja neuroverkkoja hyödyntänyt menetelmä todettiin toimivaksi. (Boutemedjet et al. 2019, s. 473–474, 481)

3.2 Lentokoneen moottorijärjestelmien mallintaminen ja optimointi

Lentokoneen voimanlähteillä eli moottoreilla on hyvin suuri merkitys lentokoneen suorituskykyyn, jonka mittareina toimii esimerkiksi suuri työntövoima, vähäinen polttoaineen kulutus, pienempi lentomelu ja pienemmät päästöt. Pienempi polttoaineen kulutus mahdollistaa myös entistä pidemmän toiminta-ajan lentokoneelle. Erityisesti siviili-ilmailussa lentokoneiden kehitys on viimeisen reilun kymmenen vuoden aikana keskittynyt pitkälti uusien moottorien suunnitteluun. (Parker 2015, s. 31–32) Merkkinä tästä toimii erityisesti Airbusin ja Boeingin uusimmat kapearunkolentokoneperheet Airbus A320neo ja Boeing B737 MAX, jotka on jatkokehitelty yhtiöiden aikaisemmista Airbus A320 ja Boeing B737 -malleista. Suurimpana erona näissä uusissa lentokoneissa on nimenomaan uudet suuremmat ja tehokkaammat moottorit, jotka kykenevät valmistajiensa mukaan jopa 15–20 % polttoainesäästöihin edeltäjiinsä verrattuna. (Airbus 2022; Boeing 2022). Lisäksi tulevaisuudessa myös vaihtoehtoiset energianlähteet kuten sähkö ja vety tuovat omat haasteensa lentokoneiden voimanlähteiden suunnitteluun.

Lentokonemoottorin mallintaminen ja optimointi koneoppimisen näkökulmasta on läheisessä suhteessa aerodynaamiseen optimointiin. Tutkimusta on tehty moottoreiden suunnittelussa erityisesti korvikemallien luomisen ja niiden soveltuvuuden arvioinnin parissa (Cao & Bai 2020; Kim et al. 2020). Suunnittelun lisäksi tutkimusta on tehty erityisesti kunnossapidon näkökulmasta, ja erilaiset koneoppimiseen perustuvat ennustavat mallit ovat jo varsin yleisessä käytössä moottoreiden kunnossapidon optimoinnissa (Khan et al. 2021). Tässä työssä tarkoitus on kuitenkin pitää rajaus nimenomaan järjestelmäsuunnittelussa, joten koneoppimisen merkitystä kunnossapitoon ei tarkastella sen enempää.

Toisaalta moottorin optimoinnissa on myös selviä eroja aerodynaamiseen optimointiin. Aerodynaamisessa mallinnuksessa käytetyt mallit ovat usein melko universaaleja käyttökohteesta riippumatta (Grauer & Morelli 2014, s. 1). Lisäksi koko aerodynaamisen rakenteen riippuvuussuhteet ovat usein selvästi rajattavissa (esimerkiksi siipirunkoyhdistelmä). Lentokoneen moottori on jo äärimmäisen monimutkainen järjestelmäkokonaisuus, jossa on huomattava määrä osajärjestelmien riippuvuussuhteita. Lisäksi moottorin toimintaan vaikuttavat myös ulkoiset ohjauksjärjestelmät. Käytännössä moottorin optimointiongelmia käsitelläänkin usein tarkastelemalla yhtä komponenttia kerrallaan ja sen vaikutusta koko moottorijärjestelmään (Min et al. 2017, s. 53; Jafari & Nikolaidis 2019, s. 48).

Tulokset koneoppimismallien, kuten neuroverkkojen käytöstä lentokoneen mallintamisessa ja optimoinnissa ovat lupaavia. Cao & Bai (2020, s. 20) nostivat aihetta käsittelevässä tutkimuksessa esiin erityisesti nopeuden paranemisen neuroverkkojen avulla. Neuroverkoihin perustuvat korvikemallit suoriutuivat vastaavasta laskennasta erittäin nopeasti (noin 0.13 s) numeerisiin mallinnusmenetelmiin verrattuna (4760 s). Optimoitujen korvikemallien todettiin myös vastaavan erittäin hyvin (yli 98 %) varmuudella numeerisia mallinnusmenetelmiä. (Cao & Bai 2020, s. 19–20) Myös siirto-oppimiseen perustuvien korvikemallien on todettu olevan kilpailukykyisiä perinteisten numeeristen mallinnusmenetelmien kanssa (Min et al. 2017, s. 59).

Toisessa tutkimuksessa Bazazzadeh et al. (2011, s. 1–2, 5) tutkivat sumeaan logiikkaan perustuvan ohivirtausmoottorin ohjausjärjestelmän rakentamista hyödyntäen neuroverkkoa. Ohjausjärjestelmän toiminta perustui polttoaineen virtausnopeuden säätämiseen riippuen tehovivun asennosta. Tavoitteena oli saada moottorin suoritusarvot vastaamaan tehovivusta pyydettyjä arvoja mahdollisimman nopeasti ja tarkasti. Polttoaineen virtausnopeudella on suora vaikutus moottorin oleellisiin suoritusarvoihin kuten työntövoimaan, pyörimisnopeuteen ja turbiinin sisäänottoilman lämpötilaan. Neuroverkon avulla määriteltiin polttoaineen virtausfunktio kyseisiin suoritusarvoihin perustuen. Suoritusarvot syötettiin neuroverkolle sisäänmenotasolla ja ulostulotasolta haettiin polttoaineen virtausnopeus. Käytännössä neuroverkko siis simuloi moottoria takaperin ottaen moottorin ulostuloarvot sisään ja syöttäen moottorin sisääntuloarvon ulos. (Bazazzadeh et al. 2011, s. 1–2, 5)

3.3 Ilma-aluksen törmäyksenestojärjestelmän parantaminen

Avioniikka on myös yksi merkittävistä koneoppimistutkimuksen osa-alueista lentokonetekniikassa. Erilaiset avioniikkajärjestelmät voivat kuitenkin olla keskenään hyvinkin erilaisia, joten tässä työssä avioniikan tarkastelu keskittyy yksittäiseen järjestelmään: ilma-aluksen törmäyksenestojärjestelmään. Kyseinen järjestelmä on tällä hetkellä murroksessa ja uuteen nykyisen järjestelmän korvaavaan järjestelmään on suunnitteilla useita koneoppimista hyödyntäviä ominaisuuksia.

3.3.1 Ilma-aluksen törmäyksenestojärjestelmä yleisesti

Ilma-aluksen törmäyksenestojärjestelmä (engl. *Airborne Collision Avoidance System*, ACAS; myös *Traffic Collision Avoidance System*, TCAS) on ilma-aluksesta löytyvä järjestelmä, joka varoittaa lentäjää muista mahdollisista törmäyskurssilla olevista ilma-

aluksista (engl. *Traffic Advisory, TA*), sekä antaa myös ohjeita törmäyksen välttämiseksi (engl. *Resolution Advisory, RA*). Järjestelmä on pakollinen kaikissa Euroopassa ope-roivissa maksimilentoalennusmassaltaan 5700 kg tai enemmän painavissa, tai 19 mat-kustajaa tai enemmän kuljettavissa siviililentokoneissa. (Federal Aviation Administrati-on 2011, s. 6, 9)

Järjestelmän toimintaperiaate perustuu tutkavastaajan eli transponderin lähettämiin ja vastaanottamiin korkeus- ja sijaintitietoihin lähettyvillä olevista ilma-aluksista. Nykyinen TCAS II 7.1 järjestelmä pystyy seuraamaan 30:tä transponderilla varustettua lentoko-netta noin 55 km etäisyyteen asti. Järjestelmän sisältämät algoritmit laskevat jokaisen lähettyvillä olevan lentokoneen lentoradan, ja laskee aikaperusteisesti oman lentoko-neen ja muiden ympärillä olevien lentokoneiden etäisyyttä. Mikäli aikaperusteinen etäi-syys menee alle tietyn rajan, jotka on esitetty taulukossa 2, lähettää järjestelmä joko TA- tai RA-varoituksen lentäjälle. (Federal Aviation Administration 2011, s. 22–23)

Taulukko 2. *Yksinkertaistettu taulukko TCAS II 7.1 -järjestelmän aikaperusteisesta varoituskynnyksestä (muokattu lähteestä Federal Aviation Administration 2011, s. 23)*

Korkeus merenpinnasta (ft)	Aika mahdolliseen kohtaamiseen (s)	
	TA	RA
< 1 000 (maan pinnasta)	20	Ei sovellettavissa
1 000–2 350 (maan pinnasta)	25	15
2 350–5 000	30	20
5 000–10 000	40	25
10 000–20 000	45	30
20 000–42 000	48	35
> 42 000	48	35

Lisäksi kahden eri lentokoneen ACAS järjestelmät keskustelevat keskenään RA-tilanteissa. Esimerkiksi kaksi lentokonetta, jotka lähestyvät toisiaan samalla korkeudella vaakalennossa ja ovat törmäyskurssilla saavat erilaiset ohjeet tilanteen välttämiseksi. Toinen kone saa ohjeen nousta ja toinen kone saa ohjeen laskeutua. Järjestelmä myös laskee vaaditun nousu- tai laskunopeuden, joka ilmoitetaan lentäjälle visuaalisesti. Len-

täjän on aina lähtökohtaisesti seurattava ACAS järjestelmän antamia ohjeita, paitsi jos ohjeiden seuraaminen vaarantaisi lennon turvallisuutta muuten. Esimerkiksi mikäli lennonjohto antaa RA-varoituksesta poikkeavan selvityksen, tulee lentäjän seurata ACAS järjestelmää kyseisessä tilanteessa. (Federal Aviation Administration 2011, s. 34, 37–39)

Nykyinen TCAS II 7.1 järjestelmä omaa kuitenkin myös rajoitteita, jotka tulee ottaa huomioon erityisesti lentoliikenteen kehittyessä tulevaisuudessa. Nykyinen järjestelmä käyttää perinteisiä algoritmeja lentoratojen laskemiseen, kykenee väistöliikkeisiin vain korkeussuunnassa. Siviililentokoneisiin asennetut järjestelmät pystyvät tunnistamaan transponderilla varustettuja ilma-aluksia riippumatta ilma-aluksen tyypistä tai koosta, mutta esimerkiksi kahden miehittämättömän ilma-aluksen välistä törmäystä nykyinen järjestelmä ei pysty estämään. Lisäksi nykyinen järjestelmä vaatii muutamaa poikkeusta lukuun ottamatta ihmisen toimintaa törmäyksen välttämiseksi. Näitä ongelmia ratkaisemaan ollaankin kehittämässä uutta ACAS X järjestelmää, jonka on tarkoitus viedä nykyistä järjestelmää eteenpäin Markovin päätösprosessin ja dynaamisen ohjelmoinnin avulla. Lisäksi ACAS X:n on tarkoitus tuoda törmäyksenestojärjestelmät myös muihin ilma-alusluokkiin kuten pienkoneisiin, helikoptereihin (ACAS X_P) ja miehittämättömiin ilma-aluksiin (ACAS X_U). (Eurocontrol 2013)

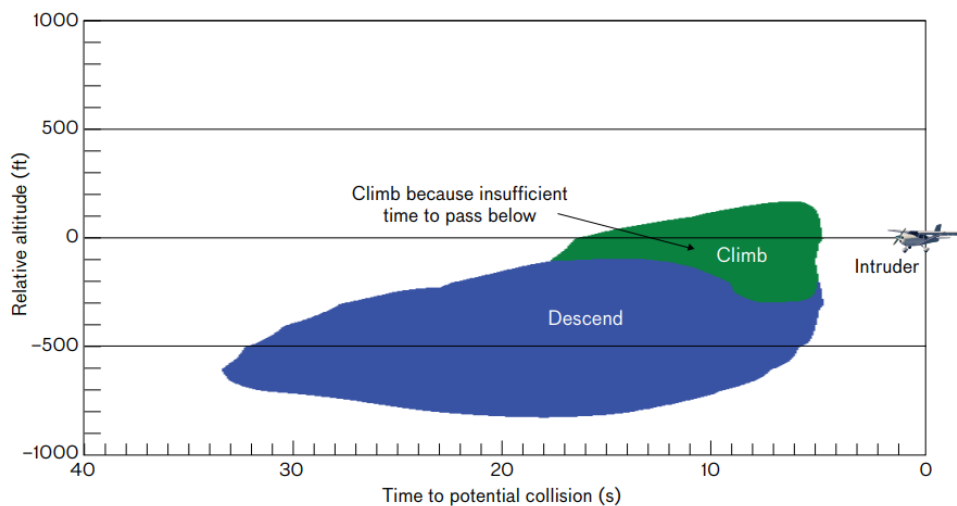
3.3.2 Koneoppivien menetelmien hyödyntäminen liikenteen törmäyksenestojärjestelmässä

TCAS 7.1 II järjestelmässä monimutkaiset toimintamenetelmät ja logiikka tuli ohjelmoida suoraan järjestelmän sisään. Ihmisen kyky käsitellä suuri määrä monimutkaisia ketjuja on kuitenkin rajallinen (Klingberg 2000), eikä nykyistä järjestelmää voida pitää optimoituna. Ilmaliikenteen lisääntyessä sekä tutka- ja satelliittiteknologian kehittyessä, lentokoneet lentävät yhä lähempänä toisiaan, mikä on johtanut lukuisiin määriin vääriä TCAS-hälytyksiä tilanteissa, joissa lentokoneet eivät ole olleet vaarassa törmätä. (Kochenderfer et al. 2012, s. 18–20; Eurocontrol 2013)

ACAS X järjestelmän suurin tavoite on pyrkiä optimoimaan TCAS 7.1 II järjestelmän päätöksentekoa, jotta se pystyisi laskemaan mahdollisia tapahtumaketjuja reaaliajassa tarkemmin. Nykyisessä ACAS X suunnitelmassa tämä on toteutettu luomalla numeeriset hakutaulukot (engl. *numeric lookup table*) käyttämällä Markovin päätösprosessia ja dynaamista ohjelmointia. Työn rajauksen vuoksi näiden menetelmien teoriaan ei tässä työssä syvennyttä sen tarkemmin, mutta menetelmät ovat vahvasti sidoksissa erityisesti koneoppivan järjestelmän vahvistusoppimiseen (Eurocontrol 2013; Vamvoudakis et al.

2021). Markovin päätösprosessi käy läpi miljoonia eri tiloja ja tilanteita ja luo näiden pohjalta jokaiselle tilalle parhaimman mahdollisen toimintamenetelmän. Nämä toimintamenetelmät sisällytetään itse järjestelmään numeeristen hakutaulukkojen muodossa, ja lentokone sijoittaa sensoreistaan saatavan datan oikeaan hakutaulukkoon reaaliajassa. (Kochenderfer et al. 2012, s. 20–26)

Kuvassa 2 on esitetty yksi hakutaulukko graafisesti, joka on luotu tilanteeseen, jossa ACAS X järjestelmän omaava lentokone nousee 7.62 m/s (1500 jalkaa minuutissa) ja ”tunkeutuja” eli törmäyskurssilla oleva lentokone lentää vaakalentoa. Pystyakselilla nähdään koneiden välinen korkeus ja vaaka-akselilla aika kohtaamiseen. Kuvasta nähdään sinisellä alue, jossa kone antaisi käskyn vähentää korkeutta ja vihreällä alue, jossa kone antaisi käskyn lisätä korkeutta. (Kochenderfer et al. 2012, s. 23–25)



Kuva 2. Graafinen esitys hakutaulukosta tilanteessa, jossa nouseva lentokone kohtaa vaakalennossa olevan lentokoneen. Pystyakselilla lentokoneiden välinen korkeus jaloissa, vaaka-akselilla aika mahdolliseen törmäykseen. Vihreällä merkitty nousevan väistökäskyn alue, ja sinisellä merkitty laskevan väistökäskyn alue. (Kochenderfer et al. 2012, s. 25)

Nykyistä ACAS X suunnitelmaa on myös lähdetty kehittämään eteenpäin syvemmillä koneoppimisen menetelmillä, kuten neuroverkoilla. Miehitämättömiin ilma-aluksiin suunniteltu ACAS X_U järjestelmä sisältäisi myös sivuttaissuuntaisia korjausliikkeitä korkeussuuntaisten korjausliikkeiden lisäksi. Tämä tosin lisää myös tarvittavien numeeristen taulukoiden määrää sekä niiden viemää tallennustilaa avioniikkajärjestelmässä. Yhdeksi ratkaisuksi tähän ongelmaan on tutkittu neuroverkkojen käyttöä, jossa neuroverkot korvaisivat numeeriset taulukot ja suorittaisivat tarvittavat laskelman törmäyksenestojärjestelmälle reaaliajassa. (Julian et al. 2019)

Koska neuroverkko toimisi osana järjestelmää, olisi sen käyttöön vaadittava muistitila huomattavasti pienempi (alle 200 MB riippuen neuroverkon koosta) verrattuna numeeristen taulukoiden vaatimaan muistitilaan (2 GB) (Julian et al. 2019, s.3–4). Toisaalta neuroverkkojen käytön turvallisuudesta osana törmäyksenestojärjestelmää käydään edelleen aktiivista keskustelua. Ensimmäiset neuroverkkoja hyödyntävät prototyypit todettiin turvallisuudeltaan riittämättömiksi, mutta menetelmää on jatkokehitetty, ja joitakin lupaavia tuloksia myös turvallisuuden paremmin takaavista neuroverkkoja soveltavista menetelmistä on saatu. Työ on kuitenkin vielä kesken, eikä yhtäkään neuroverkkoja hyödyntävää menetelmää ole vielä sertifioitu käytettäväksi törmäyksenestojärjestelmissä. (Lopez et al. 2021; Damour et al. 2021; Bak & Tran 2022)

4. KONEOPPIMISEN TULEVAISUUS LENTOKONETEKNIKASSA

Tarkastellaan lopuksi tutkimustuloksia ja mahdollisia johtopäätöksiä. Lisäksi käydään läpi kirjoittajan omaa pohdintaa aiheesta ja tutkielman suorittamisesta.

Koneoppimisen mahdollisuudet lentokonetekniikassa ovat hyvin laajat. Erilaisia käyttökohteita on useita, ja tutkimusta uusista käyttökohteista tehdään jatkuvasti. Tässä tutkielmassa tarkastelu jouduttiin rajaamaan melko tarkasti nimenomaan järjestelmien suunnitteluun. Tutkielman perusteella voidaan todeta, että koneoppiminen toi selviä hyötyjä ja mahdollisuuksia erityisesti suunnittelupuolella nopeamman mallintamisen ja optimoinnin avulla. Aiempaa tutkimusta on tehty erityisesti numeeristen menetelmien korvaamisessa erilaisilla hyvin suunnitelluilla koneoppimiseen pohjautuvilla korvikemalleilla, ja tutkimusten tulokset ovat olleet lupaavia. Koneoppivat menetelmät suoriutuivat tehtävistä huomattavasti numeerisia menetelmiä nopeammin, ja tulokset olivat hyvin lähellä numeerisilla menetelmillä saatuja tuloksia.

Lisäksi suoritettiin tarkastelu yksittäisen avioniikkajärjestelmän osalta, johon suunniteltiin koneoppimisen sisällyttämistä osana järjestelmän päätöksentekoa. Koneoppivien menetelmien katsottiin tuovan selvää hyötyä järjestelmän toimintaan parantaen sen suorituskykyä ja luotettavuutta. Toisaalta erityisesti ilmailun raskaan sääntelyn takia voi vielä mennä pitkään, ennen kuin kriittisesti lennon kulkuun vaikuttavia koneoppivia päätöksentekomalleja tullaan näkemään osana lentokoneiden järjestelmiä.

Lähitulevaisuudessa koneoppimisen hyödyntämisen käyttökohteet todennäköisesti rajoittuvatkin ulkoisiin suunnittelutehtäviin mallinnuksen ja optimoinnin parissa. Erilaisia datapohjaisia malleja on verrattain helppo ja nopea luoda, mikäli käytettävissä on vain luotettavaa dataa kattavasti. Tosin koneoppimisen käytön mahdolliset riskit tulee ottaa myös huomioon, ja sen hyödyntäminen osana suunnitteluprosessia vaatii ymmärrystä ja tarkkaa suunnittelua sen määrittelemiseksi, mitä koneoppivalla menetelmällä halutaan saavuttaa. Lisäksi koneoppivien menetelmien käyttäjän tulee ymmärtää mallin käyttäytymistä sekä pystyä arvioimaan menetelmien antamien tuloksien turvallisuutta ja luotettavuutta.

Ajan mittaan, kun tieto koneoppivista menetelmistä lisääntyy ja niiden luotettavuutta pystytään paremmin arvioimaan, tullaan varmasti näkemään koneoppimista reaaliajaj-

sa hyödyntäviä järjestelmiä osana lentokoneen järjestelmäkokonaisuutta. Tätä ennen kuitenkin erityisesti ilmailun sääntelyn täytyy pystyä vastaamaan koneoppimisen mahdollisiin ongelmakohtiin kuten vastuukysymyksiin, etiikkaan ja läpinäkyvyyden puutteeseen. Lisäksi järjestelmien käyttäjien, erityisesti lentäjien, mutta myös matkustajien, luottamus koneoppimista hyödyntäviin järjestelmiin täytyy pystyä voittamaan. (EASA 2020)

Muita mielenkiintoisia koneoppimisen aihealueita lentokonetekniikassa, joita tässä tutkielmassa ei käsitelty työn rajauksen vuoksi ovat esimerkiksi koneoppimisen hyödyntäminen lentokoneiden kunnossapidossa ja valmistuksessa. Lisäksi digitaalisten kakosten käyttö lentokonetekniikassa, ja niiden mahdollinen yhdistäminen koneoppiviin menetelmiin on yksi tähän työhön läheisesti liittyvä aihealue, jonka kuitenkin päätin jättää rajauksen ulkopuolelle. Tarkoituksena ei myöskään ollut tarkastella ihmisen korvaamista lentotehtävistä esimerkiksi koneoppimisen ja autonomisen ilmailun yhdistämisen kautta, vaan tutkia koneoppimista yhtenä työkaluna tukemassa ihmisen ja koneen välistä vuorovaikutusta.

5. YHTEENVETO

Tämän kandidaatintyön tavoitteena oli tehdä katsaus tämänhetkiseen tilaan koneoppimisen hyödyntämisestä lentokonetekniikassa, erityisesti lentokoneiden järjestelmissä sekä niiden suunnittelussa. Tarkempi perehtyminen aiheeseen osoitti, että aihe on hyvin ajankohtainen koneoppivien menetelmien suosion ollessa kasvussa. Tämän vuoksi myös tutkimuksen määrä aiheen parissa erityisesti viimeisen kymmenen vuoden aikana on lisääntynyt huomattavasti. Kiinnostus koneoppimisen hyödyntämiseen lentokonetekniikassa on suurta ja mahdolliset sovelluskohteet ovat monipuoliset.

Aiheen taustoittamiseksi työn ensimmäisessä osassa käsiteltiin koneoppimisen ominaispiirteitä sekä tiedon käsittelyä koneoppimisen näkökulmasta. Erilaisia koneoppimisen menetelmiä ja algoritmeja on useita kymmeniä, ja niille jokaiselle löytyy useita eri sovelluskohteita. Tässä työssä tarkempaan esittelyyn valittiin kaksi hyvin yleistä koneoppimisen menetelmää: geneettiset algoritmit ja neuroverkot. Lisäksi esiteltiin myös kaksi erilaista koneoppimisen kannalta hyödyllistä menetelmää tiedonkäsittelyyn liittyen: tiedonlouhinta ja olemassa olevan tiedon yhdistäminen.

Koneoppivalle järjestelmälle ominaista on sen kyky oppia sille syötettyä dataa, ja tunnistaa siitä toistuvuuksia ja riippuvuussuhteita. Koneoppivan järjestelmän oppiminen tapahtuu pääsääntöisesti kolmella tavalla: ohjatusti, ohjaamattomasti ja vahvistavasti. Eri oppimismenetelmiä käytetään riippuen koneoppivan järjestelmän käyttökohteesta ja opetukseen käytettävissä olevasta datasta. Lentoliikenne on otollinen kohde koneoppiville sovelluksille, lentoliikenteestä saatavilla olevan suuren datamäärän takia.

Työn toisessa osassa käsiteltiin tarkemmin lentokoneen eri järjestelmiä sekä koneoppimisen hyödyntämistä järjestelmien suunnittelussa. Tarkempaan järjestelmätarkasteluun valikoitui lopulta aerodynamiikan ja lentokoneen moottoreiden mallinnus ja optimointi, sekä koneoppimisen hyödyntäminen ilma-aluksen törmäyksenestojärjestelmän päätöksentekoprosessissa. Lopuksi vielä tarkasteltiin tutkimustuloksia sekä käytiin läpi yleistä pohdintaa koneoppimisen tulevaisuudesta lentokonetekniikassa perustuen tutkielman aikana tehtyihin havaintoihin.

Työssä tarkastellut tutkimuskysymykset liittyivät tällä hetkellä käytössä oleviin tai tutkimuksen alla oleviin koneoppiviin sovelluksiin lentotekniikassa, sekä sopivimpien järjestelmäsovellusten löytämiseen. Tutkimuskysymyksiin pystyttiin löytämään vastaukset,

mutta erityisesti sopivimpien järjestelmäsovellusten löytäminen vaatisi vielä kattavampaa tutkimusta eri menetelmien ja järjestelmäsovellusten vertailussa keskenään.

Koneoppimisen merkitys lentokonetekniikassa sisältää suuria mahdollisuuksia. Käyttösovellukset tällä hetkellä, ja lähitulevaisuudessa, rajoittuvat pääosin erilaisiin mallinnus- ja optimointiongelmien ratkaisemiseen. Hyviä tuloksia oltiin saatu erityisesti koneoppimisen soveltamisesta korvikemalleissa numeeristen menetelmien korvaamiseen. Neuroverkoilla ja geneettisillä algoritmeilla tehdyt sovellukset kykenivät suoriutumaan samoista mallinnus ja optimointitehtävistä huomattavasti numeerisia menetelmiä nopeammin.

Lupaavia tuloksia erityisesti simulaatioiden kautta oli myös saatu koneoppimisen sisällyttämisestä osaksi lentokonejärjestelmiä. Kuitenkin, jotta reaaliajassa ratkaisuja tekevä koneoppiva järjestelmä voitaisiin sisällyttää osaksi lentokoneen järjestelmiä, täytyy sen luotettavuus ja turvallisuus pystyä varmistamaan aukottomasti. Toistaiseksi näyttääkin siltä, ettei tällaisia erityisesti syväoppivia menetelmiä kuten neuroverkkoja soveltavia lentokonejärjestelmäsovelluksia tulla näkemään vielä aivan lähitulevaisuudessa.

Suuri osa työssä tarkastelluista tutkimuksista keskittyivät tutkimuksiin siviili-ilmailun parissa. Tutkimukset sisälsivät tuloksia sekä miehitystyistä, että miehittämättömistä ilmauksista. Työssä rajattiin tarkoituksellisesti pois muita mahdollisia koneoppimisen sovelluskohteita lentokonetekniikassa, kuten lentokoneiden kunnossapidon ja valmistuksen optimointi. Lisäksi vaikka työssä käsitellyissä tutkimuksissa käsiteltiin miehittämättömiä ilma-aluksia, ei työssä käsitelty autonomista lentoliikennettä, kuten lentäjän korvaamista lentokoneissa koneoppivilla järjestelmillä.

Koneoppimisen hyödyntäminen lentokonetekniikassa on vielä suhteellisen varhaisessa vaiheessa. Tutkimuskohteita tulevaisuuden tutkimuksia varten on useita. Erityisesti tarkemmat tutkimukset koneoppimisen sisällyttämisestä osaksi lentokoneen järjestelmiä ovat tarpeen luotettavuuden ja turvallisuuden varmistamiseksi. Myös erilaisten järjestelmien suunnittelua tukevien koneoppivien mallien tutkimista tulisi jatkaa parhaiden menetelmien löytämiseksi. Digitaalisten kaksosten ja koneoppimisen suhdetta lentokonetekniikassa voitaisiin myös selvittää tarkemmin.

Työstä saadut tulokset koneoppimisen hyödyntämisestä lentokoneiden järjestelmäsuunnittelussa vaikuttavat lupaavilta. Erilaiset käyttökohteet ovat monipuolisia järjestelmäsuunnittelusta järjestelmäsovelluksiin. Koneoppivia järjestelmiä suunnitellessa tulee ottaa huomioon menetelmän soveltuvuus sekä pyrkiä luomaan ymmärrys koneoppivan järjestelmän toiminnasta parhaimman tuloksen saavuttamiseksi.

LÄHTEET

- Ailisto, H., Heikkilä, E., Helaakoski, H., Neuvonen, A. & Seppälä, T. (2018). Tekoälyn kokonaiskuva ja osaamiskartoitus. Valtioneuvoston kanslia. Saatavissa (viitattu 4.3.2022): <https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/160925/46-2018-Tekoalyn%20kokonaiskuva.pdf?sequence=1&isAllowed=y>.
- Airbus (2022). A320neo The most successful commercial aircraft family ever. Saatavissa (viitattu 26.4.2022): <https://aircraft.airbus.com/en/aircraft/a320/a320neo>.
- Alpaydin, E. (2021). Koneoppiminen. Terra Cognita Oy.
- Anderson, G.R. & Aftosmis, M.J. (2015). Adaptive Shape Parameterization for Aerodynamic Design. NASA. pp. 56.
- Anderson, M.B. (2003). Genetic Algorithms in Aerospace Design: Substantial Progress, Tremendous Potential. Sverdrup Technology Inc.
- Bak, S. & Tran, H.-D. (2022). Neural Network Compression of ACAS Xu Early Prototype is Unsafe: Closed-Loop Verification through Quantized State Backreachability. arXiv. Saatavissa: <https://arxiv.org/pdf/2201.06626.pdf>
- Bazazzadeh, M., Badihi, H. & Shahriari, A. (2011). Gas Turbine Engine Control Design Using Fuzzy Logic and Neural Networks. International journal of aerospace engineering, pp. 1–12.
- Bertoni, A., Hallstedt, S.I., Dasari, S.K. & Andersson, P. (2020). Integration of value and sustainability assessment in design space exploration by machine learning: an aerospace application. Design Science, 6.
- Boeing (2022). The new Boeing 737 MAX Family. Saatavissa (viitattu 26.4.2022): <https://www.boeing-me.com/en/products-and-services/commercial-airplanes/737-MAX.page>
- Boutemedjet, A., Samardžić, M., Rebhi, L., Rajić, Z. & Mouada, T. (2019). UAV aerodynamic design involving genetic algorithm and artificial neural network for wing preliminary computation. Aerospace science and technology, 84, pp. 464–483.
- Brunton, S.L., Nathan Kutz, J., Manohar, K., Aravkin, A.Y., Morgansen, K., Klemisch, J., Goebel, N., Buttrick, J., Poskin, J., Blom-Schieber, A.W., Hogan, T. & McDonald, D. (2021). Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. AIAA journal, 59, no. 8, pp. 2820–2847.
- Burmester, G., Ma, H., Steinmetz, D. & Hartmann, S. (2018). Big Data and Data Analytics in Aviation. Springer International Publishing.
- Cao, D. & Bai, G. (2020). A Study on Aeroengine Conceptual Design Considering Multi-Mission Performance Reliability. Applied sciences, 10, no. 13, pp. 4668-.
- Chen, M.-S., Han, J. & Yu, P.S. (1996). Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering, 8, no. 6, pp. 866–883. Saatavissa: <https://doi.org/10.1109/69.553155>.
- Damour, M., De Grancey, F., Gabreau, C., Gauffriau, A., Ginestet, J.-B., Hervieu, A., Huriaux, T., Pagetti, C., Ponsolle, L. & Clavière, A. (2021). Towards Certification of a Reduced Footprint

ACAS-Xu System: A Hybrid ML-Based Solution. Springer International Publishing. Saatavissa (viitattu 22.4.2022): https://doi.org/10.1007/978-3-030-83903-1_3.

Dangeti, P. (2017). *Statistics for machine learning: techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R*. Packt Publishing.

Dong, Y., Tao, J., Zhang, Y., Lin, W. & Ai, J. (2021). Deep Learning in Aircraft Design, Dynamics, and Control: Review and Prospects. *IEEE transactions on aerospace and electronic systems*, 57, no. 4, pp. 2346–2368.

EASA (2020). *Artificial Intelligence Roadmap A human-centric approach to AI in aviation*. EASA. Saatavissa (viitattu 24.4.2022): <https://www.easa.europa.eu/sites/default/files/dfu/EASA-AI-Roadmap-v1.0.pdf>.

Eurocontrol (2013). *ACAS X - The future of airborne collision avoidance*. Eurocontrol. Saatavissa: <https://skybrary.aero/sites/default/files/bookshelf/2390.pdf>.

Federal Aviation Administration (2011). *Introduction to TCAS II Version 7.1*. U.S Department of Transportation. Federal Aviation Administration. Saatavissa (viitattu 27.3.2022): https://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf.

Gawron, V. (2019). *Automation in Aviation - Accident Analyses*. Mitre Saatavissa: <https://www.mitre.org/sites/default/files/pdf/pr-16-3426-lessons-lost-accident-analysis.pdf>.

Ge, Z., Song, Z., Ding, S.X. & Huang, B. (2017). Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*, 5, pp. 20590–20616.

Gerdes, M., Scholz, D. & Galar, D. (2016). Effects of condition-based maintenance on costs caused by unscheduled maintenance of aircraft. *Journal of quality in maintenance engineering*, 22, 4, pp. 394–417.

Grauer, J.A. & Morelli, E.A. (2014). *A Generic Nonlinear Aerodynamic Model for Aircraft*. NASA/Langley Research Center.

Gudivada, V., Apon, A. & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, 10, no. 1, pp. 1–20.

Hall, P., Gill, N. & Cox, B. (2020). *Responsible Machine Learning*. 1st edition. O'Reilly Media, Inc.

Halle, M., Thielecke, F. & Lindenau, O. (2014). Comparison of real-time flight loads estimation methods. *CEAS aeronautical journal*, 5, no. 4, pp. 501–513.

Hilbert, M. & Lopez, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science (American Association for the Advancement of Science)*, 332, 6025, pp. 60–65.

Jafari, S. & Nikolaidis, T. (2019). Meta-heuristic global optimization algorithms for aircraft engines modelling and controller design; A review, research challenges, and exploring the future. *Progress in Aerospace Sciences*, 104, pp. 40–53.

Jordan, M.I. & Mitchell, T.M. (2015). *Machine learning: Trends, perspectives, and prospects*. *Science (American Association for the Advancement of Science)*, 349, 6245, pp. 255–260.

Julian, K.D., Kochenderfer, M.J. & Owen, M.P. (2019). Deep Neural Network Compression for Aircraft Collision Avoidance Systems. *Journal of guidance, control, and dynamics*, 42, no. 3, pp. 598–608.

- Kääriäinen, J., Aihkisalo, T., Halén, M., Holmström, H., Jurmu, P., Matinmikko, T., Seppälä, T., Tihinen, M. & Tirronen, J. (2018). Ohjelmistorobotiikka ja tekoäly - soveltamisen askelmerkkejä. Valtioneuvoston kanslia. Saatavissa (viitattu 4.3.2022): <https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobotiikka%20ja%20tekoaly.pdf?sequence=1&isAllowed=y>.
- Khan, K., Sohaib, M., Rashid, A., Ali, S., Akbar, H., Basit, A. & Ahmad, T. (2021). Recent trends and challenges in predictive maintenance of aircraft's engine and hydraulic system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 43, 8.
- Kielitoimiston sanakirja (2022). Algoritmi. Saatavissa (viitattu 13.3.2022): <https://www.kielitoimistonsanakirja.fi/algoritmi>.
- Kim, S., Kim, K. & Son, C. (2020). Transient system simulation for an aircraft engine using a data-driven model. *Energy (Oxford)*, 196, pp. 117046-.
- Klingberg, T. (2000). Limitations in information processing in the human brain: neuroimaging of dual task performance and working memory tasks. Elsevier.
- Kochenderfer, M.J., Holland, J.E. & Chryssanthacopoulos, J.P. (2012). Next-Generation Airborne Collision Avoidance System. Saatavissa: https://www.ll.mit.edu/sites/default/files/page/doc/2018-05/19_1_1_Kochenderfer.pdf.
- Kwon, S.J. (2011). Artificial neural networks. Nova Science Publishers.
- Li, J., Bouhlel, M.A. & Martins, J.R.R.A. (2019). Data-Based Approach for Fast Airfoil Analysis and Optimization. *AIAA journal*, 57, 2, pp. 581–596.
- Lopez, D.M., Johnson, T.T., Bak, S., Tran, H.-D. & Hobbs, K.L. (2021). Neural Network Verification Methods for Closed-Loop ACAS Xu Properties. *AIAA Scitech Forum*, 4.1.2021, pp. 26.
- Min, A.T.W., Sagarna, R., Gupta, A., Ong, Y.-S. & Goh, C.K. (2017). Knowledge Transfer Through Machine Learning in Aircraft Design. *IEEE computational intelligence magazine*, 12, 4, pp. 48–60.
- Mishra, S. & Sahoo, S. (2017). Genetic Algorithm: An Efficient Tool for Global Optimization. *Advances in Computation Sciences and Technology*, 8, pp. 2201–2211.
- Ogle, R.A., Morrison, D. "Trey" & Carpenter, A.R. (2008). The relationship between automation complexity and operator error. *Journal of hazardous materials*, 159, 1, pp. 135–141.
- Parker, C. (2015). Aircraft: design, technology and safety. Nova Publishers.
- Pietikäinen, M. & Silvén, O. (2019). Tekoälyn haasteet - Koneoppimisesta ja konenäöstä tunnetekoaalyyn. Oulun yliopisto. Saatavissa (viitattu 23.3.2022): <http://jultika.oulu.fi/files/isbn9789526224824.pdf>.
- Sari, M. & Tuna, C. (2018). Prediction of Pathological Subjects Using Genetic Algorithms. *Computational and mathematical methods in medicine*, pp. 1–9.
- Shalev-Shwartz, S. & Ben-David, S. (2014). Understanding machine learning: from theory to algorithms. Cambridge University Press.
- Sivanandam, S.N. (2008). Introduction to Genetic Algorithms. Springer Berlin Heidelberg.
- Skinner, S.N. & Zare-Behtash, H. (2018). State-of-the-art in aerodynamic shape optimisation methods. *Applied soft computing*, 62, pp. 933–962.

The Federal Democratic Republic of Ethiopia Ministry of Transport (2020). Accident to the B737-8 (MAX) Registered ET-AVJ operated by Ethiopian Airlines. Saatavissa (viitattu 13.3.2022): https://reports.aviation-safety.net/2019/20190310-0_B38M_ET-AVJ_Interim.pdf.

Vamvoudakis, K.G., Wan, Y., Lewis, F.L. & Cansever, D. (2021). Handbook of reinforcement learning and control. Springer.

Wang, P. (2019). On Defining Artificial Intelligence. Journal of Artificial General Intelligence, 10, pp. 37.

Weiss, K., Khoshgoftaar, T.M. & Wang, D. (2016). A survey of transfer learning. Journal of big data, 3, 1, pp. 1–40.

Wuest, T., Weimer, D., Irgens, C. & Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. Production & manufacturing research, 4, 1, pp. 23–45.