TRAFICOM

Liikenne- ja viestintävirasto
Kyberturvallisuuskeskus

Tekoälyn soveltamisen kyber-turvallisuus ja riskienhallinta

9/2021

Sisällys

Julkaisupäivämäärä
22.10.2021

Julkaisun nimi
Tekoälyn soveltamisen kyberturvallisuus ja riskienhallinta

Tekijät
Antti Vähä-Sipilä, F-Secure Consulting
Samuel Marchal; Matti Aksela, F-Secure AI Centre of Excellence

Toimeksiantaja ja asettamispäivämäärä
Liikenne- ja viestintävirasto Traficom

Julkaisusarjan nimi ja numero	ISSN(verkkojulkaisu) 2669-8781
Traficomin tutkimuksia ja selvityksiä 9/2021	ISBN(verkkojulkaisu) 978-952-311-771-6

Asiasanat
Tekoäly, koneoppiminen, kyberturvallisuus, tietoturva, riskienhallinta, tietosuoja

Tiivistelmä

Tekoäly on entistä useammin osa tietojärjestelmiä sen tarjoamien uusien mahdollisuuksien vuoksi. Tekoälyyn ja koneoppimiseen liittyvien tietoturvariskien tunnistaminen ja hallitseminen on tärkeää, jotta tekoälyjärjestelmiä voidaan hyödyntää mahdollisimman turvallisesti ja myös saada näiden järjestelmien tuoma mahdollinen hyöty nykyistäkin laajempaan käyttöön. Tekoälyn ja koneoppimisen etiikka ja riskit ovat olleet viime vuosina suuren huomion kohteena. Eettisiä periaatteita kodifioidaan myös lainsäädäntöön ja erilaisiin käytännesääntöihin. Myös tekoälyn turvallisuuden standardointityö on aktiivista sekä yleisesti tekoälyjärjestelmille että toimialakohtaisesti. Sekä tekoälyä käyttäviltä että perinteisiltä järjestelmiltä vaaditaan selitettävyyttä, vikasietoisuutta ja tarkkailtavuutta.

Tekoälyn tietoturvallisuus ja tietosuoja lepäävät vahvasti perinteisen tietoturvallisuuden varassa. Monet ehdotetuista lähestymistavoista kannattaisikin ottaa käyttöön myös perinteisten järjestelmien kehityksessä. Tekoälyjärjestelmissä näiden ominaisuuksien saavuttaminen voi kuitenkin vaatia perinteisistä järjestelmistä poikkeavia lähestymistapoja. Tekoälyn tietoturvariskit ovat sekä systeemisiä että toteutukseen liittyviä. Systeemiset riskit liittyvät siihen, miten tekoäly toimii oikein osana muita järjestelmiä ja yhteistoiminnassa ihmisten kanssa. Toteutuksen riskit liittyvät opetusdatan hallintaan ja luottamuksellisuuteen, koneoppimismallien eheyteen ja niiden oikeaan toimintaan myös vihamielisissä tilanteissa.

Koneoppimismallit oppivat datasta. Opetusdataan vaikuttamalla on mahdollista vaikuttaa koneoppimista käyttävän järjestelmän toimintaan. Riskejä voidaan hallita tekoälyjärjestelmien kehitystyössä oikeilla teknisillä ratkaisuilla sekä tietoturva- ja tietosuoja-aktiviteeteilla. Järjestelmiä kehittävät tahot voivat löytää omat riskinsä ja hallita niitä tehokkaasti, kunhan riskien syntymekanismit on tunnettu ja hallintakeinot on tosiasiallisesti otettu osaksi kehitysprosesseja. Hallintakeinot voivat olla ennaltaehkäiseviä ja reaktiivisia ja useimmat niistä ovat samoja kuin perinteisillä järjestelmillä. Poikkeukset liittyvät esimerkiksi testausjärjestelyihin.

Selvitystä varten kerättiin näkemyksiä useista organisaatioista: Aalto-yliopiston Mobile Cloud Computing -ryhmä, Awake.ai, CSC - Tieteen ja tietotekniikan keskus, Oulun satama, Reaktor, TietoEVRY, VTT ja Wärtsilä. Näiden lisäksi näkemyksiä kerättiin muusta kahdesta organisaatiosta, jotka toimivat tietoliikenne- ja rahoitusaloilla.

Yhteyshenkilöt	**Raportin kieli**	**Luottamuksellisuus**	**Kokonaissivumäärä**
Markus Mettälä, Emma Hokkanen	suomi	Julkinen	40

Jakaja	**Kustantaja**
Liikenne- ja viestintävirasto Traficom, Kyberturvallisuuskeskus	Liikenne- ja viestintävirasto Traficom, Kyberturvallisuuskeskus

Utgivningsdatum
22.10.2021

Publikation
Cybersäkerhet och riskhantering vid tillämpning av artificiell intelligens

Författare
Antti Vähä-Sipilä, F-Secure Consulting
Samuel Marchal; Matti Aksela, F-Secure AI Centre of Excellence

Tillsatt av och datum
Liikenne- ja viestintävirasto Traficom

Publikationsseriens namn och nummer **Traficoms forskningsrapporter och utredningar** 9/2021	ISSN(webbpublikation) 2669-8781 ISBN(webbpublikation) 978-952-311-771-6

Ämnesord
artificiell intelligens, maskininlärning, informationssäkerhet, cybersäkerhet, artificiell intelligens och etik, riskhantering, data

Sammandrag

Artificiell intelligens är allt oftare en del av olika informationssystem, tack vare de nya möjligheter AI medför. Det är viktigt att kunna identifiera och hantera risker för informationssäkerhet i anknytning till artificiell intelligens och maskininlärning, så att man kan utnyttja system med artificiell intelligens på ett så tryggt sätt som möjligt, och samtidigt göra det möjligt att utnyttja dessa system i en allt större utsträckning. Etik och risker kring artificiell intelligens och maskininlärning har under de senaste åren varit föremål för mycket uppmärksamhet. Etiska principer kodifieras också i lagstiftning och i olika uppförandekoder. Det pågår aktivt arbete med att standardisera säkerhetspraxis kring artificiell intelligens, både i allmänhet för system med artificiell intelligens och specifikt inom olika branscher. Både system med artificiell intelligens och traditionella system måste vara förklarbara, feltoleranta och observerbara. Informationssäkerheten och dataskyddet för artificiell intelligens baserar sig i dagsläget i hög grad på traditionell informationssäkerhet. Många av de föreslagna sätten att närma sig problemen skulle det löna sig att ta i bruk också i utvecklingen av traditionella system. För att dessa egenskaper ska uppnås i system med artificiell intelligens kan det dock krävas att man tar sig an problematiken på annorlunda sätt än för traditionella system. Riskerna med informationssäkerhet för artificiell intelligens är kopplade till både systemet och själva genomförandet. Systemiska risker handlar om hur artificiell intelligens kan fungera korrekt som en del av andra system och i växelverkan med människor. Risker i samband med genomförandet har att göra med hur träningsdata hanteras och hålls konfidentiell och med dataintegriteten i maskininlärningsmodellerna och att de fungerar korrekt även i fientliga situationer. Maskininlärningsmodeller lär sig av data. Genom att påverka träningsdata kan man påverka funktionen hos ett system som använder sig av maskininlärning. Man kan hantera riskerna under utvecklingen av systemen med artificiell intelligens genom tekniska lösningar och genom olika informationssäkerhets- och dataskyddsaktiviteter. De som utvecklar systemen kan också själva hitta risker och hantera dem effektivt, förutsatt att riskernas uppkomstmekanismer är kända och metoderna för att hantera riskerna faktiskt har införlivats i utvecklingsprocessen. Hanteringsmetoder kan vara förebyggande och reaktiva, och i de flesta fall är de likadana som för traditionella system. Det finns undantag till exempel i fråga om testarrangemangen.

Kontaktperson	**Språk**	**Sekretessgrad**	**Sidoantal**
Markus Mettälä, Emma Hokkanen	finska	offentlig	40

Distribution	**Förlag**
Transport och kommunikationsverket Traficom, Cybersäkerhetscentret	Transport och kommunikationsverket Traficom, Cybersäkerhetscentret

Date of publication
22.10.2021

Title of publication
Cyber security and risk management in the application of AI

Author(s)
Antti Vähä-Sipilä, F-Secure Consulting
Samuel Marchal; Matti Aksela, F-Secure AI Centre of Excellence

Commissioned by, date
Liikenne- ja viestintävirasto Traficom

Publication series and number	ISSN(online) 2669-8781
Traficom Research Reports 9/2021	ISBN(online) 978-952-311-771-6

Keywords
AI, machine learning, information security, cyber security, AI ethics, risk management, data

Abstract

Due to the new opportunities provided by AI, it has become an increasingly common part of information systems. Identifying and managing information security risks related to AI and machine learning is important in order to be able to utilise AI systems as securely as possible. They are also crucial in trying to utilise any benefits from AI systems even more comprehensively than currently. The ethics and risks related to AI and machine learning have been a hot topic in recent years. Ethical principles will also be codified in legislation and various codes. The standardisation work for AI security in both AI systems in general and branch-specifically is carried out actively. Systems that use AI, as well as those that do not, need to be explainable, fault-tolerant and monitorable. The information security and data protection of AI rely heavily on traditional information security. Many of the proposed approaches should, indeed, also be implemented in the development of traditional systems. Achieving these features in artificial intelligence systems can, however, require approaches that differ from those of traditional systems. Information security risks related to AI are both systemic and connected to implementation. Systemic risks are related to how AI works correctly as part of other systems and in cooperation with people. The risks connected to implementation have to do with the management and confidentiality of training data as well as the integrity of machine learning models and their correct operation, including in hostile situations. Machine learning models learn from data. The operation of a system using machine learning can be influenced by influencing the training data. In AI system development work, risks can be managed by using the correct technical solutions as well as information security and data protection activities. The parties developing these systems can identify their own risks and manage them efficiently, provided that the birth mechanisms of the risks are known and the management measures have been implemented in the development processes in practice. The management measures may be pre-emptive and reactive, and most of them are the same as with traditional systems. Exceptions are connected to test configurations, for example.

Contact person	**Language**	**Confidence status**	**Pages, total**
Markus Mettälä, Emma Hokkanen	Finnish	public	40

Distributed by	**Published by**
Transport and Communications Agency, Cyber Security Centre Finland	Transport and Communications Agency, Cyber Security Centre Finland

Tekoäly ja koneoppiminen käsitteinä

Tekoäly on koneiden toimintaa, joka muistuttaa jollakin tavalla ihmisen älykkyyttä. Tekoäly ei nykyisellään ole yleistettävää, käsitemalleihin pohjautuvaa älykkyyttä, vaan se ratkaisee sovelluskohtaisia ongelmia. Tekoälyn toteutusmenetelmiä on monia, joista koneoppiminen on ominaisuuksiensa vuoksi tietoturvamielessä kiinnostavin. Koneoppimisjärjestelmien sydämenä on koneoppimismalliksi kutsuttu tietorakenne, joka muodostetaan eli opetetaan esimerkkien eli oppimisdatan pohjalta. Nämä erityispiirteet tuovat järjestelmiin uusia riskejä, vaikka suurin osa toteutuksen riskeistä onkin samoja kuin perinteisissä järjestelmissä.

Tekoäly

Tekoäly-termiä (artificial intelligence, AI) käytetään sellaisesta koneen toiminnasta, joka näyttää ihmisen tai muun eläimen älykkäältä käytökseltä. Tällä hetkellä tekoäly rajoittuu kapeisiin sovelluksiin, kuten esimerkiksi jonkin ilmiön havaitsemiseen tietomassasta tai datan generointiin esimerkkien pohjalta. Vaikka se suoriutuukin usein näistä rajatuista tehtävistä ihmistä paremmin, vaikkapa nopeammin ja väsymättä, tekoäly ei vielä kuitenkaan ole ”yleistä” tai ”todellista” älykkyyttä. Tekoäly ei tarkoita sitä, että järjestelmä osaisi yhtäkkiä tehdä johtopäätöksiä, joita sitä ei ole suunniteltu tekemään.

Suurimpia eroja nykyisen tekoälyn ja inhimillisen älykkyyden välillä on kyky siirtää opittua ongelmanratkaisukykyä kokonaan uuteen viitekehykseen yleiskäsitteiden kautta. Nykyiset tekoälyjärjestelmät saattavat epäonnistua täysin, mikäli ne saavat syötteen, jota niiden luokittelujärjestelmä ei pysty käsittelemään oikein. Tämä on samalla yksi merkittävimmistä riskilähteistä nykyisissä tekoälyjärjestelmissä. Esimerkiksi nykyinen koneoppimiseen perustuva tekoälyn opetusprosessi ei opeta koneelle käsitteitä vaan siinä luodaan mahdollisesti hyvinkin monimutkainen tapa päätyä havainnoista päätökseen. Järjestelmä muodostaa nämä päätöspolut havaintojen perusteella matemaattisten mallien kautta. Päätöspolut eivät välttämättä perustu samoihin syötteiden ominaisuuksiin, mihin ihmisaivot päätöksensä perustaisivat, eivätkä ne myöskään synnytä ”maalaisjärkeä” tai ”yleistä elämänkokemusta” (1).

Toinen ero ”yleisen” älykkyyden ja nykyisen tekoälyn välillä on järjestelmä, jolla äly kytkeytyy ympäröivään maailmaan. Ihmisaivoihin on kiinteästi integroitunut edistynyt sensomotorinen järjestelmä, jonka avulla ne voivat reagoida älykkäästi tavoilla, joilla konepohjaiset sensori- ja aktuaattorijärjestelmät eivät tällä hetkellä voi. (1) Ainakin ihmisen tietoisuudella on myös tunteita ja muita mielentiloja sekä käsitys mielen teoriasta. Nykyisillä tekoälyjärjestelmillä ei näitä piirteitä esiinny.

Jos ja kun tekoäly joskus saavuttaa ”yleisen” tai ”todellisen” älykkyyden tason, tekoälyn tietoturva ja eettiset kysymykset muodostuvat huomattavasti nykyistä monimutkaisemmiksi. Tässä katsauksessa rajataan tekoälyn riskin arviointi nykyisiin, rajattujen käyttötarkoitusten tekoälyjärjestelmiin.

Tekoäly-termi ei itsessään kerro, miten tekoäly on teknisesti toteutettu. Rajatusti älykkäältä näyttäviä toimintoja voidaan toteuttaa monilla eri lähestymistavoilla. Nykyään tekoälyjärjestelmän toteutusmenetelmänä on usein *koneoppiminen*, jolla tarkoitetaan monenlaisia tapoja, joilla järjestelmä voi oppia syötedatasta sen sijaan, että päätöspolut määriteltäisiin valmiiksi. Suosittelujärjestelmät käyttävät usein tilastollisia menetelmiä, joihin ei välttämättä sisälly koneoppimista. Esimerkiksi aikaisempaa ostokäyttäytymistä voidaan verrata muiden asiakkaiden käyttäytymiseen ja asiakkaat voidaan segmentoida niin, että tilastollisesti voidaan ennustaa, mitkä muut tuotteet asiakasta saattaisivat kiinnostaa. Joissakin järjestelmissä päätöksenteko taas voidaan määritellä selkeiden sääntöjen kautta, jolloin voidaan käyttää logiikkaohjelmointia ja symbolista laskentaa. Vaikka tällainen järjestelmä onkin lähempänä ”perinteistä” tietokoneohjelmaa, sen käyttötapaus voi silti saada koneen vaikuttamaan älykkäältä.

Sääntely tulee koskemaan tekoälyä toteutustavasta riippumatta: Euroopan komission ehdotus tekoälyasetukseksi (2) tunnistaa laajan kirjon tekoälyn toteutustapoja.

Käytännön tekoälyjärjestelmät tarvitsevat myös tapoja saada syötteitä ja toisaalta toimia päätösten pohjalta. Kun tekoälyjärjestelmää tarkastellaan tietoturvan kannalta, esimerkiksi sensorit, aktuaattorit (toimilaitteet[1]) ja niiden ja tekoälyn väliset tietoliikenneyhteydet on otettava tarkastelun piiriin.

Esimerkki sensorien moninaisuudesta on Liikenne- ja viestintäministeriön autonomisten alusten sääntelystä kertova raportti (3), jonka kappale 2.3 luettelee alusten koneoppimisjärjestelmän syötelähteitä.

Koneoppiminen

Tietoturvamielessä kiinnostavin tekoälyn toteutusmenetelmien luokka on *koneoppiminen* (machine learning, ML). Tällä hetkellä monet tekoälysovellukset perustuvat neuroverkkopohjaisiin[2] koneoppimismalleihin, mutta kaikki koneoppimismallit eivät ole neuroverkkoja. Koneoppimismalli muodostuu mallin opetuksessa käytetyn datan perusteella. Koneoppimisjärjestelmä opetetaan tekemään päätöksiä antamalla järjestelmälle toistuvasti syötteitä ja säätämällä päätösmallia, kunnes järjestelmä antaa toivotun lopputuloksen (tarkemmin ks. kappale Koneoppimisjärjestelmän elinkaari). Järjestelmä oppii syötteiden piirteistä, mutta järjestelmän suunnittelijat eivät välttämättä pysty, osaa tai halua vaikuttaa siihen, mitkä syötteiden piirteistä vaikuttavat päätökseen tai miten malli kunkin piirteen tulkitsee. Koneoppimisjärjestelmillä voidaankin luoda myös sovelluksia, jotka pystyvät käsittelemään dataa ilman että datan rakennetta pitää erikseen tarkasti kuvata - esimerkiksi kuvia, puhetta ja luonnollista kieltä.

Tekoälyjärjestelmien riskeistä moni liittyy erityisesti siihen, että niiden päätöksenteon perusteiden selittäminen on vaikeaa ja koneoppimismalleille on suhteellisen helppo luoda vihamielisiä (*adversarial*) syötteitä, jotka saavat ne tekemään vääriä tai yllättäviä päätöksiä.

Erityisesti englanninkielisiä termejä ”AI” ja ”ML” käytetään usein sekaisin. Kaikki tekoälyjärjestelmät eivät kuitenkaan ole koneoppimisjärjestelmiä ja kaikki automaattiset päätökset eivät ole tekoälyjärjestelmien tekemiä. Esimerkiksi verotuksessa tietokone tekee laskelmia, jotka perustuvat täysin käsin määriteltyihin sääntöihin. Vaikka päätökset ovatkin käytännössä automatisoituja, ne eivät ole ”älykkäitä” eivätkä ne perustu koneoppimiseen. Toisaalta koneoppimisen menetelmiä voitaisiin käyttää esimerkiksi veronkierron ja veropetosten havaitsemiseen. Tässä katsauksessa olemme pyrkineet valitsemaan mahdollisimman tarkan termin aina, kun viittaamme tekoälyn ja/tai koneoppimisen riskeihin.

Tämä katsaus painottaa erityisesti koneoppimisjärjestelmien riskejä. Monet riskeistä esiintyvät muillakin tavoilla toteutetuissa tekoälyjärjestelmissä, mutta koneoppimisessa esiintyy sellaisia riskejä, joita muissa toteutustavoissa ei ole juuri sen takia, että opetusdatalla on niin suuri merkitys mallin toimintaan.

1 Aktuaattoreita eli toimilaitteita ohjataan sensorien tiedon perusteella. Esimerkiksi lämpömittarin lukemien perusteella voitaisiin ohjata moottoria, joka avaa ja sulkee kuumavesilinjan venttiiliä. Moottori on tässä tapauksessa toimilaite.

2 Neuroverkko on rakenne, joka on saanut innoituksensa eläimen hermosoluista ja niiden välisistä yhteyksistä. Synteettiset neuroverkot sisältävät usein kerroksellisesti järjestettyjä neuroneita, jotka ovat yksinkertaisia, toisiinsa kytkettyjä laskentayksiköitä.

Yleisimmät riskit, uhat ja väärinkäyttömahdollisuudet

Useimmat tekoälyjärjestelmien riskeistä ovat samoja kuin perinteistenkin järjestelmien riskit ja eettisetkin kysymykset seurailevat yleisiä tekniikan etiikan teemoja. Aiheesta on viime vuosina kirjoitettu paljon. Yleisimmissä tekoälyn käyttötarkoituksissa ratkaisu eettisiin kysymyksiin tulee löytymään suoraan lainsäädännöstä. Mitä poikkeuksellisempi käyttötarkoitus, sitä todennäköisemmin systeemitason pohdinta on tarpeen.

Systeemiset riskit liittyvät siihen, miten tekoäly toimii oikein osana muita järjestelmiä. Tekoälyjärjestelmän ja ihmisen tai yhteiskunnan rajapinnat vaativat erityishuomiota, koska ne ovat usein monimutkaisia ja kumpikin osapuoli voi olla omalla tavallaan vaikeasti analysoitavissa. Merkittävimmät tekoälyjärjestelmille ominaiset riskit ja eettiset kysymykset liittyvät nimenomaan koneoppimisen käyttöön, erityisesti koneoppimismallin läpinäkyvyyteen ja selitettävyyteen. Tekoälyjärjestelmiltä vaaditaan myös vikasietoisuutta, joka on tärkeä piirre myös perinteisissä järjestelmissä. Nämä kaikki ominaisuudet yhdessä määräävät, onko järjestelmä hallittavissa - ja voidaanko sen oikeasta ja eettisestä käytöstä varmistua.

Tekoälyn etiikka

Tekoälyn eettiset kysymykset liittyvät sen korkean tason systeemisiin riskeihin. Ylimmällä tasolla tekoälyn käytön eettiset kysymykset eivät eroa muista tekniikan käytön etiikan kysymyksistä. Eettisyyden arviointi perustuu aina kulloinkin vallitseviin arvoihin, ja arvot muuttuvat ajan myötä. Tekoälyn eettisissä pohdinnoissa on myös joskus näkyvissä, että tekoälyn kyvyt oletetaan suuremmaksi kuin mitä ne tällä hetkellä ovat.

Eettisen arvioinnin tarkoitus tietoturvanäkökulmasta on pystyä välttämään ja hallitsemaan tekniikan tuomia riskejä. Jotkin näistä riskeistä ovat samankaltaisia kuin ne, jotka voitaisiin välttää tietoturvan teknisillä toteutusmenetelmillä. Esimerkiksi koneoppimisjärjestelmä saattaa syrjiä henkilöryhmää joko siksi, että se on opetettu datalla, jossa on syrjiviä piirteitä, tai koska mallin eheyteen pyritään vaikuttamaan hyökkäämällä (ks. kappale Koneoppimismallin ja opetusdatan eheys). Riskienhallinnan kannalta keskittyminen ainoastaan hyökkäyksiin voi johtaa siihen, että samaan lopputulokseen johtavat systeemiset riskit jäävät käsittelemättä.

Useimmat eettisen arvioinnin taustalla vaikuttavat arvot on osittain kirjattu EU:n perusoikeuskirjaan, perustuslakiin ja eurooppalaisiin sekä kansallisiin lakeihin ja asetuksiin sekä jopa yksityisten toimijoiden menettelyohjeisiin. Nämä antavat yksilöille tiettyjä oikeuksia ja vapauksia. Tekoälyn eettinen arviointi voidaan aiemmin tunnetussa käyttötapauksessa useimmiten tehdä pelkästään laillisuus- ja sääntöjenmukaisuusarviointina. Joissakin tapauksissa tekoälyn vaikutus voi olla hyvin sovelluskohtainen ja sovellusalue uudenlainen, jolloin valmiista tulkinnoista ei suoraa ratkaisua vielä löydy. Tällöin apua ja inspiraatiota voi hakea esimerkiksi Euroopan perusoikeusviraston tekoälyraportista (4) ja sen kattavista viiteluetteloista.

Yksi ydinkysymys on, mihin tekoälyä saa ylipäänsä käyttää. Saksan liittovaltion liikenteen ja digitaalisen infrastruktuurin ministeriön alainen automatisoidun ja verkotetun ajamisen etiikkakomissio julkaisi vuonna 2017 raportin (5), jonka mukaan monimutkaisia eettisiä päätöksiä ei saa tehdä eikä ohjelmoida etukäteen. Tekoälyjärjestelmä ei esimerkiksi saisi päättää elämän ja kuoleman kysymyksistä. Luonnos Euroopan unionin tekoälyasetukseksi (2) kieltää tekoälyn käytön joissakin tapauksissa, jotka liittyvät tiedostamattomaan vaikuttamiseen, henkilöryhmän ominaisuuksien hyväksikäyttöön, sosiaaliseen pisteyttämiseen tai biometrisiin etätunnistusjärjestelmiin julkisissa tiloissa.

Eettisten ohjeiden asettamisen ongelma näkyy tekoälyasetuksen saamassa kritiikissä, jonka mukaan se on samaan aikaan liian lavea ja liian spesifinen. Asetusehdotusta on kritisoitu siitä, että sen määritelmä tekoälylle on liian lavea (6). Toisaalta on kritisoitu myös sitä, että asetus kieltää epäeettisiksi mielletyt käyttötarkoitukset vain tekoälysovelluksilta eikä myös perinteisiltä tietojärjestelmiltä.

Viime vuosien aikana tekoälyn ja koneoppimisen etiikkaa on käsitelty runsaasti. Kirjallisuustutkimukset vuosilta 2019 (7) ja 2020 (8) löysivät vastaavasti 84 ja 36 dokumenttia, joissa ehdotettiin eettisiä periaatteita tai ohjeita. Dokumentteja ovat julkaisseet sekä valtiolliset että yksityisen ja kolmannen sektorin toimijat.

Yksityisyyden suoja ja tietosuoja ovat tekoälyn etiikkaa käsittelevissä dokumenteissa usein toistuvia teemoja. Tämä ei ole yllättävää, sillä tekoäly perustuu usein data-analyysiin. Usein esille tulevat reiluuden, ihmisarvon ja syrjimättömyyden näkökulmat ovat samoja oikeuksia ja vapauksia kuin mitä henkilötietolainsäädäntö pyrkii takaamaan. Tietosuojavaikutusten arviointi ja tekoälyn riskien tarkastelu onkin jo monissa organisaatioissa yhdistetty (ks. kappale Palvelumuotoilun ja käyttötapausten määrittämisen taso).

Tekoälyn eettiset kysymykset systeemisten riskien analyysinä

Tekoälyn eettisiä kysymyksiä lähestytään eri lähteissä eri tavoin. Kuvan 1. malli pyrkii jäsentämään keskustelua tekoälysovellusten eettisten kysymysten tarkastelusta osana riskianalyysiä. Uudenlaiset sovelluskohteet vaativat tarkastelun laajentamista kuvassa oikealle.

Kuva 1: Malli tekoälysovelluksen eettisten kysymysten tarkasteluun osana riskianalyysiä

Kuvan oikeassa laidassa keskustelu perustuu arvoihin ja pyrkii vastaamaan siihen kysymykseen, mihin ja miten tekoälyä saa käyttää. Kysymykset keskittyvät esimerkiksi tekoälyjärjestelmän ja ympäröivän maailman yhteistoimintaan, sen vaikutuksiin käyttäjiin ja muihin yksilöihin, yhteiskuntaan ja ympäristöön. Nämä kysymykset ovat useammin sovelluskohtaisia. Dokumentoituja suomalaisia esimerkkejä tämän tyyppisestä analyysistä ovat Aurora AI -ohjelman etiikkatyöryhmän väliraportti (9), joka pui tekoälyn käytön vaikutuksia julkisten palveluiden käytön optimointiin sekä liikenne- ja viestintäministeriön raportti (3), jossa käsitellään autonomisten alusten tekoälyn eettisiä vaatimuksia esimerkiksi sensoreiden ja navigoinnin suhteen.

Tuotekehityksessä kuvan oikean laidan teemoista tulisi keskustella mieluiten tuotehallintaprosessin aikaisessa vaiheessa, esimerkiksi palvelumuotoilun ja liiketoiminta-aihioiden luomisen yhteydessä tai tietosuoja-asetuksen kuvaaman tietosuojavaikutusten arvioinnin (*data protection impact assessment,* DPIA) yhteydessä (sovittamisesta tuoteturvallisuusprosesseihin (*Security Development Lifecycle*, SDLC), ks. kappale Palvelumuotoilun ja käyttötapausten määrittämisen taso).

Kaavion vasemmassa reunassa kysymys on järjestelmän käytännön hallittavuudesta. Hallittavuus syntyy järjestelmän toiminnan läpinäkyvyydestä (*transparency*), tarkasteltavuudesta (*observability*) sen elinkaaren yli sekä vikasietoisuudesta (*robustness* tai *resilience*, joskus myös *reliability*), joka on järjestelmän kyky toimia jatkuvasti sille tarkoitetulla tavalla. Hallittavuuteen liittyvät kysymykset voidaan usein helpommin yleistää sovellusalueriippumattomiksi, eikä keskustelu näiden osalta aina vaadi arvopohdintaa. Tämä tekee näiden ohjeiden kirjoittamisesta helpompaa. Tuotekehityksessä vasemman reunan teemoja voidaan tuoda lähemmäksi toteutuksen suunnittelua ja toteutusta, esimerkiksi osana teknistä uhkamallinnusta (*threat modeling*); sen roolista tuotekehityksessä ks. kappale Arkkitehtuurin ja toteutuksen taso.

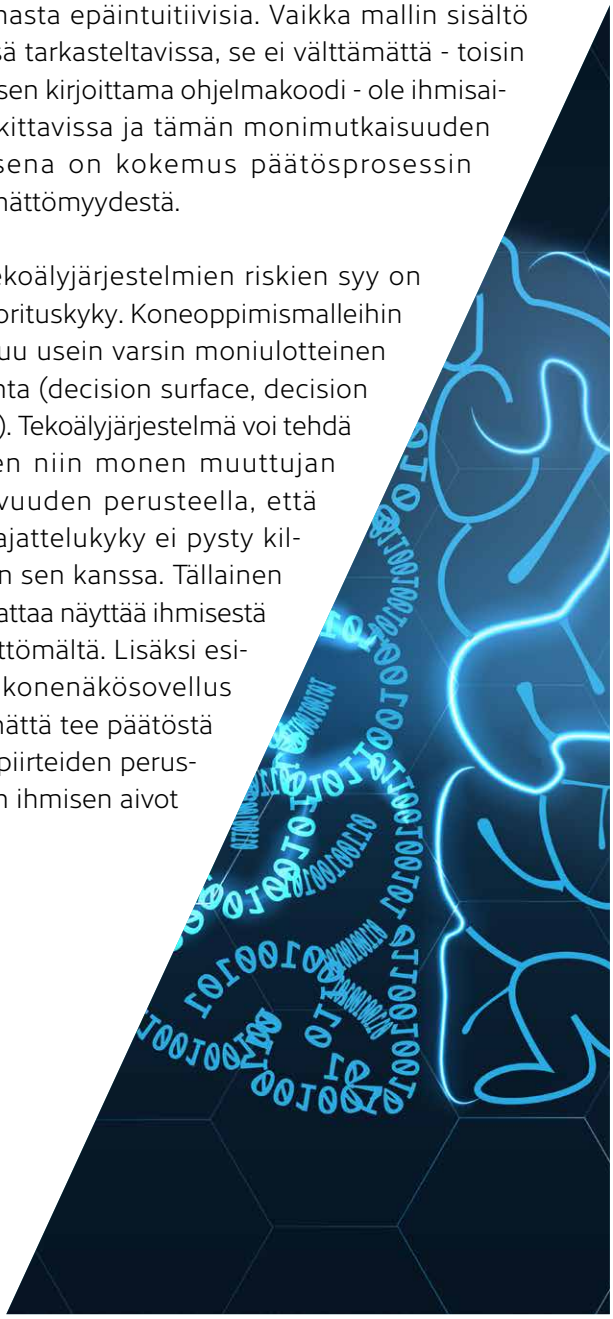
Eräät eettiset ohjeet luettelevat jopa toteutusvaatimuksia. Esimerkiksi suositukset sekä EU:sta ja Yhdysvaltain puolustusministeriöstä (10) ja (11), joskin hieman eri termein, vaativat riskianalyysin, riskien hallintamenetelmien toteutuksen sekä vikasietoisuuden testauksen, ks. kappale Testauksen ja tuotantoonviennin taso.

TEKOÄLYN JA KONEOPPIMISEN SYSTEEMISET RISKIT

Koneoppimisjärjestelmien tekemien päätösten hankala selitettävyys (*explainability*) on yksi merkittävistä syistä, miksi tekoälyjärjestelmiä pidetään riskialttiimpina kuin ”perinteisiä” tietojärjestelmiä. Päätöksen perusteet saattavat olla koodattuna esimerkiksi neuroverkon neuronien keskinäisiin yhteyksiin ja painoihin, eikä ihmisellä välttämättä ole kykyä hahmottaa, miten tällainen järjestelmä tuottaa tietyillä syötteillä tietyn tuloksen. Ilmiönä tässä ei ole mitään uutta - jopa ihmisen kirjoittamiin algoritmeihin perustuvat tietojärjestelmät näyttävät usein läpinäkymättömiltä laatikoilta, erityisesti jos ohjelmakoodi muistuttaa spagettia ja dokumentaatio puuttuu.

Koneoppimisjärjestelmien menetelmille on tyypillistä, että järjestelmä on oppinut valtavan määrän parametreja datasta. Päätökseen vaikuttavat parametrit ja niiden keskinäiset yhteydet voivat myös olla ihmisen näkökulmasta epäintuitiivisia. Vaikka mallin sisältö on sinänsä tarkasteltavissa, se ei välttämättä - toisin kuin ihmisen kirjoittama ohjelmakoodi - ole ihmisaivojen tulkittavissa ja tämän monimutkaisuuden seurauksena on kokemus päätösprosessin läpinäkymättömyydestä.

Toinen tekoälyjärjestelmien riskien syy on niiden suorituskyky. Koneoppimismalleihin muodostuu usein varsin moniulotteinen päätöspinta (decision surface, decision boundary). Tekoälyjärjestelmä voi tehdä päätöksen niin monen muuttujan ja ulottuvuuden perusteella, että ihmisen ajattelukyky ei pysty kilpailemaan sen kanssa. Tällainen päätös saattaa näyttää ihmisestä selittämättömältä. Lisäksi esimerkiksi konenäkösovellus ei välttämättä tee päätöstä samojen piirteiden perusteella kuin ihmisen aivot tekisivät.



Tekoälyjärjestelmien hallittavuus

Jotta tekoälyjärjestelmää ja sen vaikutuksia - negatiivisia tai positiivisia - voidaan hallita, järjestelmältä edellytetään ennustettavuutta ja tarkasteltavuutta. Järjestelmä, joka toimii odottamattomasti tai näennäisen satunnaisesti, ei ole hallittavissa.

Useimmat eettiset ohjeet edellyttävät tekoälyjärjestelmältä vikasietoisuutta (*robustness*, joskus suomeksi myös *jykevätekoisuus*, englanninkielisessä tietosuoja-asetuksessa *resilience*). Esimerkiksi EU:n korkean tason tekoälyasiantuntijaryhmän ohjeet (10) vaativat ”teknistä vikasietoisuutta ja turvallisuutta” ja ”varasuunnitelmaa”. Vikasietoista järjestelmää on vaikea saada toimimaan odottamattomalla tavalla, se sietää vääriä syötteitä ja sen virhetilanteet on hyvin määritelty. Vikasietoisuus on yleistettävissä mihin tahansa tekoälyjärjestelmään sovelluskohteesta riippumatta.

Vikasietoisuuden pettämisellä voi olla monenlaisia seurauksia. Se voi johtaa hengen, terveyden tai ympäristön turvallisuuden *(safety*[3]) riskeihin tai tietoturva- ja tietosuojariskeihin. Tyypillisesti vikasietoisessa järjestelmässä on turvallinen tila, johon järjestelmä siirtyy havaittaessa virhe.

Vaikka järjestelmä toimisi odotetulla tavalla, se voi silti tehdä virheitä. Koneoppimisjärjestelmissä sataprosenttinen erehtymättömyys ei ole edes realistinen tavoite. Kaikkia mahdollisia yllättäviä käytönaikaisia skenaariota ei usein ole mahdollista edes opettaa järjestelmälle, ja yleistyvyysongelma tekee tästä koneoppimisen kannalta usein perinteisiä järjestelmiä haastavamman. Vaikka tekoälyjärjestelmän toiminta olisi hyvin määritelty alueella, josta on olemassa dataa, sen käyttäytyminen tutun datan ulkopuolella voi olla hyvinkin yllättävää. Olennaista vikasietoisuudessa on, että virheiden määrä ja järjestelmän tarkkuus pysyy hyväksyttävällä ja ennakkoon määritellyllä tasolla. Vikasietoisuuden pitäisi kattaa sekä järjestelmän hyväntahtoiset ja vahingossa tapahtuvat käyttövirheet, että pahantahtoiset hyökkäykset.

Vikasietoisuuden käsite ei ole rajoittunut tekoälyjärjestelmiin. Perinteisissäkin järjestelmissä erityisesti syötteiden käsittelyyn ja virhetilanteisiin liittyy usein heikkouksia, jotka aiheuttavat sivuvaikutuksia, joita hyökkääjä voi käyttää hyväksi. Vikasietoisuuden testaus (*robustness testing*) on vakiintunut tietojärjestelmien testauksen osa-alue. Useimmiten kuitenkin

3 Sekä *safety* että *security* kääntyvät suomeksi termiksi *turvallisuus*. Tässä katsauksessa on käytetty englanninkielistä termiä eron selkeyttämiseksi silloin, kun termeillä on sekoittumisen vaara.

vaatimukset ja lainsäädäntö määrittelevät tietoturvan luottamuksellisuuden, eheyden ja saatavuuden kautta, ja vikasietoisuus nähdään yleisempänä laadullisena vaatimuksena.

Järjestelmän vikasietoisuuden tulisi säilyä koko sen elinkaaren yli. Esimerkiksi eräs tekoälysuositus Yhdysvaltain puolustushallinnosta (11) edellyttää ”jatkuvaa riskien arviointia, jatkuvaa valvontaa ja testausta, jotta nähdään, vastaako järjestelmä edelleen käyttötarvettaan”. Järjestelmien elinkaaren syklistä luonnetta on avattu kappaleessa Koneoppimisjärjestelmän elinkaari.

Toinen usein toistuva teema eettisissä ohjeissa on läpinäkyvyyden (*transparency*) ja selitettävyyden (*explainability*) vaatimus. Vaikkakin nämä liittyvät eniten aiemmin mainittuihin systeemisiin riskeihin, selitettävyys ja läpinäkyvyys ovat tärkeitä myös poikkeamien havaitsemiselle, hallinnalle ja tutkimukselle. Esimerkiksi (11) toteaa, että tekoälyjärjestelmän toiminnan läpinäkyvyys saattaa auttaa onnettomuustutkintaa. Erityisesimerkkinä läpinäkyvyystarpeesta on autonomisen ajamisen raportti (5), jonka mukaan järjestelmissä, joissa ihminen ja kone vuorottelevat, on aina oltava selvää, kumpi kulloinkin oli ohjaimissa.

Läpinäkyvyyden tulisi myös kattaa koko elinkaari. Esimerkiksi EU:n korkean tason tekoälyasiantuntijaryhmän ohjeet (10) vaativat, että järjestelmien tulisi olla ”jäljitettäviä”, esimerkiksi niiden suunnitteluprosessien ”riittävän ymmärryksen” suhteen. Tekoälyasetusehdotus (2) vaatii jäljitettävyyttä lokitietojen kautta ja vaatii suuririskisiltä järjestelmiltä kaikkien tekoälylle tehtyjen syötteiden lokitusta.

Autonomisen ajamisen raportti (5) antaa mahdollisuuden tulkintaan, että järjestelmän toimittajalle saattaa siirtyä vastuuta järjestelmän toiminnasta. Tällöin tekoälyjärjestelmä nähtäisiin itse asiassa luojansa (ohjelmoijansa, opettajansa) agenttina - ei siis käyttäjänsä apuvälineenä tai itsellisenä olentona. Jos järjestelmän toimittaja tämän vuoksi joutuu puolustelemaan järjestelmänsä toimintaa esimerkiksi oikeudessa, läpinäkyvyysvaatimuksesta syntyy myös vastuuriskin hallintakeino.

Läpinäkyvyys muuttuu entistä monimutkaisemmaksi hajautetussa järjestelmässä, jossa data ja jopa päätökset saattavat tulla useilta eri komponenteilta, joilla kullakin saattaa olla toisistaan eriävä alkuperä. Antaako järjestelmävalmistaja tekoälynsä tehdä päätöksiä muiden valmistajien sensoridatan pohjalta?

Esimerkiksi autonomisissa ajoneuvoissa reunaprosessointi[4] voi mahdollistaa sensoridatan keruun usealta datan tuottajalta eli joukkoistamisen ja jakamisen muille ajoneuvoille. Joukkoistaminen voi lisätä turvallisuutta, koska ajoneuvojen näkökenttä voi laajentua ympäröivien ajoneuvojenkin sensoreihin. Sensoridatan alkuperästä ja eheydestä varmistuminen muuttuu kuitenkin monimutkaiseksi, ja se nostaa esille kysymyksiä datan omistajuudesta.

4 Reunaprosessointi (*edge computing*) tarkoittaa laskennan suorittamista lähellä datalähteitä sen sijaan, että laskenta suoritetaan esimerkiksi kaukana sijaitsevalla palvelimella. Jos pilvipalvelu itsessään sijaitsee ”kaukana”, esimerkiksi toisessa maanosassa, käsitteellisesti ”pilven reunalla” tapahtuva laskenta voi tapahtua vaikkapa laitteen välittömässä läheisyydessä tai verkkoyhteyden tarjoajan tiloissa.

Tekoälyn ja koneoppimisen tietoturvan sääntely ja standardointi

Tekoälyn sääntely ja standardointi liittyy läheisesti tietosuojan sääntelyyn, onhan esimerkiksi koneoppiminen datalähtöistä. EU:n tekoälyasetusehdotus on merkittävä tulevaisuuden sääntelyn lähde. Tietoturvan osalta suurin osa perinteisten järjestelmien tietoturvasääntelystä ja -standardoinnista on suoraan sovellettavissa tekoälyjärjestelmiin, mutta joillakin aloilla on katsottu tarpeelliseksi ottaa kantaa nimenomaan tekoälyn ja erityisesti koneoppimisen erityispiirteisiin. Koska tekoäly on autonomisen liikenteen merkittävä mahdollistaja, safety-tyyppiset turvallisuusriskit ohjaavat tekoälyn tietoturvastandardointia.

Euroopan komission tekoälyasetusehdotus

Tätä kirjoitettaessa syksyllä 2021 kaikkein puhutuin tulevaisuuden sääntelyn väline on Euroopan komission ehdotus yhtenäistetyiksi tekoälyn säännöiksi (2). Koska kyseessä olisi asetus, siitä tulisi ehdotuksen tultua hyväksyttyä sovellettavaa lainsäädäntöä kaikissa EU:n jäsenmaissa. Ehdotus voi muuttua huomattavasti sen käsittelyn edetessä.

Tämänhetkinen ehdotus kieltäisi joitakin tekoälyn sovelluskohteita ja määrittelisi osan sovelluksista ”suuririskisiksi”. Suuririskisille järjestelmille vaadittaisiin riskienhallintajärjestelmä. Lisäksi niiden datanhallinnalle, testaukselle, dokumentaatiolle, jäljitettävyydelle, läpinäkyvyydelle ja luotettavuudelle asetettaisiin erityisiä vaatimuksia.

Nykyisen ehdotuksen 15. artikla käsittelee kyberturvallisuutta. Tekoälyjärjestelmille olisi tehtävä ”vararatkaisut”, kuten ”varasuunnitelmat tai vikavarmistussuunnitelmat”, ja niiden olisi ”kestettävä asiaankuulumattomien ulkopuolisten tahojen yritykset muuttaa järjestelmän käyttöä tai suorituskykyä hyödyntämällä järjestelmän haavoittuvuuksia”. Resitaalin 51 mukaan vaatimukset kattaisivat myös tekoälyjärjestelmän käyttämän taustainfrastruktuurin. Jos järjestelmän tekoälyosuus on sertifioitu EU:n kyberturvallisuuden sertifiointijärjestelmässä (12), 15. artiklan vaatimusten katsottaisiin täyttyvän.

Teknisistä hyökkäyksistä koneoppimismallia kohtaan nykyinen asetusehdotus nostaa esille lähinnä mallin eheyteen liittyvät hyökkäykset: datan myrkytyksen, mallinväistöhyökkäykset ja huonosti tehdyt mallit. Näitä hyökkäyksiä on tarkemmin avattu kappaleessa Koneoppimismallin ja opetusdatan eheys.

EU:n yleinen tietosuoja-asetus

Koska monet tekoälyjärjestelmät käsittelevät henkilötietoja, EU:n yleisellä tietosuoja-asetuksella (GDPR) (13) on huomattava vaikutus tekoälyn tietoturvasääntelyyn.

Tietosuoja-asetuksen 22. artikla antaa yksilöille joitakin oikeuksia, jos he ovat automaattisen päätöksenteon kohteina, mikäli päätöksellä on oikeus- tai muita merkittäviä vaikutuksia. Vaikka automaattinen päätöksenteko voikin olla sallittua, yksilöillä on oikeus saada ihminen mukaan päätöksentekoon. Automaattinen päätöksenteko ei välttämättä tarkoita tekoälyä, vaan se kattaa myös ”perinteiset” tietojärjestelmät.

Henkilötietojen käsittelyn tietoturva on määritelty tietosuoja-asetuksen 32. artiklassa. Tekoälysovellus voi käsitellä tietoja useassa eri vaiheessa. Esimerkiksi kuvan 1 järjestelmässä opetus- ja validointidatan keruu ja tallennus, opetus ja järjestelmän käyttö saattavat kaikki päätyä tämän artiklan alaisuuteen.

Rekisterinpitäjän täytyy luottamuksellisuuden, eheyden, saatavuuden ja vikasietoisuuden lisäksi varmistua myös henkilötietoon pääsevien henkilöiden käyttäytymisestä. Tämä sisältää tekoälyjärjestelmien suunnittelijat, jotka joutuvat käsittelemään henkilötietoja

järjestelmän ja sen mallien kehittämiseksi. Käytännössä tutkiva data-analytiikka sisältää runsaasti kokeilevaa tutkimusta ja algoritmisuunnittelua, joten todisteellisen pääsynhallinnan suunnitteleminen saattaa olla vaikeaa.

Tietosuoja-asetuksen näkökulmasta myös henkilötiedoilla opetettu koneoppimisen malli saattaa itsessään olla henkilötieto. Vanhan tietosuojadirektiivin 29. artiklalla perustetun tietosuojatyöryhmän (Article 29 Working Party) anonymisointitekniikkamielipiteen (14) kohta A.3 käsittelee tiedon yleistämistä anonymisoinnin välineenä. Vaikkakin koneoppimisen malli on yksinkertaista taulukkoa monimutkaisempi, se on kuitenkin käsitteellisesti verrattavissa siihen. Tilannetta voi tulkita niin, että henkilötiedoilla opetettu koneoppimisen malli ei ole välttämättä anonyymi vaan mahdollisesti pseudonyymiä dataa. Lisäksi esimerkiksi syväoppimiseen pohjautuvat kielimallit voivat oppia hyvinkin yksityiskohtaista informaatiota, jolloin malli saattaa sisältää henkilötietoja jopa alkuperäisessä muodossaan. Sovelluskohteesta riippuen on siis mahdollisesti tarpeen anonymisoida opetusdata ennen kuin malli opetetaan.

Tekoälyn riskienhallinnan standardointi

YLEISET STANDARDIT

ISO ja IEC ovat perustaneet yhteisen teknisen komiteansa alakomitean ISO/IEC JTC 1/SC 42 standardisoimaan tekoälyn käyttöä. Alakomitean tekoälyn luotettavuuden työryhmä (WG) 3 työskentelee useiden etiikkaan, riskiin ja vikasietoisuuteen liittyvien hankkeiden parissa. Tätä kirjoitettaessa kesällä 2021 alakomitea on ilmoittanut työstävänsä seuraavia standardeja (suomenkieliset nimet epävirallisia käännöksiä):

- ISO/IEC 23894, Riskienhallinta
- ISO/IEC 24028, Katsaus tekoälyn luotettavuuteen
- ISO/IEC 24029, Neuroverkkojen vikasietoisuuden arviointi
- ISO/IEC 24368, Katsaus eettisiin ja yhteiskunnallisiin kysymyksiin
- ISO/IEC 5469, Toiminnallinen turvallisuus (safety) ja tekoälyjärjestelmät

Euroopan unioni ylläpitää ICT-alueen standardoinnin jatkuvaa suunnitelmaa. Suunnitelmaan kuuluu verkkosivusto, jolla on hyvä tekoälyn standardisoinnin tilan yleiskatsaus (15). Sivulla on kattava tekoälyn etiikkaan ja tietoturvaan liittyvien yleisstandardien ja standardointiaktiviteettien lista.

MAALIIKENNE

YK:n Euroopan talouskomission (UNECE) ajoneuvosääntelyn harmonisoinnin kansainvälisen foorumin (WP.29[5]) kyberturvallisuusohjeiden (16) on nimenomaisesti kerrottu koskevan autonomisia ajoneuvoja. WP.29:n sääntely vaatii kyberturvallisuuden hallintajärjestelmän ja luettelee useita teknisiä riskejä, jotka on vähintään otettava huomioon. Luetelluista riskeistä harvat liittyvät tekoälyyn, mutta esimerkiksi väärennetyn sensoridatan riskit voivat luonnollisesti vaikuttaa niiden perusteella toimivien tekoälyjärjestelmien toimintaan.

ISO/SAE 21434 (17) on maantieajoneuvojen kyberturvallisuusstandardi. Standardissa ei ole erityisiä vaatimuksia nimenomaan autonomisille ajoneuvoille, mutta ”autonominen ajaminen väärään paikkaan” on annettu esimerkkinä ”vakavasta” tietoturvariskin vaikutuksesta. Standardin vaatimukset kattavat autonomisen auton järjestelmistä miltei kaikki, joissa on sähköä ja jonkinlaista tiedonsiirtoa tai -käsittelyä, joten tekoälyjärjestelmä, sensorit, toimilaitteet ja näiden väliset väylät ovat sen piirissä. Standardi vaatii kattavan tietoturvapolitiikan, riskienhallintamallin, uhkamallinnuksen ja vaatimusten validoinnin, mutta se ei ota kantaa toiminnallisuuteen muutoin kuin esimerkkien kautta.

5 UNECE WP.29 ja Euroopan tietosuojaneuvoston edeltäjä Article 29 Working Party (A29WP) ovat samankaltaisesta nimestään huolimatta eri toimielimiä.

MERENKULKU

Merenkulun alalla Kansainvälisellä merenkulkujärjestöllä (IMO, International Maritime Organization) on useita tekoälyyn liittyviä hankkeita, joista autonomisten pinta-alusten (MASS, Maritime Autonomous Surface Ships) osa-alue on kyberturvallisuusmielessä olennaisin. Autonomisiin aluksiin liittyvien kokeilujen ohjeistuksessa (18) tietoturvaan viitataan hyvin yleisellä tasolla. Tekoälyn osalta huomiota kiinnitetään koneiden ja ihmisten yhteistoimintaan ja virhetoimintojen hallintaan. IMO on myös julkaissut merenkäynnin kyberturvallisuuden ohjeet (19). Ohjeistus määrittelee riskienhallintajärjestelmän ja viittaa myös automaatioon, mutta ei varsinaisesti ota erikseen kantaa tekoälyyn.

Euroopan meriturvallisuusvirasto (EMSA, European Maritime Safety Agency) on julkaissut raportin autonomisten alusten riskeistä ja sääntelystä (20). Suomessa Liikenne- ja viestintäministeriön raportti (3) on käsitellyt sääntelyä autonomisten alusten näkökulmasta.
Merenkulun luokituslaitoksilla on omia standardejaan, jotka saattavat ottaa kantaa tekoälyjärjestelmiin, mutta esimerkiksi luokituslaitos DNV-GL:n luokitussääntöjen (21) kyberturvallisuusvaatimukset pohjautuvat IEC 62443-sarjaan (22), joka on tarkoitettu teollisuusautomaation turvallisuuteen. Ne eivät ota tekoälyn erityispiirteitä esille, vaikka toki kattavat autonomisetkin alukset.

ILMAILU

Lentoliikenteessä eurooppalainen ilmailustandardointifoorumi EUROCAE on perustanut työryhmän WG-114 (23) standardoimaan tekoälyn käyttöä ilmailussa. Standardointi tulee kattamaan muun muassa ilmailussa käytettävien tekoälyä hyödyntävien tuotteiden turvallisuusriskien arvioinnin ja järjestelmien sertifioinnin.

Euroopan unionin lentoturvallisuusvirasto EASAn tekoälytiekartta (24) erittelee tekoälyn ja nimenomaisesti koneoppimisen riskejä ilmailussa ja nimeää luottamuksen lähteiksi koneoppimisen oikeellisuuden varmistamisen (*learning assurance*), mallien selitettävyyden ja *safety*-turvallisuusriskien hallinnan, ja on julkaissut neuroverkkojen oppimisen oikeellisuuden varmistamisesta erillisen raportin (25).

Eurooppalainen ilmailutekoälyn korkean tason työryhmä on julkaissut raportin tekoälyn käytöstä ilmailussa (26). Vaikka kyseessä ei ole normatiivinen dokumentti, sen kappale 7 antaa hyvät suuntaviivat sille, mihin tekoälyn turvallisuuden osalta tullaan kiinnittämään huomiota.

MUU SEKTORIKOHTAINEN SÄÄNTELY

Esimerkkinä autonomisen liikenteen ulkopuolelta Yhdistyneiden kuningaskuntien terveys- ja sosiaaliministeriö on julkaissut ohjeistusta datapohjaisille terveyteen liittyville järjestelmille (27). Tekoälyyn liittyvä ohjeistuksen osa liittyy algoritmiseen läpinäkyvyyteen ja selitettävyyteen. Ohjeistuksen tietoturvavaatimukset ovat yleisiä vaatimuksia.

Tekoäly- ja koneoppimisjärjestelmien tietoturvariskit

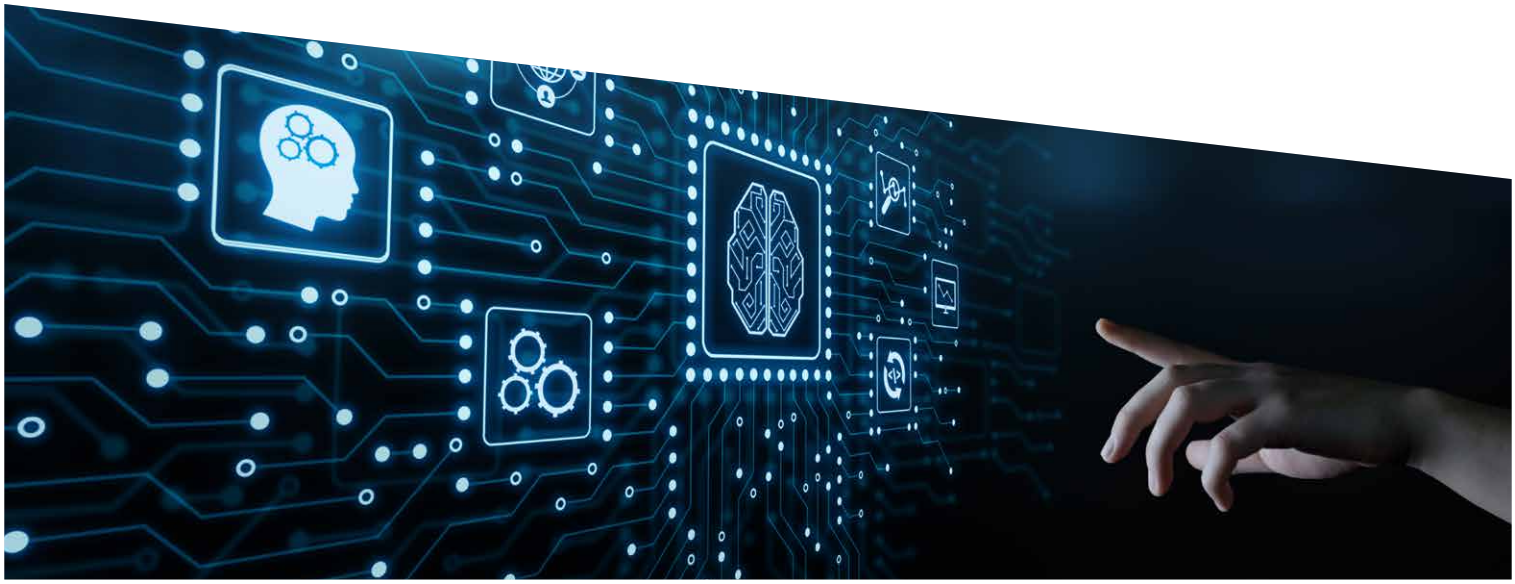
Suurin osa tietoturvariskeistä ja niiden hallintamenetelmistä on tekoälyjärjestelmissä samoja kuin perinteisissäkin järjestelmissä. Koneoppimisjärjestelmissä opetusdatan tulkinta, koneoppimismallien sisältämät tiedot, ennalta opetettujen mallien käyttö ja opetusprosessit ovat perinteisistä järjestelmistä poikkeavia osa-alueita ja vaativat erityishuomiota. Näiden osa-alueiden riskejä voidaan kuitenkin edelleen jäsentää perinteisen luottamuksellisuuteen, eheyteen ja saatavuuteen perustuvan tietoturvan kolmijaon kautta. Koneoppimisjärjestelmien osalta erityisiä teknisiä riskejä on ennen kaikkea opetusdatan ja koneoppimismallin eheydessä sekä hajautetun järjestelmän osien saatavuudessa.

Keskeiset tekniset riskit ovat koneoppimismallin myrkyttyminen väärällä datalla, mallin varastaminen, mallin väistäminen vihamielisellä syötteellä ja sensoreiden ja päätöksenteon välisen kommunikaation epäonnistuminen. Näiden riskien vaikutus riippuu koneoppimisen käyttötarkoituksesta ja niiden vaikutukset ovat usein sidoksissa jonkin toisen riskin vaikutuksiin. Osa riskeistä on luonteeltaan tietoturvariskejä, jotkin voivat olla puhtaammin henkilötietojen käsittelyyn liittyviä tietosuojariskejä.

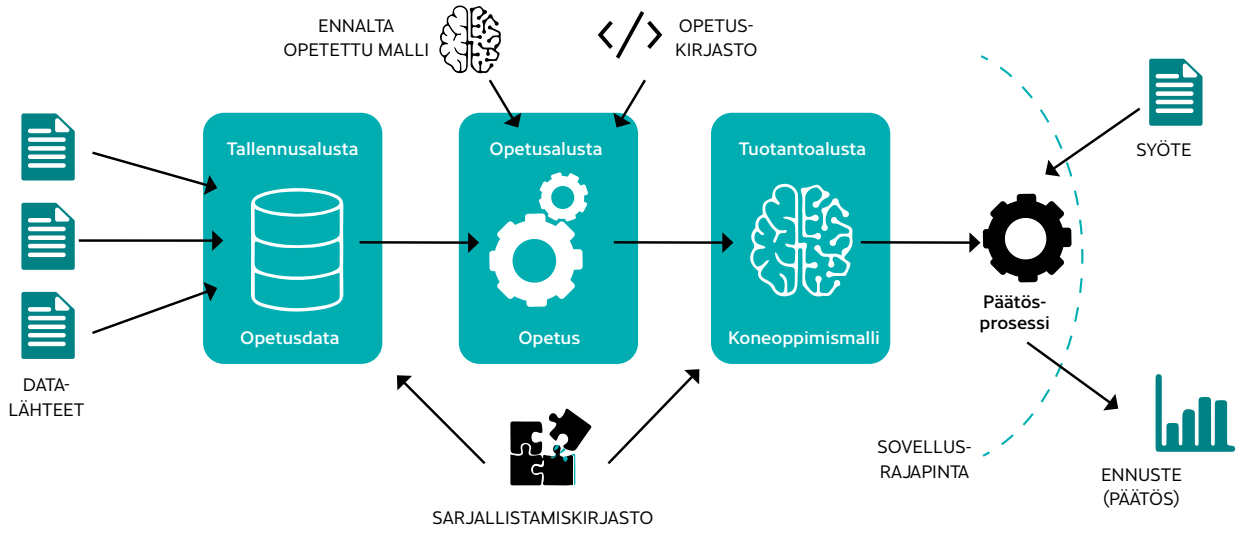
Hyökkäyspinta

Järjestelmän tietoturvariskejä voidaan tarkastella hyökkäyspinnan[6] (*attack surface*) kautta. Hyökkäyspinta koostuu kaikesta järjestelmän toiminnallisuudesta, jonka kanssa hyökkääjä voi toimia ja vaikuttaa järjestelmän toimintaan. Tyypillisessä tietojärjestelmässä hyökkäyspinta koostuu rajapinnoista, kuten käyttöliittymästä ja sovellusrajapinnoista, ja kaikista paikoista, joissa hyökkääjä pääsisi kosketuksiin siirrettävän tai tallennettavan tiedon tai ajettavan koodin kanssa.

Tekoälyjärjestelmien hyökkäyspinta on pääosin samanlainen kuin perinteisten tietojärjestelmienkin. Seuraavalla sivulla oleva kuva esittää yksinkertaistetun koneoppimisjärjestelmän hyökkäyspinnan läpi järjestelmän elinkaaren. Koneoppimisen erityispiirteiden tarkastelun mahdollistamiseksi kuvaan on valittu nimenomaan koneoppimisella toteutettu tekoälyjärjestelmä.



6 Hyökkäyspinta-termiä ei pidä sekoittaa aiemmin mainittuun koneoppimisjärjestelmien päätöspintaan.



Kuva 2: Koneoppimisjärjestelmän hyökkäyspinta

Kuvassa oikealla näkyvät koneoppimismallin tekemät päätökset. Malli on osa järjestelmän käytönaikaista toiminnallisuutta. Mallille annetaan syötteitä, tyypillisesti sovellusrajapinnan eli kutsurajapinnan kautta (merkitty kuvassa katkoviiva), ja se tuottaa tuloksia pyytäjälle.

Kuvassa on hyvin yksinkertainen järjestelmä. Oikeat järjestelmät ovat usein paljon monimutkaisempia tietovirtoineen. Liikenne- ja viestintäministeriön autonomisten alusten sääntelyä käsittelevän raportin (3) kappaleessa 2.1 on havainnollinen esimerkkikuva järjestelmistä ja tietovirroista, joita autonomisesti navigoiva alus tarvitsee.

Ennen kuin koneoppimismallia voidaan käyttää, se on opetettava. Opetuksen aikana mallille annetaan opetussyötteitä ja mallin parametreja säädetään niin, että malli alkaa tuottaa odotettuja vastauksia. Tätä prosessia kuvataan kuvan keskimmäisessä osassa. Opettamista voidaan helpottaa rakentamalla malli ennalta opetetun mallin päälle. Osana opetusta malli myös validoidaan ja testataan erillisillä, näitä varten varatuilla dataryhmillä. Koneoppimismallia voidaan myös opettaa tai validoida uudelleen myöhemmin. Kuvaan on merkitty datan käsittelyssä käytetty sarjallistamiskirjasto [7]sekä opetukseen käytetty opetuskirjasto, jotka ovat käytännössä ulkoisia ohjelmakoodiriippuvuuksia.

Opettamiseen ja validointiin tarvittava data saadaan kuvan vasemmassa reunassa kuvatuista datalähteistä ja tallennetaan jonkinlaiseen tallennuspaikkaan.

Kuvan 2 osat voidaan jakaa kolmeen kategoriaan:

1. Osat, joita esiintyy sekä ”perinteisissä” että koneoppimista käyttävissä järjestelmissä. Vaikka näissä osissa ei ole mitään koneoppimisen kannalta erityistä, niiden tietoturva on tekoälyjärjestelmän osalta edelleen kriittistä. Näitä osia ovat:
   - Ulkoiset ohjelmakirjastot ja työkalut, joita käytetään esimerkiksi opetukseen ja datan käsittelyyn
   - Ohjelmisto- ja tallennusalustat (ml. pilvipalvelut), joita käytetään esimerkiksi tiedon tallennukseen ja koodin ajamiseen
   - Kyselyrajapinta, jonka kautta tekoälyjärjestelmälle tehdään pyyntöjä

2. Osat, joita esiintyy sekä ”perinteisissä” että koneoppimista käyttävissä järjestelmissä, mutta joiden luonne (esimerkiksi datan muoto tai tulkinta) ja tätä myöten riskien yksityiskohdat ovat koneoppimisjärjestelmissä erilaisia:
   - Koneoppimismallin syötteet, jotka vastaavat perinteisen järjestelmän syötteitä
   - Päätökset, jotka vastaavat perinteisen järjestelmän tuloksia
   - Koneoppimismalli itsessään, joka tuottaa syötteiden pohjalta päätöksen
   - Datalähteet

3. Ainoastaan koneoppimisjärjestelmissä esiintyvät osat, joihin ”perinteisten” järjestelmien tietoturvan analyysi ei välttämättä tarjoa valmiita välineitä:
   - Datan tulkinta opetus- ja validointidataksi
   - Ennalta opetetut mallit
   - Opetus- ja validointiprosessit, myöskin ajonaikainen oppiminen järjestelmän päätöksiin käytettävästä datasta

7 Sarjallistaminen tarkoittaa tietokoneen muistissa olevan tietorakenteen muuttamista merkkisarjaksi, joka voidaan tallentaa esimerkiksi tiedostoon ja lukea sieltä taas muistiin.

Koneoppimisen erityispiirteet

Suurin ero perinteiseen ohjelmistoon on, että koneoppimismalli oppii opetusdatastaan. Tällä on neljä olennaista seurausta perinteiseen tietojärjestelmään verrattuna, jotka on otettava huomioon koneoppimisjärjestelmän tietoturvan suunnittelussa.

Ensinnäkin, koska malli opetetaan opetusdatalla, koneoppimismalli sisältää jotakin informaatiota opetusdatasta, ja tätä informaatiota vuotaa myös mallin antamiin päätöksiin. Tämä tarkoittaa sitä, että koneoppimismallia ja sen päätöksiä voidaan mahdollisesti käyttää oppimisdatan ja datalähteiden luottamuksen rikkomiseen, vaikka opetusdata itsessään olisikin hyvin turvattu. Konkreettinen, joskin ääriesimerkki tästä on tapaus, jossa Internetistä löytyvällä tekstimateriaalilla opetettu GPT-2-malli tuottaa tuloksissaan kokonaisia lainauksia oppimisdatasta (28). Tätä riskiä on kuvattu tarkemmin kappaleessa Koneoppimismallin luottamuksellisuus.

Toiseksi, koska mallin toiminta opitaan opetusdatan perusteella, opetusdataa, datalähteitä tai ennalta opetettuja malleja (tai näiden opetusdataa tai datalähteitä) manipuloiva hyökkääjä pystyy vaikuttamaan myös lopulliseen malliin. Mikäli opetusdata ei ole enää eheää tai sen alkuperästä ei ole varmuutta, myöskään mallin eheydestä tai alkuperästä ei ole varmuutta. Näitä riskejä on kuvattu tarkemmin kappaleessa Koneoppimismallin ja opetusdatan eheys.

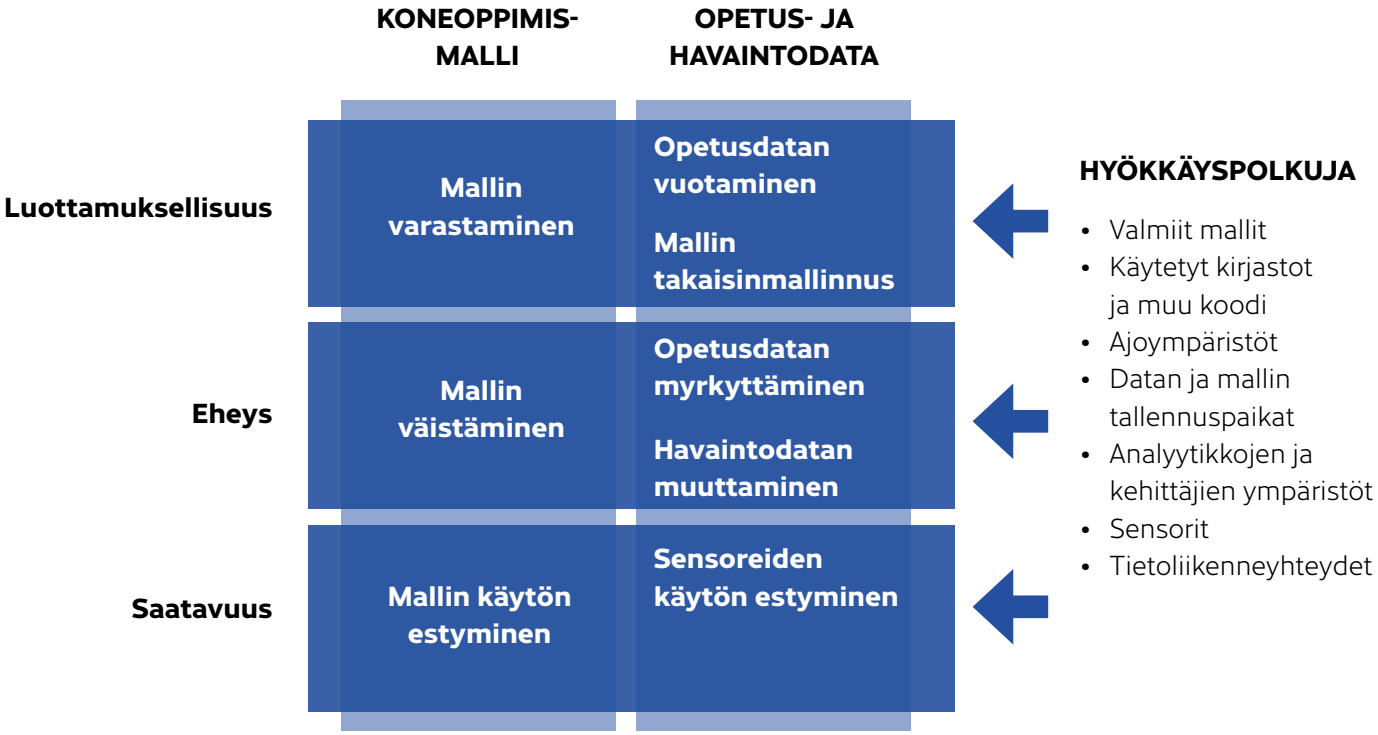
Kolmanneksi, koska koneoppimismallin sisäistä toimintaa ei yleisesti ottaen pystytä selittämään samalla tasolla kuin sarjaa yksinkertaisia loogisia sääntöjä, ei myöskään ole mahdollista varmistaa mallin oikeellisuutta annettua säännöstöä vasten. Koneoppimismallien validointi voi antaa tilastollista näkemystä siitä, toimiiko malli odotetulla tavalla, mutta validoinnilla ei voida todistaa sen oikeellisuutta kaikille mahdollisille syötteille. Tämän vuoksi myös esimerkiksi ennalta opetettujen mallien oikeellisuuden varmistaminen on vaikeaa, mikä heijastuu järjestelmän testauksen haasteisiin (ks. kappale Koneoppimisjärjestelmän elinkaari). Kuten kaikessa muussakin ohjelmistotuotannossa, toimitusketjuhyökkäys on mahdollinen ja mallien luonteen vuoksi vaikeasti havaittavissa.

Neljänneksi, mallin syötteiden oikeellisuutta on hankala todentaa. Tämä pätee sekä opetuksessa että mallin varsinaisessa käytössä käytettyihin syötteisiin. Syy siihen, miksi koneoppimista ylipäänsä käytetään, on usein siinä, että se on käyttökelpoinen syötteille ja ilmiöille, joita on vaikea kuvata määrämuotoisesti. Esimerkiksi ”puhelinnumero” on helppo kuvata syntaktisesti ja sille voidaan rakentaa syötetarkastus, mutta koneoppimisjärjestelmien syötteet voivat tyypillisesti olla vaikkapa ”pikseleiden väriarvoja, jotka esittävät kokonaisuutena ihmisen kasvokuvaa”, josta järjestelmän tulisi sitten tunnistaa henkilö. Tällaista syötettä on mahdoton kuvata yksinkertaisella säännöstöllä. Tämän vuoksi syötteiden rajapintoihin on vaikea toteuttaa syötteentarkastusta, joka havaitsisi esimerkiksi hyökkääjän manipuloimaa kuvadataa. Tämän tekee vielä haastavammaksi se, että emme välttämättä edes tiedä, mitä piirteitä kuvadatasta malli käyttää päätöksensä pohjana.

Riskit

Koneoppimiseen pohjautuvan tekoälyjärjestelmän tietoturvariskejä voidaan tarkastella luottamuksellisuuden, eheyden ja saatavuuden (nk. CIA-malli, confidentiality, integrity, availability) kautta. Tarkastelu täytyy ulottaa sekä koneoppimismalliin itseensä että mallin opetusdataan.

Kuva 3: Koneoppimisjärjestelmän riskejä. havainnollistaa pääasiallisia koneoppimiseen liittyviä riskejä.



Kuva 3: Koneoppimisjärjestelmän riskejä.

KONEOPPIMISMALLIN LUOTTAMUKSELLISUUS

Koneoppimismallien opettamiseen ja niiden opetusdatan keräämiseen kuluneet investoinnit saattavat tehdä mallista itsestään liiketoiminnallisesti arvokkaan. Esimerkiksi hyvän kuvan- tai tekstintunnistusmallin kehittäminen vaatii osaamista, opetusdataa ja runsaasti laskentakapasiteettia. Hyvin opetettu ja toimiva malli saattaa itsessään myös olla kilpailuetu.

Koneoppimismallin luottamuksellisuuden murtuminen - esimerkiksi mallin varastaminen - vertautuu immateriaaliomaisuuden (esimerkiksi liikesalaisuuden) varastamiseen, ja varastetun mallin julkaisu saattaa myös vertautua tekijänoikeudella suojatun materiaalin julkaisuun.

Koneoppimismallin joutuminen vääriin käsiin voi myös mahdollistaa sen, että mallia käyttävää järjestelmää vastaan on helpompi hyökätä muilla tavoin. Hyökkääjä voi esimerkiksi testata etukäteen mallin päätökset ja muokata syötteitään niin, että malli tekee hyökkääjälle edullisia päätöksiä. Tällöin mallin luottamuksellisuuden menetys voi olla vain yksi askel pidemmässä hyökkäyspolussa, jonka tavoite on muualla, esimerkiksi mallinväistöhyökkäyksessä, josta myöhemmin eheyttä käsittelevässä osiossa.

Koneoppimismallit ovat ohjelmistoja ja kuten muidenkin ohjelmien tapauksessa, niiden luottamuksellisuus saattaa murtua, jos hyökkääjä pääsee käsiksi alustaan, jolla niitä opetetaan tai käytetään tai jonne ne on

tallennettu. Luottamuksellisuuden lisäksi alustojen turvallisuus vaikuttaa myös koneoppimisen eheyteen (ks. kappale Koneoppimismallin ja opetusdatan eheys). Alusta voi olla oma, suljettu järjestelmänsä, mutta nykyään tyypillisempää on käyttää jotakin julkista pilvipalvelua joko infrastruktuuripilvenä (IaaS, Infrastructure as a Service), jonka päälle oma koneoppimisratkaisu rakennetaan, tai pilvipalveluntarjoajan alusta- tai ohjelmistopilvenä (MLaaS, Machine Learning as a Service). On myös yleistä käyttää alihankkijaa koneoppimisratkaisun kehittämiseen, jolloin turvallisuus riippuu myös sen tietoturvan tasosta.

Koneoppimismalli sijaitsee käytönaikaisesti joko taustajärjestelmässä (usein ”pilvessä” tai sen ”reunalla”), josta sitä käytetään verkkoyhteyden yli lähettämällä pyyntöjä sen sovellusrajapintaan, tai sitten se tuodaan lähelle käyttäjää esimerkiksi sisällyttämällä se käyttäjän laitteeseen. Esimerkiksi älykaiuttimet sisältävät mallin, joka tunnistaa herätesanan (esim. Amazonin tuotteissa ”Alexa!”) paikallisesti, ja äänikomento lähetetään taustajärjestelmään vasta herätesanan jälkeen. Mallin sijainti määräytyy todennäköisesti pääasiassa sovellusalueen mukaan, mutta erityisesti laitteeseen asennettuna voi olla tarpeen ottaa huomioon, että laite itsessään voi olla kokonaan hyökkääjän hallussa, ja mallin varastaminen voi tämän vuoksi olla helpompaa. Tietosuojamielessä laitteeseen asennettu malli voi joissakin tapauksissa taas olla parempi vaihtoehto, koska mahdollisesti henkilötietoja sisältävää dataa ei tällöin tarvitse lähettää taustajärjestelmiin; ks. kappale Arkkitehtuurin ja toteutuksen taso.

Mallin varastaminen voi tapahtua joko konkreettisesti varastamalla mallin toteutus tai hyökkääjä voi rakentaa mallista kopion (*model stealing*) lähettämällä mallille syötteitä, tallentamalla mallin vastaukset ja opettamalla näillä oman kopiomallinsa tekemään vastaavat päätökset. Tämänkaltaiseen hyökkäykseen ei tarvita erityistä pääsyä mallin toteutukseen, vaan teoriassa pelkkä pääsy kyselyrajapintaan riittää. Taustajärjestelmässä kyselyrajapinnan takana sijaitsevaa mallia on helpompi puolustaa tältä hyökkäykseltä, koska kyselyjen määrää voidaan rajoittaa esimerkiksi rajapinta-avaimilla ja kyselymääriä rajaamalla (*throttling*).

OPETUSDATAN LUOTTAMUKSELLISUUS

Koska opetusdatasta jää malliin ja sitä kautta mallin päätöksiin informaatiota, opetusdatan luottamuksellisuutta uhkaa myös opetusdatan päättelyhyökkäys (*training data inference attack*). Hyökkääjä, jolla on koneoppimismalli hallussaan, voi suorittaa hyökkäyksen kolmella eri tavalla. Mallin kääntämisessä (*model inversion*) hyökkääjä rekonstruoi opetusdataa takaisinmallintamalla (*reverse engineering*) koneoppimismallin. Tätä hieman kevyempi vaihtoehto on attribuuttien päättely (*attribute inference*), jossa hyökkääjä vastaavasti päättelee mallista joitakin opetusdatan piirteitä. Mikäli hyökkääjällä on pääsy vain mallin rajapintaan koko mallin sijaan, tietoalkion päättely (*member inference*) on edelleen mahdollinen hyökkäys. Tässä hyökkääjä pyrkii selvittämään, käytettiinkö jotakin tiettyä datapistettä mallin opetukseen. Mikäli opetusdatan luottamuksellisuudella on sovellusalueella merkitystä, nämä hyökkäysmahdollisuudet vaikuttavat järjestelmän arkkitehtuurissa siihen, missä malli sijaitsee ja miten sen kyselyrajapinta avataan.

Tekoälyä voi mahdollisesti käyttää myös järjestelmään tallennettujen luottamuksellisten tietojen vuotamisen välillisesti. Voidaan hyvin kuvitella esimerkiksi tilanne, jossa hyökkääjä pyrkii käyttämään verkkokauppaa täsmälleen samoin kuin joku toinen henkilö, ja suosittelujärjestelmä alkaakin näyttää hyökkääjälle tämän toisen henkilön edellisiä ostoksia. Mitä harvinaisempi kohdehenkilön ostoprofiili on, sitä todennäköisemmin hyökkäys onnistuu.

Hyökkääjä, joka pystyy vaikuttamaan opetusdataan tai mallin opetukseen, voi myös tehdä mallista helpomman takaisinmallinnettavan. Tämä voi tapahtua niin, että malli alkaa ”vuotaa” enemmän informaatiota opetusdatasta päättelyhyökkäyksissä tai malli saattaa itsessään päätyä sisältämään suoranaisia otteita opetusdatasta. Myös tilanne, jossa useat tahot opettavat samaa mallia omalla opetusdatallaan hajautetusti (*federated learning*) voi naiivisti toteutettuna mahdollistaa sen, että nämä tahot voivat kyetä päättelemään asioita muiden tahojen opetusdatasta.

Opetusdata itsessään toki tarvitsee tietoturvatoimia kuten mikä tahansa muu luottamuksellisuutta vaativa perinteisen tietojärjestelmän data. Haasteena opetusdatan ja minkä tahansa muun data-analytiikassa käytetyn datan osalta on, että opetusvaiheessa dataa on todennäköisesti tarpeen käsitellä melko laajasti. Esimerkiksi datan siistiminen ja laadunvarmistus, tilastoharhojen ja vinoumien löytäminen ja poistaminen, mahdollisesti tarvittava anonymisointi ja datan tallennusmuotojen muuttaminen ja yhdenmukaistaminen ovat usein luonteeltaan tutkivaa työtä, joka saattaa vaatia kertakäyttöistenkin työkalujen toteutusta. Mikäli organisaation data-analyysiin tarjotut työkalut eivät ole riittäviä, painetta datan kopiointiin esimerkiksi

data-analyytikon työkoneelle tai kolmannen osapuolen pilvipalveluun voi esiintyä.

Nykyinen paine saada hyvälaatuista opetusdataa saattaa helposti johtaa myös siihen, että eri toimijat haluavat käyttää dataa sellaisiin tarkoituksiin, johon sitä ei alun perin ole tarkoitettu - esimerkiksi sosiaalisen median videoita tai videoneuvotteluja voitaisiin käyttää opetusdatana. Tämä ei aina ole suoranaisesti luottamuksellisuuskysymys, onhan data esimerkiksi sosiaalisessa mediassa usein saatettu julkisesti saataville, mutta se voi olla tietosuoja- ja kuluttajansuojakysymys. Tietosuoja-asetus on rajoittanut määrittelemättömiä tulevaisuuden käyttötarkoituksia henkilötietojen suhteen, mutta kaikkialla maailmassa rajoituksia datan luovalle uudelleenkäytölle ei ole.

KONEOPPIMISMALLIN JA OPETUSDATAN EHEYS

Suurin osa suoranaisesti koneoppimiseen liittyvistä hyökkäysmenetelmistä liittyy mallin eheyteen. Kuten kappaleessa Tekoäly ja koneoppiminen käsitteinä todettiin, tekoälyn läpinäkyvyys, selitettävyys ja vikasietoisuus ovat olennaisia tekoälyjärjestelmän hallinnan kannalta. Koneoppimismallin eheyden rikkoutuminen vähentää luottamusta koneoppimismallin päätöksiin joko kokonaisuutena tai osalla syötteistä ja vaikuttaa näin koko järjestelmän tärkeimpiin hallittavuuden periaatteisiin.

Kuten aiemminkin on jo useasti todettu, koneoppimismalli on opetusdatansa ja opetusprosessinsa tulos, joten mikä tahansa järjestelmän osa, joka voi vaikuttaa opetukseen tai malliin, saattaa vaikuttaa mallin eheyteen. Opetusdatan lisäksi mallin opetukseen tai tallennukseen käytettävien alustojen hallinta voi mahdollistaa hyökkääjälle mallin eheyden murtamisen (samoin kuin luottamuksellisuuden menetyksen, ks. kappale Koneoppimismallin luottamuksellisuus).

Käytännön esimerkkinä eheyden murtumisesta voisi olla esimerkiksi hypoteettinen tilanne, jossa autonominen ajoneuvo tulkitsisi liikennemerkin väärin, esimerkiksi stop-merkin nopeusrajoitusmerkkinä.

Mallin eheyttä vastaan voidaan hyökätä esimerkiksi dataa myrkyttämällä (*data poisoning*). Tässä hyökkäyksessä mallin opetusdataa muutetaan, tyypillisesti joko lisäämällä opetusdataan uutta materiaalia tai muuttamalla olemassa olevaa dataa. Datan myrkytyksellä pyritään yleensä joko vähentämään mallin päätöksien tarkkuutta (tätä kutsutaan tässä yhteydessä myös palvelunestohyökkäykseksi, *denial of service*) tai lisäämään malliin takaovi (*back door*), jolloin tietty syöte aiheuttaa hyökkääjän haluaman päätöksen, vaikka malli muuten näyttäisi toimivan oikein. Liikennemerkkiesimerkissä myrkyttäminen voisi tapahtua esimerkiksi lisäämällä opetusdataan valokuvia stop-merkeistä, jotka on opetusdatassa merkitty nopeusrajoitusmerkeiksi.

Hyökkääjälle tyypillisin tapa myrkyttää dataa on käyttää hyväkseen datalähteitä, joista tulevan datan alkuperää ja oikeellisuutta ei pystytä varmistamaan. Joidenkin koneoppimismallien, esimerkiksi syvien neuroverkkojen (*deep neural networks*), opetukseen tarvitaan suuria datamääriä. Koneoppimisen kehittäjät pyrkivät tällöin keräämään dataa hyvin laajalti, esimerkiksi julkisista datalähteistä tai yksittäisiltä asiakkailta tai heidän laitteistaan. Tämä antaa hyökkääjälle mahdollisuuksia syöttää opetusdataan haluamaansa dataa, mahdollisesti vieläpä niin, ettei hyökkääjän tarvitse varsinaisesti hyökätä mitään taustajärjestelmää vasten ja niin, että pahantahtoinen toiminta voi olla uskottavasti kiistettävissä. Hyökkääjä saattaa esimerkiksi lisätä lähettämäänsä dataan kohinaa tai väärentää opetusdataksi kerättävien havaintojen luokituksia (*labels*).

Esimerkki väärennetyistä luokituksista voisi olla hypoteettinen pankkipalvelu, jossa tilitapahtumille voi itse antaa luokituksen (“elintarvikkeet”, “lastenhoitokulut”). Mikäli asiakkaiden itsensä antamia luokkia käytetään järjestelmän opettamiseen, pienikin määrä tahallisesti väärin luokiteltuja tilitapahtumia voi johtaa muiden asiakkaiden tilitapahtumien väärään luokitteluun. Jos käyttöliittymä on tarkoitettu näennäisesti vain käyttäjän omien tilitapahtumien luokittelun korjaamiseen, on hyvin vaikea argumentoida, että hyökkääjä olisi edes tehnyt mitään väärää.

Toinen hyökkäys mallin eheyttä vastaan on mallinväistöhyökkäys (*evasion*). Tällöin hyökkääjä valitsee tai luo koneoppimismallille käytönaikaisia syötteitä, jotka on suunniteltu niin, että malli tulkitsee ne väärin tai antaa tulkinnoilleen matalan luottamustason. Liikennemerkkiesimerkissä hyökkääjä voisi mahdollisesti muokata oikeaa stop-merkkiä lisäämällä merkkiin häiriötekijöitä

(*perturbations*) siten, että auton järjestelmä luokittelee sen nopeusrajoitusmerkiksi. Mikäli hyökkääjällä on hallussaan kohteena oleva koneoppimismalli, tarvittavat häiriötekijät voidaan tehokkaasti laskea sen avulla (ks. kappale Koneoppimismallin luottamuksellisuus). Häiriötekijät voidaan yleensä myös laskea niin, että ihminen ei todennäköisesti erota niitä muokatusta syötteestä - edellisessä esimerkissä liikennemerkistä. (29)

Koneoppimismallin eheys voi murtua myös siksi, että hyökkääjä pystyy vaikuttamaan johonkin mallin opetuksessa käytettävään ohjelmistokirjastoon kuten opetus-, optimointi-, sarjallistamis- tai muuhun kirjastoon. Nämä kirjastot ovat usein ulkopuolisten tekemiä ja avointa lähdekoodia. Hyökkäys kirjastoja vastaan vertautuu ohjelmistoriippuvuuksien (*dependencies*) toimitusketjuhyökkäyksiin (*supply chain attacks*). Jos mallin opetus perustuu ennalta opetettuun malliin, myös tämän ennalta opetetun mallin eheyden murtuminen johtaa sen pohjalta kehitetyn mallin eheyden murtumiseen.

KONEOPPIMISMALLIN SAATAVUUS

Saatavuus on järjestelmän ominaisuus olla tarvittaessa käytettävissä ja toimintakykyinen. Saatavuutta heikentävät hyökkäykset voivat olla vakavia riskejä esimerkiksi silloin, kun tekoälyjärjestelmää käytetään aikakriittisessä sovelluksessa. Esimerkiksi autonominen ajoneuvo ei välttämättä ehdi pysäyttää tai siirtää vastuuta ihmiselle ajoissa, jos sen järjestelmä ei enää pystykään muodostamaan päätöksiä ajoissa. Jos tekoälyä taas käytetään avustavana tekniikkana, saatavuusongelma saattaa heikentää järjestelmän kokonaisturvallisuutta - esimerkiksi ihmisen ohjaama ajoneuvo ei enää pystyisikään havaitsemaan jalankulkijoita edessään, jos ajoneuvossa tällainen ominaisuus olisi.

Saatavuushyökkäys koneoppimisjärjestelmiä vastaan voidaan tehdä samoin kuin muita tietojärjestelmiä vastaan. Rajapintoihin voidaan esimerkiksi lähettää häiritsevän suuria määriä kyselyitä tai esimerkiksi viallisilla kyselyillä voidaan aiheuttaa virhetila, joka estää rajapinnan toimimisen. Hyökkäys voi kohdistua kyselyrajapinnan sovelluskerroksen lisäksi myös alempiin protokollakerroksiin.

Riippuen sovelluksesta koneoppimisjärjestelmiä vastaan voidaan joskus rakentaa syötteitä, joiden käsittely kestää koneoppimismallilta kauan. Tällöin hyökkääjä pyrkii maksimoimaan koneoppimismallin päätöksen tekemiseen vaadittavan ajan. Näin mallin kyky vastata todellisiin kyselyihin pienenee.

Saatavuushyökkäys voidaan suunnata myös järjestelmän syötteisiin, kuten esimerkiksi autonomisen ajoneuvon sensoreihin (*sensor blinding*) tai sensoridataa siirtävään tiedonsiirtokanavaan. Myös koneoppimismallin päätöksien kommunikointiin voidaan vaikuttaa, jolloin koneoppimisjärjestelmä tekee edelleen päätöksiä, mutta esimerkiksi koneen toimilaitteet eivät pysty vastaanottamaan niitä eivätkä pysty toimimaan niiden mukaan.

Saatavuuteen liittyvät riskit ovat leimallisesti usein järjestelmän integraatioon liittyviä riskejä, koska realistiset hyökkäysvaihtoehdot kattavat usein myös järjestelmän sensorit, toimilaitteet ja tiedonsiirron. Saatavuusriskejä voidaankin joutua torjumaan myös laitteiston ja mekaanisen suunnittelun osa-alueilla.

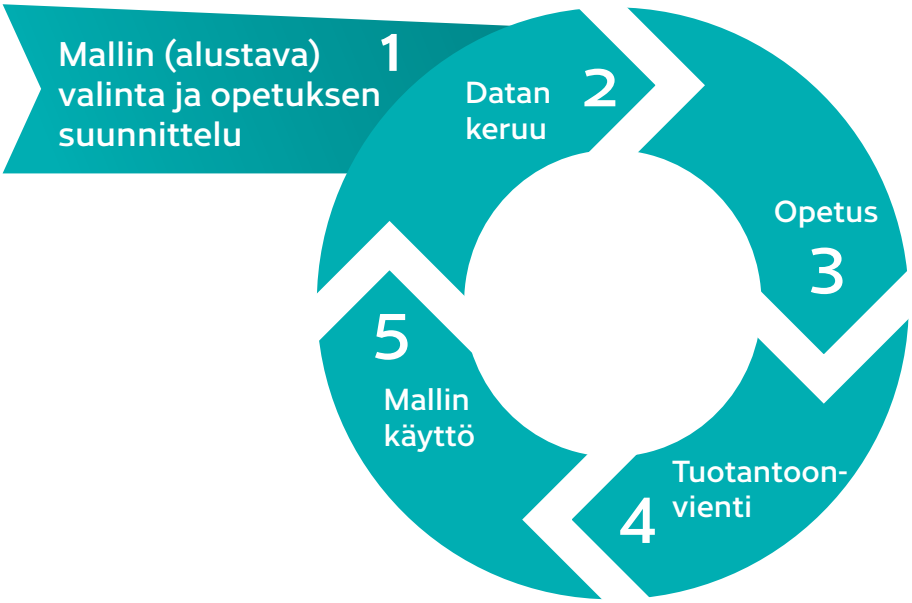
Esimerkiksi ajoneuvoissa on runsaasti perinteistä hyökkäyspintaa, kuten radiorajapintoja ja tietoliikenneväyliä. Koska ajoneuvo on fyysinen esine ja mahdollisesti hyökkääjän hallussa, hyökkäyspinta on altis myös paikalliselle hyökkäykselle. Etäohjatun ajoneuvon osalta täytyy myös huomioida tilanne, jossa ajoneuvo siirtyy matkapuhelinverkon kuuluvuusalueen ulkopuolelle. Esimerkiksi autonomisella aluksella ei avomerellä välttämättä ole käytettävissään riittävän nopeaa tietoliikenneyhteyttä sensoridatan välitykseen.

Koneoppimisjärjestelmän elinkaari

Tekoälyjärjestelmien riskejä on tärkeää lähestyä elinkaarianalyysin kautta. Erityisesti koneoppimisjärjestelmillä elinkaareen sisältyy vaiheita, joita perinteisillä tietojärjestelmillä ei ole, kuten opetusdatan keruu ja mallin opettaminen. Koneoppimisjärjestelmät saattavat jatkaa kehittymistään myös käyttöönoton jälkeen. Myös ympäröivä maailma saattaa muuttua, jolloin mallin vastaavuus reaalimaailman ilmiöiden kanssa vanhentuu. Tämä asettaa uusia vaatimuksia esimerkiksi järjestelmien testaukselle, jonka täytyy sellaisessa tapauksessa olla myöskin jatkuvaa.

Koneoppimisjärjestelmän elinkaaressa on viisi vaihetta:

1. **Mallin suunnittelu:** valitaan käytettävän mallin tyyppi ainakin alustavasti, opetusalgoritmi, optimointitapa ja määritellään opetettavat piirteet.
2. **Datan keruu:** Kerätään opetusdata, määritetään datan arvot (*label*), puhdistetaan data ja muutetaan se koneoppimismallille kelpaavaan muotoon (*feature extraction*). Tässä vaiheessa data voidaan myös jakaa opetus-, validointi- ja testidataan.
3. **Opetus:** Valitaan lopullinen mallin tyyppi ja opetetaan mallia valitun opetusalgoritmin avulla. Opetusalgoritmi säätää mallin parametreja opetuksen perusteella ja mallin suorituskykyä pyritään parantamaan säätämällä sen hyperparametreja sekä toistamalla opetus- ja validointivaiheita.
4. **Käyttöönotto ja integraatio:** Koneoppimismalli viedään toimintaympäristöönsä ja integroidaan sitä ympäröivään tekoälyjärjestelmään.
5. **Ennusteiden ja päätösten tekeminen (*inference*):** Koneoppimismalli tekee ennusteita tai päätöksiä syötteidensä pohjalta.



Kuva 4: Koneoppimisjärjestelmän elinkaaren vaiheet

Kuvassa 4 on kuvattu koneoppimisjärjestelmän elinkaaren vaiheet silmukkana. Joissakin tapauksissa koneoppimismallia päivitetään keräämällä uutta opetusdataa ja päivittämällä mallia tämän avulla. Tätä voidaan joutua tekemään esimerkiksi siksi, että toimintaympäristö muuttuu tai mallilta vaadittavat päätökset muuttuvat (*model decay*). Mallin jatkuva evoluutio johtaa siihen, että mallin suorituskykyä ja turvallisuutta on myös arvioitava yhä uudelleen - vähintään jokaisen opetuskerran jälkeen. Arviointia voidaan myös joutua tekemään uudelleen ajan mittaan ympäristön (ja sitä kautta syötteiden) muuttuessa, vaikka malli itsessään ei muuttuisikaan. Näitä ilmiöitä kutsutaan ympäröivän maailman muuttuessa konseptin ajelehtimiseksi

(*concept drift*) ja oppimisdatan muuttuessa kovariansin siirtymäksi (*covariate shift*).

Koneoppimisjärjestelmän tuotekehitysprosessissa tietoturvan uudelleenarviointi olisi mielekästä sitoa kehitysprosessiin niin, että sitä tehdään iteratiivisesti myös ”ylläpitovaiheessa”. Mikäli mallia päivitetään jatkuvasti uuden datan perusteella, teoriassa esimerkiksi safety-tyyppiseen turvallisuuteen liittyvät tekoälyn testit pitäisi ajaa jokaisen muutoksen jälkeen. Käytännössä tämä vaatii testauksen täyttä automaatiota.

Mikäli koneoppimisjärjestelmän elinkaari jakaantuu usealle toimijalle, riskit saattavat syntyä vääristä olettamuksista ja epäselvyyksistä heidän välillään. Esimerkiksi ulkopuolisen tahon ennalta opetetun mallin käyttäminen tarkoittaa, että kyseisen mallin opetusdataa ei välttämättä ole käytettävissä testaukseen. Pääsy opetusdataan kannattaa varmistaa myös, jos mallin opetus tehdään kokonaan toisen tahon toimesta ja malli joudutaankin opettamaan uudelleen, mahdollisesti eri pohjalle. Alihankintasopimuksissa pääsy opetusdataan voi muodostaa toimittajaloukun.

Vaikkakaan tämä ei ole suoranaisesti tietoturvariski, se voi olla liiketoiminnallinen jatkuvuusriski. Myös testausvastuu, pääsy testidataan ja testidatan ylläpito ajan mittaan tulisi selkeyttää ja vastuuttaa samoista syistä.

Koneoppimismallin käyttönotto (”tuotantoonvienti”) on tietoturvariskeiltään verrattavissa minkä tahansa muun ohjelmiston käyttöönottoon. Jos malli on esimerkiksi tallennettu versionhallintaan ja se viedään sieltä esimerkiksi pilvipalveluun, organisaatiolla on todennäköisesti jonkinlainen integraatio- ja tuotantoonvientiputki (*deployment pipeline*, *continuous integration/continuous delivery*, CI/CD), joka huolehtii tästä automaattisesti tai puoliautomaattisesti. Mikäli hyökkääjä saa tällaisen järjestelmän hallintaansa, hän voi todennäköisesti hyökätä tekoälyjärjestelmää vastaan monin eri keinoin.

Mielenkiintoinen kysymys on, miten lisääntyvä ohjelmiston määrä näkyy esimerkiksi ajoneuvokatsastuksessa. Jo nykyään ohjelmistopäivityksellä voidaan merkittävästi muuttaa ajoneuvojen ominaisuuksia, esimerkiksi huippunopeutta ja kiihtyvyyttä - autonomisten ajoneuvojen maailmassa muutokset voivat olla vielä suurempia. Onko tulevaisuuden ajoneuvokatsastuksella roolia ajoneuvon ohjelmistokonfiguraation varmistamisessa, esimerkiksi ohjelmistopäivitysten ajantasaisuuden osalta?

Yleisimpien riskien ehkäisy

Tekoälyjärjestelmien tietoturvariskejä ehkäistään pääosin samoin kuin perinteisten järjestelmien riskejä. Erityisesti koneoppimisjärjestelmät vaativat kuitenkin myös niille räätälöityjä lähestymistapoja, jotka vaihtelevat arkkitehtuurisista päätöksistä koneoppimismallien luonnissa ja käytössä tehtäviin tietoturvaa lisääviin toimenpiteisiin. Arkkitehtuuriset tietoturvakontrollit liittyvät ennen kaikkea mallien sijaintiin järjestelmässä ja niiden syötteiden kontrollointiin. Koneoppimismalleihin liittyvät tietoturvakontrollit taas liittyvät siihen, miten mallia opetetaan ja miten opetusdataa käsitellään ennen opetusta.

Tekoälyjärjestelmien riskien ehkäisytavat voidaan jakaa kahteen pääryhmään:

1. Järjestelmätason ehkäisymenetelmät, jotka pätevät osittain myös muihin tekoälyjärjestelmiin kuin koneoppimisjärjestelmiin ja muistuttavat perinteisten järjestelmien riskien ehkäisyä. Nämä voidaan edelleen jakaa
   - Arkkitehtuurisiin lähestymistapoihin, jotka liittyvät järjestelmäsuunnitteluun ja käyttöönottoon
   - Tietoturvaominaisuuksiin, jotka ovat riskejä vähentäviä toiminnallisuuksia

2. Koneoppimismallitason ehkäisymenetelmät, jotka sopivat nimenomaisesti koneoppimista käyttävien tekoälyjärjestelmien riskien ja koneoppimishyökkäysten ehkäisyyn.

Useita koneoppimiseen liittyviä riskejä voidaan ehkäistä käyttäen pelkästään järjestelmätasonkin menetelmiä. Joskus kuitenkin tarvitaan myös koneoppimismallitason riskienhallintaa.

Tällä hetkellä ei kuitenkaan ole olemassa mitään yksittäistä tapaa täysin ehkäistä kaikkia koneoppimiseen liittyviä riskejä. Koneoppimismalleja vastaan tehdyt hyökkäykset ovat aktiivinen tutkimusalue, ja uusia tuloksia julkaistaan säännöllisesti. Paras tulos saavutetaan yhdistelemällä eri tasoisia riskienhallintamenetelmiä lähtien tekoälyn käyttötapauksen valinnasta koneoppimismallin opetusmenetelmiin.

Arkkitehtuuriset valinnat

Arkkitehtuuritason riskienhallinnan päätavoitteena on pienentää hyökkäyspintaa (ks. kappale Hyökkäyspinta). Arkkitehtuurin tasolla on tyypillistä, että järjestelmän kehittäjän on valittava kahdesta eri lähestymistavasta. Kummallakin on yleensä sekä hyviä että huonoja puolia ja riippuu sovelluskohteesta ja muista riskienhallintatavoista, kumpi lähestymistapa kannattaa valita.

KESKITETTY VS. HAJAUTETTU OPETUS

Mikäli koneoppismallin opetus tehdään keskitetysti, opetusprosessia voidaan täysin hallita. Tämä tekee mallin eheyden (ks. kappale Koneoppimismallin ja opetusdatan eheys) turvaamisen helpommaksi, mutta toisaalta opetusdata on tällöin mallin opettajien saatavilla. Tämä voi olla ongelma esimerkiksi, jos opetusdata sisältää henkilötietoja tai muita luottamuksellisia tietoja, joita opetusdatan lähteet eivät halua tai voi luovuttaa. Hajautettu opetus, jossa datalähteet itse suorittavat osan opetuksesta, mahdollistaa sen, ettei opetusdataa tarvitse luovuttaa datalähteiden ulkopuolelle. Samalla se kuitenkin antaa datalähteille laajemman mahdollisuuden vaikuttaa opetukseen ja avaa ikkunan myrkytyshyökkäyksille ei vain datan vaan myös mallin päivitysten kautta.

DATALÄHTEIDEN RAJAAMINEN

Tekoälyjärjestelmän data-analyysiin (esimerkiksi koneoppimismallin opetukseen) käytettyjä datalähteitä voidaan myös rajata niin, että käytetään vain jollakin tavalla luotettavaksi tiedettyjä datalähteitä. Tämä vähentää mallin eheyden riskiä (ks. kappale

Autonomisia autoja varten on suunniteltu reunaprosessointia, jossa prosessointitehoa tuodaan teiden varsille tai esimerkiksi julkisen liikenteen busseihin, jotka toimisivat liikkuvina palvelinkeskuksina. Hyökkääjä pääsee helpommin fyysisesti käsiksi kirjaimellisesti tien reunalla nököttävään laatikkoon kuin hyvin vartioituun palvelinkeskukseen.

Koneoppimismallin ja opetusdatan eheys), mutta toisaalta pienempi datamäärä ja vinoutuneempi data saattavat alentaa tekoälyjärjestelmän suorituskykyä erityisesti sen tarkkuuden (*accuracy*) osalta.

Vinoutumia sisältävän datan tunnistaminen voi kuitenkin olla vaikeaa. Ihmistenkään suorittama arviointi ei välttämättä tunnista kaikkia vinoumia. Kriteerit, joiden pohjalta datalähteen luotettavuutta arvioidaan, voivat myös itse aiheuttaa datan vinoumia.

PÄÄTÖKSENTEON FYYSINEN SIJAINTI

Tekoälyjärjestelmän päätöksenteko voidaan tehdä joko taustajärjestelmässä, reunajärjestelmässä (*edge*) tai esimerkiksi asiakkaan laitteessa. Asiakkaan laitteessa tehtävä päätöksenteko poistaa tarpeen lähettää kyselyitä verkon yli taustajärjestelmiin, jolloin esimerkiksi henkilötietojen käsittelyn turvallisuus tekoälyjärjestelmässä on helpommin toteutettavissa (ks. kappale Koneoppimismallin luottamuksellisuus). Reunajärjestelmässä tehtävä päätöksenteko voi vähentää tiedonsiirron viiveitä ja vaikkapa henkilötietojen kertymistä yhteen paikkaan, mutta toisaalta reunajärjestelmät voivat olla fyysisesti alttiimpia hyökkäyksille. Keskitetty taustajärjestelmä mahdollistaa tekoälyjärjestelmän päätöksenteon suojaamisen parhaiten, mutta samalla se tuottaa suurimmat tiedonsiirtoviiveet ja pakottaa siirtämään mahdollisesti luottamuksellisen tiedon taustajärjestelmään päätöksentekoa varten.

Tietoturvaominaisuudet

Tietoturvaominaisuuksien tarkoitus on hankaloittaa hyökkäyspinnan kautta tapahtuvia hyökkäyksiä. Verrattuna arkkitehtuurisiin valintoihin nämä ominaisuudet eivät pakota yhtä syvällisiin keskusteluihin järjestelmän tavoitteista, mutta niiden toteutus vaatii toki ajallisia investointeja ja hyvän ymmärryksen siitä, miten ne vaikuttavat riskiin juuri kyseisellä sovelluksella.

KYSELYMÄÄRIEN RAJAAMINEN

Mallin varastamista ja opetusdatan takaisinlaskentaa voidaan ehkäistä rajaamalla mallille tehtävien kyselyiden määrää. Tämä onnistuu yleensä vain tilanteissa, joissa malli sijaitsee taustajärjestelmässä. Rajapinnoille voidaan asettaa maksimikyselymääriä tietyssä aikaikkunassa tai rajata kyselyiden määrää esimerkiksi samasta verkko-osoitteesta. Rajaukset on kuitenkin suunniteltava hyvin, jotta ne eivät tee palvelunesto-hyökkäyksistä helpompia.

SYÖTTEIDEN PUHDISTUS

Ennen kuin syöte annetaan koneoppimisjärjestelmälle, syötettä voidaan käsitellä joidenkin hyökkäysten hankaloittamiseksi. Esimerkiksi kuvasta voidaan poistaa tai siihen voidaan lisätä kohinaa, jolloin kohinalta näyttävät hyökkääjän mallinväistöhyökkäyksen suorittamiseksi lisäämät häiriötekijät (ks. kappale Koneoppimismallin ja opetusdatan eheys) poistuvat tai peittyvät.

Sensori itsessään saattaa toimia tällaisena suodattimena. Esimerkiksi kamerakuvan kohinaisuus ja matala erottelukyky voivat poistaa tai vaimentaa häiriötekijöitä.

POIKKEAVIEN SYÖTTEIDEN TUNNISTUS

Syötteitä voidaan myös tarkkailla tilastollisesti normaalista poikkeavien syötteiden havaitsemiseksi. Tällä voidaan pyrkiä estämään mallin varastamista ja mallinväistöhyökkäyksiä (ks. kappaleet Koneoppimismallin luottamuksellisuus ja Koneoppimismallin ja opetusdatan eheys). Ongelmana tässä lähestymistavassa on, että joskus poikkeama on todellinen - esimerkiksi turvallisuuskriittinen järjestelmä ei välttämättä voi olettaa toimintaympäristönsä olevan aina normaalissa tilassa. Lisäksi poikkeamia strukturoimattomassa syötteessä kuten äänen bittivirrassa on vaikea havaita - sen vuoksi koneoppimista on todennäköisesti alun perin alettu sovelluksessa käyttääkin.

Autonomisen ajoneuvon tekoälyn järkevyystarkastus voi esimerkiksi yhdistää kahden eri sensorityypin havaintoja, esimerkiksi näkyvän valon kameran kuvaa ja LIDAR-pistepilveä liikenteen havaitsemiseksi tai verrata navigaatiojärjestelmältä saatavaa sijainti- ja korkeustietoa korkearesoluutioiseen maanpinnan korkeuskarttaan.

Erityisen hankalaksi tilanne muodostuu, jos sensoridatan laatu tai eheys heikkenee ilman, että se on helposti havaittavissa. Jos lumi pyryttää ajoneuvon kamerat umpeen, tilanne on hyvin selkeä. Toisaalta aktiivinen hyökkääjä, joka korvaisi kamerakuvan aiemmalla samasta kamerasta napatulla tallenteella, ei olisi välttämättä helposti havaittavissa. Joissakin tapauksissa sensorien on mahdollisesti tarjottava mahdollisuus sensoridatan tekniseen todentamiseen, erimerkiksi digitaalisin allekirjoituksin.

POIKKEAVIEN ENNUSTEIDEN TUNNISTUS

Järjestelmässä voidaan myös tunnistaa poikkeavia ennusteita ja tehdä niille jonkinlaisia järkevyystarkistuksia. Tähän voidaan käyttää yksinkertaisimmillaan sääntöjä ”mahdottomista” tilanteista. Järkevyystarkastuksissa voidaan käyttää myös koneoppimismalleja, jolloin järjestelmää valvovat mallit eivät tee varsinaisia päätöksiä, mutta niillä on jonkinlainen veto-oikeus tuloksiin. Myös useampia malleja voidaan yhdistää päätöksen tekemiseen (*ensemble*), jolloin yksittäisen mallin virheet eivät välttämättä vaikuta tulokseen.

Järjestelmälle voidaan myös määritellä useita eri toimintatiloja, esimerkiksi ”suorituskykyinen tila” ja ”turvallisempi tila”, joiden välillä järjestelmä vaihtaa tilaa havaitessaan poikkeaman. Tilan vaihto voi esimerkiksi tarkoittaa mallin vaihtoa huonompaan mutta hyökkäyksiä kestävämpään malliin, tai järjestelmä voi poistaa tekoälyjärjestelmän kokonaan tai osittain käytöstä.

ENNUSTEIDEN LAADUN PIILOTTAMINEN JA HEIKENNYS

Jos käytössä oleva koneoppimismalli pystyy tuottamaan arvion ennusteensa todennäköisyydestä, koneoppimismallin kyselyrajapintaa voidaan rajata siten, että se palauttaa vain ennusteen eikä tätä todennäköisyyttä. Tämä pienentää mallista vastausten mukana vuotavan informaation määrää ja vaikeuttaa tehokkaasti mallin varastamista ja opetusdatan takaisinmallinnusta.

Jos todennäköisyystieto on sovellukselle tärkeä, mallin palauttamiin ennusteisiin voidaan sen poistamisen sijaan lisätä kohinaa. Riittävällä määrällä kyselyitä hyökkääjä saattaa kuitenkin pystyä poistamaan vastauksissa olevan kohinan.

Mallitason ehkäisymenetelmät

Koneoppimismalleja voidaan muuttaa niin, että niitä vastaan on hankalampi hyökätä. Tällä hetkellä tunnetuista menetelmistä useimmat liittyvät opetusprosessin muokkaamiseen. Mallitason menetelmien käytöllä on usein myös ei-toivottuja vaikutuksia. Usein malli, joka on vähemmän altis hyökkäyksille, on myös tarkkuudeltaan huonompi. Jonkin tietyn hyökkäyksen hankaloittaminen saattaa puolestaan helpottaa jotakin toista hyökkäystä. Tämän vuoksi kohteena olevan sovelluksen koneoppimisjärjestelmän uhkamalli[8] (*threat model*) on pidettävänä selkeänä mielessä. Tarkkuudeltaan huonompi ja yksinkertaisempi malli saattaa olla myös vikasietoisempi, joten tarkkuuden heikkeneminenkään ei ole yksiselitteisesti aina huono asia.

MALLIN REGULARISOINTI

Koneoppimismallia voidaan regularisoida (*regularization*) opetuksen aikana. Tällöin koneoppimismallia ohjataan tasapainoisempaan suuntaan ”rankaisemalla” liian monimutkaista mallia sen lisäksi, että palkitaan mallin tarkkuudesta. Jos hyökkääjä on pystynyt myrkyttämään vain pienen osa opetusdatasta, regularisointi saattaa onnistua jättämään juuri myrkytetyt osat huomiotta. Liiallinen regularisointi voi heikentää mallin oppimiskykyä ja mahdollisesti estää mallia pääsemästä optimaaliseen ratkaisuun - regularisointia käytetäänkin myös mallin ylisovittamisen (*overfitting*) ehkäisemiseen.

DIFFERENTIAALINEN TIETOSUOJA

Koneoppimismallia voidaan opettaa käyttämällä differentiaalisen tietosuojan (*differential privacy*) menetelmiä. Näiden menetelmien tarkoituksena on estää yksittäisen opetusdatan datapisteen ominaisuuksien paljastuminen kuitenkin säilyttäen opetusdatan yleiset ominaisuudet. Henkilötietoihin viittaavasta nimestään huolimatta menetelmiä voi käyttää muunkin luottamuksellisen informaation suojaamiseksi.

Differentiaalisen tietosuojan voi intuitiivisesti ymmärtää esimerkiksi niin, että jos opetusdatasta otetaan jokin datapiste, se on totta vain tietyllä todennäköisyydellä, ja vastaavasti jollakin todennäköisyydellä se on esimerkiksi satunnaista dataa. Vaihtoehtoisesti voidaan ajatella, että opetusdatasta voi poistaa minkä tahansa yksittäisen datapisteen datan tilastollisten ominaisuuksien siitä kärsimättä.

Hajautettua opetusta tai datankeruuta käytettäessä differentiaalisen tietosuojan menetelmillä voidaan suojata yksittäisiä datapisteitä opetusvaiheessa. Keskitettyä opetusta käytettäessä tällä voidaan ehkäistä sitä, että takaisinmallintamalla voitaisiin paljastaa yksittäisen datapisteen ominaisuuksia.

Yksinkertaisimmillaan differentiaalisen tietosuojan menetelmä voi olla esimerkiksi kohinan lisäämistä opetusdataan tai opetusvaiheessa neuroverkon mallin painoihin. Klassinen esimerkki on muuttaa tietty määrä opetusdataa vääräksi jollakin todennäköisyydellä jo datan keräysvaiheessa, jolloin yksittäisen - hyökkääjälle vuotaneen tai takaisinmallinnetun - opetusdatan pisteen osalta on jokin todennäköisyys, että se ei pidäkään paikkaansa.

Differentiaalisen tietosuojan menetelmien vaikutus mallin suorituskykyyn muistuttaa aiemmin mainittua regularisointia. Ne heikentävät mallin suorituskykyä, joten saavutetun turvallisuuden ja mallin käyttökelpoisuuden väliltä on löydettävä sopiva tasapaino.

VIHAMIELINEN OPETUS

Malleja voidaan myös tarkoituksellisesti opettaa vihamielisellä opetusdatalla (*adversarial training*). Mallista tulee näin kestävämpi erityisesti mallinväistöhyökkäyksiä (ks. kappale Koneoppimismallin ja opetusdatan eheys) vastaan, mutta toisaalta se heikentää mallin suorituskykyä, koska vihamielinen opetusdata ei luonnollisestikaan kuvaa haluttua todellisen maailman ilmiötä, vaan mallia opetetaan tällöin sietämään valitun tyyppisiä häiriötekijöitä.

8 Uhkamalli tarkoittaa yksinkertaistaen ymmärrystä siitä, mitkä asiat voivat mennä pieleen ja mitkä niiden tietoturvavaikutukset ovat. Jokin tietoturvariski saattaa olla jollekin sovellukselle täysin merkityksetön tai sen hyödyntäminen mahdotonta, jolloin sanotaan, että se ”ei kuulu uhkamalliin”. Kun uhkamalli on hyvin ymmärretty, myös puolustukselliset toimenpiteet voidaan kohdistaa tehokkaasti. Termien samankaltaisuudesta huolimatta uhkamalli ei ole koneoppimismalli.

Tekoälyn riskien hallinta tuotekehitysprosessissa

Tekoälyn tietoturva- ja tietosuojariskien hallinnan periaatteet tuotekehityksessä muistuttavat suuresti perinteisen ohjelmistokehityksen tietoturvaa. Painopiste on kuitenkin turvallisen ohjelmoinnin sijasta käyttötapausten ja vaatimusten määrittelyssä (palvelumuotoilussa ja tuotteenhallinnassa) sekä turvallisessa arkkitehtuurisuunnittelussa. Opetusdatan hallinnan ja data-analyysin osalta tietoturva muistuttaa pääosin perinteistä IT-järjestelmien tietoturvaa ja painottuu esimerkiksi pääsynhallintakysymyksiin ja auditoitavuuteen.

Koneoppimisjärjestelmillä tietoturvatestausta voidaan osin joutua lähestymään perinteisestä järjestelmästä poikkeavalla tavalla, koska koneoppimismalleja on vaikea tutkia ja testauksen on joskus tarpeen jatkua myös senkin jälkeen, kun järjestelmä on jo otettu käyttöön. Esimerkiksi staattisen analyysin työkalujen käyttöarvo koneoppimismallien analyysissä on rajallinen. Dynaamista testausta taas voi hankaloittaa testidatan puute ja ajan mittaan sen eriytyminen todellisen maailman ilmiöistä.

Elinkaaren hallinta koneoppimisjärjestelmissä voi olla perinteisiä järjestelmiä hankalampaa siksi, että jos järjestelmä oppii käytön aikana, se saattaa omaksua uusia ominaisuuksia ohi ”virallisen” tuotehallinnallisen päätöksenteon.

Tietoturvan organisointi IT- ja tuotekehitysorganisaatioiden yli

Tekoälyn ja erityisesti koneoppimismallien tietoturvariskien hallinta tuotekehitysprosessissa muistuttaa suuresti ohjelmistojen tuotekehitysprosessin turvaamista. Joitakin merkittäviä eroja kuitenkin on.

Tekoälyjärjestelmien alustaturvallisuudesta huolehtiminen muistuttaa suuresti perinteisen tuotekehityksen tietoturvaa, mutta siinä on hieman enemmän elementtejä perinteisen ”IT-turvallisuuden” puolelta. Suurin syy tähän on data-analytiikka ja esimerkiksi juuri koneoppimismallien kehityksen vaatima opetusdatan prosessointi.

Datamassoja hallitaan usein perinteisen IT-turvallisuuden puolella, jossa hoidetaan esimerkiksi työssä käytettävien tietokoneiden tietoturva, tietokantojen ja muiden tiedontallennuspaikkojen pääsynhallinta, MLaaS-alustojen pääsynhallintaan käytetty federoitu identiteetinhallinta ja integraatio- ja tuotantoonvientijärjestelmien ylläpito ja pääsynhallinta. IT-turvallisuus vaikuttaa tällöin suoraan tuoteturvallisuuteen, kuten kappaleessa Riskit on kuvattu.

Organisaatioiden olisi suositeltavaa käsitellä tietoturvaa kokonaisuutena niin, että tuotekehityksen ja IT:n tietoturva eivät ole toisistaan erillisiä saarekkeita. Myös siirtymä pilvinatiiviin (*cloud native*) maailmaan vaatii samankaltaista kokonaisuuden hallintaa, joten tekoäly ei ole ainoa syy tuotekehityksen ja IT:n raja-aitojen purkuun.

Tietoturvasta ja toisaalta data-analytiikasta vastaavien henkilöiden näkemykset sopivasta pääsynhallinnan tasosta voivat usein olla eriäviä. Kokonaisuuden kannalta paras ratkaisu ei todennäköisesti ole kieltää kaikkea. Perinteisen ohjelmistotuotannon alalla monet organisaatiot lähestyvät asiaa niin, että tuotekehitykselle tarjotaan sisäisesti tuotteistettuna kehittäjille mieluisia ja turvalliseksi todettuja ratkaisuja. Data-analyytikot, -insinöörit ja muut tekoälyjärjestelmiä kehittävät henkilöt kannattaisi ottaa mukaan tietojärjestelmien hankintaan ja tietoturvan suunnitteluun. Datan käsittelyn tarpeiden selvittäminen konkreettisen päivittäisen työnteon tasolla voi säästää hankaluuksilta myöhemmin, kun turvallisiksi tehdyt järjestelmät päätyvät data-analyytikkojen ensisijaiseksi valinnaksi käyttömukavuutensa vuoksi.

Palvelumuotoilun ja käyttötapausten määrittämisen taso

Tekoälyjärjestelmien systeemisten riskien arviointi on luontevaa tehdä silloin, kun järjestelmän käyttötapauksia mietitään. Riskien lopulliset vaikutukset juontavat yleensä juurensa nimenomaan käyttötapauksiin eivätkä nimenomaisiin tekoäly- tai koneoppimisalgoritmeihin. Liian suurta systeemistä riskiä voidaan mahdollisesti torjua käyttötapausta muuttamalla, joka on teknisesti kestävä keino ja voi tulla huomattavasti edullisemmaksi kuin reagoida asiaan myöhemmin korjaamalla sitä järjestelmän heikkoutena tai haavoittuvuutena.

Käyttötapausten miettimisen yhteydessä on usein luontevaa myös miettiä negatiivisia käyttötapauksia (kirjallisuudessa *misuse case* tai *attacker story*). Negatiivisessa käyttötapauksessa toivottuun tulokseen ei jostakin syystä päästä. Syy voi olla satunnainen ulkoinen tapahtuma, kuten toimilaitteen rikkoutuminen, tai johtua aktiivisen hyökkääjän toiminnasta.

Euroopan komission tekoälyasetusehdotus (2) asettaa suuririskisille tekoälyjärjestelmille vaatimuksia muun muassa läpinäkyvyyden ja käyttäjien informoinnin suhteen. Näiden vaatimusten huomioon ottaminen palvelumuotoilussa johtaa todennäköisesti parempaan lopputulokseen kuin niiden lisääminen järjestelmän käyttötapauksiin jälkikäteen. Jos järjestelmä on niin uudenlainen, että sen vaikutuksia perusoikeuksiin on syytä tutkia, Euroopan perusoikeusviraston tekoälyraportti (4) kuvaa perusoikeusvaikutusten analyysin (*fundamental rights impact assessment*), jota voi käyttää keskustelun pohjana.

Turvallisen tuotekehityksen prosesseissa on jo nykyisellään jossain määrin vastaava aktiviteetti, tietosuojavaikutusten arviointi. Tämä suomenkielinen termi tarkoittaa joko tietosuoja-asetuksen tarkoittamaa *data protection impact assessmentia* (DPIA) tai sitten yleisempää *privacy impact assessmentia* (PIA). Näiden termien taakse kätkeytyvä käytännön toiminta ja sen sitominen tuotekehitysprosessiin vaihtelee organisaatioittain, mutta tyypillisesti tämä tehdään jonkinlaiselle tuotteen toiminnallisuuden ylätason aihiolle selkeänä erillisenä toimenpiteenä. Ketterässä tuotehallintaprosessissa kohteena saattaa tyypillisesti olla tuotteen kehitysjonossa liiketoiminnallisen arvon kuvaus (usein käytetään termiä *epic*) tai käyttäjätarinan kuvaus (*user story*).

Tekoälyasetusehdotuksen 9. artikla vaatii suuririskisiltä järjestelmiltä hieman tätä muistuttavan aktiviteetin, riskien tunnistamisen ja analysoinnin. Monissa organisaatioissa tätä aktiviteettia on jo soviteltu tietosuojavaikutusten arvioinnin yhteyteen, koska sen ajatellaan tapahtuvan tuotekehityksessä ajallisesti samoihin aikoihin, ja tekoälyjärjestelmän käsitellessä henkilötietoja näillä molemmilla aktiviteeteilla saattaa olla yhteistä pohjaa.

Tämän kehityskulun on myös jo ajateltu johtavan siihen, että organisaation tietosuojatoiminnat kehittyvät kohti yleisempää ”datan hallinnan, jatkuvuussuunnittelun ja vaatimustenmukaisuuden” funktiota, jonka vastuualueeseen kuuluisivat myös Euroopan unionin tulevaisuuden datalainsäädännön vaatimukset.

Tässä lähestymistavassa kannattaa kuitenkin pitää mielessä, että tekoälyn - ja erityisesti koneoppimisen - riskit eivät ole pelkästään kysymyksiä lain- ja vaatimustenmukaisuudesta vaan jotkin näistä riskeistä ovat puhtaasti teknisiä. Toki tietosuojankin alalla on teknisiä kysymyksiä (näiden ratkomisesta on käytetty termiä *privacy engineering*), mutta suurin osa tietosuojan alan teknisistä haasteista on puhtaammin perinteisiä tietoturvahaasteita. Organisaation rakentaessa yleistä datanhallintafunktiota sen tulisi ottaa huomioon riittävä teknisen osaamisen taso. Samaten tämän toiminnan selkeä suhde palvelumuotoiluun kannattaa määritellä.

Esimerkkinä tekoälyn käyttöönoton palvelumuotoilusta on Oulun satama. Sataman kaltaisessa toimintaympäristössä on paljon pieniä ja keskisuuria yrityksiä, joilla on usein omat tietojärjestelmänsä ja omat pelkonsa tietojen käytöstä.

Luottamuksen rakentaminen tekoälyyn perustuville tuotteille on tärkeää. Palvelumuotoilumielessä haastetta lähestytään rakentamalla palveluita tarkasti valittujen syötteiden päälle niin, että saavutettavasta hyödystä on alusta lähtien selvä työhypoteesi. Näin vältetään esimerkiksi laajamittaisen - ja mahdollisesti turhan - datan keruun riskejä.

Toinen palvelumuotoilutason riskienhallintapäätös on rajoittaa käsiteltävä sensoridata ensi vaiheessa vain visuaaliseen dataan, jolloin tekoälyn sovellukset ovat lähellä lisättyä (augmented) älykkyyttä. Sataman kaltaisessa toimintaympäristössä on paljon visuaalisesti havaittavia ja ”julkisesti” nähtävillä datapisteitä kuten rahtikonttien merkintöjä. Näihin keskittymällä tekoälyn päätökset ovat helpommin ihmisten ymmärrettävissä ja järjestelmässä voidaan välttää syötteitä, jotka ovat liikesalaisuuksia.

Arkkitehtuurin ja toteutuksen taso

Nykyaikaisessa tuotekehitysprosessissa tietoturvaa pyritään usein luomaan tekemällä riskianalyysiä jossakin arkkitehtuurisuunnittelun, toiminnallisen suunnittelun ja toteutuksen välimaastossa. Tästä käytetään usein termejä uhkamallinnus (*threat modeling*), arkkitehtuurin riskianalyysi (*architectural risk analysis*) tai tietoturvan suunnitteluperiaatteiden katselmointi (*security design review*). Toteutustavat vaihtelevat yksittäisistä koko tuotetta koskevista analyyseistä iteratiiviseen, vuorollaan kutakin uutta toiminnallisuuden osaa tutkivaan lähestymistapaan.

Tekoälyn riskienhallinnassa tämän tason riskianalyysi on ehkäpä tärkeimmässä roolissa, ainakin jos tekoälyn sovellusalue ei ole itsessään uudenlainen tai korkeariskinen. Arkkitehtuurin analyysin tasolla voidaan löytää useimmat tekoälyjärjestelmän ulkoisten rajapintojen riskit sekä myös valtaosa tekoäly- tai koneoppimisjärjestelmän sisäisistä riskeistä (ks. kappale Hyökkäyspinta). Myös koneoppimismallien heikkouksien riskit (ks. kappale Riskit) on todennäköisesti helpointa ymmärtää ja suojauskeinot (ks. kappale Yleisimpien riskien ehkäisy) valita tällä abstraktion tasolla. Yhdysvaltalainen tutkijaryhmä on julkaissut erinomaisen taksonomian ja listan koneoppimisjärjestelmien arkkitehtuuritason riskeistä (30).

Hyvä lähestymistapa täysin uuden koneoppimisjärjestelmän luomiseen voi joskus olla luoda prototyyppi järjestelmästä ensin, jopa niin, että tekoälykomponentit on korvattu väliaikaisilla ”tyhmillä” komponenteilla (mock object). Tämä pakottaa järjestelmän suunnittelijat miettimään järjestelmäintegraatiota kokonaisuutena. Kun yleinen toiminnallisuus on varmistettu, tekoälyn lisääminen tuottaa järjestelmälle sen lopullisen toimintakyvyn. Samalla ”tyhmien” komponenttien käyttäminen prototyypissä tekee konkreettiseksi sen, miten järjestelmä saattaa toimia tekoälyn toimiessa väärin.

Fyysisille järjestelmille voidaan luoda myös virtuaalinen malli (digitaalinen kaksonen, digital twin), jossa voidaan testata tekoälyä ja muutoksia ilman, että fyysistä maailmaa altistetaan esimerkiksi turvallisuusriskeille.

Kun tekoälyä tuodaan olemassa olevaan järjestelmään eikä organisaatiolla ole aiempaa kokemusta tekoälystä, yksi riskienhallintatapa voi olla tuoda tekoäly ensin mukaan optimointikeinona tai esimerkiksi käytettävyyden parantajana. Jos tekoäly ei toimikaan, tuloksena on tällöin lähinnä järjestelmän toiminnan lievä huonontuminen eikä esimerkiksi täydellinen toimimattomuus.

Koneoppimisjärjestelmien elinkaarelle tyypillisen syklisyyden (ks. kappale Koneoppimisjärjestelmän elinkaari) vuoksi arkkitehtuurin ja järjestelmäsuunnittelun riskianalyysiä voi olla tarpeen tehdä iteratiivisesti.

Tietoturvan uhkamallinnusmenetelmiin on usein pyritty lisäämään elementtejä, joilla voitaisiin löytää myös tietosuojariskejä. Ajatuksena on, että tietosuojariskien tunnistamista tehtäisiin aiemmin mainitun tietosuojavaikutusten arvioinnin (DPIA/PIA) tason lisäksi myös teknisemmällä tasolla. Koneoppimisen pohjautuessa data-analyysiin tietosuoja-asioiden lisääminen uhkamallinnukseen voi olla hyödyllistä. Esimerkiksi Microsoftin suositun datavirtojen analyysissä käytettävän STRIDE-uhkamallinnuskehikon (31) oheen on ehdotettu LINDDUN- (31) ja TRIM (32) -nimisiä tietosuojariskien tunnistamisen malleja. On täysin mahdollista lisätä myös muita koneoppimiseen liittyviä riskejä tietoturvan uhkamallinnukseen vastaavalla tavalla.

Testauksen ja tuotantoonviennin taso

Koneoppimismallien suorituskyvyn ja tietoturvan testaaminen on haastavaa (ks. mallin toiminnallisuudesta varmistuminen kappaleessa Hyökkäyspinta ja muuttuvan toimintaympäristön tuomat haasteet kappaleessa Koneoppimisjärjestelmän elinkaari). Perinteisessä ohjelmistokehityksessä käytettyjä staattisen analyysin[9] menetelmiä ei nykyisellään voida käyttää mallien oikeellisuuden tutkimiseen. Testauksen on käytännössä oltava dynaamista, jolloin mallille annetaan testidatan syötteitä ja mallin ratkaisuja verrataan odotettuun toimintaan.

Testitulokset eivät kerro absoluuttisia totuuksia siinä mielessä, että se takaisi mallin olevan aina oikeassa. Testitulos ilmaisee järjestelmän oikean toiminnan tietyssä määrässä tapauksia. Perinteisenkin ohjelmistokehityksen puolella vähänkään monimutkaisemman ohjelman oikeaksi todistaminen on toki myös vaikea tehtävä, joskin tätä ideaalia lähemmäksi voidaan ihmisten kirjoittamassa ohjelmakoodissa päästä staattisella kuin dynaamisella analyysillä.

Testidatan valinta voi myös olla ongelmallista. Koska koneoppimista käytetään tyypillisesti tapauksissa, joissa syötteiden rakennetta ei voida tiukasti määritellä (ks. kappale Hyökkäyspinta), koneoppimismallin mahdollinen syöteavaruus on paljon suurempi kuin oikeiden syötteiden kirjo. Testidata on usein opetusdatasta testaustarkoituksiin ”säästetty” osa, joka saattaa myös - erityisesti ajan mittaan - alkaa poiketa todellisen maailman datasta, jolla koneoppimismallia käytetään. Testidataa on siis päivitettävä ympäröivän maailman muuttuessa, jotta mallin suorituskykyä ja turvallisuutta voidaan luotettavasti mitata.

Perinteisen ohjelmistokehityksen tietoturvatestausta voidaan jossain määrin sitoa vaatimusten tai jopa tuotantoonviennin aikatauluihin. Mikäli tuotteen tehtävälista ja muutosten hallinta on hyvin hoidettu, tuotteeseen ei pitäisi ilmestyä tuntematonta toiminnallisuutta itsestään. Koneoppimisjärjestelmissä, joita opetetaan jatkuvasti osana järjestelmän toimintaa, tämä ei välttämättä enää pidäkään paikkaansa vaan koneoppimismalli saattaa oppia toimimaan aiemmasta eroavalla tavalla ”omin päin”. Mikäli tällaista jatkuvasti oppivaa järjestelmää suunnitellaan, myös sen testauksen jatkuvatoimisuus tulisi varmistaa.

Poikkeamien hallinnan taso

Jos organisaatiolla on ajantasainen poikkeamien hallinnan prosessi, todennäköisesti se kelpaa tekoälyjärjestelmiin varsin hyvin.

Tekoälyjärjestelmien poikkeamat (*incidents*) voivat johtua hyökkäysten lisäksi myös käytettyjen mallien vanhenemisesta tai niiden väärinkäytöstä (33). Mallien vanheneminen tai väärinkäyttö ei välttämättä ole tietoturvapoikkeama, mutta niiden vaikutus saattaa olla sama kuin tietoturvapoikkeaman. Lisäksi, koska hyökkäysdataa ja tavallista syötedataa ei syöteavaruuden laajuuden vuoksi myöskään voi välttämättä erottaa toisistaan, hyökkäyksen tunnistaminen monitorointijärjestelmällä voi olla vaikeampaa.

Syytä poikkeamien hallinnan prosessien kehittämiseen voi olla, jos tuotteiden toiminnallisten poikkeamien ja tietoturvapoikkeamien hallinta ovat esimerkiksi eri prosessit, joista ovat vastuussa eri organisaatiot. Tuotteiden toiminnan monitorointi voi myös olla alue, jota ei välttämättä tehdä tietoturva mielessä.

9 Staattinen analyysi (*static analysis*) tarkoittaa ohjelmakoodin, yleensä lähdekoodin, arviointia silloin, kun se ei ole ajossa. Staattinen analyysi vaihtelee yksinkertaisesta syntaktisesta eli kieliopin tarkastuksesta (*linting*) syötteiden simuloituun seuraamiseen ohjelmakoodin läpi (*taint checking*). Koneoppimismallien analyysissä ei kumpikaan näistä strategioista toimi halutulla tavalla. Staattisen analyysin vastapari on dynaaminen analyysi (*dynamic analysis*), jossa ohjelma on ajossa ja sen käyttäytymistä eri syötteillä voidaan tarkkailla.

Riskien itsearviointityökalu

Oheinen tekoälyn käytön riskien itsearviointityökalu on tarkoitettu koostamaan tämän katsauksen huomioita yhteen. Arviointityökalu koostuu sarjasta kysymyksiä ja niitä vastaavista mahdollisista huomioista, jotka ohjaavat käyttäjää oikeaan suuntaan riskien tunnistamisen ja hallinnan osalta.

Työkalu nostaa ensin esille yleisempiä riskejä ja vasta sitten teknisempiä kysymyksiä. Näin jonkin yksittäisen teknisen riskin ilmetessä on helpompi sanoa, onko sillä järjestelmälle mitään merkitystä. Jos esimerkiksi järjestelmällä ei ole mitään riskejä, joihin luottamuksellisuuden menetys johtaisi, ei myöskään luottamuksellisuuteen liittyvistä teknisistä riskeistä välttämättä tarvitse huolestua.

On tärkeä huomata, että työkalu ei pysty johdattamaan käyttäjäänsä uudenlaisen sovellusalueen monimutkaisten systeemisten riskien löytöprosessin läpi.

Itsearviointityökalu

Voisiko tekoälyjärjestelmän tekemä päätös tai toimintahäiriö aiheuttaa

- laki- tai sopimusrikkomuksen
- ei-halutun oikeusvaikutuksen
- riskin fyysiselle tai henkiselle terveydelle (esimerkiksi jos ohjelmisto on turvakomponentti)
- riskin henkilön turvallisuudelle (esimerkiksi työturvallisuudelle)
- riskin henkilön fyysiselle koskemattomuudelle tai yksityisyydensuojalle
- riskin ympäristölle (esimerkiksi jos ohjelmisto on turvakomponentti)
- negatiivisen vaikutuksen ihmisten välisiin sosiaalisiin suhteisiin
- riskin henkilöiden (esimerkiksi EU:n perusoikeuskirjassa mainituille) oikeuksille ja vapauksille?

Nämä ovat esimerkkejä korkean tason systeemisistä vaikutuksista, joiden huomioon ottaminen toki on tarpeen muidenkin kuin tekoälyjärjestelmien suhteen.

Tekoälyjärjestelmien yhteydessä nämä asiat ovat sääntelyssä pinnalla, koska koneoppimisjärjestelmän suunnittelussa ei mallin tasolla välttämättä edes pyritä sataprosenttiseen oikeellisuuteen ja koska päätöksen perusteiden selittäminen saattaa olla hankalaa.

Jos jokin näistä on mahdollinen, seuraavien kysymysten kannalta on hyödyllistä miettiä, miksi se on mahdollinen. Onko syy tekninen (liittyy esimerkiksi luottamuksellisuuteen, eheyteen tai saatavuuteen), vai onko syy rakennettu sisään käyttötapauksiin, jolloin syy voi olla systeeminen tai välillinen?

Mitä riskienhallintaelementtejä tekoälyjärjestelmän kehitysprosessissa on?

- Systeemisten vaikutusten arviointi
- Tietosuojavaikutusten arviointi (DPIA/PIA), jos käytetään henkilötietoja
- Arkkitehtuurin riskianalyysi, uhkamallinnus tai design-katselmointi
- Alustojen ja ajoympäristöjen turvallisuuden arviointi tai tarkastus
- Järjestelmän ja rajapintojen tietoturvatestaus
- Poikkeamien hallinnan suunnittelu

Olemassa olevan turvallisen tuotekehitysprosessin aktiviteetteja voi lähestyä tekoälyn erityispiirteiden kautta ja niihin voi olla hyödyllistä tehdä lisäyksiä tai muutoksia (ks. kappale Tekoälyn riskien hallinta tuotekehitysprosessissa).

Jos tuotekehitysprosessissa ei vielä tehdä tietoturva-aktiviteetteja, sovellusalueesta riippuen näiden suunnittelua kannattaa harkita.

Näiden korkeamman tason ja riskienhallinnallisten pohdintojen jälkeen on helpompi nähdä metsä puilta teknisissä kysymyksissä:

Käytetäänkö tekoälyn kehityksessä

- Dataa, joka on relevanttia, korkealaatuista ja kattavaa
- Selkeää ja oikeasuhtaista pääsynhallintaa dataan
- Menetelmiä, joilla seurataan tekoälyn suorituskykyä (tarkkuus, herkkyys, täsmällisyys)
- Menetelmiä, joilla seurataan, onko tekoälyn toimintaympäristö edelleen riittävän samanlainen kuin mihin järjestelmä kehitettiin?

Nämä kysymykset eivät liity pelkästään tietoturvaan, mutta näitä periaatteita seuraamalla tekoälyjärjestelmästä tulee vikasietoisempi ja tätä kautta turvallisempi.

Mitä datalähteitä käytetään opetukseen ja järjestelmän syötteinä?

Esimerkkejä:

- Kokonaan organisaationne sisäisesti tuotettua dataa
- Asiakkailta saatua dataa
- Asiakkaiden käyttäytymisestä kerättyä dataa
- Ulkoisista lähteistä saatua luottamuksellista dataa
- Julkista dataa

Ulkopuolelta saatu data voi altistaa koneoppimismallin hyökkäyksille (ks. Riskit).

Jos opetusdataa luokitellaan (*labelling*), miten se tapahtuu?

Esimerkkejä:

- Sisäisesti ihmistyönä
- Sisäisesti koneellisesti
- Ulkoisesti asiakkaiden toimesta
- Ulkoisesti alihankittuna
- Ulkoisesti tuntemattomien toimesta (crowdsourcing)

Ulkopuolinen luokittelu voi altistaa koneoppimismallin hyökkäyksille (ks. Riskit). Koneellinen luokittelu voi siirtää koneen virheet malliin.

Käytetäänkö koneoppimismallin pohjana ennalta opetettuja malleja?

- Sisäistä tai luotettavasta lähteestä hankittua mallia
- Julkisesta lähteestä kuten mallikirjastosta saatua mallia

Ennalta opetettu koneoppimismalli voi altistaa koneoppimismallin hyökkäyksille (ks. Riskit).

Onko koneoppimismallissa tai opetusdatassa luottamuksellista dataa?

- luottamuksellista dataa (esimerkiksi salassa pidettävää dataa, liikesalaisuuksia)
- tekijänoikeudella suojattua dataa
- henkilötietoja
- dataa, jolla on itsessään kaupallista arvoa

Arkkitehtuurisuunnittelussa ja tietoturvasuunnittelussa kannattaa ottaa huomioon luottamuksellisuuteen liittyvien hyökkäysten riski ratkaisuvaihtoehtoja (ks. kappaleet Koneoppimismallin luottamuksellisuus, Yleisimpien riskien ehkäisy ja Poikkeamien hallinnan taso).

Pidättekö kirjaa, mistä datalähteistä mikin data tai sen luokitus (*label*) on peräisin?

- Datan/luokituksen alkuperä on todennettu
- Datalähde ja data/luokitus on kirjanpidollisesti yhdistettävissä
- Dataa/luokitusta ja datalähdettä ei voida jälkeenpäin yhdistää

Mikäli jokin datalähde tai luokittelija osoittautuu jälkikäteen ongelmalliseksi (hyökkääjäksi), sieltä tulleen datan poistaminen ja mallin uudelleen luonti vaatii jonkinlaisen kirjanpidon datalähteistä.

Onko mietitty, mitä tapahtuu, jos tekoälysovellus tai sen käyttämä malli

- lopettaa toimimisen (esimerkiksi vastaamisen) kokonaan
- hidastuu merkittävästi
- alkaa tehdä merkittävässä määrin vääriä päätöksiä?

Suunnittelussa kannattaa ottaa huomioon eheys- ja saatavuusriskit sekä näiden ehkäisemisen vaihtoehdot (ks. kappaleet Koneoppimismallin ja opetusdatan eheys, Koneoppimismallin saatavuus, Yleisimpien riskien ehkäisy ja Poikkeamien hallinnan taso). Jos hyväksyttävien virheiden määrää on vaikea määritellä tai virheitä edes huomata, tämä on erityisesti koneoppimisen sovellusten suhteen olennaista selvittää.

Käytetäänkö datan käsittelyyn tai muuhun tekoälyjärjestelmän luomiseen ulkopuolisia ohjelmistoja tai kirjastoja?

- Suljettua koodia
- Valmispalveluita (SaaS, PaaS, MLaaS)
- Avointa lähdekoodia

Ulkopuolinen koodi tuo tullessaan toimitusketjuhyökkäyksen (supply chain attack) riskit. Riski riippuu siitä, mihin ohjelmistoa käytetään ja missä sitä ajetaan. Datan kannalta riski voi olla esimerkiksi datan luottamuksellisuudelle tai saatavuudelle (mm. kiristyshaittaohjelmat).

Valmispalvelut voivat mahdollisesti altistaa jatkuvuusriskille, jos palvelu esimerkiksi aiheuttaa toimittajaloukun ja lakkaa toimimasta.

Kuinka usein koneoppimismallia opetetaan uudelleen?

- Jatkuvasti (online)
- Erikseen tarvittaessa
- Kehitysprosessin tai kalenterin mukaan

Mallin validointi ja testaus sekä sen suorituskyvyn seuraaminen saattaa olla tarpeen järjestää samalla strategialla ja erityisesti huomioida jatkuvan oppimiksen aiheuttamat vaatimukset. Ks. kappale Testauksen ja tuotantoonviennin taso.

Onko datan (oppimisdatan, kyselyiden ja esimerkiksi sensoridatan) keruun, siirron ja tallennuksen osalta uskottava tarina

- luottamuksellisuus
- eheys
- saatavuus

Yksi menetelmä on piirtää datan matka tietovirtakaaviona (data flow diagram) ja selittää jokaiselle datavirralle tekninen argumentti, miksi kukin näistä piirteistä on kunnossa. Jos tarinan artikulointi tai kaavion piirto on vaikeaa, se vihjaa, että sitä ei ehkä ole täysin vielä mietitty.

Missä muodossa data esitetään tekoälyjärjestelmälle?

- Raakana tai strukturoimattomana
- Jollakin tavalla ennakkoon jäsenneltynä tai suodatettuna

Datan jäsentäminen tai suodattaminen voi olla tietoturvaominaisuus (ks. kappale Tietoturvaominaisuudet), mutta jäsentämisessä voi myös olla perinteisiä tietoturvahaavoittuvuuksia. Erityisesti mallinväistöhyökkäystä voi olla vaikeampi tunnistaa raa'asta tai strukturoimattomasta datasta.

Miten kyselyt toimitetaan tekoälyjärjestelmälle?

- Verkon yli avoimeen rajapintaan
- Verkon yli suljettuun rajapintaan
- Paikalliseen rajapintaan (esimerkiksi laitteessa)
- Tekoälyjärjestelmälle annetaan pääsy tietokantaan tai tiedostoihin

Verkon yli saavutettavassa rajapinnassa voidaan tehdä tietoturvaan vaikuttavia toimenpiteitä (ks. kappale Tietoturvaominaisuudet), mutta yleisistä rajapintojen turvallisuusasioista[10] on huolehdittava. Paikallinen rajapinta voi olla esimerkiksi henkilötietojen osalta helpompi ratkaisu, mutta esimerkiksi koneoppimismallin luottamuksellisuuden kannalta hankalampi (ks. kappale Koneoppimismallin luottamuksellisuus). Mikäli tekoälyjärjestelmä saa itse pääsyn toisiin järjestelmiin, tunnusten hallintaan ja toimitusketjuriskeihin täytyy suunnata erityistä huomiota.

Missä muodossa tekoäly palauttaa päätöksensä tai ennusteensa?

- Käyttöliittymän kautta, jonka ulkoasua tai sisältöä ennusteet muokkaavat
- Tuloksina, jotka ovat esimerkiksi generoituja kuvia tai tekstiä
- Ennusteina, jotka ovat binäärisiä luokituksia
- Ennusteina, joissa on mukana todennäköisyys (mahdollisesti eri ratkaisuvaihtoehdoille)

Mitä monipuolisempaa tietoa tekoälyn ennusteista tai päätöksistä hyökkääjä voi saada, sitä helpommaksi monet luottamuksellisuuteen ja mallin väistämiseen tähtäävät hyökkäykset muuttuvat. Ks. kappale Riskit.

10 Rajapintojen turvallisuuteen liittyvät mm. tiedonsiirron salaus, palvelunestohyökkäysten huomioon ottaminen (mm. kuormantasaus, skaalautuminen, alueellisesti suunnatut palvelut), rajapinnan vikasietoisuus viallisia kutsuja vastaan, rajapinnan toiminnan valvonta ja auditointilokitus, rajapinnan todentaminen kutsujille ja suljetuissa rajapinnoissa kutsujan todennus ja valtuutus.

Viitteet

(1) Melanie Mitchell. *Why AI is Harder Than We Think*. arXiv:2104.12871v2. 2021. https://arxiv.org/pdf/2104.12871.pdf

(2) Euroopan komissio. *COM(2021) 206 final: Ehdotus Euroopan parlamentin ja neuvoston asetukseksi tekoälyä koskevista yhdenmukaistetuista säännöistä (tekoälysäädös) ja tiettyjen unionin säädösten muuttamisesta.* https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:52021PC0206. 2021.

(3) Liikenne- ja viestintäministeriö. *Charting Regulatory Frameworks for Maritime Autonomous Surface Ship Testing, Pilots, and Commercial Deployments.* 2020. http://urn.fi/URN:ISBN:978-952-243-610-8

(4) Euroopan unionin perusoikeusvirasto (FRA). *Getting The Future Right: Artificial Intelligence and Fundamental Rights*. 2020.

(5) Ethik-Kommission Automatisiertes und Vernetztes Fahren, Bundesministerium für Verkehr und digitale Infrastruktur. A report. 2017. https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/bericht-der-ethik-kommission.pdf.

(6) Teknologiateollisuus. *AI Act–Focus on the Process and Predictability*. 2021. https://teknologiateollisuus.fi/sites/default/files/inline-files/One_pager%20AI%20Act_May2021.pdf

(7) Anna Jobin, Marcello Ienca, Effy Vayena. *The global landscape of AI ethics guidelines.* Nature Machine Intelligence, 1 (9), 389–399. 2019. https://doi.org/10.1038/s42256-019-0088-2

(8) Jessica Fjeld, Nele Achten, Hannah Hilligoss, Adam Nagy, Madhulika Srikumar. *Principled Artificial Intelligence: Mapping Consensus in Ethical and Rights-Based Approaches to Principles for AI.* Berkman Klein Center Research Publication No. 2020-1. 2020. http://doi.org/10.2139/ssrn.3518482

(9) *Aurora AI -ohjelman etiikka- ja yhteiskuntavastuutyöryhmän 1. väliraportti.* 2021. https://auroraaifin.slack.com/files/UCJ3TDEKS/F01SNE53F6V/auroraain_etiikkaryhm__n_v__liraportti__marraskuu_2020_-_maaliskuu_2021_.pdf

(10) High-Level Expert Group on Artificial Intelligence (AI HLEG). *Ethics Guidelines for Trustworthy AI.* 2019. https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60419

(11) Defense Innovation Board, Department of Defense (United States). *AI Principles: Recommendations on the Ethical Use of Artificial Intelligence by the Department of Defense*. 2019
.

(12) Euroopan parlamentin ja neuvoston asetus 2019/881. *Euroopan unionin kyberturvallisuusvirasto ENISAsta ja tieto- ja viestintätekniikan kyberturvallisuussertifioinnista sekä asetuksen (EU) N:o 526/2013 kumoamisesta (kyberturvallisuusasetus).* https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32019R0881

(13) *Euroopan parlamentin ja neuvoston asetus 2016/679 luonnollisten henkilöiden suojelusta henkilötietojen käsittelyssä sekä näiden tietojen vapaasta liikkuvuudesta ja direktiivin 95/46/EY kumoamisesta (yleinen tietosuoja-asetus).* https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32016R0679

(14) Article 29 Data Protection Working Party. *WP216: Opinion 05/2014 on Anonymisation Techniques.* 2014. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

(15) European Commission and the European Multi-Stakeholder Platform on ICT Standardisation. *Rolling Plan for ICT Standardisation: Artificial Intelligence*. https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/artificial-intelligence. 2021.

(16) UNECE World Forum for Harmonization of Vehicle Regulations (WP.29). *Proposal for a new UN Regulation on uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system.* https://unece.org/DAM/trans/doc/2020/wp29grva/ECE-TRANS-WP29-2020-079-Revised.pdf. 2020.

(17) ISO/SAE 21434:2021. *Road vehicles — Cybersecurity engineering*. https://www.iso.org/standard/70918.html. 2021.

(18) International Maritime Organization (IMO). *Interim Guidelines for MASS Trials*. 2019. https://wwwcdn.imo.org/localresources/en/MediaCentre/HotTopics/Documents/MSC.1-Circ.1604%20-%20Interim%20Guidelines%20For%20Mass%20Trials%20(Secretariat).pdf

(19) International Maritime Organization. *Guidelines on Maritime Cyber Risk Management*. 2017. https://wwwcdn.imo.org/localresources/en/OurWork/Security/Documents/MSC-FAL.1-Circ.3%20-%20Guidelines%20On%20Maritime%20Cyber%20Risk%20Management%20(Secretariat).pdf

(20) European Maritime Safety Agency (EMSA). SAFEMASS: *Study of the risks and regulatory issues of specific cases of MASS*. 2020. http://emsa.europa.eu/we-do/safety/ship-safety-standards/item/3892-safemass-study-of-the-risks-and-regulatory-issues-of-specific-cases-of-mass.html

(21) DNV-GL. *Rules for Classification: Ships. Part 6: Additional Class Notations. Chapter 5: Equipment and design features.* 2018. https://rules.dnv.com/docs/pdf/DNV/RU-SHIP/2018-07/DNVGL-RU-SHIP-Pt6Ch5.pdf

(22) IEC 62443-standardisarja. https://webstore.iec.ch/searchform&q=62443

(23) EUROCAE. Työryhmän WG-114 (Artificial Intelligence) kuvaus. https://eurocae.net/about-us/working-groups/

(24) European Union Aviation Safety Agency (EASA). *Artificial Intelligence Roadmap: A human-centric approach to AI in aviation*. 2020. https://easa.europa.eu/ai

(25) EASA AI Task Force. *Concepts of Design Assurance for Neural Networks (CoDANN) II*. 2021. https://www.easa.europa.eu/sites/default/files/dfu/ddln_easa_codann2_public.pdf

(26) European Aviation Artificial Intelligence High Level Group. *The FLY AI Report: Demystifying and Accelerating AI in Aviation/ATM*. 2020. https://www.eurocontrol.int/sites/default/files/2020-03/eurocontrol-fly-ai-report-032020.pdf

(27) Department of Health & Social Care, UK. *A guide to good practice for digital and data-driven health technologies*. https://www.gov.uk/government/publications/code-of-conduct-for-data-driven-health-and-care-technology/initial-code-of-conduct-for-data-driven-health-and-care-technology. 2021.

(28) Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, Colin Raffel. *Extracting Training Data from Large Language Models*. arXiv:2012.07805v2. 2021. https://arxiv.org/pdf/2012.07805.pdf

(29) Ian Goodfellow, Jonathon Shlens, Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv:1412.6572v3. https://arxiv.org/pdf/1412.6572.pdf

(30) Gary McGraw, Harold Figueroa, Victor Shepardson, Richie Bonnett. *An Architectural Risk Analysis of Machine Learning Systems: Toward More Secure Machine Learning*. 2020. https://berryvilleiml.com/docs/ara.pdf

(31) Adam Shostack. *Threat Modeling: Designing for Security.* Wiley. 2014.

(32) F-Secure. *Elevation of Privacy: Privacy Cards for Software Developers*. 2019. https://github.com/F-Secure/elevation-of-privacy

(33) Patrick Hall, Andrew Burt. *What to Do When AI Fails*. 2021. O'Reilly. https://learning.oreilly.com/videos/meet-the-expert/0636920557937/

Liikenne- ja viestintävirasto Traficom
Kyberturvallisuuskeskus
PL 320, 00059 TRAFICOM
p. 029 534 5000
traficom.fi

Liikenne- ja viestintävirasto
Kyberturvallisuuskeskus