Liite: Tekoäly automaation tietoturvan kehittämisessä

Pietari Sarjakivi, SVP Labs, Nixu

Tekoäly parantaa tehokkuutta ja muuttaa meidän tapaamme tehdä työtä. Sen vaikutukset ulottuvat lähes jokaiselle toimialalle muun muassa robotiikan, älykkäiden hakutoimintojen, konenäön ja luonnollisen kielen käsittelyn kehittymisen myötä. Automaation tietoturvassa tekoälyllä voidaan jo tänä päivänä tehostaa poikkeamantunnistamista, sekä ei-aktiivisesta toimijasta johtuvan käyttökatkojen ennustamista. Edellä mainituista esimerkeistä huolimatta suoranaista kannattavuusloikkaa ei tekoälyn kehityksellä olla toistaiseksi saavutettu automaation tietoturvan saralla. Tästä huolimatta tekoälyn mahdollisuuksien ymmärtäminen on tärkeää, jotta ympäristöjä pystytään puolustamaan tehokkaasti myös tulevaisuudessa, kun sekä suojeltava ympäristö että hyökkääjä käyttää tekoälyä. Automaatiojärjestelmien rakentajien tulee varautua tekoälyn tuomiin muutoksiin, sillä heidän rakentamillaan järjestelmillä oletetaan olevan IT-järjestelmiä pidempi elinkaari.

Mitä tekoäly oikeastaan on?

Tämän kappaleen tavoitteena on antaa yleiskäsitys aiheesta ja helpottaa käytännön sovelluksia kuvaavan kappaleen ymmärtämistä. Tekoälyn perusteita voi opiskella tarkemmin esimerkiksi Helsingin Yliopiston ilmaisen Elements of AI kurssin kautta (University of Helsinki & Reaktor, 2020).

Tekoälystä on puhuttu antiikin Kreikan ajoista asti. Vaikka konsepti vakiintui nykyisen kaltaiseksi 1950-luvun tutkijoiden kuten John McCarthy (McCarthy et al., 1955) ja Alan Turingin (Turing, 1950) julkaisujen myötä, on sen käytännön sovellusten kehitys edennyt melko hitaasti ja aaltoilevasti. Merkittäviä tekoälyn käytännön sovellusten kehitykseen liittyviä tapahtumia ovat olleet muun muassa IBM Deep Bluen voitto shakissa suurmestari Garri Kasparovia vastaan vuonna 1997 (Korf, 1997) sekä Googlen AlphaGon voitto strategiapeli GOssa lajin ylivoimaista mestaria Lee Sedolia vastaan vuonna 2016 (Silver et al., 2016). Jos tarkastelemme tekoälysovellusten ominaisuuksia edellä mainittujen esimerkkien kautta, havaitsemme hyvin tekoälyyn liittyvät rajoitukset.

Ensinnäkin molemmat esimerkit olivat pelejä, joissa on selkeät säännöt ja kenttä, jotka luovat rajat sovellukselle. Vaikka pelin sisällä on huomattava määrä vaihtoehtoja ja tapoja pelata, ei sovellus voi hyödyntää pelin ulkopuolisia keinoja tai osaa tulkita ottelun merkitystä isommassa kuvassa ja vaikkapa geopoliittisessa kontekstissa. Voidaan siis sanoa, että molemmat ovat kapean eli heikon tekoälyn (Narrow AI) sovelluksia. Mikäli kyseessä olisi vahva tai yleinen tekoäly (Artificial Generic

Intelligence, AGI), pystyisi se selviytymään monenlaisista ja yllättävistä tehtävistä, joihin ei voida asettaa kattavia sääntöjä, vaan sovelluksen pitäisi itse pystyä päättelemään historian ja kulttuurilliseen tuntemukseen pohjautuen, miten tilanteessa tulee toimia.

Toisena huomiona on sovelluksen tapa oppia. Esimerkiksi DeepBlue ratkaisun on väitetty pohjautuvan ihmisen kirjoittamiin sääntöihin, eikä näin ollen olisi tekoälyä sanan varsinaisessa merkityksessä. Toisaalta se on hyvä osoitus, että tehokas automatisointi hyvällä sisäänrakennetulla tai käyttäjän opettamalla logiikalla voi saavuttaa loistavan lopputuloksen. Toinen esimerkki, AlphaGO, hyödynsi koneoppimista (Machine Learning) tai tarkemmin ottaen neuroverkkopohjaista (Artificial Neural Network, ANN) syväoppimista (Deep Learning), ollen näin oppikirjaesimerkki siitä, mitä käsitämme tekoälyllä tänä päivänä. Koneoppiminen jaetaan tyypillisesti kolmeen kategoriaan sen oppimismenetelmän mukaisesti: Ohjattu oppiminen (Supervised Learning), Ohjaamaton oppiminen (Unsupervised Learning) ja Vahvistusoppiminen (Reinforcement Learning). Yhteistä näille malleille on, että opetusdataa tarvitaan paljon. Näin ollen maailmassa kerätyn datan määrän huima kasvu ja laskentatehon lisääntyminen ovat osaltaan vauhdittaneet koneoppimisen ja neuroverkkojen kehitystä.

Useimmat tuntemamme tekoälysovellukset ovatkin tarkemmin ottaen koneoppimista. Humoristisesti yleistäen voidaankin sanoa, että koneoppiminen tehdään Pythonilla ja tekoäly Powerpointilla.

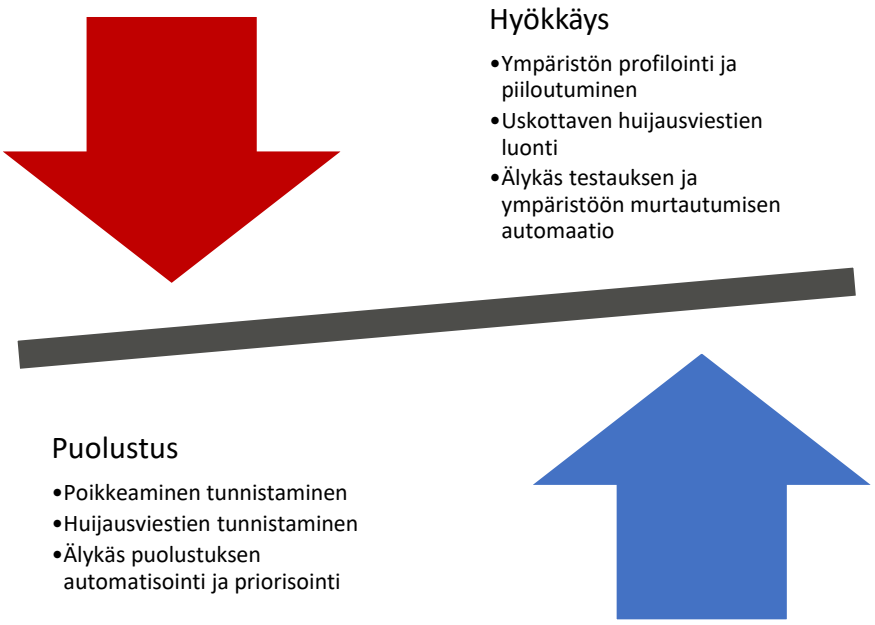
Tekoälyn sovelluksia?

Tekoälyn käyttö ei ole itseisarvo, vaan sillä tulisi parantaa jonkin tärkeän prosessin tai toiminnallisuuden suorituskykyä. Tästä syystä on erityisen tärkeää tietää, millaisiin sovelluksiin tämän päivän tekoälyratkaisut kykenevät. Yksi tapa lähestyä asiaa on katsoa tutkimus- ja konsultointiyritys Gartnerin tuottamaan hype-käyrää, joka on esitelty kuvassa 1 (Gartner, 2020).

Kuva 1 Gartner Hype Cycle for Artificial Intelligence, 2020

Vaikka hype-käyrältä löytyykin monta Industry 4.0 muutosta tukevaa asiaa, kuten konenäkö (Computer Vision) ja älykäs robotiikka (Smart Robotics), useat listatut asiat ovat vasta konseptitasolla, eivätkä niiden sovellukset ole vielä vakiintuneet. Tekoälyn hype-käyrällä onkin listattu enemmän tekniikoita kuin varsinaisia sovelluksia. Useat yritykset ja yhteisöt panostavat tekoälyn ja siihen liittyvien sovellusten kehitykseen merkittävästi, joka osaltaan kiihdyttää kehitystä lisää. Juuri kehityksen vauhti ja tekoälyllä tavoiteltujen asioiden vaikuttavuus ovat isossa mittakaavassa niin merkittäviä asioita, että mikään toimia-ala ei pysty tulevaisuudessa välttymään tältä muutokselta.

Tekoälyn kehitys voidaan tietoturvan näkökulmasta katsoa olevan kaksiteräinen miekka, kuten Kuva 2 esittää. Toisaalta se mahdollistaa puolustajan nopeamman ja älykkään toiminnan kustannustehokkaasti, jolloin asiantuntijat voivat keskittyä rutiinitoimenpiteiden sijasta kehittäviin toimenpiteisiin. Toisaalta myös hyökkääjä voi tehostaa omaa toimintaansa käyttäen samoja menetelmiä kuin puolustaja. Lisäksi hyökkääjä voi kohdistaa hyökkäyksensä puolustajan tekoälyratkaisuihin, heikentäen puolustuskykyä huomaamattomasti. Erityisen tehokkaan tästä hyökkäyksestä tekee tekoälyratkaisuille luonteenomainen tehtyjen päätösten vaikeasti tarkistettava päättelyketju. Hyökkääjän ja puolustajan roolit menevät myös osittain ristiin, sillä puolustajan täytyy pystyä näkemään oma ympäristönsä myös hyökkääjän silmin, kun taas hyökkääjän tulee jatkuvasti puolustaa omaa operaatiotaan. Joidenkin maiden lainsäädäntö sallii myös puolustuksen aktiiviset vastatoimet ja niin sanotun HackBackin.



Kuva 2 Yleisimmät tekoälyn puolustukselliset ja hyökkäykselliset sovellukset kyberturvassa

Koneoppimisen hyödyt tulevat erityisen hyvin esille poikkeamien tunnistamisessa, jossa suhteellisen muuttumattomassa ympäristössä on paljon tapahtumia, eli opetusmateriaalia, ja käytössä on paljon laskentatehoa. Nämä ehdot täyttyvät erityisen hyvin automaatioympäristöjen verkkoliikenteen valvonnassa, jossa itsenäinen sensori seuraa esimerkiksi verkkoliikenneprofiilin muutosta, sekä verkossa olevia laitteita ja palveluita. Poikkeamat verkkoliikenteessä eivät aina tarkoita hyökkäystä tai muuta aktiivisen toimijan tekemään toimenpidettä, vaan voivat paljastaa ympäristön saatavuuden kannalta merkittäviä alkavia ongelmatilanteita. Koneoppimiseen pohjautuvaa käyttäytymisprofiilia voidaan muodostaa myös päätelaitteista ja järjestelmistä kerättyjen lokien perusteella, mutta näissä molemmissa on omat haasteensa. Päätelaitteella samankaltaisia tapahtumia on melko paljon, mutta turvallisuuteen käytettävissä oleva laskentateho on rajallinen. Tämä on erityisen haastava ongelma automaatioympäristöissä, joissa saatavuutta ei voida vaarantaa päätelaitteen liiallisen kuormituksen takia. Lokien keräämiseen ja keskittämiseen pohjautuvissa järjestelmissä laskentakapasiteettia on helppo lisätä, mutta tarpeellinen määrä riittävän muuttumattomia tapahtumia on vaikeampi saavuttaa. Molemmissa tapauksissa käyttötapaus voidaan rajoittaa vielä kapeammaksi ja keskittyä esimerkiksi käyttäjän kirjautumiseen liittyvään dataan. Yleisesti ottaen puhtaasti koneoppimiseen perustuva poikkeamienhallinta ei vielä ole riittävällä tasolla, vaan parhaan mahdollisen tuloksen saavuttamiseksi sitä tulee vahvistaa uhkatietoa tai tunnettuja hyökkäysmenetelmiä tunnistavalla sääntöpohjaisella havainnoinnilla.

Hyökkääjät pystyvät vaikuttamaan koneoppimiseen pohjautuvaan poikkeamien tunnistukseen ainakin kahdella keinolla. He voivat koettaa opettaa puolustusjärjestelmän tunnistamaan heidän liikenteensä osana normaaliksi luokiteltua liikennettä, jolloin heidän toimistaan ei nouse poikkeamia. Toisaalta he

voivat itse oppia operoimaan normaaliksi luokiteltujen toimien puitteissa, välttäen näin ollen havaituksi tulemisen. Jotta hyökkääjät tietävät mitä luokitellaan normaaliksi, tulee heidän pystyä profiloimaan ympäristöä tai päästä käsiksi profiilitietoon, esimerkiksi murtautumalla puolustusjärjestelmiin. Mikäli turvajärjestelmiin päästään murtautumaan, on ympäristön suojaaminen lähtökohtaisesti huomattavan haastavaa. Profilointiin ja hyökkäysliikenteen ja hyökkäyskoodin kätkemiseen on olemassa useita erilaisia ilmaisiakin työkaluja jo tänä päivänä.

Koneoppimisen yksi laajimmalle levinnyt käyttötapaus tietoturvan alalla on sähköpostien roskapostisuodatus, jossa käytännössä luokitellaan viestit yksinkertaisesti roskaposteiksi ja asiallisiksi viesteiksi pohjautuen suodattimen oppimiin malleihin. Samaa logiikkaa käyttää osa muunkinlaisistakin huijauksen havainnointityökaluista. Tässäkin käyttötapauksessa parhaan lopputuloksen saa jakamalla uhkatietoa järjestelmien välillä.

Suodatuksen kehittymisen myötä hyökkääjät joutuvat panostamaan huijausviestiensä laatuun. Erityisesti tällä alalla tekoäly on tuonut huomattavasti uusia työkaluja hyökkääjälle, kun luonnollisen kielen käsittely (NLP) ja generatiiviset kilpailevat verkostot (GAN) ovat luoneen hyökkääjälle mahdollisuuden tuottaa toisen henkilön kirjoitusasua muistuttavia viestejä sekä erittäin todenmukaisia kuva-, ääni- ja videoväärennyksiä (Deep Fakes).

Älykäs ja oppiva toiminnan automaatio tehostaa niin puolustajan kuin hyökkääjänkin toimia. Esimerkkinä tästä mainittakoon Security Orchestration, Automation & Response (SOAR) työkaluihin rakennetut pelikirjat ja automatisoidut työvuot, joiden avulla puolustaja pystyy esimerkiksi rikastuttamaan käsiteltävän tietoturvapoikkeaman tietoja älykkäillä verkkohauilla ja historiakorreloinneilla. Lisäksi SOAR-työkalut pystyvät priorisoimaan poikkeamia oppimansa perusteella, jolloin puolustajan ajankäyttö tehostuu.

Toinen esimerkki älykkään automaation kehittymisestä löytyy ohjelmistojen ja ympäristöjen testaamisesta eli haavoittuvuuksien löytämisestä. Perinteisten dynaamisten ja staattisten ohjelmistotestaustyökalujen lisäksi yhä useampi työkalu pystyy hyökkäämään kokonaista ympäristöä vastaan älykkäästi tai pyrkii etsimään reittiä ulos suljetusta ympäristöstä. Testausautomaation kehittyminen tukee hyvin modernia ohjelmistokehitystä, jossa koodin kirjoittamisesta tuotantoon siirtymisen aikaväli kutistuu myös turvakriittisillä aloilla. Uusien haavoittuvuuksien lisäksi jotkut testaustyökalut osaavat etsiä lähdekoodin seasta tunnetusti haavoittuvia koodikirjastojen versioita ja koodinpätkiä. Kehittyneimmissäkin älykkäissä testausautomaation ratkaisuissa on silti kyse enintään kapean tekoälyn sovelluksista, jotka rajoittavat toimensa niille annettuihin tehtäviin ja sääntöihin.

Keskeisimmät tekoälyyn liittyvät termit

- Artificial Intelligence (AI), Tekoäly. Laaja kattotermi, joka erityisesti puhekielessä kattaa kaikki ohjelmistopohjaiset itsenäiset ja älykkäät ratkaisut.

- Weak / narrow AI, kapea tai heikko tekoäly. Tekoälyn muoto, joka keskittyy selkeästi rajattuun tehtävään. Moni nykypäivän käytännön sovelluksista kuuluu tähän kategoriaan.
- Artificial Generic Intelligence (AGI), vahva tekoäly. Laaja-alainen tekoälyn muoto, jolla on riittävä ymmärrys toimia vähemmän säädellyssä ympäristössä. AGI:n tulee selvitä yllättävistä tehtävistä päättelemällä oikea ratkaisu historialliseen ja kulttuurilliseen tuntemukseen pohjautuen. Tällaiseen tekoälyyn viitataan populaarikulttuurissa, eikä se ole todellisuutta tänä päivänä.
- Machine Learning (ML), koneoppiminen. Tekoälyn yleisin muoto, missä ohjelma oppii sille syötetystä datasta sille määritetyn algoritmin pohjalta ennustamaan tai luokittelemaan asioita.
- Artificial Neural Networks (ANN), neuroverkot. Biologisten neuroverkkojen tietotekninen mallinnus, jolla saadaan aikaan tehokkaita syväoppimisjärjestelmiä. Neuroverkot koostuvat joukoista toisiinsa yhdistettyjä yksinkertaisia elementtejä (neuroneja), jotka ottavat tietoa vastaan yhdestä suunnasta ja lähettävät tietoa toiseen suuntaan. Älykkyys muodostuu yksinkertaistaen neuronien ketjutuksesta ja piilokerroksista, ei yksittäisen neuronin sisältämästä painoarvosta.
- Deep Learning, syväoppiminen. Koneoppimismenetelmiä, joissa algoritmien perusteella luodaan neuroverkkoja monimutkaiseen ongelmanratkaisuun. Useat käytännön sovellukset, kuten puheen- ja kuvantunnistus, pohjautuvat syväoppimiseen.
- Deep fake, syvävääreennys. GAN-tekniikalla toteutettu aidon oloinen väärennös, esimerkiksi muunnelma olemassa olevasta videosta.
- Supervised Learning, ohjattu oppiminen. Yksi kolmesta koneoppimisen perusmenetelmästä. Tässä menetelmässä sovellukselle syötetään dataa ja luokiteltu kuvaus, jonka jälkeen sovellus oppii tunnistamaan samanlaiset. Esimerkiksi, onko kuvassa koira vai ei.
- Unsupervised Learning, ohjaamaton oppiminen. Yksi kolmesta koneoppimisen perusmenetelmästä. Tässä menetelmässä sovellukselle syötetään luokittelematonta dataa, josta sovelluksen tehtävänä on muodostaa ryhmiä samanlaisuuksien perusteella.
- Reinforcement Learning, vahvistusoppiminen. Yksi kolmesta koneoppimisen perusmenetelmästä. Tässä menetelmässä sovellus saa palautteen tekemästään päätöksestä viiveellä, esimerkiksi pelin lopussa. Tämän pohjalta sovellus voi kehittää omaa päätöksentekokykyään niin kauan, kun esimerkkidataa on saatavilla.
- Generative adversarial network (GAN), generatiivinen kilpaileva verkosto. Koneoppimismalli, jossa yleensä kaksi eri algoritmia kehittävät toisiaan, esimerkiksi siten, että toinen piirtää aidon näköisen kuvan ja toinen antaa sille arvion ja syöttää tuloksen takaisin ensimmäiselle algoritmille.
- Natural language processing (NLP), Luonnollisen kielen käsittely. Tekoälyn ja kielitieteiden alalaji, jolla pyritään kehittämään tietokoneen kykyä kommunikoida ihmisen kanssa.

Lähteet

Gartner. (2020). *2 Megatrends Dominate the Gartner Hype Cycle for Artificial Intelligence, 2020*. https://www.gartner.com/smarterwithgartner/2-megatrends-dominate-the-gartner-hype-cycle-for-artificial-intelligence-2020/

Korf, R. (1997). *Does Deep Blue use AI?* AAAI Technical Report, WS-97-04, 2.

McCarthy, J., Minksy, M., Rochester, N., and Shannon, C. (1955*). A Proposal for the Dartmouth Summer Research Project on Artificial intelligence.* Technical report.

Turing, A. M. (1950). *Computing Machinery and Intelligence.* Mind, New Series, 59(236), 433–460.

University of Helsinki, & Reaktor. (2020). *Elements of AI. A Free Online Introduction to Artificial Intelligence for Non-Experts*. https://www.elementsofai.com

...

Tämä on liite kirjaan:
Automaation tietoturva – Kriittisen tuotannon turvaaminen
(ISBN: 978-952-5183-58-0, ISSN 1455-6502, SAS julkaisusarja nro 51, © Suomen Automaatioseura ry, www.automaatioseura.fi/AutomaationTietoturva)