Lauri Hasa

Simuloitu ihminen koneoppimismallin koulutuksessa

aktiivisen oppimisen ja koneopettamisen käyttökelpoisuus satunnaismetsämallissa

Tekniikan ja innovaatiojohtamisen yksikkö
Pro gradu -tutkielma
Tietojärjestelmätiede, Kauppatieteiden maisteri

Vaasa 2026

VAASAN YLIOPISTO
Tekniikan ja innovaatiojohtamisen yksikkö
Tekijä: Lauri Hasa
Tutkielman nimi: Simuloitu ihminen koneoppimismallin koulutuksessa: aktiivisen oppimisen ja koneopettamisen käyttökelpoisuus satunnaismetsämallissa
Tutkinto: Kauppatieteiden maisteri
Koulutusohjelma: Tietojenkäsittelytieteet
Opintosuunta: Tietojärjestelmätiede
Työn ohjaaja: Juho-Pekka Mäkipää
Valmistumisvuosi: 2026 **Sivumäärä:** 59

TIIVISTELMÄ:
Koneoppimisen yleistymisen myötä tarve koneoppimisen kehittämiselle on kasvanut. Yksi mahdollinen tapa kehittää koneoppimismallien tehokkuutta ja lisätä ymmärrystä on lisätä ihminen joka tuntee relevantin aihealueen mukaan mallin kehitykseen. Ihmisen osallistumista voidaan tarkastella eri tasoilla, riippuen siitä, kuka johtaa oppimisprosessia. Tutkielma on kontrolloitu koe, jossa ihmisen osallistumista koneoppimismallin toimintaan simuloidaan, ja siten tarkastellaan ihmisen käyttökelpoisuutta verrattaessa optimoituun koneoppimismalliin. Koneoppimismalli jota tutkielmassa käytetään on satunnaismetsä, sillä se saavuttaa hyvän suorituskyvyn, mutta se on myös tulkittava malli.

Tutkielmassa suoritettu simulaatio osoittaa, että myös niissä tilanteissa kun ihminen tekee virheitä, vertautuu koneopettaminen silti hyvin optimoituun malliin. Simuloidussa koneopetustilanteessa, missä opettajan tarkkuus oli 80% ja 60%, saavutetaan samankaltainen suorituskyky erittäin nopeasti. Kuitenkin optimoitu malli saavuttaa suorituskyvyn, johon enemmistö muista tavoista eivät aivan yletä. Sen takia on tarkoituksenmukaista tarkastella opetustehokkuutta hyödyntäen normalisoitua aluetta oppimiskäyrän alapuolella (Area Under the Learning Curve, AULC). Epävarmuuteen perustuvalla näytteenvalinnalla koulutettu aktiivisen oppimisen malli hyödyntää vain noin 18% harjoitusdatasta, mutta saavuttaa kuitenkin yhtä hyvän F1-makrokeskiarvon. Koneopettamisen lopullinen F1-makrokeskiarvo jää hyvin vähän optimoidun mallin arvosta vajaaksi, mutta ero on satunnaisvaihtelun rajoissa, käytännössä F1-makrokeskiarvot ovat saman tasoisia.

Tutkielman tulokset osoittavat aktiivisen oppimisen eli datan luokittelun olevan tällä aineistolla hyvin tehokas menetelmä, saavuttaen optimoitua mallia vastaavan suorituskyvyn käyttämällä vain pientä osaa harjoitusdatasta. Tutkielma tarjoaa näyttöä siitä, että koneopettaminen on vikasietoinen menetelmä silloin kun virheet ovat vähämerkityksellisiä piirteitä, eivätkä suoranaisesti haitallisia. Tätä tutkittiin pakottamalla simuloidulle opettajalle virheitä, eli simulaatioissa tarkkuudet oli 100%, 80% ja 60%. Kun tarkkuus oli pienempi, virheet valittiin vähämerkityksellisten piirteiden joukosta satunnaisesti. Piirteiden informatiivisuus oli mahdollista laskea, sillä satunnaismetsä mallina mahdollistaa sen. Koneopetussimulaatiossa mallit saavuttivat melkein optimoidun mallin suorituskyvyn jo muutamalla piirteellä. Täyden tarkkuuden opettaja saavutti sen nopeimmin, vain kahdella piirteellä.

AVAINSANAT: Koneoppiminen, koneopettaminen, aktiivinen oppiminen, Human-in-the-loop

Sisällys

Kuviot

Taulukot

1 Johdanto

Tässä luvussa luodaan katsaus tutkielman taustaan, rakenteeseen, tavoitteeseen sekä käytettyyn aineistoon ja menetelmään. Tutkimuksen keskiössä on ihmisen ja koneoppimismallin välisen vuorovaikutuksen tehostaminen ja arvioiminen vertailemalla proaktiivista koneopettamista (Machine Teaching) ja reaktiivista aktiivista oppimista (Active Learning). Tavoitteena on selvittää, miten simuloidun ihmisen antamat konseptit, eli tässä tutkielmassa piirteet, vaikuttavat koneoppimismallin suorituskykyyn ja oppimiseen verrattuna perinteiseen havaintopohjaiseen oppimiseen. Lisäksi tutkimuksessa arvioidaan opettajan epävarmuuden vaikutusta mallin suorituskykyyn.

1.1 Tutkimuksen tausta

Koneoppiminen on noussut monissa tehtävissä, kuten esimerkiksi konenäössä, luonnollisen kielen käsittelyssä ja puheenkäsittelytehtävissä, viimeisintä tekniikkaa edustavaksi teknologiaksi. Koneoppimisen ainutlaatuiset haasteet kuitenkin viittaavat siihen, että käyttäjätiedon sisällyttäminen järjestelmään voi olla hyödyllistä (Wu ja muut, 2022, s. 364). Zanzotto (2019, s. 245) varoittaa siirtymän ihmisvetoisesta ohjelmoinnista kohti tekoälyn autonomista oppimista olevan mahdollisesti vaarallinen ihmiskeskeisen tekoälyn näkökulmasta (Human-in-the-loop AI). Tämä kehitysaskel voi olla ongelmallinen, sillä autonominen oppiminen poistaa ihmisen roolin ja vähentää vuorovaikutuksen hallittavuutta. Zanzotto (2019, s. 244) myös huomauttaa nykyisten järjestelmien hyödyntävän usein ihmisen tietämystä heidän tietämättään, eli varastamalla tietoa käyttäjiltä, jotka eivät ole tietoisia roolistaan järjestelmän ”opettajina”.

Ihmisen ja koneoppimismallien vuorovaikutusta on tutkittu tarkastelemalla kumpi osapuoli hallitsee oppimisprosessia. Mosqueira-Rey ja muut (2023, s. 3006) luokittelevat aktiivisen oppimisen menetelmänä, jossa järjestelmä säilyttää hallinnan oppimisprosessista. Vaikka tämä onkin suosittu lähestymistapa, sisältää se kuitenkin olennaisia rajoitteita. Tegen ja muut (2021, s. 1215) huomauttavat, että aktiivinen

oppiminen olettaa ihmisen toimivan prosessissa aina oikeassa olevana "oraakkelina", mikä ei vastaa todellista päätöksentekoa, jossa virheiltä ei usein voi välttyä. Tätä olettamusta ei haasteta tässä tutkimuksessa, vaan juuri tämän ajatuksen pohjalta vertaillaan oraakkelia hyödyntävää aktiivista oppimista koneopettamiseen, jossa ihminen ei ole oraakkeli, vaan tekee virheitä.

Vastakohtana tälle järjestelmän hallitsemalle oppimisprosessille toimii Holmbergin (2020, luku 1) mukaan koneopettaminen, jossa asiantuntija johtaa oppimisprosessia rajoittamalla tietoa, joka koneoppimismallille annetaan. Tämä mahdollistaa siirtymän pelkästä datan merkitsemisestä kohti syvällisempää tiedonsiirtoa, esimerkiksi kuten Simard ja muut (2017, luku 3) ehdottavat, antamalla järjestelmälle tietoa piirteiden muodossa. Mosqueira-Rey ja muut (2023, s. 3022) toteavat koneopettamisen rajoituksena olevan epätäydellisen tietämyksen kanssa työskentely. Tämän takia on tärkeää simuloida tilannetta, jossa ihminen ei ole täysin oikeassa (oraakkeli), sillä siten on mahdollista selvittää tarjoaako koneopettaminen hyötyjä, vaikka opettaja tekisi virheitä.

Aihetta on tärkeää tutkia lisää, sillä olemassa oleva tutkimus keskittyy tilanteisiin, jossa ei oteta huomioon sitä, että ihmiset tekevät virheitä. Lisäksi aiheena koneopettamista ei ole vielä tutkittu kattavasti, vaikka se tarjoaakin mahdollisuuksia koneoppimisen tehostamiseksi esimerkiksi oppimisnopeuden, datan puutteellisuuden ja asiantuntijatiedon hyödyntämisen näkökulmista. Aihe on myös käytännössä merkittävä, koska oikean elämän tilanteissa täydellisesti merkittyä dataa tai virheetöntä asiantuntijatietoa ei ole yleensä saatavilla. Näiden syiden takia on hyvin oleellista tutkia, kuinka hyvin koneopettaminen toimii tilanteessa, jossa ihmisen osallistuminen sisältää epävarmuutta ja virheitä. Mikäli koneopettaminen ja aktiivinen oppiminen mahdollistavat kilpailukykyisen suorituskyvyn myös epätäydellisellä simuloidulla ihmisen osallistumisella, voitaisiin niitä hyödyntää laajemmin osa-alueilla, joissa datan käsittely on kallista ja aikaa vievää.

1.2 Tutkimuksen tavoite

Tutkielman päätavoite on tarkastella koneoppimismallien oppimisprosessia lisäämällä simuloitu ihminen mukaan prosessiin. Tutkimuksen tavoitteena on osoittaa ihmisen osallistumisen olevan hyödyllistä vertaamalla simuloidun koneopettamisen kouluttamaa mallia ja aktiivisella oppimisella koulutettua mallia optimoituun perusmalliin. Koneopettamisen simulaatiosta tehdään realistisempi lisäämällä sille virheitä, ja tämän kautta pyritään osoittamaan koneopettamisen käyttökelpoisuus myös niissä tilanteissa, joissa ihminen voisi tehdä virheitä. Tutkielma pyrkii myös selvittämään aktiivisen oppimisen ja koneopettamisen eroja, ja vertailemaan niitä.

Tutkielmassa sekä aktiivinen oppiminen, että koneopettaminen ovat ihmisen osallistumisen kannalta simuloitu, eli kone saa tiedon suoraan. Aktiivisessa oppimisessa mallin on mahdollista tarkistaa tavoitesyöte automaattisesti. Koneopettamisen saralla asiantuntijan tieto simuloidaan käyttämällä optimoidun mallin tuottamaa piirteiden tärkeysjärjestystä, joka toimii arviona ideaalista asiantuntijatiedosta. Syötteeseen lisätään myös virhemarginaali, jolloin koneelle opetetaan vähämerkityksellisiä piirteitä.

Tutkimuskysymykset on rajattu seuraaviksi**:**

TK1: Miten opettajan tekemät virheet piirteiden valinnassa vaikuttavat koneopettamisen suorituskykyyn verrattuna optimoituun koneoppimiseen?

TK2: Miten simuloitu proaktiivinen piirteiden valinta eli koneopettaminen vertautuu suorituskyvyltään reaktiiviseen havaintojen merkintään ja kuinka vikasietoinen menetelmä on opettajan tekemille virheille?

Tutkimuksen tulokset osoittavat, että sekä aktiivisella oppimisella ja koneopettamisella voidaan saavuttaa kilpailukykyinen suorituskyky verrattuna optimoituun satunnaismetsämalliin. Tulokset viittaavat myös siihen, että koneopettaminen voi säilyä hyödyllisenä menetelmänä tilanteissa, joissa opettaja tekee virheitä piirteiden

valinnassa. Tulos tukee ajatusta siitä, että ihmisen osallistumista koneoppimisessa voidaan tarkastella realistisemmasta näkökulmasta, ilman olettamaa oraakkelista.

1.3 Tutkimuksen rakenne ja rajaus

Tutkielma on jaettu viiteen kappaleeseen. Kappale yksi on johdanto, jossa esitellään tutkielman taustaa, tavoitteita ja tutkimuskysymyksiä, sekä aiheen rajausta. Kappaleessa yksi kerrotaan myös miksi aihetta tulee tutkia, ja kerrotaan aiheesta yleisesti, jotta lukija ymmärtää mistä on kyse, ja jotta lukija ymmärtää aiheen tärkeyden sekä tutkielman hyödynnettävyyden.

Toinen luku keskittyy teoreettiseen viitekehykseen, jossa määritellään tutkielmalle olennaisia käsitteitä ja teorioita, sekä kerrotaan niiden käytöstä tutkielmassa. Olennaisia käsitteitä tutkielmassa ovat muun muassa koneoppiminen, human-in-the-loop, aktiivinen oppiminen ja koneopettaminen.

Kolmas luku kertoo tutkimusmenetelmästä, jossa kerrotaan tarkasti simulaation luomisesta, toiminnasta ja tulkinnasta. Kappaleessa kerrotaan myös kontrolloidusta kokeesta, ja siitä miten se soveltuu tähän tutkielmaan.

Neljännessä kappaleessa käsitellään edeltävässä kappaleessa määritellyillä tavoilla saatuja tuloksia. Viidennessä kappaleessa käsitellään tulosten vaikutuksia, sekä mitä jatkossa tulisi tutkia lisää, ja mitä rajoituksia tutkielmassa on.

Holmberg (2020, luku 1) jakaa ihmisen ja koneoppimismallien yhteistyön kolmeen osaan: aktiivinen oppiminen, interaktiivinen koneoppiminen (Interactive Machine Learning) ja koneopettaminen. Interaktiivinen koneoppiminen on Holmbergin (2020, luku 1) mukaan ihmisen ja koneen yhteistyön päätepisteiden, aktiivisen oppimisen ja koneopettamisen välissä. Lisäksi, Amershi ja muut (2014, s. 106) toteavat, että interaktiivisen koneoppimisen nopeat, keskittyneet ja inkrementaaliset oppimissyklit johtavat siihen,

että on vaikea erottaa ihmisen ja koneen vaikutus malliin. Tämän takia interaktiivinen koneoppiminen on rajattu tutkielman ulkopuolelle, ja tutkielma keskittyy nimenomaan aktiiviseen oppimiseen, sekä koneopettamiseen.

Tutkielmassa oletetaan täydellinen tieto kohdemallista ja oppijasta, mutta tavoitteena ei ole kuitenkaan suunnitella optimaalista koulutustietoaineistoa. Tutkielmassa opettajan rooli rajoittuu mallin ohjaamiseen piirteiden valinnan kautta. Mosqueira-Reyn ja muiden (2023, s. 3021) mukaan nykyään koneopettamisella tarkoitetaan koneoppimisen kontekstissa lähinnä ideaa siitä, että opettaja opettaa koneoppimisenmallin koneoppimisalgoritmille. He toteavat myös että opettajan on tarkoitus olla ihminen, mutta on myös mahdollista että algoritmi simuloi opettajaa. Tässä tutkielmassa käytettävä tulkinta koneopettamisesta, jossa algoritmi simuloi opettajaa koneoppimisprosessissa opettaen mallia on siis reilu, sekä yleisesti hyväksytty.

2 Koneoppiminen

Koneoppiminen on prosessi, jossa tietokoneet selvittävät, miten suorittaa tehtäviä ilman, että niitä on erikseen koulutettu tekemään niitä (Trisal & Mandloi, 2021, s. 343). Eli, koneoppimisessa mallin tavoitteena on tuottaa ennusteita ilman täsmällistä ohjelmointia (Trisal & Mandloi, 2021, s. 343). Koneoppiminen ja ajatus siitä, että kone oppii abstraktin konseptin datasta, ja hyödyntää sitä uusissa tilanteissa ei ole uusi, ja se onkin ollut olemassa jo ainakin 1950-luvulta lähtien (Badillo ja muut, 2020, s. 871).

El Hassanin ja muiden (2025, s. 320) mukaan koneoppimisalgoritmit voidaan jakaa neljään kategoriaan: valvottuun oppimiseen, valvomattomaan oppimiseen, puolivalvottuun oppimiseen ja vahvistusoppimiseen (supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning).

2.1 Valvottu oppiminen

Valvottu oppiminen on koneoppimisen menetelmä, jossa malli oppii valmiiksi merkitystä datasta (Rincy & Gupta, 2020, s. 1). Oppimisalgoritmi saa siis syötteenä joukon piirteitä sekä niihin liittyvät oikeat tulokset, ja se oppii vertaamalla omaa tuottamaansa tulosta oikeisiin tuloksiin havaitakseen virheet, jonka jälkeen algoritmi muuttaa mallia vastaavasti (Nasteski, 2017, s. 4). Valvotun oppimisen tehtävät voidaan jakaa regressioon ja luokitteluun, luokittelussa ennustettava tulos on diskreetti, ja regressiossa ennustettava tulos on jatkuva arvo (Nasteski, 2017, s. 5). Rincyn ja Guptan (2020, s. 2) mukaan valvotun koneoppimisen algoritmit voidaan luokitella eri ryhmiin niiden toimintaperiaatteiden perusteella, kuten esimerkiksi päätöspuihin (decision trees), sääntöpohjaisiin luokittelijoihin (rule-based classifiers) ja neuroverkkoihin (neural networks). Valvotun oppimisen päätehtävänä on muodostaa estimaattori, joka pystyy ennustamaan kohteen tavoitemuuttujan sen piirteiden perusteella (Nasteski, 2017, s. 4).

2.1.1 Luokittelu

Luokittelu on koneoppimisen menetelmä, joka pyrkii ennustamaan mihin ryhmään data-esiintymät kuuluvat (Aized Amin Soofi & Arshad Awan, 2017, s. 459). Kuten Nasteski (2017, s. 2) toteaa, luokittelu on valvotun koneoppimisen tehtävä, jossa tavoitteena on oppia funktio, joka luokittelee vektorin yhteen useista ennalta määritellyistä luokista hyödyntäen useita funktion tulo – lähtö esimerkkejä. Ayodele (2010, s. 24) mukaan yleisiä luokitteluun käytettyjä koneoppimisalgoritmejä ovat logistinen regressio, Naiivi Bayes -luokittelija, perseptroni ja tukivektorikone, jotka kuuluvat lineaaristen luokittelijoiden joukkoon. Ayodele (2010, s. 24) mainitsee myös kvadraattiset luokittelijat, k-means klusteroinnin, tehostuksen, päätöspuun, satunnaismetsän, neuroverkon ja Bayesin verkon. Vaikka Ayodele (2010, s. 24) sisällyttää k-means klusteroinnin luokittelualgoritmien joukkoon, menetelmä on kirjallisuudessa kuitenkin yleisesti määritelty valvomattoman oppimisen osa-alueeksi, kuten esimerkiksi Eckhardt ja muut (2023, s. 377) toteavatkin.

2.1.2 Päätöspuu

Päätöspuu on perusteellinen ja intuitiivinen koneoppimisalgoritmi, jota käytetään sekä luokittelussa että regressiossa (Dwaraka Srihith ja muut, 2023, s. 29). Blockeelin ja muiden (2023, s. 2) mukaan päätöspuu on menetelmä funktion f(x):n tuloksen laskemiseksi. Menetelmä koostuu syötteelle x toistuvasti suoritettavista testeistä, joissa kunkin testin tulos määrää seuraavan testin, kunnes lopullinen arvo f(x) voidaan määrittää. Dwaraka Srihithin ja muiden mukaan (2023, s. 30) päätöspuu koostuu juurisolmusta (root node), sisäsolmuista (internal nodes) ja lehtisolmuista (leaf nodes). Heidän mukaan juurisolmu on ylin solmu, ja se edustaa koko aineistoa, sisäsolmut edustavat yksittäisiin ominaisuuksiin perustuvia testejä tai päätöksiä. Sisäsolmuihin tulee yksi terä yläsolmusta (parent node), ja niistä lähtee kaksi tai useampi terää alisolmuihin (child nodes). Lisäksi he toteavat, että lehtisolmuilla ei ole alisolmuja ja niissä muodostuu lopullinen ennuste, joka on luokittelussa luokkatunniste ja regressiossa numeerinen arvo.

Luokittelutehtävissä lehtisolmun luokaksi määräytyy solmun enemmistöluokka (Dwaraka Srihith ja muut, 2023, s. 33). Yksi tärkeä päätöspuiden ominaisuus on se, että niiden tulos on yksinkertainen ja helposti ymmärrettävä (Blockeel ja muut, 2023, s. 2).

2.1.3 Yhdistelmämalli

Yhdistelmäoppiminen (ensemble learning) pohjautuu ajatukseen siitä, että joukko tietää paremmin kuin yksilö. Teoria joukkoälystä toteaa, että yhdistämällä useiden yksilöiden tietoa, saadaan parempia päätöksiä kuin vain yhden yksilön päätöksellä (Kumar ja muut, 2022, s. 1). Yhdistelmämalli on joukko samasta harjoitusdatasta oppineita malleja, joiden tulokset yhdistetään hyödyntämällä esimerkiksi painotettua keskiarvoa, keskiarvoa, äänestystä tai todennäköisyyttä (Kumar ja muut, 2022, s. 2). Yhdistelmämallien päätavoite on siis parantaa yksittäisten mallien suorituskykyä yhdistämällä malleja, tuottaen uuden, paremman suorituskyvyn omaavan mallin (Galar ja muut, 2012, s. 467). Yhdistelmämallien mukana tuleva suorituskyvyn paraneminen perustuu yleensä varianssin vähenemiseen, mikä puolestaan vähentää ylisovittamisen (overfitting) vaaraa (Galar ja muut, 2012, s. 467).

2.1.4 Satunnaismetsä

Fawagreh ja muut (2014, s. 604) kertovat, että satunnaismetsä on yhdistelmämalli jota käytetään sekä luokitteluun että regressiotehtäviin, ja että sen kehitti Breiman vuonna 2001. Satunnaismetsä koostuu useista päätöspuista, jossa jokainen puu muodostetaan toisistaan riippumattomasti satunnaisesti valitun otoksen perusteella siten, että otokset on poimittu samasta todennäköisyysjakaumasta (Breiman, 2001, s. 5). Salman ja muut (2024, s. 72) toteavat satunnaismetsän vähentävän päätöspuiden välistä korrelaatiota käyttämällä satunnaista havaintojen ja piirteiden valintaa. Cutler ja muut (2012, viitattu teoksessa Salman ja muut, 2024, s. 72) toteavat, että nämä satunnaistamisen keinot vähentävät puiden välistä korrelaatiota, mikä vähentää ylisovituksen mahdollisesti

aiheuttamia virheitä, ja parantaa mallin tarkkuutta. Luokittelutehtävissä satunnaismetsän lopullinen luokka määräytyy enemmistöäänestyksen perusteella, jossa päätöspuiden yleisimmin ennustama luokka valitaan lopulliseksi metsän ennusteeksi (Salman ja muut, 2024, s. 74). Yhdistämällä puita ja niiden ennusteita enemmistöäänestyksellä on saavutettu merkittävästi parempia tuloksia, kuin vain yksittäistä puuta käyttäen (Breiman, 2001, s. 5).

2.2 Valvomaton oppiminen

Toinen keskeinen koneoppimisen menetelmä on valvomaton oppiminen. Valvomaton oppiminen on koneoppimisen lähestymistapa rakenteiden ja säännönmukaisuuksien havaitsemiseen aineistoista, joissa datapisteet ovat ilman valmiita tunnisteita tai rakenteita (Naeem ja muut, 2023, s. 911). Eckhardt ja muut (2023, s. 377) esittelevät neljä keskeistä valvomattoman koneoppimisen metodia: k-means klusterointi, hierarkinen klusterointi, pääkomponenttianalyysi ja faktorianalyysi (k-means clustering, hierarchical clustering, principal component analysis, factor analysis). Valvomattoman oppimisen tavoitteena on löytää piileviä rakenteita tunnisteettomasta datasta tai muodostaa malli, joka kuvaa syöttödatan todennäköisyystiheysjakaumaa (Rincy ja Gupta, 2020, s. 3).

2.3 Puolivalvottu oppiminen

Rincyn ja Guptan (2020, s. 5) mukaan puolivalvottu koneoppiminen on valvotun ja valvomattoman koneoppimisen ketju, ja puolivalvotussa koneoppimisessa merkitty data on hyvin niukkaa, samalla kun on erittäin suuri määrä merkitsemätöntä dataa. Puolivalvottu oppiminen on prosessi, jossa pyritään muodostamaan parempi malli hyödyntämällä sekä merkittyä että merkitsemätöntä dataa (Prakash & Nithya, 2014, s. 25). Merkittyä dataa on usein vaikea ja työläs hankkia, kun taas merkitsemätöntä dataa on runsaasti saatavilla. Tämän vuoksi puolivalvottu oppiminen tarjoaa perustellun

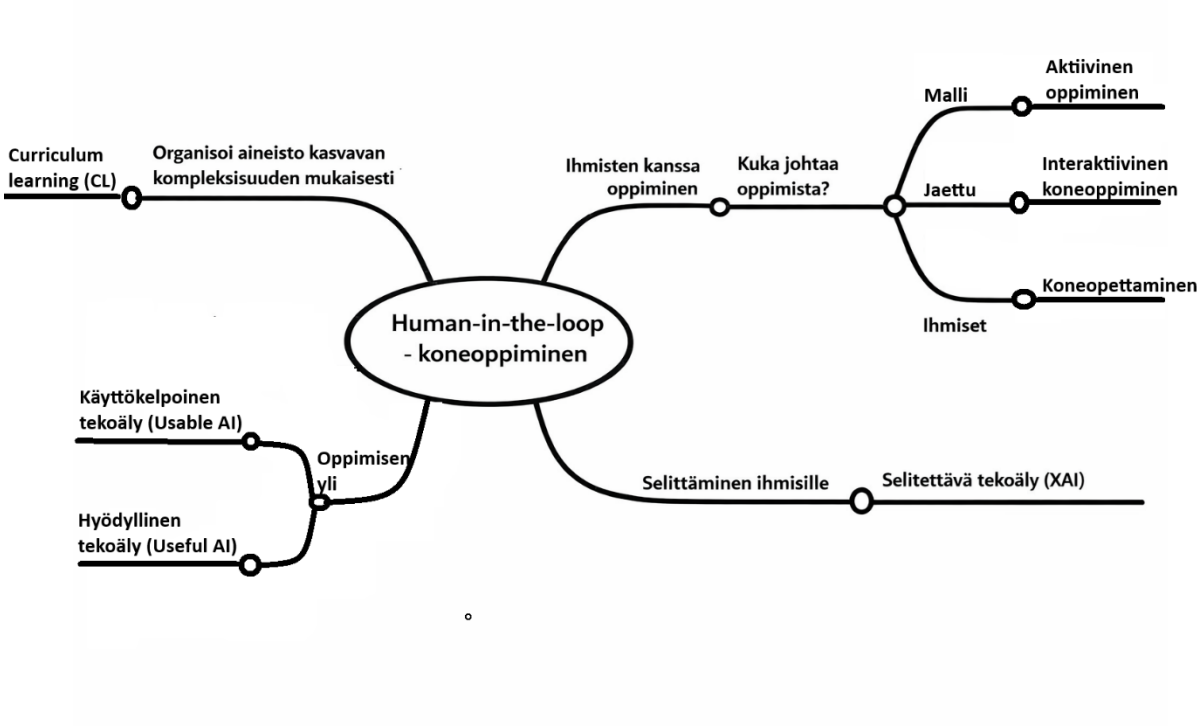
lähestymistavan inhimillisen työmäärän vähentämiseen ja ennustustarkkuuden parantamiseen (Prakash & Nithya, 2014, s. 25).

2.4 Vahvistusoppiminen

Viimeinen koneoppimisen kategoria on vahvistusoppiminen. Vahvistusoppimisessa agentti oppii vuorovaikutuksessa ympäristön kanssa valitsemaan toimintoja, jotka maksimoivat saadun palkkion (Rincy & Gupta, 2020, s. 4). Ympäristön antaman vahvistussignaalin tarkoituksena vahvistusoppimisessa on arvioida älykkään agentin toiminnan laatua, mutta ei kertoa agentille, miten oikea toiminta tulisi tehdä (Qiang & Zhongli, 2011, s. 1).

2.5 Human-in-the-loop -koneoppiminen

Mosqueira-Reyn ja muiden (2023, s. 3006) mukaan perinteisessä koneoppimisessa ihmisen rooli rajoittuu usein mallin kehityksen alkuvaiheisiin. He myös toteavat, että tällaisessa lähestymistavassa mallit eivät välttämättä skaalaudu hyvin, niistä voi tulla staattisia ja vaikeasti arvioitavia sekä suorituskyky voi huonontua. Human-in-the-loop (HITL) -koneoppiminen pyrkii integroimaan merkityksellistä ihmisten osaamista koko koneoppimisen sykliin, esimerkiksi datan keräämistä, algoritmin säätämistä tai parametrien valitsemista hyödyntäen (Wang ja muut, 2022, luku 1). Wang ja muut (2022, luku 1) toteavat myös, että tämän aiheen perimmäinen tavoite on uudelleentutkia ja uudelleenkehystää koneoppimisen työnkulku ihmiskeskeisestä näkökulmasta. Mosqueira-Rey ja muut (2023, s. 3006-3007) toteavat HITL-ajattelun keräävän alleen monia uudenlaisia ihmisen ja koneen vuorovaikutuksia, ja että tämän tarkoituksena on tehdä koneoppimisesta tehokkaampaa, mutta myös ihmisistä tehokkaampia.

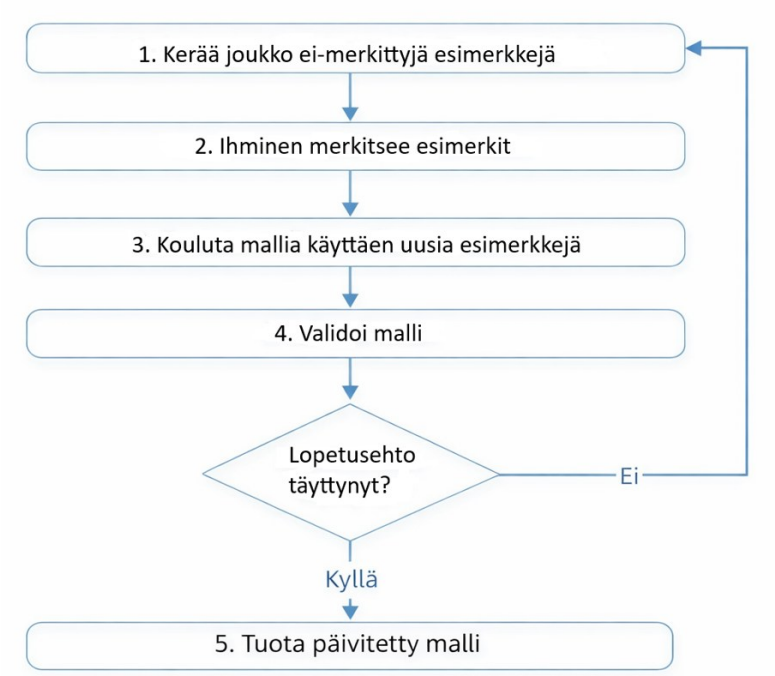


Kuvio 1. Human-in-the-loop -koneoppimisen miellekartta (Mosqueira-Rey ja muut, 2023, s. 3007).

Kuvion 1 mukaisesti, Mosqueira-Rey ja muut (2023, s. 3006-3007) jakavat HITL-koneoppimisen eri kategorioihin. He myös jakavat sen alakohtiin kuvion 1 mukaisesti, esimerkiksi riippuen tavoista joilla ihmiset voivat osallistua oppimisprosessiin ja kuka loppupeleissä johtaa oppimista. Mosqueira-Reyn ja muiden (2023, s. 3006-3007) mukaan oppimisen hallitsemisen mukaan jaeteltuna, voi HITL-koneoppimisen jakaa aktiiviseen oppimiseen, interaktiiviseen koneoppimiseen ja koneopettamiseen. Lisäksi he liittävät kokonaisuuteen Curriculum Learning:in (CL), selitettävän tekoälyn (Explainable AI, XAI) sekä käsitteet käyttökelpoinen tekoäly (Usable AI) ja hyödyllinen tekoäly (Useful AI), jotka laajentavat tarkastelua oppimisalgoritmien ulkopuolelle ihmiskokemukseen ja yhteiskunnalliseen kontekstiin.

2.6 Aktiivinen oppiminen

Koneoppimismallien kouluttaminen tarkoittaa useissa käytännön tapauksissa työskentelyä datan kanssa, jota on merkitty rajallisesti. Koska datan merkinnät voivat olla kalliita, tietoaineistot jaetaan usein pienempään merkittyyn osaan, ja suurempaan ei-merkittyyn osaan (Werner ja muut, 2025, s. 1). Aktiivinen oppiminen on koneoppimisen menetelmä, jossa oppija pyytää opettajana toimivaa oraakkelia merkitsemään valittuja tapauksia jotka ovat epävarmoja, ja jotka antavat oppimisprosessille relevanttia tietoa (Mosqueira-Rey ja muut, 2023, s. 3008). Mosqueira-Reyn ja muiden (2023, s. 3008-3009) mukaan tämän merkitsemisen avulla oppija parantaa oppimisprosessiaan käyttäen vähemmän harjoitusesimerkkejä. He toteavat myös aktiivisen oppimisen olevan hyvin tehokas tilanteissa joissa on paljon merkitsemätöntä dataa saatavilla, ja sen merkitseminen on kallista tai aikaa vievää. Lisäksi he toteavat, että tässä tekniikassa oppija hallitsee oppimisprosessia, ja että aktiivisen oppimisen voisi luokitella puolivalvottuun oppimiseen, koska aktiivinen käyttää sekä merkittyä, että merkitsemätöntä dataa.



Kuvio 2. Vaiheet mallin päivittämiseksi aktiivisessa oppimisessa (Mosqueira-Rey ja muut, 2023, s. 3010).

Kuvio 2 havainnollistaa aktiivisen oppimisen vaiheet, joita toistetaan kunnes lopetusehto on täyttynyt. Kuviossa puhutaan ihmisessä, mutta myös simuloitua ihmistä hyödynnetään usein aktiivisen oppimisen tilanteissa. Munro (2020, viitattu teoksessa Mosqueira-Rey ja muut, 2023, s. 3010) erottelee kolme näytteenvalinnan menetelmää: satunnaisuuteen perustuva näytteenvalinta, epävarmuuteen perustuva näytteenvalinta ja monimuotoisuuteen perustuva näytteenvalinta (random sampling, uncertainty sampling, diversity sampling).

2.6.1 Epävarmuuteen perustuva näytteenvalinta

Sun ja Zhou (2015, s. 122) kertovat epävarmuuteen perustuvan näytteenvalinnan perustuvan luokkien posterioritodennäköisyysjakauman arviointiin ja sen valitsevan ne esimerkit, jotka ovat kaikkein epävarmimpia posterioritodennäköisyyksien arvojen perusteella. Heidän mukaansa binäärisessä luokitteluongelmassa valitut esimerkit ovat niitä, joiden luokkajäsenyyden todennäköisyys on lähimpänä arvoa 0,5. Mosqueira-Reyn

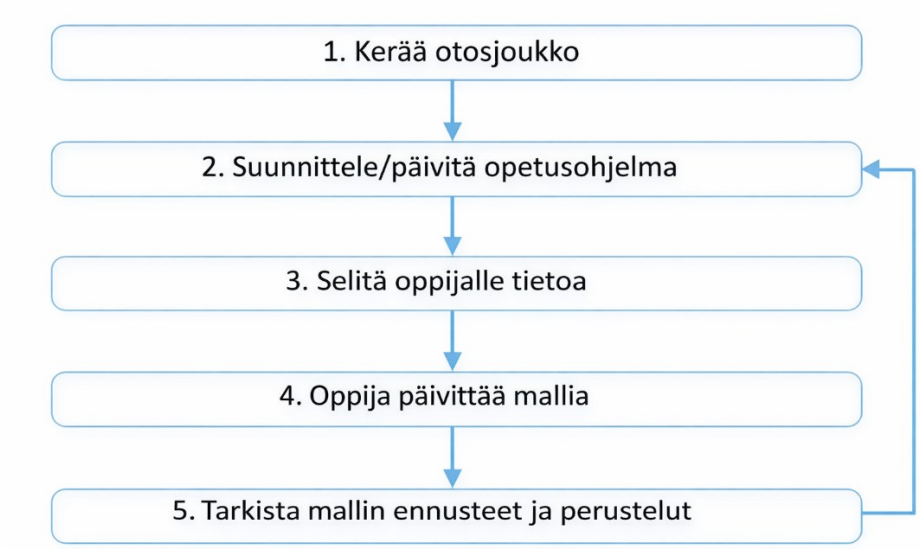
ja muiden (2023, s. 3010) mukaan epävarmuuteen perustuvan näytteenvalinnan kategoriaan kuuluu pienin varmuus, varmuuden marginaali, varmuuksien suhde ja entropia (least confidence, margin of confidence, ratio of confidence, entropy).

2.7 Koneopettaminen

Koneopettaminen on myös tapa siirtää tietoa ihmiseltä koneelle, eroten siten aktiivisesta oppimisesta, että oppimisprosessia johtaa ihminen (Mosqueira-Rey ja muut, 2023, s. 3021). Zhu (2015, s. 4083) määrittelee koneopettamisen optimaalisen harjoitusdatan löytämiseksi, kun koneoppimisalgoritmi ja tavoitemalli ovat annettuina. Tässä tulkinnassa opettajalla on täydellinen tieto sekä oppijasta että tavoitemallista, ja opetus tapahtuu koulutusesimerkkejä antamalla (Zhu, 2015, s. 4083). Tämä lähestymistapa ei kuitenkaan täysin vastaa ihmiskeskeistä koneopettamista, jossa opettajalla ei välttämättä ole täydellistä tietoa tavoitemallista, vaan sitä rakennetaan iteratiivisesti vuorovaikutuksessa oppijan kanssa.

Simardin ja muiden (2017, luku 1) mukaan koneoppimismallien rakentaminen vaatii syvää koneoppimisosaamista, ja jotta kasvava tarve koneoppimismalleille voitaisiin täyttää, on tarpeellista kasvattaa sitä määrää ihmisiä jotka osaavat opettaa koneita. He tarkoittavat tällä sitä, että koneoppimisjärjestelmien rakentaminen tulisi tehdä mahdolliseksi eri aihealueiden asiantuntijoille, joilla ei ole koneoppimisosaamista. Simardin ja muiden (2017, luku 1) mukaan koneopettaminen keskittyy uusien koneoppimisalgoritmien luomisen ja parantamisen sijaan opettajien tehokkuuteen suhteessa oppijoihin. He ehdottavat koneopettamisen mittareiksi tuottavuutta, tulkittavuutta, kestävyyttä ja skaalautuvuutta ongelman monimutkaisuuden tai osallistujien määrän kasvaessa. Mosqueira-Rey ja muut (2023, s. 3026) toteavatkin tämän lähestymistavan keskeisen hyödyn olevan ihmisten luontaisten opettamisen kykyjen hyödyntäminen, jotta myös ilman koneoppimistaustaa olevat henkilöt voivat siirtää tietoa tietokonejärjestelmälle samalla tavoin kuin he opettaisivat toista ihmistä. Wall ja muut (2019, s. 579) tukevat tätä ajatusta, heidän mukaansa koneopettamisessa

opettaja voi opettaa mallia eri tavoin. Heidän tulkintansa mukaan koneopettamisen kontekstissa opettajalla tarkoitetaan aihealueen asiantuntijaa, joka kouluttaa koneoppimismallia koneopettamisprosessin avulla.



Kuvio 3. Kaaviomainen esitys koneopettamisprosessista (Mosqueira-Rey ja muut, 2023, s. 3025).

Kuvio 3 kuvaa koneopettamisprosessia vaiheittain, kuten aktiivisessa oppimisessakin, koneopettaminen on iteratiivinen lähestymistapa. Koneopettamisessa opettaja voi olla aktiivinen osallistuja, joka valitsee harjoitusdatajoukkoon sisällytettäviä elementtejä, merkiten ja valiten milloin ja miten ennustevirheitä korjataan luomalla semanttisesti merkityksellisiä piirteitä havaintojensa pohjalta (Wall ja muut, 2019, s. 579). Simardin ja muiden (2017, luku 4) mukaan opettajan tehtävä on siirtää tietoa oppimiskoneelle siten, että se kykenee tuottamaan konseptia lähestyvän mallin. He määrittelevät konseptin tarkoittamaan sääntöä tai funktiota, joka määrittää, miten havainnot luokitellaan.

Koneopettaminen ei siis rajoitu pelkästään harjoitusdatan valintaan tai luokitteluun, vaan opettaja voi siirtää tietoa myös piirteiden avulla. Simard ja muut (2017, luku 4) tarkoittavat piirteellä konseptia, joka antaa jokaiselle esimerkille skalaariarvon, ja heidän mukaansa piirteitä käytetään osoittamaan konseptia, kun halutaan korostaa sen käyttöä

koneoppimismallissa. Osana oppimisprosessia opettaja voi muokata, luoda uusia tai olla huomioimatta piirteitä, ja siten saavuttaa kestävän mallin (Simard ja muut, 2017, luku 5).

2.7.1 Koneopettaminen piirteiden valinnalla

Koneopettamisessa tiedonsiirto ei rajoitu vain esimerkkien merkitsemiseen, vaan ihmisopettajan on mahdollista opettaa esimerkiksi piirteillä, tekemällä pareittain vertailuja tai säännöillä (Zhu ja muut, 2018, s. 7). Tämä edellyttää sen, että oppimisalgoritmi on varustettu vastaanottamaan tämänlaisia opetusviestejä (Zhu ja muut, 2018, s. 7). Piirteiden valintaa ei voi käyttää siis koneopettamiseen kaikissa tilanteissa, sillä se vaatii sopivan oppimisalgoritmin.

Opettaja voi koneopettamisprosessissa valita millä esimerkeillä mallia opetetaan, milloin ja miten mallia päivitetään tai korjataan, sekä hajottaa tai yhdistää piirteitä tai tehtäviä ja niin edelleen (Ramos ja muut, 2020, s. 419). Koneopettaminen piirteiden valinnalla on siis yksi monista tavoista siirtää asiantuntijoiden tietoa oppimisalgoritmille. Ramos ja muut (2020, s. 424) kertovat myös semanttisista piirteistä koneopettamisessa, semanttisilla piirteillä tarkoitetaan funktiota, joka edustaa ihmisille ymmärrettävää konseptia (human concept), ja joka palauttaa tietorakenteen jota oppimisalgoritmi voi hyödyntää. Semanttiset piirteet eroavat piirteiden valinnasta siten, että ne eivät rajoitu aineistossa jo oleviin piirteisiin. Kuitenkin ihmisille ymmärrettävän konseptin määrittely ei ole yksiselitteistä, ja sen tulkinta voi vaihdella kontekstista ja käyttäjästä riippuen.

2.8 Luokittelukoneoppimismallien mittaaminen

Koneoppimismallien mittaamisella on kaksi päätarkoitusta, huonosti suoriutuvien metodien hylkääminen ja lupaavien metodien optimointi (Rainio ja muut, 2024). Rainion ja muiden (2024) mukaan binäärisissä luokittelutehtävissä yleisimmät suorituskyvyn mittarit ovat tarkkuus (accuracy), herkkyys (recall), sisäinen tarkkuus (precision) ja

spesifisyys (specificity). Heidän mukaansa näillä mittareilla mitataan prosentuaalisesti oikein luokiteltuja tapauksia kaikista tapauksista, oikeita positiivisia tapauksia, oikeita negatiivisia tapauksia ja positiivisesti luokiteltuja tapauksia. Oikea positiivinen (True Positive, TP) on oikein ennustettu positiivinen tulos, oikea negatiivinen on oikein ennustettu negatiivinen tulos (True Negative, TN), väärä positiivinen (False Positive, FP) on negatiivinen tapaus joka ennustettiin positiiviseksi ja väärä negatiivinen (False Negative, FN) on positiivinen tapaus joka ennustettiin negatiiviseksi (Rainio ja muut, 2024). Rainio ja muut (2024) määrittelevät tarkkuuden, herkkyyden, sisäisen tarkkuuden ja spesifisyyden kaavat seuraavanlaisiksi:

$$Tarkkuus = \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1]. \quad (1)$$

$$Herkkyys = \frac{TP}{TP + FN} \in [0, 1]. \quad (2)$$

$$Sisäinen\ tarkkuus = \frac{TP}{TP + FP} \in [0, 1]. \quad (3)$$

$$Spesifisyys = \frac{TN}{TN + FP} \in [0, 1]. \quad (4)$$

Herkkyyttä, sisäistä tarkkuutta ja spesifisyyttä käytetään usein pareittain, kuten esimerkiksi sisäinen tarkkuus ja herkkyys, tai spesifisyys ja herkkyys (Rainio ja muut, 2024). Rainion ja muiden (2024) mukaan herkkyys ja spesifisyys paljastavat mallista enemmän kuin vain tarkkuus, varsinkin jos positiivisten ja negatiivisten instanssien määrä on hyvin epätasapainoinen. He esittelevät myös F1-arvon (F1-Score), joka on sisäisen tarkkuuden ja herkkyyden harmoninen keskiarvo. F1-arvon kaava on:

$$F1 - arvo = \frac{2 \cdot Sisäinen\ tarkkuus \cdot Herkkyys}{Sisäinen\ tarkkuus + Herkkyys} \in [0, 1]. \quad (5)$$

Tutkielmassa hyödynnetään F1-arvon sijaan F1-makrokeskiarvoa (Macro-F1). Opitz ja Burst (2021, s. 1) määrittelevät F1-makrokeskiarvon kaavan seuraavanlaiseksi:

$$F1 - makrokeskiarvo = \frac{1}{n}\sum_{x} F1_x\,, \quad (6)$$

missä n on luokkien lukumäärä ja $F1_x$ on luokan x F1-arvo.

F1-makrokeskiarvoa hyödynnetään usein tilanteissa joissa luokkien jakaumat ovat epätasapainoisia (Opitz & Burst, 2021, s. 3). F1-makrokeskiarvoa voi kuitenkin hyödyntää myös tilanteissa, joissa luokkia on vain kaksi, ja jakauma on tasainen.

Koneoppimismalleja voi mitata myös tutkimalla niiden oppimiskäyriä. Oppimiskäyrä on tärkeä graafinen visualisaatio, joka voi antaa lisätietoa oppimiskäyttäytymisestä tuottamalla kuvaajan yleistämissuorituskyvystä suhteessa koulutusesimerkkien määrään (Viering & Loog, 2022, s. 1). Vieringin ja Loogin (2022, s. 2) mukaan voi olla hyödyllistä tiivistää oppimiskäyrät yhdeksi numeroksi, yksi suosittu mittari tätä varten on alue oppimiskäyrän alapuolella (Area Under the Learning Curve, AULC). Tämän mittarin laskemiseksi on ensin määritettävä useita näytekokoja, jonka jälkeen jokaisella näytekoolla lasketaan suorituskyvyn keskiarvo, jotta saadaan oppimiskäyrän alle jäävä pinta-ala. Alue oppimiskäyrän alapuolella tekee siten erikoisen oletuksen, että kaikki näytekoot ovat yhtä todennäköisiä. Eli pelkästään lopullisen suorituskyvyn arvioinnin sijaan, alue oppimiskäyrän alapuolella mittaa suorituskyvyn keskiarvoa koko oppimisprosessilta, minkä takia tämä toimii hyvin mittarina kun halutaan verrata eri metodien näytteiden hyödyntämisen tehokkuutta.

2.9 Koneoppimisen ja ihmisen vuorovaikutuksen tutkimus

Aiempi tutkimus ihmisen ja koneoppimismallien vuorovaikutuksesta on keskittynyt pitkälti mallien suorituskyvyn parantamiseen. Tarkemmin sanoen, aktiiviseen oppimiseen keskittyvä tutkimus on keskittynyt niukan datan käytön optimointiin, ja eri näytteenvalinnan tapojen vertailuun. Werner ja muut (2025, s. 6) tutkivat aktiivista oppimista verraten sitä eri menetelmiin joilla yritetään ratkoa niukan datan ongelmaa, ja he huomasivat, että aktiivinen oppiminen häviää suorituskyvyssään datan

augmentaatiolle (Data Augmentation, DA) ja puolivalvotulle oppimiselle. He kuitenkin toteavat myös sen, että aktiivinen oppiminen tarjoaa lisähyötyä, kun sen yhdistää näihin eri menetelmiin. Heidän tutkielmansa tulos ei ole suoraan tälle tutkielmalle keskeinen, mutta se osoittaa hyvin nykyisen aktiivisen oppimisen tutkimuksen rajoituksia. Heidän tuloksensa myös toimii perusteena sille, että aktiivista oppimista hyödynnetään tässä tutkielmassa vertailukohtana eikä päämenetelmänä.

Simardin ja muiden (2017, luku 2) mukaan koneopettamisen tutkimuksessa pyritään tekemään opettajasta tehokkaampi koneoppimismallien luomisessa. Mosqueira-Rey ja muut (2023, s. 3023-3024) kuvaavat koneopettamisen tutkimuksen kehitystä, heidän mukaansa koneopettamisen tutkimus on kehittynyt eräpohjaisesta lähestymistavasta ”loputtomaksi silmukaksi”, ja että tätä lähestymistapaa tutki ensimmäistä kertaa loppukäyttäjien kanssa Wall ja muut (2019).

Olemassa olevassa tutkimuksessa on kuitenkin puutteita, sekä aktiivisessa oppimisessa että koneopettamisessa. Aktiivista oppimista ja koneopettamista tutkitaan usein siitä lähtökohdasta, että opettajan antama tieto on täydellistä, mikä ei vastaa oikean elämän tilanteita. Devidzen ja muiden (2020, s. 2647) mukaan koneopettamisen tutkimuksessa usein oletetaan opettajan tiedon oppijasta ja tehtävästä olevan täydellistä. Heidän tutkimuksessa onkin keskeisenä aiheena selvittää, miten tehokkaasti epätäydellinen opettaja saa opetettua oppijaa. Devidze ja muut (2020, s. 2649) tutkivat epätäydellisyyttä neljällä tavalla, lisäämällä kohinaa oppijan lähtötiedossa, oppimisnopeudessa, käyttäen rajallista määrää merkittyä dataa sekä lisäämällä kohinaa piirre-esityksessä. Tässä tutkielmassa opettajan epätäydellisyyttä lähestytään toisesta, vähemmän tutkitusta näkökulmasta. Esimerkkien sijaan, opettaja opettaa oppijalle piirteitä, eli siirretty tieto on eri muodossa. Tämän lisäksi oppijana toimii satunnaismetsämalli, kun taas Devidzen ja muiden (2020, s. 2648) tutkimuksessa hyödynnettiin version-space mallia oppijana.

Aktiivisen oppimisen yleistä olettamaa ihmisen täydellisestä tiedosta ei tässä tutkielmassa haasteta, vaan aktiivista oppimista hyödynnetään vertailukohtana koneopettamiselle, mikä mahdollistaa laajemman kuvan saamisen simuloidusta ihmisen osallistumisesta koneoppimismallin toimintaan. Tämä tutkimusasetelma mahdollistaa näiden vastakkaisten menetelmien vertailun, sillä tutkielmassa käytetään samaa mallia ja samaa aineistoa. Yhdistämällä näiden menetelmien vertailuun koneopettamisen virheellisyyden simuloinnin, saadaan tutkielmasta empiiristä tietoa siitä, miten epätäydellinen opettajan tieto vaikuttaa koneopettamisen suorituskykyyn ja miten tämä vertautuu aktiiviseen oppimiseen suorituskyvyn ja vikasietoisuuden osalta.

3 Tutkimusmenetelmä

Tässä luvussa käsitellään valittua tutkimusmenetelmää, sekä käytettyä aineistoa ja simulaatiossa käytettyjä eri menetelmiä. Tutkielmassa hyödynnetty tutkimusmenetelmä on kontrolloitu koe (Controlled Experiment). Kontrolloidun kokeen avulla on tarkoitus selvittää miten simuloitu asiantuntijaihminen vaikuttaa koneoppimismallin suorituskykyyn, kun mallina on satunnaismetsä. Simulaatiosta on pyritty poistamaan kaikki satunnaisten tekijöiden vaikutukset, jotta kokeesta saatava tulos vastaisi tutkimusmenetelmää mahdollisimman tarkasti, ja jotta se tuottaisi käyttökelpoisia tuloksia. Aiemman luvun teoria tukee tutkimuksellista asettelua, ja se antaa vertauskohdan tutkielmassa saatuihin tuloksiin.

Kontrolloidun kokeen avulla pyritään vastaamaan tutkielmassa asetettuihin tutkimuskysymyksiin vertailemalla aktiivisen oppimisen, koneopettamisen ja optimoidun perusmallin suorituskykyä samoissa olosuhteissa. Ensimmäiseen tutkimuskysymykseen vastataan tarkastelemalla, miten simuloidut virheet piirteiden valinnassa vaikuttavat koneopettamisen suorituskykyyn verrattaessa optimoituun perusmalliin. Toiseen tutkimuskysymykseen vastataan vertaamalla koneopettamisen suorituskykyä myös aktiiviseen oppimiseen, ja arvioimalla kuinka hyvin koneopettaminen säilyttää suorituskykynsä tilanteissa, joissa myös virheitä on simuloitu.

3.1 Kontrolloitu koe

Kontrolloitu koe tarkoittaa menetelmää, jossa mahdollisimman monta tutkittavaan ilmiöön liittyvää muuttujaa ovat tutkijan hallinnassa (Järvinen, 2001, s. 48). Tämä soveltuu tutkimusmenetelmänä tähän tutkielmaan hyvin, sillä tutkielman asetelma mahdollistaa muuttujien hallitsemisen, sekä satunnaisten muuttujien poistamisen. Järvisen (2001, s. 48) mukaan kontrolloidussa kokeessa on riippumattomia muuttujia, sekä riippuvia muuttujia. Tässä tutkielmassa riippumattomina muuttujina toimii koneoppimismallin opetustapa, eli onko kyseessä aktiivisen oppimisen instanssien

merkitseminen, vai koneopettamisen tapa piirteitä hyödyntäen. Riippuvana muuttujana toimii mallin suorituskyky, sillä se muuttuu riippuen opetustavasta. Lisäksi kokeessa käytetään vertailukohtana perusmallia, joka koulutetaan ilman aktiivista oppimista tai koneopettamista. Tämän avulla voidaan arvioida, tuovatko tarkastellut menetelmät lisäarvoa verrattuna perinteiseen koneoppimiseen. Muut mahdolliset muuttujat, kuten aineisto, koneoppimismalli ja arviointimenetelmät pidetään vakiona, jotta koe pysyisi mahdollisimman kontrolloituna.

3.2 Aineisto

Tutkielmassa käytetty tietoaineisto on synteettinen, ja se simuloi kyberturvallisuushyökkäysten ennustamista. Tietoaineisto sisältää verkkoliikennetietoa sekä tietoa käyttäjien käyttäytymisestä. Tietoaineistona toimii käyttäjän Dinesh Naveen Kumar Samudrala julkaisema tietoaineisto: Cybersecurity Intrusion Detection Dataset *(Kaggle, 2025).* Tietoaineistossa on 9537 instanssia, ja puuttuvia tietoja on vain yhdessä piirteessä, jonka takia tietoaineisto soveltuu hyvin käytettäväksi kontrolloidussa kokeessa. Tietoaineistossa on myös yhdenmukainen muotoilu ja tavoitemuuttuja on tasapainoinen jakautuen siten, että noin 45% edustaa positiivista tulosta (arvo 1) ja 55% negatiivista tulosta (arvo 0). Positiivisia tuloksia on 4264 ja negatiivisia tuloksia on 5273.

Tietoaineistossa on 11 piirrettä, joista tavoitemuuttuja, jota pyritään ennustamaan, on attack_detected. Piirre attack_detected kuvaa sitä, että havaittiinko kyseisessä instanssissa hyökkäys vai ei. Piirre arvolla 1 tarkoittaa että hyökkäys on havaittu, ja kun piirteen arvo on 0 niin kyseessä ei ole hyökkäys. Session_id piirrettä ei käytetä tutkielmassa, sillä se on vain tunniste istunnolle. Eli ennustavia piirteitä, joita hyödynnetään on yhdeksän.

3.3 Aineistoanalyysi

Aineiston analysointi perustuu teoriaa testaavaan tutkimukseen, jossa eri koulutusmenetelmien suorituskykyjä mitataan samoilla mittareilla. Tutkielmassa pyrittiin välttämään laajaa datan käsittelyä, sillä tavoite ei ole saada paras mahdollinen suorituskyky, vaan vertailla opetusmenetelmiä. Tämän takia esikäsittely pidettiin mahdollisimman yksinkertaisena, eikä kokeessa suoritettu kattavaa piirteiden suunnittelua (feature engineering). Jos aineistoon olisi tehty kattavaa käsittelyä suorituskyvyn maksimoimiseksi, se olisi myös voinut aiheuttaa epäselvyyttä esimerkiksi piirteiden tärkeysjärjestykseen.

Analyysin lähtökohtana ovat tutkimuskysymykset. Ensimmäiseen tutkimuskysymykseen vastataan vertaamalla simuloitua koneopettamista optimoituun perusmalliin suorituskykyjen perusteella. Toiseen tutkimuskysymykseen vastataan vertaamalla koneopettamisen tuloksia aktiivisen oppimisen tuloksiin sekä tarkastelemalla koneopettamisen suorituskykyä opettajan tekemien virheiden näkökulmasta.

Suorituskykyä arvioidaan käyttämällä viittä yleistä binääriluokittelun mittaria, tarkuutta, herkkyyttä, sisäistä tarkkuutta, spesifisyyttä ja F1-makrokeskiarvoa. F1-makrokeskiarvo oli mittareista keskeisin, sillä se tasapainottaa molempien luokkien F1-arvot, mikä on perusteltua ottaen huomioon aineiston luokkatasapainon. Oppimisprosessin tehokkuutta arvioidaan hyödyntämällä normalisoitua aluetta oppimiskäyrän alapuolella (AULC), joka lasketaan puolisuunnikassääntöä (trapezoidal rule) hyödyntäen. Tämä on keskeinen mittari vertailussa, sillä sen avulla on mahdollista selvittää eroja oppimistehokkuudessa tilanteissa joissa eri menetelmät saavat tietoa eri muodoissa ja eri tahdissa.

3.4 Koeasetelma

Tutkielmassa hyödynnetty koneoppimismalli on satunnaismetsä, sillä se soveltuu hyvin numeeristen ja kategoristen piirteiden käsittelyyn. Satunnaismetsä on vakaa ja paljon tutkittu malli, jonka vuoksi se sopii hyvin kontrolloituun kokeeseen, jossa pyritään minimoimaan satunnaisvaihtelun vaikutus tutkimukseen. Satunnaismetsä on yhdistelmämalli, ja sen hyvä suorituskyky yhdistettynä sen kelvolliseen tulkittavuuteen ovat myös mallin valintaan vaikuttaneita avaintekijöitä. Satunnaismetsän avulla on myös mahdollista tuottaa tärkeysjärjestys piirteistä, mikä mahdollistaa piirteisiin perustuvan koneopettamisen, sillä tässä tutkimuksessa ihmisen tietämys on simuloitu muuttujien poistamiseksi. Samaa mallia käytetään kaikissa kokeen vaiheissa, eli perusmallissa, aktiivisessa oppimisessa ja koneopettamisessa. Tällä pyritään varmistamaan että vertailu mallien välillä on mahdollisimman reilu ja erot johtuvat vain opetustavasta. Malleissa on erotuksena se, että aktiivisen oppimisen ja koneopettamisen tapauksissa ei suoriteta optimointia, vaan annetaan mallien oppia simuloidun opettajan avulla.

3.4.1 Aktiivinen oppiminen

Tutkielmassa keskeinen aktiivinen oppiminen perustuu epävarmuuteen perustuvaan näytteenvalintaan, jossa koneoppimismalli kysyy simuloidulta ihmiseltä vastausta kaikista epävarmimpiin instansseihin. Jokaisella kierroksella valitaan 50 kaikista epävarminta tapausta, joihin simuloitu ihminen antaa oikean merkinnän.

Itse opettaminen tapahtuu jakamalla harjoitusdata siten, että aluksi malli koulutetaan käyttäen 5% harjoitusdatasta, jonka jälkeen tämän mallin suorituskykyä verrataan alkuperäisestä tietoaineistosta erotettuun testidataan. Tässä kohdassa oppimiskäyrä tallennetaan. Tämän jälkeen malli tarkastelee harjoitusdatan ei-merkittyjä instansseja, ja arvioi niistä 50 epävarminta, jonka jälkeen mallin oppimiskäyrää taas tarkkaillaan. Tämä silmukka toistuu kunnes 20 kierrosta on käyty läpi, eli 1000 simuloidun ihmisen merkitsemää instanssia ja alun 5% harjoitusdatasta on käyty läpi. Koko harjoitusdataa ei

siis käytetä mallin kouluttamiseen, sillä näin pyritään osoittamaan aktiivisen oppimisen tehokkuus, sekä mahdollinen kustannustehokkuus eri tilanteissa.

Aktiivisessa oppimisessa toistetaan koe 10 kertaa eri satunnaisilla alkujoukoilla, ja raportoidut tulokset ovat näiden kokeiden keskiarvoja. Tällä pyritään tasoittamaan sitä satunnaisuutta, minkä lähtömallin koulutus aiheuttaa, jossa malli koulutetaan 5% harjoitusdatasta satunnaisesti. Jotta tulos olisi vielä vakuuttavampi, lisättiin epävarmuuteen perustuvan näytteenvalinnan lisäksi aktiivisen oppimisen malli satunnaisuuteen perustuvalla näytteenvalinnalla, jotta saataisiin aktiiviselle oppimiselle myös relevantti vertauskohta.

3.4.2 Koneopettaminen

Simuloitu piirteisiin perustuva koneopettaminen saa olennaiset piirteet optimoidulta satunnaismetsämallilta. Oletuksena tutkielmassa on siis se, että asiantuntijaihminen tietää merkittävät piirteet, mutta tutkielmassa tutkitaan myös tilannetta, jossa ei ole täydellistä tietoa, vaan simuloidulle ihmiselle annetaan 80% ja 60% tarkkuus, jolloin joko 20% tai 40% piirteistä valitaan satunnaisesti vähämerkityksellisten piirteiden joukosta. Vähämerkitykselliset piirteet arvotaan siten, että samaa piirrettä ei ikinä käytetä yhtä kertaa enempää. Tässä tutkielmassa konsepteiksi tulkitaan piirteet, ja näitä piirteitä annetaan mallille vain osittain. Valittujen piirteiden perusteella muodostettiin pienempi aineisto, jota hyödynnettiin satunnaismetsämallin kouluttamiseksi, ja mallin suorituskykyä arvioidaan testiaineistolla, kuten muitakin malleja. Koneopettamisen koe toistetaan 30 kertaa, sillä virheet, eli vähiten merkittävät piirteet, valitaan satunnaisesti piirteiden tärkeysjärjestyksen pohjapuoliskolta. Virheelliset piirteet eivät siis välttämättä ole kaikkein vähämerkityksellisimpiä, mikä tekee simulaatiosta realistisemman. Suorittamalla koe 30 kertaa ja laskemalla tulosten keskiarvo, saamme luotettavaa tietoa, vaikka itse ohjelma ajetaan vain kerran. Lopussa kaikki koneopetussimulaatiot oppivat samat piirteet, koska piirteitä lisätään mallille yksitellen, kunnes ne loppuvat.

3.4.3 Toteutus

Tutkielman kontrolloitu koe on toteutettu Python-ohjelmointikieltä hyödyntäen, käyttäen useita suosittuja koneoppimiskirjastoja. Scikit-learn -kirjastoa käytettiin satunnaismetsämallissa, GridSearchCV-hyperparametrioptimoinnissa, ristiinvalidoinnissa, mittarien tuottamisessa, koulutus- ja testidatan jakamisessa, kirjastosta hyödynnettiin myös OneHotEncoderia ja StandardScaleria datan esikäsittelyssä.

Pandas ja NumPy -kirjastoja käytettiin datan käsittelyyn, esimerkiksi tietoaineiston lataamiseen, ja ei-informatiivisen piirteen poistamiseen. Näiden kirjastojen lisäksi käytettiin myös Matplotlib ja seaborn -kirjastoja visualisointeja varten. Python-koodi ajettiin paikallisessa kehitysympäristössä Windows-käyttöjärjestelmässä. Kokeen tulokset tuotettiin teksti- ja kuvamuodossa, jotka tallennettiin paikallisesti.

Koska kokeessa pyritään hallittuun tilanteeseen ja toistettavuuteen, annettiin kaikille relevanteille funktioille, kuten satunnaismetsälle, harjoitus- ja testidatan jaolle (train_test_split) ja GridSearchCV:lle sama random_state (arvo 42), jolla varmistetaan että kokeen tulokset ovat aina samat, riippumatta siitä, kuinka monta kertaa koe suoritetaan.

Koe etenee vaiheittain, alkaen aineiston latauksesta ja esikäsittelystä. Aineisto on CSV-tiedosto, joka ladataan hyödyntäen Pandas-kirjastoa. Aineistossa tavoitemuuttujana toimii ”attack_detected”, joka kertoo onko hyökkäystä havaittu. Esikäsittely alkaa sillä, että piirre ”session_id” poistetaan, sillä se ei sisällä ennustamisen kannalta merkitsevää tietoa. Esikäsittely jatkuu aineiston jakamisella koulutus- ja testiosioihin, 80% koulutukseen ja 20% testaamiseen. Testiaineistoa ei siis käytetä mallin kouluttamiseen, vaan ainoastaan lopullisen suorituskyvyn arviointiin, ja lisäksi asettamalla arvon ”stratify=y” varmistettiin se, että testi- ja koulutusaineistoissa on sama tavoitemuuttujan jakauma. Numeeriset piirteet skaalattiin, ja kategoriset piirteet

muunnettiin OneHotEncoding -menetelmän avulla myös numeerisiksi. Puuttuvat numeeriset arvot imputoitiin hyödyntäen mediaania, ja kategoriset arvot imputoitiin hyödyntäen yleisintä kategoriaa.

Optimoidun satunnaismetsämallin hyperparametrit optimoitiin hyödyntäen Scikit-learn -kirjaston GridSearchCV:n ristiinvalidointia, jossa taitoksia (fold) oli viisi. Hyperparametrit joita optimoitiin olivat n_estimators, max_depth ja min_samples_split. Aktiivinen oppiminen ja koneopettaminen toteutettiin siten, että ne käyttävät samaa esikäsiteltyä aineistoa, mutta hieman yksinkertaisempaa satunnaismetsää kuin optimoitu perusmalli. Aktiivisessa oppimisessa ja koneopettamisessa n_estimators on asetettu arvoon 50. Aktiivisen oppimisen toistettavuus varmistettiin aiemmin mainitulla kymmenen iteraation keskiarvolla, ja koneopettamisessa 30 toiston keskiarvolla. Aktiivisen oppimisen kohdalla varmistettiin myös se, että jokaisessa iteraatiossa valittava lähtödata on aina satunnainen.

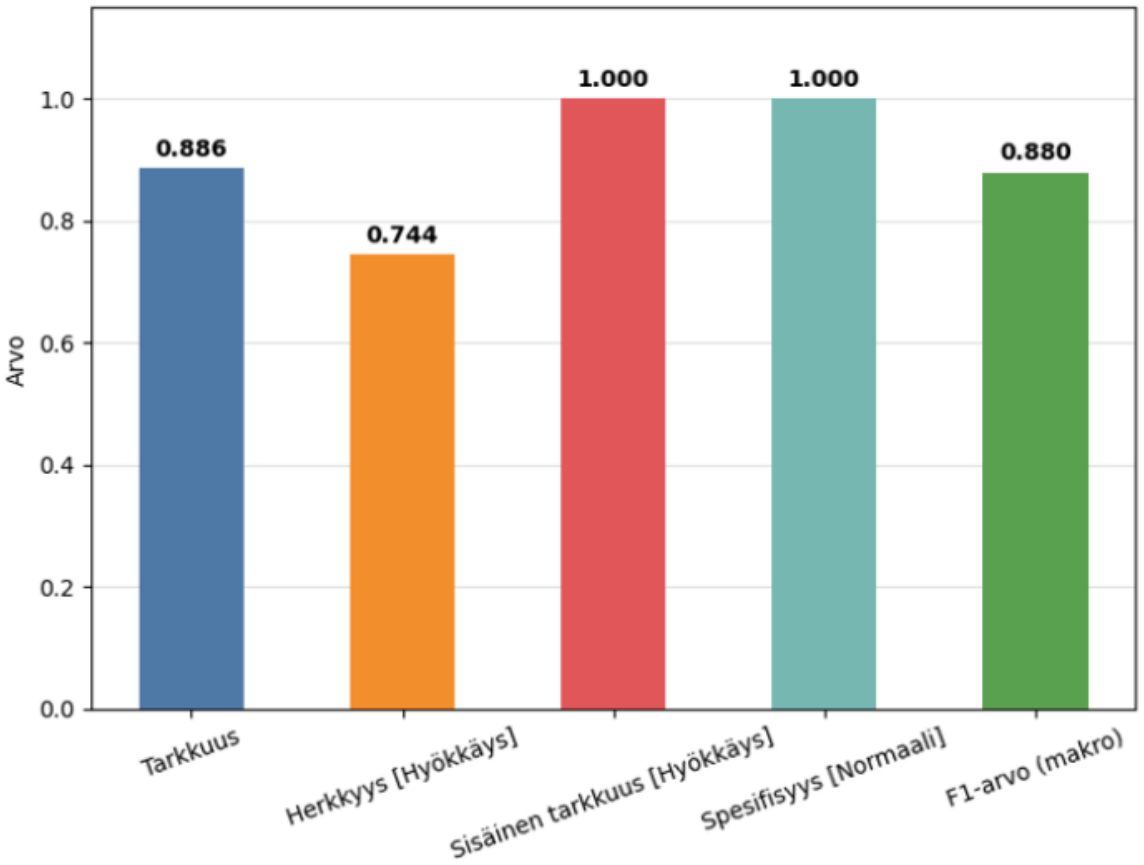
Mallien suorituskykyä mitattiin aiemmin määritetyillä mittareilla (tarkkuus, herkkyys, sisäinen tarkkuus, spesifisyys, F1-makrokeskiarvo), ja näiden lisäksi normalisoidulla analyysilla alueesta oppimiskäyrän alapuolella (AULC). Suorituskyvyn mittarit on määritelty heti koodin alussa, jotta matemaattiset kaavat ovat helposti nähtävillä. Näillä implementaation tavoilla ja ohjelmointilogiikalla varmistettiin se, että ainoana muuttujana mallien välillä on koulutustapa.

4 Tulokset

Tässä luvussa käydään läpi tulokset, jotka saatiin edellisessä kappaleessa kerrotuilla menetelmillä. Tämän lisäksi tuloksia verrataan keskenään, sekä avataan niiden merkitystä. Tulokset esitellään loogisessa järjestyksessä seuraten edellisessä luvussa olevaa menetelmien järjestystä. Tulokset siis esitellään aloittaen optimoidusta satunnaismetsämallista, sillä se toimii lähtökohtana tulosten vertailulle. Tämän jälkeen avataan aktiivisen oppimisen tuloksia, sillä niitä hyödynnetään myös vertailukohtana koneopettamiselle. Koneopettamisen tulokset esitellään viimeisenä, jonka jälkeen saatuja tuloksia vertaillaan keskenään, sekä kerrotaan niiden tulkitsemisesta. Kaikkia menetelmiä arvioidaan aiemmissa luvuissa esitellyillä mittareilla, joka mahdollistaa eri mallien reilun arvioimisen.

4.1 Optimoitu satunnaismetsämalli

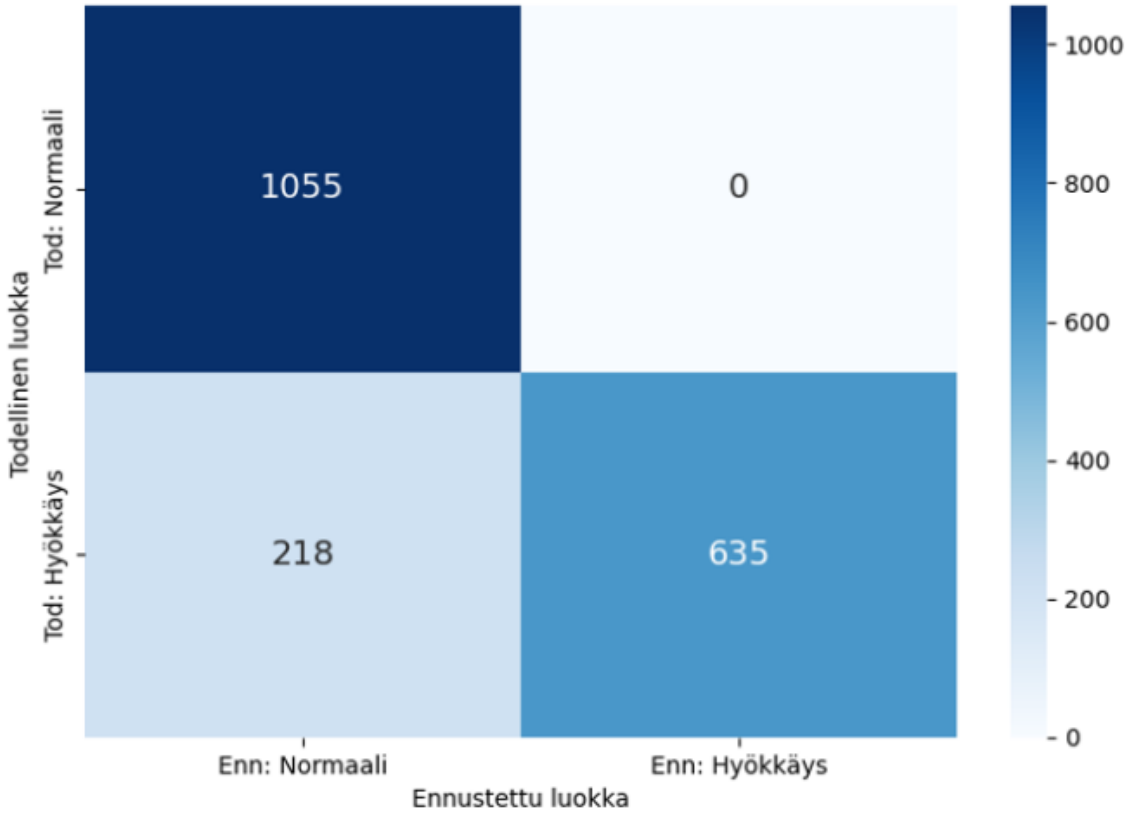
Tutkielmassa lähtökohtana toimiva optimoitu satunnaismetsämalli saavuttaa tarkkuuden 0,886. Tämä kertoo siitä, että malli onnistuu ennustamaan suuren enemmistön instansseista oikein.



Kuvio 4. Optimoidun satunnaismetsän suorituskyvyn mittarit.

Kuvio 4 havainnollistaa graafisesti optimoidun satunnaismetsän suorituskykyä, kuviosta myös näkee tarkat arvot joita pylväsdiagrammit esittävät. Tarkkuus on 0,886, herkkyys on 0,744, sisäinen tarkkuus on 1, spesifisyys on 1 ja F1-makrokeskiarvo on 0,880. Malli siis ennustaa kaikista tapauksista noin 89% oikein, mikä on hyvä tulos, mutta myös positiivista on se, että mikään näistä luokittelumallien mittareista ei ole merkittävän huono, joten mallin suorituskyky on yleisellä tasolla hyvä. Tuloksista huonoin on herkkyys, joka tässä tilanteessa ennustaa kaikista hyökkäyksistä noin 74% oikein, ja luokittelee loput 26% hyökkäyksistä ei-hyökkäyksiksi. Eli malli ei siis huomaa kaikkia hyökkäyksiä. Sisäinen tarkkuus on 1, mikä tarkoittaa sitä, että kun malli luokittelee instanssin hyökkäykseksi, se on aina oikeassa, eli 100% kerroista. Nämä herkkyyden ja sisäisen tarkkuuden tulokset kertovat mallista siis sen, että se ei aina havaitse kaikkia hyökkäyksiä, mutta aina kun se ennusti instanssin olevan hyökkäys, se oli oikeassa.

Mallin spesifisyys on myös 1, mikä tarkoittaa sitä että malli ei ikinä väärin ennusta tilanteen olevan hyökkäys. Eli jos kyseessä on ”normaali” tilanne, jossa ei ole hyökkäystä, malli ennustaa sen aina oikein. F1-makrokeskiarvo on 0,880, eli malli saavuttaa molemmissa luokissa keskimäärin hyvän tasapainon sisäisen tarkkuuden ja herkkyyden välillä.



Kuvio 5. Optimoidun satunnaismetsämallin sekaannusmatriisi (confusion matrix).

Kuvion 5 sekaannusmatriisi havainnollistaa sitä, miten malli luokitteli testiaineiston instanssit. Kuviossa tilanteet, joissa ei ollut hyökkäystä on merkitty normaaleiksi. Tämä matriisi kuvastaa samaa mikä jo mittareista huomattiin, eli kuten vasemmasta yläkulmasta huomaa, 1055 instanssia jossa ei ollut hyökkäystä luokiteltiin oikein. Ja kuten oikeasta yläkulmasta näkee, nolla tilannetta jossa ei ollut hyökkäystä luokiteltiin hyökkäykseksi. Eli kaikki normaalit tilanteet luokiteltiin oikein.

Matriisin alaosasta näkee todelliset hyökkäykset, joita oli 853. Näistä malli ennusti 635 oikein, eli hyökkäyksiksi, mutta malli ennusti väärin 218 hyökkäysinstanssia. Tämän takia herkkyys on pienempi kuin spesifisyys ja sisäinen tarkkuus, sillä herkkyys keskittyy nimenomaan vääriin negatiivisiin.

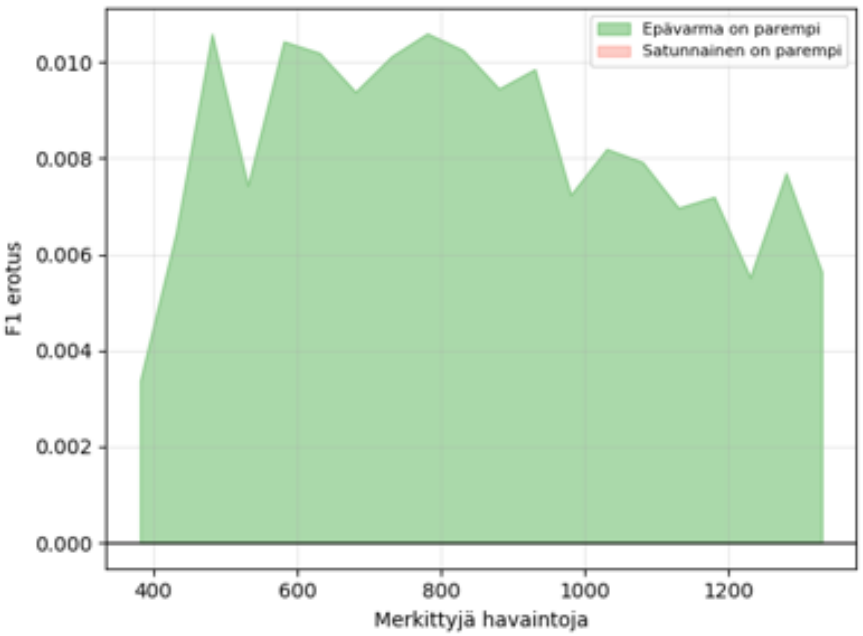
4.2 Aktiivinen oppiminen

Aktiivisen oppimisen simulaatiossa malli koulutettiin aluksi käyttäen vain 5% harjoitusdatasta, jonka jälkeen oraakkelilta (simuloitu ihminen) kysyttiin 50 epävarmimman instanssin merkintää 20 kierroksen ajan. Harjoitusdatan 7629 havainnosta siis käytettiin kokonaisuudessaan vain pieni osa, minkä avulla pyrittiin osoittamaan aktiivisen oppimisen kustannustehokkuus. Aktiivisesta oppimisesta saadut tulokset ovat 10 kokeen keskiarvo, jolloin alun koulutuksen satunnaisvaihtelun muutokset eivät tee tuloksista niin satunnaisia.

Epävarmuuteen perustuvan näytteenvalinnan aktiivinen oppiminen saavuttaa 0,885 tarkkuuden, mikä on erittäin hyvä tulos. Epävarmuuteen perustuvan näytteenvalinnan herkkyys on 0,746, sisäinen tarkkuus on 0,996, spesifisyys on 0,998 ja F1-makrokeskiarvo on 0,880. Epävarmuuteen perustuvan näytteenvalinnan aktiivinen oppiminen saavuttaa siis pienemmällä osalla tiedosta tuloksen, joka on optimoituun malliin nähden hyvin samankaltainen. Tarkkuus, sisäinen tarkkuus ja spesifisyys ovat vähän huonompia kuin optimoidussa mallissa, mutta puolestaan herkkyys on parempi ja F1- makrokeskiarvo on sama. Tämä viittaa siihen, että epävarmuuteen perustuvalla näytteenvalinnalla saadaan tällä aineistolla tehokkaammin saavutettua hyvän suorituskyvyn. Kuitenkin erot mittareissa ovat niin pieniä, että ne voidaan luokitella satunnaisvaihteluksi, joten näiden perusteelta on vaikea tehdä johtopäätöksiä menetelmien paremmuudesta.

Satunnaisuuteen perustuvaa näytteenvalintaa hyödyntävä aktiivinen oppiminen oli tutkielmassa myös koulutettu, sillä sen avulla voidaan tutkia tarkemmin miten

epävarmuuteen perustuva näytteenvalinta oikeasti oppii. Satunnaisuuteen perustuvaa näytteenvalintaa hyödyntävä aktiivinen oppiminen saavutti tarkkuuden 0,880, herkkyyden 0,747, sisäisen tarkkuuden 0,987, spesifisyyden 0,987 ja F1-makrokeskiarvon 0,874. Ero siis näiden aktiivisen oppimisen menetelmien välillä on hyvin pieni.



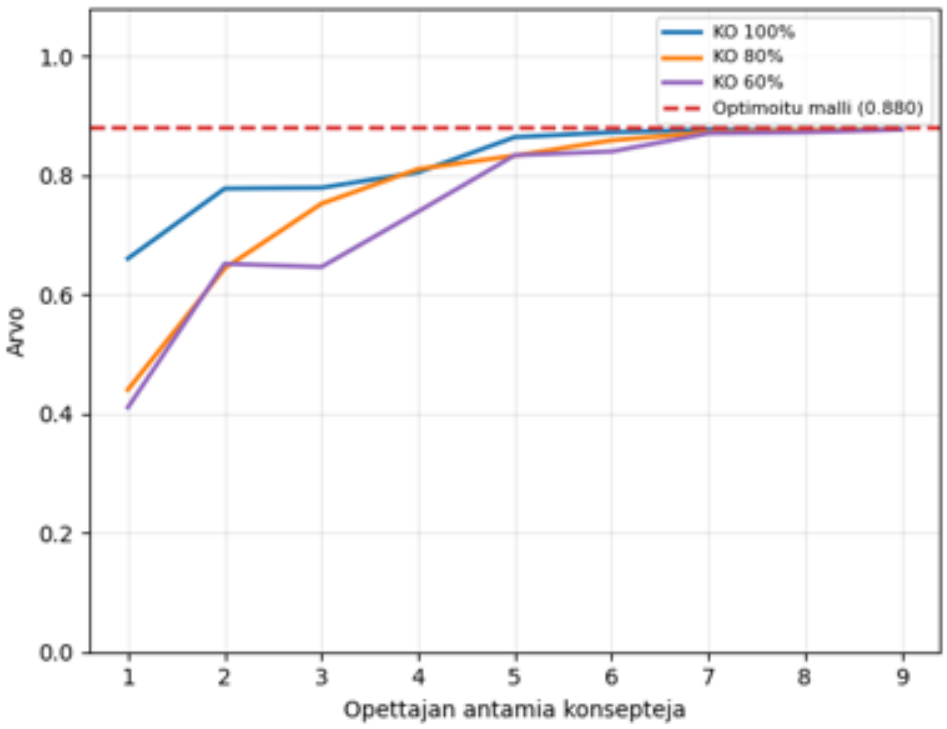
Kuvio 6. Epävarmuuteen perustuvan näytteenvalinnan F1-makrokeskiarvo miinus satunnaisuuteen perustuvan näytteenvalinnan F1-makrokeskiarvo.

Kuvio 6 kuvastaa epävarmuuteen perustuvan näytteenvalinnan ja satunnaisuuteen perustuvan näytteenvalinnan eroja aktiivisessa oppimisessa perustuen F1-makrokeskiarvoihin. Epävarmuuteen perustuva näytteenvalinta on siis parempi vaihtoehto koko koulutuksen ajan, vaikka ero onkin pieni. Kuviosta näkee, että epävarmuuteen perustuvan näytteenvalinnan suurin hyöty on oppimisprosessin keskivaiheilla, eli noin 600-800 harjoitusdatan instanssin kohdalla. Alussa F1-makrokeskiarvojen erot ovat pieniä, mikä tarkoittaa sitä että molemmilla näytteenvalinnan tavoilla malli oppii hyvin. Lopussa on myös havaittavissa että satunnaisuuteen perustuva näytteenvalinta alkaa saavuttamaan epävarmuuteen perustuvaa näytteenvalintaa. Tämän voi selittää se, että lopussa molemmilla

näytteenvalinnan tavoilla on niin monta merkittyä instanssia, että valintaperusteella ei enää ole niin suurta merkitystä mallin F1-makrokeskiarvolle.

4.3 Koneopettaminen

Koneopettamisesta saadaan kolme eri tulosta, tarkkuuksien 100%, 80% ja 60% tulokset. Kokeissa opetettiin piirteitä yksi kerrallaan, jonka takia oli mahdollista saada selkeät käyrät suoritusmittareista. Tämän lisäksi jokainen koneopettamistarkkuus toistettiin 30 kertaa, sillä huonojen piirteiden valinta on satunnaista, niin tämän avulla saatiin epävarmuuden vaikutus minimoitua.



Kuvio 7. Koneopettamisen eri tarkkuuksien F1-makrokeskiarvot verrattuna optimoidun mallin F1-makrokeskiarvoon.

Kuten kuvion 7 kaaviosta huomataan, koneopettaminen saavuttaa melkein optimoidun mallin F1-makrokeskiarvon kaikissa tapauksissa. Koneopettaminen 100% tarkkuudella, eli malli joka saa vain hyödyllisiä piirteitä tärkeysjärjestyksessä lähestyy optimoidun

mallin F1-makrokeskiarvoa kaikista tehokkaammin. Tämä tulos ei ole yllättävä, toisin kuin se, että myös vähämerkityksellisiä piirteitä oppivat mallit lähestyvät optimoidun mallin F1-makrokeskiarvoa hyvin tehokkaasti. Eli opettajan virheet voivat hidastaa oppimista, mutta ne eivät kuitenkaan estä mallia saavuttamasta hyvää suorituskykyä vähäisellä konseptien määrällä.

Täydellä tarkkuudella koulutettu malli saavuttaa 80% optimoidun mallin F1-makrokeskiarvoa jo kahdella piirteellä, kun taas 80% tarkkuudella opettava malli tarvitsi 3 piirrettä ja 60% tarkkuudella opettava malli tarvitsi 4 piirrettä sen saavuttamiseksi. Tämän selittää se, että mallien piirteiden opettamisen laskukaavan mukaan 80% tarkkuuden mallilla ensimmäinen piirre on vähämerkityksellinen, jota seuraavat piirteet ovat merkittäviä, eli kolmella piirteellä kaksi on parhaita ja yksi on vähämerkityksellinen. Ja vastaavasti, 60% tarkkuuden mallilla tarvitaan 4 piirrettä, että malli saavuttaa 80% optimoidun mallin F1-makrokeskiarvosta.

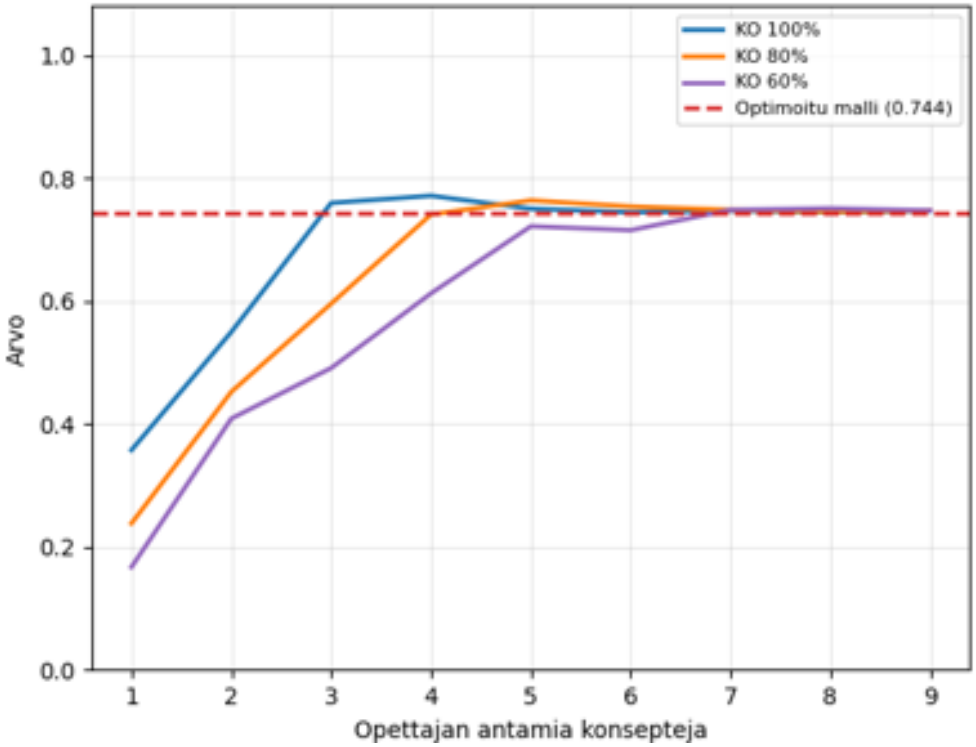
Kaikki koneopettamistapaukset lopuksi saavuttaa saman suorituskyvyn, sillä ne kaikki oppivat lopuksi kaikki yhdeksän piirrettä. Kuitenkin, lopputulos on hieman erilainen ja koneopetuksen suorituskyvyt ovat hieman eroavia.

Taulukko 1. Koneopettamisen lopulliset suorituskyvyt.

Koneopettamisen tarkkuus	Tarkkuus	Herkkyys	Sisäinen tarkkuus	Spesifisyys	F1-makrokeskiarvo
60%	0,8835	0,7474	0,9893	0,9935	0,8778
80%	0,8841	0,7473	0,9914	0,9948	0,8785
100%	0,8836	0,7468	0,9907	0,9943	0,8780

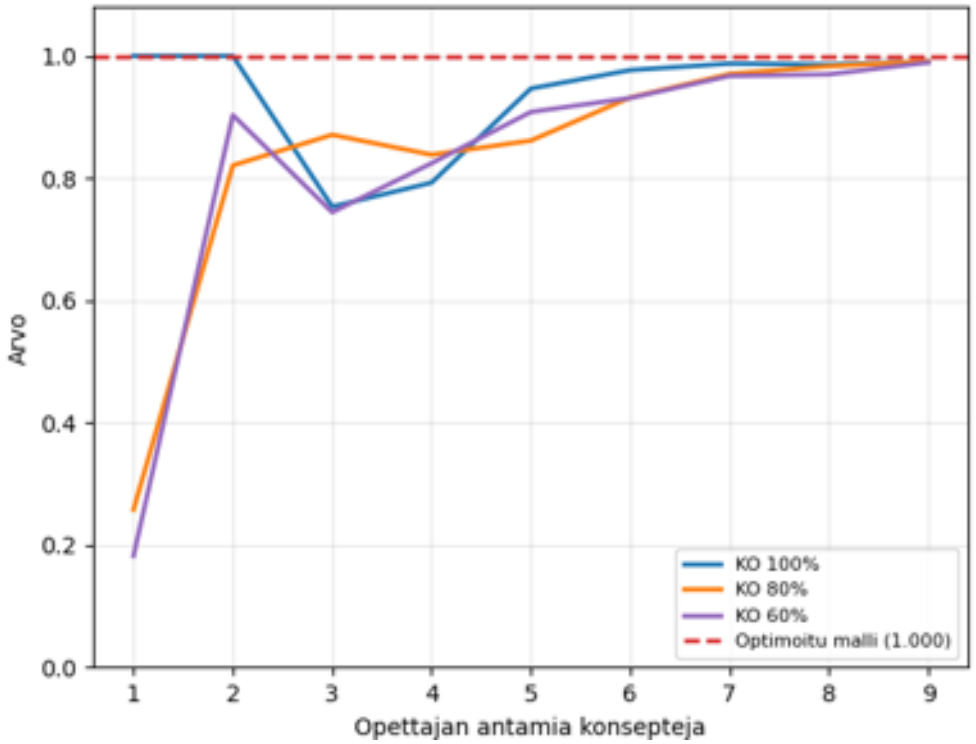
Taulukko 1 havainnollistaa koneopettamisen lopullisia mittareita. Arvojen pitäisi olla samat, sillä kaikki mallit ovat lopussa koulutettu samalla yhdeksällä piirteellä. Pieniä eroja malleissa voi selittää se, miten satunnaismetsä toimii. Metsää voi muuttaa esimerkiksi piirteiden järjestys, joka onkin tässä tilanteessa todennäköisin syy

lopputulosten pienelle vaihtelulle. Erot ovat kuitenkin minimaalisia, joten näillä tuloksilla ei ole vaikutusta tutkielman löydöksiin. Tutkimalla eri mittareita tarkemmin saa hyvän käsityksen koneopettamisen eri tarkkuuksien oppimiskäyristä.



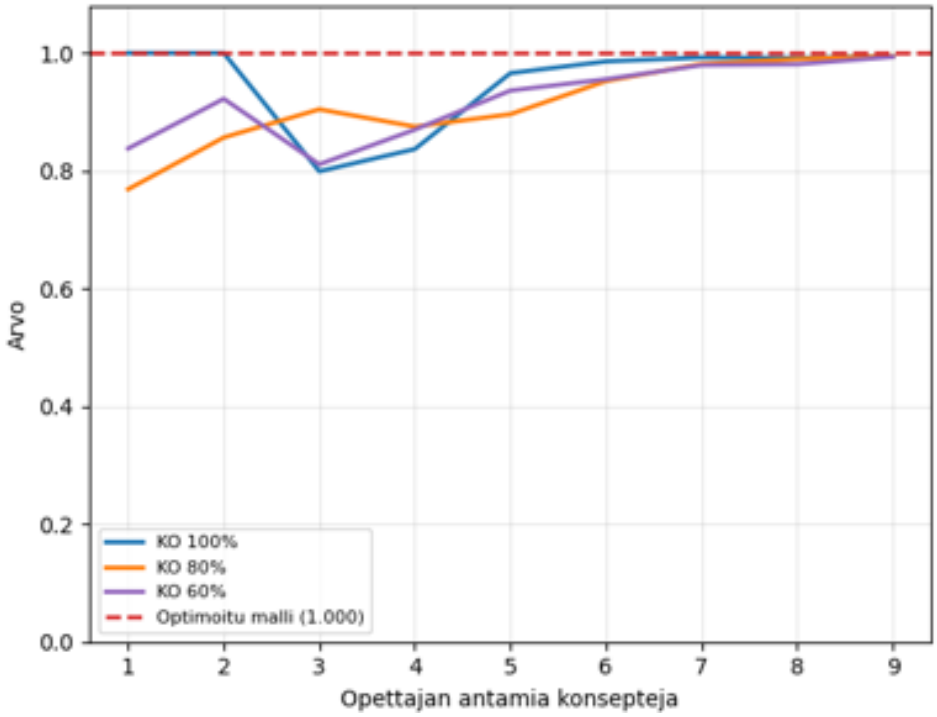
Kuvio 8. Koneopettamisen herkkyydet verrattuna optimoituun malliin.

Kuvio 8 havainnollistaa koneopettamisen herkkyyttä verrattuna optimoituun pohjamalliin. Kuviossa 100% tarkka opettaja saavuttaa optimoitua paremman mallin piirteiden 3-5 kohdalla, ja muut tarkkuudet seuraavat perässä. Kuviosta voi siis tulkita, että kolme merkittävintä piirrettä (failed_logins, login_attempts, ip_reputation_score) antavat hyvin tietoa hyökkäysten ennustamiseksi, jonka jälkeen muut piirteet saavat mallin ennustamaan niitä hieman huonommin.



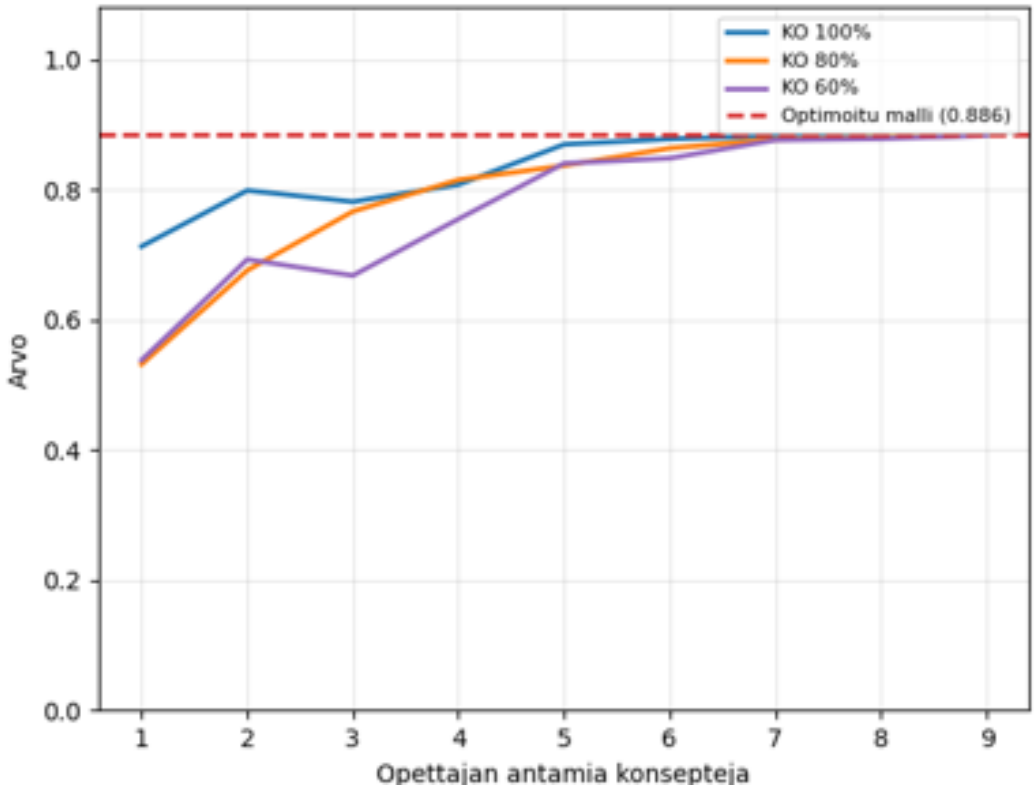
Kuvio 9. Koneopettamisen sisäinen tarkkuus verrattuna optimoituun malliin.

Kuvio 9 havainnollistaa koneopettamisen sisäisiä tarkkuuksia verrattuna optimoituun malliin. Kolmannen annetun konseptin kohdalla täysin tarkan opettajan koneopettamismallin sisäinen tarkkuus heikkenee huomattavasti, mikä heijastaa herkkyyden ja sisäisen tarkkuuden välistä suhdetta. Kolmas piirre saa mallin arvaamaan hyökkäyksiä enemmän, ja samalla se hetkellisesti luokittelee muutaman normaalin tilanteen hyökkäykseksi. Tämän sisäisen tarkkuuden heikkenemisen jälkeen kuitenkin mallit oppivat lisäpiirteistä selkeästi tehokkaasti, sillä sisäinen tarkkuus nousee lähes optimoidun mallin tasolle.



Kuvio 10. Koneopettamisen spesifisyys verrattuna optimoituun malliin.

Koneopettaminen saavuttaa pohjamallin myös spesifisyydessä, mikä tarkoittaa siis sitä, ettei malli ennusta normaalin tilanteen olevan hyökkäys. Spesifisyydessä on myös notkahdus alkuvaiheen opetuksessa, mikä viittaa samaan havaintoon kuin sisäisen tarkkuuden notkahdus. Malli kokeilee siis arvata enemmän hyökkäyksiä, ja siten luokittelee muutaman normaalin tilanteen hyökkäykseksi. Kaikki opetustarkkuudet kuitenkin pääsevät notkahduksen yli ja piirteiden lisääntyessä saavuttavat optimoidun mallin tasoa.



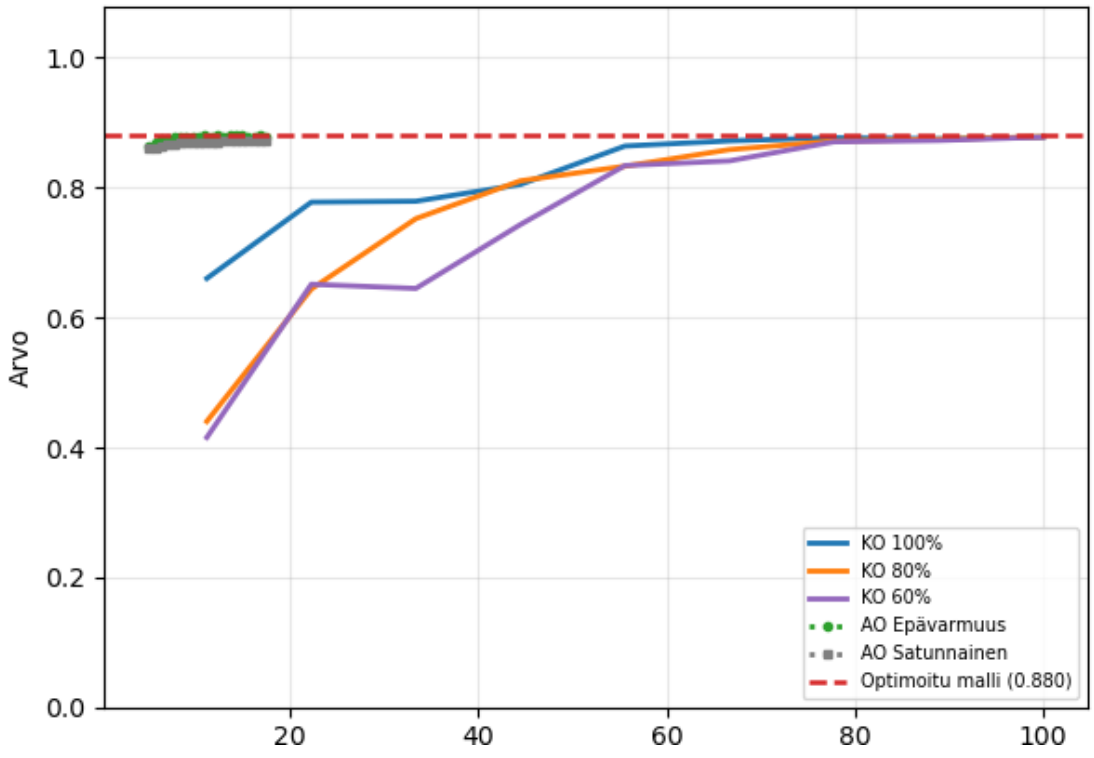
Kuvio 11. Koneopettamisen tarkkuus verrattuna optimoituun malliin.

Tarkkuus seuraa pitkälti samaa trendiä kuin muut suorituskyvyn mittarit. Viiden piirteen kohdalla kaikki koneopettamisen tarkkuudet lähes saavuttavat optimoidun mallin tarkkuuden. Lisäksi pieni notkahdus piirteen kolme kohdalla on myös tässä havaittavissa 100% tarkkuuden opettajalla, mikä voi kertoa siitä, että kun malli oppii ennustamaan myös hyökkäyksiä, niin tarkkuus voi kärsiä siitä. Tarkkuus kuitenkin kehittyy selkeästi lisäpiirteiden avulla, ja piirteiden lisääminen vähentääkin vaihtelua, ja lopussa kehitys lisäpiirteillä pienenee.

4.4 Menetelmien vertailu

Tällä aineistolla, jokainen eri koulutusmenetelmä saavuttaa lopulta lähes samankaltaisen suorituskyvyn. F1-makrokeskiarvoa vertaamalla, huomaamme lopullisten erojen olevan hyvin pieniä, optimoitu malli saavuttaa F1-makrokeskiarvon 0,880, epävarmuuteen perustuvalla näytteenvalinnalla koulutetulla aktiivisella oppimisella saavutetaan F1-

makrokeskiarvo 0,8795, satunnaisuuteen perustuvalla näytteenvalinnalla koulutettu aktiivisen oppimisen malli saavuttaa F1-makrokeskiarvon 0,8739. Koneopettamisen puolella taas 100% tarkan opettajan malli saavuttaa F1-makrokeskiarvon 0,8780, 80% tarkka opettaja saavuttaa F1-makrokeskiarvon 0,8785 ja 60% tarkan opettajan malli saavuttaa F1-makrokeskiarvon 0,8778. Nämä koneopettamisen F1-makrokeskiarvot johtuvat satunnaismetsän sisäisestä piirteiden satunnaisotannasta, joka on herkkä piirteiden sarakejärjestykselle. Kun kaikki piirteet ovat lopulta mukana, mallit ovat käytännössä samat ja erot ovat satunnaisvaihtelun rajoissa. Siksi voikin yleistää koneopettamisen saavuttavan F1-makrokeskiarvon 0,88.

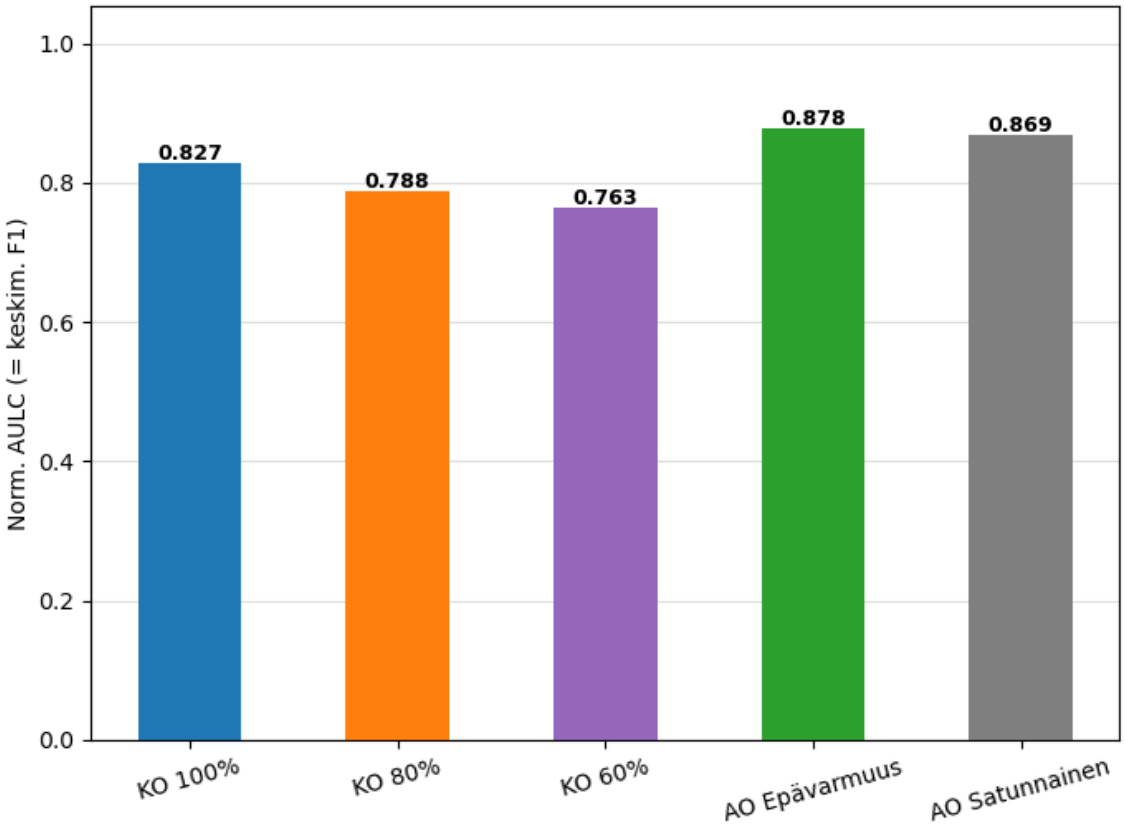


Kuvio 12. Eri menetelmien F1-makrokeskiarvot normalisoituna saadun informaation määrän mukaisesti.

Kuvio 12 visualisoi tämän koulutuksen ja F1-makrokeskiarvon kehityksen, tästä erityisen mielenkiintoinen huomio on aktiivisen oppimisen tehokkuus. Sillä ne menetelmät koulutetaan vain pienellä määrällä harjoitusdatasta, mutta ne saavuttavat samankaltaisen F1-makrokeskiarvon kuitenkin hyvin tehokkaasti. Alun 5%

harjoitusdatasta jolla aktiivisen oppimisen mallit koulutetaan antavat siis jo hyvän pohjan mallin toiminnalle, ja annetut merkinnät antavat myös riittävästi tietoa mallille.

Tällä aineistolla siis koulutusmenetelmä ei merkittävästi määritä saatua suorituskykyä, ne vain vaikuttavat siihen miten se saavutetaan, ja kuinka tehokkaasti. Tämän vuoksi onkin tarkoituksenmukaista tutkia eri mallien oppimiskäyriä hyödyntäen aluetta oppimiskäyrän alapuolella (AULC).



Kuvio 13. Normalisoitu AULC (F1-makrokeskiarvo) molemmilla tutkittavilla menetelmillä.

Kuvio 13 tiivistää menetelmien F1-makrokeskiarvon koko oppimisprosessin ajalta, suurempi arvo tarkoittaa sitä, että menetelmä saavuttaa suuren F1-makrokeskiarvon nopeasti ja säilyttää sen koko oppimisprosessin ajan. Pienempi arvo tarkoittaa sitä, että vaikka menetelmä saavuttaisi hyvän F1-makrokeskiarvon lopulta, niin se viettää oppimisprosessin alkuvaiheen heikommalla tasolla, oppiminen ei siis ole niin tehokasta koko prosessin aikana. Parhaan arvon saavuttaa epävarmaan näytteenvalintaan

perustuva aktiivinen oppiminen, ero satunnaiseen näytteenvalintaan perustuvaan aktiiviseen oppimiseen on kuitenkin pieni. Koneopettamisen menetelmät puolestaan seuraavat oletettua järjestystä, jossa 100% tarkkuudella opetettu malli oppii koko prosessin aikana parhaiten, ja muut tarkkuudet ovat heikompia. Opettajan virheet siis hidastavat oppimisprosessia selkeästi, vaikka ne eivät estäkään lopullista suorituskyvyn saavuttamista.

Normalisoitu AULC ei kuitenkaan ole täydellinen mittari, sillä koneopettaminen ja aktiivinen oppiminen saavat tiedon eri muodossa. Aktiivinen oppiminen saa tietoa merkittyjen instanssien muodossa, kun koneopettaminen puolestaan saa tiedon piirteiden muodossa. Eri menetelmien tulokset tarkoittavat siis eri asioita, joten nämä eivät suoraan kerro menetelmien paremmuudesta toisiinsa verrattuina.

Saadut tulokset antavat asetetuille tutkimuskysymyksille selkeät vastaukset. Tutkimuskysymys 1 keskittyy koneopettamisen suorituskykyyn verrattuna optimoituun satunnaismetsämalliin, tutkien nimenomaan virheiden vaikutusta piirteiden valinnassa. Tulosten perusteella opettajan tekemät virheet oppimisprosessin aikana heikensivät mallin suorituskykyä hyvin vähän. Kun tuloksia verrataan F1-makrokeskiarvoa tarkastellen, huomataan tämä selkeästi, mikä viittaa siihen, että koneopettaminen säilyttää tässä simulaatiossa toimintakykynsä myös tilanteissa joissa opettajan tieto on puutteellista. Tulokset siis osoittavat, että koneopettaminen on ainakin tässä tutkimusasetelmassa hyvin kilpailukykyinen mallin opettamismenetelmä, verrattuna optimoituun malliin.

Toinen tutkimuskysymys keskittyy koneopettamisen vertaamiseen aktiivisen oppimisen kanssa, ja koneopettamisen vikasietoisuuden tutkimiseen. Tulokset osoittavat, että aktiivinen oppiminen epävarmuuteen perustuvalla näytteenvalinnalla saavutti optimoidun mallin F1-makrokeskiarvon, ja koneopettaminen oli myös hyvin lähellä, saavuttaen yli 99% optimoidun mallin F1-makrokeskiarvosta. Aktiivinen oppiminen käytti vain pientä osaa harjoitusdatasta, mikä tukee menetelmän tutkittua

oppimistehokkuutta ja kustannustehokkuutta. Koneopettamisen tulokset puolestaan osoittivat että mallia voidaan erittäin hyvin ohjata myös piirteiden avulla ilman täydellistä tietoa. Tätä tukee tulos siitä, että kaikki koneopettamisen tarkkuudet saavuttavat 80% optimoidun mallin suorituskyvystä viimeistään viidennen opetetun piirteen kohdalla, vaikka mukaan olisikin tullut virheitä.

5 Diskussio

Tutkielmassa suoritettiin kontrolloitu koe, jolla tutkittiin simuloidusti ihmisen osallistumista satunnaismetsämallissa. Tutkielmassa ihmisen osallistuminen oli simuloitu, sillä siten oli mahdollista poistaa mahdollisimman paljon muuttujia kokeesta. Aiheesta tehty tutkimus keskittyy pitkälti teoriaan, tai ylipäänsä näiden menetelmien hyödyntämiseen. Tutkimusta tilanteissa jossa virheitä tapahtuu on hyvin vähän, ja piirteitä hyödyntävää opettajaa ei ole tutkittu, lisäksi nimenomaan käytännön testejä ei ole suoritettu riittävästi. Optimoitu satunnaismetsämalli saavutti parhaan suorituskyvyn, mikä ei ole yllättävä tulos ottaen huomioon tutkielmassa hyödynnetyn hyperparametrien optimoinnin funktion. Tutkielmassa keskityttiin osoittamaan aktiivisen ja koneopettamisen mahdollinen hyödyllisyys menetelminä, keskittyen koneopettamiseen tilanteessa, jossa myös virheitä on simuloitu.

Simulaatio toteutettiin optimoimalla satunnaismetsämalli hyödyntämällä Scikit-learn -kirjaston GridSearchCV:n ristiinvalidointia. Aktiivinen oppiminen toteutettiin epävarmuuteen perustuvalla näytteenvalinnalla, jossa malli aluksi koulutettiin käyttämällä 5% olemassa olevasta harjoitusdatasta. Tämän lisäksi kokeeseen lisättiin myös satunnaisuuteen perustuvaa näytteenvalintaa hyödyntävä aktiivisen oppimisen tilanne, jotta olisi mahdollista saada kattavampi kuva. Koneopettaminen simuloitiin ensin selvittämällä piirteiden vaikutuksen optimoidun mallin ennustukseen, ja sitten antaen piirteitä uudelle mallille yksitellen. Koneopettamisessa hyödynnettiin kolmea eri tarkkuustasoa, 60%, 80% ja 100%. Kun verrataan eri mallien F1-makrokeskiarvoja, huomaa erojen pieneyden selkeästi. Optimoitu malli saavutti F1-makrokeskiarvon 0,880, kun taas 100% tarkkuuden koneopettaminen saavutti F1-makrokeskiarvon 0,8780. Eri tarkkuuden koneopettamisen tilanteet saavuttivat erilaiset F1-makrokeskiarvot, mutta se rajataan tässä tutkielmassa tulosten tarkastelussa ulkopuolelle, sillä piirteiden järjestys saattaa muuttaa metsää, vaikka lopullinen tulos pitäisi olla sama.

5.1 Pohdinta

Tutkielman tulokset osoittavat sen, että koneopettamisesta sekä aktiivisesta oppimisesta voi olla hyötyä nimenomaan oppimistehokkuuden näkökulmasta. Vaikka optimoitu malli suoritti parhaiten tällä aineistolla, saavuttivat nämä eri simuloidut ihmisen osallistumisen tavat lähes yhtä hyvät suorituskyvyt. Aktiivisen oppimisen tapauksessa käytetty aineisto oli vain pieni osa koko aineistosta, mikä osoittaa aktiivisen oppimisen tehokkuuden tilanteessa, kun tietoa on saatavilla vähemmän, ja jossa sitä merkitsee oraakkelina toimiva ihminen. Kokeen tulos viittaa myös siihen, että ihmisen osallistumisella koneopettamisessa voi olla potentiaalia piirteiden opettamisen muodossa, myös niissä tilanteissa jossa ihminen ei opeta vain optimaalisia piirteitä.

Työn keskeinen kontribuutio on siis epätäydellisen opettajan simulointi koneopettamisessa, sekä tämän menetelmän vertaaminen aktiivisen oppimisen menetelmiin, jossa simuloitu opettaja toimii oraakkelina. Simuloidulla aktiivisella oppimisella saatu tulos on linjassa aiemman tutkimuksen kanssa, jonka mukaan aktiivinen oppiminen on käyttökelpoinen menetelmä tilanteissa, jossa merkittyä dataa on niukasti, ja käytössä on oraakkeli. Aktiivinen oppiminen saavutti optimoidun mallin F1-makrokeskiarvon, mutta käytti vain pienen osan datasta, tukien aktiivisen oppimisen tutkimuksen näkemystä siitä, että informatiivisten instanssien valinta voi parantaa oppimistehokkuutta. Tämä on linjassa esimerkiksi Mosqueira-Reyn ja muiden (2023, s. 3011) tutkimukseen, jossa todetaan aktiivisen oppijan pyrkivän mahdollisimman suureen tarkkuuteen, käyttäen mahdollisimman vähän merkittyjä instansseja, ja siten minimoida saadun merkityn datan hinta.

Koneopettamisen kontribuutio on hyvin keskeinen. Tutkielman tulokset viittaavat siihen, että vaikka opettaja tekisi opetustilanteessa virheitä, siitä voi silti olla hyötyä. Mosqueira-Reyn ja muiden (2023, s. 3023) mukaan koneopettamisen tutkimus keskittyy opettajien tehokkuuteen, mitä on tässä tutkielmassa myös simulaatiolla tutkittu. Tässä tutkielmassa kuitenkin keskityttiin tutkimaan epätäydellisen opettajan seuraamuksia, ja pysyykö opettamisprosessi kuitenkin tehokkaana. Mosqueira-Rey ja muut (2023, s. 3024)

mainitsevat, että koneopettamisprosessissa opettajan pitäisi pystyä selittämään relevantit konseptit joiden pohjalta dokumentti on merkitty tiettyyn luokkaan kuuluvaksi. Tätä ajatusta myös hiotaan tässä tutkielmassa, jossa piirteitä käytetään konsepteina, sekä simuloidaan tilannetta jossa malli oppii myös vähemmän merkittäviä piirteitä.

Käytännössä tutkielman tulokset viittaavat siihen, että koneoppimismalleissa voidaan hyödyntää asiantuntijatietoa, vaikka tietämys ei olisi täydellistä. Koneopettamisen suorituskyky säilyi hyvänä, vaikka opettajalle simuloitiin virheitä piirteiden valinnassa. Tämä on merkittävä tulos, sillä sen pohjalta voi oikeuttaa käytännössä ihmisen osallistumisen koneoppimisprosessiin. Usein käytännön tilanteissa ei ole saatavilla yhtä kattavasti tietoa, ja toimivaa aineistoa kuin tässä tutkielmassa, joten ihmisen osallistumisesta voisi saada lisähyötyä, ja kuten tutkielma osoittaa, virheet eivät välttämättä aiheuta kriittistä haittaa mallille. Aktiivisen oppimisen tulokset puolestaan osoittavat, että hyvä suorituskyky voidaan saavuttaa pienemmällä määrällä merkittyä dataa. Tämä voi vähentää datan merkitsemiseen liittyviä erilaisia kustannuksia, erityisesti alueilla, joissa datan manuaalinen luokittelu on hidasta tai kallista.

Tulokset siis tukevat ajatusta siitä, että ihmisten osallistumista koneoppimisessa voitaisiin tarkastella realistisemmasta näkökulmasta, joka ei oleta ihmisten olevan täydellisiä oraakkeleja. Aktiivisen oppimisen ja koneopettamisen vertailu osoittaa, että molemmilla menetelmillä voidaan saavuttaa kilpailukykyinen suorituskyky myös tilanteissa, joissa käytettävissä on vähemmän merkittyä dataa tai opettaminen ei ole täydellistä.

5.2 Rajoitteet

Tutkielmassa on useita rajoitteita, jotka tarkoittavat sitä, että tutkielman tulosta voi olla vaikea yleistää laaja-alaisesti. Yksi keskeinen rajoite tutkielmassa on se, että ihmisen osallistuminen on täysin simuloitu. Oikean ihmisen käyttäytymistä on käytännössä mahdotonta simulaatiolla mallintaa täydellisesti, eli vaikka koneopettamisessa

simuloitiin ihmisen virheitä eri virhemarginaaleilla, ei kuitenkaan tulosta pysty täysin yleistämään, sillä ihmisten käyttäytyminen saattaa olla erilaista riippuen useista eri asioista, esimerkiksi ihmisen tietämyksestä. Lisäksi simuloidut virheet ovat vain käytännössä piirteitä, joilla on vähiten merkitystä ennustuksen tekoon, eli on mahdollista, että tässä aineistossa niistä ei suoranaisesti ollut haittaa, vaan niistä ei vain ollut niin paljoa hyötyä. Tämä ei siis myöskään täydellisesti mallinna ihmisen käyttäytymistä, koska ihminen voi antaa piirteenä jopa haitallista tietoa.

Rajoitteita tulee myös tutkimusaineiston näkökulmasta. Tutkielmassa käytettiin vain yhtä tietoaineistoa, mikä voi tarkoittaa sitä, että aineisto soveltuu ominaisuuksiensa takia erityisen hyvin aktiiviselle oppimiselle ja koneopettamiselle. Lisäksi aineisto on synteettinen, ja optimoidun mallin saavuttama täydellinen sisäinen tarkkuus ja spesifisyys viittaavat hyvin selkeään jakoon luokkien välillä, mikä ei välttämättä ole realistista oikeassa datassa. Aineistossa on vain yhdeksän relevanttia piirrettä joita käytetään mallien kouluttamisessa, minkä takia koneopettamisen mittareiden tulkinnassa x-akseli ei ole yksityiskohtainen. Hyödyntämällä aineistoa, jossa on enemmän piirteitä, olisi mahdollista saada tarkempi kuva oppimiskäyrästä. Tutkielmassa myös käytettiin vain yhtä koneoppimismallia. Käyttämällä useaa eri koneoppimismallia, joiden sisäinen toimintaperiaate on erilainen, olisi mahdollista saada paremmin yleistettävää tietoa aiheesta.

Tutkielman viimeinen rajoite on se, että aktiivisessa oppimisessa simuloitu ihminen toimii oraakkelina, samalla kun koneopettamisessa on annettu mallille virhemarginaali. Tässä tutkielmassa tavoitteena oli tutkia nimenomaan koneopettamista, ja käyttää aktiivista oppimista vertailukohtana, joten tämä ei ole suuri rajoite, kuitenkin selkeämmän kuvan voisi saada poistamalla oraakkeli-olettaman aktiivisesta oppimisesta.

5.3 Jatkotutkimusaiheet

Tutkielman pohjalta pystyy nostamaan useamman aiheen, jota olisi tärkeää tutkia enemmän tulevaisuudessa. Keskeisimpänä jatkotutkimusaiheena on suorittaa samanlainen koe, mutta hyödyntäen oikeita asiantuntijaihmisiä, simulaation sijaan. Oikeita ihmisiä hyödyntäen saisi kokeesta käyttökelpoisempia tuloksia, mutta se myös nostaisi muuttujien määrää erittäin paljon, joten itse toteutus voi olla hyvinkin vaikea.

Yksinkertaisimmillaan tutkimuksen tekoa voisi jatkaa tutkielman pohjalta lisäämällä tietoaineistojen määrää, hyödyntämällä eri koneoppimismalleja ja tuottamalla realistisempi simulaatio opettajan virheistä. Näillä lisäyksillä saisi kokeen tuloksesta tehtyä luotettavamman, sillä näiden muutosten avulla saisi vielä tutkielmassa mahdollisesti olevaa satunnaisuutta poistettua enemmän.

Itse simulaatioon voisi myös tulevaisuudessa tehdä muutoksia, minkä avulla voisi tutkia tarkemmin ihmisen osallistumisen mahdollisuuksia koneoppimisessa. Esimerkiksi olettamaa aktiivisen oppimisen oraakkelista voisi haastaa, ja lisätä myös virheitä aktiivisen oppimisen opettajalle. Myös mielenkiintoinen jatkotutkimusaihe voisi olla yhdistelmä, jossa hyödynnetään sekä aktiivista oppimista että koneopettamista yhdistetyssä mallissa. Yhdistetyllä mallilla voisi koittaa tutkia suorituskyvyn optimointia, ja selvittää mitä mahdollisia etuja kyseisellä menetelmällä voisi saavuttaa.

Lähteet

Aized Amin Soofi & Arshad Awan. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences*, *13*, 459–465. https://doi.org/10.6000/1927-5129.2017.13.76

Amershi, S., Cakmak, M., Knox, W. B., & Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, *35*(4), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

Ayodele, T. O. (2010). Types of Machine Learning Algorithms. Teoksessa *New Advances in Machine Learning*. IntechOpen. https://doi.org/10.5772/9385

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871–885. https://doi.org/10.1002/cpt.1796

Blockeel, H., Devos, L., Frénay, B., Nanfack, G., & Nijssen, S. (2023). Decision trees: From efficient prediction to responsible AI. *Frontiers in Artificial Intelligence*, *6*, 1124553. https://doi.org/10.3389/frai.2023.1124553

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Devidze, R., Mansouri, F., Haug, L., Chen, Y., & Singla, A. (2020). Understanding the Power and Limitations of Teaching with Imperfect Knowledge. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2647–2654. https://doi.org/10.24963/ijcai.2020/367

Dinesh Naveen Kumar Samudrala. (2025). *Cybersecurity Intrusion Detection Dataset*. Noudettu 24. huhtikuuta 2026 osoitteesta https://www.kaggle.com/datasets/dnkumars/cybersecurity-intrusion-detection-dataset

Dwaraka Srihith, P. Vijaya Lakshmi, A. David Donald, T. Aditya Sai Srinivas, & G. Thippanna. (2023). *A Forest of Possibilities: Decision Trees and Beyond*. https://doi.org/10.5281/ZENODO.8372196

Eckhardt, C. M., Madjarova, S. J., Williams, R. J., Ollivier, M., Karlsson, J., Pareek, A., & Nwachukwu, B. U. (2023). Unsupervised machine learning methods and emerging applications in healthcare. *Knee Surgery, Sports Traumatology, Arthroscopy*, *31*(2), 1795. https://doi.org/10.1007/s00167-022-07233-7

El Hassani, M., El Faquih, L., & Machkour, N. (2025). Machine Learning Techniques in Expert Systems: Comparative Insights. *Procedia Computer Science*, *265*, 318–325. https://doi.org/10.1016/j.procs.2025.07.187

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science & Control Engineering*, *2*(1), 602–609. https://doi.org/10.1080/21642583.2014.956265

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285

Holmberg, L., Davidsson, P., & Linde, P. (2020). A Feature Space Focus in Machine Teaching. *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 1–2. https://doi.org/10.1109/PerComWorkshops48775.2020.9156175

Järvinen, P. (2001). *On research methods*. Opinpajan kirja. Noudettu 20. huhtikuuta 2026 osoitteesta https://researchportal.tuni.fi/fi/publications/on-research-methods-3/

Kumar, S., Kaur, P., & Gosain, A. (2022). A Comprehensive Survey on Ensemble Methods. *2022 IEEE 7th International conference for Convergence in Technology (I2CT)*, 1–7. https://doi.org/10.1109/I2CT54291.2022.9825269

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., & Fernández-Leal, Á. (2023). Human-in-the-loop machine learning: A state of the art. *Artificial Intelligence Review*, *56*(4), 3005–3054. https://doi.org/10.1007/s10462-022-10246-w

Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, *13*(1), 911–921. https://doi.org/10.12785/ijcds/130172

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

Opitz, J., & Burst, S. (2021). *Macro F1 and Macro F1* (arXiv:1911.03347). arXiv. https://doi.org/10.48550/arXiv.1911.03347

Prakash, V. J., & Nithya, L. M. (2014). A Survey On Semi-Supervised Learning Techniques. *International Journal of Computer Trends and Technology*, *8*(1), 25–29. https://doi.org/10.14445/22312803/IJCTT-V8P105

Qiang, W., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, 1143–1146. https://doi.org/10.1109/MEC.2011.6025669

Rainio, O., Teuho, J., & Klén, R. (2024). Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, *14*(1), 6086. https://doi.org/10.1038/s41598-024-56706-x

Ramos, G., Meek, C., Simard, P., Suh, J., & Ghorashi, S. (2020). Interactive machine teaching: A human-centered approach to building machine-learned models. *Human–Computer Interaction*, *35*(5–6), 413–451. https://doi.org/10.1080/07370024.2020.1734931

Rincy, T., & Gupta, R. (2020). A Survey on Machine Learning Approaches and Its Techniques: *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 1–6. https://doi.org/10.1109/SCEECS48394.2020.190

Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning*, *2024*, 69–79. https://doi.org/10.58496/BJML/2024/007

Simard, P. Y., Amershi, S., Chickering, D. M., Pelton, A. E., Ghorashi, S., Meek, C., Ramos, G., Suh, J., Verwey, J., Wang, M., & Wernsing, J. (2017). *Machine Teaching: A New*

Paradigm for Building Machine Learning Systems (arXiv:1707.06742). arXiv. https://doi.org/10.48550/arXiv.1707.06742

Sun, S., & Zhou, J. (2015). Gaussian process versus margin sampling active learning. *Neurocomputing*, *167*, 122–131. https://doi.org/10.1016/j.neucom.2015.04.086

Tegen, A., Davidsson, P., & Persson, J. A. (2021). Active Learning and Machine Teaching for Online Learning: A Study of Attention and Labelling Cost. *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1215–1220. https://doi.org/10.1109/ICMLA52953.2021.00197

Trisal, A., & Mandloi, D. (2021). MACHINE LEARNING: AN OVERVIEW. *International Journal of Research -GRANTHAALAYAH*, *9*(7), 343–348. https://doi.org/10.29121/granthaalayah.v9.i7.2021.4120

Viering, T., & Loog, M. (2022). *The Shape of Learning Curves: A Review* (arXiv:2103.10948). arXiv. https://doi.org/10.48550/arXiv.2103.10948

Wall, E., Ghorashi, S., & Ramos, G. (2019). Using Expert Patterns in Assisted Interactive Machine Learning: A Study in Machine Teaching. Teoksessa *Human-Computer Interaction – INTERACT 2019* (Vol. 11748, s. 578–599). Springer International Publishing. https://doi.org/10.1007/978-3-030-29387-1_34

Wang, J., Guo, B., & Chen, L. (2022). *Human-in-the-loop Machine Learning: A Macro-Micro Perspective* (arXiv:2202.10564). arXiv. https://doi.org/10.48550/arXiv.2202.10564

Werner, T., Schmidt-Thieme, L., & Yalavarthi, V. K. (2025). *The Role of Active Learning in Modern Machine Learning* (arXiv:2508.00586). arXiv. https://doi.org/10.48550/arXiv.2508.00586

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, *135*, 364–381. https://doi.org/10.1016/j.future.2022.05.014

Zanzotto, F. M. (2019). Viewpoint: Human-in-the-loop Artificial Intelligence. *Journal of Artificial Intelligence Research*, *64*, 243–252. https://doi.org/10.1613/jair.1.11345

Zhu, X. (2015). Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education. *Proceedings of the AAAI Conference on Artificial Intelligence*, *29*(1). https://doi.org/10.1609/aaai.v29i1.9761

Zhu, X., Singla, A., Zilles, S., & Rafferty, A. N. (2018). *An Overview of Machine Teaching* (arXiv:1801.05927). arXiv. https://doi.org/10.48550/arXiv.1801.05927

Liitteet

Liite 1: Ilmoitus tekoälyavusteisten teknologioiden käytöstä tutkielmassa.

Tämän tutkielman valmistelun aikana kirjoittaja käytti Anthropic Claude -palvelua tekstin jäsentelyyn ja kielenhuoltoon, sekä koodin kirjoittamisen työkaluna. Käytetty generatiivinen tekoälymalli oli Opus 4.7. Työkalun/palvelun käytön jälkeen kirjoittaja tarkisti ja muokkasi sisällön tarpeen mukaan ja ottaa täyden vastuun julkaisun sisällöstä.