Matemaattiset menetelmät yhdenvertaisuuden edistämiseksi koneoppimisen sovelluksissa

Emma-Leena Ahmaoja

Diplomityö
Toukokuu 2024

Tarkastajat:
Prof. Ion Petre
Prof. Vesa Halava

Matematiikan ja tilastotieteen laitos

TURUN YLIOPISTO
Matematiikan ja tilastotieteen laitos

Emma-Leena Ahmaoja: Matemaattiset menetelmät yhdenvertaisuuden edistämiseksi koneoppimisen sovelluksissa
Diplomityö, 43 s.
Matematiikka
Toukokuu 2024

Tekoälyjärjestelmien ja -sovellusten käyttö arkipäivän tilanteissa lisääntyy jatkuvasti, mikä korostaa tarvetta huomioida oikeudenmukaisuus niiden suunnittelussa ja toteutuksessa. Tavoitteena on varmistaa, että nämä järjestelmät eivät aiheuta päätöksissään syrjintää tiettyjä ihmisryhmiä kohtaan.

Tässä tutkielmassa tarkastellaan erilaisia oikeudenmukaisuuden määritelmiä sekä syitä datavinoumien muodostumiseen. Lisäksi esitetään joitain kirjallisuudessa ehdotettuja algoritmisia menetelmiä, joilla oikeudenmukaisuutta voidaan edistää koneoppimisen sovelluksissa. Näitä menetelmiä tarkastellaan koneoppimisprosessin eri vaiheiden näkökulmasta. Esikäsittelyn aikaisista menetelmistä tutkielmaan valikoitui NIFTY-algoritmi, kun taas käsittelyn aikaisten menetelmien osalta tarkastellaan adversatiivisia vinoumien korjaustapoja. Lopuksi jälkikäsittelymenetelminä tutkielmassa esitetään ortogonaaliluokittelija ja monitarkkuuden käsite.

Asiasanat: algoritminen oikeudenmukaisuus, oikeudenmukainen koneoppiminen, reiluusmetriikat.

Sisällys

1 Johdanto

Viimeisen vuosikymmenen aikana tekoäly on ottanut huomattavia askeleita eteenpäin. Tätä äkillistä kehitystä selittävät etenkin digitalisaation myötä suurentuneet datamassat, laskentatehon kehitys sekä edistyneet koneoppimisteknologiat [8]. Algoritmista päätöksentekoa käytetään jo huomattavissa määrin ratkaisemaan haastavia ongelmia yhteiskunnallisesti merkittävillä sektoreilla. Koneoppimista hyödynnetään esimerkiksi rikosten uusimisasteiden tarkastelussa [2], keskustelujen ja asiakastuen automatisoinnissa [3], työhakemusten seulonnassa [9] sekä henkilöasiakkaiden luottoluokituksen arvioimisessa [7]. Yhtä aikaa tekoälyjärjestelmien yleistymisen kanssa on herännyt huoli siitä, että koneoppimisalgoritmit tekevät syrjiviä päätöksiä. Kun algoritmin ennustetarkkuutta painotetaan, on huomioitava, että vahvasti historialliseen dataan perustuvat koneoppimismenetelmät voivat periyttää olemassa olevia vinoumia ennusteisiinsa, mikä puolestaan johtaa syrjinnän kasvuun yhteiskunnassa.

Nykyisessä Suomen yhdenvertaisuuslaissa (1325/2014) on asetettu kolme velvollisuutta: yhdenvertaisuuden edistäminen, syrjinnän ja vastatoimien kielto sekä syrjinnän kohteeksi joutuneen oikeusturvan tehostaminen [25]. Lakia sovelletaan julkisessa ja yksityisessä toiminnassa ja siten yhdenvertaisuuden edistämisvelvoite ulottuu koskemaan myös yleisesti käytössä olevien tekoälyjärjestelmien kehitystä ja ylläpitoa. Jos koneoppimisalgoritmin kouluttamiseen käytetyt tiedot ovat puolueellisia, algoritmi voi oppia nämä vääristymät ja kohdistaa syrjiviä päätöksiä lailla suojattuihin ryhmiin, vaikka järjestelmän suunnittelijat eivät itse olisikaan tehneet syrjiviä päätöksiä. Erityisesti arkaluonteisissa sovelluksissa tällaisilla vääristyneillä päätöksillä voi olla elämää muuttavia vaikutuksia, kun asianomaisten henkilöiden lailliset oikeudet vaarantuvat vääristyneen tuloksen vuoksi [27].

Tämän tutkielman päätavoitteena on perehtyä algoritmista reiluutta käsittelevään kirjallisuuteen ja tarkastella niitä matemaattisia menetelmiä, joilla väestöryhmien yhdenvertaisuutta voidaan algoritmitasolla edistää datan vinoumia korjaamalla.

2 Algoritminen syrjintä

Algoritmisella vinoumalla tarkoitetaan tilaa, jossa tiettyyn ihmisryhmään kohdistuu systemaattista haittaa esimerkiksi epätarkkojen ennusteiden tai negatiivisia seurauksia aiheuttavien päätösten muodossa [15]. Kun kuvatunlaisia vinoumia esiintyy systemaattisesti tekoälyjärjestelmässä, syntyy algoritmista syrjintää. Tällaisessa tapauksessa algoritmi tai tekoälyjärjestelmä aiheuttaa yksilöön tai ihmisryhmään kohdistuvaa syrjintäperusteista epäoikeudenmukaista kohtelua, kuten on käynyt esimerkiksi Yhdysvaltojen tuomioistuinten käyttämän COMPAS-työkalun kanssa.

Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) on ohjelma, joka mittaa henkilön riskiä syyllistyä uuteen rikokseen ja jonka tuloksia käytetään tutkintavankeutta ja vapauttamista koskevan päätöksenteon apuna [2]. Ohjelmistoa koskevassa tutkimuksessa on havaittu sen olevan puolueellinen afroamerikkalaisia kohtaan, sillä ohjelma antaa todennäköisemmin väärän positiivisen afroamerikkalaisilla rikoksentekijöillä kuin valkoihoisilla rikoksentekijöillä [2]. Nämä vääristyneet ennusteet johtuvat datan tai algoritmien piilevistä tai huomiotta jätetyistä vääristymistä.

2.1 Välitön ja välillinen syrjintä

Tekoälysovellus voidaan katsoa *välittömästi syrjiväksi*, jos kielletyn syrjintäperusteen käyttö tai sen ilmentyminen päätöksentekomallissa aiheuttaa syrjintäperusteeseen pohjautuvaa erilaista kohtelua ihmisten välillä [15]. Siinä erottuvat kaksi syrjinnän muotoa: muodollinen eriarvoinen kohtelu samanlaisissa tilanteissa olevia ihmisiä kohtaan sekä aikomus syrjiä [4]. Esimerkiksi rodun tai sukupuolen käyttö muuttujana aiheuttaa sen, että tulosteen arvot perustuvat kiellettyyn informaatioon. Välittömänä syrjintänä voidaan pitää myös sitä, jos mallin tuottama tuloste itsessään sisältää kielletyn syrjintäperusteen tai eksplisiittistä tietoa siitä [15]. Poikkeuksena tähän määrittelyyn tulee huomata yhdenvertaisuuslain 11 § ja 12 § asettamat oikeuttamisperusteet, joiden nojalla yksilön erilainen kohtelu voi kuitenkin olla hyväksyttävää.

Päätöksentekotapaukset, joissa syrjintäperusteita ei automaattisesti sisällytetä tiedonhankintaprosessiin, voivat yhtälailla johtaa järjestelmällisesti epäedullisempiin päätöksiin suojattujen luokkien jäsenten kannalta. Tämä on mahdollista, sillä näennäisesti neutraali järjestelmä voi koulutusprosessin aikana oppia kielletylle syrjintäperusteelle korvannaismuuttujan (engl. *proxy*) [4, 15]. Tässä tapauksessa on kyse *välillisestä syrjinnästä* ja se voi johtua esimerkiksi välinpitämättömyydestä tai heikosta panostuksesta algoritmien pohjadatana käytettäviin tietoihin [13]. Esimerkiksi

yksilön etnisen taustan poistaminen aineistosta ei yksinomaan riitä, sillä esimerkiksi postinumero voi toimia korvaavana indikaattorina etnisyydelle. Tämä tarkoittaa, että postinumeron perusteella voidaan tehdä arvailuja henkilön etnisestä taustasta. Näin ollen kyseisellä alueella asuvat saattavat joutua epäsuoran syrjinnän kohteeksi, vaikka nimenomaan etniseen taustaan liittyvistä tiedoista olisi periaatteen tasolla luovuttu. Tällöin tekoälysovelluksen tuottamilla päätöksillä on suhteettoman epäsuotuisa vaikutus tiettyihin ihmisryhmiin, vaikka noudatettavat periaatteet ja käytänteet ovat näennäisesti neutraaleja [4].

Päätöksentekijät eivät siis aina tarkoituksellisesti aiheuta syrjintää, vaan se voi johtua tiedonlouhinnan tapauksista, joissa suojattu tieto tai sen arvot korreloivat voimakkaasti suojattuun luokkaan kuulumisen kanssa [4]. Syrjintää voi siis ilmetä koneoppimismalleissa, vaikka tarkoitus olisi vain maksimoida tarkkuutta. Valtioneuvoston toteuttamassa kartoituksessa nostetaan etenkin välillinen syrjintä keskeiseksi huolenaiheeksi syrjintäriskien hallinnassa ja ehkäisyssä.

2.2 Erilaisia vinoumia

Tekoälyärjestelmissä vinoumilla viitataan tiettyihin datan ryhmiin, jotka ovat koneoppimismalleissa ylipainotettuja tai -edustettuja, tai muuttujiin, jotka ovat keskeisiä kohteena olevan ilmiön kuvaamiseksi, mutta joita opitut mallit eivät ole asianmukaisesti sisällyttäneet [27]. Jos tietty luokka tai ryhmä on aliedustettuna aineistossa, koulutettu malli saattaa sen yleisestä korkeasta tarkkuudesta huolimatta suoriutua tämän ryhmän kohdalla huonosti [27]. Tämä on yksi syy, miksi algoritminen päätöksenteko nähdään joskus ei-toivotun syrjinnän aiheuttajana ja epäluotettavana [18, 23]. Algoritmeilla ei ole kykyä havaita päätöksenteon koko kontekstia, ja oletetaan, että taustalla oleva tietojen tuottamisprosessi on oikeudenmukainen. On kuitenkin useita COMPAS-ohjelman kaltaisia tapauksia, joissa ohjelmiston koulutusdata on sisältänyt historiallisia, edustuksellisia tai mittauksellisia vinoumia johtaen täten epäoikeudenmukaisiin päätöksiin [18].

On tärkeää huomata, etteivät vinoumat nouse pelkästään käytetystä datasta tai yksittäisistä koulutusprosessin aktiviteeteista, kuten datan käsittelystä tai mallin opettamisesta, vaan ne elävät näiden aktiviteettien yhdistelmien mukana [15]. Vinoumat ovatkin monivaiheisten tuotantoketjujen tulosta ja ne voivat uusiutua, vahvistua tai korjaantua käytetystä mallinnusmenetelmästä ja oppimisalgoritmista riippuen.

Vinoumat ovat yleisiä, ja ne voivat johtua virheellisestä suunnittelusta tai niitä voi ilmetä huollisenkin suunnittelun jälkeen, sillä tekoälysovellusten monimutkaisuus, iteratiivisuus sekä useiden toimijoiden yhdistyminen tekee vinoumien tunnista-

misesta ja ehkäisystä haastavaa [15]. On myös huomattava, etteivät kaikki vinoumat johda syrjintään oikeudellisessa mielessä vaan niitä voidaan rakentaa myös tarkoituksellisesti syrjinnän ehkäisemiseksi. Tällöin voidaan puhua positiivisesta erityiskohtelusta eli toimenpiteistä, joilla parannetaan syrjinnälle alttiiden ryhmien asemaa ja turvataan tosiasiallinen yhdenvertaisuus, tai kohtuullisista mukautuksista, joilla voidaan turvata vammaisen ihmisen yhdenvertaisuutta [15]. Vinoumien syntymekanismit on kuitenkin hyvä ymmärtää mahdollisten syrjintäriskien tunnistamiseksi.

Yksi tapa erilaisten vinoumien luokitteluun on jaotella ne sen mukaan, missä järjestelmän kehitysprosessin vaiheessa ne rakentuvat. Taulukossa 1 on esitetty data-aineistossa esiintyvät vinoumat, jotka kulkeutuvat datan mukana käytettävään algoritmiin, kun taas taulukossa 2 luetellaan ne vinoumat, jotka syntyvät algoritmin käytöstä ja siirtyvät eteenpäin vaikuttaen järjestelmän käyttäjiin. Viimeisenä taulukkoon 3 on koottu käyttäjien itsensä synnyttämiä vinoumia, jotka siirtyvät heistä takaisin data-aineistoon.

Taulukko 1: Datasta algoritmiin -tyypin vinoumat (Ojanen, A. et al. (2022)).

Mittausvinouma	Vinouma, joka juontaa siitä, miten ominaisuuksia tai ilmiöitä raportoidaan, valikoidaan, hyödynnetään, operationalisoidaan tai mitataan.
Puuttuvan muuttujan vinouma	Vinouma, joka seuraa merkityksellisen muuttujan jättämisestä pois mallista.
Edustavuusvinouma	Vinouma, joka juontaa näytteenotosta populaatiosta datankeruun aikana.
Yhdistämisvinouma	Vinouma, joka juontaa siitä, että populaatiota koskevista havainnoista tehdään päätelmiä yksilöistä.
Otantavinouma	Vinouma, joka juontaa epäedustavasta tai eisattumanvaraisesta näytteenotosta (vrt. edustavuusvinouma).
Pitkittäisdatan vinouma	Vinouma, joka voi seurata esimerkiksi heterogeenisten kohorttien analysoinnista poikkileikkausanalyysin menetelmin pitkittäisanalyysin sijaan.

Linkitysvinouma	Vinouma, joka syntyy, kun toimijoiden yhteyksiä, toimintaa ja vuorovaikutusta tarkastelemalla tuotettu tieto toimijaverkoston (esim. sosiaalisen median alustan) ominaisuuksista ei edusta toimijoiden todellista käyttäytymistä tai antaa siitä väärän kuvan.

Taulukko 2: Algoritmista käyttäjään -tyypin vinoumat (Ojanen, A. et al. (2022)).

Algoritmin vinouma	Vinouma, joka syntyy algoritmin käyttämisen tuloksena syötedatan mahdollisista vinoumista riippumatta.
Käyttäjäinteraktiovinouma	Vinouma, jossa ympäristön ja toimijan vuorovaikutuksen syntyviä havaintoja tulkitaan yksinomaan käyttäytymistä edustavana.
Populaarisuusvinouma	Vinouma, joka perustuu siihen, että suosittu sisältö (esim. verkkokaupoissa) saa enemmän näkyvyyttä.
Ilmentyvä vinouma	Vinouma, joka syntyy käyttäjien tai populaation ja käytetyn järjestelmän vuorovaikutuksessa, ja ilmenee esimerkiksi muutoksina populaation käyttäytymisessä tai sosiaalisissa ja kulttuurisissa arvoissa.
Arviointivinouma	Vinouma, joka syntyy koneoppimisalgoritmin avulla tuotettua mallia arvioitaessa esimerkiksi soveltumattomin mittarein tai vertailuarvoin.

Taulukko 3: Käyttäjästä dataan -tyypin vinoumat (Ojanen, A. et al. (2022)).

Historiallinen vinouma	Vinouma, joka seuraa historiallisten ja olemassa olevien yhteiskunnallisten, rakenteellisten ja/tai sosiaalisten erojen mallintamisesta (jopa silloin, kun mallintamisprosessi on täydellisen tarkka).

Populaation vinouma	Vinouma, joka seuraa siitä, kun mallinnettavan kohdepopulaation (esim. sovelluksen käyttäjät) tilastolliset, demografiset tai muut piirteet eroavat populaatiosta yleisesti.
Itsevalikoitumisvinouma	Yhdenlainen edustavuus- tai otantavinouma, joka syntyy, kun otos perustuu yksilöiden itsevalikoitumiseen (esim. Facebook-ryhmään valikoituu homogeeninen joukko yksilöitä).
Sosiaalinen vinouma	Vinouma, joka syntyy, kun yksilön arviointikykyyn tai toimintaan vaikuttavat havainnot muiden toimijoiden toiminnasta (esim. verkkokaupassa yksilön antamaan tuotearvioon voivat vaikuttaa muiden antamat arviot).
Käyttäytymisvinouma	Vinouma, joka seuraa toimijoiden eroavasta käyttäytymisestä eri alustoilla, eri konteksteissa tai eri datajoukoissa.
Ajallinen vinouma	Vinouma, joka seuraa muutoksista toimijoiden käyttäytymisen ajallisessa tarkastelussa.
Sisällöntuottamisvinouma	Vinouma, joka juontaa rakenteellisista, leksikaalisista, semanttisista tai syntaktisista eroista toimijoiden tavoissa tuottaa sisältöä (esim. eri ikäryhmien eroavaisuudet kielenkäytössä).

3 Oikeudenmukaisuus

Päätöksenteon yhteydessä oikeudenmukaisuudella tarkoitetaan sitä, ettei yksilöä tai ryhmää kohtaan ilmene minkäänlaisia ennakkoluuloja tai suosimista, jotka perustuisivat heidän luontaisiin tai hankittuihin ominaisuuksiin [19]. On tärkeää huomata, että eri ryhmillä ei välttämättä ole samoja mahdollisuuksia saavuttaa tavoitteita ja siksi oikeudenmukaisuuden saavuttamiseksi algoritmisessa päätöksenteossa tulisikin varmistaa, että jokaisella ryhmällä on yhtäläiset mahdollisuudet [18].

Tämän ajatuksen muuntaminen toteutettavaksi ja matemaattisesti johdonmukaiseksi määritelmäksi, joka on vielä algoritmin kielellä ymmärrettävissä, on haastava tehtävä. Toisin sanoen, algoritmi ei saa toimia eri tavalla eri väestöryhmille, sillä tiettyyn väestöryhmään kuulumisen ei tule vaikuttaa saatuun tulokseen millään tavalla. Oikeudenmukainen koneoppiminen pyrkii siis tuottamaan sellaisia ennustemalleja, jotka eivät suojattuihin attribuutteihin perustuen aiheuta ennakkoluuloja tai suosimista mitään yksilöä tai ryhmää kohtaan [19].

Oletetaan valvottu oppimisympäristö (engl. *supervised setting*), jossa x on ominaismuuttuja ja y tavoitemuuttuja. Olkoon $(y_1, x_1), \ldots, (y_n, x_n)$ riippumaton ja identtisesti jakautunut opetusnäyte, joka on poimittu tuntemattomasta jakaumasta $\mathbb{P}$. Käytetään nyt empiiristä jakaumaa $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i,y_i}$. Ennusteen laatua mitataan käyttäen tappiofunktiota (engl. *loss function*) $\ell : (y, \hat{y}) \mapsto \ell(y, \hat{y}) \in \mathbb{R}^+$, joka määrittää virheen muuttujan $\hat{y}$ ennustuksessa, kun y on havaittu. Olkoon $\mathcal{F}$ valittujen algoritmien luokka ja $\hat{f}_n$ paras malli, joka voidaan arvioida minimoimalla luokan $\mathcal{F}$, tappiofunktion sekä mahdollisen rangaistuksen yli. Silloin

$$\hat{f}_n \in \arg\min_{f \in \mathcal{F}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \lambda \text{rangaistus}(f) \right\},$$

jossa λ tasapainottaa kummankin termin vaikutusta saadakseen kompromissin algoritmin vinouden ja tehokkuuden välillä [5]. Orakkelisääntö (engl. *oracle rule*) on paras, vielä tuntematon sääntö, jonka voisi rakentaa, jos todellinen jakauma olisi tiedossa [5]. Merkitään

$$f^* \in \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbb{P}} \left\{ \ell(y, f(x)) + \lambda \text{rangaistus}(f) \right\}.$$

Ennusteet annetaan kaavalla $\hat{y} = \hat{f}_n(x)$. Koneoppimisteorian tulokset varmistavat, että asianmukaisesti valitulla sääntöjoukolla $\mathcal{F}$, ylimääräinen riski

$$\mathbb{E}_{\mathbb{P}} \left\{ \ell(y, \hat{f}_n(x)) \right\} - \mathbb{E}_{\mathbb{P}} \left\{ \ell(y, f^*(x)) \right\}$$

on pieni [5]. Täten matemaattiset takuut takaavat, että optimaalinen ennustemalli toistaa oppimisnäytteestä opitut käytökset uusille havainnoille. Se muokkaa todellisuutta oppimissäännön mukaisesti ilman kyseenalaistamista tai muutosta.

3.1 Oikeudenmukaisuuden määritelmä

Erilaiset oikeudenmukaisuuden määritelmät voidaan karkeasti jakaa kolmeen luokkaan: 1) ryhmäoikeudenmukaisuus, joka korostaa, että vähemmistöryhmien tulisi saada samanlaista kohtelua kuin etuoikeutettujen ryhmien, 2) yksilöoikeudenmukaisuus, joka edellyttää, että samankaltaisia yksilöitä tulisi kohdella samankaltaisesti, ja 3) kontrafaktuaalinen oikeudenmukaisuus, joka kuvastaa ajatusta siitä, että päätös yksilöön liittyen on oikeudenmukainen silloin, kun yksilön suojatun attribuutin arvon muuttaminen ei vaikuta päätökseen [1].

Olkoon $(\omega \subset \mathbb{R}^d, \beta, \mathbb{P})$ todennäköisyysavaruus, jossa β on Borelin σ-algebra avaruuden $\mathbb{R}^d$ osajoukoille ja $d \geq 1$. Olkoon $\hat{f}_n$ algoritmi, joka pyrkii ennustamaan muuttujaa $y \in \mathcal{Y} \in \mathbb{R}^d$ havaintojen $x \in \mathcal{X} \subset \mathbb{R}^d$ perusteella. Oikeudenmukaisuus määritellään silloin suhteessa satunnaismuuttujaan $s \in \mathcal{S}$, jota kutsutaan suojatuksi attribuutiksi (engl. *protected attribute*). Suojatut attribuutit pitävät sisällään sellaisia tietoja datasta, joita voidaan sosiokulttuurisesti pitää arkaluonteisina koneoppimisen sovelluksissa [6]. Esimerkkejä ovat mm. sukupuoli, etnisyys, siviilisääty ja ikä. Täydellisen oikeudenmukaisuuden käsite (engl. *full fairness*) edellyttää, että ennuste $\hat{y} = f(x, s)$ on täysin riippumaton suojatusta muuttujasta s [5].

3.2 Reiluusmetriikat

Mahdollisia mallien vinoumia pyritään tunnistamaan reiluusmetriikoiden (engl. *fairness metrics*) avulla, joista suuri osa on kehitetty luokittelu- ja regressioalgoritmien kontekstiin [15]. Näihin lukeutuu useita yksilö- ja ryhmätason matemaattisia mittareita sekä tilastollisia testejä, joilla arvioidaan tekoälyjärjestelmien yhdenvertaisuutta laskentaprosessien muuttujien välisten suhteiden sekä tulosteiden jakaumien perusteella. Lisäksi mittareita voidaan käyttää mallin opetusvaiheessa algoritmien optimointikriteereinä.

Olkoon $y \in \{0, 1\}$ binäärinen ennustemuuttuja ja $s \in \{0, 1\}$ mallin pohjalta saatu binäärinen satunnaismuuttuja. Oletetaan, että

$$s_i = \begin{cases} 1, & \text{kun } s_i \text{ edustaa epäsuotuisaksi oletettua väestöryhmää} \\ 0, & \text{kun } s_i \text{ edustaa perusjoukkoa.} \end{cases}$$

3.2.1 Tilastollinen pariteetti

Tilastollinen pariteetti (engl. *statistical parity*) käsittelee muuttujien y ja s välistä riippumattomuutta, merkitään

$$\hat{y} \perp\!\!\!\perp s.$$

Luokittelija täyttää tämän määritelmän, jos sekä suojattuihin että suojaamattomiin ryhmiin kuuluvilla kohteilla on yhtä suuri todennäköisyys tulla määritetyiksi positiiviseen ennustettuun luokkaan [22]. Esimerkiksi eri ikäryhmien tai sukupuolten välillä positiivisen ennusteen todennäköisyyden tulee olla yhtä suuri:

$$\mathbb{P}(\hat{y} = 1 \mid s = 0) = \mathbb{P}(\hat{y} = 1 \mid s = 1).$$

Tilastollisesta pariteetista käytetään kirjallisuudessa myös termejä demografinen pariteetti (engl. *demographic parity*) ja ryhmäoikeudenmukaisuus (engl. *group fairness*).

3.2.2 Ennustava pariteetti

Ennustava pariteetti (engl. *predictive parity*) tarkastelee positiivisten ennustearvojen tasa-arvoisuutta ryhmien välillä, merkitään

$$y \perp\!\!\!\perp s \mid \hat{y}.$$

Luokittelija täyttää määritelmän, jos sekä suojattuihin että suojaamattomiin ryhmiin kuuluvilla kohteilla on yhtä suuri positiivisen ennustearvon todennäköisyys eli todennäköisyys, että positiiviseen luokkaan ennustetulla kohteella on todellisuudessa positiivinen luokka [22]. Toisin sanoen, esimerkiksi pääsykoevaiheessa eri opiskelijaryhmien valmistumisprosenttien tulee olla yhtäläiset sisään hyväksyttävien opiskelijoiden osalta. Tämä pätee, kun

$$\mathbb{P}(y = 1 \mid \hat{y} = 1, s = 0) = \mathbb{P}(y = 1 \mid \hat{y} = 1, s = 1).$$

3.2.3 Virhetasojen yhtäläisyys

Virhetasojen yhtäläisyys (engl. *equality of odds*) tarkastelee riippumattomuutta suojatun attribuutin s ja tuloksen y välillä, kun otetaan huomioon kohteen todellinen arvo, merkitään

$$\hat{y} \perp\!\!\!\perp s \mid y.$$

Määritelmän mukaan eri väestöryhmillä tulisi olla yhtäläiset virhetasot [22]. Toisin sanoen, todennäköisyys väärälle positiiviselle ja väärälle negatiiviselle tulee olla yhtä suuri eri väestöryhmien välillä:

$$\mathbb{P}(\hat{y} = 1 \mid y = i, s = 0) = \mathbb{P}(\hat{y} = 1 \mid y = i, s = 1), \text{ kun } i = 0, 1.$$

Käytännön esimerkkinä voidaan antaa yksityishenkilön luotonhakutilanne. Jotta määritelmä täyttyy, sekä aidosti korkeaan luottoluokitukseen kuuluvan hakijan todennäköisyys tulla määritetyksi oikein korkeaan luottoluokkaan että matalan luottoluokituksen omaavan hakijan todennäköisyys tulla virheellisesti määritetyksi korkeaan luottoluokkaan, tulee olla yhtä suuret sekä mies- että naispuolisille hakijoille.

3.2.4 Kalibraatio

Kalibraatiossa väestöryhmien väliset tulokset heijastavat populaation todellista todennäköisyysjakaumaa [15]. Mallin todennäköisyystuloksia säädellään varmistaen, että ennustettujen positiivisten tulosten osuus vastaa positiivisten esimerkkien osuutta kaikissa aineiston ryhmissä. Oikeudenmukaisuuden kannalta tämän tulee päteä myös populaation aliryhmiin. Algoritmi $\hat{f}_n$ on *kalibroitu*, kun kaikilla r

$$\mathbb{P}(\hat{y} = 1 \mid r, s = 0) = \mathbb{P}(\hat{y} = 1 \mid r, s = 1).$$

Luotonhakutilanteessa tämä määritelmä tarkoittaa, että millä tahansa tuloksella r, todennäköisyys korkealle luottoluokitukselle tulee olla yhtä suuri sekä mies- että naispuolisille hakijoille [22].

3.2.5 Positiivisen ja negatiivisen luokan tasapaino

Edellä esitellyn virhetasojen yhtäläisyyden yleistykseksi on esitelty positiivisen ja negatiivisen luokan tasapainoa (engl. *balance for positive/negative class*). Matemaattisesti tämä tasapaino ilmaistaan odotusarvojen yhtäsuuruuden kautta [5]:

$$\mathbb{E}(r \mid y = i, s = 0) = \mathbb{E}(r \mid y = i, s = 1), \text{ kun } i \in \{0, 1\}.$$

4 Algoritmiset ratkaisut

Tekoälyn vahvuus on sen kyky tunnistaa säännönmukaisuuksia datan sisällä. Se pystyy tunnistamaan sellaisia ilmiöitä, jotka eivät ole suoraan luettavissa datasta sekä tekemään johtopäätöksiä niiden pohjalta. Tätä vahvuutta voidaan syrjinnän näkökulmasta pitää myös heikkoutena, sillä algoritmien tuottamat tulokset ovat usein vaikeita tai jopa mahdottomia selittää. Kuten todettu, ei voida luottaa, että pelkkä syrjintäperusteisen luokan poistaminen koneoppimismallista ratkaisisi syrjintään liittyvät ongelmat, sillä mallien käyttämät korvannaismuuttujat saattavat epäsuorasti johtaa välilliseen syrjintään [4, 6, 15]. On myös esitetty huoli syrjivien päätöksentekosääntöjen naamioimisesta useiden korvannaismuuttujien taakse, mikä tekee mahdollisen syrjinnän todistamisesta vaikeaa [15].

Luvussa 2.2 tarkasteltiin erilaisia vinoumia ja niiden syntymekanismeja. Kuten todettiin, vinoumia esiintyy alkuperäisissä datajoukoissa, niitä syntyy datankeruun aikana ja ne voivat uusiutua tai vahvistua koneoppimisprosessin aikana [15]. Siksi onkin tärkeää, että yhdenvertaisuuteen kiinnitetään huomiota sovelluksen kehitysprosessin jokaisessa vaiheessa. Seuraavaksi tarkastellaan algoritmisia keinoja, joilla datassa esiintyviä vinoumia voidaan korjata prosessin eri vaiheissa. Näistä käytetään myös nimitystä oikomismenetelmät.

Datan esikäsittely (engl. *pre-processing*) on tekniikka, joka pyrkii tuottamaan alkuperäisdatasta valittua oppimistavoitetta varten hienosäädettyä koulutusdataa, joka voidaan myöhemmin antaa syötteenä koulutusalgoritmille [6, 16, 19]. Tavoitteena on esikäsitellä koulutusdataa, jotta mikä tahansa tällä datalla koulutettu luokittelualgoritmi tuottaisi oikeudenmukaisia lopputuloksia. Näiden menetelmien vahvuus onkin siinä, että niitä voidaan käyttää minkä tahansa luokittelualgoritmin kanssa. Heikkoutena puolestaan on, että ne voivat heikentää tulosten selitettävyyttä ja prosessin lopussa saavutettuun tarkkuuteen liittyy suurta epävarmuutta, sillä menetelmät eivät ole optimaalisesti räätälöityjä käytetylle koneoppimismallille [19].

Käsittelyn aikaiset menetelmät (engl. *in-processing*) pyrkivät poistamaan vinoumat ja saavuttamaan oikeudenmukaisen luokittelun muokkaamalla luokittelijan koulutusmenetelmää [5, 16, 19]. Nämä menetelmät ovat tyypillisesti käytössä valvotuissa oppimisympäristöissä, joissa oppimistavoite on tiedossa. On tärkeää huomioida, että saadut koulutetut mallit takaavat tällöin oikeudenmukaisuuden ainoastaan kyseistä tavoitetta kohtaan.

Samoin kuin esikäsittelyn tekniikkoja, myös jälkikäsittelymenetelmiä (engl. *post-processing*) voidaan käyttää minkä tahansa koneoppimismallin kanssa. Ne pyrkivät parantamaan ennusteen oikeudenmukaisuutta soveltamalla muunnoksia mallin tuloksiin [5, 6, 16]. Jälkikäsittely on yksi joustavimmista lähestymistavoista, sillä se ei

vaadi pääsyä varsinaisiin algoritmeihin tai koneoppimismalleihin. Tämän ansiosta ne ovat sovellettavissa myös silloin, kun koko kehitysputki ei ole näkyvillä. Koska jälkikäsittelymenetelmiä sovelletaan kuitenkin vasta oppimisprosessin myöhäisessä vaiheessa, niillä saadaan yleensä muita menetelmiä huonompia tuloksia [6].

Käsittelyn aikaiset menetelmät voidaan matemaattisesti nähdä reiluina riskin minimointiongelmina (engl. *fair risk minimization problem*), kun taas datan esi- ja jälkikäsittely perustuvat optimaaliseen kuljetukseen (engl. *optimal transport*) [5].

4.1 Datan esikäsittely

Datan esikäsittelyyn liittyvät menetelmät tunnistavat ongelman olevan usein itse datassa, tiettyjen suojattujen attribuuttien ollessa vääristyneitä, syrjiviä ja/tai epätasapainossa [6]. Esikäsittelymenetelmät pyrkivät täten muuttamaan suojattujen attribuuttien otosjakaumia tai suorittamaan erityisiä muunnoksia dataan syrjinnän poistamiseksi ilman oletuksia jatkossa käytettävästä mallinnustekniikasta [6]. Menetelmien tavoitteena on päästä kouluttamaan perinteistä koneoppimismallia ns. puhdistetulla ja korjatulla datasarjalla [6, 16, 19].

Ennen varsinaista mallin opetusvaihetta opetusdatan laatua tulisi arvioida, jotta voidaan varmistaa datan luotettavuus ja käyttökelpoisuus. Erityistä huomiota tulee kiinnittää datan tarkkuuteen ja populaation edustavuuteen, sekä datan validiteettiin ja reliabiliteettiin [15]. On syytä tarkastella, mistä data tulee ja kuka on vastuussa sen keräämisestä. Lisäksi halutaan kiinnittää huomiota siihen, onko dataan sisällytetty informaatio sopivaa ja asianmukaista kehitettävän algoritmin tarkoitukseen. Tätä varten tulee havainnoida myös mahdollisia mittausvirheitä ja vanhentuneita datapisteitä [15]. Nämä vaiheet auttavat hahmottamaan datan alkuperää sekä arvioimaan, onko data luotettavaa ja relevanttia tehtävän kannalta.

Seuraavaksi voidaan tarkastella mitkä populaation ryhmät ovat opetusdatassa edustettuina ja mitkä aliedustettuina. Tähän liittyen voidaan pohtia mm. miltä maantieteellisiltä alueilta ja millä ajanjaksolla dataa on kerätty. Lisäksi voidaan tarkastella datan puutteita, jotta varmistetaan datan edustavan monipuolisesti erilaisia ryhmiä ja tunnistetaan mahdolliset epätäydellisyydet [15].

Edellä kuvatut vaiheet voidaan tiivistää valtioneuvoston kartoituksessa havaittuihin neljään ydinongelmaan. Kiellettyjä syrjintäperusteita sisältävä opetusdata, epäedustava opetusdata, populaation rakenteellisia eroja heijastava opetusdata sekä syrjiviä stereotypioita tai muuten loukkaavaa ja halventavaa sisältöä sisältävä opetusdata sisältävät kaikki merkittäviä riskejä negatiivisten yhdenvertaisuusvaikutusten syntymiselle [15]. Tällaiset data-aineistot ovat ongelmallisia, sillä niiden myötä algoritmi voi oppia toisintamaan olemassaolevia ja/tai historiallisia syrjiviä yhteis-

kunnallisia rakenteita. Yhdeksi ratkaisuksi tähän haasteeseen on esitetty NIFTY-menetelmä, joka voi merkittävästi parantaa esityksen oikeudenmukaisuutta ja stabiiliutta heikentämättä mallin ennustetehokkuutta [1].

4.1.1 NIFTY

Nykypäivän koneoppimismallien on täytettävä korkeat tekniset vaatimukset, kuten korkea ennustetarkkuus ja rajalliset laskentavaatimukset, mutta samalla on varmistettava, etteivät ne syrji väestön alaryhmiä. Graafiset neuroverkot (GNN, engl. *graph neural network*) ovat tällä hetkellä tehokkain ratkaisu teknisten vaatimusten täyttämiseksi, vaikka on osoitettu, että ne perivät ja vahvistavat yhteiskunnan epätasa-arvoa heijastavia datan sisältämiä vinoumia [8]. Tämän luvun teoria perustuu artikkeliin [1].

Syväoppiminen graafeissa perustuu siihen, että malli oppii esittämään graafin solmut piilotettuina esityksinä eli upotuksina (engl. *embedding*) vektoriupotusavaruudessa, missä upotusavaruuden geometria on optimoitu heijastamaan graafin topologiaa sekä solun attribuuttitietoja. Graafiset neuroverkot voidaan usein esittää viestinvälitysverkkoina, jotka on määritelty koulutettavien operaattorien MSG, AGG ja UPD avulla. K-kerroksisessa graafisessa neuroverkossa operaattoreita sovelletaan rekursiivisesti graafiin $\mathcal{G}$ määrittäen, kuinka neuroverkon viestit liikkuvat solmujen välillä (MSG), miten eri naapurien viestit kootaan yhdeksi aggregaattiviestiksi (AGG), ja miten solun edustukset päivitetään aggregoidun viestin perusteella saapumaan viimeisen muunnoskerroksen lopullisiin soluesityksiin (UPD). Tällaisessa verkossa ennusteet saattavat voimakkaasti korreloida suojattujen attribuuttien kanssa aiheuttaen merkittävää syrjivää vinoumaa.

NIFTY (engl. *uNIfying Fairness and stabiliTY*) on menetelmä, jota voidaan käyttää GNN-tyyppistä arkkitehtuuria hyödyntävissä koneoppimismalleissa oikeudenmukaisten ja stabiilien esitysten oppimiseksi. Se hyödyntää keskeistä yhteyttä kontrafaktuaalisen oikeudenmukaisuuden ja mallin stabiiliuden välillä.

Olkoon $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ suuntaamaton graafi, joka koostuu solmujoukosta $\mathcal{V}$ ja kaarijoukosta $\mathcal{E}$. Olkoon $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ kaikkia joukon $\mathcal{V}$ solmuja kuvastava vektorijoukko, missä $\mathbf{x}_v \in \mathbf{X}$ on M-dimensioinen vektori, joka sisältää solmun $v \in \mathcal{V}$ attribuuttiarvot. Merkitään $N = |\mathcal{V}|$ kuvaamaan joukon solmujen määrää graafissa, ja olkoon $\mathbf{A} \in \mathbb{R}^{N \times N}$ graafin vierusmatriisi, jossa

$$\mathbf{A}_{uv} = \begin{cases} 1, & \text{jos solmujen } u \text{ ja } v \text{ välillä on jokin kaari } e \in \mathcal{E} \\ 0, & \text{muutoin.} \end{cases}$$

Lisäksi määritellään binäärinen esiintymävektori $\mathbf{I}_u \in \{0, 1\}^N$, joka tallentaa kaikki

solmuun u kohdistuvat kaaret seuraavasti:

$$\mathbf{I}_{uv} = \begin{cases} 1, & \text{jos solmujen } u \text{ ja } v \text{ välillä on jokin kaari } e \in \mathcal{E} \\ 0, & \text{muutoin.} \end{cases}$$

Olkoon $\mathcal{G}' = (\mathcal{V}, \mathcal{E}', \widetilde{\mathbf{X}})$ puolestaan laajennettu graafi, jossa jokaiselle graafin $\mathcal{G}$ solmulle $u \in \mathcal{V}$ on luotu vastaava solmu muuttamalla lievästi attribuuttien arvoja, niihin liittyviä kaaria ja/tai muuttamalla solmun u suojattua attribuuttia. Merkitään $\widetilde{\mathbf{X}}$ kuvaamaan tätä muokattua vektorijoukkoa ja $\widetilde{\mathbf{A}}$ graafin $\mathcal{G}'$ vierusmatriisia. Lisäksi käytetään seuraavia merkintöjä:

$\mathcal{N}_u$: solmun u välittömien naapurien joukko $\mathcal{N}_u = \{v \in \mathcal{V} \mid A_{uv} = 1\}$

$\mathbf{m}_{uv}^k$: solmujen u ja v välinen viesti kerroksella k, $\mathbf{m}_{uv}^k = \text{MSG}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1})$

$\mathbf{m}_u^k$: joukon $\mathcal{N}_u$ viesteistä koottu aggregaattiviesti, $\mathbf{m}_u^k = \text{AGG}(\mathbf{m}_{uv}^k \mid u \in \mathcal{N}_u)$

$\mathbf{h}_u^k$: kerroksen k tuottama esitys annetulle solmulle u, $\mathbf{h}_u^k = \text{UPD}(\mathbf{m}_u^k, \mathbf{h}_u^{k-1})$

$\mathbf{z}_u$: viimeisen kerroksen tuottama esitys annetulle solmulle u

$\tilde{\mathbf{z}}_u$: viimeisen kerroksen tuottama esitys annetulle solmulle u graafissa $\mathcal{G}'$

f : luokittelija, joka kuvaa solmun u esitystä z_u tietylle luokkanimikkeelle $\hat{y}_u$.

Määritelmä 4.1 ([1], Kontrafaktuaalinen oikeudenmukaisuus)**.** Kooderifunktio (engl. *encoder*) ENC täyttää kontrafaktuaalisen oikeudenmukaisuuden, jos seuraava ehto pätee annetulle solmulle u:

$$\text{ENC}(u) = \text{ENC}(\tilde{u}^s),$$

missä $\tilde{u}^s$ on laajennetun graafin solmu, joka on generoitu muuttamalla solmun u suojatun attribuutin s arvoa, kun kaikki muut arvot pysyvät ennallaan.

Funktiota pidetään Lipschitz-jatkuvuuden käsitteen mukaan stabiilina, jos kohtuullisen pienen häiriön aiheuttaminen mille tahansa annetulle tapaukselle ei radikaalisti muuta funktion tulosta. Graafi-edustusten tapauksessa tämä tarkoittaa, että solmun ominaisuuksiin ja/tai siihen liittyviin kaariin tehtävien pienten muutosten ei tulisi merkittävästi muuttaa saatuja esityksiä.

Määritelmä 4.2 ([1], Stabiili kooderifunktio)**.** Olkoon $\tilde{u}$ solmu laajennetussa graafissa ja L Lipschitzin vakio. Olkoon $\tilde{\mathbf{b}}_u = [\mathbf{x}_{\tilde{u}}; \mathbf{I}_{\tilde{u}}]$ ja $\mathbf{b}_u = [\mathbf{x}_u; \mathbf{I}_u]$ sisältäen tiedon solmujen $\tilde{u}$ ja u attribuuteista ja liittyvistä kaarista. Kooderifunktio ENC on stabiili Lipschitz-jatkuvuuden käsitteen mukaan, jos

$$||\text{ENC}(\tilde{u}) - \text{ENC}(u)||_p \leq L||\tilde{\mathbf{b}}_u - \mathbf{b}_u||_p, \tag{1}$$

missä $||\cdot||_p$ on ns. p-normi, missä $||x||_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ ja $p \geq 1$.

Stabiiliuden vahvistamiseksi muutetaan solmun attribuuttiarvoja poimimalla satunnainen attribuuttimaskivektori $\mathbf{r} \in \{0,1\}^M$ Bernoulli-jakaumasta, ts. $\mathbf{r} \sim \mathcal{B}(p_n)$, missä p_n kuvaa todennäköisyyttä itsenäisesti häiritä jokaista attribuuttia vektorissa $\mathbf{x}_u$, pois lukien sen suojattu attribuutti. Laajennettu attribuuttivektori määritellään siten kaavalla $\tilde{\mathbf{x}}_u = \mathbf{x}_u + \mathbf{r} \odot \delta$, missä $\delta \in \mathbb{R}^M$ on otettu normaalijakaumasta.

Stabiiliutta voidaan yhä vahvistaa graafin rakennetta muuttamalla. Luodaan vierusmatriisi $\tilde{\mathbf{A}} = \mathbf{A} \odot \mathbf{R}_e$, jossa $\mathbf{R}_e \sim \mathcal{B}(1 - p_e)$ on Bernoulli-jakaumasta poimittu satunnainen binäärimaski $\mathbf{R}_e \in \{0,1\}^{N \times N}$ ja p_e ilmaisee todennäköisyyden sille, että valittu kaari jätetään pois.

Oletetaan, että suojattu attribuutti s on binäärimuuttuja eli $s \in \{0,1\}$. Kääntämällä binäärimuuttujan arvo 0:sta 1:een tai päinvastoin saadaan solmu $\tilde{u}^s$, joka merkitsee muutettua suojattua attribuuttia. Tämän muutoksen tarkoitus on vahvistaa funktion oikeudenmukaisuutta.

Kooderifunktio ENC tuottaa laajennetun graafin edustuksia $\tilde{z}_u$ jokaisella iteraatiolla. Generoimalla laajennettuja graafeja, NIFTY voi sisällyttää tarkoituksenmukaista vinoumaa alla olevaan graafiseen neuroverkkoon oppiakseen upotuksia, jotka ovat muuttumattomia kontrafaktuaalisten solmujen yhdistelmälle sekä satunnaisille häiriöille graafin rakenteessa. Täysin yhdistetystä neuroverkon kerroksesta koostuvaa ennustefunktiota $t : \mathbb{R}^d \to \mathbb{R}^d$ käytetään sitten esitysten muuntamiseen ja sovittamiseen toisiinsa.

Oikeudenmukaisuuden ja stabiiliuden vahvistamiseksi esitellään kolmipohjainen tavoitefunktio, joka maksimoi yhteneväisyyttä alkuperäisen graafin ja laajennettujen (kontrafaktuaalisen ja kohinaa sisältävän) esitysten välillä:

$$\mathcal{L}_s = \mathbb{E}_u[\frac{1}{2}(D(t(\mathbf{z}_u), \mathrm{sg}(\tilde{\mathbf{z}}_u)) + D(t(\tilde{\mathbf{z}}_u), \mathrm{sg}(\mathbf{z}_u)))], \tag{2}$$

missä $t(\mathbf{z}_u)$ ja $t(\tilde{\mathbf{z}}_u)$ ovat solmujen u ja $\tilde{u}$ muunnetut esitykset, D on kosinietäisyys, ja stopgrad (sg) estää gradienttien taaksepäin leviämisen. Stopgrad merkitsee, että solmuesityksiä $\tilde{z}_u$ pidetään vakioina, kun toimitaan esityksen $t(z_u)$ kanssa ja päinvastoin.

Lopulta saadaan NIFTY:n kaikenkattava tavoitefunktio:

$$\min_{\theta_{\mathrm{ENC}}, \theta_t, \theta_f} \mathbb{E}_u[(1-\lambda)\mathcal{L}_c] + \lambda \mathcal{L}_s, \tag{3}$$

missä $\{\theta_{\mathrm{ENC}}, \theta_t, \theta_f\}$ merkitsee kooderin ENC, ennustefunktion t ja luokittelijan f koulutettavia parametreja, $\mathcal{L}_c$ on binäärisen ristientropian tappio, ja odotusarvo otetaan graafin $\mathcal{G}$ koulutussolmuista. Regularisointikerroin λ ohjaa kompromissia luokittelutappion $\mathcal{L}_c$ ja kolmipohjaisen tavoitteen $\mathcal{L}_s$ välillä.

NIFTY pyrkii paitsi optimoimaan tavoitefunktiota, myös vahvistamaan oikeudenmukaisuutta ja stabiiliutta suoraan GNN-arkkitehtuurissa muokkaamalla jokaisen kerroksen UPD-operaattoria. Menetelmä parantaa neuroverkon viestinvälitystä suorittamalla kerroskohtaista painonormitusta Lipschitz-vakiota käyttäen.

Tarkastellaan AGG-operaattoria täysin kytkettynä kerroksena, ts. jokainen kerroksen neuroni on yhteydessä kaikkiin edellisen ja seuraavan kerroksen neuroneihin, ja oletetaan UPD-operaattori epälineaariseksi aktivaatiofunktioksi σ. Tällöin viestin välitysvaihe voidaan esittää muodossa

$$\begin{aligned} \mathbf{h}_u^k &= \mathrm{UPD}(\mathrm{AGG}(\mathrm{MSG}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1}) \mid v \in \mathcal{N}_u), \mathbf{h}_u^{k-1}) \\ &= \sigma(\mathbf{W}_a^k \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1}), \end{aligned} \tag{4}$$

missä $\mathbf{W}_n^k$ on solmun u matriisimuotoinen painokerroin kerroksella k, ja $\mathbf{W}_a^k$ kerroksen k itsehuomio-matriisi (engl. *self-attention weight matrix*).

Määritelmän 4.2 perusteella tiedetään, että kun paikallinen verkkoalue ja solmun u attribuuttivektori laajenevat vektorista $\mathbf{b}_u$ vektoriin $\tilde{\mathbf{b}}_u$, Lipschitz-vakio L antaa ylärajan solmun u solmu-upotuksen mahdolliselle muutokselle. Toisin sanoen, Lipschitz-vakio L edustaa pienintä arvoa, jolle yhtälö 1 pätee. Tätä tietoa hyödyntäen NIFTY rajoittaa solmun u upotuksen muutosta Lipschitz-normalisoimalla kooderin painomatriisit jokaisella kerroksella. Tämä rajoittaa alkuperäisten ja häirittyjen solmujen upotusten eroa ja luo yhteyden stabiiliuden ja kontrafaktuaalisen oikeudenmukaisuuden välille, jotta samankaltaisista syötteistä syntyisi samankaltaisia ennusteita. Käytetään Lipschitz-vakiota normalisoimaan itsehuomio-matriisi $\mathbf{W}_a^k$, jotta

$$\tilde{\mathbf{W}}_a^k = \frac{\mathbf{W}_a^k}{\sigma(\mathbf{W}_a^k)}. \tag{5}$$

Saadaan päivitetty UPD-operaattori

$$\mathbf{h}_u^k = \sigma(\tilde{\mathbf{W}}_a^k \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1}).$$

Algoritmiin 1 on kuvattu NIFTY:n koulutusmenetelmä pseudokoodilla.

Lause 4.3 ([1], Stabiili NIFTY)**.** Olkoot $\tilde{u}$ solmu laajennetussa graafissa, $\mathbf{W}_a^k$ solmun u matriisimuotoinen painokerroin kerroksella k, ja $\tilde{\mathbf{b}}_u = [\mathbf{x}_{\tilde{u}}; \mathbf{I}_{\tilde{u}}]$ ja $\mathbf{b}_u = [\mathbf{x}_u; \mathbf{I}_u]$ sisältäen tiedon solmujen $\tilde{u}$ ja u attribuuteista ja liittyvistä kaarista. Jos epälineaarinen aktivaatiofunktio σ on Lipschitz-jatkuva, niin NIFTY-kehyksen oppimat esitykset ovat stabiileja:

$$\|\mathrm{ENC}(\tilde{u}) - \mathrm{ENC}(u)\|_p \leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p \|(\tilde{\mathbf{b}}_u - \mathbf{b}_u)\|_p. \tag{6}$$

Algorithm 1: NIFTY

Data: Graafi $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathbf{X})$; regularisointikerroin λ; suojattu attribuutti s; aineiston koulutuskertojen määrä num_epoch

Result: Optimoidut malliparametrit $\{\theta_{\text{ENC}}, \theta_t, \theta_f\}$; reilut ja stabiilit esitykset z_u, missä $u \in \mathcal{G}$

Alustetaan $k = 1, u = 1, ep = 1$
while $ep \leq num\_epoch$ **do**
  while *kerros* $k \leq K$ **do**
    Lipschitz-normalisoidaan kooderin ENC painomatriisit $\mathbf{W}_a^k$ yhtälön 5 mukaisesti;
    k += 1;
  end
  while *solmu* $u \leq |\mathcal{V}|$ **do**
    Muutetaan attribuutteja ja graafin rakennetta, jotta saadaan $\tilde{u}$;
    Muutetaan suojattua attribuuttia, jotta saadaan $\tilde{u}^s$;
    Koodataan $\mathbf{z}_u = \text{ENC}(u), \tilde{\mathbf{z}}_u = \text{ENC}(\tilde{u}), \tilde{\mathbf{z}}_u^s = \text{ENC}(\tilde{u}^s)$;
    Muunnetaan upotukset: $t(\mathbf{z}_u), t(\tilde{\mathbf{z}}_u), t(\mathbf{z}_{\tilde{u}}^s)$;
    u += 1;
  end
  Lasketaan kolmipohjainen yhtäläisyys yhtälön 2 mukaan;
  Sovelletaan luokittelijaa f seuraavasti: $\hat{y}_u = f(\text{ENC}(u))$;
  Päivitetään $\{\theta_{\text{ENC}}, \theta_t, \theta_f\}$ yhtälön 3 mukaisesti;
  ep += 1;
end

Todistus. Yhtälön 4 perusteella tiedetään, että laajennetun graafin solmulle $\tilde{u}$ tuotettu edustustuloste kerroksella k voidaan merkitä

$$\tilde{\mathbf{h}}_u^k = \sigma(\mathbf{W}_a^k \tilde{\mathbf{h}}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(\tilde{u})} \mathbf{h}_v^{k-1}).$$

Viestinvälityksen jälkeen saatu upotusten välinen erotus kerroksessa k on nyt

$$\tilde{\mathbf{h}}_u^k - \mathbf{h}_u^k = \sigma(\mathbf{W}_a^k \tilde{\mathbf{h}}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(\tilde{u})} \mathbf{h}_v^{k-1}) - \sigma(\mathbf{W}_a^k \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1}).$$

Ottamalla tästä yhtälöstä normi ja olettamalla, että σ on Lipschitz-normalisoitu, saadaan

$$\|\tilde{\mathbf{h}}_u^k - \mathbf{h}_u^k\|_p \leq \|\mathbf{W}_a^k(\tilde{\mathbf{h}}_u^{k-1} - \mathbf{h}_u^{k-1}) + \mathbf{W}_n^k(\sum_{v \in \mathcal{N}(\tilde{u}} \mathbf{h}_v^{k-1} - \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1})\|_p.$$

Koska kaaren p_e pudottamisen todennäköisyys on hyvin pieni, huomataan, että epäyhtälön toinen termi on lähellä nollaa ja se voidaan pudottaa pois. Nyt Cauchy-Schwarzin epäyhtälöä hyödyntämällä saadaan

$$\|\tilde{\mathbf{h}}_u^k - \mathbf{h}_u^k\|_p \leq \|\mathbf{W}_a^k(\tilde{\mathbf{h}}_u^{k-1} - \mathbf{h}_u^{k-1})\| \leq \|\mathbf{W}_a^k\|_p\|(\tilde{\mathbf{b}}_u - \mathbf{b}_u)\|_p.$$

Kooderi ENC on siis pohjimmiltaan kerroksissa $1, \ldots, K$ sovellettavien viestinvälitysfunktioiden peräkkäinen yhdistelmä. Lisäksi kahden Lipschitz-jatkuvan funktion, joilla on Lipschitz-vakiot L_1 ja L_2, yhdistelmä on on uusi Lipschitz-jatkuva funktio, jolla on Lipschitz-vakio $L_1 \times L_2$. Täten saadaan

$$\begin{aligned}\|\mathrm{ENC}(\tilde{u}) - \mathrm{ENC}(u)\|_p = \|\tilde{\mathbf{z}}_u - \mathbf{z}_u\|_p &= \|\tilde{\mathbf{h}}_u^K - \mathbf{h}_u^K\|_p \\ &\leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p\|(\tilde{\mathbf{b}}_u - \mathbf{b}_u)\|_p,\end{aligned}$$

missä K on verkon viimeinen kerros. □

Jos $p = 2$, edellä esitetyssä yhtälössä oleva Lipschitz-vakio on yhtä suuri kuin painomatriisien $\mathbf{W}_a^k$ suurimpien spektrinormien tulo, ja sitä voidaan approksimoida pienellä määrällä potenssimenetelmän (engl. *power method*) iteraatioita. Suoritetaan siis spektraalinen normalisointi kunkin kerroksen painoille ja käytetään normalisoituja painoja $\mathbf{W}_a^k$ kunkin kerroksen UPD-askeleessa.

Lause 4.4 ([1], NIFTY:n kontrafaktuaalinen oikeudenmukaisuus)**.** Olkoon s suojattu attribuutti ja σ epälineaarinen aktivaatiofunktio. Jos σ on Lipschitz-jatkuva, NIFTY-kehyksen oppimien esitysten kontrafaktuaalista epäreiluutta voidaan rajoittaa seuraavasti:

$$\|\mathrm{ENC}(\tilde{u}^s) - \mathrm{ENC}(u)\|_p \leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p,$$

missä $\tilde{u}^s$ on laajennetun graafin solmu, joka luodaan muuttamalla solmun u suojatun attribuutin arvoa pitäen samalla kaikki muu vakiona.

Todistus. Yhtälön

$$\|\mathrm{ENC}(\tilde{u}^s) - \mathrm{ENC}(u)\|_p \leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p\|(\tilde{\mathbf{b}}_u^s - \mathbf{b}_u)\|_p$$

todistus on analoginen edellä esitetyn samanmuotoisen yhtälön 6 kanssa. Suojatun attribuutin arvo on käännetty, mutta muutoin solmu $\tilde{u}^s$ on täsmälleen sama kuin solmu u, joten $\|(\tilde{\mathbf{b}}_u^s - \mathbf{b}_u)\|_p = 1$. Täten

$$\|\mathrm{ENC}(\tilde{u}^s) - \mathrm{ENC}(u)\|_p \leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p\|(\tilde{\mathbf{b}}_u^s - \mathbf{b}_u)\|_p$$

$$\|\mathrm{ENC}(\tilde{u}^s) - \mathrm{ENC}(u)\|_p \leq \prod_{k=1}^{K} \|\mathbf{W}_a^k\|_p.$$ □

4.1.2 NIFTY:n vahvistaminen oikeudenmukaisen attribuuttikäsittelyn ja vääristyneen kaarihylkäyksen avulla

NIFTY:n oikeudenmukaisuuden vahvistamiseksi on esitetty kaksi strategiaa artikkelissa [8], johon tämän luvun teoria perustuu. Ensimmäinen strategia pyrkii oppimaan alkuperäisten solmuattribuuttien oikeudenmukaisen muunnoksen solmujen ominaisuuksia häiritsemällä. Tavoitteena on luoda solmuattribuuttien harhaton esitys, joka säilyttää kuitenkin alkuperäisten attribuuttien piirteet. Toinen strategia puolestaan keskittyy tasapainottamaan graafin homofiilisia ja heterofiilisia yhteyksiä. Tehokas tapa tämän saavuttamiseksi on muuttaa graafin topologiaa pudottamalla graafin kaaria, mikä estää alaryhmien solmujen kardinaalisuuden aiheuttamat vääristymät.

Oikeudenmukaista attribuuttikäsittelyä varten otetaan käyttöön täysin kytketty piilotettu kerroksellinen autokooderifunktio, jolle

$$[\dot{\mathbf{z}}_1, \ldots, \dot{\mathbf{z}}_N]^T = \mathrm{ReLU}([\mathbf{z}_1, \ldots, \mathbf{z}_N]^T \mathbf{W}_e)\mathbf{W}_d,$$

missä $\mathbf{W}_e$ on kooderin ja $\mathbf{W}_d$ on dekooderin paino. Painot koulutetaan artikkelissa [11] esiteltyä ADAM-metodia käyttäen minimoimaan tappiofunktio

$$\lambda L_r + (1 - \lambda) L_f,$$

missä regularisointikerroin λ ohjaa kompromissia rekonstruktion tarkkuuden (L_r) ja piilotetun esityksen oikeudenmukaisuuden ($\mathrm{ReLU}([\mathbf{z}_1, \ldots, \mathbf{z}_N]^T \mathbf{W}_e)$) välillä. Rekonstruktion tarkkuutta mitataan keskimääräisellä neliövirheellä ja esityksen oikeudenmukaisuutta minimoitavaksi halutulla reiluusmetriikalla.

Tilastollisen pariteetin tapauksessa saadaan

$$\left\| \hat{\mathbb{E}}_{v|s_v=0} \mathrm{ReLU}(\mathbf{z}_v^T \mathbf{W}_e) - \hat{\mathbb{E}}_{u|s_u=1} \mathrm{ReLU}(\mathbf{z}_u^T \mathbf{W}_e) \right\|^2,$$

eli solmu, jossa suojatun attribuutun $s = 0$ keskimääräisen edustuksen on oltava samanlainen kuin solmun, jossa $s = 1$. Jotta oikeudenmukainen attribuuttikäsittely ei vaikuttaisi NIFTY:n laskennallisiin vaatimuksiin, autokooderia ei liitetä osaksi laajempaa menettelyä vaan malliparametrit koulutetaan sen suhteen erikseen.

NIFTY:n olennaisena osana on graafin topologian muokkaaminen kaaria poistamalla. Se kuitenkin keskittyy stabiiliuden vahvistamiseen, eikä ota huomioon pudotetun kaaren solmujen suojattuja attribuutteja. Seuraavaksi esiteltävä vääristynyt kaarihylkäys on strategia, joka pyrkii muokkaamaan NIFTY:n perinteistä kaarihylkäystä niin, että graafin topologia ei vinoutuisi kaaria pudotettaessa vaan oikeudenmukaisuus tulisi huomioiduksi tässäkin vaiheessa. Graafin kaarien tarkalla

uudelleenkytkennällä voidaan saavuttaa koulutettu malli, jolla on alkuperäistä paremmat oikeudenmukaisuusominaisuudet ja siksi strategia pakottaa graafille halutun homofilia-asteen. Syntyvän graafin kaarten on siis kuljettava sellaisten solmujen välillä, joilla on samoja suojattujen attribuuttien arvoja.

Homofilian periaatteen mukaan yksilöt ovat todennäköisemmin vuorovaikutuksessa sellaisten yksilöiden kanssa, jotka jakavat samoja suojattuja attribuutteja johtaen epätasa-arvoisiin vaikutuksiin. Vääristynyt reunahylkäys pohjautuu artikkelissa [20] esiteltyyn FairDrop-algoritmiin, mutta ottaa lisäksi huomioon hyperparametrin ρ, joka määrittää sallittujen homofiilisten yhteyksien enimmäissuhteen.

Otetaan huomioon NIFTY:n mukainen vierusmatriisi $\mathbf{A}$ ja laajennetun graafin $\mathcal{G}'$ vierusmatriisi $\tilde{\mathbf{A}}$. Olkoon

$$\mathcal{E}^{\tilde{\mathbf{A}}} = \{(u_i, u_j) : (u_i, u_j) \in \mathcal{E}, \tilde{a}_{i,j} = a_{i,j} = 1\}$$

niiden kaarien joukko, joita NIFTY ei ole pudottanut ja

$$\mathcal{E}^{\neg\tilde{\mathbf{A}}} = \{(u_i, u_j) : (u_i, u_j) \in \mathcal{E}, \tilde{a}_{i,j} = 0, a_{i,j} = 1\}$$

niiden kaarien joukko, jotka NIFTY pudotti. Joukon $\mathcal{E}^{\tilde{\mathbf{A}}}$ perusteella lasketaan homofiilisten kaarien prosenttiosuus. Jos osuus on suurempi kuin ρ, niin algoritmi korvaa joukon $\mathcal{E}^{\tilde{\mathbf{A}}}$ homofiilisiä yhteyksiä joukon $\mathcal{E}^{\neg\tilde{\mathbf{A}}}$ heterofiilisillä yhteyksillä, kunnes haluttu arvo ρ saavutetaan.

4.2 Käsittelyn aikaiset menetelmät

Kuten todettu, syrjiviä vinoumia voi syntyä tai vahvistua myös mallin opetusvaiheessa. Siksi mallin oikeudenmukaisuutta arvioitaessa on huomioitava useita keskeisiä näkökohtia. Ensinnäkin on tärkeää tunnistaa mallinnuksen yhteydessä perustasot mallinnettavalle ilmiölle eri väestöryhmissä [15]. Algoritmien yhdenvertaisuusvaikutusten arvioinnissa tulisi tutkia ja hyödyntää reiluusmetriikoita ja valitut metriikat on olennaista määritellä huolellisesti, kun huomioidaan järjestelmän yhteiskunnallinen käyttökonteksti sekä sovellettava lainsäädäntö [15]. Lisäksi on tärkeää määritellä vertailuluokkien, kuten sukupuolen ja iän ominaisuudet mallin oikeudenmukaisuuden arvioimiseksi [15]. Samalla tulee varmistaa, ettei mallin muuttujina käytetä kiellettyjä syrjintäperusteita ilman oikeudellisesti perusteltua ja hyväksyttävää syytä [15]. Lopuksi on suoritettava intersektionaalinen arviointi, jossa otetaan huomioon eri alaryhmien välinen moniperusteinen syrjintä [15].

Mallin valinnassa on tärkeää pohtia, kuinka suorityskykyä ja reiluutta mitataan ja arvioidaan sekä määritellä niitä koskevat tavoitearvot, kuten mikä on hyväksyttävä kompromissi ennusteiden reiluuden ja tarkkuuden välillä [21]. Mallin toiminta

tulee dokumentoida selkeästi ja lisäksi määrittää mitä mallin tulee ennustaa, luokitella tai suositella, sekä valita sopivat kohdemuuttujat ja luokittelutehtävät [15]. Valitusta tekoälymallista tulee lisäksi tunnistaa ja dokumentoida sen rajoitukset sekä sellaiset piirteet, jotka voivat johtaa syrjiviin tuloksiin [15]. On myös tärkeää pohtia, miten näihin haavoittuvuuksiin voidaan puuttua erityisesti, jos algoritmi oppii jatkuvasti ympäristöstään [15]. Näiden tietojen avulla tutkijat voivat arvioida mallin soveltuvuutta annettuun tehtävään. Koska menetelmillä on erilaisia vaikutuksia tarkkuuteen ja oikeudenmukaisuuteen, tutkijoita suositellaan valitsemaan oikeat menetelmät tutkimustavoitteidensa perusteella [21]. Tavoitteena tulisi olla mahdollisimman yksinkertaisen, aiemmin validoidun ja läpinäkyvän mallin käyttäminen.

Kun kaikki edellä mainitut näkökulmat otetaan mallin kehitysvaiheessa huomioon, mahdolliset syrjivät vinoumat ja syrjintäriskit voidaan tunnistaa ja ehkäistä. Käsittelyn aikaisissa menetelmissä tyypillisiä keinoja oikeudenmukaisuuden lisäämiseksi ovat tappiofunktioon kohdistuvat rangaistukset tai rajoitukset, joilla suojatun ominaisuuden määrittelemien olosuhteiden välistä ennustuskyvyn eroa pyritään kaventamaan [21].

Olkoon $(x_1, s_1, y_1), ..., (x_n, s_n, y_n)$ riippumaton ja identtisesti jakautunut joukko havaintoja, jotka on poimittu tuntemattomasta jakaumasta $\mathbb{P}$. Oletetaan empiiriseksi jakaumaksi $\mathbb{P}_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i,s_i,y_i}$. Lähes oikeudenmukainen malli saadaan minimoimalla empiirinen riski

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i, s_i)),$$

jossa $\ell : (y, \hat{y}) \mapsto \ell(y, \hat{y}) \in \mathbb{R}^+$ on ennusteen laatua mittaava tappiofunktio ja suojatun muuttujan s vaikutus ennusteeseen $\hat{y}$ on kontrolloitu [5]. Optimointiongelman relaksointi mahdollistaa opitun algoritmin oikeudenmukaisuuden tason hallinnan. Tämä voidaan saavuttaa

(i) säätämällä täyden oikeudenmukaisuusehdon raja-arvoa

$$\min_{f \in \mathcal{F}} R_n(f) \tag{7}$$

missä δ on riippuvuuden mitta, $\mathcal{E} \leq 0$ edustaa oikeudenmukaisuuden tasoa ja $\delta(f(x, s), s, y) \leq \mathcal{E}$; tai

(ii) sisällyttämällä riippumattomuus suoraan tavoitteeseen rangaistuksena

$$\min_{f \in \mathcal{F}} R_n(f) + \lambda\delta(f(x, s), s, y), \tag{8}$$

missä $\lambda > 0$ tasapainottaa molempien ehtojen vaikutusta saavuttaakseen kompromissin algoritmin vinouden ja tehokkuuden välillä [5].

Aiemmin luvussa 4.1.1 esitetty NIFTY -algoritmi pyrkii oikaisemaan datan vinoumia luomalla täysin uuden datan esityksen. Käsittelyn aikaiset menetelmät sen sijaan pyrkivät pakottamaan mallin tuottamaan oikeudenmukaisia tuloksia lisäämällä oppimismekanismiin oikeudenmukaisuusrajoituksia tai muokkaamalla tavoitefunktiota. Joissakin menetelmissä rajoitettuja optimointiongelmia muunnetaan Lagrangen kertoimien avulla tai lisäämällä tavoitteeseen rangaistuksia [5]. Toisissa menetelmissä taas pyritään maksimoimaan järjestelmän kykyä ennustaa kohdetta samalla kun minimoidaan järjestelmän kykyä ennustaa suojattua attribuuttia adversatiivisia tekniikkoja käyttäen [5]. Tarkastellaan seuraavaksi menetelmiä, jotka pyrkivät lieventämään vinoumia oppimalla tapauksen uudelleenpainotusfunktion adversatiivista oppimista hyödyntäen.

4.2.1 Adversatiivinen vinoumien korjaus

Adversatiivinen oppiminen on suosittu menetelmä neuroverkkomallien kouluttamisessa [14]. Siinä funktion f tavoite on hämätä vastustusfunktiota g luomalla satunnaisesta kohinasta $z \sim p(z)$ alkuperäistä dataa $\mathcal{X}$ muistuttavaa synteettistä dataa samalla kun vastustusfunktio pyrkii erottamaan todellisen ja synteettisen datan määräämällä $g(f(z)) = 0$ ja $g(x) = 1$ [14]. Tämän jälkeen oppiminen etenee yhteisen tavoitefunktion

$$V(g, f) = \mathbb{E}_{p(x)}[\log(g(x))] + \mathbb{E}_{p(z)}[\log(1 - g(f(z)))]$$

minimointina ja maksimointinta [14].

Tässä luvussa esitellään artikkeliin [26] pohjautuen ensimmäinen adversatiivinen tekniikka reiluuden saavuttamiseksi. Tavoitteena on kouluttaa ennustefunktio f mallintamaan tulostetta mahdollisimman tarkasti noudattaen annettua reiluusmetriikkaa. Oletetaan, että malli koulutetaan muokkaamalla painot w tappion $L_P(\hat{y}, y)$ minimoimiseksi gradienttipohjaista menetelmää käyttäen. Ennusteen tulostekerrosta käytetään sitten vastustusfunktion g syötteenä. Oletetaan, että vastustusfunktiolla on tappiotermi $L_A(\hat{s}, s)$ ja painot u. Tilastollista pariteettia varten vastustusfunktio g pyrkii ennustamaan suojattua attribuuttia s ennusteen $\hat{y}$ pohjalta samalla kun funktion f tavoitteena on estää tämä. Virhetasojen yhtäläisyyttä tutkiessa funktiolle g taas annetaan pääsy todelliseen arvoon y rajoittaen näin ennusteessa $\hat{y}$ oleva suojattu tieto siihen, joka on jo olemassa funktiossa y. Tappiotermin minimoimiseksi painokerroin u päivitetään jokaisella koulutuskerralla gradientin $\nabla_u L_A$ mukaisesti. Painoa w muokataan seuraavasti:

$$\nabla_w L_P - \text{proj}_{\nabla_w L_A} \nabla_w L_P - \alpha \nabla_w L_A, \tag{9}$$

missä α on jokaisella askeleella säädettävissä oleva hyperparametri ja $\text{proj}_v x = 0$, jos $v = 0$.

Lause 4.5 ([26])**.** Olkoon $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^n$ mallin koulutukseen käytettävä datajoukko. Tutkitaan seuraavia ehtoja:

(i) suojattu attribuutti s on diskreetti

(ii) vastustusfunktio g on koulutettu tilastollisen pariteetin mukaan

(iii) vastustusfunktio g on riittävän vahva, jotta se on konvergenssissa oppinut satunnaisfunktion A, joka minimoi ristientropian tappion $\mathbb{E}_{(x,y,s)\sim\mathcal{D}}[-\log P(A(\hat{y}) = s)]$; ts. se saavuttaa optimaalisen suorituskyvyn, jolla muuttujan s voi ennustaa ennusteen $\hat{y}$ perusteella

(iv) Ennustefunktio hämää vastustusfunktion täysin; vastustusfunktio saavuttaa tappion $h(s)$ eli suojatun attribuutin s entropian.

Jos kaikki ehdot täyttyvät, niin $\hat{y} \perp\!\!\!\perp s$ eli ennustefunktio täyttää tilastollisen pariteetin.

Todistus. Jos vastustusfunktio g piirtää satunnaisfunktion $A(\hat{y}) \sim (s \mid \hat{y})$, niin sen tappio on täsmälleen ehdollinen entropia

$$\begin{aligned} h(s \mid \hat{y}) &= \mathbb{E}[-\log P(s \mid \hat{y})] \\ &= \mathbb{E}[-\log P(A(\hat{y}) = s \mid= \hat{y})], \end{aligned}$$

missä odotusarvo otetaan yli otosjakauman $(x, y, s) \sim \mathcal{D}$. Oletetaan sitten, että y on riippuvainen suojatusta attribuutista s. Tällöin $h(s \mid \hat{y}) < h(s)$, joten vastustusfunktion tappio voi saavuttaa pienemmän arvon kuin $h(s)$ mikä on ristiriidassa ehdon (iv) kanssa. □

Lause 4.6 ([26])**.** Jos edellä esitetyt ehdot (ii) - (iv) korvataan vastaavilla virhetasojen yhtäläisyyttä koskevilla oletuksilla eli vastustusfunktiolle tarjotaan molemmat arvot $\hat{y}$ ja y ja että vastustusfunktio ei voi saavuttaa parempaa tappiota kuin $h(s \mid y)$, niin silloin $(\hat{y} \perp\!\!\!\perp s) \mid y$ eli ennustefunktio täyttää virhetasojen yhtäläisyyden vaatimukset.

Todistus. Jos vastustusfunktio g piirtää satunnaisfunktion $A(\hat{y}, y) \sim (s \mid \hat{y}, y)$, niin sen tappio on täsmälleen ehdollinen entropia

$$\begin{aligned} h(s \mid \hat{y}, y) &= \mathbb{E}[-\log P(s \mid \hat{y}, y)] \\ &= \mathbb{E}[-\log P(A(\hat{y}) = s \mid= \hat{y}, y)], \end{aligned}$$

missä odotusarvo otetaan yli otosjakauman $(x, y, s) \sim \mathcal{D}$. Nyt jos $\hat{y} \perp\!\!\!\perp s \mid y$, niin $h(s \mid \hat{y}, y) < h(s \mid y)$. Täten vastustusfunktion tappio voi saavuttaa pienemmän arvon kuin $h(s \mid y)$. □

Lauseet 4.5 ja 4.6 osoittavat, että riittävän voimakas vastustusfunktio, joka on koulutettu riittävän suurella harjoitusjoukolla, kykenee täyttämään tilastollista pariteettia ja virhetasojen yhtäläisyyttä koskevat ennustefunktion rajoitukset, jos vastustus- ja ennustefunktiot saavuttavat konvergenssin.

4.2.2 Tapausten uudelleenpainotus

Tässä luvussa tarkastellaan FAIR-menetelmää artikkelin [17] pohjalta. FAIR ei suorita painotusta esikäsittelynä, vaan painojen estimointi integroidaan koko oppimisprosessiin. Menetelmän pääperiaatteena on painottaa uudelleen kunkin tapauksen log-likelihood oikeudenmukaisuuden ja ennustuskyvyn välisen kompromissin mukaisesti, jotta tulosteelle saadaan oikeudenmukainen ja käyttökelpoinen ennustemalli.

Olkoon $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{s}_i)\}_{i=1}^{N} \sim P(\mathbf{x}, \mathbf{y}, \mathbf{s})$ todellista perusjakaumaa noudattava datajoukko, joka koostuu syöteominaisuuksista $\mathbf{x}$, tulosteista $\mathbf{y}$ ja suojatuista attribuuteista $\mathbf{s}$. FAIR koostuu kolmesta neuroverkosta: painotusverkkoa $f_\theta(\mathbf{x})$ käytetään painojen määrittämiseen kullekin tapaukselle, herkkyysverkkoa $h_\psi(\mathbf{x})$ ennustamaan suojattua attribuuttia ja ennusteverkkoa $g_\phi(\mathbf{x})$ tulosteen ennustamiseen. Painotusverkko $f_\theta(\mathbf{x})$ tuottaa tapauksen x kohdalla kyseisen tapauksen painon $w_\mathbf{x} \in [0, 1]$, kun taas ennuste- ja herkkyysverkko tuottavat vastaavasti ennusteet tulosteiden ja suojattujen ominaisuuksien suhteen. Merkitään näiden verkkojen mallintamia todennäköisyysfunktioita $P_\psi(\mathbf{y} \mid \mathbf{x})$ ja $P_\psi(\mathbf{s} \mid \mathbf{x})$.

Oikeudenmukaisuustavoitteen saavuttamiseksi FAIR painottaa tapausten logaritmisia todennäköisyyksiä antaen matalat painot niille tapauksille, joilla on vahva yhteys suojattuihin attribuutteihin, mutta ei vastaavaa yhteyttä tulosteen kanssa. Vastaavasti suuret painot annetaan niille tapauksille, joilla on vahva yhteys tulosteen kanssa, mutta ei suojattuihin attribuutteihin.

Yksinkertaistetussa viitekehyksessä oletetaan, että jokaiselle tapaukselle $\mathbf{x}$ on määritetty skalaarinen painoarvo $f_\theta(\mathbf{x}) \in [0, 1]$ painotusverkon avulla. Tällöin FAIR-skalaarin mukainen optimointitehtävä on muotoa

$$(\theta^*, \phi^*, \psi^*) = \arg\min_{\theta,\phi} \max_{\psi} \mathbb{E}_{\substack{\mathbf{x},\mathbf{y},\mathbf{s}\sim P(\mathbf{x},\mathbf{y},\mathbf{s}) \\ w\sim f_\theta(\mathbf{x})}} [w \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\psi(\mathbf{y} \mid \mathbf{x}))],$$

missä α säätelee ennusteverkon oikeudenmukaisuuden ja ennustuskyvyn välistä kompromissia.

Viitekehystä voidaan laajentaa käyttämällä skalaarisen painoarvon sijaan satunnaismuuttujia. Tällöin painotusverkon f_θ tuloste mallintaa tapausten todennäköisyysjakaumaa $P(w_{\mathbf{x}} \mid \mathbf{x})$. FAIR-Bernoulli olettaa, että tapausten logaritminen todennäköisyys suhteessa suojattuihin attribuutteihin ja tulosteisiin painotetaan sellaisilla kokonaisluvuilla $w_{\mathbf{x}} \in \{0, 1\}$, että $P_\theta(w_{\mathbf{x}} = 1 \mid \mathbf{x}) = f_\theta(x)$ pätee. Toisin sanoen painojen ehdollinen todennäköisyys noudattaa Bernoulli-jakaumaa $\mathcal{B}(f_\theta(\mathbf{x}))$. FAIR-Bernoullin adversatiivinen tappiofunktio $\mathcal{L}^{\mathcal{B}}_\alpha(\theta, \phi, \psi)$ saadaan kaavasta

$$\mathbb{E}_{\substack{\mathbf{x},\mathbf{y},\mathbf{s}\sim P(\mathbf{x},\mathbf{y},\mathbf{s}) \\ w\sim P_\theta(w|\mathbf{x})}}[w \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x}))].$$

Vastaava adversatiivinen optimointitehtävä on silloin

$$(\theta^*, \phi^*, \psi^*) = \arg\min_{\theta,\phi} \max_{\psi} \mathcal{L}^{\mathcal{B}}_\alpha(\theta, \phi, \psi),$$

missä $\mathcal{B}$ korostaa Bernoulli-oletusta.

Tappion optimoimiseksi on laskettava gradientit θ, ϕ ja ψ suhteen. Kaksi jälkimmäistä voidaan laskea perinteisellä takaisinpäinlaskennalla, mutta gradientin laskeminen θ suhteen on hankalampaa, koska θ määrittelee w-jakauman, jonka yli odotusarvo otetaan. Tästä syystä tappiofunktion gradientti $\nabla_\theta \mathcal{L}_\alpha(\theta, \phi, \psi)$ johdetaan seuraavasti:

$$\begin{aligned}
\nabla_\theta \mathcal{L}_\alpha(\theta, \phi, \psi) &= \nabla_\theta \mathbb{E}_{\substack{\mathbf{x},\mathbf{y},\mathbf{s}\sim P(\mathbf{x},\mathbf{y},\mathbf{s}) \\ w\sim P_\theta(w|\mathbf{x})}}[w \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x}))] \\
&= \mathbb{E}_{\mathbf{y},\mathbf{x},\mathbf{s}} \left[\int_w \nabla_\theta P_\theta(w \mid \mathbf{x}) \cdot w \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x})) dw \right].
\end{aligned}$$

Koska $\nabla_\theta P_\theta(w \mid \mathbf{x}) = P_\theta(w \mid \mathbf{x}) \cdot \frac{\nabla_\theta P_\theta(w|\mathbf{x})}{P_\theta(w|\mathbf{x})} = P_\theta(w \mid \mathbf{x}) \cdot \nabla_\theta \log P_\theta(w \mid \mathbf{x})$, saadaan

$$\nabla_\theta \mathcal{L}_\alpha(\theta, \phi, \psi) = \mathbb{E}_{\substack{\mathbf{x},\mathbf{s},\mathbf{y}\sim P(\mathbf{x},\mathbf{s},\mathbf{y}) \\ w\sim P_\theta(w|\mathbf{x})}}[w \cdot \nabla_\theta \log P_\theta(w \mid \mathbf{x}) \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x}))],$$

mikä mahdollistaa stokastisen gradienttimenetelmän käytön. FAIR-Bernoullin pseudokoodi pisteytysfunktiolla on esitetty Algoritmissa 2.

Olkoon $w \in [0, 1]$. Merkitään kaikki tapauskohtaiset painot sisältävää vektoria $\mathbf{w}$. Lisäksi, jos vektori on vastaavan optimaalisen ratkaisun osa, merkitään $\mathbf{w}^*$. Oletetaan nyt säännelty tappiofunktio

$$\mathcal{L}_\alpha(\mathbf{w}, \phi, \psi) : \sum_{(\mathbf{x},\mathbf{y},\mathbf{s})\in D} w_{\mathbf{x}}[\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x})], \tag{10}$$

jolle $\|\theta\|_2^2 + \|\phi\|_2^2 + \|\psi\|_2^2 < \lambda$.

Algorithm 2: FAIR-Bernoulli

Data: Oppimisasteet $\gamma_\theta, \gamma_\phi, \gamma_\psi$; datajoukko $\mathcal{D}$; hyperparametri α; todennäköisyysmalli P tapausten painoille; iteraatioiden lkm M

Result: Parametrit θ, ϕ, ψ

Asetetaan alkuarvo $i = 0$

Alustetaan θ, ϕ, ψ

while $i \leq M$ **do**

  Poimitaan otos $B \subseteq \mathcal{D}$

  Poimitaan otos $w_{\mathbf{x}} \sim \mathcal{P}(f_\theta(\mathbf{x}))\forall \mathbf{x} \in B$

  Päivitetään $d_\theta \leftarrow$
  $\gamma_\theta \frac{1}{|B|} \sum_{(\mathbf{x},\mathbf{y},\mathbf{s})\in B} [w_{\mathbf{x}} \nabla_\theta \log P_\theta(w_{\mathbf{x}} \mid \mathbf{x}) \cdot (\alpha \log P_\psi(\mathbf{s} \mid \mathbf{x}) - \log P_\phi(\mathbf{y} \mid \mathbf{x}))]$

  Päivitetään $d_\phi \leftarrow \gamma_\phi \nabla_\phi \mathcal{L}_\alpha^{\mathcal{P}}(\theta, \phi, \psi, B)$

  Päivitetään $d_\psi \leftarrow -\gamma_\psi \nabla_\psi \mathcal{L}_\alpha^{\mathcal{P}}(\theta, \phi, \psi, B)$

  Päivitetään $(\theta, \phi, \psi) \leftarrow (\theta, \phi, \psi) - (d_\theta, d_\phi, d_\psi)$

  i += 1;

end

Lause 4.7 ([17])**.** Jokaiselle tapaukselle $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ pätee $w^*_{\mathbf{x}} = 1$ tai $w^*_{\mathbf{x}} = 0$ tai $\alpha \log P_{\psi^*}(\mathbf{s} \mid \mathbf{x}) = \log P_{\phi^*}(\mathbf{y} \mid \mathbf{x})$.

Todistus. Tarkastellaan osittaista derivaattaa optimaalisessa ratkaisussa:

$$\frac{\partial \mathcal{L}_\alpha}{\partial w_{\mathbf{x}}}(\mathbf{w}^*, \phi^*, \psi^*) = \alpha \log P_{\psi^*}(\mathbf{s} \mid \mathbf{x}) - \log P_{\phi^*}(\mathbf{y} \mid \mathbf{x})$$

Jos derivaatta on negatiivinen, on olemassa $d > 0$, jolle pätee

$$\mathcal{L}_\alpha(\mathbf{w}^* + d\mathbf{e}_{\mathbf{x}}, \phi^*, \psi^*) < \mathcal{L}_\alpha(\mathbf{w}^*, \phi^*, \psi^*),$$

missä $\mathbf{e}_{\mathbf{x}} = (0, ..., 1, ..., 0) \in \mathbb{R}^{|D|}$, kun 1 on koordinaatissa, joka vastaa painoa $w_{\mathbf{x}}$. Siten, jos $w^*_{\mathbf{x}} < 1$ pätee, voidaan $w^*_{\mathbf{x}}$ kasvattaa tappion pienentämiseksi ja $(\mathbf{w}^*, \phi^*, \psi^*)$ ei ole optimaalinen ratkaisu, jolloin päädytään ristiriitaan. Siksi $w^*_{\mathbf{x}} = 1$ on oltava voimassa. Jos derivaatta on positiivinen, voidaan vastaavasti osoittaa $w^*_{\mathbf{x}} = 0$. Jos derivaatta on nolla, lause pätee sen kolmannen tapauksen perusteella. □

Lisäksi todistus osoittaa, että optimaalinen paino on 1, jos $\alpha \log P_{\psi^*}(\mathbf{s} \mid \mathbf{x}) < \log P_{\phi^*}(\mathbf{y} \mid \mathbf{x})$ ja 0, jos $\alpha \log P_{\psi^*}(\mathbf{s} \mid \mathbf{x}) > \log P_{\phi^*}(\mathbf{y} \mid \mathbf{x})$. Tämä ominaisuus on oleellinen tulosten tarkastelussa.

Lemma 4.8 ([17])**.** Jos λ on äärellinen, niin on olemassa negatiiviset vakiot $c_\theta, c\prime_\phi, c_\psi$ ja $c\prime_\psi$, joille pätee $c_\phi \leq \log P_\phi(\mathbf{y} \mid \mathbf{x}) \leq c\prime_\phi$ ja $c_\psi \leq \log P_\psi(\mathbf{s} \mid \mathbf{x}) \leq c\prime_\psi$ kaikilla $\mathbf{x}, \mathbf{y}$ ja $\mathbf{s}$ sekä kaikilla ϕ ja ψ, jotka täyttävät ehdon 10.

Todistus. Merkitään symbolilla $\mathcal{B}$ palloa, joka on määritelty kaavalla $\|\theta\|_2^2 + \|\phi\|_2^2 + \|\psi\|_2^2 \leq \lambda$, ja joka kuvaa optimointiongelman mahdollisten ratkaisujen joukkoa. Olkoon $\bar{g}_\phi(\mathbf{x})$ verkko $g_\phi(\mathbf{x})$, joka mallintaa tulostetta y siten, että sigmoidifunktio on poistettu ulostulosta, ja $\bar{h}_\psi(\mathbf{x})$ verkko $h_\psi(\mathbf{x})$, joka mallintaa suojattua attribuuttia s siten, että sigmoidifunktio on poistettu ulostulosta. Koska $\mathcal{B}$ on kompakti joukko ja $\bar{g}_\phi(\mathbf{x})$ ja $\bar{h}_\psi(\mathbf{x})$ ovat jatkuvia funktioita, ne molemmat saavuttavat äärelliset minimi- ja maksimiarvonsa pallossa $\mathcal{B}$. Koska $\log P_\psi(\mathbf{s} \mid \mathbf{x})$ ja $\log P_\phi(\mathbf{y} \mid \mathbf{x})$ ovat jatkuvia funktioita verkoille $\bar{g}_\phi(\mathbf{x})$ ja $\bar{h}_\psi(\mathbf{x})$, jotka kuvaavat funktioiden $\bar{g}_\phi$ ja $\bar{h}_\psi$ arvot koko niiden määrittelyalueella $(-\infty, \infty) \to (-\infty, 0)$, funktiot $\log P_\psi(\mathbf{s} \mid \mathbf{x})$ ja $\log P_\phi(\mathbf{y} \mid \mathbf{x})$ saavuttavat negatiiviset ja äärelliset minimi- ja maksimiarvonsa pallon $\mathcal{B}$ sisällä. Tarvittavat vakiot ovat siis olemassa, joten lemma on todistettu. □

Lemman 4.8 avulla voidaan todistaa seuraava lause, joka määrittelee optimaalisten painojen käyttäytymisen hyperparametin α arvojen ääripäässä.

Lause 4.9 ([17])**.** Jos λ on äärellinen, niin $\mathbf{w}^* = 0$, kun $\alpha = 0$.

Todistus. Lemman 4.8 mukaan $P_\psi(\mathbf{s} \mid \mathbf{x})$ on rajoitettu, joten

$$\begin{aligned}(\mathbf{w}^*, \phi^*, \psi^*) &= \arg\min_{\mathbf{w},\phi} \max_{\psi} \mathcal{L}_\alpha(\mathbf{w}, \phi, \psi) \\ &= \arg\min_{\mathbf{w},\phi} - \sum_{(\mathbf{x},\mathbf{y},\mathbf{s}) \in D} w_{\mathbf{x}} \cdot \log P_\phi(\mathbf{y} \mid \mathbf{x})\end{aligned}$$

on voimassa, kun $\alpha = 0$. Lisäksi Lemman 4.8 mukaan $\log P_\phi(\mathbf{y} \mid \mathbf{x})$ on negatiivinen, joten $-\sum_{(\mathbf{x},\mathbf{y},\mathbf{s}) \in \mathcal{D}} w_{\mathbf{x}} \cdot \log P_\phi(\mathbf{y} \mid \mathbf{x}) \geq 0$. Täten arvosta ϕ huolimatta, sen minimiarvo on 0, kun $\mathbf{w} = 0$. □

Lause 4.10 ([17])**.** Jos λ on äärellinen, niin jokaiselle tapaukselle $(\mathbf{x}, \mathbf{y}, \mathbf{s} \in \mathcal{D})$ on voimassa, että

$$\frac{\partial \mathcal{L}_\alpha}{\partial w_{\mathbf{x}}}(\mathbf{x}_\alpha^*, \phi_\alpha^*, \psi_\alpha^*) \to -\infty,$$

kun $\alpha \to \infty$.

Todistus. Tutkitaan osittaista derivaattaa suhteessa vektoriin $w_{\mathbf{x}}$ optimaalisessa tilanteessa:

$$\frac{\partial \mathcal{L}_\alpha}{\partial w_{\mathbf{x}}}(\mathbf{x}_\alpha^*, \phi_\alpha^*, \psi_\alpha^*) = \alpha \log P_{\psi_\alpha^*}(\mathbf{s} \mid \mathbf{x}) - \log P_{\phi_\alpha^*}(\mathbf{y} \mid \mathbf{x}).$$

Lemman 4.8 mukaan kaikille mahdollisille ψ ja ϕ on olemassa sellaiset vakiot $c\prime_\psi < 0$ ja c_ϕ, että $\log P_\psi(\mathbf{s} \mid \mathbf{x}) \leq c\prime_\psi$ ja $\log P_\phi(\mathbf{y} \mid \mathbf{x} \geq c_\phi$. Siten $\alpha \log P_{\psi_\alpha^*}(\mathbf{s} \mid \mathbf{x}) \to -\infty$, kun $\alpha \to \infty$ ja toinen termi on rajoitettu, joten osittaisderivaatan raja-arvo on $-\infty$. □

Lause 4.11 ([17])**.** Jos λ on äärellinen, niin jokaiselle tapaukselle $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{D}$ pätee, että $w_{\mathbf{x},\alpha}^* \to 1$ kun $\alpha \to \infty$.

Todistus. Lauseen 4.10 mukaan osittaisderivaatan raja-arvo optimaalisessa tilanteessa on negatiivinen, kun $\alpha \to \infty$. Tällöin raja-arvon määritelmän perusteella on olemassa sellainen $\alpha_0 \in \mathbb{R}$, että

$$\frac{\partial \mathcal{L}_\alpha}{\partial w_{\mathbf{x}}}(\mathbf{w}^*_\alpha, \phi^*_\alpha, \psi^*_\alpha) < 0$$

kaikilla $\alpha > \alpha_0$. Koska derivaatta painon $w_{\mathbf{x}}$ suhteen on negatiivinen, pätee jokaiselle tällaiselle parametrille α, että $w^*_{\mathbf{x},\alpha} = 1$. Näin ollen voidaan päätellä, että jokaiselle $\mathcal{E} > 0$ on olemassa sellainen α_0, että $w^*_{\mathbf{x},\alpha} > 1 - \mathcal{E}$ kaikilla $\alpha > \alpha_0$. Näin ollen raja-arvon määritelmästä saadaan, että $w^*_{\mathbf{x},\alpha} \to 1$, kun $\alpha \to \infty$. □

Merkitään $\mathcal{D}_\alpha$ kuvaamaan kaikkien niiden tapausten $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ joukkoa, jotka täyttävät ehdon $\alpha \log P_{\psi^*_\alpha}(\mathbf{s} \mid \mathbf{x}) - \log P_{\phi^*_\alpha}(\mathbf{y} \mid \mathbf{x}) \leq 0$. Muissa tapauksissa pätee $w^*_{\mathbf{x},\alpha} = 0$. Näin ollen optimaaliset ratkaisut ϕ^* ja ψ^* riippuvat joukon $\mathcal{D}_\alpha$ tapauksista. Jakauma $P_\alpha(\mathbf{x}, \mathbf{y}, \mathbf{s})$ voidaan nyt määritellä jakauman $P(\mathbf{x}, \mathbf{y}, \mathbf{s})$ rajoituksena, missä joukon $\mathcal{D}_\alpha$ ulkopuoliset pisteet on asetettu nollaksi ja normalisoitu. Lause 4.12 kuvaa jakauman $P_\alpha(\mathbf{x}, \mathbf{y}, \mathbf{s})$ reiluuden käyttäytymistä TIL-mittarin suhteen.

Lause 4.12 ([17])**.** Jos λ on äärellinen, niin tilastollinen pariteetti $\mathrm{TIL}_\alpha = |P_\alpha(\hat{\mathbf{y}} = 1 \mid \mathbf{s} = 0) - P_\alpha(\hat{\mathbf{y}} = 1 \mid \mathbf{s} = 1)|$ pienenee nollaan, kun α pienenee.

Todistus. Sivuutetaan. Lause on todistettu artikkelissa [17]. □

Lauseet 4.7-4.12 selittävät miten mallin keskeiset elementit ovat vuorovaikutuksessa toisiinsa, kun ne laitetaan liikkeelle hyperparametrin α vaikutuksesta. Käy ilmi, että α voidaan ymmärtää tapauksen ennustettavuuden ja oikeudenmukaisuuden suhdetta koskevana kynnysarvona, jonka perusteella malli päättää, pitäisikö tapaus hylätä vai käyttää oppimiseen. Erityisesti, kun ehto $\frac{\log P_\phi(\mathbf{y}|\mathbf{x})}{\log P_\psi(\mathbf{s}|\mathbf{x})} < \alpha$ täyttyy, tapaus katsotaan intuitiivisesti riittävän oikeudenmukaiseksi suhteessa sen ennustettavuuteen kohdemuuttujan suhteen. Hyperparametrin arvon muuttaminen ohjaa siis sekä mallin oikeudenmukaisuuden että ennustuskyvyn välistä kompromissia ja hyperparametrin pienentäminen vähentää myös sen jakauman epäoikeudenmukaisuutta, josta optimaalinen luokittelija opitaan.

Yksinkertaisemmin sanottuna, jotta suhde olisi matala, tapauksen ennustettavuuden tulisi olla korkea ja sen epäoikeudenmukaisuuden tulisi olla matala. Kun α on nolla, yhtäkään tapausta ei pidetä riittävän oikeudenmukaisena, sillä ei ole mahdollista, että log-likelihood osoittajassa olisi täsmälleen nolla tai log-likelihood nimittäjässä ääretön. Toisessa ääripäässä, kun $\alpha = \infty$, oikeudenmukaisuus jätetään huomiotta ja kaikkia tapauksia käytetään oppimiseen. Välissä olevilla parametrin α arvoilla joitakin tapauksia hylätään ja joitakin käytetään. Tämä keskeinen oivallus

liittyy paitsi hyperparametriin α, myös siihen, miten malli arvioi tapauksen ennustettavia ominaisuuksia ja niiden välistä kompromissia, minkä jälkeen se painottaa niitä eri tavoin.

4.3 Jälkikäsittely

Jälkikäsittelymenetelmät liittyvät nimensä mukaisesti mallin tulosten jälkikäsittelyyn oikeudenmukaisuuden varmistamiseksi. Nämä menetelmät tunnistavat, että koneoppimismallin todellinen tuotos saattaa olla epäoikeudenmukainen yhtä tai useampaa suojattua attribuuttia ja sen alaryhmää kohtaan. Parantaakseen ennusteen oikeudenmukaisuutta, jälkikäsittelymenetelmät pyrkivät soveltamaan sopivia algoritmeja mallin tuloksiin [5, 6, 16]. Tämän tavoitteen saavuttamiseksi erilaiset jälkikäsittelymenetelmät voivat mm. kääntää joitakin luokittelijan päätöksiä, asettaa kynnysarvoja uudelleen tai generoida kokonaan erilliset ryhmäkohtaiset luokittelijat [16]. Koska jälkikäsittely tarvitsee pääsyn ainoastaan ennusteisiin sekä tietoon suojatuista attribuuteista, se on luetelluista luokista ainoa, jonka menetelmät ovat sovellettavissa myöhemmin luvussa 5.3 esiteltävään mustan laatikon ongelmaan [6].

4.3.1 Ortogonaaliluokittelija

Seuraavaksi esitellään yksinkertainen, mutta tehokas jälkikäsittelymenetelmä, joka hyödyntää hyvin koulutettua mallia löytääkseen ortogonaalisen mallin. Tämä löydetty malli kykenee tekemään ennusteita tietyistä ominaisuuksista riippumatta. Oikeudenmukaisuustavoitteen saavuttamiseksi malli voi esimerkiksi luopua kielletyistä syrjintäperusteista. Luvun teoria perustuu artikkeliin [24].

Oletetaan, että meillä on pääluokittelija w_1 ja täydellinen luokittelija w_x, joka ennustaa saman nimikkeen kaiken saatavilla olevan tiedon perusteella. Täydellinen luokittelija voidaan kouluttaa normaalisti ilman rajoituksia, jonka jälkeen ortogonaalinen luokittelija w_2 saadaan vähentämällä täydestä luokittelijasta pääluokittelijan osuus eli $w_2 = w_x \setminus w_1$.

Määritelmä 4.13 ([24], Ortogonaalinen satunnaismuuttuja)**.** Olkoot $x \in \mathcal{X}$ ja $z_1, z_2 \in \mathcal{Z}$ jatkuvia muuttujia. Satunnaismuuttujat z_1 ja z_2 ovat ortogonaalisia jakauman $P(x, y)$ suhteen, jos seuraavat ehdot täyttyvät:

(*i*) on olemassa diffeomorfismi $f : \mathcal{Z}_1 \times \mathcal{Z}_2 \to \mathcal{X}$, jolle $f(z_1, z_2) = x$

(*ii*) muuttujat z_1 ja z_2 ovat tilastollisesti riippumattomia annetun muuttujan y suhteen, ts. $z_1 \perp\!\!\!\perp z_2 \mid y$.

Olkoon $\mathcal{D} = \{(x_t, y_t)\}_{t=1}^n$ jakaumasta $P(x, y)$ poimittu riippumaton ja identtisesti jakautunut datajoukko, joka sisältää syöteominaisuudet $x \in \mathcal{X}$ ja tulosteet $y \in \mathcal{Y}$. Ortogonaaliluokittelijan saamiseksi koulutetaan ensin täydellinen luokittelija $\hat{w}_x$ datajoukon $\mathcal{D}$ perusteella. Olkoot $z_1(x)$ ja $z_2(x)$ keskenään ortogonaalisia satunnaismuuttujia jakauman $P(x, y)$ suhteen. Luokittelijoita kutsutaan seuraavasti: pääluokittelija $w_1(x)_y = P(y \mid z_1(x))$, ortogonaaliluokittelija $w_2(x)_y = P(y \mid z_2(x))$ ja täydellinen luokittelija $w_x(x)_y = P(y \mid x)$.

Olkoot $i, j \in \mathcal{Y}$. Mielivaltaiseen pisteeseen x kohdistettujen tiheyksien $P(x \mid i)$ ja $P(x \mid j)$ suhde voidaan esittää Bayesin optimaalisen luokittelijan $w(x)_i = \Pr(i \mid x)$ avulla muodossa

$$\frac{P(x \mid i)}{P(x \mid j)} = \frac{P(j)w(x)_i}{P(i)w(x)_j}.$$

Vastaavasti

$$\frac{P(z_1(x) \mid i)}{P(z_1(x) \mid j)} = \frac{P(j)w_1(x)_i}{P(i)w_1(x)_j}, \forall i, j.$$

Nämä yhdistämällä saadaan laskettua luokkaehtoisen jakauman $P(z_2(x) \mid y)$ tiheyssuhteet ja edelleen ortogonaaliluokittelija w_2. Koska diffeomorfismi f sallii muuttujanvaihdon, saadaan $P(x \mid i) = P(z_1 \mid i) * P(z_2 \mid i) * J_f(z_1, z_2)$, missä J_f on diffeomorfismikuvauksen Jacobin tilavuus. Nyt

$$\frac{P(z_2 \mid i)}{P(z_2 \mid j)} = \frac{P(z_1 \mid i) * P(z_2 \mid i) * J_f(z_1, z_2)}{P(z_1 \mid j) * (P(z_2 \mid j) * J_f(z_1, z_2)} \Big/ \frac{P(z_1 \mid i)}{P(z_1 \mid j)} = \frac{w_x(x)_i}{w_x(x)_j} \Big/ \frac{w_1(x)_i}{w_1(x)_j}.$$

Yllä olevassa yhtälössä Jacobin tilavuustermit kumoavat toisensa, sillä diffeomorfismi f on yhteinen kaikille luokille. Luokittelijan ortogonalisointi edellyttää, että luokkaehdollisten jakaumien tukipisteet ovat päällekkäiset, jolloin luokittelijat tulosteet $w_x(x)_i$ ja $w_1(x)_i$ pysyvät nollasta poikkeavina kaikilla $x \in \mathcal{X}$ ja $i \in \mathcal{Y}$ arvoilla. Saadaan

$$\Pr(i \mid z_2(x)) = \Pr(i) \frac{w_x(x)_i}{w_1(x)_i} \Big/ \sum_j \left(\Pr(j) \frac{w_x(x)_j}{w_1(x)_j} \right). \tag{11}$$

Olkoon $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^n$ jakaumasta $P(x, y, s)$ poimittu riippumaton ja identtisesti jakautunut datajoukko, joka sisältää myös suojatun attribuutin $s \in \mathcal{S}$. Tavoitteena on oppia luokittelija, joka on tarkka tulosteen y suhteen ja oikeudenmukainen suojatun attribuutin s suhteen. Halutaan siis löytää ortogonaaliluokittelija täysin epäreilulle luokittelijalle w_1, joka käyttää ennustamiseen vain suojattuja attribuutteja. Datajoukosta saadaan suoraan epäreilu luokittelija $w_1(x)_y = p(y \mid s(x))$. Seuraavaksi osoitetaan, että epäoikeudenmukaisen luokittelijan ortogonaaliluokittelija noudattaa virhetasojen yhtäläisyyttä täyttäen täten yhden oikeudenmukaisuuden määritelmistä.

Väite 4.14 ([24])**.** Jos on olemassa ortogonaalinen satunnaismuuttuja jakauman $P(x, y)$ suhteen, niin silloin ortogonaaliluokittelija $w_x \setminus w_1$ täyttää virhetasojen yhtäläisyyden määritelmän.

Todistus. Olkoon $z(x)$ ortogonaalinen satunnaismuuttuja ja $w_x \setminus w_1(x) = Pr[y \mid z(x)]$. Koska $z(x) \perp\!\!\!\perp s \mid y$, niin $w_x \setminus w_1(x) \perp\!\!\!\perp s \mid y$. Näin ollen luokittelijan $w_x \setminus w_1$ ennustus on ehdollisesti riippumaton suojatusta attribuutista s todellisen arvon y suhteen, mikä täyttää virhetasojen yhtäläisyyden määritelmän. □

4.3.2 Moniluokkatarkkuus

Tarkastellaan seuraavaksi menetelmää, joka tarkastelee osapopulaatioita intersektionaalisesta näkökulmasta. Tässä luvussa esitetty teoria perustuu artikkeliin [10].

Oletetaan musta laatikko, jolle on annettu luokittelija f_0 sekä suhteellisen pieni validointijoukko luokiteltuja näytteitä, jotka on poimittu edustavasta jakaumasta $\mathcal{D}$. Jakaumaa $\mathcal{D}$ voidaan pitää todellisena jakaumana, jonka perusteella lopullisen mallin tarkkuutta arvioidaan. Tavoitteena on selvittää luokittelijan f_0 moniluokkatarkkuus (engl. *multiaccuracy*) eli sen oikeudenmukaisuus kaikkien syöteavaruuden osapopulaatioiden $S \subseteq \mathcal{X}$ välillä. Jos tarkastus paljastaa, ettei f_0 täytä moniluokkatarkkuuden vaatimuksia, pyritään luomaan uusi monitarkka luokittelija f ilman negatiivisia vaikutuksia niihin osapopulaatioihin, joissa f_0 oli jo tarkka. Yksinkertaisen algoritmin avulla tunnistetaan ne jakauman $\mathcal{D}$ osajoukot, joissa f_0 tekee eniten virheitä. Tämän tiedon avulla luokittelijaan f_0 kohdistetaan iteratiivinen jälkikäsittelymenetelmä, joka käyttää multiplikatiivisia painoja parantaakseen auditoijan havaitsemia epäoptimaalisia ennusteita, kunnes luokittelija täyttää moniluokkatarkkuuden ehdot.

Määritelmä 4.15 ([10], Moniluokkatarkkuus)**.** Moniluokkatarkkuus edellyttää, että ennusteet ovat puolueettomia jokaisessa tunnistettavassa osapopulaatiossa. Olkoon $\alpha \geq 0$ ja olkoon $\mathcal{C} \subseteq [-1, 1]^{\mathcal{X}}$ funktioiden luokka syötejoukossa. Luokittelija $f : \mathcal{X} \to [0, 1]$ on $(\mathcal{C}, \alpha)$-monitarkka jos kaikille arvoille $c \in \mathcal{C}$ pätee

$$\mathbf{E}_{x \sim \mathcal{D}}[c(x) \cdot (f(x) - y(x))] \leq \alpha.$$

$(\mathcal{C}, \alpha)$-monitarkkuus takaa, että luokittelija f näyttäytyy puolueettomana luokan $\mathcal{C}$ määrittelemän tilastollisten testien luokan mukaan. Käytännössä moniluokkatarkkuus voi rodun ja sukupuolen määrittelemien populaatioiden lisäksi taata tarkkuuden useamman syrjintäperusteen (esim. rotu, sukupuoli, ikä ja uskonnollinen vakaumus) leikkauspisteen määrittelemille osapopulaatioille. Tämä tarjoaa vahvan suojan moniperusteista syrjintää vastaan, sillä moniluokkatarkkuudessa osapopulaatioita tarkastellaan intersektionaalisesta näkökulmasta.

Opetusalgoritmia $\mathcal{A}$ käytetään auditoimaan luokittelijan f moniluokkatarkkuutta. Algoritmi saa pienen otoksen jakaumasta $\mathcal{D}$ ja pyrkii oppimaan funktion h, joka korreloi jäännösfunktion $f - y$ kanssa.

Määritelmä 4.16 ([10], Moniluokkatarkkuuden auditointi)**.** Olkoon $\alpha > 0, m \in \mathbb{N}$ ja $\mathcal{A} : (\mathcal{X} \times [-1, 1])^m \to [-1, 1]^{\mathcal{X}}$ opetusalgoritmi. Oletetaan, että $D \sim \mathcal{D}^m$ on joukko riippumattomia otoksia. Luokittelija $f : \mathcal{X} \to [0, 1]$ läpäisee $(\mathcal{A}, \alpha)$-monitarkkuus auditoinnin, jos

$$\mathbf{E}_{x \sim \mathcal{D}} \left[h(x) \cdot (f(x) - y(x)) \right] \leq \alpha,$$

kun $h = \mathcal{A}(D; f - y)$.

Esitetään seuraavaksi monitarkka BOOST-algoritmi, jolla esikoulutettu malli voidaan jälkikäsitellä moniluokkatarkkuuden saavuttamiseksi. Algoritmille annetaan mustan laatikon mukainen käyttöoikeus alkuperäiseen luokittelijaan $f_0 : \mathcal{X} \to [0, 1]$ ja opetusalgoritmiin $\mathcal{A} : (\mathcal{X} \times [-1, 1])^m \to [-1, 1]^{\mathcal{X}}$. Mallin käsittely aloitetaan jakamalla syöttöavaruus $\mathcal{X}$ alkuperäisen luokittelijan f_0 perusteella kahteen osaan: $\mathcal{X}_0 = \left\{ x \in \mathcal{X} : f_0(x) \leq \frac{1}{2} \right\}$ ja $\mathcal{X}_1 = \left\{ x \in \mathcal{X} : f_0(x) > \frac{1}{2} \right\}$. Syöttöavaruuden kahtiajako varmistaa, että algoritmin tulostama luokittelija f säilyttää alkuperäisen luokittelijan f_0 tarkkuuden.

Opetusalgoritmia $\mathcal{A}$ käytetään nyt iteratiivisesti etsimään syöttöavaruudesta funktioita, jotka korreloivat merkittävästi jäännösennusteen $f - y$ kanssa. Jos algoritmi tunnistaa merkittävän osapopulaation, jolle nykyinen luokittelija on epätarkka, se palauttaa funktion $h : \mathcal{X} \to [-1, 1]$ ja päivittää ennusteet multiplikatiivisesti sen mukaan. Jokaisella iteraatiolla käytetään tuoretta otosta $D_t \sim \mathcal{D}^m$, jotta taataan funktion h yleistettävyys.

Algoritmi pysähtyy iteraatiokertojen täyttyessä tai sen tuottaessa luokittelijan $f : \mathcal{X} \to [0, 1]$, joka läpäisee $(\mathcal{A}, \alpha)$-monitarkkuuden auditoinnin millä tahansa tarkkuusparametrilla $\alpha > 0$. Monitarkka BOOST-algoritmin pseudokoodi on esitetty Algoritmissa 3.

Algorithm 3: Monitarkka BOOST

Data: Alkuperäinen luokittelija $f_0 : \mathcal{X} \to [0, 1]$; auditoija $\mathcal{A} : (\mathcal{X} \times [-1, 1])^m \to [-1, 1]^{\mathcal{X}}$; tarkkuusparametri $\alpha > 0$; validointidata $D = D_0, \ldots, D_T \sim \mathcal{D}^m$; iteraatioiden lkm T

Result: Monitarkka luokittelija f_t

Ositetaan $\mathcal{X}_0 \leftarrow \left\{x \in \mathcal{X} : f_0(x) \leq \frac{1}{2}\right\}$

Ositetaan $\mathcal{X}_1 \leftarrow \left\{x \in \mathcal{X} : f_0(x) > \frac{1}{2}\right\}$

Asetetaan $\mathcal{S} \leftarrow \{\mathcal{X}, \mathcal{X}_0, \mathcal{X}_1\}$

Alustetaan $t = 0$

while $t \leq T$ **do**

  Kun $S \in \mathcal{S}$, asetetaan $h_{t,S} \leftarrow \mathcal{A}(D_t; (f_t - y)_S)$

  Asetetaan $S^* \leftarrow \arg\max_{S \in \mathcal{S}} \mathbf{E}_{x \sim D_t} \left[h_{t,S}(x) \cdot (f_t(x) - y(x))\right]$

  if $\mathbf{E}_{x \sim D_t} \left[h_{t,S^*}(x) \cdot (f_t(x) - y(x))\right] \leq \alpha$ **then**

    Palauta f_t

  end

  Päivitetään $f_{t+1}(x) = e^{-\eta h_{t,S^*}} \cdot f_t(x) \forall x \in S^*$

  $t \mathrel{+}= 1$;

end

5 Haasteet

Aiemmissa luvuissa tarkasteltiin erilaisia tapoja, joilla koneoppimismallien oikeudenmukaisuutta voidaan pyrkiä parantamaan. Seuraavaksi käsitellään haasteita, joita tällaisten oikeudenmukaisuustoimenpiteiden yhteydessä usein kohdataan.

5.1 Oikeudenmukaisuuden mahdottomuusteoria

Intuitiivisesti ajateltuna luvussa 3.2 esitetyt reiluusmetriikat vaikuttaisivat pyrkivän samaan yhteiseen tavoitteeseen eli korkeaan ennustetarkkuuteen riippumatta siitä, mihin aliryhmään kohdehenkilö kuuluu. Sen perusteella voisi luulla, että toistensa muunnelmilta vaikuttavat määritelmät olisi helppo saavuttaa yhtäaikaisesti. Kirjallisuudessa on kuitenkin huomattu erilaisten reiluusmetriikoiden johtaneen niin kutsuttuihin oikeudenmukaisuuden mahdottomuusteorioihin (engl. *impossibility theorems of fairness*) [5, 12]. Tämä tarkoittaa, että yleisesti ottaen useat reiluusmetriikat ovat keskenään yhteensopimattomia ja ne voidaan täyttää samanaikaisesti vain tietyissä erittäin rajoitetuissa tapauksissa.

Ehdollisen todennäköisyyden määritelmästä $\mathcal{L}(y, \hat{y} \mid s)$ voidaan johtaa kolme keskeistä oikeudenmukaisuusmittareihin viittaavaa todennäköisyysjakaumaa [5]. Saadaan

$$\begin{aligned}\mathcal{L}(y, \hat{y} \mid s) &= \mathcal{L}(y \mid \hat{y}, s) \times \mathcal{L}(\hat{y} \mid s) \\ &= \mathcal{L}(\hat{y} \mid y, s) \times \mathcal{L}(y \mid s),\end{aligned}$$

jossa $\mathcal{L}(y \mid \hat{y}, s)$ viittaa ennustavaan pariteettiin, $\mathcal{L}(\hat{y} \mid s)$ tilastolliseen pariteettiin, $\mathcal{L}(\hat{y} \mid y, s)$ virhetasojen yhtäläisyyteen ja $\mathcal{L}(y \mid s)$ todellisen kohteen jakaumaan kussakin ryhmässä.

Tutkitaan seuraavaksi kolmea esitystä, jotka heijastavat erilaisia reiluuden käsitteitä, kun $s \in \{0, 1\}$. Merkitään

OP_s: ryhmäkohtaiset oikeat positiiviset $\mathbb{P}(\hat{y} = 1 \mid y = 1, s)$

VP_s: ryhmäkohtaiset väärät positiiviset $\mathbb{P}(\hat{y} = 1 \mid y = 0, s)$

PE_s: ryhmäkohtaiset positiiviset ennustearvot $\mathbb{P}(y = 1 \mid \hat{y} = 1, s)$.

Tarkastellaan, voiko ennustemuuttuja täyttää samanaikaisesti virhetasojen yhtäläisyyden sekä tilastollisen pariteetin määritelmän.

Väite 5.1. Jos s on riippuvainen muuttujasta y ja $\hat{y}$ on riippuvainen muuttujasta y, niin virhetasojen yhtäläisyys tai tilastollinen pariteetti toteutuu, mutta ei molemmat [5].

Todistus. Huomataan, että jos $s \perp\!\!\!\perp \hat{y}$ ja $\hat{y} \perp\!\!\!\perp y$, niin joko $s \perp\!\!\!\perp y$ tai $\hat{y} \perp\!\!\!\perp y$. □

Tarkastellaan sitten, voiko ennustemuuttuja täyttää samanaikaisesti sekä tilastollisen pariteetin että ennustavan pariteetin määritelmän.

Väite 5.2. Jos s on riippuvainen muuttujasta y, niin tilastollinen pariteetti tai ennustava pariteetti toteutuu, mutta ei molemmat [5].

Todistus. Riittää huomata, että jos $s \not\perp\!\!\!\perp y$ ja $s \perp\!\!\!\perp y \mid \hat{y}$, niin $s \not\perp\!\!\!\perp \hat{y}$. □

Tarkastellaan vielä, voiko ennustemuuttuja täyttää samanaikaisesti ennustavan pariteetin ja virhetasojen yhtäläisyyden määritelmän.

Väite 5.3. Jos s on riippuvainen muuttujasta y, niin ennustava pariteetti tai virhetasojen yhtäläisyys toteutuu, mutta ei molemmat [5].

Todistus. Koska $s \perp\!\!\!\perp \hat{y} \mid y$ ja $s \perp\!\!\!\perp y \mid \hat{y}$, niin $s \perp\!\!\!\perp (\hat{y}, y)$. Tästä seuraa, että $s \perp\!\!\!\perp y$. □

Huomataan, ettei reiluusmetriikkojen yhtäaikainen käyttö ole triviaalia. Kirjallisuudessa ei myöskään vallitse yksimielistä näkemystä siitä, mikä oikeudenmukaisuuden määritelmä tulisi asettaa etusijalle. On myös korostettu, etteivät matemaattiset määritelmät aina vastaa sosiaalisia, taloudellisia tai oikeudellisia näkemyksiä oikeudenmukaisuudesta [6]. Lisäksi on huomattu, että esimerkiksi ryhmien välistä oikeudenmukaisuutta parannettaessa ryhmien sisäiset ongelmat vahvistuvat [6]. On myös esitetty huoli siitä, että algoritmien käyttö päätöksenteossa luo uusia syrjinnän muotoja, jolloin epäsuotuisat vaikutukset voivat kohdistua sellaisiin ryhmiin, joita nykyinen yhdenvertaisuuslaki ei edes tiedosta eikä siten myöskään suojele [15]. Tämä tekee oikeudenmukaisuuden tavoittelusta koneoppimisessa haastavaa, sillä yhden vinouman korjaaminen saattaa hyvinkin johtaa toisen vinouman esiintymiseen.

Ratkaisuksi ongelmaan on ehdotettu keinoja reiluusmetriikkojen yhdistelmien tarkasteluun. Kattavaa meta-arviointia ja monialaista tutkimusta mittareiden kehittämisestä, validoinnista ja kategorisoinnista pidetään merkityksellisenä apuna tutkijoille ja alan ammattilaisille vaihtoehtoisten mittareiden valintaan [6, 13]. Tämän lisäksi empiirisen näytön tukemat työkalupakit ja viitekehykset reiluusmetriikkojen vertailuun ja valintaan nähdään tarpeellisina käytännön toimijoille ja poliittisille päätöksentekijöille [6, 13].

5.2 Suorituskyvyn ja tarkkuuden heikkeneminen

Oikeudenmukaisuuskriteerien liittäminen koneoppimismallin koulutusprosessiin ei ole yksiselitteistä myöskään mallin suorituskyvyn kannalta. Mallia koulutettaessa

on korostettava joko oikeudenmukaisuutta tai mallin suorituskykyä, sillä toisen parantaminen useimmiten heikentää toista. Intuitiivisesti voidaan huomata, että luokittelija, joka ei ota huomioon oikeudenmukaisuutta vaan maksimoi tarkkuuden saavuttaa paremman tarkkuuden kuin vastaava luokittelija, johon lisätään oikeudenmukaisuusrajoituksia [16]. On kuitenkin tärkeää tiedostaa, että mallin korkea tarkkuus on saattanut aiemmin perustua jossain määrin syrjintään. Kun tätä syrjintää pyritään vähentämään, mallin tarkkuuden lasku on väistämätöntä. [6].

Esimerkiksi terveydenhuollossa ja rikosoikeudessa epätarkoilla päätöksillä voi kuitenkin olla vakaviakin vaikutuksia yksilöihin ja yhteiskuntaan, joten suorityskyvyn heikentyminen ei lähtökohtaisesti ole toivottavaa. Siksi täydellisen oikeudenmukaisuuden tavoittelun ei välttämättä pidä edes olla tavoitteena, vaan tärkeämpää on löytää tasapaino oikeudenmukaisuuden ja tarkkuuden välillä lieventämällä oikeudenmukaisuuden käsitettä [5].

5.3 Mustan laatikon ongelma

Tekoälysovellusten läpinäkymättömyys on keskeinen huolenaihe alan kirjallisuudessa. Suuret datamassat, lukuisat parametrit, monimutkaiset tilastolliset mallit sekä tiedonkäsittelyn kompleksisuus vaikeuttavat sekä mallin toiminnan ymmärtämistä että tulosteiden jäljittämistä ja niiden tulkintaa [15, 27]. Tätä ongelmaa nimitetään *mustan laatikon ongelmaksi*.

Läpinäkyvyys on on avainasemassa vastuuvelvollisuuden helpottamisessa ja sitä voidaan arvioida sekä koko mallin että yksittäisten komponenttien, kuten parametrien tai koulutusalgoritmien tasolla [13]. Malleille tulisikin olla ominaista niiden selkeys ja alhainen laskennallinen monimutkaisuus, mutta toisaalta läpinäkyvyyttä voidaan tarkastella myös tarjoamalla intuitiivinen selitys kaikille mallin osille [13]. Vaikka tulkittavissa olevien syväoppimismallien kehittämisessä on tapahtunut valtavaa edistystä, useimmat olemassa olevat tutkimukset keskittyvät käyttämään herkkyysanalyysiä osoittaakseen korrelaation mallin syötteiden ja ennusteiden välillä [27]. Tulkittavuuden osalta kvantitatiivinen kehys kuitenkin puuttuu, eikä sille ole myöskään keksitty yleisesti hyväksyttyä määritelmää [27]. Lisäksi läpinäkyvyyden puutetta algoritmeissa ja malleissa voidaan itsessään pitää riskinä syrjiville vaikutuksille ja se voi jo sellaisenaan johtaa päätelmään syrjinnästä [15]. Yritysten voi sen takia olla vaikeaa tai jopa mahdotonta tarjota objektiivista oikeutusta päätöksilleen, kun taas päätöksen kohteena olevan henkilön mahdollisuudet riitauttaa itseä koskevat päätökset voivat rajoittua [15].

Algoritmisen päätöksenteon läpinäkymättömyyttä voidaan pitää tarkoituksellisena, kun algoritmin keksijät pykivät suojelemaan immateriaalioikeuksiaan [13].

Tämän tyyppisen läpinäkymättömyyden vähentämiseksi esitetään EU:n uuden tietosuoja-asetuksen (GDPR) kaltaista lainsäädäntöä, joka määrittelee henkilölle oikeuden "selitykseen" eli tietoon siitä, kuinka henkilötietoja käsitellään [13].

Toisena ratkaisuna mustan laatikon ongelmaan on esitetty sovellusten ymmärrettävyyttä parantavia *selitysmenetelmiä*, joita pidetään myös hyvien käytänteiden mukaisina [15]. Ne eivät suoraan vaikuta mallin ennusteisiin, mutta voivat auttaa mahdollisten vinoumien ja virheiden tunnistamisessa ja korjauksessa. Selitysmenetelmien avulla on mahdollista parantaa tiedonsaantia päätöksenteon perusteista sekä helpottaa mallien analysointia ja sen myötä arvioida mahdollisen syrjinnän ilmentymistä järjestelmän tulosteissa [15]. Jos selitysmenetelmillä ei saada ratkaisua tulkittavuusongelmaan, tulee läpinäkymättömyyden torjumiseksi edellyttää vaihtoehtoisten helpommin ymmärrettävien koneoppimismallien käyttöä, vaikka niiden tarkkuus saattaa olla alhaisempi kuin mustan laatikon mallien [13].

Niin kutsutun "algoritmisen lukutaidottomuuden" ongelmaa voitaisiin helpottaa koulutusohjelmia tehostamalla sekä antamalla riippumattomille asiantuntijoille mahdollisuus tarjota ohjausta niille, joihin algoritminen päätöksenteko vaikuttaa, sillä suurimmalla osalla ihmisistä ei ole teknisiä taitoja ymmärtää algoritmeja tai koneoppimismallien rakenteita [13].

5.4 Teknologian ulkopuoliset haasteet

Tekoälysovellusten tekniset riskit ilmenevät usein vasta järjestelmän käyttöönoton jälkeen. Monet sosiotekniset tekijät, kuten käyttäjien arviointikyky, sovelluksen käyttökonteksti ja -skaala sekä kohdepopulaatio, vaikuttavat syrjintäriskien realisoitumiseen [15]. On myös todettu, että mikään yksittäinen menetelmä ei ole muita parempi kaikissa tapauksissa vaan tulokset riippuvat valituista reiluusmetriikoista, tietokokonaisuudesta ja muutoksista harjoittelu- ja testidatassa [16]. Lisäksi olemassa oleva algoritmista oikeudenmukaisuutta käsittelevä kirjallisuus ja tutkimustyö nojautuu voimakkaasti vertailututkimukset mahdollistaviin yleisiin ja helposti sovellettavissa oleviin tietoaineistoihin. Tähän verrattua oikeiden mallien kouluttaminen tuo mukanaan lisähaasteita, kun käytettävissä oleva todellinen data edustaa aiempia päätöksiä ja niihin liittyvät vinoumat vahvistuvat, tai kun koulutus- ja käyttöönottovaiheessa käytettyjen aineistojen jakaumat eroavat merkittävästi toisistaan [6]. Tämän takia olosuhteet ja syrjinnän vaikutusten arviointi tulisi aina tulkita tapauskohtaisesti riskien käsitteellistämiseksi. Tähän vaaditaan tiivistä yhteistyötä ja inhimillistä vuorovaikutusta alan asiantuntijoiden kesken [27].

Lisäksi tekoälyavusteisessa päätöksenteossa käyttäjän, niin kutsutun päätöksentekijän, rooli tulosten tulkinnassa on merkittävä. Käyttäjien on havaittu olevan tai-

puvaisia luottamaan sokeasti järjestelmien automaattisesti tuottamiin ennusteisiin, sillä niitä pidetään objektiivisina ja neutraaleina [15]. Tätä nimitetään *automaatioharhaksi*. Käyttäjän kognitiot ja tulkinnat ovatkin siten avainasemassa syrjivien vaikutusten realisoitumisessa. Esimerkiksi käyttäjän järjestelmäosaaminen, hänen tulkintansa algoritmin tulosteesta sekä järjestelmän käyttöliittymäominaisuudet vaikuttavat syrjintäriskiin merkittävästi [15]. Mikäli järjestelmässä esiintyy syrjiviä vinoumia, käyttäjän kyky arvioida tulosteita yhdenvertaisuusnäkökulmasta on erityisen tärkeä. Onkin todettu, ettei ”human-in-the-loop” -periaate yksinään riitä estämään syrjintää [15].

6 Yhteenveto

Algoritmista päätöksentekoa on viimevuosina tutkittu kansainvälisesti useista näkökulmista. Tässä työssä keskityttiin näkökulmaan, joka tutkii päätöksenteon oikeudenmukaisuutta koneoppimismallin koulutusprosessin eri vaiheissa. Tutkielman alussa tarkasteltiin oikeudenmukaisuuden matemaattista määrittelyä sekä yleisimmin oikeudenmukaisuuden mittaamiseksi käytettyjä reiluusmetriikoita. Luvussa 4 keskityttiin tarkastelemaan joitakin matemaattisia menetelmiä, joilla algoritmista reiluutta voidaan vahvistaa prosessin eri vaiheissa ja viimeisenä käytiin läpi yleisesti tunnistettuja haasteita, joita tekoälyjärjestelmien oikeudenmukaisuuteen liittyen on havaittu.

Yhdeksi työn keskeiseksi huomioksi voidaan nostaa kuhunkin erityistilanteeseen sopiva oikeudenmukaisuuskäsitys ja sitä vastaava reiluusmetriikka. On tärkeää ymmärtää, ettei täysin neutraalia ja vinoutumatonta järjestelmää ole, ja siksi oikeudenmukaisuutta tuleekin tarkastella eettisestä näkökulmasta kysyen, mikä on hyväksyttävää ja mikä ei. Kuten edellä todettiin, erilaisia metriikoita ei yhtäaikaisesti voida täyttää ja näin ollen malli voi olla oikeudenmukainen yhden määritelmän mukaan, mutta samalla epäoikeudenmukainen toisen määritelmän nojalla. Tämä korostaa tapauskohtaisuuden merkitystä ja tarvetta arvioida kuhunkin tilanteeseen sopivaa oikeudenmukaisuuskäsitystä.

Vaikka työssä esitellyt menetelmät tarjoavat tietoa algoritmisen reiluuden haasteista ja mahdollisuuksista, on tärkeää huomata, että ne edustavat vain pintaraapaisua jo olemassa oleviin menetelmiin. Uusia menetelmiä algoritmiselle reiluudelle kehitetään jatkuvasti, ja alan tutkimuksessa kiinnitetään koko ajan entistä enemmän huomiota algoritmisen reiluuden syvälliseen ymmärtämiseen ja parantamiseen.

Nämä näkökulmat yhdessä muodostavat monimutkaisen ja ajankohtaisen keskustelun oikeudenmukaisuudesta, yhdenvertaisuudesta ja teknologian roolista näiden periaatteiden toteutumisessa nykypäivän yhteiskunnassa.

Kirjallisuutta

[1] Agarwal, C., Lakkaraju, H., & Zitnik, M. (2021). *Towards a Unified Framework for Fair and Stable Graph Representation Learning.* arXiv.Org. https://doi.org/10.48550/arxiv.2102.13186

[2] Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016, May 23). *Machine Bias.* ProPublica.org. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, luettu 21.3.2024

[3] Arora, A. S., Jentjens, S., Arora, A., McIntyre, J. R., & Sepehri, M. (2022). *Artificial Intelligence (AI) in Marketing: How AI Supports Marketers Throughout the Consumer Journey.* In Managing Social Robotics and Socio-Cultural Business Norms (pp. 75–90). Switzerland: Springer International Publishing AG. https://doi.org/10.1007/978-3-031-04867-8_6

[4] Barocas, S., & Selbst, A. D. (2016). *Big Data's Disparate Impact.* California Law Review, 104(3), 671–732. https://doi.org/10.15779/Z38BG31

[5] Eustasio del Barrio, Gordaliza, P., & Jean-Michel Loubes. (2020). *Review of Mathematical frameworks for Fairness in Machine Learning.* arXiv.Org. https://doi.org/10.48550/arxiv.2005.13755

[6] Caton, S., & Haas, C. (2023). *Fairness in Machine Learning: A Survey.* ACM Computing Surveys. https://doi.org/10.1145/3616865

[7] Dastile, X., Celik, T., & Potsane, M. (2020). *Statistical and machine learning models in credit scoring: A systematic literature survey.* Applied Soft Computing, 91, 106263-. https://doi.org/10.1016/j.asoc.2020.106263

[8] Franco, D., D'Amato, V. S., Pasa, L., Navarin, N., & Oneto, L. (2024). *Fair graph representation learning: Empowering NIFTY via Biased Edge Dropout and Fair Attribute Preprocessing.* Neurocomputing (Amsterdam), 563, 126948-. https://doi.org/10.1016/j.neucom.2023.126948

[9] Garg, S., Sinha, S., Kar, A. K., & Mani, M. (2022). *A review of machine learning applications in human resource management.* International Journal of Productivity and Performance Management, 71(5), 1590–1610. https://doi.org/10.1108/IJPPM-08-2020-0427

[10] Kim, M. P., Ghorbani, A., & Zou, J. (2018). *Multiaccuracy: Black-Box Post-Processing for Fairness in Classification.* Ithaca: Cornell University Library, arXiv.org. https://doi.org/10.48550/arxiv.1805.12317

[11] Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization.* arXiv.Org. https://doi.org/10.48550/arxiv.1412.6980

[12] Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). *Inherent Trade-Offs in the Fair Determination of Risk Scores.* arXiv.Org. https://doi.org/10.48550/arxiv.1609.05807

[13] Lepri, B., Oliver, N., Letouzé, E., Pentland, A., & Vinck, P. (2018). *Fair, Transparent, and Accountable Algorithmic Decision-making Processes: The Premise, the Proposed Solutions, and the Open Challenges.* Philosophy & Technology, 31(4), 611–627. https://doi.org/10.1007/s13347-017-0279-x

[14] Madras, D., Creager, E., Pitassi, T., & Zemel, R. (2018). *Learning Adversarially Fair and Transferable Representations.* arXiv.Org. https://doi.org/10.48550/arxiv.1802.06309

[15] Ojanen, A., Sahlgren, O., Vaiste, J., Björk, A., Mikkonen, J., Kimppa, K., Laitinen, A., & Oljakka, N. (2022). *Algoritminen syrjintä ja yhdenvertaisuuden edistäminen: Arviointikehikko syrjimättömälle tekoälylle.* Valtioneuvoston kanslia, Helsinki. https://urn.fi/URN:NBN:fi:tuni-202310098703

[16] Pessach, D., & Shmueli, E. (2022). *A Review on Fairness in Machine Learning.* ACM Computing Surveys, 55(3), 1–44. https://doi.org/10.1145/3494672

[17] Petrović, A., Nikolić, M., Radovanović, S., Delibašić, B., & Jovanović, M. (2022). *FAIR: Fair adversarial instance re-weighting.* Neurocomputing (Amsterdam), 476, 14–37. https://doi.org/10.1016/j.neucom.2021.12.082

[18] Radovanovic, S., Savic, G., Delibasic, B., & Suknovic, M. (2022). *FairDEA—Removing disparate impact from efficiency scores.* European Journal of Operational Research, 301(3), 1088–1098. https://doi.org/10.1016/j.ejor.2021.12.001

[19] Saraswat, A., Pal, M., Pokhriyal, S., & Abhishek, K. (2023). *Towards fair machine learning using combinatorial methods.* Evolutionary Intelligence, 16(3), 903–916. https://doi.org/10.1007/s12065-022-00702-5

[20] Spinelli, I., Scardapane, S., Hussain, A., & Uncini, A. (2021). *FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning.* arXiv.Org. https://doi.org/10.48550/arxiv.2104.14210

[21] Sun, Y., Fung, B. C. M., & Haghighat, F. (2022). *In-Processing fairness improvement methods for regression Data-Driven building Models: Achieving uniform energy prediction*. Energy and Buildings, 277, 112565-. https://doi.org/10.1016/j.enbuild.2022.112565

[22] Verma, S., & Rubin, J. (2018). *Fairness definitions explained*. Proceedings - International Conference on Software Engineering, 1–7. https://doi.org/10.1145/3194770.3194776

[23] Wang, R., F Maxwell Harper, & Zhu, H. (2020). *Factors Influencing Perceived Fairness in Algorithmic Decision-Making: Algorithm Outcomes, Development Procedures, and Individual Differences*. arXiv.Org. https://doi.org/10.48550/arxiv.2001.09604

[24] Xu, Y., He, H., Shen, T., & Jaakkola, T. (2022). *Controlling directions orthogonal to a classifier*. arXiv.org. https://doi.org/10.48550/arxiv.2201.11259

[25] Yhdenvertaisuuslaki 1325/2014

[26] Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). *Mitigating unwanted biases with adversarial learning*. Ithaca: Cornell University Library, arXiv.org. https://doi.org/10.48550/arXiv.1801.07593

[27] Zhang, X., Chan, F. T. S., Yan, C., & Bose, I. (2022). *Towards risk-aware artificial intelligence and machine learning systems: An overview*. DECISION SUPPORT SYSTEMS, 159, 113800-. https://doi.org/10.1016/j.dss.2022.113800