Ajoneuvojen luokittelu värähtelysignaalin avulla

Valtteri Virtanen

Pro gradu -tutkielma
Huhtikuu 2019

MATEMATIIKAN JA TILASTOTIETEEN LAITOS
TURUN YLIOPISTO

Abstrakti

Innoroad kehittää tienmittausteknologiaa, joka toimiessaan mahdollistaa tiellä kulkevan liikenteen, tien kunnon ja sääolosuhteiden automatisoidun seurannan. Monet käytössä olevat menetelmät joko mittaavat tien kuntoa tai seuraavat liikennevirtaa. Tutkimuksen tavoitteena oli kehittää tilastollinen malli, jolla voidaan automaattisesti tunnistaa tietä käyttävän ajoneuvon kokoluokka sen aiheuttaman värähtelyn perusteella. Automaattisesti toimiva malli auttaisi tienkäytön seuraamisessa ja tien kulumisen arvioinnissa. Tavoitteena oli myös löytää tunnistamisen kannalta tärkeät taajuudet, mikä mahdollisesti auttaisi värähtelymittareiden kehittämisessä.
Tutkimusaineisto kerättiin Muoniosta valtatieltä 21 tien varteen ja sen yli asennetuilla värähtelymittareilla. Aineistona käytettiin ajoneuvojen ylityshetkiä. Raakamuotoisen datan lisäksi mallinnuksessa käytettiin myös taajuustasolle muunnettua aineistoa sekä tietoa havaitusta rengasparien lukumäärästä. Näihin aineistoihin sovitettiin erilaisia tilastollisia- ja koneoppimismalleja, joilla pyrittiin löytämään eri ajoneuvoluokille ominaisia piirteitä. Lisäksi yritettiin eri menetelmillä löytää taajuuksia, joilla eri ajoneuvoluokille ominaiset piirteet ilmenevät.
Tutkimuksen perusteella ajoneuvoluokkien automaattinen tunnistaminen onnistuu tyydyttävällä luotettavuudella värähtelysignaalin avulla. Suurin ongelma tunnistamisessa oli kevyiden (mm. porras- ja viistoperäiset- sekä farmariautot) ja keskikokoisten (mm. paketti- ja tila-autot sekä katumaasturit) ajoneuvojen erottaminen toisistaan. Ajoneuvoluokkien tunnistuksen kannalta tärkeät taajuudet sijaitsivat välillä 1 Hz - 15 906 Hz. Huomattiin kuitenkin, että enemmistö näistä taajuuksista sijaitsee käytetyn taajuusvälin 1 Hz - 16 000 Hz alkupäässä. Ajoneuvoluokan tunnistaminen käytetyillä malleilla tapahtuu siis matalammilla taajuuksilla.

Sisältö

1 Johdanto

Yleisiä tapoja mitata autotien kuntoa ovat laserkeilaus (laser scanning), pudotuspainolaite (falling weight deflectometer, FWD) ja maatutkaus (ground-penetrating radar, GPR) (Lång ym. 2013). Näissä mittauksissa ajetaan tietä pitkin ajoneuvolla, johon mittalaitteet on asennettu. Näillä menetelmillä mittaaminen on hidasta ja mittausten automatisointi on hankalaa. Yleisiä tapoja seurata liikennevirtaa ovat mikroaaltoilmaisimet, kamerat ja induktiosilmukat (Alkila ym. 2014). Nämä menetelmät ovat nopeampia ja helpommin automatisoitavissa. Innoroad kehittää mittausteknologiaa, joka yhdistää autotien kunnon mittaamisen ja liikennevirran seuraamisen. Toimiessaan teknologia mahdollistaa tiellä kulkevan liikenteen, tien kunnon ja sääolosuhteiden automatisoidun seurannan, mikä auttaa suunnittelemaan mm. teiden huoltoa ja liikenteen ohjausta. Teknologia perustuu tien poikki ja sen reunalla kulkeviin mittareihin, jotka tallentavat tietä käyttävien ajoneuvojen aiheuttaman värähtelysignaalin.
Tämän tutkimuksen tavoitteena oli kehittää tilastollinen malli, jolla voidaan automaattisesti tunnistaa ajoneuvon kokoluokka (kevyt/keskikokoinen/raskas) mittareiden keräämästä värähtelysignaalista. Tien kulumisen arvioiminen helpottuu, jos tiedetään kuinka paljon ja minkä kokoisilla ajoneuvoilla tietä on käytetty. Ei ole kuitenkaan välttämätöntä tunnistaa jokaista ajoneuvoa, vaan tarpeeksi hyvät arviot ajoneuvojen määristä ja kokoluokista riittävät. Tutkimuksen toisena tavoitteena oli löytää ajoneuvon kokoluokan tunnistuksen kannalta tärkeät taajuudet. Tätä tietoa voitaisiin jatkossa käyttää esimerkiksi mittareiden parantamiseen.
Aineiston analyysiin käytettiin tilastollisia menetelmiä (pääkomponenttianalyysi, k:n lähimmän naapurin -menetelmä ja yleistetty lineaarinen malli), koneoppimismenetelmiä (satunnaismetsä ja Boruta) sekä neuroverkkoja (konvoluutio- ja tavallinen neuroverkko).
Tutkielma etenee suraavasti. Luvussa 2 esitellään aineisto sekä kerrotaan, miten aineisto esikäsiteltiin. Tutkielmassa käytettyjä menetelmiä avataan luvussa 3. Luvussa 4 kuvaillaan aineistoa pääkomponenttianalyysillä. Luvussa 5 aineistoa luokitellaan erilaisilla menetelmillä sekä etsitään luokittelun kannalta tärkeitä taajuuksia. Luokittelun ja tärkeiden taajuuksien tunnistuksen tuloksia esitellään myös luvussa 5. Tulosten tulkintaa sekä pohdintaa niiden yleistettävyydestä ja merkityksestä jatkokehityksen kannalta on luvussa 6.

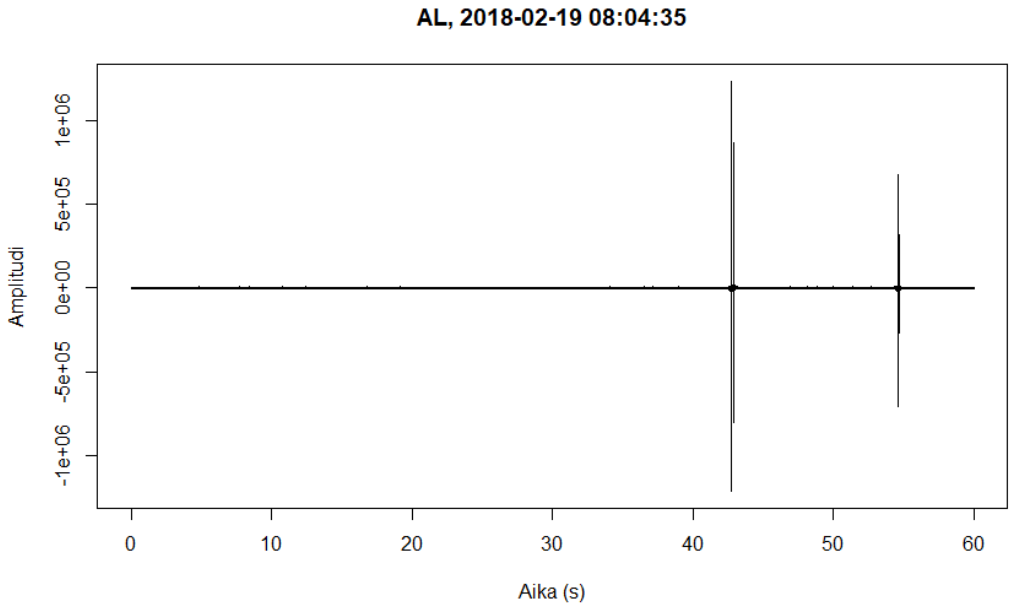
2 Aineisto

Kuva 1: Riistakameran ottamista kuvista rajattuja ajoneuvoja.

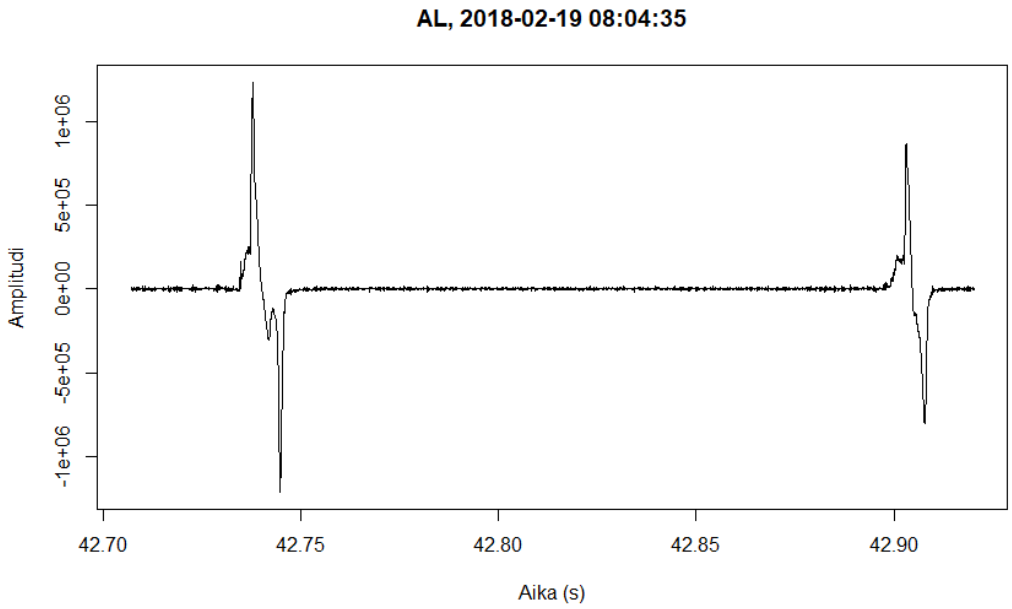
Aineisto kerättiin Muonioon valtatien 21 varteen sijoitetusta testipisteestä mittareilla, jotka rekisteröivät tietä käyttävän ajoneuvon aiheuttamaa värähtelyä. Molemmin puolin autotietä tien suuntaisesti oli asennettu kaksi mittaria BL ja BR, ja molemmille kaistoille poikittain oli asennettu toiset kaksi mittaria AL ja AR. Apuna oli myös tien viereen asennettu riistakamera, jonka avulla voitiin luokitella ajoneuvoja Kuvan 1 kaltaisista kuvista. Mittarit AL ja BR mittasivat kameraa kohti tulevan liikenteen aiheuttamaa värähtelyä ja mittarit AR ja BL kamerasta poispäin menevän liikenteen aiheuttamaa värähtelyä. Kustakin mittarista saatiin havaintoina sen tallentama värähtely (Kuva 2). Näytteenottotaajuus oli 32 000 Hz eli yksi mittari rekisteröi minuutissa 1 920 000 yksittäistä värähtelyn datapistettä. Aineistoa tarkasteltaessa BL- ja BR-mittaukset vaikuttivat melko samanlaisilta eri ajoneuvoille, joten päädyttiin analysoimaan AL- ja AR-mittauksia.
Analyysissä käytettiin havaintoja seitsemältä päivältä vuoden 2018 tammi-helmikuussa. Käytetyt havainnot valittiin valoisuuden ja näkyvyyden perusteella siten, että ajoneuvot oli helppo silmämääräisesti tunnistaa kuvista. Aineistosta poistettiin analyysiä haittaavia mittauksia, kuten mittauksia liian lähekkäin toisiaan ajavista ajoneuvoista tai liian kevyistä ajoneuvoista (esimerkiksi pyöräilijä).
Ajoneuvot jaettiin kuvien perusteella käsin neljään luokkaan: kevyet, keskikokoiset, raskaat ja tyhjät (Taulukko 1). Tyhjiä mittauksia valittiin kummastakin mittarista 100 kappaletta, joka on vähän pienempi kuin eri luokkien mittauksien lukumäärän keskiarvo (105). Tällä pyrittiin siihen, että malli ottaisi huomioon tyhjät mittaukset, mutta ei painottaisi niitä liikaa. Tyhjien mittausten mukaanottamisen tarkoituksena oli, että malli tunnistaisi myös hetket, joina mittareiden ohi ei mene ajoneuvoja, eikä yrittäisi luokitella niitä muihin ajoneuvoluokkiin. Peräkärryt jätettiin pois analyysistä, koska mittauksia oli liian vähän luokittelua varten (8 AL-mittausta ja 11 AR-mittausta).

Taulukko 1: Ajoneuvoluokat. Kevyet-luokkaan luettiin mm. porras- ja viistoperäiset- sekä farmariautot, keskikokoiset-luokkaan mm. paketti- ja tila-autot sekä katumaasturit, raskaat-luokkaan rekka- ja linja-autot sekä muut raskaan liikenteen ajoneuvot ja tyhjät-luokkaan mittauksia hetkiltä, joina mittareiden antureiden yli ei mennyt ajoneuvoja.

Mittari	**Kevyet**	**Keskikokoiset**	**Raskaat**	**Tyhjät**	**Yhteensä**
AR	208	87	21	100	416
AL	206	79	29	100	414
Yhteensä	414	166	50	200	830



Kuva 2: AL-mittarin rekisteröimä värähtely, jossa näkyy keskikokoisen ajoneuvon ylitys noin 43 sekunnin kohdalla ja kevyen ajoneuvon ylitys noin 55 sekunnin kohdalla.

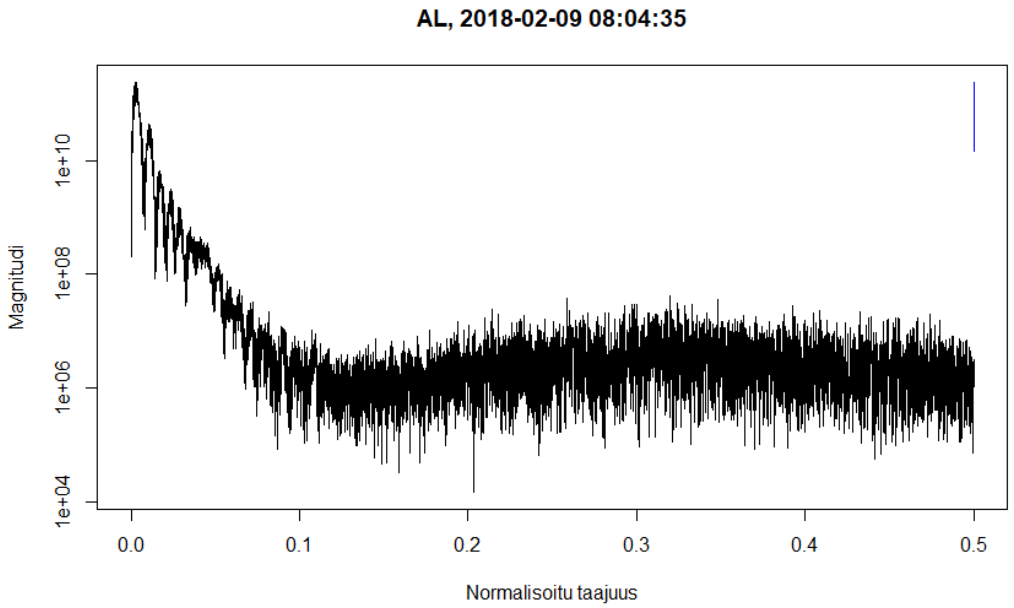


Kuva 3: Keskikokoisen ajoneuvon ylitys rajattuna kuvasta 2.

Aineiston esikäsittely

Mittarit rekisteröivät värähtelyä koko ajan, oli tiellä liikennettä tai ei, joten ensin aineistosta rajattiin ajoneuvojen ylitykset (Kuva 3). Ylityksen katsottiin alkaneen kohdasta, jossa signaalin amplitudi ylitti 20 000 Pa, ja loppuneen, kun signaalin amplitudi oli alle 20 000 Pa 0,55 sekunnin ajan. Eri rajoja testattiin pienellä otoksella ajoneuvoja ja näiden rajojen havaittiin toimivan parhaiten.
Ylityksen signaali muutettiin sen sisältämiksi taajuuksiksi, koska taajuuksien avulla voitiin tutkia signaalin spektriä. Spektriä haluttiin tutkia siksi, että oletettiin ajoneuvoluokkien välisten erojen näkyvän erilaisena värähtelynä tietyillä taajuuksilla. Muunnos tehtiin nopealla diskreetillä Fourier-muunnoksella (Fast Discrete Fourier Transform, FFT), joka muuntaa signaalin aikainformaation kunkin taajuuden magnitudi- ja vaihekomponentiksi. Tässä analyysissä käytettiin hyväksi vain magnitudien neliöitä eli tehospektriä (power spectrum). Jokaiselle ylitykselle muodostettiin Kuvan 4 mukainen periodogrammi, joka on tehospektrin jakauman estimaatti.
Analyysiä varten koottiin kolme eri tavoilla käsiteltyä aineistoa: havaittu aineisto (H), spektriaineisto (S) sekä spektriaineisto ja piikit (SP) (Taulukko 2). H-aineistoon kuuluivat värähtelymittaukset ajoneuvojen ylityshetkiltä, S-aineisto käsitti värähtelymittausten periodogrammit ja SP-aineisto sisälsi sekä periodogrammit että tiedon niin sanottujen piikkien (signaalin amplitudi yli 20 000 Pa) lukumäärästä. Erilaisia aineistoja käytettiin erilaisista syistä. H-aineistoa käytettiin, koska haluttiin nähdä, onnistuuko luokittelu käsittelemättömällä aineistolla. S-aineistoa käytettiin, koska oletettiin eri painoisten ajoneuvojen aiheuttavan eri taajuista värähtelyä, mikä auttaisi luokittelussa.

SP-aineistoa käytettiin, koska oletettiin piikkien lukumäärän helpottavan luokittelua. Esimerkiksi jos signaalissa näkyy yhdeksän piikkiä, voidaan suurella varmuudella sanoa ajoneuvon olevan raskaan liikenteen ajoneuvo, sillä muun kaltaisilla ajoneuvoilla on harvoin yhdeksää rengasparia.



Kuva 4: Kuvan 3 keskikokoisen ajoneuvon ylityksestä muodostettu periodogrammi.

Taulukko 2: Analyysissä käytetyt aineistot. Periodogrammeille tehtiin luonnolliset logaritmimuunnokset ja ne keskitettiin vähentämällä niistä keskiarvo kaikkien peridogrammien yli ja skaalattiin jakamalla ne keskihajonnalla kaikkien periodogrammien yli. Piikeillä tarkoitetaan kohtia, joissa signaalin amplitudi ylittää arvon 20 000. Niiden avulla havaitaan, kuinka monta rengasparia ajoneuvolla oli.

Aineisto	**Sisältö**
H	Signaali
S	Spektrin periodogrammi
SP	Spektrin periodogrammi ja tieto piikkien lukumäärästä

3 Menetelmät

Tässä luvussa kerrotaan analyysissä käytetyistä menetelmistä. Menetelmillä pyrittiin tunnistamaan, mihin kokoluokkaan ajoneuvo kuuluu sen aiheuttaman värähtelyn perusteella, sekä erittelemään, mitkä taajuudet ovat tunnistuksen kannalta tärkeitä. Selvennetään kuitenkin ensin luvussa käytettyjä menettelyitä ja termejä.

3.1 Yleisiä käsitteitä

Havainto

Havainnolla tarkoitetaan yleisesti ajoneuvon ylityksestä mitattua värähtelyä. Se voi kuulua H-, S- tai SP-aineistoon.

Muuttuja

Muuttuja on havainnon osa.

Luokka

Luokilla tarkoitetaan ajoneuvoluokkia, joihin havainnot pyritään luokittelemaan.

Luokittelu

Luokittelulla tarkoitetaan mittausten jakamista luokkiin. Esimerkiksi mittauksen signaalista tai signaalin spektristä voidaan mahdollisesti tunnistaa yksittäinen tai useampi raskaille ajoneuvoille tyypillinen piirre ja siten luokitella mittaus raskaiden ajoneuvojen luokkaan.

Luokitteluvirhe

Väärin luokiteltujen mittausten suhteellista osuutta kutsutaan luokitteluvirheeksi. Jos esimerkiksi kymmenestä mittauksesta kaksi luokitellaan väärin, on luokitteluvirhe 0,2.

Kategorinen tarkkuus

Oikein luokiteltujen mittausten osuutta kaikista mittauksista kutsutaan kategoriseksi tarkkuudeksi.

Tappiofunktio

Tappiofunktio rankaisee mallia väärin luokitelluista mittauksista.

Optimaatiofunktio

Optimaatiofunktio yrittää minimoida jonkin tappiofunktion arvon esimerkiksi muuttamalla neuroverkon painoja.

Opetus

Luokittelumallin opetuksessa käytetään opetusaineistoa sekä optimaatio- ja tappiofunktiota. Ensin mallille syötetään opetusaineisto, jonka se luokittelee. Sitten tappiofunktio rankaisee mallia, minkä jälkeen optimaatiofunktio muuttaa mallia. Tätä toistetaan, kunnes malli on niin sanotusti oppinut luokittelemaan aineiston tarpeeksi hyvin.

Opetusaineisto

Opetusaineisto on se osa aineistoa, jota käytetään mallin opettamiseen.

Validointiotos

Tässä tutkimuksessa neuroverkkoja opetettaessa opetusaineisto jaetaan ristiinvalidoinnilla opetus- ja validointiotoksiin. Opetusotoksella opetetaan mallia ja validointiotoksella validoidaan malli. Mallin validoinnilla tarkoitetaan mallin ennusteiden vertaamista todellisiin arvoihin.

Testiaineisto

Testiaineisto on se osa aineistoa, jota käytetään mallin lopulliseen testaamiseen. Testiaineistoa ei käytetä mallin opetuksessa.

Ristiinvalidointi

Ristiinvalidoinnilla tarkoitetaaan tässä tutkimuksessa 10-sarakkeista ristiinvalidointia. Tällöin aineisto jaetaan satunnaisesti kymmeneen yhtä suureen otokseen, joista yhdeksää käytetään opetusotoksena ja yhtä validointiotoksena. Tämä toistetaan kymmenen kertaa siten, että jokainen otoksista on vain yhden kerran validointiotoksena. Näin saaduista kymmenestä tuloksesta (esimerkiksi todennäköisyydestä) lasketaan keskiarvo, jota käytetään lopullisena ristiinvalidoituna tuloksena.

Syöte

Neuroverkon sisääntulokerroksella luetaan havainto, jonka jälkeen parametrivektorista tai -matriisista käytetään nimitystä syöte. Syötteen muoto riippuu neuroverkon kerroksien tyypeistä ja kerroksien neuronien lukumääristä, joten halutaan erottaa syöte havainnosta.

Aktivaatiofunktio

Aktivaatiofunktio määrää neuronilta ulostulevan syötteen. Esimerkiksi binäärinen aktivaatiofunktio määrää nollaa pienemmän sisääntulevan syötteen summan nollaksi ja nollaa suuremman tai yhtä suuren summan yhdeksi:

$$f(x_{si}) = \begin{cases} 0, & \text{kun} \quad x_{si} < 0 \\ 1, & \text{kun} \quad x_{si} \geq 1 \end{cases},$$

missä x_{si} on neuronille i sisääntulevan syötteen summa.

ROC-käyrä

Binäärisessä mallissa, jossa vasteen luokat ovat positiivinen ja negatiivinen, todellinen positiivinen taso (true positive rate) eli sensitiivisyys on se osuus kaikista oikeasti positiivisista tapauksista, joka luokitellaan oikein positiivisiksi. Väärä positiivinen taso (false positive rate) on se osuus kaikista oikeasti negatiivisista tapauksista, joka luokitellaan väärin positiivisiksi. ROC-käyrä saadaan piirtämällä todellinen positiivinen taso y-akselille ja väärä positiivinen taso x-akselille eri kynnysarvoilla (threshold).

AUC

ROC-käyrän alle jäävä pinta-ala AUC kertoo, millä todennäköisyydellä satunnaisesti valitun positiivisen havainnon tuottama pistemäärä (score) on suurempi kuin satunnaisesti valitun negatiivisen havainnon tuottama pistemäärä. AUC:n arvo on todennäköisyys välillä $0,5 - 1$.

Selitetyn devianssin osuus, D^2

Devianssi kertoo, kuinka paljon sovitettu malli poikkeaa saturoidusta mallista, jossa jokaiselle muuttujalle on oma parametri siten, että malli sovittuu täydellisesti. Sovitetun mallin devianssi lasketaan saturoidun mallin ja sovitetun mallin uskottavuuksien avulla:

$$Dev_{fitted} = 2(log(L(\hat{\boldsymbol{\theta}}_{sat})) - log(L(\hat{\boldsymbol{\theta}}_{fitted}))),$$

jossa $\hat{\boldsymbol{\theta}}_{sat}$ on saturoidun mallin parametrien suurimman uskottavuuden estimaattien vektori ja $\hat{\boldsymbol{\theta}}_{fitted}$ on sovitetun mallin parametrien suurimman uskottavuuden estimaattien vektori. Saturoidun mallin uskottavuus on yksi, jolloin kaava supistuu muotoon

$$Dev_{fitted} = -2log(L(\hat{\boldsymbol{\theta}}_{fitted})).$$

Vastaavasti niin sanotun nolla-mallin (mallissa on pelkästään vakiotermi) devianssi saadaan kaavasta

$$Dev_{null} = -2log(L(\hat{\boldsymbol{\theta}}_{null})),$$

jossa $\hat{\boldsymbol{\theta}}_{null}$ on nolla-mallin parametrien suurimman uskottavuuden estimaattien vektori.

Selitetyn devianssin osuus kertoo, kuinka suuren osuuden sovitettu malli selittää devianssista. Se lasketaan sovitetun mallin ja nolla-mallin devianssien avulla:

$$D^2_{fitted} = \frac{Dev_{null} - Dev_{fitted}}{Dev_{null}},$$

jossa D^2_{fitted} on sovitetun mallin selitetyn devianssin osuus, Dev_{null} on nolla-mallin devianssi ja Dev_{fitted} on sovitetun mallin devianssi.

Ohjaamaton menetelmä

Ohjaamattomalla (unsupervised) menetelmällä tarkoitetaan menetelmää, joka ei hyödynnä vastemuuttujan tietoja. Esimerkiksi tässä tutkimuksessa tämä tarkoittaa, että

menetelmä ei käytä tietoa ajoneuvojen oikeista luokista. Pääkomponenttianalyysi on esimerkki tällaisesta menetelmästä. Muut tässä tutkimuksessa käytetyt luokittelumenetelmät ovat ohjattuja (supervised) mentelmiä, koska ne hyödyntävät tietoa ajoneuvojen oikeista luokista (opetusaineistoa).

Vahvasti olennainen muuttuja

Muuttuja x on vahvasti olennainen (strongly relevant), jos sen jättäminen pois yksistään heikentää ideaalin luokittimen ennustustarkkuutta (Rudnicki, Wrzesień ja Paja 2015).

Heikosti olennainen muuttuja

Muuttuja x on heikosti olennainen (weakly relevant), jos se ei ole vahvasti olennainen ja on olemassa sellainen muuttujien osajoukko $S, x \notin S$, että ideaalin luokittimen ennustustarkkuus on heikompi joukolla S kuin joukolla $S \cup x$ (Rudnicki, Wrzesień ja Paja 2015).

Epäolennainen muuttuja

Muuttuja x on epäolennainen (irrelevant), jos se ei ole vahvasti olennainen tai heikosti olennainen (Rudnicki, Wrzesień ja Paja 2015).

Minimal optimal -ongelma

Etsitään vahvasti olennaisista muuttujista muodostuva joukko ja sellainen heikosti olennaisten muuttujien osajoukko, että jäljelle jäävät heikosti olennaiset muuttujat sisältävät pelkästään hyödytöntä informaatiota.

All relevant -ongelma

Etsitään kaikki vahvasti ja heikosti olennaiset muuttujat (Rudnicki, Wrzesień ja Paja 2015).

3.2 Fourier-muunnos

Aineiston esikäsittelyssä käytettiin nopeaa diskreettiä Fourier-muunnosta (Fast Discrete Fourier Transform, FFT), joka muuntaa signaalin aikainformaation kunkin taajuuden magnitudi- ja vaihekomponentiksi. Tässä analyysissä käytettiin hyväksi vain magnitudien neliöitä eli tehospektriä.

$$FFT : y_h = \sum_{n=1}^{N} z_n * exp\{\frac{-2\pi i(n-1)(h-1)}{N}\},$$

jossa y_h on tehospektrin h :s muuttuja, missä $h = \{1, \ldots, h, \ldots, H\}$, i on imaginaariyksikkö ja z_n on signaalin n :s muuttuja, missä $n = \{1, \ldots, n, \ldots, N\}$ (Cooley ja Tukey 1965).

3.3 Pääkomponenttianalyysi

Pääkomponenttianalyysissa (Principal Component Analysis, PCA) ideana on korvata alkuperäiset muuttujat niiden lineaarikombinaatioilla (ns. pääkomponenteilla) menettämällä mahdollisimman vähän informaatiota. Tällöin muuttujien lukumäärä tyypillisesti pienenee huomattavasti, mutta eri luokille ominaiset piirteet säilyvät. PCA:ssa muuttujaavaruuden akseleita kierretään siten, että ensimmäinen dimensio osoittaa suurimman varianssin suuntaan. Sama toistetaan muille dimensioille siten, että ne pysyvät kohtisuorassa aiempia dimensioita vastaan. PCA on järkevä dimension supistusmenetelmä, mikäli varianssin voidaan olettaa vastaavan informaatiota.

Holland (2008) määrittelee PCA:n seuraavasti: Olkoon $\boldsymbol{X}$ $N \times p$ havaintomatriisi, jonka kovarianssimatriisi on $\boldsymbol{\Sigma}$. Pääkomponentit muodostetaan seuraavasti:

1. Ensimmäinen pääkomponentti $\boldsymbol{y}_1$ on muuttujien $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ lineaarikombinaatio

   $$\boldsymbol{y}_1 = u_{1,1}\boldsymbol{x}_1 + \cdots + u_{1,p}\boldsymbol{x}_p = \boldsymbol{X}\boldsymbol{u}_1,$$

   jossa $\boldsymbol{u}_1$ on ensimmäisen pääkomponentin painovektori. Painovektori $\boldsymbol{u}_1$ etsitään siten, että $u_{1,1}^2 + \cdots + u_{1,p}^2 = 1$ ja $Var(\boldsymbol{y}_1) = Var(\boldsymbol{X}\boldsymbol{u}_1) = \boldsymbol{u}_1^T\boldsymbol{\Sigma}\boldsymbol{u}_1$ on mahdollisimman suuri.

2. Muodostetaan toinen pääkomponentti vastaavasti. Nyt painovektori $\boldsymbol{u}_2$ etsitään siten, että $u_{2,1}^2 + \cdots + u_{2,p}^2 = 1$, $Var(\boldsymbol{y}_2)$ on mahdollisimman suuri ja $corr(\boldsymbol{y}_1, \boldsymbol{y}_2) = corr(\boldsymbol{X}\boldsymbol{u}_1, \boldsymbol{X}\boldsymbol{u}_2) = \boldsymbol{u}_1^T\boldsymbol{\Sigma}\boldsymbol{u}_2 = 0$.

3. Muodostetaan kolmas pääkomponentti kuten aiemmin. Painovektori $\boldsymbol{u}_3$ etsitään siten, että $u_{3,1}^2 + \cdots + u_{3,p}^2 = 1$, $Var(\boldsymbol{y}_3)$ on mahdollisimman suuri ja $corr(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3) = corr(\boldsymbol{X}\boldsymbol{u}_1, \boldsymbol{X}\boldsymbol{u}_2, \boldsymbol{X}\boldsymbol{u}_3) = \boldsymbol{u}_1^T\boldsymbol{\Sigma}\boldsymbol{u}_3 = \boldsymbol{u}_2^T\boldsymbol{\Sigma}\boldsymbol{u}_3 = 0$.

4. Jatketaan samalla tavalla, kunnes lineaarikombinaatioiden lukumäärä on p.

5. Saadaan $\boldsymbol{Y}$, joka on $N \times p$ pääkomponenttimatriisi. $\boldsymbol{Y}$:n kovarianssimatriisi on diagonaalimatriisi, jonka diagonaalilukujen $\{d_1, \ldots, d_p\}$ summa on $\boldsymbol{Y}$:n kokonaisvarianssi. Pääkomponentin niin sanottu lataus (loading) kertoo, kuinka suuren osan kokonaisvarianssista pääkomponentti selittää. Esimerkiksi ensimmäisen pääkomponentin lataus on

   $$100 * \frac{d_1}{d_1 + \cdots + d_p}\%.$$

   PCA:ssa tavoitteena on muuttujien lukumäärän pieneminen, joten vain osa pääkomponenteista valitaan kuvaamaan aineistoa. Valintakriteerinä voi olla esimerkiksi, että valitaan ne pääkomponentit, joiden lataus on suurempi kuin yksi prosentti. Toinen valintakriteeri voisi olla, että valitaan järjestyksessä pääkomponentteja, kunnes kumulatiivinen selitetyn varianssin osuus saavuttaa jonkin ennalta määrätyn rajan.

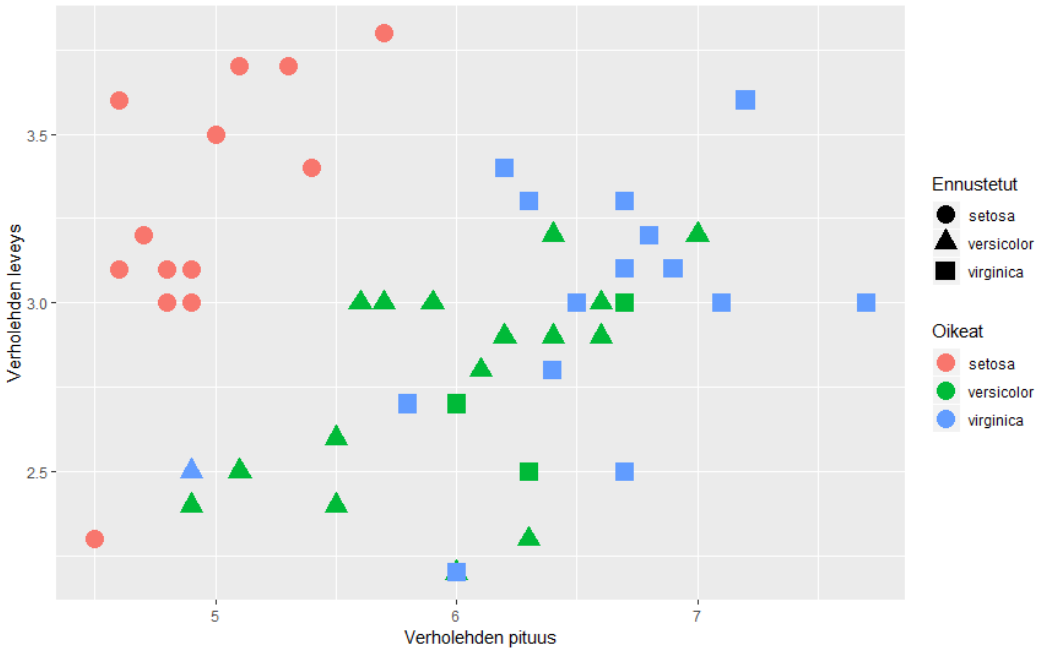
Luokkien välisiä eroja voidaan tarkastella kuvaamalla pääkomponentit xy-koordinaatistossa.

3.4 k:n lähimmän naapurin menetelmä

Menetelmää, jossa havainnot jaetaan ryhmiin etsimällä jokaiselle havainnolle euklidisen etäisyyden perusteella k lähintä havaintoa (Kuva 5), kutsutaan k:n lähimmän naapurin menetelmäksi (k nearest neighbours, knn). Jos yksi havainto sijaitsee pisteessä $\boldsymbol{a} = (a_1, a_2)$ ja toinen havainto pisteessä $\boldsymbol{b} = (b_1, b_2)$, niiden euklidinen etäisyys on $d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$.

Olkoot $\boldsymbol{X}$ $N \times p$ havaintomatriisi ja $k, 1 \leq k \leq N$, valittu lähimpien naapureiden lukumäärä. Tällöin k :n lähimmän naapurin -menetelmä on seuraava:

1. Jaetaan havainnot opetus- ja testiaineistoihin.
2. Etsitään jokaiselle testiaineiston havainnolle euklidisen etäisyyden perusteella k lähintä opetusaineiston havaintoa.
3. Luokitellaan jokainen testiaineiston havainto luokkaan, johon enemmistö sen k :sta lähimmästä opetusaineiston havainnosta kuuluu.



Kuva 5: Iris-aineistolla (Fisher 1936) tehty esimerkki luokittelusta knn:llä. Menetelmää käytettiin aineiston kaikille muuttujille, joita ovat verholehden leveyden ja pituuden lisäksi terälehden leveys ja pituus sekä tieto kukan lajikkeesta (Iris setosa, versicolor tai virginica). Kukkien verholehtien leveys on y-akselilla ja verholehtien pituus x-akselilla. Kukkien oikeat luokat on merkitty eri väreillä ja knn:n muodostamat luokat eri muodoilla.

3.5 Satunnaismetsä

Breimanin satunnaismetsä-algoritmi (random forest, rf) luokittelee havainnot eri luokkiin päätöspuiden avulla. Tavoitteena on tehdä päätöspuista mahdollisimman syviä, koska silloin ne sisältävät mahdollisimman paljon informaatiota aineistosta. Puun syvyydellä tarkoitetaan jakojen määrää. Jaolla tarkoitetaan aineiston jakamista kahteen osaan jonkin ominaisuuden perusteella. Satunnaismetsä on harvoin tarkkuudeltaan optimaalinen koneoppimisratkaisu, mutta sen etuja ovat suoraviivainen soveltaminen ja arvio annetun luokittelun tarkkuudesta, joten menetelmää voidaan hyödyntää alustavan arvion muodostamiseen saavutettavissa olevasta tarkkuudesta. Hyvä tasapaino AUC:n, prosessointiajan ja muistin käytön välille saadaan, kun päätöspuita on 64 - 128 (Oshiro, Perez ja Baranauskas 2012).

Efron ja Hastie (2016) esittävät satunnaismetsä-algoritmin näin (Kuva 6):

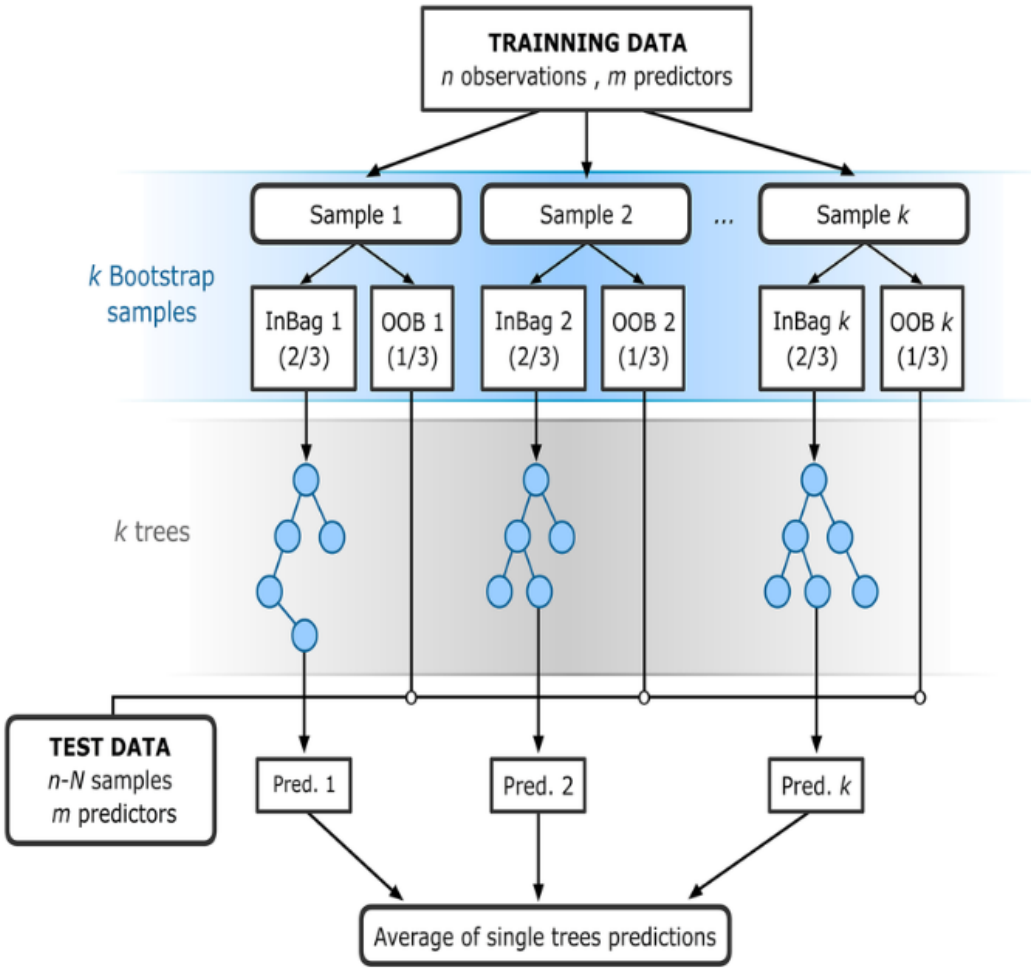
1. Olkoon opetusaineisto $\boldsymbol{h} = (\boldsymbol{X}, \boldsymbol{y})$, jossa $\boldsymbol{X}$ on $N \times p$ havaintomatriisi ja $\boldsymbol{y}$ on vastemuuttujan vektori $\in R^N$. Kiinnitetään $m \leq N$ ja puiden lukumäärä k.

2. Tehdään puut $b = 1, 2, \ldots, k$ seuraavasti:

a) Muodostetaan opetusaineiston bootstrap-muunnos $\boldsymbol{h}_b^*$ ottamalla satunnaisesti takaisinpanolla j $(< N)$ havaintoa.
b) Käytetään $\boldsymbol{h}_b^*$:n aineistoa ja muodostetaan mahdollisimman syvä puu $\hat{r_b}$ valitsemalla ennen kutakin jakoa satunnaisesti m muuttujaa.
c) Tallennetaan puu ja jokaisen opetusaineiston muuttujan bootstrap-esiintymistiheydet, eli kuinka monta kertaa kukin opetusaineiston havainto sisältyi bootstrap-otokseen.

3. Lasketaan satunnaismetsäsovite ennustuspisteessä x_0 keskiarvona $\hat{r_{rf}}(x_0) = \frac{1}{k} \sum_{b=1}^{k} \hat{r_b}(x_0)$.

4. Lasketaan OOB_i (out-of-bag) virhe kaikille opetusaineiston vastemuuttujille y_i käyttämällä sovitetta $\hat{r_{rf}}$, joka saadaan laskemalla keskiarvo niistä $\hat{r_b}(x_i)$:stä, joissa havainto i ei ollut bootstrap-otoksessa. OOB kokonaisvirhe on keskiarvo OOB_i virheistä.



Kuva 6: Satunnaismetsä-algoritmin toiminta (kuva kopioitu Rodríguez Galiano ym. 2016). Kuvassa algoritmin vaihe 1 on ylhäällä valkoisella pohjalla, vaihe 2a on sinisellä pohjalla, vaiheet 2b ja 2c ovat harmaalla pohjalla ja vaiheet 3 ja 4 ovat alhaalla valkoisella pohjalla.

3.6 Boruta

Boruta on satunnaismetsä-algoritmin ympärille rakennettu muuttujanvalinta-algoritmi (feature selection algorithm). Sen avulla voidaan poistaa ne muuttujat, jotka ovat luokittelun kannalta epäolennaisia. Satunnaismetsä-algoritmista voidaan laskea OOB-kokonaisvirheeseen perustuva muuttujan tärkeys (variable importance) seuraavalla tavalla:

1. Lasketaan OOB-kokonaisvirhe.
2. Permutoidaan muuttujan i arvot satunnaisesti ja lasketaan uudelleen OOB-kokonaisvirhe. Tällä tavalla aiheutunutta OOB-kokonaisvirheen kasvun määrää pidetään muuttujan i tärkeyden määränä.

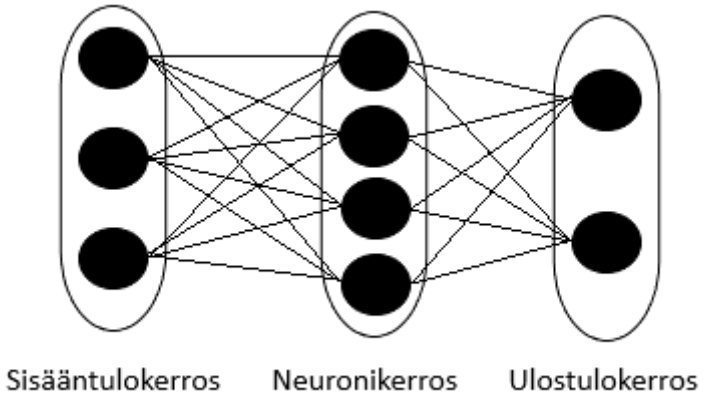
Borutassa muuttujien tärkeydet keskitetään vähentämällä niistä kaikkien muuttujien tärkeyksien keskiarvo ja skaalataan jakamalla ne kaikkien muuttujien tärkeyksien keskihajonnalla. Näin saadaan muuttujien tärkeyksien standardipistemäärät (Z score). Muuttujien tärkeyksien standardipistemääriä käytetään muuttujien tärkeyksien mittana, koska standardipistemäärä ottaa huomioon OOB-kokonaisvirheen kasvun keskiarvon heilahtelut satunnaismetsän puiden välillä. Standardipistemäärää ei voida kuitenkaan suoraan käyttää muuttujan tärkeyden mittana, sillä satunnaismetsän antamat muuttujien tärkeydet eivät ole normaalijakautuneita. Tämä huomioidaan kasvattamalla satunnaisuutta lisäämällä havaintoihin kopiot (varjomuuttujat) kaikista muuttujista. Varjomuuttujien arvoja sekoitetaan satunnaisesti siten, että ne eivät korreloi alkuperäisten muuttujien kanssa. Satunnaismetsä sovitetaan koko aineistolle ja muuttujien tärkeyttä verrataan siihen varjomuuttujaan, jolla on suurin tärkeys, koska varjomuuttujien tärkeys voi poiketa nollasta vain satunnaisvaihtelun takia. (Kursa, Jankowski ja Rudnicki 2010).
Boruta eroaa perinteisistä muuttujanvalinta-algoritmeistä siten, että se yrittää ratkaista all relevant -ongelmaa. Suurin osa perinteisistä muuttujanvalinta-algoritmeistä yrittää ratkaista minimal optimal -ongelmaa.

Borutan toiminta:

1. Kasvatetaan havaintoja lisäämällä kopio (varjomuuttuja) jokaisesta muuttujasta. Varjomuuttujia lisätään aina vähintään viisi kappaletta, vaikka muuttujia olisi alle viisi kappaletta.
2. Sekoitetaan varjomuuttujia satunnaisesti siten, että ne eivät korreloi alkuperäisten muuttujien kanssa.
3. Ajetaan satunnaismetsä koko aineistolle ja lasketaan muuttujien standardipistemäärät.
4. Etsitään varjomuuttuja, jolla on suurin standardipistemäärä (MZSA), ja merkitään osuma jokaiselle muuttujalle, jonka standardipistemäärä on suurempi kuin MZSA:n.

5. Verrataan kunkin muuttujan osumien lukumäärää binomijakaumaan $Bin(n, p)$, jossa n on iteraatioiden lukumäärä ja p on 0,5.
6. Muuttujat, joiden osumien lukumäärä on merkitsevästi pienempi kuin binomijakauman odotusarvo, merkitään merkityksettömiksi ja ne poistetaan aineistosta.
7. Merkitään tärkeiksi muuttujat, joiden osumien lukumäärä on merkitsevästi suurempi kuin binomijakauman odotusarvo.
8. Poistetaan kaikki varjomuuttujat.
9. Toistetaan kohtia 1 - 8, kunnes kaikki muuttujat on merkitty joko tärkeiksi tai merkityksettömiksi tai kunnes valittu toistojen määrä on saavutettu.

3.7 Neuroverkko



Kuva 7: Esimerkki yksinkertaisesta neuroverkosta. Ympyrät kuvaavat neuroneja ja ympyröiden väliset viivat synapseja.

Neuroverkko (Neural Network, NN) on epälineaarinen malli, joka koostuu sisääntulokerroksesta (input layer), yhdestä tai useammasta neuronikerroksesta (densely-connected layer tai hidden layer) ja ulostulokerroksesta (output layer). Kerrokset muodostuvat niin kutsutuista neuroneista ja ne on kytketty toisiinsa niin kutsutuilla synapseilla (Kuva 7), joille neuroverkko määrittää kytkentäkohtaiset painot. Neuroverkkoon voidaan myös lisätä muita kerroksia, kuten otoksen normalisointikerros kiihdyttämään konvergointia tai katokerros estämään ylisovittumista.

Neuroverkon toiminta (Kuva 7):

1. Sisääntulokerroksella luetaan havainnon muuttujat ja siirretään ne neuronikerrokselle. Jokainen kuvan 7 sisääntulokerroksen ympyrä vastaa yhtä muuttujaa.
2. Kerrotaan syöte synapsikohtaisilla painoilla.

3. Neuronikerroksella summataan kaikki edelliseltä kerrokselta tulevat syötteet ja lisätään niihin neuronikohtaiset harhaparametrit. Sen jälkeen ennalta määrätty aktivaatiofunktio laskee kaikkien syötteiden ja harhaparametrien summalle epälineaarisen kuvauksen, joka siirretään seuraavalle neuronikerrokselle tai ulostulokerrokselle.
4. Kerrotaan syöte synapsikohtaisilla painoilla.
5. Ulostulokerrokselta saadaan mallin tulos, eli esimerkiksi luokittelumallissa havainnon todennäköisyys kuulua tiettyyn luokkaan.

Luokittelun opettaminen neuroverkolle:

1. Opetusaineisto jaetaan satunnaisesti opetus- ja validointiotoksiin, joiden koot ovat ennalta määrättyjä.
2. Luetaan seuraava havainto neuroverkkoon.
3. Ennalta valittu tappiofunktio laskee mallin tappion.
4. Ennalta valittu optimaatiofunktio yrittää minimoida tappion päivittämällä synapsien painot.
5. Kohtia 2 - 4 toistetaan, kunnes koko opetusaineisto on luettu neuroverkkoon.
6. Malli validoidaan validointiotoksella.
7. Toistetaan kohtia 2 - 6, kunnes validaatiovirhe alkaa nousta.

3.7.1 Mallin kerrokset

Neuronikerros

Neuronikerros koostuu neuroneista, jotka käsittelevät edelliseltä kerrokselta saamansa syötteen ja lähettävät sen eteenpäin seuraavalle kerrokselle. Synapsit kytkevät eri neuronikerrokset toisiinsa ja niille määritellään kytkentäkohtainen paino, jolla kerroksien välillä kulkeva syöte kerrotaan. Neuronikerrokselta ulostuleva syöte on vektori, jonka pituus on kerroksella olevien neuronien lukumäärä. Neuronin i syöte, kun verkossa siirrytään kerrokselta k kerrokselle $k+1$:

$$x_{si}^{(k+1)} = [\boldsymbol{w}_i^{(k+1)}]^T \boldsymbol{x}_u^{(k)} + b_i^{(k+1)},$$
$$x_{ui}^{(k+1)} = f(x_{si}^{(k+1)}),$$

joissa $x_{si}^{(k+1)}$ on kerroksen $k+1$ neuronille i sisääntuleva syöte, $\boldsymbol{x}_u^{(k)}$ on kerroksen k ulostulon syöte, $[\boldsymbol{w}_i^{(k+1)}]^T$ on painovektori, jossa ovat kerroksien k ja $k+1$ välissä olevien neuroniin i kytkeytyvien synapsien painot, $b_i^{(k+1)}$ on kerroksen $k+1$ neuronin i harhaparametri, $x_{ui}^{(k+1)}$ on kerroksen $k+1$ neuronin i ulostulon syöte (output) ja f on kerroksen $k+1$ aktivaatiofunktio. (Srivastava ym. 2014.)

Otoksen normalisointikerros

Neuroverkkoa opetettaessa mallin parametrit päivitetään, kun kaikki opetusotoksen havainnot on luettu neuroverkkoon. Jokainen mallin kerros yrittää mallintaa edelliseltä kerrokselta tulevaa syötettä, joten mallin parametrien muuttuessa myös neuroverkon kerrosten aktivaatiofunktioiden arvojen jakaumat muuttuvat, mikä hidastaa mallin konvergointia. Aktivaatiofunktioiden jakaumien muuttumista kutsutaan sisäisen kovariaatin siirtymiseksi (internal covariate shift). Otoksen normalisointikerroksella (batch normalization layer) sisääntuleva syöte normalisoidaan vähentämällä siitä sen keskiarvo ja jakamalla se sen keskihajonnalla. Tämä vähentää sisääntulevan syötteen sisäisen kovariaatin siirtymistä. (Ioffe ja Szegedy 2015)

Katokerros

Katokerroksella (dropout layer) osa edelliseltä kerrokselta ulostulevasta syötteestä asetetaan nollaksi, mikä ehkäisee mallin ylisovittumista. Nollaksi asetettavan syötteen osuutta voidaan säädellä muuttamalla kadon tasoa (dropout rate). Neuronin i syöte, kun verkossa siirrytään neuronikerrokselta $k-1$ katokerroksen k kautta neuronikerrokselle $k+1$ ja kadon taso on p :

$$
\begin{aligned}
r_j^{(k)} &\sim Bernoulli(p), \\
\tilde{\boldsymbol{x}}_{\boldsymbol{ui}}{}^{(k)} &= \boldsymbol{r}^{(k)} * \boldsymbol{x}_{ui}^{(k-1)}, \\
x_{si}^{(k+1)} &= [\boldsymbol{w}_i^{(k+1)}]^T \tilde{\boldsymbol{x}}_{ui}^{(k)} + b_i^{(k+1)}, \\
x_{ui}^{(k+1)} &= f(x_{si}^{(k+1)}),
\end{aligned}
$$

jossa $\boldsymbol{r}^{(k)}$ on riippumattomien Bernoulli-jakautuneiden muuttujien vektori, jonka arvot ovat yksi todennäköisyydellä p, ja $*$ tarkoittaa alkiokohtaista tuloa. (Srivastava ym. 2014.)

ReLU (Rectified Linear Unit)

ReLU on yksinkertainen aktivaatiofunktio, jota käytetään yleensä ensimmäisen ja viimeisen kerroksen välisissä neuronikerroksissa. Se asettaa syötteen nollaa pienemmät summat nolliksi, mutta nollaa suuremmat tai yhtä suuret summat pysyvät samoina.

$$
f(x_{si}) = \begin{cases} 0, & \text{kun} \quad x_{si} < 0 \\ x_{si}, & \text{kun} \quad x_{si} \geq 0 \end{cases},
$$

$$
f'(x_{si}) = \begin{cases} 0, & \text{kun} \quad x_{si} < 0 \\ 1, & \text{kun} \quad x_{si} \geq 0 \end{cases}.
$$

Kun neuroni asetetaan nollaksi, myös sen gradientti on nolla. Eli neuronin tila ei muutu mallin päivittäessä parametreja ja neuroni niin sanotusti kuolee pois. Tässä on sekä hyvä että huono puoli. Mallin laskennallinen nopeus paranee, kun osaa neuroneista ei tarvitse käyttää. Toisaalta mallista tulee passiivinen, jos liian suuri osa neuroneista kuolee pois. (Efron ja Hastie 2016).

Softmax

Softmax on multinomisen yleistetyn lineaarisen regressiomallin kanonisen linkkifunktion käänteisfunktio. Se on aktivaatiofunktio, jota käytetään yleensä viimeisessä kerroksessa, kun vaste on moniluokkainen. Se antaa jokaiselle havainnolle todennäköisyydet kuulua kuhunkin luokkaan. Esimerkiksi ulostulokerroksen neuronille m sisääntulevan syötteen x_{sm} eksponentti jaetaan kaikille ulostulokerroksen neuroneille sisääntulleiden syötteiden $\boldsymbol{x}_s$ eksponenttien summalla:

$$f^{(K)}(x_{sm}^{(K)}; \boldsymbol{x}_s^{(K)}) = \frac{e^{x_{sm}^{(K)}}}{\sum_{l=1}^{M} e^{x_{sl}^{(K)}}},$$

jossa M on ulostulokerroksen neuroneiden lukumäärä, $k \in \{1, \ldots, K\}$ on neuroverkon kerroksien lukumäärä (Efron ja Hastie 2016).
Tämä tehdään kaikille neuroneille, jolloin saadaan todennäköisyysjakauma. Ulostulokerroksen neuroneiden lukumäärän M pitää olla sama kuin luokkien lukumäärä. Tällöin neuronin m saama todennäköisyys kertoo todennäköisyyden kuulua luokkaan m.

3.7.2 Laskennalliset menetelmät

Kategorinen ristientropia

Kategorinen ristientropia (categorical cross-entropy loss) on tappiofunktio, joka mittaa sellaisen luokittelumallin suoritusta, jonka ulostulo on todennäköisyys eli ulostulo saa arvoja välillä $[0, 1]$. Kategorinen ristientropia määritellään seuraavasti, kun luokkien lukumäärä M on suurempi kuin kaksi:

$$tappio = \frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} \mathbf{I}_{\boldsymbol{x}_i,m} \log(p_{\boldsymbol{x}_i,m}),$$

jossa $\mathbf{I}$ on binäärinen indikaattori $\mathbf{I} = \begin{cases} 1, & \text{kun} \quad \boldsymbol{x}_i \notin m \\ 0, & \text{kun} \quad \boldsymbol{x}_i \in m \end{cases}$, $p_{\boldsymbol{x}_i,m}$ on ennustettu todennäköisyys, että $\boldsymbol{x}_i \in m$, M on luokkien lukumäärä, $\boldsymbol{x}_i$ on havainto i, N havaintojen lukumäärä ja m on luokka. (de Boer et al. 2005.)
Kategorinen ristientropia kasvaa, kun ennustettu luokka poikkeaa oikeasta luokasta ja rankaisee erityisesti ennusteita, jotka ovat varmoja (suuri todennäköisyys) mutta vääriä.

Adam (Adaptive Moment Estimation)

Adam on optimaatiofunktio, joka yhdistää ideoita AdaGrad- ja RMSprop-optimaatiofunktioista (Chen, viitattu 26.3.2019). Adam laskee mukautuvan oppimisvauhdin (adaptive learning rate) jokaiselle parametrille.
Kingma ja Ba (2014) esittävät Adamin näin: Olkoon mallin iteraatiokerta $t \in \{1, \ldots, T\}$. Kaikki vektorien väliset operaatiot ovat alkiokohtaisia.

1. Lasketaan tappiofunktion gradientit mallin parametrien suhteen:

$$\boldsymbol{g}_t = \bigtriangledown_{\boldsymbol{\theta}} f_t(\boldsymbol{\theta}_{t-1}),$$

jossa $\boldsymbol{g}_t$ on parametrien gradienttivektori iteraatiokerralla t, $\boldsymbol{\theta}_{t-1}$ on mallin parametrivektori iteraatiokerralla $t-1$ ja f_t on tappiofunktio iteraatiokerralla t.

2. Päivitetään gradienttien ensimmäisten momenttien harhaiset estimaatit:

$$\boldsymbol{m}_t = \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t,$$

jossa $\boldsymbol{m}_t$ on harhainen gradienttien ensimmäisten momenttien estimaattivektori iteraatiokerralla t, $\beta_1 \in [0,1)$ on säädettävä hyperparametri ja $\boldsymbol{m}_{t-1}$ on harhainen gradienttien ensimmäisten momenttien estimaattivektori iteraatiokerralla $t-1$.

3. Päivitetään gradienttien toisten momenttien harhaiset estimaatit:

$$\boldsymbol{v}_t = \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2,$$

jossa $\boldsymbol{v}_t$ on harhainen gradienttien toisten momenttien estimaattivektori iteraatiokerralla t, $\beta_2 \in [0,1)$ on säädettävä hyperparametri ja $\boldsymbol{v}_{t-1}$ on harhainen gradienttien toisten momenttien estimaattivektori iteraatiokerralla $t-1$.

4. Lasketaan gradienttien päivitettyjen ensimmäisten momenttien harhattomat estimaatit:

$$\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m}_t}{1-\beta_1^t},$$

jossa $\hat{\boldsymbol{m}}_t$ on gradienttien päivitettyjen ensimmäisten momenttien harhaton estimaattivektori iteraatiokerralla t.

5. Lasketaan gradienttien päivitettyjen toisten momenttien harhattomat estimaatit:

$$\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v}_t}{1-\beta_2^t},$$

jossa $\hat{\boldsymbol{v}}_t$ on gradienttien päivitettyjen toisten momenttien harhaton estimaattivektori iteraatiokerralla t.

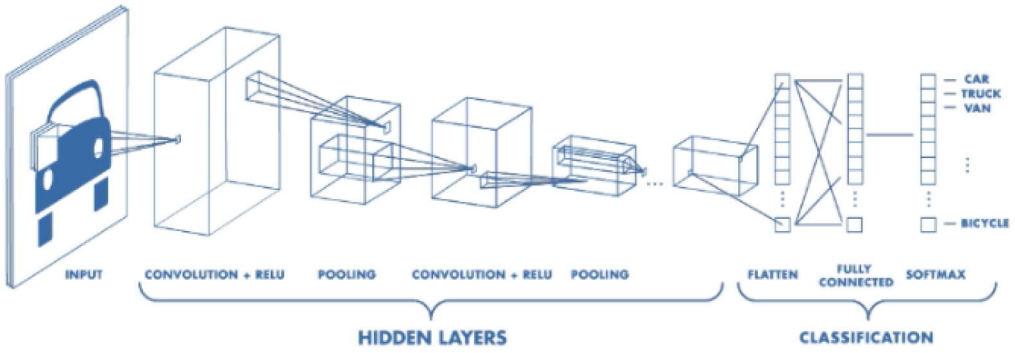
6. Päivitetään mallin parametrit:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \frac{\alpha \hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon},$$

jossa $\boldsymbol{\theta}_t$ on mallin parametrivektori iteraatiokerralla t, α on oppimisvauhti ja $\epsilon > 0$ on erittäin pieni lähellä nollaa oleva positiivinen luku, joka estää nollalla jakamisen.

3.8 Konvoluutioneuroverkko

Konvoluutioneuroverkkoa (Convolutional Neural Network, CNN) voidaan käyttää esimerkiksi ajoneuvojen luokitteluun ajoneuvoista otettujen kuvien perusteella (Kuva 8). Konvoluutioneuroverkossa sisääntuleva kuva käsitellään yhdellä tai useammalla konvoluutio- ja yhdistämiskerroksella. Sen jälkeen se siirretään vektorisointikerroksen kautta neuronikerrokselle, jonka jälkeen konvoluutioneuroverkko toimii kuten tavallinen neuroverkko.



Kuva 8: Esimerkki konvoluutioneuroverkosta, jossa on kaksi konvoluutiokerrosta (kuva kopioitu The MathWorks, viitattu 6.2.2019).

3.8.1 Mallin kerrokset

Konvoluutiokerros

Konvoluutiokerrokselle (convolutional layer) sisääntuleva kuva konvoloidaan käyttämällä suodattimia (filter), joiden koko ja lukumäärä on ennalta määrätty. Olkoon x kuva, joka esitetään $k \times k$ matriisina ja f suodatin, joka on $q \times q$ matriisi, jossa $q < k$. Tällöin konvoloitu kuva on $k \times k$ matriisi $\tilde{\boldsymbol{x}}$, jossa $\tilde{x}_{i,j} = \sum_{l=1}^{q} \sum_{l'=1}^{q} x_{i+l,j+l'} f_{l,l'}$. Konvoluutiokerros laskee konvoluution käyttämällä p erillistä $q \times q \times e$ suodatinta. Suodattimet tuottavat kuvien järjestetyn joukon (array), jonka ulottuvuus on $p \times k \times k$. Tässä e on väriulottuvuus eli esimerkiksi $e = 3$ (punainen, vihreä ja sininen). (Efron ja Hastie 2016.)

Yhdistämiskerros

Yhdistämiskerroksella (pooling layer) pienennetään konvoloitua kuvaa korvaamalla sen jokaisen ruudun jokainen ei-päällekkäinen $k \times k$ -kokoinen lohko sen suurimmalla arvolla (Efron ja Hastie 2016). Olkoon yhdistämiskerrokselle sisääntulevan konvoloidun kuvan

osa

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}.$$

Ennalta on päätetty, että kuva jaetaan 2×2 -kokoisiin lohkoihin. Jokainen lohko korvataan sen suurimmalla arvolla, joten yhdistämiskerrokselta lähtevä vastaava konvoloidun kuvan osa on

$$\begin{bmatrix} 6 & 8 \\ 14 & 16 \end{bmatrix}.$$

Vektorisointikerros

Vektorisointikerros (flatten layer) muuttaa syötteen matriisimuodosta vektorimuotoon, jolloin se voidaan siirtää neuronikerrokselle (Efron ja Hastie 2016).

3.9 Glmnet

Glmnet (Friedman, Hastie ja Tibshirani 2010) on tilastollisen R-ohjelmointikielen paketti, jolla aineistoon voidaan sovittaa yleistetty lineaarinen malli. Mallin sovituksessa käytetään optimaatiofunktiota, joka minimoi suurimman uskottavuuden sakkotermin. Sakkotermiksi voidaan valita lasso-, ridge- tai elastic-net-sakkotermi, joka on lasso- ja ridge-sakkotermien yhdistelmä. Analyysissä käytettiin multinomiaalista logistista regressiomallia ja lasso-sakkotermiä. Multinomiaaliselle logistiselle regressiomallille voidaan valita joko lasso-sakkotermi jokaiselle parametrille erikseen tai jokaiselle luokalle erikseen. Analyysissä käytettiin lasso-sakkotermiä jokaiselle parametrille erikseen, koska näin sovitetut mallit olivat tarkempia ja niiden selitetyn devianssin osuus oli suurempi. Lassoa käyttämällä saatiin myös tietää, mitkä parametrit ovat mallin kannalta tärkeimpiä, sillä lasso valitsee malliin vain tietyt parametrit ja asettaa muut parametrit nolliksi.

3.9.1 Malli

Olkoon $Y \in \{1, 2, \ldots, M\}$ luokat, joihin esimerkiksi mittaukset halutaan luokitella. Todennäköisyydet voidaan esittää log-lineaarisesti

$$Pr(Y = m|X = x) = \frac{e^{\beta_{0m} + \boldsymbol{\beta}_m^T \boldsymbol{x}}}{\sum_{i=1}^{M} e^{\beta_{0i} + \boldsymbol{\beta}_i^T \boldsymbol{x}}}.$$

Havainnoille $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ voidaan kirjoittaa negatiivisen log-uskottavuuden regularisoitu muoto

$$l(\{\beta_{0m}, \boldsymbol{\beta}_m\}_{m=1}^M) = -\frac{1}{N}\sum_{i=1}^{N} \log Pr(Y = y_i|\boldsymbol{x}_i; \{\beta_{0m}, \boldsymbol{\beta}_m\}_{m=1}^M) + \lambda \sum_{m=1}^{M} ||\boldsymbol{\beta}_m||_1,$$

jossa $\boldsymbol{\beta}_m$ luokan m kerroinvektori, ja $\lambda \sum_{m=1}^{M} ||\boldsymbol{\beta}_m||_1$ on lasso-sakkotermi jokaiselle parametrille erikseen.

Olkoon $\boldsymbol{R}$ $N \times M$ indikaattorivastematriisi, jossa $r_{im} = I(l_i = m)$. Tällöin negatiivinen log-uskottavuus saadaan muotoon

$$l(\{\beta_{0m}, \boldsymbol{\beta}_m\}_1^M) = -\frac{1}{N}\sum_{i=1}^{N}[\sum_{m=1}^{M} r_{im}(\beta_{0m} + \boldsymbol{\beta}_m^T \boldsymbol{x}_i) - \log(\sum_{m=1}^{M} e^{\beta_{0m} + \boldsymbol{\beta}_m^T \boldsymbol{x}_i})] + \lambda \sum_{m=1}^{M} ||\boldsymbol{\beta}_m||_1.$$

3.9.2 Laskennalliset menetelmät

Glmnet käyttää multinomiaalisessa logistisessa regressiomallissa cyclic coordinate descent -algoritmiä (CCD) minimoidessaan lasso-mallin mukaista (negatiivista) uskottavuuslauseketta. CCD:ssä kierretään (cycle) toistuvasti ennustavien muuttujien yli jossain vakioidussa mutta mielivaltaisessa järjestyksessä, jossa j:nnessä kohdassa päivitetään luokan j parametrit ja pidetään muiden luokkien parametrit vakioina sen hetkisissä arvoissaan. Päivitettäessä parametreja $(\beta_{0j}, \boldsymbol{\beta}_j)$, muodostetaan neliöllinen funktio

$$\boldsymbol{Q}_j(\beta_{0j}, \boldsymbol{\beta}_j) = -\frac{1}{2N}\sum_{i=1}^{N} w_{ij}(g_{ij} - \beta_{0j} - \boldsymbol{\beta}_j^T \boldsymbol{x}_i)^2 + C(\{\tilde{\beta_{0m}}, \tilde{\beta}_m\}_{m=1}^M),$$

jossa C on $(\beta_{0j}, \boldsymbol{\beta}_j)$:stä riippumaton vakio, $g_{ij} = \tilde{\beta_{0j}} + \tilde{\beta_j}^T \boldsymbol{x}_i + \frac{r_{ij} - \tilde{p_j}(\boldsymbol{x}_i)}{\tilde{p_j}(\boldsymbol{x}_i)(1 - \tilde{p_j}(\boldsymbol{x}_i))}$ ja $w_{ij} = \tilde{p_j}(x_i)(1 - \tilde{p_j}(x_i))$, jossa $\tilde{p_j}(\boldsymbol{x}_i)$ on ehdollisen todennäköisyyden $Pr(Y = j|\boldsymbol{x}_i)$ sen hetkinen estimaatti.

Jokaiselle λ:n arvolle luodaan silmukka (loop), joka kiertää arvojen $j \in \{1, .., M\}$ yli ja laskee osittaisen neliöllisen approksimaation Q_j:lle sen hetkisillä parametreilla $(\tilde{\beta_0}, \tilde{\beta})$: $Q(\beta_{0j}, \boldsymbol{\beta}_j)$. Sen jälkeen käytetään coordinate descent -algoritmiä (CD) minimoimaan painotettu lasso

$$minimize_{(\beta_{0j}, \boldsymbol{\beta}_j) \in \mathbb{R}^{p+1}}\{Q(\beta_{0j}, \boldsymbol{\beta}_j) + \lambda||\boldsymbol{\beta}_j||_1\}.$$

CD on erityisen nopea lassolle, koska kunkin koordinaattiakselin suuntaiselle minimoijalle on käytettävissä eksplisiittinen lauseke. Tällöin ei tarvita iteratiivista hakua jokaista koordinaattia pitkin. Se käyttää myös hyväkseen ongelman parametriavaruuden harvuutta (sparsity). Tarpeeksi suurilla λ arvoilla suurin osa kertoimista on nollia, eikä

niitä tulla liikuttamaan nollasta. (Hastie ja Qian 2014; Hastie, Tibshirani ja Wainwright 2015; Efron ja Hastie 2016.)

3.10 Yleistetty pienimmän neliösumman menetelmä

Yleistettyä pienimmän neliösumman menetelmää (generalized least squares, GLS) voidaan käyttää regressiokertoimien estimoimiseen lineaarisessa mallissa, vaikka jäännökset olisivat korreloituneita.

Olkoon $\boldsymbol{a}$ $n \times 1$ vastevektori, jonka odotusarvo $E(\boldsymbol{a}) = \boldsymbol{X}\beta$; $\boldsymbol{X}$ $n \times p$ kovariaattimatriisi, jonka ensimmäinen sarake on selittävien muuttujien vektori ja loput $p-1$ saraketta voivat esimerkiksi olla ensimmäisen sarakkeen muunnoksia; $\boldsymbol{\beta}$ $p \times 1$ regressiokertoimien vektori ja $\boldsymbol{e}$ $n \times 1$ satunnaisvirheiden vektori, jonka odotusarvo $E(\boldsymbol{e}) = 0$. GLS:ssä ei tarvitse tehdä jakaumaoletusta satunnaisvirheille.

Olkoon satunnaisvirheiden $\boldsymbol{e}$ varianssi-kovarianssi-matriisi $Cov(\boldsymbol{e}) = \boldsymbol{\Sigma}$. GLS:ssä minimoidaan $(\boldsymbol{a} - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{a} - \boldsymbol{X}\boldsymbol{\beta})$ $\boldsymbol{\beta}$:n suhteen. Olkoon varianssi-kovarianssi-matriisi $\boldsymbol{\Sigma}$ tunnettu. Tällöin regressiokertoimien $\boldsymbol{\beta}$ estimaattori on

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{a}$$

ja $\hat{\boldsymbol{\beta}}$:n estimoitu kovarianssimatriisi on

$$\hat{Cov}(\hat{\boldsymbol{\beta}}) = (\boldsymbol{X}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{X})^{-1}.$$

$\hat{\boldsymbol{\beta}}$ on $\boldsymbol{\beta}$:n harhaton estimaatti riippumatta $\boldsymbol{\Sigma}$:n valinnasta, eli $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$. (Orsini ym. 2006.)

3.11 Šidákin menetelmä

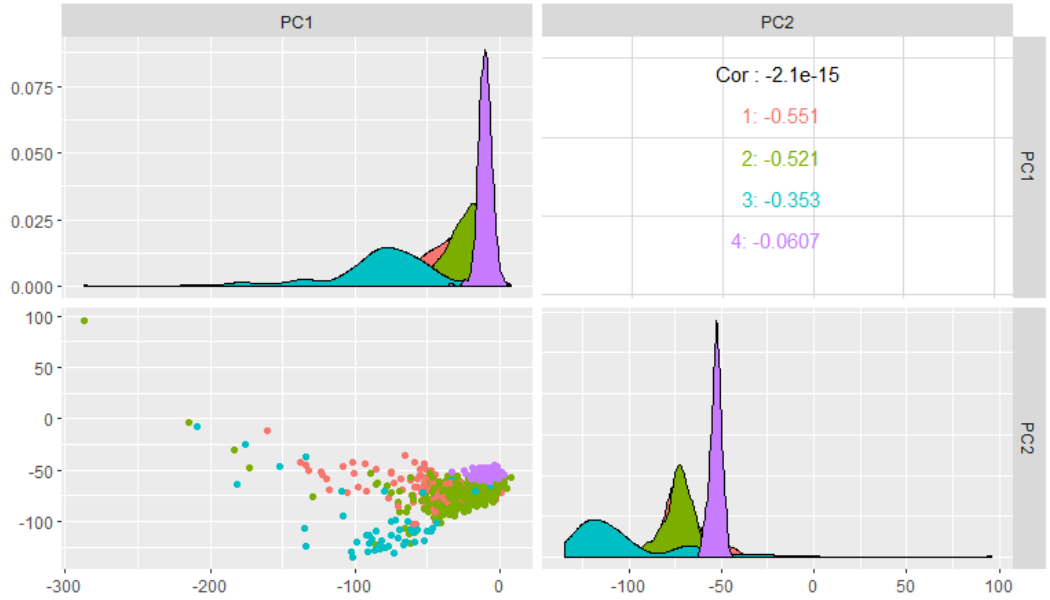
Tyypin I -virheellä tarkoitetaan nollahypoteesin hylkäämistä, vaikka se olisi tosi. Tyypin I -virheen todennäköisyyttä kutsutaan merkitsevyystasoksi. Kun tarkastellaan monta lineaarisen mallin lineaarikombinaatiota samanaikaisesti, yksittäisen lineaarikombinaation sijasta merkitsevyystaso liitetään koko tarkasteltavien lineaarikombinaatioiden joukkoon. Tarkasteluiden samanaikaisuus voidaan ottaa huomioon esimerkiksi Šidákin menetelmällä:

$$\alpha_c = 1 - (1 - \alpha)^{\frac{1}{j}},$$

jossa α_c on yksittäisen tarkastelun korjattu merkitsevyystaso, α on alunperin valittu merkitsevyystaso ja j on tarkasteluiden lukumäärä (Šidák 1967). Šidákin menetelmällä voidaan varmistaa, että tarkasteltavien lineaarikombinaatioiden joukkoon liityvä merkitsevyystaso on pienempi tai yhtä suuri kuin valittu merkitsevyystaso. (Grönroos 2003.)

4 Aineiston kuvailu ohjaamattomilla menetelmillä

Kaikille aineistoille tehtiin pääkomponenttianalyysit, joissa yritettiin pääkomponenttien avulla löytää eroja luokkien välillä. H-aineiston havainnot skaalattiin jakamalla ne havaintojen keskihajonnalla. S- ja SP-aineistojen havainnot skaalattiin jo aineiston esikäsittelyvaiheessa. Kuvasta 9 nähdään, miten ensimmäisen ja toisen pääkomponentin avulla nähdään eroja luokkien välillä. Kaksi ensimmäistä pääkomponenttia selittivät kuitenkin melko vähän kokonaisvarianssista (Taulukko 3). S- ja SP-aineistoissa oli selkeää erottuvuutta luokkien välillä. H-aineistossa ei näkynyt selkeää eroa eri luokkien välillä.



Kuva 9: S-aineiston ensimmäinen ja toinen pääkomponentti. Keskikokoiset-luokka on merkitty punaisella, kevyet-luokka vihreällä, raskaat-luokka sinisellä ja tyhjät-luokka violetilla. Kuvan oikeassa yläkulmassa näkyvät ensimmäisen ja toisen pääkomponentin välinen korrelaatio koko aineistossa sekä kussakin luokassa.

Taulukko 3: Ensimmäisen ja toisen pääkomponentin selittämä osuus varianssista eri aineistoilla ja niiden muuttujien sijainti, joilla on suurimmat lataukset. Suurimpien latausten sijainti kertoo niiden muuttujien sijainnin, jotka selittävät suurimman osan varianssista.

Aineisto	**1. PK (%)**	**2. PK (%)**	**Suurimpien latausten sijainti**
H	5,75	5,06	22 443 - 24 008
S	11,76	4,16	63 - 142
SP	11,76	4,16	62 - 143

5 Analyysi

Analyysissä pyrittiin luokittelemaan mittaukset oikeisiin ajoneuvoluokkiin ja tunnistamaan ne taajuudet, joita menetelmät käyttävät luokittelussa. Luokittelussa käytetyt menetelmät olivat knn, rf, CNN, NN, glmnet, NN Borutalla valituille muuttujille ja NN glmnetin valitsemille muuttujille. Tärkeiden taajuuksien tunnistuksessa käytetyt menetelmät olivat Boruta, glmnet, CNN- ja NN-mallit, joissa osa spektristä korvattiin nollilla, ja NN-mallin painojen tarkastelu. Jako opetus- ja testiaineistoihin oli kaikissa analyyseissä 80%/20%. Kaikissa neuroverkkomalleissa opetusaineiston jako opetus- ja validointiaineistoihin oli myös 80%/20%.

5.1 Luokittelu

5.1.1 Mallit

k:n lähimmän naapurin menetelmä

Kokeiltiin knn-menetelmää H-, S- ja SP-aineistoille. Naapurien lukumäärä k valittiin ristiinvalidoinnilla, jossa kriteerinä oli luokitteluvirhe. H-aineistolle saatiin näin $k = 3$, S-aineistolle $k = 7$ ja SP-aineistolle $k = 7$.

Satunnaismetsä

Luokiteltiin H-, S- ja SP-aineistojen mittaukset satunnaismetsä-algoritmillä. Luokitteluun käytettiin 64 päätöspuuta.

Konvoluutioneuroverkko

Sovitettiin kaikille aineistoille CNN-mallit. Konvoluutiokerroksien oletettiin löytävän aineistosta tärkeitä piirteitä, jotka sitten syötettäisiin neuronikerroksille. Malleja rakennettaessa kokeiltiin eri kokoisia kerroksia ja vaihdeltiin kerrosten lukumäärää. Erilaisia rakenteita tarkasteltaessa valintakriteereinä olivat mallien sovittuminen ja tarkkuus. Lopulta päädyttiin malleihin, joissa oli kolme konvoluutiokerrosta ja yksi neuronikerros (Taulukot 4 - 6). Mallit, joissa on vain yksi neuronikerros, voidaan mieltää ensimmäisen asteen lineaarisen mallin epälineaarisina yleistyksinä. Sovitetut CNN-mallit voidaan siis ymmärtää siten, että konvoluutiokerroksien poimimat piirteet syötettiin malliin, joka on ensimmäisen asteen lineaarisen mallin epälineaarinen yleistys. Mallien hyvyyden arviointikriteereinä olivat sovittuminen, kategorinen tarkkuus ja tappiofunktion arvo. Mallien aktivaatiofunktioina käytettiin ReLU- ja softmax-funktioita, optimaatiofunktiona Adam-funktiota ja tappiofunktiona kategorinen ristientropia -funktiota.

Taulukko 4: H-aineistolle sovitetun CNN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan matriisin rivien ja sarakkeiden lukumäärää tai vektorin pituutta. Suodattimien pituus on ensimmäisellä kerroksella 20, toisella 5 ja kolmannella 5.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	32 000 x 1	0	-	
Konvoluutio	31 981 x 4	84	-	4 suodatinta
Yhdistämis	3 198 x 4	0	-	
Konvoluutio	3 194 x 4	84	-	4 suodatinta
Yhdistämis	638 x 4	0	-	
Konvoluutio	634 x 4	84	-	4 suodatinta
Yhdistämis	126 x 4	0	-	
Vektorisointi	504	0	-	
Otoksen normalisointi	504	2 016	-	
Kato	504	0	-	taso 0,5
Neuroni	200	101 000	ReLU	
Ulostulo	4	804	Softmax	

Taulukko 5: S-aineistolle sovitetun CNN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan matriisin rivien ja sarakkeiden lukumäärää tai vektorin pituutta. Suodattimien pituus on ensimmäisellä kerroksella 20, toisella 5 ja kolmannella 5.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	16 000 x 1	0	-	
Konvoluutio	15 981 x 4	84	-	4 suodatinta
Yhdistämis	1 598 x 4	0	-	
Konvoluutio	1 594 x 4	84	-	4 suodatinta
Yhdistämis	318 x 4	0	-	
Konvoluutio	314 x 4	84	-	4 suodatinta
Yhdistämis	62 x 4	0	-	
Vektorisointi	248	0	-	
Otoksen normalisointi	248	992	-	
Kato	248	0	-	taso 0,5
Neuroni	200	49 800	ReLU	
Ulostulo	4	804	Softmax	

Taulukko 6: SP-aineistolle sovitetun CNN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan matriisin rivien ja sarakkeiden lukumäärää tai vektorin pituutta. Suodattimien pituus on ensimmäisellä kerroksella 20, toisella 5 ja kolmannella 5.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	16 001 x 1	0	-	
Konvoluutio	15 982 x 4	84	-	4 suodatinta
Yhdistämis	1 598 x 4	0	-	
Konvoluutio	1 594 x 4	84	-	4 suodatinta
Yhdistämis	318 x 4	0	-	
Konvoluutio	314 x 4	84	-	4 suodatinta
Yhdistämis	62 x 4	0	-	
Vektorisointi	248	0	-	
Otoksen normalisointi	248	992	-	
Kato	248	0	-	taso 0,5
Neuroni	100	24 900	ReLU	
Ulostulo	4	404	Softmax	

Neuroverkko

Kokeiltiin, paranevatko mallit, kun jätetään konvoluutio-osuus pois. Eli sovitettiin kaikille aineistoille NN-mallit. Mallien rakenteet valittiin samalla tavalla kuin CNN-mallien, mutta ilman konvoluutio-osuutta. Päädyttiin malleihin, joissa oli vain yksi neuronikerros (Taulukot 7 - 9). Aktivaatiofunktioina käytettiin ReLU- ja softmax-funktioita, optimaatiofunktiona Adam-funktiota ja tappiofunktiona kategorinen ristientropia -funktiota. Poistamalla konvoluutio-osuus saatiin parametrien lukumäärän kasvusta huolimatta nopeammin suorittavia malleja. H-aineistolle sovitettu malli ylisovittui, mutta esimerkiksi katokerroksen lisääminen huononsi sen tarkkuutta. Sen kategorinen tarkkuus validointiotoksella parani nollasta vasta, kun kategorinen tarkkuus opetusotoksella oli yli 0,90.

Taulukko 7: H-aineistolle sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	32 000	0	-
Neuroni	150	4 800 150	ReLU
Otoksen normalisointi	150	600	-
Ulostulo	4	604	Softmax

Taulukko 8: S-aineistolle sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	16 000	0	-
Neuroni	150	2 400 150	ReLU
Otoksen normalisointi	150	600	-
Ulostulo	4	604	Softmax

Taulukko 9: SP-aineistolle sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	16 001	0	-
Neuroni	150	2 400 300	ReLU
Otoksen normalisointi	150	600	-
Ulostulo	4	604	Softmax

Glmnet

Mallinnettiin aineistoja glmnetin multinomiaalisella logistisella regressiomallilla, jossa oli sakkoterminä lasso jokaiselle parametrille erikseen. λ valittiin ristiinvalidoinnilla, jossa kriteerinä oli luokitteluvirhe. Kaikissa aineistoissa päädyttiin λ:n arvoon 0,04.

Neuroverkko Borutalla valituille muuttujille

Sovitettiin NN-malli H-, S- ja SP-aineistojen muuttujille, jotka olivat Boruta-algoritmin mukaan luokittelun kannalta tärkeitä muuttujia. Mallien rakenteet valittiin samalla tavalla kuin aikaisempien CNN- ja NN-mallien rakenteet. Päädyttiin malleihin, joissa oli kaksi neuronikerrosta (Taulukot 10 - 12). Kahden neuronikerroksen mallit voidaan mieltää toisen asteen lineaarisen mallin epälineaarisina yleistyksinä. Aktivaatiofunktiona käytettiin Adamia ja tappiofunktiona kategorista ristientropiaa.

Taulukko 10: Borutalla valituille H-aineiston muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	154	0	-
Neuroni	154	23 870	ReLU
Otoksen normalisointi	154	616	-
Neuroni	150	23 250	ReLU
Otoksen normalisointi	150	600	-
Ulostulo	4	604	Softmax

Taulukko 11: Borutalla valituille S-aineiston muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	169	0	-	
Neuroni	169	28 730	ReLU	
Kato	169	0	-	taso 0,3
Otoksen normalisointi	169	676	-	
Neuroni	100	17 000	ReLU	
Otoksen normalisointi	100	400	-	
Ulostulo	4	404	Softmax	

Taulukko 12: Borutalla valituille SP-aineiston muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	166	0	-
Neuroni	166	27 722	ReLU
Otoksen normalisointi	166	664	-
Neuroni	166	27 722	ReLU
Otoksen normalisointi	166	664	-
Ulostulo	4	668	Softmax

Neuroverkko glmnetin valitsemille muuttujille

Sovitettiin NN-mallit glmnet-mallin käyttämille H-, S- ja SP-aineistojen muuttujille, joiden oletettiin olevan luokittelun kannalta tärkeimmät muuttujat. Mallien rakenteet valittiin samalla tavalla kuin aikaisempien CNN- ja NN-mallien rakenteet. Päädyttiin malleihin, joissa oli yksi neuronikerros (Taulukot 13 - 15). Aktivaatiofunktiona käytettiin Adamia ja tappiofunktiona kategorista ristientropiaa.

Taulukko 13: H-aineiston glmnet-mallin käyttämille muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	76	0	-	
Neuroni	51	3 927	ReLU	
Otoksen normalisointi	51	204	-	
Kato	51	0	-	taso 0,2
Ulostulo	4	208	Softmax	

Taulukko 14: S-aineiston glmnet-mallin käyttämille muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	25	0	
Neuroni	25	650	ReLU
Otoksen normalisointi	25	100	
Ulostulo	4	104	Softmax

Taulukko 15: SP-aineiston glmnet-mallin käyttämille muuttujille sovitetun NN-mallin rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan vektorin pituutta.

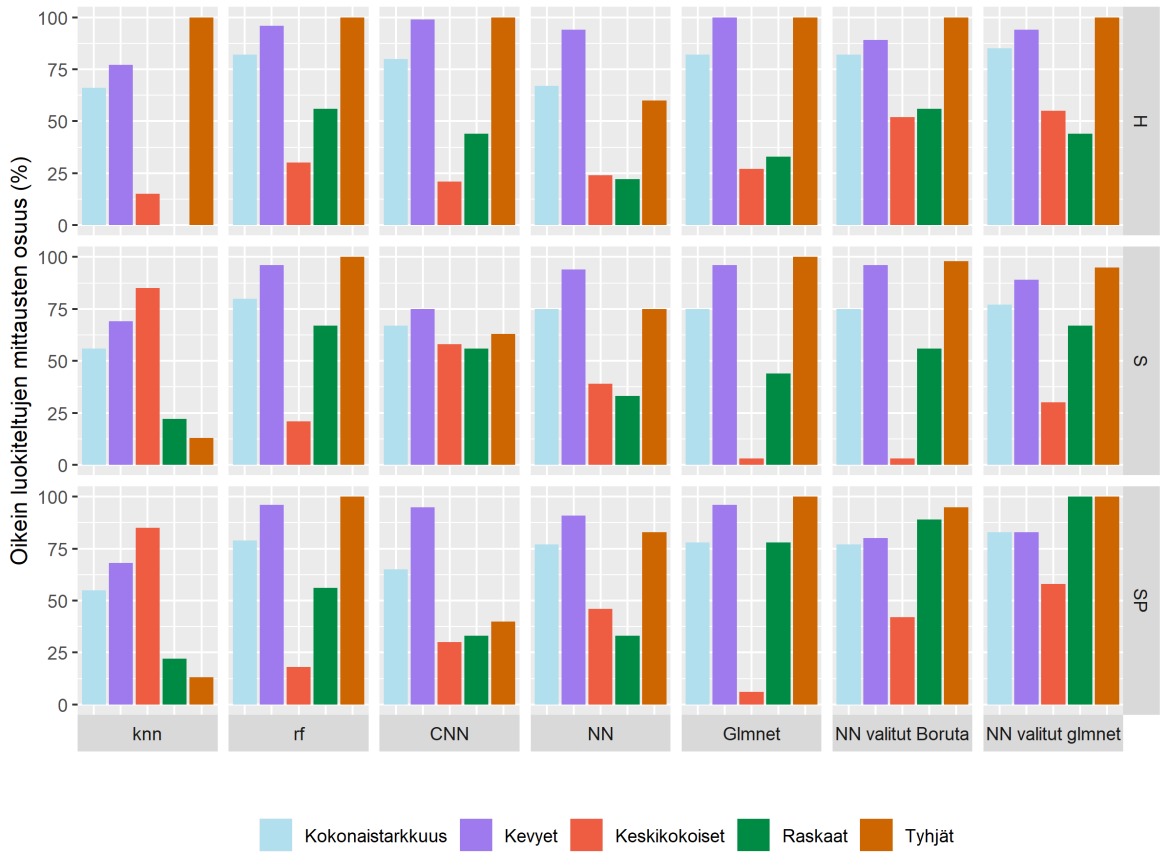
Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**
Sisääntulo	32	0	
Neuroni	30	990	ReLU
Otoksen normalisointi	30	120	
Ulostulo	4	124	Softmax

5.1.2 Tulokset

Esitellään luokittelussa eri menetelmillä saadut tulokset H-, S- ja SP-aineistoille. Taulukossa 16 ja Kuvassa 10 näkyvät mallien kategoriset tarkkuudet sekä kuinka hyvin mallit tunnistivat eri ajoneuvoluokat.

Taulukko 16: Aineistoille sovitettujen mallien tarkkuus sekä oikein luokiteltujen mittausten prosenttiosuus. Kevyet % kertoo, mikä prosenttiosuus kevyiksi ennustetuista mittauksista on oikeasti kevyitä. Vastaavasti Keskikokoiset %, Raskaat % ja Tyhjät % kertovat niitä vastaavien luokkien oikein luokiteltujen mittausten prosenttiosuuden. NN valitut Boruta tarkoittaa neuroverkkoa Borutalla valituille muuttujille ja NN valitut glmnet tarkoittaa neuroverkkoa glmnetin käyttämille muuttujille.

Malli/aineisto	**Tarkkuus (%)**	**Kevyet (%)**	**Keskikokoiset (%)**	**Raskaat (%)**	**Tyhjät (%)**
knn					
H	66	77	15	0	100
S	56	69	85	22	13
SP	55	68	85	22	13
rf					
H	82	96	30	56	100
S	80	96	21	67	100
SP	79	96	18	56	100
CNN					
H	80	99	21	44	100
S	67	75	58	56	63
SP	65	95	30	33	40
NN					
H	67	94	24	22	60
S	75	94	39	33	75
SP	77	91	46	33	83
Glmnet					
H	82	100	27	33	100
S	75	96	3	44	100
SP	78	96	6	78	100
NN valitut Boruta					
H	82	89	52	56	100
S	75	96	3	56	98
SP	77	80	42	89	95
NN valitut glmnet					
H	85	94	55	44	100
S	77	89	30	67	95
SP	83	83	58	100	100



Kuva 10: Taulukon 16 tulokset.

Malleista suurin kategorinen tarkkuus oli glmnet-mallin valitsemille H-aineiston muuttujille sovitetulla neuroverkolla. Se tunnisti hyvin kevyet- ja tyhjät-luokat, mutta huonosti keskikokoiset- ja raskaat-luokat. Toiseksi suurin kategorinen tarkkuus oli glmnet-mallin valitsemille SP-aineiston muuttujille sovitetulla neuroverkolla. Se tunnisti hyvin kevyet-, raskaat- ja tyhjät-luokat, mutta huonosti keskikokoiset-luokan. Yleisesti keskikokoiset- ja raskaat-luokat tunnistettiin huonosti ja kevyet- ja tyhjät-luokat hyvin.

5.2 Tärkeiden taajuuksien tunnistus

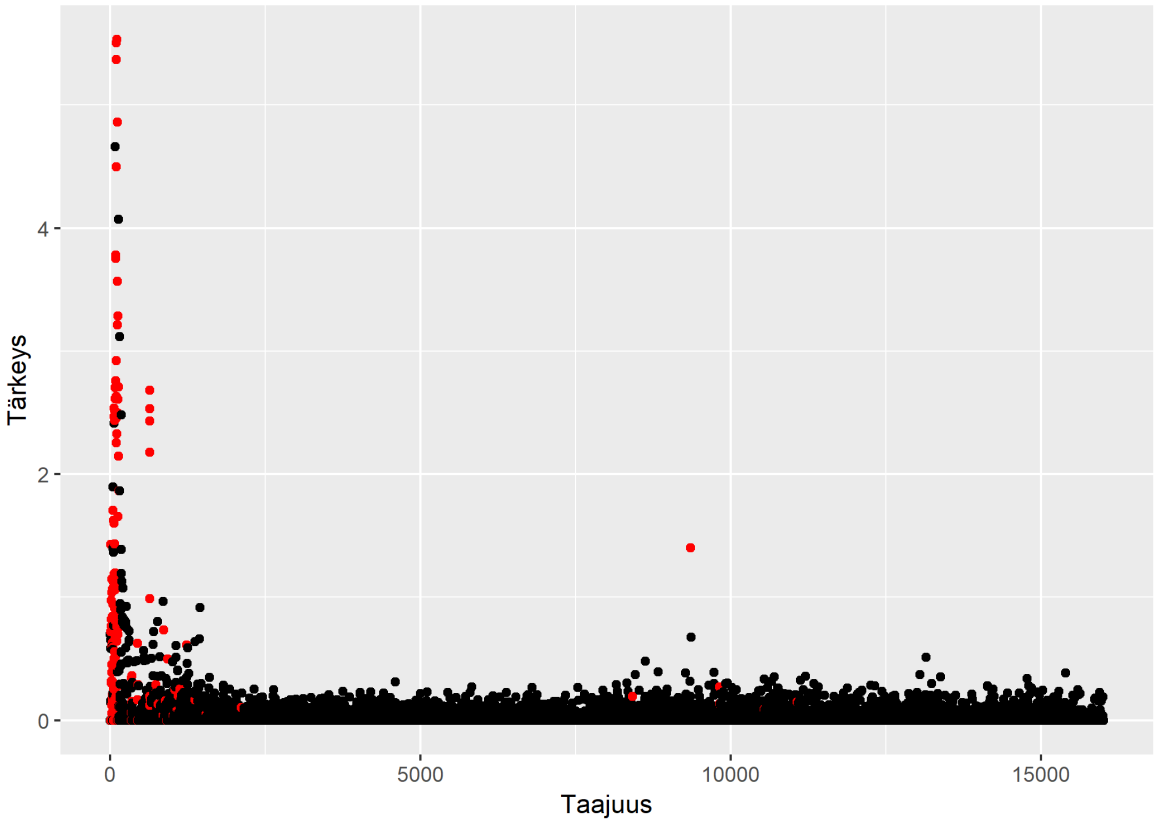
Tärkeiden taajuuksien tunnistukseen käytettiin S-aineistoa, koska se käsittää pelkästään värähtelymittausten periodogrammit. Spektriaineiston (S) muuttujia kutsutaan taajuuksiksi, koska sen muuttujat vastaavat taajuuksia. Esimerkiksi jos S-aineiston muuttuja 50 on tärkeä luokittelulle, se tarkoittaa, että 50 Hz taajuus on tärkeä luokittelulle. Tärkeiden taajuuksien tunnistukseen olisi vaihtoehtoisesti voitu käyttää SP-aineistoa. SP-aineisto käsittää kuitenkin värähtelymittausten periodogrammien lisäksi piikkien lukumäärän, joka ei liity taajuuksien tunnistukseen. Tästä syystä sen

käytöstä luovuttiin.

5.2.1 Menetelmät

Boruta

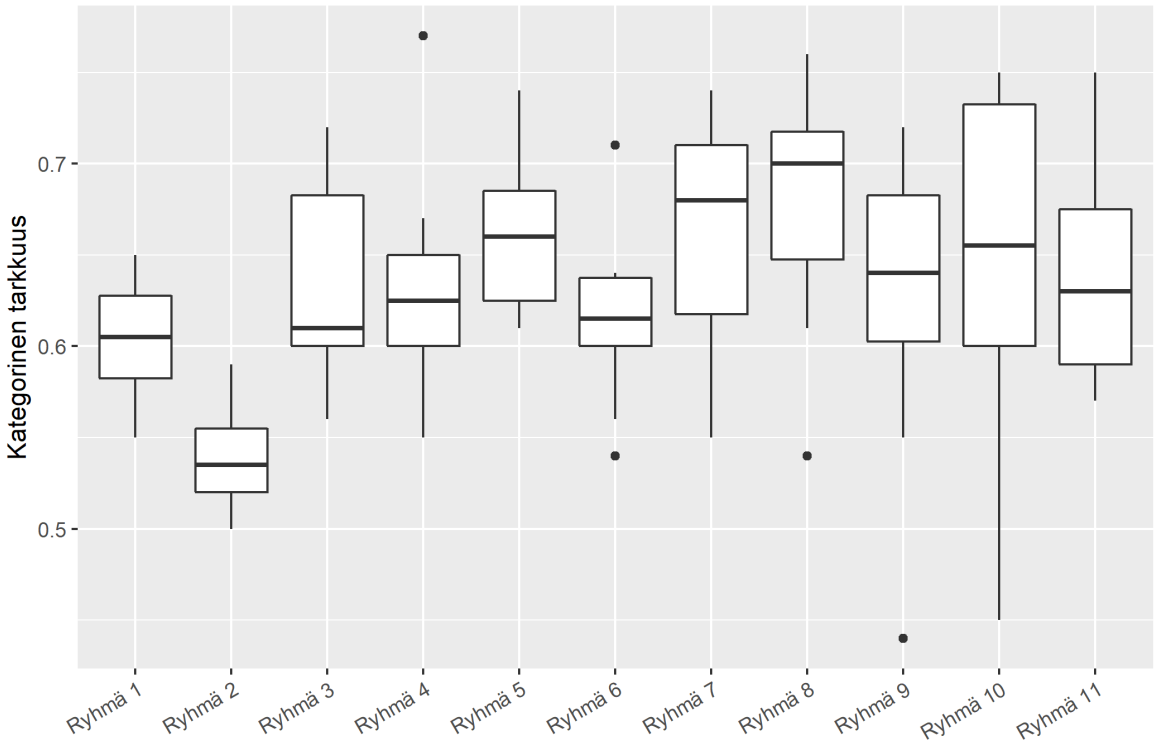
Valittiin Borutalla S-aineiston tärkeimmät taajuudet. Kuvassa 11 ja liitteessä A kuvataan, mitkä taajuudet olivat Borutan mukaan S-aineiston tärkeimpiä. Niistä nähdään, että Borutalla valitut tärkeät taajuudet ovat joko alle 2 500 Hz tai keskittyvät taajuuden 10 000 Hz molemmin puolin.



Kuva 11: Borutalla valitut tärkeät taajuudet. Kuvassa on satunnaismetsän antamat muuttujien tärkeydet (variable importance). Borutalla valitut tärkeät taajuudet on merkitty punaisella.

Konvoluutioneuroverkko ja neuroverkko

Jaettiin spektri kymmeneen 1 600 taajuuden mittaiseen taajuusalueeseen ja korvattiin jokainen taajuusalue vuorotellen saman kokoisella otoksella nollia. Tällä tavalla arvioitiin, kuinka tärkeä kukin 1600 taajuuden taajuusalue on mallille. Kutsutaan koko spektriä ryhmäksi 1 ja muokattuja spektrejä ryhmiksi 2 - 11. Malleina käytettiin samoja CNN- ja NN-malleja kuin luokittelua tehdessä (Taulukot 5 ja 8).
Arvioinnissa kaikille ryhmille sovitettiin kumpikin malli kymmenen kertaa. Neuroverkko etsii ratkaisua satunnaisista suunnista, joten näin saatiin kymmenen toisistaan poikkeavaa tappiofunktion ja kategorisen tarkkuuden arvoa. Ryhmille sovitettiin yleistetyllä pienimmän neliösumman menetelmällä estimoitu lineaarinen malli, jossa vasteena oli joko tarkkuus tai tappio, selittävänä muuttujana oli ryhmä ja jokaisella ryhmällä oli erillinen varianssiparametri. Mallista saatiin ennustetut tarkkuudet tai tappiot kaikille ryhmille. Ryhmän 1 ennustettua tarkkuutta verrattiin muiden ryhmien ennustettuihin tarkkuuksiin t-testillä.
Tilastollisesti merkitsevästi CNN-mallin tarkkuutta huononsi ainoastaan taajuuksien 1 Hz - 1 600 Hz korvaaminen nollilla. Tätä taajuusaluetta voidaan pitää mallin kannalta spektrin tärkeimpänä osana (Kuva 12 ja taulukko 17). Mallin tarkkuus parani tilastollisesti merkitsevästi, kun joko taajuudet 4 800 Hz - 6 400 Hz tai 9 600 Hz - 11 200 Hz korvattiin nollilla. Tämä johtui varmaankin siitä, että epäolennainen aineisto ei vaikuttanut malliin samalla tavalla. Yritettiin vielä tarkemmin paikallistaa tärkeät taajuudet jakamalla taajuudet 1 Hz - 1 600 Hz kymmeneen osaan. Näin saatuja uusia ryhmiä vertailtiin koko spektriin kuten aiemmin, mutta tilastollisesti merkitseviä eroja tarkkuden huonontumisessa tai tappion kasvamisessa ei löytynyt.

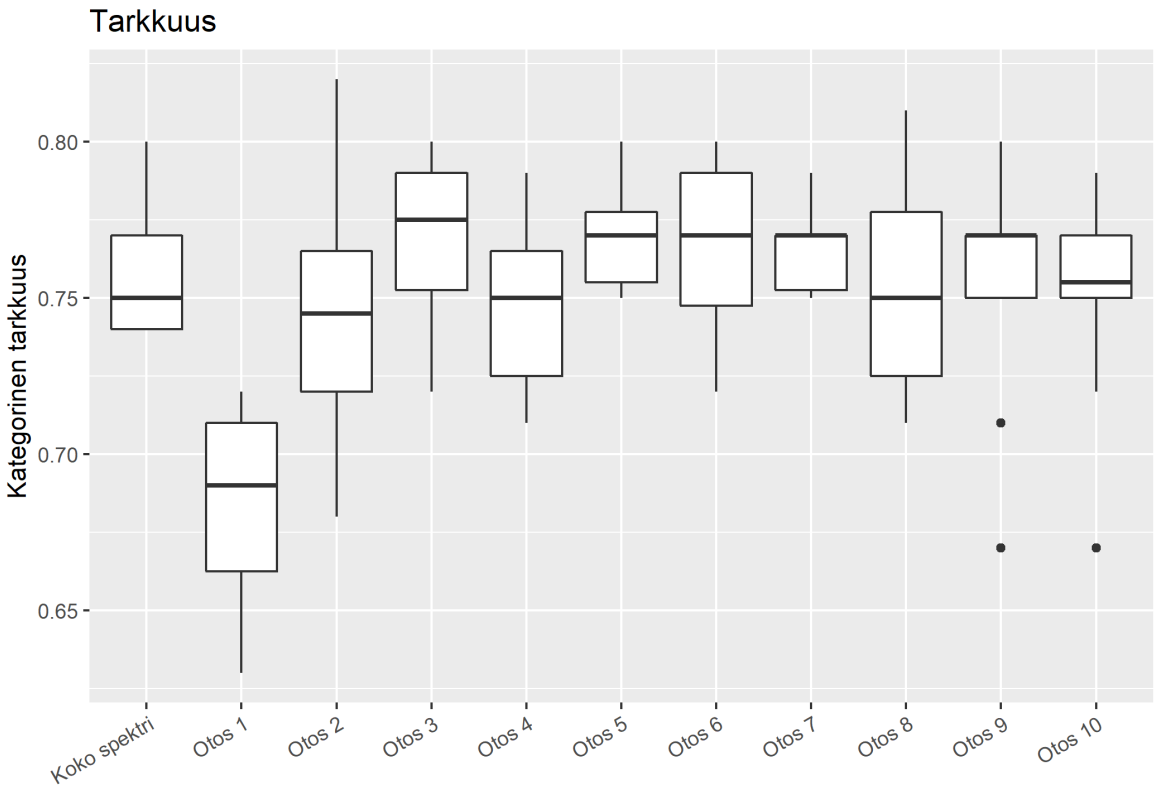


Kuva 12: Laatikkojanakuviot CNN-mallien kategorisista tarkkuuksista. Ryhmä 1 kertoo mallien kategoriset tarkkuudet koko spektrille, Ryhmä 2 kertoo kategoriset tarkkuudet, kun taajuudet 1 Hz - 1 600 Hz asetettiin nollaksi, jne. CNN-malli sovitettiin kymmenen kertaa jokaiselle ryhmälle.

Taulukko 17: Lineaarisella mallilla eri ryhmille ennustettuja kategorisia tarkkuuksia verrattiin koko spektrille ennustettuihin kategorisiin tarkkuuksiin. Kun taajuudet 1 Hz - 1 600 Hz korvattiin nollilla, ennustetut kategoriset tarkkuudet olivat tilastollisesti merkitsevästi huonompia kuin koko spektrille ennustetut kategoriset tarkkuudet. Vertailujen samanaikaisuus huomioitiin korjaamalla tulokset Šidákin menetelmällä.

Taajuudet (Hz)	**Estimaatti**	**95 % lv alaraja**	**95 % lv yläraja**	**p-arvo**
1 - 1 600	-0,064	-0,105	-0,023	$< 0{,}001$
1 600 - 3 200	0,033	-0,027	0,093	0,722
3 200 - 4 800	0,027	-0,037	0,091	0,930
4 800 - 6 400	0,059	0,010	0,108	0,008
6 400 - 8 000	0,019	-0,040	0,078	0,988
8 000 - 9 600	0,057	-0,011	0,125	0,174
9 600 - 11 200	0,076	0,008	0,144	0,017
11 200 - 12 800	0,023	-0,059	0,105	0,996
12 800 - 14 400	0,043	-0,052	0,138	0,887
14 400 - 16 000	0,037	-0,024	0,098	0,590

Tilastollisesti merkitsevästi NN-mallin tarkkuutta huononsi ja tappiofunktion arvoa kasvatti taajuuksien 1 Hz - 1 600 Hz korvaaminen nollilla. Jatkettin tärkeiden taajuuksien etsimistä jakamalla taajuudet 1 Hz - 1 600 Hz kymmeneen osaan ja vertaamalla otoksia koko spektriin kuten aiemmin. Tilastollisesti merkitsevästi NN-mallin tarkkuutta huononsi ja tappiota kasvatti taajuuksien 1 Hz - 160 Hz korvaaminen nollilla (Kuva 13 ja taulukko 18). Jatkettin jakamalla taajuudet 1 Hz - 160 Hz kahdeksaan osaan ja vertaamalla otoksia koko spektriin kuten aiemmin. Minkään näiden 20 taajuuden mittaisen taajuusalueen korvaaminen nollilla ei tilastollisesti merkitsevästi huonontanut NN-mallin tarkkuutta tai kasvattanut tappiota verrattuna koko spektriin.



Kuva 13: Laatikkojanakuviot NN-mallien kategorisista tarkkuuksista. Ryhmä 1 kertoo mallien kategoriset tarkkuudet koko spektrille, Ryhmä 2 kertoo kategoriset tarkkuudet, kun taajuudet 1 Hz - 160 Hz asetettiin nollaksi, jne. NN-mallli ajettiin kymmenen kertaa jokaiselle ryhmälle.

Taulukko 18: Lineaarisella mallilla eri ryhmille ennustettuja kategorisia tarkkuuksia verrattiin koko spektrille ennustettuihin kategorisiin tarkkuuksiin. Kun taajuudet 1 Hz - 160 Hz korvattiin nollilla, ennustetut kategoriset tarkkuudet olivat tilastollisesti merkitsevästi huonompia kuin koko spektrille ennustetut kategoriset tarkkuudet. Vertailujen samanaikaisuus huomioitiin korjaamalla tulokset Šidákin menetelmällä.

Taajuudet (Hz)	**Estimaatti**	**95 % lv alaraja**	**95 % lv yläraja**	**p-arvo**
1 - 160	-0,073	-0,108	-0,038	$< 0{,}001$
160 - 320	-0,015	-0,056	0,026	0,969
320 - 480	0,012	-0,018	0,042	0,948
480 - 640	-0,008	-0,040	0,024	0,998
640 - 800	0,013	-0,011	0,037	0,742
800 - 960	0,009	-0,024	0,042	0,996
960 - 1 120	0,008	-0,014	0,030	0,971
1 120 - 1 280	-0,003	-0,038	0,032	1,000
1 280 - 1 440	-0,003	-0,042	0,036	1,000
1 440 - 1 600	-0,007	-0,043	0,029	1,000

Neuroverkon painot

Tarkasteltiin koko spektrille sovitetun NN-mallin (Taulukko 8) sisääntulokerroksen painoja ja yritettin niiden avulla tunnistaa tärkeitä taajuuksia. Kun malli sovitettiin uudelleen, huomattiin, että suurimmat ja pienimmät painot eivät olleet aina samoilla taajuuksilla (Taulukko 19). Täten ei voitu erottaa tiettyjä taajuuksia, jotka vaikuttaisivat eniten malliin.

Taulukko 19: S-aineistolle sovitetun NN-mallin sisääntulokerroksen painot. Jokaiselle taajuudelle oli 16 000 painoa, joten käytettiin niiden itseisarvojen keskiarvoa kuvaamaan taajuuden saamaa painoa.

Taajuus (Hz)	**Taajuuden paino**
Suurimmat	
13 041	0,64
5 677	0,62
2 684	0,62
5 070	0,62
14 335	0,62
13 007	0,61
3 431	0,61
5 915	0,61
5 676	0,61
8 671	0,61
Pienimmät	
5 160	0,52
3 332	0,52
14 167	0,52
6 488	0,52
14 418	0,52
8 354	0,52
12 834	0,52
12 884	0,52
12 023	0,52
2 789	0,52

Pyrittiin erottamaan tietyt taajuudet muista lisäämällä NN-malliin paikallisesti kytketty kerros (locally connected layer), joka regularisoi painot lassolla (Taulukko 20). Lasson λ:ksi valittiin 0,001, joka on mahdollisimman suuri, mutta ei vaikuta mallin tarkkuuteen. Kerroksella käytettiin 16 000 suodatinta, jolloin neuronin saama paino vastasi taajuuden saamaa painoa. Painoja rajoitettiin siten, että niiden neliöiden summa oli yksi. Näin estettiin optimaatiofunktiota asettamasta kaikkia painoja nolliksi, kun se minimoi sakkofunktion arvoa.
Paikallisesti kytketty kerros toimii kuten konvoluutiokerros. Erona on, että konvoluutiokerros jakaa samat paikalliset suodattimet kaikkien ulostulopisteiden kesken, kun taas paikallisesti kytketyssä kerroksessa on omat paikalliset suodattimet jokaiselle ulos-

tulopisteelle. Esimerkki: olkoot suodattimet f_1, f_2, f_3 ja f_4 1×1 kokoisia ja kerrokselle sisääntuleva kuva 2×2 matriisi

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}.$$

Tällöin konvoluutiokerroksella lasketaan

$$\left[\begin{bmatrix} 1*f_1 & 2*f_1 \\ 3*f_1 & 4*f_1 \end{bmatrix}, \begin{bmatrix} 1*f_2 & 2*f_2 \\ 3*f_2 & 4*f_2 \end{bmatrix}, \begin{bmatrix} 1*f_3 & 2*f_3 \\ 3*f_3 & 4*f_3 \end{bmatrix}, \begin{bmatrix} 1*f_4 & 2*f_4 \\ 3*f_4 & 4*f_4 \end{bmatrix} \right],$$

kun taas paikallisesti kytketyllä kerroksella lasketaan

$$\begin{bmatrix} 1*f_1 & 2*f_2 \\ 3*f_3 & 4*f_4 \end{bmatrix}.$$

Taulukko 20: S-aineistolle sovitetun NN-mallin, johon lisättiin paikallisesti kytketty kerros, rakenne. Ulostulon muoto tarkoittaa kerrokselta ulos tulevan matriisin rivien ja sarakkeiden lukumäärää tai vektorin pituutta.

Kerroksen tyyppi	**Ulostulon muoto**	**Parametrien lkm**	**Aktivaatiofunktio**	**Muuta**
Sisääntulo	16 000 x 1	0	-	
Paikallisesti kytketty	16 000 x 1	32 000	-	16 000 suodatinta
Vektorisointi	16 000	0	-	
Neuroni	150	2 400 150	ReLU	
Otoksen normalisointi	150	600	-	
Ulostulo	4	604	Softmax	

Painot muuttuivat aina, kun malli sovitettiin uudelleen. Tämän takia sovitettiin malli kymmenen kertaa ja valittiin tärkeiksi taajuuksiksi ne taajuudet, joiden painojen itseisarvot olivat jokaisella kerralla 1600 suurimman painon joukossa (Taulukko 21).

Taulukko 21: Taajuudet, jotka saivat suurimpia painoja jokaisella sovituksella.

1	32	33	34	183	737	1065	1095	1134	1217	1461	1534
1799	2342	2478	2490	2613	2617	2684	2862	2939	3258	3430	3524
3546	3572	3796	3798	3941	4084	4158	4164	4645	4676	4913	5072
5613	5614	5841	5914	5952	6091	6383	6560	6705	6715	6879	7126
7192	7753	7997	8310	8423	8672	8692	8904	9100	9321	9349	9495
9547	9548	9642	9878	10140	10219	10482	10556	10582	10685	10727	11144
11168	11183	11434	11696	11697	11781	11997	12103	12659	12823	13007	13041
13143	14133	14231	14813	14882	14923	15038	15143	15512	15535	15560	15594
15614	15883	15906									

Glmnet

Katsottiin, mitä taajuuksia S-aineistolle sovitettu glmnet-malli käytti luokitteluun (Taulukko 22). Nämä taajuudet olivat glmnet-mallin mukaan luokittelun kannalta tärkeimpiä. Glmnetillä saadut tärkeät taajuudet ovat joko alle 2 500 Hz tai keskittyvät taajuuden 10 000 Hz molemmin puolin.

Taulukko 22: Glmnetin valitsemat tärkeät taajuudet.

11	14	27	34	71	72	92	96	108	109	122	123
124	439	638	1080	1134	1198	1217	1720	2114	2115	7996	7997
8624	9100	9350	9353	9547	9548	9642	10219	10297	10298	10556	10669
10987	11183	11184	11229	11696	11698	13041	14134				

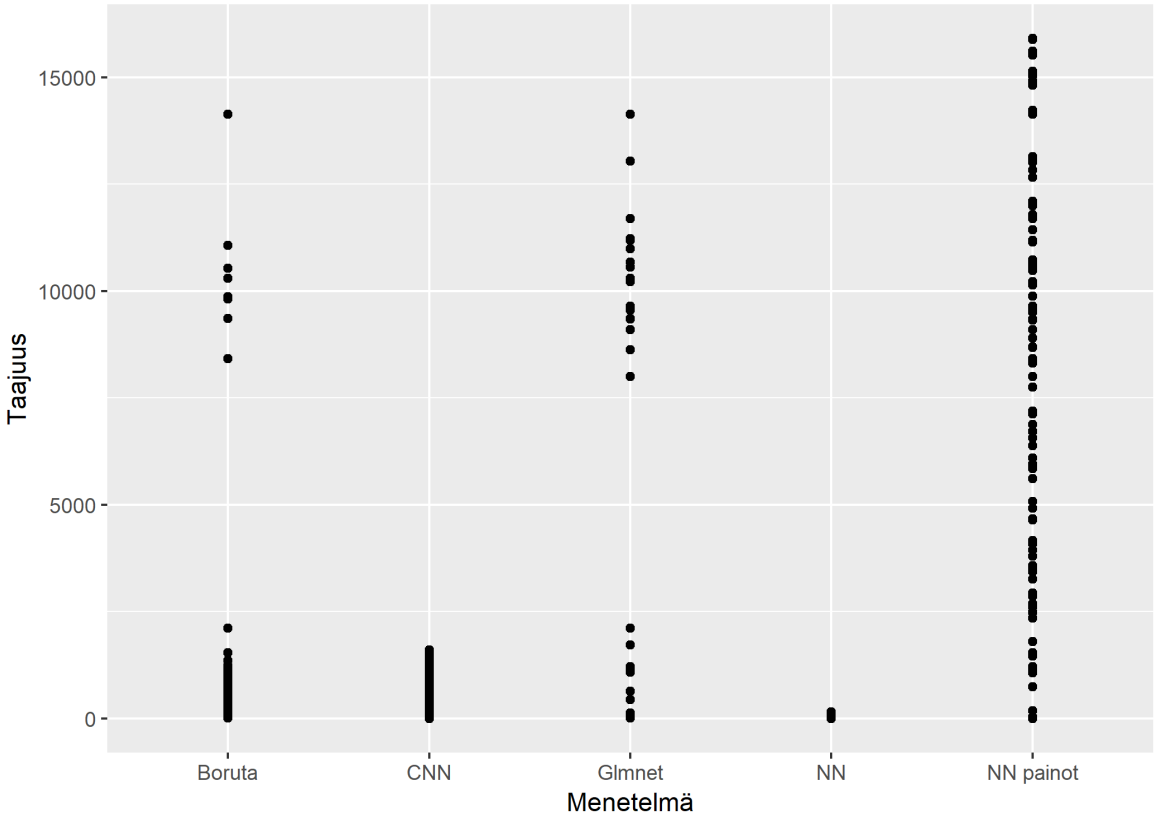
5.2.2 Tulokset

Talukossa 23 näkyy kullakin menetelmällä valittujen tärkeiden taajuuksien alueet. Alueet kattavat taajuudet 1 Hz - 15 906 Hz.

Taulukko 23: Alueet, joille tärkeät taajuudet jakaantuvat kunkin menetelmän mukaan. Taajuuksien osuus kertoo kullakin menetelmällä valittujen tärkeiden taajuuksien osuuden kaikista mitatuista taajuuksista.

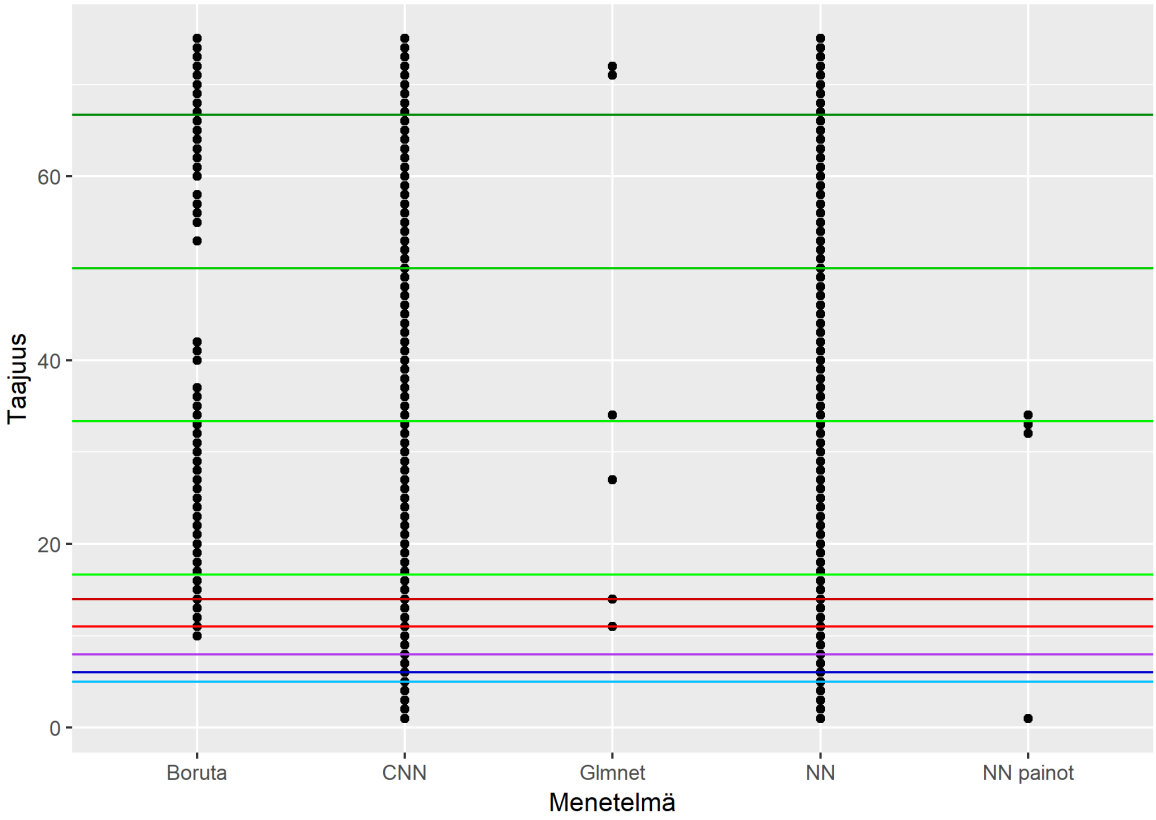
Menetelmä	**Taajuusalue (Hz)**	**Taajuuksien osuus (%)**
rf	32 - 108	0,11
Boruta	10 - 14 134	1,06
CNN	1 - 1 600	10
NN	1 - 160	1
Glmnet	11 - 14 134	0,28
NN painot	1 - 15 906	0,62

Suurin osa valituista tärkeistä taajuuksista on matalampia kuin 2 500 Hz. Kuvasta 14 nähdään, että Borutalla ja glmnetillä saadut tärkeät taajuudet ovat joko alle 2 500 Hz tai keskittyvät taajuuden 10 000 Hz molemmin puolin. Neuroverkon painoja tarkastelemalla valitut taajuudet jakaantuvat melko tasaisesti koko taajuusalueelle.



Kuva 14: Kunkin menetelmän valitsemat tärkeät taajuudet.

Kuvaan 15 on merkitty ajoneuvojen renkaiden pyörimisnopeuksia sekä moottorin pyörimisnopeuksia. Kaikilla menetelmillä, paitsi NN painot -menetelmällä, tärkeiksi valitut taajuudet 13 Hz ja 11 Hz vastaavat henkilöauton renkaiden pyörimisnopeutta 100 km/h ja 80 km/h nopeuksissa. Kaikilla menetelmillä on valittu tärkeäksi taajuudeksi jokin taajuuden 33,3 Hz lähellä oleva taajuus. Taajuus 33,3 Hz vastaa auton moottorin pyörimisnopeutta 2 000 rpm.



Kuva 15: Kunkin menetelmän valitsemat tärkeät taajuudet, jotka ovat alle 75 Hz. Kuvaan on merkitty vihreän eri sävyillä auton moottorin pyörimisnopeudet (tummimmasta vaaleimpaan 4 000 rpm, 3 000 rpm, 2 000 rpm ja 1000 rpm), tummanpunaisella viivalla 100 km/h liikkuvan henkilöauton renkaan pyörimisnopeus, vaaleanpunaisella 80 km/h liikkuvan henkilöauton renkaan pyörimisnopeus, violetilla 60 km/h liikkuvan henkilöauton tai 100 km/h liikkuvan raskaanliikenteen suurirenkaisen ajoneuvon renkaan pyörimisnopeus, tummansinisellä 80 km/h liikkuvan raskaanliikenteen suurirenkaisen ajoneuvon renkaan pyörimisnopeus ja vaaleansinisellä 60 km/h liikkuvan raskaanliikenteen suurirenkaisen ajoneuvon renkaan pyörimisnopeus.

6 Yhteenveto ja päätelmät

Tämän tutkimuksen tavoitteena oli kehitettää tilastollinen malli, jolla voidaan automaattisesti tunnistaa ajoneuvon kokoluokka autotien varteen asennettujen värähtelymittareiden keräämästä signaalista sekä löytää ne taajuudet, jotka ovat tärkeitä ajoneuvon kokoluokan tunnistamisen kannalta.
Glmnet on tilastollisen R-ohjelmointikielen paketti, jolla aineistoon voidaan sovittaa yleistetty lineaarinen malli. Mallin sovituksessa käytetään optimaatiofunktiota, joka minimoi suurimman uskottavuuden sakkotermin. Tässä tutkielmassa sakkoterminä

käytettiin lassoa, joka valitsee malliin vain tietyt muuttujat. Luokittelussa parhaat tulokset saatiin menetelmällä, jossa sovitettiin neuroverkko glmnetin valitsemille muuttujille. Parhaat kokonaistarkkudet saatiin, kun malli sovitettiin aineistolle, jossa käytettiin raaka-muotoista värähtelysignaalia (H-aineisto), ja aineistolle, jossa käytettiin värähtelysignaalin periodogrammia sekä tietoa piikkien lukumäärästä (SP-aineisto). Piikeillä tarkoitetaan kohtia, joissa värähtelysignaalin amplitudi ylittää arvon 20 000. Niiden avulla havaitaan, kuinka monta rengasparia ajoneuvolla on. H-aineistolle sovitetun mallin kokonaistarkkuus oli 0,85 ja SP-aineistolle sovitetun mallin 0,83. Eri ajoneuvoluokkien (kevyet, keskikokoiset, raskaat ja tyhjät) oikein luokiteltujen mittausten osuuksia tarkasteltaessa SP-aineistolle sovitettu malli vaikuttaa paremmalta. Oikein luokiteltujen mittauksien osuuksien kaikkien ajoneuvoluokkien keskiarvo on SP-aineistolla 85,25 % ja H-aineistolla 73,25 %. Tämän takia glmnetin valitsemille SP-aineiston muuttujille sovitettua neuroverkkoa voidaan pitää parhaana mallina. H-aineistolle sovitettu malli luokitteli oikein 94 % kevyet-luokan ajoneuvoista, joten suurempi kokonaistarkkuus johtuu kevyet-luokan ajoneuvojen suuresta lukumäärästä verrattuna muihin ajoneuvoluokkiin.

Tärkeitä taajuuksia löytyi kaikilla menetelmillä enemmän käytetyn taajuusvälin (1 Hz - 16 000 Hz) alkupäästä kuin loppupäästä. Osa tärkeistä taajuuksista keskittyi myös taajuuden 10 000 Hz ympärille. Kaikki eri menetelmillä saadut tärkeät taajuudet kattava alue oli 1 Hz - 15 906 Hz. Eri menetelmien valitsemien tärkeiden taajuuksien osuus kaikista mitatuista taajuuksista oli 0,11 % - 10 %. Parhaassa luokittelumallissa käytetyt taajuudet valittiin glmnetillä ja niiden osuus kaikista mitatuista taajuuksista oli vain 0,28 %.

Tutkimuksen kevyet ajoneuvot tunnistetaan melko hyvin, keskikokoiset huonosti ja raskaat ja tyhjät erinomaisesti. Suurin ongelma tunnistuksessa on kevyiden ja keskikokoisten ajoneuvojen erottaminen toisistaan. Valittu malli luokittelee kevyet ajoneuvot paremmin kuin keskikokoiset, mutta se johtuu luultavasti kevyiden ajoneuvojen suuresta lukumäärästä verrattuna keskikokoisiin ajoneuvoihin. Oletettavasti näiden luokkien erottamista toisistaan vaikeuttaa se, että ajoneuvot luokiteltiin käsin kuvien perusteella. Näiden luokkien ajoneuvojen massat ovat myös hyvin lähellä toisiaan, jolloin esimerkiksi täyteen lastatun farmariauton ja tyhjän pakettiauton erottaminen toisistaan värähtelyn avulla on vaikeaa. Koska kevyiden ja keskikokoisten ajoneuvojen massat ovat niin lähellä toisiaan, voidaan miettiä, onko näiden luokkien erottaminen toisistaan tarpeellista tien kunnon mittaamisen ja liikennevirran seuraamisen kannalta. Jos on, niin tutkimuksen perusteella ajoneuvoluokkien automaattinen tunnistaminen värähtelysignaalin avulla onnistuu huonolla luotettavuudella. Jos ei, niin tunnistaminen onnistuu hyvällä luotettavuudella.

Ajoneuvon liike-energia vaikuttaa eniten signaalissa esiintyvien piikkien kokoihin, joten tieto ajoneuvon massasta tai vauhdista voisi parantaa luokittelutarkkuutta. Ajoneuvon vauhti voitaisiin esimerkiksi laskea kahden peräkkäisen anturin mittaamien signaalien piikkien aikaerosta. Ajoneuvon massan estimoimiseksi voitaisiin järjestää koe, jossa mittareiden yli ajettaisiin erilaisilla ajoneuvoilla. Ajoneuvojen massat olisivat tiedossa ja niitä voitaisiin muutella järjestelmällisesti.

Luokittelun kannalta tärkeät taajuudet jakaantuvat koko käytetylle taajuusalueelle, mutta painottuvat sen alkupäähän. Jatkossa voitaisiin kokeilla esimerkiksi 1 Hz - 3 000 Hz taajuusaluetta, jolloin mitattaisiin edelleen suurinta osaa tärkeistä taajuuksista, mutta epäolennaisten taajuuksien määrä aineistossa vähenisi. Tämä säästäisi tallennustilaa ja nopeuttaisi datansiirtoa.
Aineisto kerättiin Muoniosta valtatie 21:n varrelta seitsemältä päivältä tammihelmikuussa ja sää pysyi samanlaisena aineiston keräämisen aikana. Tämän perusteella tutkielman tulokset eivät ole yleistettävissä muille teille, muina vuodenaikoina tai erilaisen sään vallitessa. Jotta tulokset olisivat yleistettävissä, käytettyä teknologiaa ja mallia pitäisi kokeilla ainakin eri materiaalista valmistetuilla teillä, uusilla tai eri tavalla kuluneilla teillä, teillä, joissa on eri nopeusrajoitus, ja pidemmältä ajalta kerätyllä aineistolla, jolloin vuodenaikojen ja sään vaihtelut voitaisiin ottaa huomioon.

Viitteet

Alkila, H. ym. 2014. "Ajoneuvoliikenteen Automaattinen Videolaskenta." Hämeen ammattikorkeakoulu.

Chen, H. Viitattu 26.3.2019. "Investigation of Stochastic Gradient Descent in Neural Networks."

Cooley, J. W. ja Tukey, J. W. 1965. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Mathematics of Computation* 19 (90). JSTOR: 297–301.

de Boer, P-T, Kroese, D., Mannor, S. ja Rubinstein, R. Y. 2005. "A Tutorial on the Cross-Entropy Method." *Annals of Operations Research* 134 (1). Springer Netherlands: 19–67. doi:10.1007/s10479-005-5724-z.

Efron, B. ja Hastie, T. 2016. *Computer Age Statistical Inference*. Vol. 5. Cambridge University Press.

Fisher, R. A. 1936. "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7 (2). Wiley Online Library: 179–88.

Friedman, J., Hastie, T. ja Tibshirani, R. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. http://www.jstatsoft.org/v33/i01/.

Grönroos, M. 2003. *Johdatus Tilastotieteeseen: Kuvailu, Mallit Ja Päättely*. Finn Lectura.

Hastie, T. ja Qian, J. 2014. "Glmnet Vignette." *Retrieved June* 9 (2016): 1–30.

Hastie, T., Tibshirani, R. ja Wainwright, M. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC press.

Holland, S. M. 2008. "Principal Components Analysis (Pca)." *Department of Geology, University of Georgia, Athens, GA*, 30602–32501.

Ioffe, S. ja Szegedy, C. 2015. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *arXiv Preprint arXiv:1502.03167*.

Kursa, M. B., Jankowski, A. ja Rudnicki, W. R. 2010. "Boruta–a System for Feature Selection." *Fundamenta Informaticae* 101 (4). IOS Press: 271–85.

Lång, K. ym. 2013. "Ainetta Rikkomattomat Tutkimusmenetelmät Katuinfran Tutkimuksissa." Tampereen ammattikorkeakoulu.

Orsini, N., Bellocco, R., Greenland, S. ym. 2006. "Generalized Least Squares for Trend Estimation of Summarized Dose-Response Data." *Stata Journal* 6 (1). Stata press 4905 Lakeway Parkway, College Station, TX 77845 USA: 40.

Oshiro, T. M., Perez P. S. ja Baranauskas J. A. 2012. "How Many Trees in a Random

Forest?" In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 154–68. Springer.

Rodríguez Galiano, V. F., Castillo, M. S., Dash, J., Atkinson, P. ja Zujar, J. O. 2016. "Modelling Interannual Variation in the Spring and Autumn Land Surface Phenology of the European Forest." *Biogeosciences, 13, 3305-3317.* Copernicus GmbH.

Rudnicki, W. R., Wrzesień, M. ja Paja, W. 2015. "All Relevant Feature Selection Methods and Applications." In *Feature Selection for Data and Pattern Recognition*, 11–28. Springer.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. ja Salakhutdinov, R. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The Journal of Machine Learning Research* 15 (1). JMLR. org: 1929–58.

Šidák, Z. 1967. "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions." *Journal of the American Statistical Association* 62 (318): 626–33.

The MathWorks, Inc. 1994-2018. Viitattu 6.2.2019. https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html.

Liitteet

A Tärkeät taajuudet Boruta

Taulukko 24: Borutalla valitut tärkeät taajuudet.

10	11	12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33	34	35
36	37	40	41	42	53	55	56	57	58	60	61	62
63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99	100	101	102
103	104	105	106	107	108	109	110	111	112	113	114	115
117	119	120	121	122	123	124	125	126	127	128	131	133
136	137	138	141	142	143	170	171	233	342	343	344	345
346	385	439	440	441	442	443	444	538	539	635	636	637
638	639	640	641	642	734	735	736	737	774	850	851	862
863	888	911	929	1019	1046	1080	1095	1096	1123	1124	1226	1227
1234	1361	1543	2112	8411	9353	9818	9819	9868	10299	10535	11063	14134

B Autonrenkaiden pyörimisnopeus

Autonrenkaan koko merkitään renkaaseen esimerkiksi näin 195/65R15. Tämä tarkoittaa, että renkaan poikkileikkausleveys on $195mm$, renkaan profiilisuhde eli renkaan korkeus suhteessa poikkileikkausleveyteen on 65 prosenttia ja vanteen koko eli renkaan reiän halkaisija on 15 tuumaa. Näiden tietojen avulla voidaan laskea renkaan säde, joka on esimerkin tapauksessa $(195mm * 0,65 * 2 + 15 * 25,4mm)/2 = 317,25mm$. Autonrenkaan pyörimisnopeus voidaan laskea kaavalla:

$$\omega = \frac{v}{2\pi r},$$

jossa ω on pyörimisnopeus, v on tangenttinopeus (eli tässä ajoneuvon nopeus) ja r on renkaan säde. Henkilöautojen renkaiden pyörimisnopeudet ovat melkein samat, vaikka vanteet olisivat eri kokoiset. Tämä johtuu siitä, että suuremmilla vanteilla käytetään matalampia renkaita, jotta renkaat mahtuvat rengaskoteloon. Esimerkiksi, kun ajoneuvon nopeus on 100 km/h, 195/65R15 kokoisten renkaiden pyörimisnopeus on noin 13,93529 kierrosta sekunnissa ja 205/55R16 kokoisten renkaiden pyörimisnopeus on noin 13,99263 kierrosta sekunnissa. Raskaan liikenteen ajoneuvoilla rengaskoko voi suurimmillaan olla esimerkiksi 425/65R22.5. Tällaisen renkaan pyörimisnopeus on noin 7.866496 kierrosta sekunnissa, kun ajoneuvon nopeus on 100 km/h.