Tarkan liikennedatan tilastollinen analyysi ja ennustaminen koneoppimismenetelmin

Suvi Laine

Perustieteiden korkeakoulu

Kandidaatintyö
Espoo 27.8.2020

Vastuuopettaja

Apulaisprofessori Pauliina Ilmonen

Työn ohjaaja

DI Henri Salmenjoki

Aalto-yliopisto
Perustieteiden
korkeakoulu

Copyright © 2020 Suvi Laine

The document can be stored and made available to the public on the open internet pages of Aalto University.
All other rights are reserved.

Aalto-yliopisto
Perustieteiden korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Tekniikan kandidaatintyön tiivistelmä

Tekijä Suvi Laine

Työn nimi Tarkan liikennedatan tilastollinen analyysi ja ennustaminen koneoppimismenetelmin

Koulutusohjelma Teknillinen fysiikka ja matematiikka

Pääaine Matematiikka ja systeemitieteet **Pääaineen koodi** SCI3029

Vastuuopettaja Apulaisprofessori Pauliina Ilmonen

Työn ohjaaja DI Henri Salmenjoki

Päivämäärä 27.8.2020 **Sivumäärä** 31+3 **Kieli** Suomi

Tiivistelmä

Liikenteen tilan ja ruuhkien täsmällisten ennusteiden avulla voidaan esimerkiksi optimoida matka-aikoja, säädellä polttoaineenkulutusta ja suositella reittejä ruuhkien perusteella. Informaatiota liikenneruuhkien muodostumisesta voidaan hyödyntää myös kaupunkisuunnittelussa. Liikenneaineistoa voidaan kerätä joko perinteisin menetelmin esimerkiksi tiekameralla tai ajoneuvoihin kiinnitetyillä sensoreilla.

Tämän kandidaatintyön tutkimusaineistona on sensorein kerätty liikenneaineisto, josta saadaan myös tieto sääolosuhteista. Työn tavoitteena on tunnistaa ja ennustaa liikenteen tilaa sensoriajoneuvojen avulla kerätystä liikenneaineistosta. Ennustaminen toteutetaan 15 minuutin aikasegmenteissä. Lisäksi työssä vertaillaan yksittäisten ajoneuvojen pohjalta saatuja ennusteita kokonaisten aikasegmenttien ennusteisiin. Aineisto on rajattu Landvetterin lentokentän läheiselle moottoritielle Göteborgiin, josta on tutkimusaineistossa tammikuun 2019 ja huhtikuun 2020 väliltä noin 13 miljoonaa havaintoa. Aineistoa on jaettu itään ja länteen päin ajaviin.

Lopputuloksena työssä saadaan kolme luokittelijaa liikenteen tilan tunnistukseen. Parhaiten tehtävään soveltuvat AdaBoost ja lineaarinen erotteluanalyysi, kun taas tukivektorikoneen tarkkuudet jäävät matalammiksi. Saadut ennustustarkkuudet ovat parempia kuin pelkän suuremman luokan ennustaminen tai satunnainen arvaus. Yksittäisten ajoneuvojen perusteella saadut ennusteet ovat hieman huonompia kuin kokonaisten aikasegmenttien. Länsisuunnan ennusteet ovat hieman parempia kuin itään päin ajavien pohjalta saadut. Voidaan todeta, että työssä käytetyillä ruuhkan määritelmillä pystytään erottelemaan aikasegmenttejä niiden liikenteen tilan perusteella.

Avainsanat koneoppiminen, tukivektorikone, lineaarinen erotteluanalyysi, AdaBoost, ruuhkan ennustaminen

Aalto-yliopisto
Perustieteiden
korkeakoulu

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the bachelor's thesis

Author Suvi Laine

Title Statistical Analysis And Prediction of Traffic Data Using Machine Learning Methods

Degree programme Engineering Physics and Mathematics

Major Mathematics and Systems Sciences **Code of major** SCI3029

Teacher in charge Asst. Prof. Pauliina Ilmonen

Advisor MSc (Tech.) Henri Salmenjoki

Date 27.8.2020 **Number of pages** 31+3 **Language** Finnish

Abstract

Accurate predictions of traffic state and congestion can be used to optimize travel times, regulate fuel consumption and recommend travel routes based on traffic congestion. Information about the formation of traffic congestions can also be used in urban planning. Traffic data can be gathered with traditional methods, such as road cameras, or by collecting data directly from moving vehicles.

This thesis uses a floating car data set collected with sensors attached to probe vehicles. Information about weather conditions is also included in the data set. The goal of this thesis is to identify and predict traffic state with a floating car data set. The prediction was executed in 15 minute time segments. Predictions based on individual probe vehicles were also compared to predictions from full time segment data. The analysis was limited to a highway near Landvetter airport near Gothenburg in Sweden and there are about 13 million data points between January 2019 and April 2020. The data set was divided into two based on the heading of the vehicle.

This study results in three classifiers for traffic state identification. The best classifiers by performance are AdaBoost and linear discriminant analysis while support vector machine performs slightly worse. The prediction accuracies were better than random guess or predicting only the larger class. The results from individual vehicles were slightly less accurate than predictions based on whole time segments. In conclusion, based on the definitions of traffic congestion that were used, it is possible to separate time segments by their traffic state.

Keywords machine learning, support vector machine, linear discriminant analysis, AdaBoost, traffic congestion prediction

Sisältö

Tiivistelmä **3**

Tiivistelmä (englanniksi) **4**

Sisältö **5**

1 Johdanto **6**

2 Aikaisempi tutkimus **7**

3 Menetelmät **8**
3.1 Koneoppiminen ja ohjattu oppiminen 8
3.2 Tukivektorikone 8
3.3 AdaBoost 10
3.4 Lineaarinen erotteluanalyysi 12
3.5 k:n osituksen ristiinvalidointi 13

4 Aineiston kuvaus **14**

5 Tulokset **22**
5.1 Koko aikasegmentin tulokset 22
5.2 Yksittäisten ajoneuvojen tulokset 24
5.3 Tulosten analysointi 26

6 Yhteenveto **28**

A Luokittelijoiden vertailu **32**
A.1 Jakaumamääritelmän luokittelijoiden vertailu 32
A.2 Häntämääritelmän luokittelijoiden vertailu 33

B Koneoppimismenetelmien tarkkuudet **34**

1 Johdanto

Kaupungistuminen on maailmanlaajuisesti johtanut lisääntyneisiin liikenneruuhkiin kaupunkialueilla. Viime vuosina keskimääräinen autoilija Yhdysvalloissa on vuosittain kuluttanut arviolta yli 50 tuntia enemmän autoiluun ruuhkan takia ja liikenneruuhkan vuosittainen kustannus autoilijaa kohden on yli 1000 $ [1]. Ongelma voidaan ratkaista reaaliaikaisella ruuhkan ennustamisella, jolloin autoilija voi valita vaihtoehtoisen reitin ruuhkan välttämiseksi. Täsmällisiä liikenteen tilan ennusteita voidaan siis hyödyntää matka-aikojen optimoinnissa, polttoaineenkulutuksen säätelyssä sekä tieonnettomuuksien välttämisessä [2, 3]. Informaatiota liikenneruuhkien muodostumisesta hyödynnetään myös kaupunkisuunnittelussa toimivien tieverkkojen rakentamisen tukena.

Liikenneanalyysia varten aineisto voidaan kerätä ajoneuvoihin kiinnitettävillä sensoreilla tai perinteisemmin kiinnitetyillä mittauspisteillä. Sensoreilla kerätyllä aineistolla on etuja perinteisiin liikennedatan keräysmenetelmiin verrattuna. Perinteiset liikenteen valvonnan laitteet, kuten nopeuskamerat ja induktiosilmukat, ovat kalliita asentaa ja huoltaa sekä vaativat olemassa olevaa infrastruktuuria eivätkä tällöin ole saatavissa kaikille alueille. [4, 5] Tässä työssä käytettävä tutkimusaineisto on ammattiajossa oleviin ajoneuvoihin kiinnitetyillä sensoreilla kerättyä, joten työ keskittyy tutkimaan tälläisen liikenneaineiston ominaisuuksia. Havainnot on mitattu Ruotsissa ja työ keskittyy tarkastelemaan aineistoa Göteborgin Landvetterin lentokentän läheisyydestä moottoritieltä. Työn tutkimusaineistossa on mukana myös sensoreilla mitattu tieto tien kunnosta ja sääolosuhteista, jotka usein puuttuvat samalla menetelmällä kerätyistä aineistoista.

Tutkimuksen päätavoitteena on selvittää, voidaanko sensoriajoneuvoilla kerätyn tutkimusaineiston avulla tulkita ja ennustaa liikenteen tilaa. Havaintoja käsiteltiin keskiarvoina 15 minuutin aikasegmenteissä nopeuteen pohjautuvien työssä määriteltyjen ruuhkan määritelmien avulla. Työssä vertaillaan eri koneoppimisluokittelijoiden, tukivektorikoneen, AdaBoostin sekä lineaarisen erotteluanalyysin, soveltuvuutta tähän tehtävään. Lisäksi tutkitaan, onnistuuko liikenteen tilan ennustaminen myös yksittäisen auton havaintojen perusteella. Tutkimuksessa on otettava huomioon aineiston alkuperä ja ominaisuudet. Aineistossa on ennalta tuntematon määrä havaintoja kerääviä sensoriajoneuvoja eikä Göteborgin lähialueilla moottoritiellä välttämättä synny samanlaisia ruuhkia kuin suurkaupungeissa tai keskustassa.

Kandidaatintyön rakenne on seuraava. Ensimmäisenä käsitellään aikaisempaa kirjallisuutta liikenteen tilan ja ruuhkan ennustamisesta sekä liikenneaineistoon pohjautuvaa tutkimusta. Sen jälkeen esitellään tutkimuksessa käytetyt koneoppimismenetelmät ja teoriaa mallien evaluoimisesta. Tutkimusaineistoa kuvaillaan osiossa ”Aineiston kuvaus”. Osiossa ”Tulokset” esitetään saadut tulokset sekä arvioidaan niitä. Lopulta yhteenvedossa kootaan työn tärkeimmät johtopäätökset. Lisäksi pohditaan aineistopohjaisen liikenteen tilan ennustamisen kehitysnäkymiä tulevaa tutkimusta varten.

2 Aikaisempi tutkimus

Liikennejärjestelmä on dynaaminen systeemi, joten sen tilan ennustamisessa tulee ottaa huomioon useita asioita. Erityyppisistä liikenneaineistosta on tehty paljon tutkimuksia, joissa pyritään ennustamaan liikenteen tilaa. Useimmiten liikenneaineistot on kerätty joko fyysiseltä mittauspisteeltä tien varrelta, esimerkiksi kamerasta, älykännykän paikannuksella tai sensoriajoneuvojen avulla kuten tämän työn aineistossa.

Keskeinen ongelma liikenteen tilan tunnistuksessa on se, miten ruuhka ilmenee aineistosta. Hongsakham et al. [6] jakaa aineiston havainnot ruuhkaisiin ja ruuhkattomiin ihmisen harkintakyvyn perusteella. Ihmisen harkintakykyä voidaan käyttää hyväksi myös sovelluksien kautta. Esimerkiksi Waze-sovellus antaa käyttäjien ilmoittaa liikenneruuhkia sovellukseen ja vastapalveluksena lähettää käyttäjille tietoa ruuhkista [7]. Ihmislähtöinen arviointi nojaa siihen, että ihmisellä on paras harkintakyky liikenteen tilan poikkeavuuksien tunnistamisessa.

Suurelle aineistolle on kuitenkin erittäin aikaavievää käydä aineisto käsin läpi. Sensoriautojen avulla kerätään yleisesti matka-aikoja ja nopeuksia tiesegmenteillä, sillä näiden avulla voidaan saada informaatiota liikenteen tilasta [5, 8, 9]. Esimerkiksi alentuneet nopeudet tai tiekamerasta nähtävä pysähtelevä liike voidaan määritellä ruuhkaksi [4]. Perinteisillä liikenneaineiston keräysmenetelmillä voidaan myös mitata liikenteen virtausta esimerkiksi ohimenevien ajoneuvojen määränä aikayksikössä, jonka avulla ruuhkaisuutta voidaan arvioida [10].

Liikenteen tilaa on kuitenkin onnistuttu ennustamaan erityyppisistä aineistoista. de Fabritiis et al. [5] käyttivät aineistonaan 600 000 yksityisautosta kerättyä aineistoa, jonka avulla määriteltiin tiesegmenteille keskinopeuksia kolmen minuutin aikaintervalleilla. Tutkimuksessa tehtiin lyhytaikaisia ennustuksia liikenteen tilasta nopeuteen perustuen hahmonsovituksella (englanniksi pattern matching) ja neuroverkolla. Pienin keskivirhe, noin 2%-8%, saavutettiin 15 minuutin ennustukselle neuroverkoilla. Lisäksi liikenneaineistoa voidaan käsitellä aikasarjana. Esimerkiksi Williams ja Hoel [10] käyttivät kameralla ja induktiokierteellä mitatun aineiston pohjalta liikenteen lyhytaikaiseen ennustamiseen ARIMA-mallia eli integroitua autoregressiivistä liukuvan keskiarvon mallia. Ennustamisessa käytettiin vain liikenteen määrää, mutta aikasarjamallin todettiin tutkimuksessa ennustavan lyhyttä 15 minuutin aikaväliä yhtä hyvin kuin esimerkiksi neuroverkko.

Wang ja Tsai [4] toteuttivat ruuhkan tunnistuksen kävelytielle asetetun tiekameran avulla kerätyllä aineistolla. Kamera tarkasteli yksittäisen ajoneuvon nopeutta sekä liikerataa korkeintaan 80 metrin päähän. Ennustamiseen käytettiin tukivektorikonetta, satunnaismetsää sekä AdaBoost-menetelmää, ja tarkkuudeksi saatiin korkeimmillaan 88,94%. Ongelmana menetelmässä oli sen heikko yleistyvyys, sillä se oli tarkka vain yksittäisellä tiesegmentillä, kuten myös monet muut liikenteen ennustamisen mallit. Esimerkiksi spatiaalisella analyysilla voidaan tukea muita menetelmiä yleistyvyyden parantamiseksi ja mallin laajentamiseksi muille tiesegmenteille. [2]

Vaikka sensoriajoneuvoilla kerätyllä aineistolla on onnistuttu tarkassa ruuhkan tunnistuksessa, nousee tälläisen aineiston ongelmaksi riittävä ajoneuvojen määrä liikenteessä ennustusajanhetkillä. Sensoreilla kerätyn liikennedatan tarkkuus riippuu siitä, kuinka suuri osuus liikenneverkon autoista on varustettu sensoreilla. Cheu et al.

[11] ehdottaa, että riittävän tarkan nopeuden arvioimiseksi tarvitaan simulaatioiden perusteella vähintään kymmenen sensoriautoa tarkasteltavalla tiesegmentillä jokaiselle aikavälille. Riittävän sensoriautomäärän saavuttaminen koetaankin tämäntyyppisen liikennedatan heikkoudeksi [4]. Sensoriajoneuvoilla kerättyä aineistoa voidaan kuitenkin käyttää myös täydentämään perinteisillä menetelmillä kerättyjä havaintoja. [5, 11]

Yhteenvetona voidaan todeta, että liikenteen ennustamisessa toimivina menetelminä käytetään usein esimerkiksi aikasarjamalleja, tilastollisia menetelmiä, neuroverkkoja tai tukivektorikonetta. Aineiston keräämiseen on useita eri menetelmiä, joten sopiva menetelmä valitaan tutkimuksen tavoitteen perusteella. Tässä työssä käytettävällä sensoriajoneuvoista kerätyllä aineistolla on saatu tarkkoja ennusteita, mutta aineistonkeräysmenetelmän heikkoudeksi liikenteen tilan ennustamisessa tulkitaan havaintojen liian pieni määrä suhteutettuna liikenteen todelliseen virtaukseen. Seuraavaksi käsitellään tässä tutkimuksessa vertailtujen menetelmien teoreettista taustaa. Tässä työssä on vertailtu myös esittelemättä jätettyjä koneoppimismalleja, ja liitteessä A on esitelty niiden tuloksia.

3 Menetelmät

3.1 Koneoppiminen ja ohjattu oppiminen

Koneoppimisessa opetetetaan algoritmia olemassa olevan aineiston pohjalta. Koneoppimismallit ovat algoritmeja, jotka kykenevät opetuksen jälkeen tunnistamaan havaintoaineistosta säännönmukaisuuksia. Tässä työssä käytetään ohjatun oppimisen menetelmiä liikenneaineiston luokitteluun. Ohjatussa oppimisessa aineiston luokat tiedetään ennalta. Havainnot jaetaan opetusaineistoon, jolla algoritmia koulutetaan, sekä testausaineistoon, jolla algoritmin suorituskykyä arvioidaan. Tehtävään sopivan koneoppimismallin löytäminen saattaa olla hankalaa. [12]

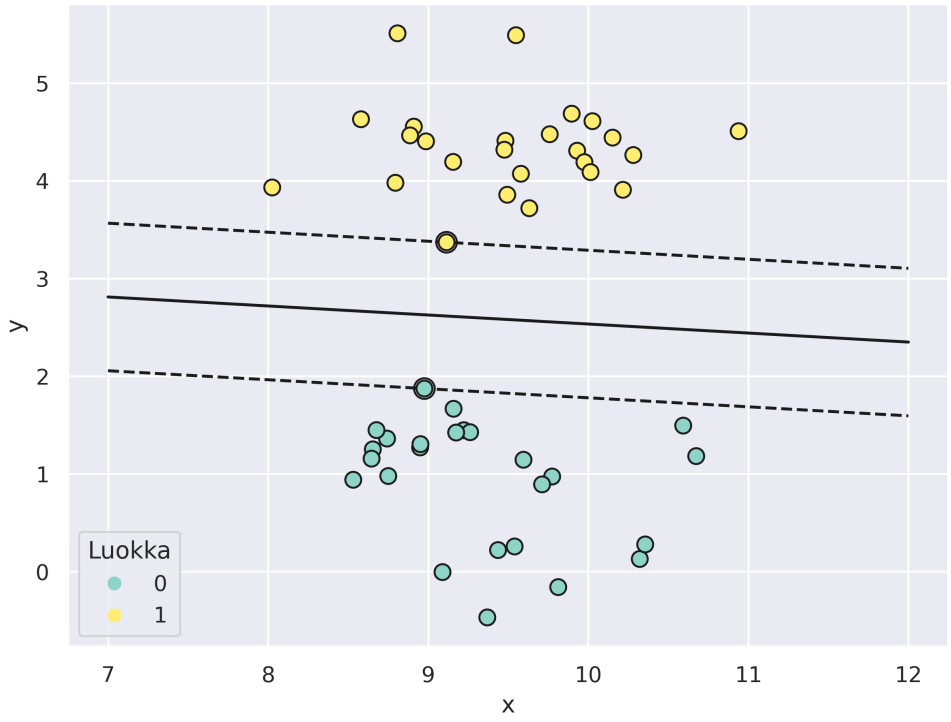
Koneoppimista käytetään erityisesti luokitteluun ja regressioon. Lisäksi sillä on useita sovelluksia, kuten puheentunnistus, osakemarkkinoiden ennustaminen ja konenäkö. [12] Koneoppimisella voidaan ratkaista ongelmia, joiden ratkaiseminen on ihmiselle tai yksinkertaiselle tietokoneohjelmalle vaikeaa, sillä algoritmit pyrkivät löytämään aineistostaan säännönmukaisuuksia ja malleja. Perusteellisempi johdatus koneoppimiseen löytyy esimerkiksi kirjasta Alpaydin ja Bach [12].

3.2 Tukivektorikone

Tukivektorikone (englanniksi support vector machine, lyhenne SVM) on monipuolinen luokittelijamalli, joka soveltuu myös regressio-ongelmiin ja hahmontunnistukseen [13]. Tukivektorikoneen luokitteluperiaatteen ymmärtämiseen tarvitaan neljä käsitettä: päätöstaso, maksimaalinen yhdensuuntaisten marginaalitasojen välimatka, joustava marginaali sekä ydinfunktio.

Kun SVM:lle annetaan valmiiksi luokiteltu opetusjoukko, se etsii optimaalisen päätöstason, jonka perusteella uusia havaintoja voidaan luokitella. Päätöstasolle haetaan mahdollisimman suuret marginaalit siten, että luokkien päätöstasoa lähimpien

pisteiden, tukivektoreiden, etäisyys päätöstasoon on mahdollisimman suuri. Kuvassa 1 nähdään kaksi luokkaa, jotka on erotettu päätöstasolla. Katkoviivalla näkyvät yhdensuuntaiset marginaalitasot, joiden välimatka on mahdollisimman suuri tälle havaintojoukolle. Lisäksi kuvassa on merkittynä tukivektorit, eli marginaalitasojen välimatkaa rajoittavat havainnot. [14]



Kuva 1: Esimerkki tukivektorikoneen toiminnasta lineaarisella ydinfunktiolla. Päätöstaso merkitty kuvassa viivalla ja marginaalit katkoviivalla. Korostetut näytepisteet marginaalin päällä ovat tukivektorit.

Yllä esitelty luokitinmalli ei kuitenkaan sovellu näytejoukolle, joka ei separoidu täydellisesti. Tällöin tarvitaan joustavan marginaalin luokitin (englanniksi Soft Margin SVM), jonka Cortes ja Vapnik [15] esittävät laajennuksena joustamattomaan tukivektorikoneeseen. Joustavan luokittelijan SVM hakee maksimaalisen marginaalin lisäksi minimaalisen virhehavaintojen, eli luokkansa ulkopuolisten pisteiden, määrän minimoimalla jokaiselle havainnolle i määrättyjen slack-muuttujien $\xi_i \geq 0, i = 1, ..., m$ summan. Kun havainto i on päätöstason mukaan oikein luokiteltu, se saa arvokseen $\xi_i = 0$. Olkoon luokittelijan opetusjoukko $(\boldsymbol{x}_i, y_i), i = 1, ..., m$, missä $\boldsymbol{x}_i \in \mathbb{R}^n$ ja $\boldsymbol{y} \in \{1, -1\}$. Optimaalisen joustavan marginaalin päätöstason $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, missä

$\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}$ ratkaiseminen vaatii seuraavan optimointitehtävän ratkaisua:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \end{aligned} \tag{1}$$

jossa kerroin $C > 0$ on virhetermin rangaistusparametri. Kerroin C kontrolloi mallin kompleksisuuden ja empiirisen virheen tasapainoa. Suurempi C:n arvo siis pienentää marginaalien välimatkaa. [16, 15]

Jos kaikki aineisto ei ole lineaarisesti separoituvaa, tarvitaan tällöin ydinfunktio

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j),$$

jossa ϕ on funktio $\phi : \mathbb{R}^d \mapsto \mathbb{R}^l$, joka muuntaa opetusjoukon vektorit $\boldsymbol{x}_i$ korkeampaan dimensioon. Ydinfunktio muuntaa ongelman korkeampaan dimensioon sopivilla kantafunktioilla, jolloin ongelma voidaan mahdollisesti ratkaista lineaarisesti uudessa kannassa [12]. Tavallisimmat ydinfunktiot ovat lineaarinen $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T\boldsymbol{x}_j$, polynomiaalinen $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\gamma\boldsymbol{x}_i^T\boldsymbol{x}_j + r)^d$ sekä radiaalikantafunktio $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$, joissa $d, \gamma > 0$ ovat parametrejä. Kun ydinfunktiota käytetään luokittelussa, muuntuu ratkaistavan optimointiongelman (1) ensimmäinen rajoite muotoon

$$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i.$$

Tässä työssä joustavan marginaalin SVM toteutettiin Python-ohjelmointikielen scikit-learn -kirjastolla. [17] Ydinfunktiona käytettiin lineaarista funktiota, sillä se antoi parhaat tarkkuudet. Lisää tietoa tukivektorikoneista löytyy esimerkiksi Cortes ja Vapnik [15] sekä Vapnik [18].

3.3 AdaBoost

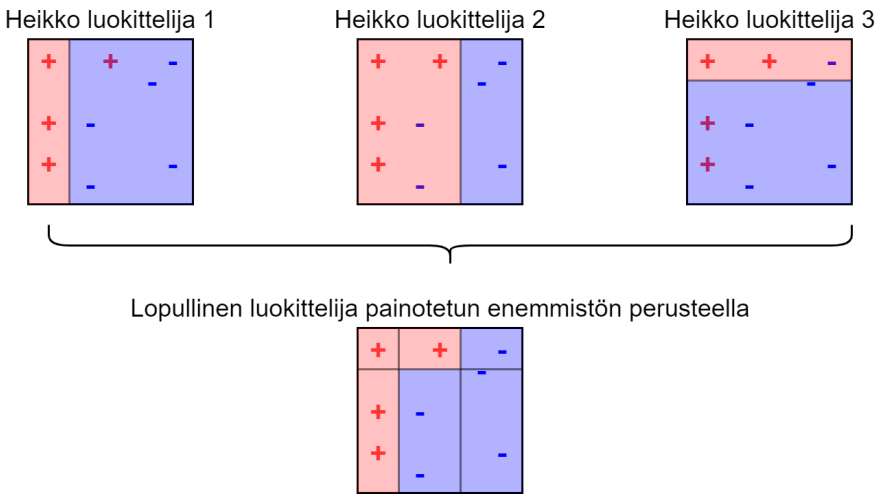
Freund ja Schapire [19] esittelivät mukautuvan tehostamisalgoritmin, AdaBoostin, vuonna 1995. Tehostaminen on yleisesti menetelmä, jolla parannetaan jonkin koneoppimisalgoritmin suorituskykyä [20]. Tehostamismenetelmät hyödyntävät heikkoja oppijoita tai luokittelijoita, eli sellaisia koneoppimismenetelmiä, joiden tarkkuus on vain hieman parempi kuin satunnainen arvaus [21]. Menetelmä syntyi, kun Schapire [22] selvitti, että heikoista oppijoista voidaan koostaa vahva oppija.

Olkoon opetusjoukko $(\boldsymbol{x}_i, y_i), i = 1, ..., m$, missä $\boldsymbol{x}_i \in \mathbb{R}^n$ ja $\boldsymbol{y} \in \{1, -1\}$. AdaBoost kutsuu toistuvasti iteraatioiden $t = 1, ..., T$ aikana heikkoa oppijaa siten, että jokaisen kierroksen jälkeen väärin luokiteltujen havaintojen painoarvo oppimisessa kasvaa. Heikon luokittelijan tavoitteena on löytää heikko hypoteesi $h_t : X \to \{-1, +1\}$, jonka hyvyyttä voidaan arvioida virheellä

$$\epsilon_t = Pr_{i\sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i)\neq y_i} D_t(i), \tag{2}$$

missä $D_t(i)$ on iteraation i painoarvojen jakauma opetusjoukon yli. Ensimmäisellä iteraatiolla käytetään tasajakaumaa $D_1(i) = 1/m$. [21] Kuvassa 2 on havainnollistettu

tehostamisalgoritmien toimintaa. Lopullinen luokittelija muodostetaan heikkojen hypoteesien painotetun keskiarvon avulla.



Kuva 2: Tehostamismenetelmien toiminta yksinkertaistettuna. Lopullinen luokittelija muodostetaan heikkojen luokittelijoiden painotettuna enemmistönä kaikkien iteraatioiden jälkeen.

Jokaisella iteraatiolla t AdaBoost opettaa ensin heikkoa luokittelijaa jakaumalla D_t. Tällöin saadaan heikko hypoteesi $h_t : X \to \{-1, +1\}$, jonka virhe ϵ_t on kaavan (2) mukainen. Seuraavaksi valitaan parametri $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$. Parametri α_t määrittää AdaBoostin lopullisessa vaiheessa iteraation t heikon luokittelijan painoarvon. Mikäli luokittelija tekee paljon virheitä, on sen painoarvo pienempi kuin paremmin pärjäävän mallin. Lopulta päivitetään painoarvojen jakaumaa

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} exp(-\alpha_t), & \text{jos } h_t(x_i) = y_i \\ exp(\alpha_t), & \text{jos } h_t(x_i) \neq y_i \end{cases}$$
$$= \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

missä Z_t on normalisointikerroin, joka valitaan siten, että D_{t+1} pysyy jakaumana. Kun T iteraatiota on suoritettu, saadaan lopullinen hypoteesi

$$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x)).$$

Menetelmässä siis valitaan se luokka, jonka painotettu enemmistö heikoista luokittelijoista valitsi. [21]

Tehostamisalgoritmien yhdistäviä ominaisuuksia ovat niiden taipumus keskittyä luokittelussa vaikeimpiin (useimmiten väärin luokiteltuihin) havaintoihin sekä luokittelun lopputuloksen vahva riippuvuus opetusjoukosta. Eri tavalla valitulla opetusjoukolla muodostetaan hyvinkin erilaisia hypoteesejä. Tehostamisalgoritmien

etuna on kuitenkin se, että ne vähentävät harhaa ja varianssia [23, 21]. Verrattuna aiemmin esiteltyyn tukivektorikoneeseen, AdaBoost ei myöskään ylisovita yhtä herkästi. [21]

Tässä työssä AdaBoost on toteutettu scikit-learn -kirjaston avulla, jossa AdaBoostin heikkona luokittelijana on käytetty päätöspuuta. [17] Päätöspuu on heikko oppija luokitteluun ja regressioon, joka koostuu päätössolmuista ja lehdistä. Päätöspuun syvyys on lehtiä edeltävien solmujen lukumäärä. Jokaisessa solmussa aineisto jaetaan jonkin muuttujan perusteella, kuten kuvan 2 heikot luokittelijat tekevät. Päätöspuun lehdet kertovat luokittelun lopputuloksen. AdaBoostissa käytettävän päätöspuun syvyys on yksi, mikä tarkoittaa, että kukin päätöspuu tekee luokittelun vain yhden aineiston piirteen perusteella. Perusteellisempi johdatus päätöspuihin löytyy esimerkiksi kirjasta Alpaydin ja Bach [12].

3.4 Lineaarinen erotteluanalyysi

Lineaarinen erotteluanalyysi (englanniksi linear discriminant analysis, lyhenne LDA) on tilastotieteen ja koneoppimisen menetelmä, jonka tavoitteena on löytää lineaarikombinaatio sellaisista aineiston ominaisuuksista, jotka erottavat vähintään kaksi luokkaa toisistaan [24]. Menetelmä perustuu muodostettavaan Fischerin lineaariseen erottelufunktioon. Erottelufunktion z kaava on

$$z = \boldsymbol{w}^T \boldsymbol{x} = w_1 x_1 + w_2 x_2 + ... + w_p x_p, \tag{3}$$

missä $\boldsymbol{x} = x_1, ..., x_p$ on havaintovektori ja $w_i, i \in 0, 1, ..., p$ ovat optimaaliset painokertoimet erottelufunktiolle. Painokerroinvektori $\boldsymbol{w}$:n komponentit valitaan siten, että pystytään saavuttamaan maksimaalinen ero luokkien $C_k, k \in [0, 1]$ välillä. Kahden luokan tapauksessa luokkien C_k keskiarvovektorit saadaan kaavasta

$$\overline{\boldsymbol{x}}_k = \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{x}_k,$$

jossa N_k on aineiston havaintojen lukumäärä luokassa C_k. [25, 26]

LDA:ta voidaan käyttää havaintojen luokitteluun. Erotteluanalyysissa redusoidaan aineiston dimensiota projektoimalla havainnot siten, että luokkien väliset etäisyydet, eli luokkien välinen varianssi $\sigma^2_{\text{luokkien välillä}}$, maksimoituvat samalla kun jokaisen luokan sisäinen varianssi $\sigma^2_{\text{luokan sisällä}}$ minimoituu. Luokkien väliset etäisyydet saadaan yksinkertaisesti kaavasta

$$\overline{x}_0 - \overline{x}_1 = \boldsymbol{w}^T (\overline{\boldsymbol{x}}_0 - \overline{\boldsymbol{x}}_1),$$

jossa $\overline{x}_k$ on luokan C_k havaintojen keskiarvo [26]. Tämä voidaan siis merkitä

$$\overline{x}_k = \boldsymbol{w}^T \overline{\boldsymbol{x}}_k. \tag{4}$$

Luokan k sisäinen varianssi s_k voidaan kirjoittaa muodossa

$$s_k^2 = \sum_{n \in C_k} (y_n - \overline{x}_k)^2, \tag{5}$$

jossa $y_n = \boldsymbol{w}^T \boldsymbol{x}_n$ [26]. Molempien luokkien sisäinen varianssi on yksinkertaisesti $s_1^2 + s_2^2$. Maksimoitava funktio voidaan kirjoittaa muodossa

$$V = \frac{\sigma^2_{\text{luokkien välillä}}}{\sigma^2_{\text{luokan sisällä}}} = \frac{(\overline{x}_0 - \overline{x}_1)^2}{s_1^2 + s_2^2} = \frac{\boldsymbol{w}^T B \boldsymbol{w}}{\boldsymbol{w}^T S \boldsymbol{w}}, \tag{6}$$

missä S on luokkien sisäinen kovarianssimatriisi ja B luokkien välinen kovarianssimatriisi. B saadaan kaavasta

$$B = (\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_0)(\overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_0)^T$$

ja S kaavasta

$$S = \sum_{n \in C_0} (\boldsymbol{x}_n - \overline{\boldsymbol{x}}_1)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_1)^T + \sum_{n \in C_1} (\boldsymbol{x}_n - \overline{\boldsymbol{x}}_2)(\boldsymbol{x}_n - \overline{\boldsymbol{x}}_2)^T.$$

Kaavan (6) viimeinen muoto on johdettu käyttämällä kaavoja (3), (4) ja (5). Kertoimet w_i valitaan siis derivoimalla kaavaa (6) $\boldsymbol{w}$:n suhteen siten, että funktio maksimoituu. Tämän jälkeen kerrotaan saatu lauseke S^{-1}:llä, ja löydetään vektori $\boldsymbol{w}$, joka maksimoi V:n. Kahden luokan tapauksessa

$$\boldsymbol{w} = S^{-1}(\overline{x}_0 - \overline{x}_1).$$

Erotteluanalyysi pyrkii siis löytämään sellaiset luokittelusäännöt, joilla luokitteluvirheiden määrä on mahdollisimman pieni. [24, 25, 26] Lineaarinen erotteluanalyysi on tässä työssä toteutettu Pythonin scikit-learn -kirjaston avulla [17].

3.5 k:n osituksen ristiinvalidointi

Luokittelijan evaluointiin voidaan käyttää ristiinvalidointia. Yksi yleisimmistä mallin evaluoinnin ja valinnan menetelmistä on k:n osituksen ristiinvalidointi. Siinä aineisto käydään läpi k kertaa siten, että jokaisella iteraatiolla k osaan jaetusta aineistosta otetaan yksi osa testausjoukoksi ja loput $k-1$ osaa kuuluvat tällöin opetusjoukkoon. Malli sovitetaan opetusjoukkoon jokaisella iteraatiolla. [27] Ajatuksena on, että jokainen aineiston havainto kuuluu testausjoukkoon jollakin iteraatiolla. Lopputuloksena saadaan k erilaista mallia, jotka on sovitettu osittain eri opetusjoukkoon. [28]

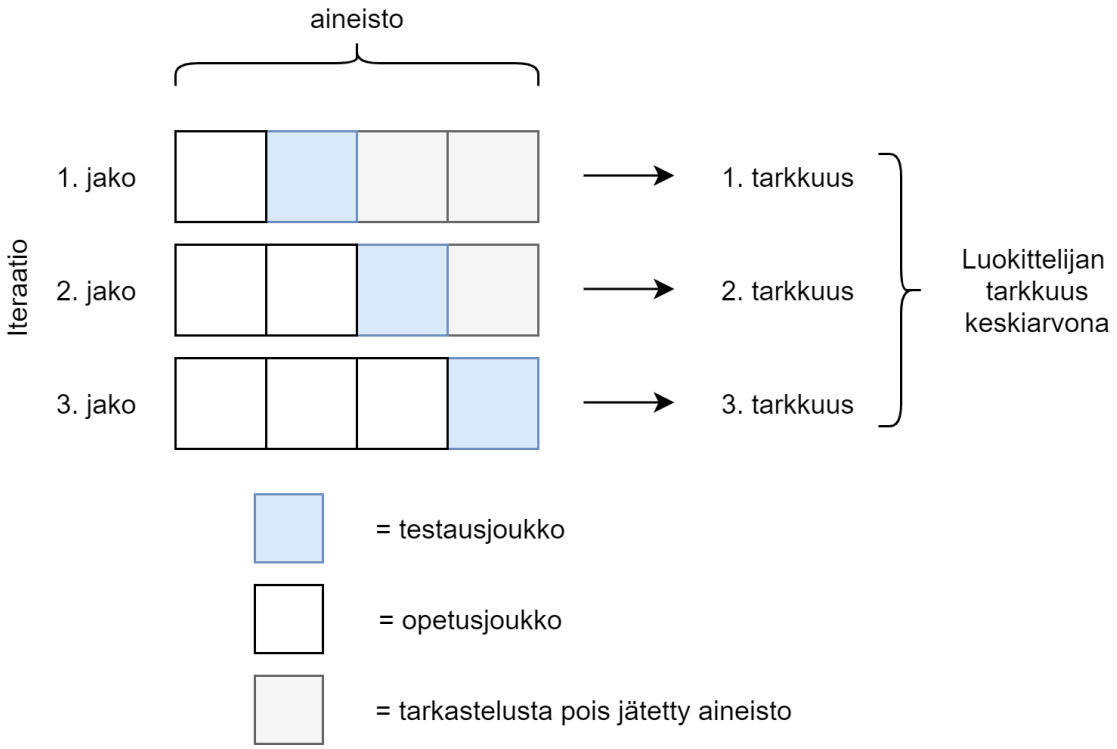
Mallin sovituksessa ja arvioinnissa halutaan välttää ylisovitusta. Ristiinvalidoinnin avulla voidaan havaita mallin ylisovitus ja arvioida mallin kykyä ennustaa uudesta aineistosta [29]. Ristiinvalidoinnissa ylisovituksesta kertoo esimerkiksi eri iteraatioiden tarkkuuksien suuri varianssi. Iteraatioiden suorituskykyä arvioitiin tarkkuuden perusteella. Tarkkuus kertoo, kuinka suuren osuuden annetun joukon havainnoista luokittelija onnistui luokittelemaan oikein. Kaavana tarkkuus voidaan kirjoittaa muotoon

$$\text{tarkkuus} = \frac{\text{oikeat luokittelut}}{\text{kaikki luokittelut}}.$$

Työssä käytetyssä aineistossa on aikariippuvuuksia, joiden takia k:n osituksen ristiinvalidoinnista on käytetty aikasarjoille soveltuvaa variaatiota. Testausjoukon jälkeinen aineisto jätetään variaatiossa huomioimatta. Jos esimerkiksi opetus- ja

testausjoukon jakaisi satunnaisesti, jäisi testausjoukkoon havaintoja, jotka luokittelija saattaisi helposti täydentää, kun opetusjoukkoon kuuluu havaintoa edeltävää ja myöhempää aineistoa. Työssä käytettiin aikasarjoille soveltuvaa viiden osituksen ristiinvalidointia, jossa opetusjoukoksi otettiin ainoastaan testausjoukkoa edeltävää aineistoa.

Kuva 3 havainnollistaa kolmen osituksen ristiinvalidoinnin etenemistä käytetyllä variaatiolla. Aineisto jaetaan kolmella tavalla testausjoukkoon sekä opetusjoukkoon, ja jokaisella iteraatiolla lasketaan luokittelijan tarkkuus testausjoukosta. Lopullisen mallin tarkkuutta voidaan vertailla toisiin malleihin iteraatioiden tarkkuuksien keskiarvona.



Kuva 3: Esimerkki kolmen osituksen ristiinvalidoinnista mallin evaluoinnissa. Osa havainnoista on jätetty huomiotta tässä variaatiossa aineiston aikariippuvuuksien takia.

4 Aineiston kuvaus

Analyysissä käytetty havaintoaineisto koostui autoihin kiinnitetyillä sensoreilla kerätyistä havainnoista vuosien 2019 ja 2020 ajalta. Kaikki tutkimuksen aineisto on peräisin sensoreja asentavalta yritykseltä, RoadCloudilta. Aineiston ensimmäinen havainto on vuoden 2019 tammikuun alusta ja viimeinen vuoden 2020 huhtikuun lopusta. Aineiston tarkastelu rajattiin Göteborgin ja Landvetterin lentokentän väliselle moottoritielle, josta oli paljon havaintoja. Moottoritiesegmentti on esitetty

kuvassa 4. Tiesegmentin nopeusrajoitus on 100 km/h ja aikavälillä segmentiltä oli noin 13 miljoonaa datapistettä. Havainnot pystytään liittämään tieverkkoon Pythonin OSMnx-kirjastolla ja sensorit keräävät aineistoa lähinnä ammattiajossa olevista ajoneuvoista, kuten takseista ja linja-autoista. [30] Liikenteen tilan tarkastelussa tiesegmentti jaettiin itään ja länteen kulkeviin ajoneuvoihin, sillä moottoritien ruuhka saattaa olla vain toiseen suuntaan. Koska osa datasta ajoittuu vuoden 2020 koronakriisin kanssa yhteen, saattaa vuoden 2020 aineistossa olla poikkeavuutta edelliseen vuoteen verrattuna. Vuosien 2019 ja 2020 kevään havaintoja vertailtaessa ei kuitenkaan huomattu suuria eroja. Kuukausien keskinopeudet olivat samaa suuruusluokkaa ja vuodelta 2020 oli enemmän aineistoa kuukausittain, mikä todennäköisesti johtuu vain lisääntyneestä sensoriajoneuvojen määrästä.

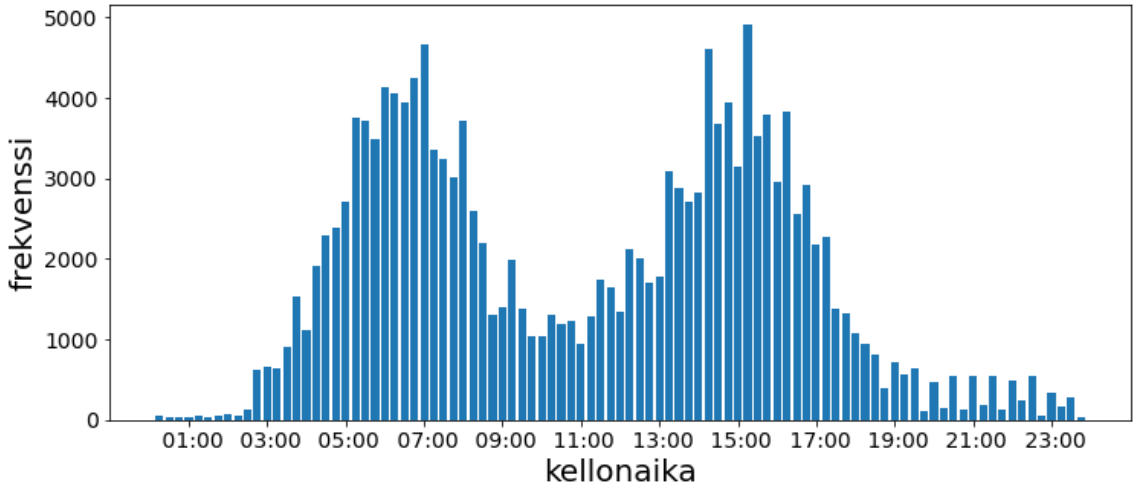
Kuva 4: Kuvassa punaisella merkittynä tiesegmentti Göteborgin ja Landvetterin lentokentän välillä, jolle aineiston käsittely rajattiin. Lentokentän sijainti on merkitty sinisellä nuolella. Tiesegmentin kokonaispituus on noin 25 kilometriä.

Havaintoaineisto sisältää tiedon tarkasta ajasta, pituus- ja leveysasteesta, nopeudesta, kiihtyvyydestä sekä suunnasta. Lisäksi aineistosta käy ilmi tien kunto, ilman lämpötila sekä kitkakerroin, jotka kertovat sääolosuhteista. Tien kunto on kategorinen muuttuja, jota kuvaavat kategoriat kuiva 'D', kostea 'M', märkä 'W', liukas 'S', jäinen 'I', loskainen 'L' sekä tulviva 'P'. Olennaisten muuttujien tilastollisia tunnuslukuja on esitetty taulukossa 1. Taulukon tunnusluvut on laskettu koko aineistosta, jossa yksittäisestä autoista saattaa olla useita havaintoja. Aineistossa on myös informaatiota, jota ei tutkimuksessa käytetty, kuten tieto kiihtyvyydestä x-, y- ja z-suunnissa, auton kallistus sekä tieto sumuvalojen, tuulilasinpyyhkimien ja hätävilkkujen tilasta. Havainnot ovat itsenäisiä pisteitä ajoneuvoista mitattuna, eivät esimerkiksi mittausasemalta tienvarrelta.

Taulukko 1: Taulukossa on esitetty havaintoaineistoa luonnehtivia tilastollisia tunnuslukuja eri muuttujille.

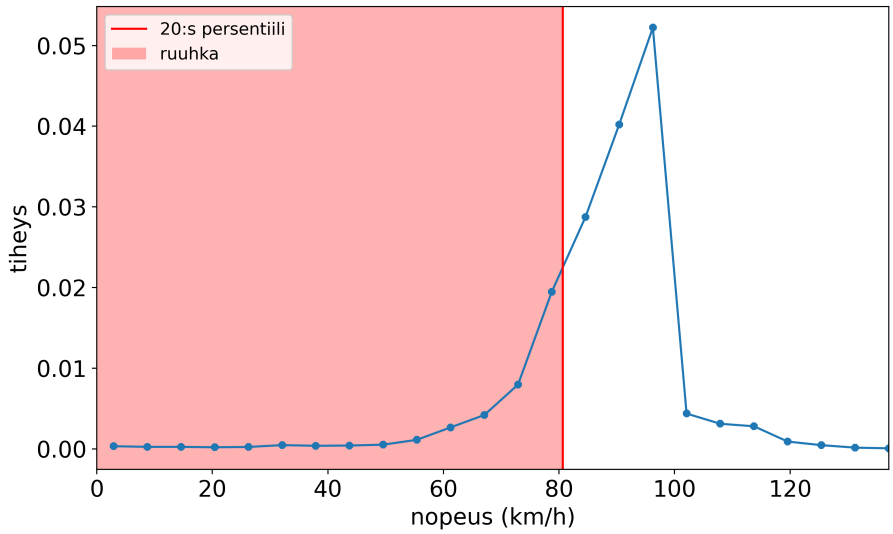
	keskiarvo	mediaani	keskihajonta	minimi	maksimi	vinous
nopeus (km/h)	87,3	90,3	12,2	0	220	-2,0
kiihtyvyys (km/h^2)	0,094	0,056	0,62	-7,22	3,25	0,88
kitkakerroin	0,81	0,85	0,069	0,073	0,85	-2,1
lämpötila (°C)	8,7	7,0	6,1	-9,0	35	0,58

Liikenteen tilaa analysoitiin 15 minuutin aikasegmenteissä, sillä se esiintyi muualla kirjallisuudessa ja todettiin relevantiksi aikaväliksi ruuhkan muodostumiselle. Ajoneuvojen lukumääräjakauma aikasegmenteittäin on esitetty kuvassa 5, josta nähdään että ajoneuvoja on moottoritiellä eniten aamu- ja iltapäivisin. Aamupiikki osuu noin kello seitsemään ja iltapäivällä autoja on eniten neljän aikaan. Ruuhkan havaitseminen aineistosta, josta tarkat automäärät eivät käy ilmi, osoittautui hankalaksi. Esimerkiksi yksittäisen ajoneuvon pysähtyminen tien reunaan saattaa näkyä aineistossa ruuhkana. Havaintoja ei myöskään voinut niiden suuren lukumäärän takia käydä käsin läpi. Oikeastaan päädyttiinkin ennustamaan poikkeavaa liikenteen tilaa objektiivisen ruuhkan sijaan. Tästä huolimatta tekstissä puhutaan jatkossa ruuhkasta.

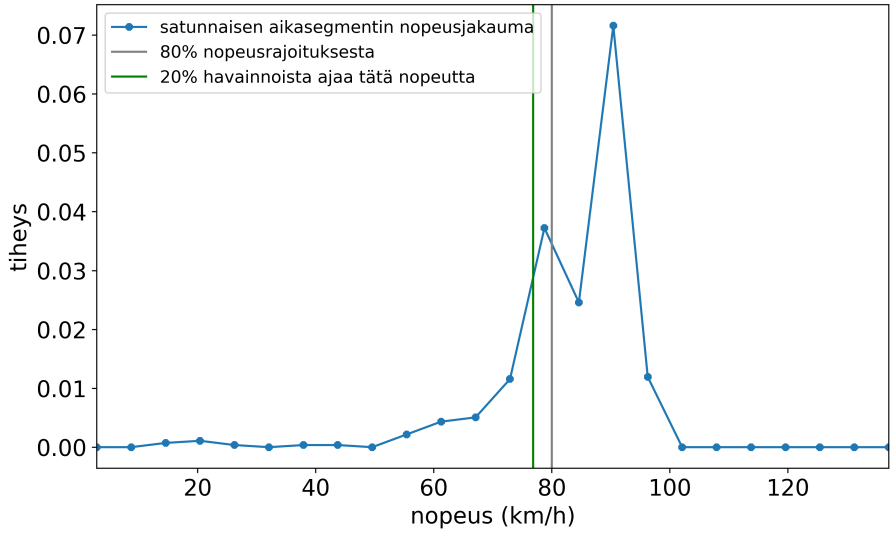


Kuva 5: Koko aineiston yksittäisten ajoneuvojen lukumäärä kellonajan suhteen 15 minuutin aikasegmenteissä.

Ruuhkan määritelmä toteutettiin nopeuteen perustuen kahdella tavalla, jotka molemmat riippuvat parametreista. Ensimmäisessä määritelmässä, joka nimettiin jakaumamääritelmäksi, ruuhkaksi määriteltiin sellainen aikasegmentti, jonka kaikkien datapisteiden keskiarvo oli pienempi kuin jokin n:s persentiili koko tiesegmentin kaikesta aineistosta. Jakaumamääritelmää on havainnollistettu kuvassa 6. Toisessa määritelmässä, häntämääritelmässä, ruuhkaksi määriteltiin sellainen aikasegmentti, jonka kaikista datapisteistä vähintään $p\%$ ajaa nopeutta, joka on alle $x \times$ nopeusrajoitus tiesegmentillä. Häntämääritelmää on havainnollistettu kuvassa 7.



Kuva 6: Esimerkki ruuhkan jakaumamääritelmästä parametrilla $n = 20$. Jos aikasegmentin keskinopeus on alle kuvaan piirretyn 20. persentiilin rajan, luokitellaan segmentti ruuhkaiseksi. Käyrä kuvaa koko aineiston nopeuksien tiheysjakaumaa.



Kuva 7: Ruuhkan häntämääritelmä havainnollistettuna. Parametreillä $p = 20\%$ ja $x = 0{,}8$ luokiteltuna segmentti on ruuhkainen, sillä vähintään 20% havainnoista nopeus on alle 0,8 × nopeusrajoitus 100 km/h. Sininen käyrä kuvaa satunnaisen aikasegmentin nopeuksien tiheysjakaumaa.

Koska optimaalisia parametrejä oli aluksi vaikea määrittää, kerättiin määritelmien mukaiset luokitukset useilla eri parametriarvoilla. Näistä arvoista otettiin keskiarvo,

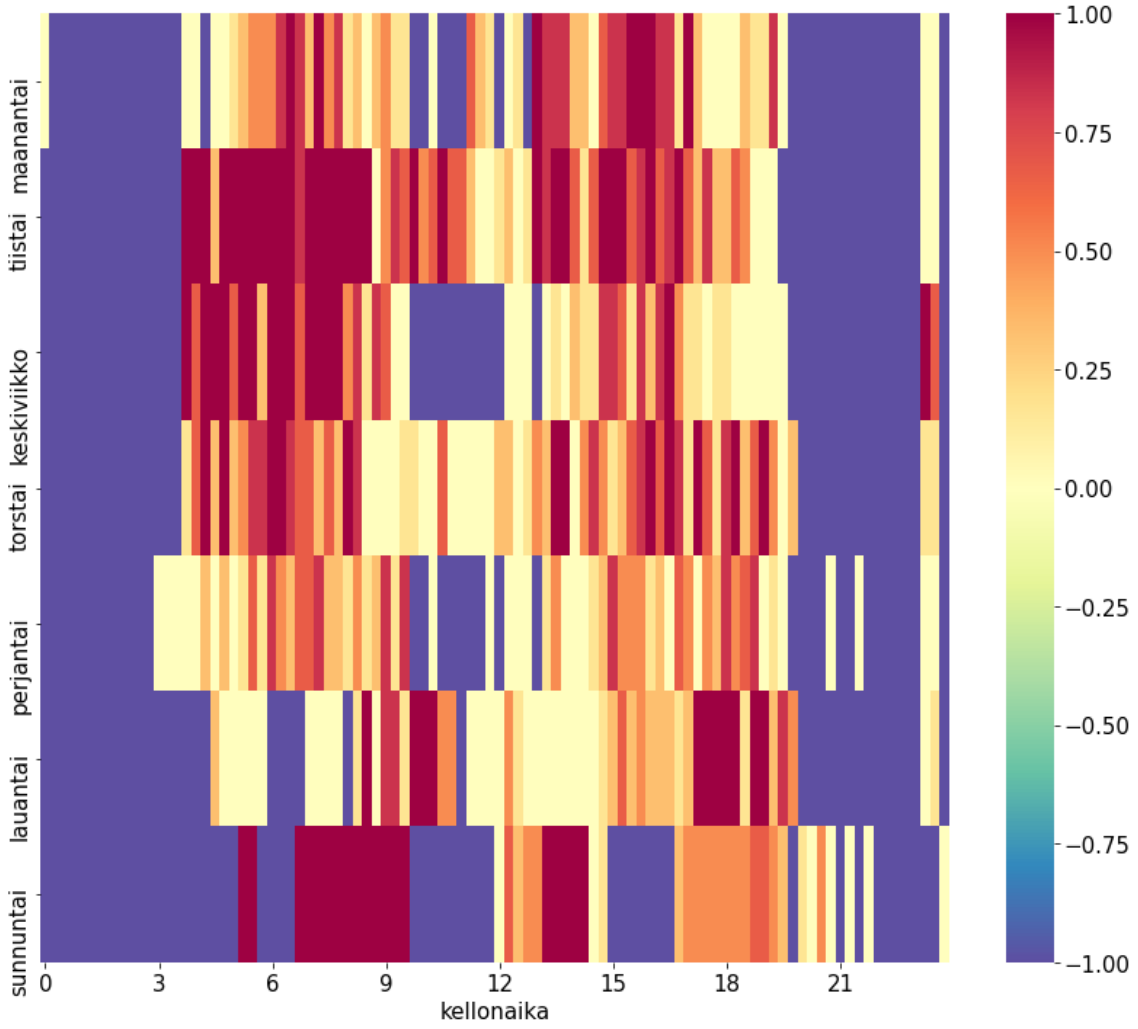
joka kuvastaa kuinka useilla eri parametriarvoilla aikasegmentin datapisteet täyttävät ruuhkan määritelmän. Näin saadaan siis luku väliltä $[0, 1]$, jossa 1 tarkoittaa, että kaikilla käytetyillä parametriarvoilla segmentti voidaan määritellä ruuhkaiseksi. Taulukossa 2 on esitelty ennustamisvaiheessa käytetyt parametrit.

Taulukko 2: Työssä ruuhkan määrittelemiseen käytettyjä parametrejä.

Häntämääritelmä, n	Jakaumamääritelmä, (p, x)
20	(20; 0,8)
30	(30; 0,8)
40	(30; 0,9)

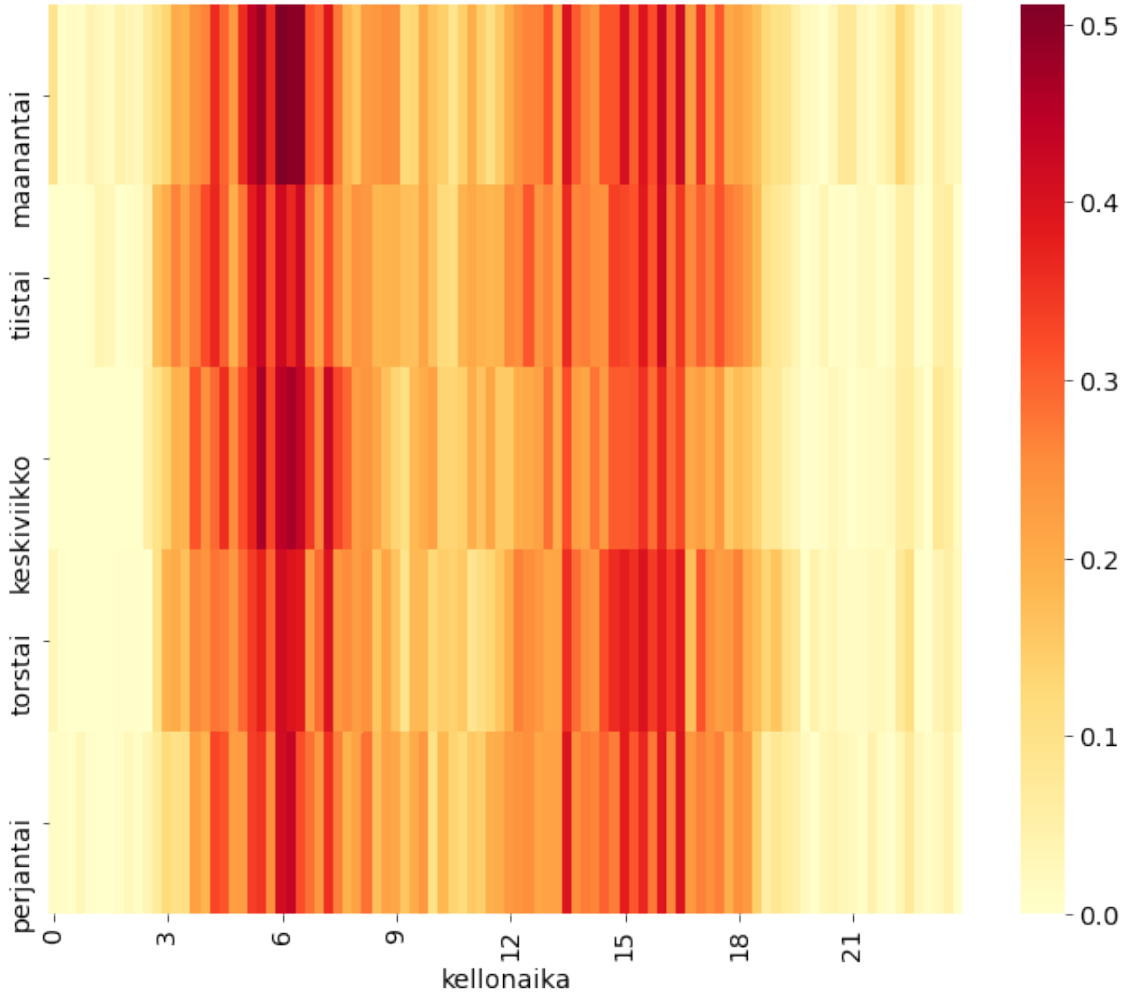
Haasteeksi aineiston käsittelyssä nousivat aikasegmentit, joilla dataa oli vähän. Yksittäiset havaintopisteet saivat esimerkiksi päivittäisen aineiston näyttämään usein siltä, että keskellä yötä olisi hyvinkin ruuhkaisia aikasegmenttejä. Aamuyön ruuhka on kuitenkin ilmiönä harvinainen ja liittyy todennäköisesti vain johonkin yksittäiseen yleisötapahtumaan, GPS-hajapisteeseen tai muuhun sensorivirheeseen. Analyysissa huomattiin siis, että riittämätön aineisto saattaa aiheuttaa määritelmällistä ruuhkaa aikasegmenteille, joissa sitä todennäköisesti ei ole. Esimerkiksi kuvasta 8 nähdään, että joulukuun 2019 toisen viikon sunnuntaiaamu näyttää varmalta ruuhkalta aamuviiden aikaan. Kuvassa on käytetty ruuhkaisuuksien laskemiseen taulukossa 2 esitettyjä parametrejä. Tämä saattaa johtua esimerkiksi pysäkillä seisovasta bussista, josta data on kerätty, mutta varsinaista selitystä ei sunnuntaiaamun ruuhkille löytynyt. On myös mahdollista, että aineistossa on virhehavaintoja, jotka todellisuudessa sijaitsevat toisella tiesegmentillä, jolla nopeusrajoitus on pienempi. Tällöin ne tulkitaan virheellisesti ruuhkaisina. Myös huono sää saattaa vaikuttaa ajonopeuksiin, mutta tämä havaitaan sensoridatasta.

Riittämätön havaintojen määrä ratkaistiin asettamalla ruuhkan määritelmän antaviin funktioihin palautusarvoksi -1, mikäli aikasegmentillä on alle 25 datapistettä. Lisäksi aineistoa siivottiin ottamalla ainoastaan arkipäivien havainnot mukaan analyysiin, sillä toistuvat ruuhkat tapahtuvat suurin piirtein samoihin kellonaikoihin arkipäivisin. Voidaan myös olettaa, että viikonloppuruuhkat ovat harvinaisia ja satunnaisia. Lisäksi puuttuvat arvot, kuten aikasegmentit, joilta puuttuu tieto lämpötilasta, tien kunnosta tai nopeuksista, yksinkertaisesti poistettiin aineistosta koneoppimismallia koulutettaessa. Myös muita lähestymistapoja, kuten interpolaatiota, kokeiltiin, mutta puuttuvia arvoja oli liikaa, jotta estimaateista saataisiin luotettavia.

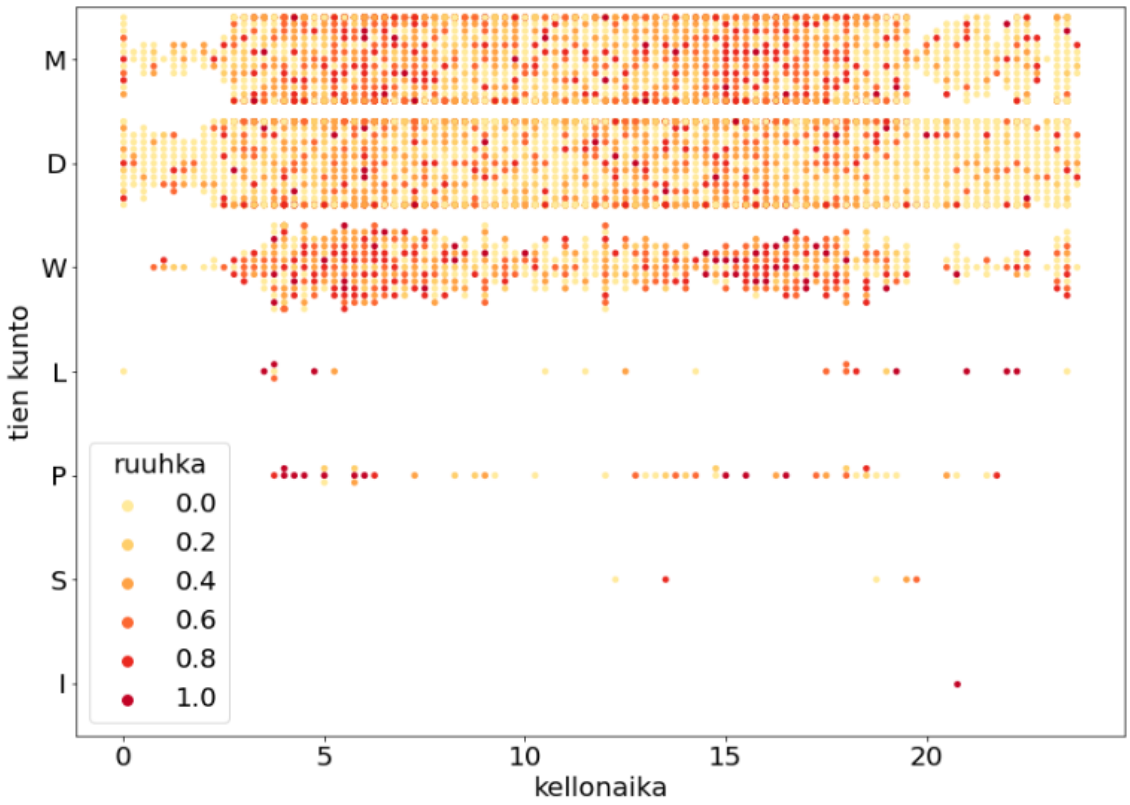


Kuva 8: Vuoden 2019 joulukuun viikon 9.-15.12. 15 minuutin aikasegmentit ruuhka-arvojen kanssa piirrettyinä. Väriasteikolla -1 tarkoittaa, ettei aikavälillä ole riittävästi aineistoa. Väriasteikon väli $[0, 1]$ taas kuvaa aikasegmentille laskettua ruuhka-arvojen keskiarvoa.

Aineistoa kuvaavia piirteitä ovat esimerkiksi edellä kuvatuilla määritelmillä lasketun ruuhkan painottuminen aikaiseen aamuun sekä iltapäivään. Ilmiö näkyy esimerkiksi kuvassa 9, jossa ruuhkaisimmat segmentit osuvat jokaisena arkipäivänä samoille kellonajoille, alkaen jo noin aamuneljästä. Sinänsä aikainen ruuhka on uskottavaa lentokentän läheisyydessä, sillä lentokenttien ruuhkaisimmat kellonajat sijoittuvat usein arkiaamuihin. Samalla huomataan, että keskimääräisesti arkipäivien ruuhkat eivät ole täysin samoihin aikoihin. Moottoritien datapisteitä tutkittaessa havaittiin, että suurimman osan ajasta tien kunto on kuiva, eli 'D', kuten kuvasta 10 voidaan havaita. Moottoriteiden kunto pidetään yleensä hyvänä, joten on loogista että tie on suurimman osan vuodesta hyvässä kunnossa ajamista varten. Samasta kuvaajasta nähdään, että aikavälin 04:00-20:00 ulkopuolella on vain vähän datapisteitä.



Kuva 9: Vuoden 2019 arkipäivien aineistosta jokaiselle 15 minuutin aikasegmentille lasketut keskiarvoiset ruuhkaisuudet kuvattuna kellonajan kanssa. Väriasteikko kuvaa segmentille laskettujen ruuhka-arvojen keskiarvoa ruuhkan jakaumamääritelmällä siten, että tummempi arvo kertoo ruuhkaisemmasta segmentistä. Jakaumamääritelmän parametreinä tässä on käytetty $n = 10, n = 20, n = 30, n = 40$ sekä $n = 50$.



Kuva 10: Tien kunto, kellonaika sekä ruuhkataso vuoden 2019 datasta 15 minuutin aikasegmenteissa tarkasteltuna. Värijakauma kuvaa segmentille laskettujen ruuhka-arvojen keskiarvoa ruuhkan jakaumamääritelmällä siten, että suurempi arvo kertoo ruuhkaisemmasta segmentistä. Kuvaajassa on useita samanarvoisia pisteitä toistensa ylä- tai alapuolelle piirrettynä.

Aineistosta haluttiin erotella samoista ajoneuvoista kerättyjä havaintoja, jotta voitaisiin tarkastella liikenteen tilan ennustamista yksittäisen ajoneuvon perusteella. Erityisesti reaaliajassa yhden auton havaintojen perusteella koko segmentin liikenteen tilan ennustaminen olisi hyödyllistä. Yksittäiset havainnot yhdistettiin toisiinsa riittävän samankaltaisen sijainnin, nopeuden sekä suunnan perusteella. Yksittäisten ajoneuvojen ruuhkaisuuksia verrattiin samojen aikasegmentin ruuhkaisuuksiin, ja niiden todettiin olevan osittain linjassa. Tämä nähdään taulukoissa 3 ja 4, joissa on esitetty yksittäisten autojen ja koko aikasegmentin ruuhkaisuuksien samankaltaisuutta. Sellaisia ajoneuvoja, joista kerättyjä havaintoja oli korkeintaan yksi, oli aineistossa paljon. Näiden ajoneuvojen ruuhkaisuudet eivät korreloineet koko aikasegmentin kanssa yhtä hyvin kuin niiden ajoneuvojen, joista oli paljon havaintoja, joten ne jätettiin pois tarkastelusta. Taulukoista nähdään, että suurin osa yksittäisistä ajoneuvoista antaa saman ruuhkaisuuden kuin koko aikasegmentin kaikki havainnot.

Taulukko 3: Yksittäisten aikasegmenttien ruuhkaisuudet verrattuna kyseisen 15 minuutin aikasegmentin ruuhkaisuuteen. Taulukossa itään päin kulkevat ajoneuvot, joiden ruuhkaisuudet on laskettu häntämääritelmällä parametreillä $p = 30$ ja $x = 0{,}8$.

	Yksittäinen auto	
Koko aikasegmentti	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	77,4%	22,6%
Ruuhkaa	30,0%	70,0%

Taulukko 4: Yksittäisten aikasegmenttien ruuhkaisuudet verrattuna kyseisen 15 minuutin aikasegmentin ruuhkaisuuteen. Taulukossa länteen päin kulkevat ajoneuvot, joiden ruuhkaisuudet on laskettu häntämääritelmällä parametreillä $p = 30$ ja $x = 0{,}8$.

	Yksittäinen auto	
Koko aikasegmentti	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	74,4%	25,6%
Ruuhkaa	29,2%	70,8%

5 Tulokset

5.1 Koko aikasegmentin tulokset

Ennustamisvaiheessa valikoitiin ruuhkan määritelmille optimaaliset parametrit siten, että 15 minuutin aikasegmentin tai yksittäisen ajoneuvon ruuhkan taso 1 kuvaa poikkeavaa liikennettä ja 0 tavanomaista liikennettä. Ennusteet laskettiin molemmilla määritelmillä kolmella eri parametrilla, jotka on esitetty taulukossa 2. Koneoppimismalleina käytettiin AdaBoostia, jonka heikko luokittelija oli yhden syvyinen päätöspuu, tukivektorikonetta lineaarisella ydinfunktiolla sekä lineaarista erotteluanalyysia. Kahta ensimmäistä käytettyä menetelmää on aiemmin käytetty onnistuneesti ruuhkan tunnistuksessa tai ennustamisessa. [2, 4] Lineaarinen erotteluanalyysi valittiin tulosten perusteella kilpailukykyiseksi verrokiksi AdaBoostille. Tukivektorikoneen parametrin suhteen päädyttiin oletusarvoon $C = 1$, sillä sen vaihtelu ei vaikuttanut merkitsevästi tuloksiin.

Opetus- ja testausjoukon attribuutit luokittelijoille koottiin aineiston piirteiden perusteella. Liikenteen ruuhkaisuuteen vaikuttavia ympäristön muuttujia ovat säätila, kellonaika, tien kunto ja erityiset lomapäivät [3]. Siispä luokittelijalle annettava aineisto järjesteltiin siten, että mukaan annettiin tieto kahden edellisen aikabinin ruuhkaisuudesta, lämpötilasta ja tien kunnosta. Lisäksi nykyisen ajanhetken kellonaika, viikonpäivä sekä kuukausi sisällytettiin opetusjoukkoon, sillä esimerkiksi kuvasta

9 nähdään ruuhkaisuuden selvää vaihtelua kellonajan ja viikonpäivän mukaan. Lisäksi kuukausi kertoo vuodenajasta, joka saattaa näkyä ajokäyttäytymisen muuttumisena esimerkiksi vaihtelevien sääolosuhteiden myötä. Eri attribuuttiyhdistelmiä kokeiltiin ja niiden vaikutusta luokittelijoiden tarkkuutteen vertailtiin, mutta tämä yhdistelmä jäi lopulliseksi.

Taulukossa 5 on esitetty kaikki viiden osituksen ristiinvalidoinnin avulla saadut keskitarkkuudet keskihajonnan kanssa. Taulukoissa 6-8 on esitetty 15 minuutin aikasegmenttien ennusteet testausjoukon, eli viimeisen ristiinvalidointikierroksen, pohjalta. Kokonaisten aikasegmenttien testausjoukkojen tarkkuudet eri parametreillä molempiin suuntiin löytyvät liitteeestä B.

Taulukko 5: Kokonaisten 15 minuutin aikasegmenttien liikenteen tilan ennusteiden 5:n osituksen ristiinvalidoinnin keskitarkkuudet ja keskihajonnat molempiin suuntiin. Korkein tarkkuus on merkitty vihreällä jokaiselle riville.

Itä (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (78%)	70,1±2,2	77,7±2,8	77,0±3,7
n=30 (62%)	59,4±2,0	62,3±5,5	63,2±2,8
n=40 (57%)	57,7±1,5	58,7±2,6	59,2±2,5
p=20, x=0,8 (58%)	59,6±1,2	61,9±2,5	60,6±2,4
p=30, x=0,8 (71%)	64,9±2,4	70,6±4,3	70,5±3,6
p=30, x=0,9 (72%)	65,3±3,5	69,7±2,5	71,8±3,8
Länsi (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (74%)	72,5±3,5	77,2±3,3	74,9±4,1
n=30 (59%)	61,2±2,4	63,9±3,7	62,6±3,7
n=40 (60%)	56,3±6,9	63,1±3,1	59,7±5,5
p=20, x=0,8 (55%)	59,4±2,0	62,8±2,8	60,3±2, 9
p=30, x=0,8 (67%)	65,9±2,5	70,7±3,6	68,5±4,6
p=30, x=0,9 (77%)	59,1±1,7	76,2±3,2	76,6±2,6

Taulukko 6: Länteen päin kulkevien kokonaisten aikasegmenttien ruuhkaisuusennusteet häntämääritelmän parametreilla $p = 20$ ja $x = 0{,}8$ AdaBoost-luokittelijalla. Suuremman luokan osuus aineistosta oli 55%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**77,7%**	22,3%
Ruuhkaa	43,9%	**56,1%**

Taulukko 7: Länteen päin kulkevien kokonaisten aikasegmenttien ruuhkaisuusennusteet häntämääritelmän parametreilla $p = 20$ ja $x = 0{,}8$ tukivektorikoneella luokiteltuna. Suuremman luokan osuus aineistosta oli 55%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**47,9%**	52,1%
Ruuhkaa	30,8%	**69,2%**

Taulukko 8: Länteen päin kulkevien kokonaisten aikasegmenttien ruuhkaisuusennusteet häntämääritelmän parametreilla $p = 20$ ja $x = 0{,}8$ lineaarisella erotteluanalyysilla luokiteltuna. Suuremman luokan osuus aineistosta oli 55%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**77,5%**	22,5%
Ruuhkaa	54,7%	**45,3%**

5.2 Yksittäisten ajoneuvojen tulokset

Yksittäisten ajoneuvojen opetusjoukkoon otettiin mukaan tiedot kyseisen ajoneuvon tallentamasta lämpötilasta ja tien kunnosta. Ajoneuvon nopeushavaintojen avulla määritettiin ruuhkan taso ja lisäksi kellonaika, kuukausi ja viikonpäivä. Ennustettava muuttuja oli seuraavan 15 minuutin aikasegmentin ruuhkaisuus samoilla ruuhkaparametreilla. Jotta tulokset olisivat vertailukykyisiä, käytettiin samoja luokittelijoita sekä ruuhkan määritelmien parametrejä kuin kokonaisille aikasegmenteille. Yksittäisten ajoneuvojen ristiinvalidointitarkkuudet on esitetty taulukossa 9. Taulukoissa 10-12 on esitetty yksittäisten autojen havaintojen perusteella saadut ennusteet länteen päin kulkeville ajoneuvoille jakaumamääritelmällä. Yksittäisten ajoneuvojen testausjoukon tarkkuudet on esitetty liitteen B taulukossa.

Taulukko 9: Yksittäisten ajoneuvojen liikenteen tilan ennusteiden ristiinvalidointitarkkuudet ja keskihajonnat molempiin suuntiin. Korkein tarkkuus on merkitty vihreällä jokaiselle riville.

Itä (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (79%)	65,4±14,4	78,8±2,8	78,9±3,0
n=30 (63%)	60,0±6,7	64,3±3,1	64,2±3,9
n=40 (55%)	55,2±5,1	58,5±3,4	58,0±3,2
p=20, x=0,8 (58%)	58,7±2,0	59,7±4,6	61,0±3,0
p=30, x=0,8 (72%)	61,9±9,1	71,7±4,0	71,8±3,9
p=30, x=0,9 (60%)	53,5±12,5	69,2±5,6	68,6±4,3
Länsi (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (76%)	64,7±13,1	67,0±25,3	77,8±5,9
n=30 (62%)	60,0±6,2	58,9±15,0	63,6±6,8
n=40 (57%)	52,7±8,4	58,6±5,6	54,2±7,4
p=20, x=0,8 (57%)	57,7±4,7	57,0±11,6	59,7±4,8
p=30, x=0,8 (69%)	61,3±8,0	62,3±20,8	70,9±6,9
p=30, x=0,9 (63%)	47,2±13,5	70,5±6,6	66,8±9,8

Taulukko 10: Länteen päin kulkevien yksittäisten autojen liikenteen tilan ennusteet jakaumamääritelmän parametrilla $n = 30$ AdaBoost-luokittelijalla. Suuremman luokan osuus aineistosta oli 61,9%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**84,8%**	15,2%
Ruuhkaa	56,0%	**44,0%**

Taulukko 11: Länteen päin kulkevien yksittäisten autojen liikenteen tilan ennusteet jakaumamääritelmän parametrilla $n = 30$ tukivektorikoneella luokiteltuna. Suuremman luokan osuus aineistosta oli 61,9%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**51,8%**	48,2%
Ruuhkaa	34,6%	**65,4%**

Taulukko 12: Länteen päin kulkevien yksittäisten autojen liikenteen tilan ennusteet jakaumamääritelmän parametrilla $n = 30$ lineaarisella erotteluanalyysilla luokiteltuna. Suuremman luokan osuus aineistosta oli 61,9%.

	Ennustettu luokka	
Todellinen luokka	Ei ruuhkaa	Ruuhkaa
Ei ruuhkaa	**91,1%**	8,9%
Ruuhkaa	85,4%	**14,6%**

5.3 Tulosten analysointi

Tukivektorikone pärjäsi kaikissa luokitteluongelmissa huonommin kuin AdaBoost ja lineaarinen erotteluanalyysi. Lineaarinen erotteluanalyysi ja AdaBoost taas saavuttivat samantasoiset tarkkuudet, vaikka menetelminä ne ovat hyvin erilaisia. Erotteluanalyysi etsii aineiston piirteistä mahdollisimman hyvin luokat erottavan kombinaation redusoimalla dimensiota, kun taas AdaBoost yhdistää heikkoja luokittelijoita koostaakseen vahvan luokittelijan. Tukivektorikoneen optimaalinen päätöstaso taas ei kyennyt erottelemaan aineistoa yhtä hyvin kuin muut luokittelijat. Tukivektorikoneen lineaarinen ydinfunktio antoi parhaat tulokset, mutta koska se ei silti yltänyt muiden luokittelijoiden tasolle, voidaan ajatella ettei aineisto ole lineaarisesti separoituvaa. Kuitenkin keskihajontoja taulukosta 5 tarkastellessa voidaan todeta, etteivät luokittelijat usein edes ole tilastollisesti merkittävästi toisiaan tarkempia.

Suhteellisesti parhaat tarkkuudet saatiin silloin, kun suuremman luokan osuus oli vain hieman yli puolet. Esimerkiksi taulukossa 5, kun suuremman luokan osuus on 77%, ei yksikään luokittelija saa tätä korkeampaa tarkkuutta. Kun toisen luokan osuus on suuri luokittelijat ennustavat suurempaa luokkaa. Jos opetusjoukon luokkien osuuksia tasaa, ei sillä vaikuta olevan selvää vaikutusta tarkkuuksiin ainakaan tukivektorikoneella, jolla tätä kokeiltiin. Lisäksi länsisuuntaan olevat ennusteet vaikuttavat hieman paremmilta kuin itään päin olevat. Tämä johtuu todennäköisesti siitä, että Göteborgin keskusta on lännessä, joten sinne on liikennettä ennalta odotettavampiin aikoihin kuin kaupungista pois päin itään.

Yksittäisten autojen aineiston perusteella tehtyjen ennusteiden tarkkuudet ovat hieman pienempiä kuin koko segmentin avulla ennustetut. Tämä oli odotettavissa, sillä suuri osa yksittäisistä ajoneuvoista koostui vain muutamasta havainnosta, jolloin ennusteet täytyy tehdä hyvin pienen havaintomäärän pohjalta. Toisaalta muutamankin havainnon lämpötila ja tien kunto todennäköisesti vastaa saman aikasegmentin keskiarvoa ja osa aikasegmenteistäkin koostuu vain muutamasta ajoneuvosta.

Mikään luokittelijamalli ei lopulta onnistunut saavuttamaan huomattavasti parempaa tarkkuutta kuin suuremman luokan ennustaminen. Algoritmejä optimoimalla, esimerkiksi parametrejä säätämällä tai luokittelijoita yhdistelemällä, voitaisiin saavuttaa korkeampia tarkkuuksia. Tuloksia kuitenkin vertailtiin sellaiseen luokittelijaan, joka arvaa jokaiselle päivän 15 minuutin aikasegmentille sen keskimääräistä ruuhkai-

suutta. Näin saadut ennusteet eivät myöskään olleet merkittävästi parempia kuin koneoppimisluokittelijoiden, joten voidaan todeta että luokittelu on jollakin tasolla haastavaa. Luokittelijat myös ennustivat oikein usein lähes pelkästään suurempaa luokkaa, kuten esimerkiksi taulukossa 12 nähdään. Tämä saattaa kertoa siitä, että aineistossa oli paljon kohinaa, jonka takia luokkien erottelu on hankalaa.

Eri muuttujien vaikutusta tarkkuuksiin kokeiltiin yksi kerrallaan. Kaikki opetusjoukkoon valitut muuttujat vaikuttivat tarkkuuksiin positiivisesti. Voidaan siis päätellä, että kellonaika, kuukausi ja viikonpäivä vaikuttavat ruuhkaisuuteen. Lisäksi tämä kertoo, että tien kunto ja lämpötila vaikuttavat ajamisnopeuksiin. Pienemmällä aineistolla ennusteista tuli epävarmempia. Joillakin aikaväleillä saatettiin saada todella hyviäkin ennusteita, mutta koska ruuhka on myös vuoden ajasta riippuvaista, ei näin saatuja ennusteita voitu pitää yhtä luotettavina eikä tuloksia pystytä toistamaan samoilla tarkkuuksilla muilla aikaväleillä. Näitä vaihteluita voi helposti tutkia tarkkuuksien keskihajonnoista, jotka olivat esimerkiksi yksittäisille autoille välillä melko suuria. Vaihtelut olivat kuitenkin suurimman osan ajasta melko pieniä, muutaman prosentin luokkaa. Luonnollisesti tarkkuudet pääsääntöisesti kasvoivat, kun ristiinvalidoinnissa opetusjoukon koko kasvoi.

Aineiston puutteellisuus oli suurin ennustustarkkuutta rajoittava tekijä. Puuttuvia arvoja oli erityisesti lämpötilassa ja tien kunnossa. Aineistosta jouduttiin pudottamaan useita rivejä pois tämän vuoksi. Lisäksi, koska aineisto koostui lähinnä ammattiajossa olevista ajoneuvoista, saattaa sen antama informatio olla vääristynyttä. Usein ruuhka-ajat osuvat arkipäivien työmatkaruuhkiin, mutta töihin kuljetaan usein taksin sijaan autolla. Sensoriajoneuvojen avulla kerätyn aineiston tarkkuus riippuukin siitä, kuinka suuri osa ajoneuvoista on varustettu sensorilla. Liikenneruuhkan havaitsemiseen riittää jo pienempi määrä sensoriautoja, mutta matalan virtauksen tilan tulkitsemiseen tarvitaan suurempi osuus sensoriautoja [5]. Tämä huomattiin käytännössä aineistosta helposti, sillä usein aineistossa näytti olevan ruuhkaisia aikasegmenttejä kellonaikoina, joina ruuhkaa tuskin on. Nämä segmentit osoittautuivat sellaisiksi, joissa havaintoja oli hyvin vähän, jolloin ruuhkaisuutta ei voida päätellä. Aiemmin todettiinkin, että riittävän tarkan nopeuden arvioimiseen tarvitaan simulaatioiden perusteella tarkasteltavaa tiesegmenttiä kohden vähintään kymmenen sensoriajoneuvoa aikavälille, mikä ei tämän tutkimuksen havaintoaineistossa usein toteudu [4].

Aineiston puuttellisuutta voitaisiin täydentää esimerkiksi lisäämällä ulkopuolista aineistoa, kuten tiekameran havaintoja liikenteen määrästä ja nopeuksista samalta tienpätkältä. Ulkopuolisella datalla liikenteen määrästä voitaisiin myös validoida aiemmin esitellyt ruuhkan määritelmät. Tämän työn laajasta aineistosta on myös todennäköisesti mahdollista löytää lisää muuttujia, joista ruuhkaa voi tulkita. Analyysistä jätettiin ulos esimerkiksi kiihtyvyys ja kitka, koska niiden mitta-asteikot olivat turhan harvoja tarkempaan analyysiin. Näistä saatettaisiin kuitenkin löytää riippuvuuksia ruuhkaan, jolloin niitä voitaisiin käyttää osana ruuhkan määritelmiä. Päällimmäiset parannusehdotukset siis olisivat aineiston laajuuden ja tarkkuuden parantaminen, aineiston täydentäminen ulkopuolisella datalla, uusien ruuhkaa indikoivien muuttujien tutkiminen sekä koneoppimisalgoritmien laajempi testaus ja optimointi.

Tulokset antavat kuitenkin pohjan sensoripohjaisen aineiston avulla saatavien tarkkojen liikenteen tilan ennusteiden saavuttamiseen. Aineiston puuttellisuudesta huolimatta liikenteen tilaa saatiin eroteltua ja ennustettua kaikilla käytetyillä ennustamismenetelmillä, vaikka tarkkuudet eivät olekaan yhtä suuria kuin samanlaisella aineistolla on mahdollista saavuttaa. [5] Keskinopeuksiin ja nopeuksien jakaumiin perustuvien ruuhkan määritelmien avulla saatiin jaoteltua aineistoa segmentteihin, joissa liikenteen tila on normaalia tai poikkeavaa. Liikenteen tilan ennustamisen kannalta tulokset kertovat myös, että yksittäisten ajoneuvojen pohjalta pystytään tekemään lähes yhtä tarkkoja ennusteita kuin laajemmasta aineistosta. Päätelmät eivät suoraan ole yleispäteviä, sillä tutkimusaineisto oli tarkasti rajattu yhdelle moottoritielle.

6 Yhteenveto

Tässä kandidaatintyössä analysoitiin sensoripohjaista liikenneaineistoa ja ennustettiin liikenteen tilaa. Aluksi vertailtiin aiemman kirjallisuuden perusteella erilaisilla menetelmillä liikenneaineistojen ominaisuuksia. Datan käsittelyä, analyysia ja ennustamista varten kirjoitettiin Python-ohjelma. Ennustamisessa käytettiin kolmea koneoppimisalgoritmia: AdaBoostia, tukivektorikonetta sekä lineaarista erotteluanalyysia. Ruuhkalle määriteltiin kaksi nopeuteen pohjautuvaa määritelmää, joita käytettiin ennustamisessa aika- ja sääinformaation ohella. Tiesegmentti jaettiin itä- ja länsisuuntaan ja aineistosta eroteltiin tutkimusta varten myös yksittäisien ajoneuvojen havaintoja. Liikenteen tilaa tutkittiin 15 minuutin aikasegmenteissä.

Parhaat tarkkuudet luokittelijoista saatiin, kun ruuhkaisen ja ruuhkattoman tilan osuudet aineistossa olivat suunnilleen yhtä suuria. Ongelmaksi epätasaisessa aineistossa koitui se, että luokittelijat ennustivat vain suurempaa luokkaa. Kaupunkiin päin kulkevan moottoritien puolen tarkkuudet olivat hieman korkeampia kuin toisen suunnan. Yksittäisten ajoneuvojen avulla saadut ennusteet olivat lähes yhtä hyviä kuin kokonaisten aikasegmenttien, vaikka niistä oli paljon vähemmän havaintoja.

Saadut ennustustarkkuudet eivät olleet yhtä korkeita kuin kirjallisuudessa samankaltaisella aineistolla on saavutettu. [5] Tulokset indikoivat kuitenkin, että tutkimusaineiston pohjalta pystytään erottelemaan ruuhkaista ja ruuhkatonta liikenteen tilaa nopeuteen perustuvilla ruuhkan määritelmillä, sillä tarkkuudet olivat satunnaista tai pelkän suuremman luokan arvausta parempia. Wang ja Tsai [4] ja Cheu et al. [11] toteavat, että sensoriaineiston suurimpia heikkouksia on riittävä havaintojen määrä. Saadut tulokset ovat linjassa näiden tutkimusten kanssa, sillä ruuhkan määritteleminen osoittautui kuitenkin hankalaksi aineiston puutteellisuuden vuoksi.

Aineiston puutteellisuutta voitaisiin paikata esimerkiksi tiekameran avulla saadulla aineistolla, jolloin ruuhkan määritteleminen helpottuisi. Työssä käytetyt ruuhkan määritelmät voitaisiin myös validoida ulkopuolisen aineiston lisäyksellä. Tiesegmentti, jota työssä käsiteltiin, oli tarkasti rajattu. Tulosten yleispätevyyden todentamiseksi tulisi tutkimusta siis laajentaa myös muille tiesegmenteille, vaikka se ei tämän työn rajoitetun aineiston puitteissa ollutkaan mahdollista.

Viitteet

[1] David Schrank, Bill Eisele, ja Tim Lomax. Urban mobility report 2019. *Texas AM Transportation Institute, College Station*, Aug 2019. URL https://trid.trb.org/view/1647076.

[2] Xiaolei Di, Yu Xiao, Chao Zhu, Yang Deng, Qinpei Zhao, ja Weixiong Rao. Traffic congestion prediction by spatiotemporal propagation patterns. Teoksessa *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, sivut 298–303, Jun 2019. doi: 10.1109/MDM.2019.00-45.

[3] Yunxiang Liu ja Hao Wu. Prediction of road traffic congestion based on random forest. Teoksessa *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, sivut 361–364, 2017.

[4] Chenqi Wang ja Hsin-Mu Tsai. Detecting urban traffic congestion with single vehicle. Teoksessa *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, sivut 233–240. IEEE, 2013.

[5] Corrado de Fabritiis, Roberto Ragona, ja Gaetano Valenti. Traffic estimation and prediction based on real time floating car data. Teoksessa *2008 11th International IEEE Conference on Intelligent Transportation Systems*, sivut 197–203, Oct 2008. doi: 10.1109/ITSC.2008.4732534.

[6] W. Hongsakham, Wasan Pattara-atikom, ja Ratchata Peachavanish. Estimating road traffic congestion from cellular handoff information using cell-based neural networks and k-means clustering. Teoksessa *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 1, sivut 13–16, May 2008. doi: 10.1109/ECTICON.2008.4600361.

[7] Waze.inc (2020). Waze social gps, maps & traffic [mobile software application]. Retrieved Aug 2020. URL https://www.waze.com/.

[8] Tomio Miwa, Yusuke Tawada, Toshiyuki Yamamoto, ja Takayuki Morikawa. En-route updating methodology of travel time prediction using accumulated probe-car data. Teoksessa *11th ITS World Congress*, 2004.

[9] Siebe Turksma. The various uses of floating car data. Teoksessa *Tenth International Conference on Road Transport Information and Control, 2000. (Conf. Publ. No. 472)*, sivut 51–55, 2000.

[10] Billy M Williams ja Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

[11] Ruey Long Cheu, Chi Xie, ja Der-Horng Lee. Probe vehicle population and sample size for arterial speed estimation. *Computer-Aided Civil and Infrastructure Engineering*, 17(1):53–60, 2002. doi: 10.1111/1467-8667.00252. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-8667.00252.

[12] Ethem Alpaydin ja Francis Bach. *Introduction to Machine Learning*. MIT Press, 2014. ISBN 9780262325745. URL http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=3339851.

[13] Bernhard E Boser, Isabelle M Guyon, ja Vladimir N Vapnik. A training algorithm for optimal margin classifiers. Teoksessa *Proceedings of the fifth annual workshop on Computational learning theory*, sivut 144–152, 1992.

[14] William S. Noble. What is a support vector machine? *Nature Biotechnology*, 24(12):1565–1567, Dec 2006. doi: 10.1038/nbt1206-1565. URL https://www.nature.com/articles/nbt1206-1565.

[15] Corinna Cortes ja Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[16] Chang Chih-Chung Hsu, Chih-Wei ja Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[17] Fabian Pedregosa, Gaël Varoquaux, Gramfort Alexandre, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[19] Yoav Freund ja Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[20] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. Teoksessa *icml*, volume 96, sivut 148–156. Citeseer, 1996.

[21] Yoav Freund, Robert Schapire, ja Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[22] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2): 197–227, 1990.

[23] Eun Bae Kong ja Thomas G Dietterich. Error-correcting output coding corrects bias and variance. Teoksessa *Machine Learning Proceedings 1995*, sivut 313–321. Elsevier, 1995.

[24] Ming Li ja Baozong Yuan. 2D-LDA: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters*, 26(5):527–532, 2005.

[25] Brian S Everitt ja Graham Dunn. *Applied multivariate data analysis*. Englewood Cliffs, N.J.: Prentice Hall, 1992. ISBN 978-0-470-71117-0.

[26] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information science and statistics. Springer, 1st ed. 2006. corr. 2nd printing edition, 2006. ISBN 9780387310732,0387310738.

[27] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

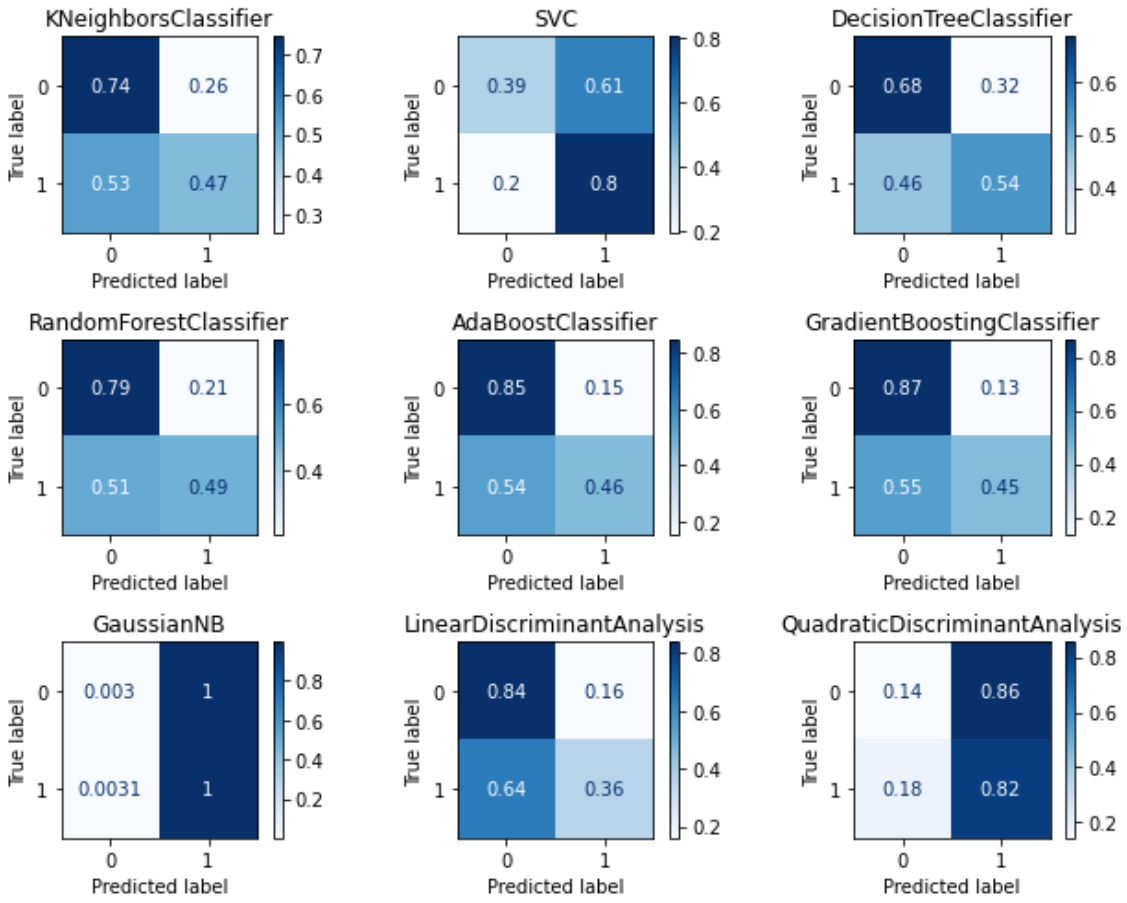
[28] Juan D Rodriguez, Aritz Perez, ja Jose A Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575, 2009.

[29] Gavin C Cawley ja Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

[30] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers Environment and Urban Systems*, 65:126–139, 07 2017. doi: 10.1016/j.compenvurbsys.2017.05.004.

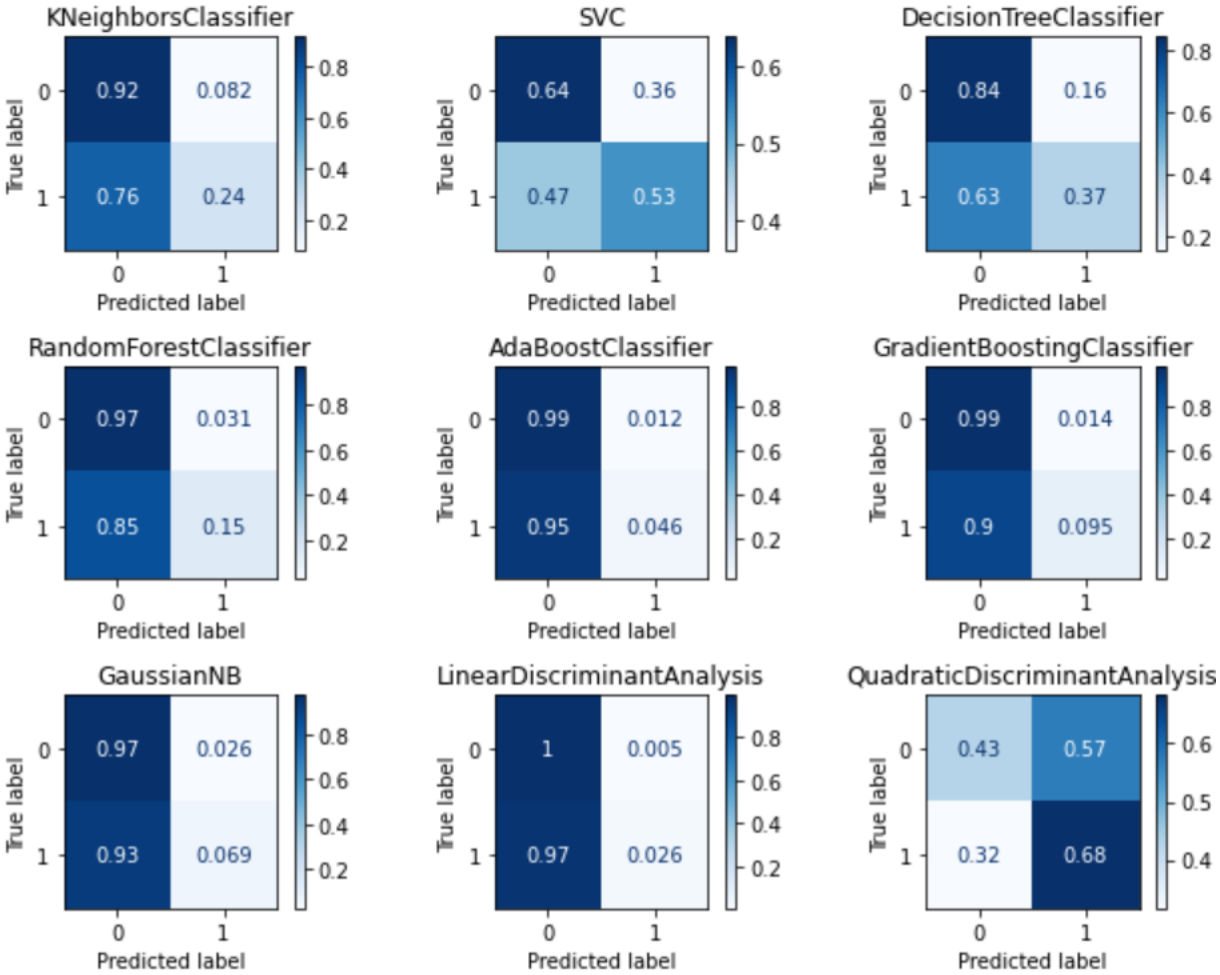
A Luokittelijoiden vertailu

A.1 Jakaumamääritelmän luokittelijoiden vertailu



Kuva A1: Erilaisten koneoppimisluokittelijoiden vertailua liikenteen tilan ennustamiseen työssä määritellyn jakaumamääritelmän perusteella länteen päin kulkevien autojen perusteella. Parametrina on käytetty $n = 20$. 59% aineistosta on suurempaa luokkaa, joka on 0 eli liikenteen tila ei ole poikkeava.

A.2 Häntämääritelmän luokittelijoiden vertailu



Kuva A2: Erilaisten koneoppimisluokittelijoiden vertailua ruuhkan ennustamiseen työssä määritellyn häntämääritelmän perusteella itään päin kulkevien autojen perusteella. Parametrina on käytetty $p = 30$ ja $x = 0{,}8$. 71% aineistosta on suurempaa luokkaa, joka on 0 eli liikenteen tila ei ole poikkeava.

B Koneoppimismenetelmien tarkkuudet

Taulukko B1: Kokonaisten aikasegmenttien testausjoukon tarkkuudet.

Itä (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (78%)	71,3	79,8	80
n=30 (62%)	60,6	65,5	65
n=40 (57%)	57	58	58,6
p=20, x=0,8 (58%)	59,5	62,6	60,4
p=30, x=0,8 (71%)	66,1	73,2	73,2
p=30, x=0,9 (72%)	64	71,9	72,1
Länsi (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (74%)	72,3	77,7	73,9
n=30 (59%)	63,4	68,5	63,8
n=40 (60%)	60,9	65,2	62,6
p=20, x=0,8 (55%)	62	68	63
p=30, x=0,8 (67%)	66,8	71,3	68
p=30, x=0,9 (77%)	69,7	77,9	77,9

Taulukko B2: Yksittäisten ajoneuvojen testausjoukon tarkkuudet.

Itä (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (79%)	59	79,1	79,3
n=30 (63%)	58,3	65,1	64,7
n=40 (55%)	61,2	61,8	60,7
p=20, x=0,8 (58%)	58	62,6	61,1
p=30, x=0,8 (72%)	58,3	71,5	71,5
p=30, x=0,9 (60%)	61,4	73,6	69,7
Länsi (suuremman luokan osuus)	**SVM (%)**	**AdaBoost (%)**	**LDA (%)**
n=20 (76%)	59,4	79,7	79,1
n=30 (62%)	60,2	71,5	66,2
n=40 (57%)	59,2	60,3	53,9
p=20, x=0,8 (57%)	59	66,8	61,9
p=30, x=0,8 (69%)	58,6	74,1	71,7
p=30, x=0,9 (63%)	58,6	64,5	62,9