Merika Peltola

JKL-Openin raportointityökalun kehittäminen

Tietotekniikan pro gradu -tutkielma

3. toukokuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Merika Peltola

Yhteystiedot: merika.m.peltola@student.jyu.fi

Ohjaaja: Timo Hämäläinen

Työn nimi: JKL-Openin raportointityökalun kehittäminen

Title in English: Improving the reporting tool on JKL-Open

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 73+13

Tiivistelmä: Tutkielmassa pyrittiin kehittämään uusi ja paranneltu versio raportointiyökalusta, joka on JKL-Open-sivustolla. Nykyinen raportointityökalu on todettu vaillinaiseksi, koska se rikkoutuu aina kun kuu vaihtuu keskellä viikkoa. Työkalussa ei myöskään huomioida kaikkea kerättyä dataa vaan esimerkiksi autot lasketaan vuoden 2009 tutkimuksen perusteella. Tutkielmassa käytettiin tutkimusmenetelmänä konstruktiivista tutkimusotetta, jossa tarkoituksena on luoda ratkaisu tosielämän ongelmaan.

Teoriaosassa tutkittiin erilaisia sensoreita, joita voidaan käyttää liikenteen havainnoimiseen. Tarkoituksena oli löytää parhaat mahdolliset sensorit, joiden avulla saadaan paljon laadukasta ja kattavaa dataa Jyväskylän kaupungin liikenteestä. Teoriaosiossa tutkittiin myös liikenteen mallintamista, jotta saataisiin selville, kuinka avointa dataa tulisi esittää ja kuinka yleisesti kerätään avointa dataa liikenteestä. Käytännön osassa esitellään, kuinka toinen tutkielman varsinaisista tuloksista, uusi raportointityökalu JKL-Open-sivustolle, toteutettiin.

Tutkielmassa tutkittiin neljää erilaista menetelmää datan analysoimiseen sekä siitä ennustamiseen. Käytännön osiossa on esitelty analysoinnin tuloksia ja kuinka SARIMA:lla ja TensorFlow:lla onnistuuttiin tekemään ennusteita tulevasta liikenteestä. Kalman-suodattimen avulla pyrittiin poistamaan häiriötekijät datasta ja iteraatiokierrosten avulla antamaan arvio tulevasta datasta samalla päivittäen edellistä tilaa. PCA:lla pystyttiin onnistuneesti tunnistamaan kerätystä datasta eri parkkihallit eli datan alkulähteet.

Tutkielmassa kehitettiin onnistuneesti uusi raportointityökalu, jolla JKL-Open-sivuston vanha raportointityökalu tullaan korvaamaan. Samalla tutkielmassa saatiin onnistuneesti tehtyä muutamia data-analyysejä ja ennustuksia. Data-analyysien ja ennustuksien tarkkuus kärsi hieman datan vähyydestä ja koronaviruksen aiheuttamista liikenteen rajoittamisista.

Avainsanat: avoin liikennedata, sensori, kulkutapajakauma, liikennelaskenta, tekoäly

Abstract: The goal of this thesis was to develop a new and improved version of the reporting tool that is on JKL-Open. The current version has been deemed incomplete because it breaks every time the month changes in the middle of the week. The tool doesn't take into consideration all the data that has been collected instead cars are calculated based on a research from 2009. The research method in this thesis is constructive research, where the goal is to create a solution for a real life problem.

One part of the theory consists of research on different types of sensors that can be used to detect traffic. The goal is to find the best possible sensors that can be used to collect a lot of high quality data that also has a high coverage on the traffic of Jyväskylä. Another part of the theory consists of research on modeling traffic, where the goals are to find out how open data should be presented and how open traffic data is collected in general. The practical part of the thesis presents how one of the results, the new reporting tool was developed.

Four different methods for data analysis and prediction were researched in this thesis. the practical part describes the results of the analysis and how predictions on upcoming traffic were successfully made with SARIMA and TensorFlow. The goal of using the Kalman filter was to eliminate the background noise and give an estimate of the next state as well as update the current state. By applying PCA to the data, parking garages were successfully recognised.

In the thesis a new reporting tool for JKL-Open was successfully created. Also a few data analyses and predictions were done successfully. The results and their preciseness were not the best due to the amount of data available. The corona virus also played a part on the preciseness of the data as there were restrictions that affected traffic, so the predictions were not able to take that into account.

Keywords: open traffic data, traffic flow, sensor, traffic counting, artificcial intelligence

Termiluettelo

Azure	Microsoftin tekemä alusta pilvipalveluille, joka mahdollistaa erilaisten palveluiden käyttämisen.
Bluetooth	Lyhyen kantaman radiotekniikkaan perustuva langaton tiedonsiirtotekniikka.
Dynamic Stixel-World	Malli, jonka avulla voidaan esittää videolla tai valokuvassa esiintyvä ympäristö kolmiulotteisesti.
FIR	Lyhenne, joka tulee sanoista far infrared. Yleisesti käytetään kuvaamaan aktiivisia infrapunasensoreita.
FPCA	Functional Principal Component Analysis eli funktionaalisten pääkomponenttien analysointi on menetelmä, jota usein käytetään data-analyysissä dominoivan ilmiön löytämiseen.
Haar wavelet	Matemaatikon mukaan nimetty joukko neliömäisiä funktioita, jotka muodostavat aallon.
Kalman-suodatin	Kalman-suodatin kykenee arvioimaan järjestelmien tilaa aiempien tulosten perusteella.
LSTM	TensorFlow:n menetelmä, joka hyödyntää lyhytaikaista muistia.
MAC	Media Access Control on yksilöivä tunniste verkkolaitteille.
Optimum Closed Cut	Metodi, jossa on optimoitu tarkasti rajattu alue liikenteestä saatavan datan täydentämiseen.
PIR	Lyhenne, joka tulee sanoista passive infrared. Yleisimmät passiiviset infrapunasensorit ovat pyroelektronisia eli ne reagoivat lämpöön.
Power BI	Microsoftin applikaatio, joka on tarkoitettu datan visualisointiin erilaisten raporttien muodossa.
RSSI	Mittaustulos, joka kertoo radiosignaalin voimakkuuden.
Screen-line laskenta	Iso laskenta-alue jaetaan pienempiin osioihin mielivaltaisten viivojen avulla.
SARIMA	Seasonal autoregressive integrated moving average on analyy-

	simalli, jota käytetään usein aikasarjojen kanssa.
TensorFlow	Avoimen lähdekoodin kirjasto, jota käytetään muun muassa sekä koneoppimisen että neuroverkkojen yhteydessä.
Wi-Fi	Yleinen kaupallinen nimitys tuotteille, jotka käyttävät WLAN-teknologiaa. Wi-Fi on Alliancen tavaramerkki, joka perustuu IEEE:n 802.11 standardiin.
WLAN	Wireless local area network eli langaton lähiverkko.

Kuviot

Sisältö

1 Johdanto

Tässä tutkielmassa käsitellään erilaisia sensoreita, jotka on tarkoitettu erilaisten kulkutapojen laskemiseen. Niiden lisäksi tutkielma käsittelee myös liikenteen mallintamista erilaisilla menetelmillä, joissa verrataan eri kulkutavoista kerättyä dataa toisiinsa. Tutkielman tutkimusmenetelmä on konstruktiivinen tutkimusote, jota useimmiten hyödynnetään liiketalouden puolella ("Kari Lukka: Konstruktiivinen tutkimusote" 2014).

Liikenteessä olevien erilaisten kulkuvälineiden seuraaminen sekä liikennelaskennan suorittaminen ovat tärkeä osa kaupunkien kulkuväylien kehittämistä. Näitä tietoja keräämällä saadaan tilastoja eri teiden käyttöasteista, joiden perusteella kyseisiä tieosuuksia voidaan parantaa. Tilastojen avulla voidaan selvittää ruuhkaisia risteyksiä, jotka hidastavat liikennettä ja joita voitaisiin parantaa liikenteen nopeuttamiseksi.

Jyväskylän kaupunki seuraa kaupungissa liikkuvia autoja, polkupyöräilijöitä ja kävelijöitä anonyymisti. Autoja seurataan liikennevaloissa olevien sensorien avulla sekä parkkihallien käyttöasteiden perusteella. Polkupyöräilijöitä seurataan pääsääntöisesti Eco-Counterin sensorien avulla, testauksessa on myös induktiosilmukoihin perustava ratkaisu. Jalankulkijoita seurataan Jyväskylän Yliopiston kehittämien WiFi-laitelaskurien avulla sekä toisinaan tehtävien manuaalisten laskentojen avulla.

JKL-Open-sivustolla on esillä eri sensoreista kerätty avoin liikennedata. Jyväskylän kaupunki on sijoittanut sensoreita sellaisiin kohteisiin, joista on järkevää kerätä liikennetietoja, kuten esimerkiksi risteyksien liikennevalot, alikulut ja suuret pyörätiet. Tämän lisäksi kaupunki kerää tietoa myös linja-autojen käyttöasteesta sekä päivittäisen liikenteen määrästä.

Tutkimus on ajankohtainen, koska keväällä 2020 koronaviruksen takia on asetettu erilaisia rajoitteita, jotka vaikuttavat suoraan liikennemääriin. Asetetut rajoitteet ovat herättäneet ihmisissä kiinnostusta liikenteen seuraamiseen, koska sen avulla voidaan nähdä konkreettinen muutos ihmisten liikkumisessa. Yleisesti tutkimus on hyödyllinen erityisesti liikennesuunnittelun kannalta, koska seuraamalla liikennettä pystytään tunnistamaan mahdolliset ongelmakohdat liikenteen sujuvuudessa, jotta ne voidaan tulevaisuudessa korjata.

1.1 Tutkimusmenetelmä

Tutkimuksessa käytetty tutkimusmenetelmä on konstruktiivinen tutkimusote, joka on yksi Case-tutkimuksen alle kuuluvista metodeista. Useimmiten konstruktiivinen menetelmä on käytössä liiketalouden puolella, mutta sen hyödyntäminen tieteellisessä tutkimuksessa kasvaa jatkuvasti. Konstruktiivisessa tutkimusotteessa tarkoituksena on tarkoituksena ratkaista jokin tosielämän ongelma. Tutkijalla on oltava syvä käytännön ja teorian tuntemus aiheesta, jotta tutkimus voidaan toteuttaa parhaalla mahdollisella tavalla. Konstruktiivisessa tutkimusotteessa läheinen yhteistyö tutkijan ja asiakkaan edustajan kanssa on keskeinen osa tutkimuksen toteuttamista. Tutkimuksen tulisi tuottaa myös jonkinlainen teoreettinen kontribuution tutkittavalla aiheelle. Vaikka tutkimus ei onnistuisi tuottamaan oikeasti toimivaa konstruktiota, voi se silti tuottaa merkittävän akateemisen kontribuution. Konstruktiivisessa tutkimusotteessa tulee siis huomioida sekä ongelman ratkaisu että teoreettisen kontribuution tekeminen. Edellä mainittujen asioiden tasapainottaminen on usein ongelmana konstruktiivisissa tutkimuksissa, välillä tutkijat keskittyvät liikaa toiseen puoleen, jolloin kokonaisuus kärsii (“Kari Lukka: Konstruktiivinen tutkimusote” 2014).

1.2 Tutkimuskysymykset

Tutkimuksen pyritään vastaamaan mahdollisimman kattavasti seuraaviin tutkimuskysymyksiin:

- Miten avointa liikennedataa voidaan hyödyntää eri kulkuvälinetyyppien käyttäjämäärien vertailemiseen sekä kuinka tieto tulisi esittää?
- Miten kerättyä liikennedataa voidaan tarkentaa paremmin vastaamaan todellisuutta?

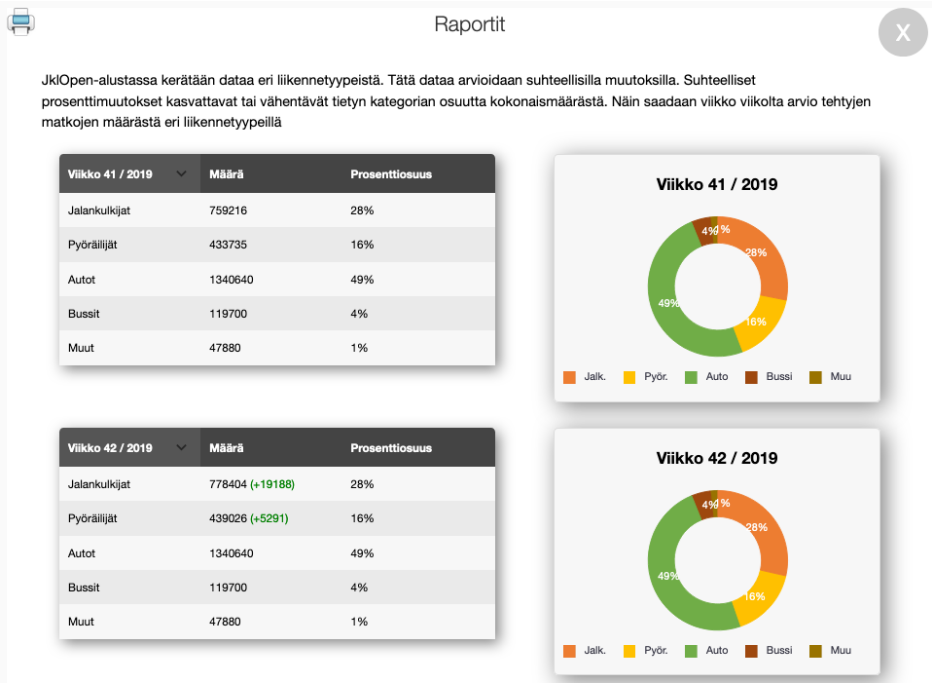
Liikennedataa pyritään tarkentamaan tutkimalla erilaisia sensoreita ja sitä, kuinka niiden havainnointikyky voidaan maksimoida. Sensorivaihtoehtojen tutkimisen ja analysoimisen jälkeen, pyritään lisäämään liikennettä havainnoivia sensoreita Jyväskylän kaupungin alueella. Liikennedatan esittämistä varten tullaan tarkastelemaan ja kokeilemaan erilaisia menetelmiä, kuten esimerkiksi Azuren ja Power BI:n käyttöä sekä JKL-Open-sivuston raportointityökalun taustatietojen tarkentamista. Samalla tarkoituksena on kehittää ja kokeilla erilaisia ratkaisuja, joiden avulla liikenteestä voitaisiin antaa tarkempia suuntaa antavia ennustuksia.

Ennustuksia ja tietoja tulisi pystyä antamaan myös sellaisissa tilanteissa, joissa jokin sensori on poissa käytöstä tilapäisesti esimerkiksi vian tai korjaustöiden takia. Datan analysointiin ja ennustamiseen tullaan käyttämään muutamia yleisesti käytössä olevia menetelmiä, kuten SARIMA ja TensorFlow.

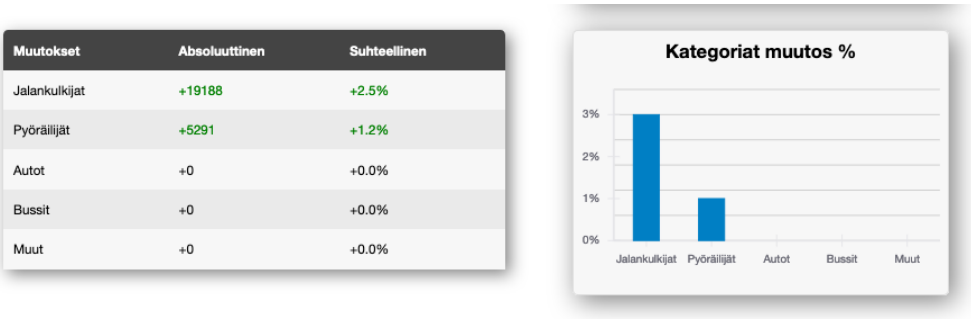
1.3 JKL-Open

JKL-Open on Jyväskylän Yliopiston, Jyväskylän kaupungin ja TNNetin yhteishankkeena kehitetty sivusto. Sivustolla esitetään avointa dataa liikenteestä, jota on kerätty erilaisten sensorien avulla. Sivusto kehittyi Future City Challenge 2018 -kilpailuun tehdyn ratkaisun pohjalta. Nykyisellä sivustolla on käytössä kaksi näkymää, toinen kevyelle liikenteelle ja toinen autoille. Sivustoa on tarkoitus kehittää tulevaisuudessa siten, että otetaan käyttöön kolmas näkymä, joka on omistettu julkiselle liikenteelle. Tutkimuksen alkaessa sivustolle tulee dataa muutamasta Eco-Counterista, WiFi-laitelaskureista, parkkihalleista sekä muutamista liikennevaloista.

JKL-Openissa on raportointityökalu, jonka avulla voidaan viikkotasolla tarkastella eri lähteiden keräämää dataa. Alkuperäinen raportointityökalu on esitetty kuvioissa 1 ja 2. Raportointityökalu toimii sillä periaatteella, että siinä verrataan kahden viikon lukemia toisiinsa. Kahdessa ensimmäisessä taulukossa esitellään eri kulkutavoista sekä havaittu määrä että kulkutavan käyttäjien osuus koko viikon havaitusta liikenteestä. Kolmannessa taulukossa esitellään eri kulkutapojen käyttäjämäärissä tapahtuneet absoluuttiset ja suhteelliset muutokset. Jokaisen taulukon pääsisältö on esitetty myös piirakkakaaviossa, josta kokonaiskuva hahmottuu paremmin.



Kuvio 1. JKL-Openin raportointityökalun alkuosa



Kuvio 2. JKL-Openin raportointityökalun loppuosa

1.4 Data-analyysi ja tekoäly

Informaatioteknologian alalla erilaiset data-analyysit sekä tekoälyyn liittyvät ratkaisut ovat tällä hetkellä hyvin trendikkäitä ja ajankohtaisia. Monet yritykset ovat alkaneet integroida tekoälyä omiin projekteihinsa aina kun se vain on mahdollista sekä yliopistojen ja yritysten välinen yhteistyö tekoälyn ja data-analyysin osalta on kasvanut huomattavasti. Data-analyysissä pyritään perinteisesti tutkimaan joko tiettyä tarkoitusta varten kerättyä dataa tai tietyistä tapahtumista kerättyä dataa. Tutkimuksen avulla on tarkoitus selvittää datassa olevat ominaisuudet, joita voidaan hyödyntää tulevaisuudessa. Yritysmaailmassa pyritään pitkälti saamaan jonkinlaista kilpailuetua ominaisuuksien avulla, kun akateemisessa maailmassa pyritään tuottamaan uutta tietoa ja tutkimusta löydettyjen ominaisuuksien ja ilmiöiden avulla.

Tekoälyn käyttämistä projekteissa ja tutkimuksissa tulee aina harkita kunnolla, koska se ei sovellu jokaiseen projektiin ja siitä saatava hyöty voi joissain tapauksissa olla aiheutuneita kuluja pienempi. Tekoälyn yksi yleisimmin käytetyistä alakategorioista on koneoppiminen, jota hyödynnetään muun muassa ennustamisessa ja datan automaattisessa luokittelussa. Tutkimuksessa hyödynnetään tekoälyä ja koneoppimista tunnistamaan kerätystä datasta eri keräyspaikat sekä ennustamaan tulevia liikennemääriä.

Yrityksellä tai tutkijalla tulee olla käytössä riittävän tehokkaat tietokoneet ja palvelimet, joissa on suuret kapasiteetit laskennalle, jotta koneoppimisesta voidaan hyödyntää. Useimmissa tapauksissa laskentaa varten täytyy joko ostaa useita koneita ja palvelimia tai vuokrata joltain pilvipalvelulta palvelin ja laskenta-aikaa. Harvemmin yritykseltä löytyy valmiiksi tekoälyn käyttämiseen soveltuvia laitteita, jotka eivät ole jatkuvasti käytössä muiden asioiden tekemiseen. Tekoälyn hyödyntämiseksi käytettävää dataa tulee olla kerättynä paljon ja pitkältä ajanjaksolta, jotta siitä voidaan saada irti luotettavia tuloksia. Datan tulee myös olla mahdollisimman lähellä alkuperäistä muoto, jotta matkalle tapahtuneet muutokset eivät pääse vaikuttamaan tuloksiin.

2 Liikennelaskennan keinot

Tässä luvussa kuvataan tarkemmin erilaisia keinoja ja teknologioita kävelijöiden, pyöräilijöiden sekä autojen havainnointiin. Jokaisen teknologian kohdalla käsitellään myös siihen liittyviä yleisesti tiedossa olevia ongelmia. Liikenteen havainnointiin on kehitetty paljon erilaisia sensoreita, mutta luvussa keskitytään yleisimmin käytössä oleviin sensoreihin, joista löytyy tutkimuksia luotettavista lähteistä.

2.1 Jalankulkijat

Jalankulkijoiden havainnoiminen tehdään useimmiten joko konenäön avulla videokuvasta tai manuaalisesti laskemalla. Jalankulkijoita voidaan havaita myös älylaitteiden avulla, käyttäen hyödyksi WiFi:n ja Bluetoothin käytössä lähetettyjä MAC-osoitteita sisältäviä viestejä. Älypuhelinten käyttäjien osuus oli vuonna 2016 jo 60 prosenttia ja nykyinen luku on paljon suurempi. Tästä johtuen älypuhelimia voidaan hyvin käyttää liikennelaskentaan. Yksistään Bluetooth ei riitä antamaan oikeanlaista kuvaa liikenteestä, koska sen avulla voidaan havaita 5-12 prosenttia älylaitteiden käyttäjistä (Lesani ja Miranda-Moreno 2018).

2.1.1 WiFi-laitelaskuri

WiFi-sensoreita käytetään yleisesti kolmenlaisiin tehtäviin: havaitsemaan liikettä, tunnistamaan havainnointialueella olevia laitteita sekä arvioimaan väkijoukkojen henkilömääriä. Kaikissa tapauksissa tulee huomioida häiriötekijät ja suodattaa ne pois datasta. Datan käsittelyyn on olemassa monenlaisia malleja ja algoritmeja, joita käytetään muun muassa deep learning-metodeissa. Joitain valmiita applikaatioita on kehitetty WiFi-signaalien havainnointiin Android-käyttöjärjestelmissä, jotka vaativat tabletin tai puhelimen toimiakseen. Applikaatio pystyy havaitsemaan vain lähellä olevat signaalit ja vaatii käyttäjän, joka käynnistää sovelluksen aina havainnoinnin ajaksi (Shlayan, Kurkcu ja Ozbay 2016).

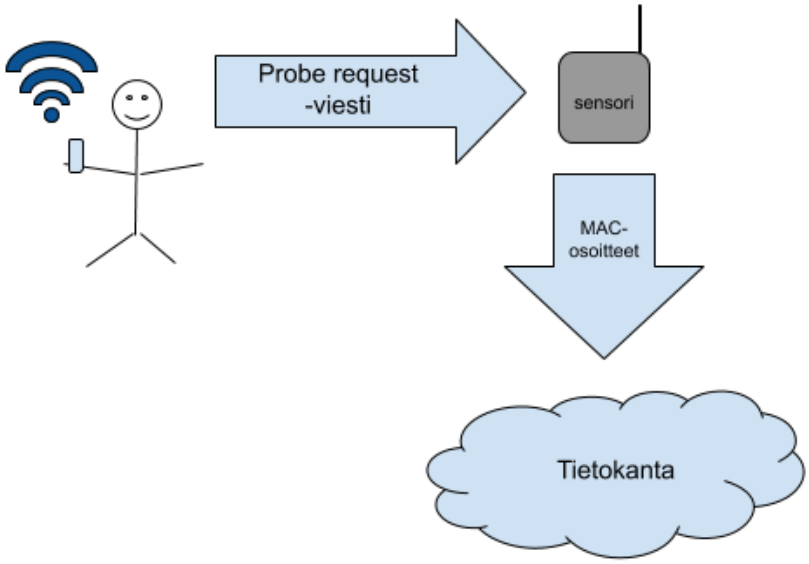
Langattomia reitittimiä voidaan käyttää ihmismäärien havainnointiin. Menetelmä edellyttää, että älylaite on yhdistettynä tiettyyn verkkoon. Mikäli laite ei ole yhdistetty tiettyyn verkkoon, ei sitä voida havainnoida reitittimen avulla. Reittimien käyttäminen ihmisten havain-

nointiin on kuitenkin ongelmallista, koska sen avulla ei voida havainnoida ihmisiä suurelta alueelta. Menetelmä ei myöskään sovellu havainnointialueen laajentamiseen, joten se sopii parhaiten hyvin rajattujen alueiden liikenteen seuraamiseen. Nämä ongelmat ovat johtaneet erillisten WiFi-signaaleja havaitsevien sensoreiden kehittämiseen. Uudet kehitetyt sensorit havaitsevat älylaitteiden lähettämiä probe request -viestejä. Näistä viesteistä kerätään ainutlaatuisia MAC-osoitteita, joiden avulla voidaan tunnistaa yksittäisiä laitteita. Laitteet lähettävät näitä viestejä, vaikka laite ei olisi kyseisellä hetkellä aktiivisesti käytössä. Ainut vaatimus viestien lähettämiselle on se, että älylaitteessa on WiFi päällä. Sensori pyrkii havaitsemaan signaaleja 2.4GHz:n ja 2.5GHz:n väliseltä taajuusalueelta, joka on jaettu 14 erilliseen tarkkailtavaan kanavaan (Lesani ja Miranda-Moreno 2018). Osa WiFiin ja probe request -viesteihin perustuvista menetelmistä jättää kanavan 14 huomioimatta, koska se ei ole käytössä useassa maassa (Oliveira ym. 2019). Kun pyritään havainnoimaan liikennettä monelta kanavalta, täytyy tehdä päätös pyrkiikö yksi laite havainnoimaan liikennettä jokaisella kanavalla vai annetaanko jokaisella kanavalle oma sensorinsa. Jos päädytään malliin, jossa yksi sensori pyrkii havaitsemaan viestejä jokaiselta kanavalta, tulee ajoitukseen kiinnittää erityistä huomiota. Aluksi tulee päättää, kuinka kauan yhtä kanavaa kuunnellaan ja kuinka kauan aikaa kulutetaan yhteen kierrokseen, jonka aikana käydään läpi kaikki kuunneltavat kanavat (Sandrasegaran ym. 2013).

Jalankulkijoita voidaan havaita myös pelkästään WiFi-signaaleja laskemalla. Menetelmää kutsutaan naiiviksi, koska siinä ei oteta huomioon häiriötekijöitä. Kun otetaan käyttöön uudenlaisia sensoreita, tulisi niiden antamia lukemia verrata oikeaksi todistettuihin laskentoihin. Osa puhelimien valmistajista on ottanut käyttöön turvakeinoja, jotta voitaisiin vaikeuttaa probe request -viesteistä saatujen MAC-osoitteiden yhdistämistä tiettyihin laitteisiin. Näihin turvatoimiin lukeutuvat mm. Applen tekemä MAC-osoitteiden satunnaistaminen silloin, kun laite lähettää viestejä eteenpäin. Tämä turvatoimi ei valitettavasti toimi kunnolla tarkoissa ja vaativissa skannauksissa, jotka vieläkin pystyvät saamaan selville oikean MAC-osoitteen (Schauer, Werner ja Marcus 2014).

WiFi probe request -viestien RSSI-lukemien perusteella on hankala arvioida kävelijöiden liikettä paikasta toiseen. Jotta lukemasta olisi hyötyä, täytyy raja-arvo valita oikein ja huomioida mahdolliset häiriöt. Jos raja-arvo on liian korkea, osa kävelijöistä jää havainnoimatta.

Liian matalalla raja-arvolla osa kävelijöistä tulee tunnistetuksi useamman kerran tai sitten kävelijöiksi lasketaan sinne kuulumattomia laitteita. Ongelmaksi muodostuu myös se, että probe request -viestien lähetystaajuuksissa on eroja Androidin ja iOS:n välillä. Jotkin Applen iOS-versiot lähettävät useammin probe request -viestejä kuin osa Android-versioista. Tästä johtuen iOS-laitteet tulevat helpommin havaituiksi (Schauer, Werner ja Marcus 2014). Koska häiriötekijät vaikuttavat WiFi-signaalin voimakkuuteen, täytyy niiden vaikutukset RSSI-lukemiin ottaa huomioon. Muutamia menetelmiä on kehitetty näiden häiriötekijöiden vaikutuksen poistamiseen. Osa menetelmistä on johdettu takaisinmallintamalla häiriötekijöitä sekä niiden vaikutusta WiFi-signaaliin. Yksi häiriötekijöiden vaikutuksen poistamiseen tarkoitettu menetelmä on Ensemble Empirical Mode Decomposition (EEDM), jossa signaaliin lisätään taustakohinaa ja signaali pilkotaan pieniin osiin. Edellä mainittuja askelia toistetaan erilaisilla taustakohinoilla kunnes oikea kaava alkuperäisin signaalin selvittämiseksi löytyy (Sandrasegaran ym. 2013).



Kuvio 3. WiFi-laitteiden probe request -viestien havainnointi sensorin avulla.

WiFi probe request -viestien havainnoimisessa on muutama ongelma, jotka pitää ottaa huomioon liikennelaskentaa tehdessä. Pelkkien viesteistä löydettävien MAC-osoitteiden avulla ei ole helppo erotella jalankulkijoita polkupyöräilijöistä, joten usein apuna joudutaan käyt-

tämään algoritmejä ja muita sensoreita. Sensorit ovat herättäneet myös huolia yksityisyyden suojasta, mutta MAC-osoitteet annetaan verkkolaitteille jo tehtaalla. Tästä syystä osoitteet eivät sisällä mitään henkilökohtaisia tietoja, vaan niiden avulla voidaan ainoastaan yksilöidä verkkolaitteet. Useimmat MAC-osoitteita keräävät sensorit koodaavat osoitteet ennen niiden eteenpäin lähettämistä. Koodaamisen avulla pyritään siihen, ettei kukaan pysty suoraan lukemaan MAC-osoitetta ja samalla yhdistä sitä tietyn henkilön käyttämään laitteeseen. Menetelmä mahdollistaa vielä yksilöllisten tunnisteiden erottamisen, koska jokaisesta MAC-osoitteesta tulee erilainen tiiviste koodauksen seurauksena. Kun osoitteet päätyvät palvelimelle, niitä ei yleensä säilytetä pitkiä aikoja vaan pyritään laskemaan erilaisten osoitteiden lukumäärä. Tämän jälkeen palvelin joko tuhoaa tiedot tai asettaa ne sellaiseen muotoon, josta niitä ei enää voida helposti takaisinmallintaa (Lesani ja Miranda-Moreno 2018).

Yleinen ongelma WiFi:n käyttämisessä jalankulkijoiden havainnointiin on se, että siinä lasketaan laitteita, joilla on WiFi yhteys päällä. Osalla jalankulkijoista on mukanaan useampi WiFi:llä varusteltu laite ja toisilla ei pakosti ole yhtään tällaista laitetta mukana. Osittain tämä tasapainottaa tuloksia, mutta antaa myös mahdollisuuden laskelmien vääristymiselle. Samoin täytyy myös ottaa huomioon, että havaintojen joukossa voi paikasta riippuen olla myös polkupyöräilijöitä tai autoissa olevia ihmisiä.

WiFi:n avulla ihmisten tunnistamista on hyödynnetty paljon erityisesti rakennuksissa ja erilaisissa tapahtumissa, kuten esimerkiksi festivaaleille. Tällaisissa paikoissa WiFi:n käyttö havainnointiin on helppoa, koska kyseessä on rajattu alue, jossa ihmiset usein viipyvät jonkin aikaa. Mitä pidemmän ajan ihmiset viipyvät tietyllä aluella, sitä varmemmin heidän älylaitteidensa probe request -viestit havaitaan. Mikäli festivaaleilla käyviä ihmisiä halutaan havainnoida WiFi:n avulla, käytetään useampaa sensoria. Useampi sensori mahdollistaa kattavamman havainnoinnin festivaalialueelta, joka on yleensä kooltansa suurempi kuin yhden sensorin havainnointialue (Farooq ym. 2015). WiFi-sensoreita voidaan käyttää paikannukseen sisätiloissa, mutta usein tarvitaan jokin toinen sensori parantamaan tuloksia (Liu ym. 2012). Sisätila havainnointi on mahdollista toteuttaa kahden sensorin avulla, jotka asennetaan vinoon siten, että niiden välille muodostuu näkymätön linja. Kun ihminen kävelee sensorien ohi, hänet voidaan havaita ja laskea. Jotta sensori kykenee havainnoimaan rinnakkain käveleviä ihmisiä, tulee sensori asentaa noin 45 asteen kulmaan (Kura, Yamaguchi ja

Shiraishi 2018).

Ihmisten havainnointiin WiFi:n avulla on olemassa kaksi erilaista mallia, aktiivinen sekä passiivinen. Passiivisessa metodissa havaitseva sensori ei kommunikoi probe request -viestejä lähettävät älylaitteen kanssa. Aktiivisessa metodissa havaitseva sensori käyttää hyväkseen probe response -injektiota, jonka tarkoituksena on saada älylaite kommunikoimaan sensorin kanssa. Tutkimusten perusteella on saatu lupaavia tuloksia siitä, että aktiivinen metodi havaitsee noin 10 prosenttia enemmän probe request -paketteja. Nykyään kuitenkin osa älypuhelinten valmistajista on tehnyt parannuksia tähän, eivätkä kaikki puhelimet enää reagoi probe response -injektioihin (Sandrasegaran ym. 2013).

2.1.2 Infrapuna

Infrapunasensoreita käytetään havainnointiin kahdella tavalla, joista ensimmäinen hyödyntää konenäköä lisäten siihen ominaisuuksia lämmön tunnistamiseen. Toinen yleinen infrapunasensori on säteisiin perustuva infrapunasensori, joka voi olla joko passiivinen tai aktiivinen. Aktiivinen mittaa jatkuvasti lämpötilaa kun taas passiivinen mittaa lämpötilan vain silloin kun jokin tapahtuma “laukaisee” sensorin. Sensoreista saadaan dataa ulos kahdella tavalla, analogisella ja binäärisellä. Usein käytetään binääristä ulostuloa, jonka avulla pystytään vain kertomaan, onko sensori havainnut lämpösignaalin vai ei. Yleisesti passiivisista infrapunasensoreista käytetään nimitystä PIR, joka tulee sanoista pyroelectric infrared tai passive infrared. Aktiivisista infrapunasensoreista käytetään usein lyhennettä FIR, joka tulee sanoista far infrared ja koostuu usein kameran avulla otettujen lämpökuvien analysoinnista. Useimmiten aktiivisten infrapunasensoreiden käyttäminen on harvinaisempaa ja kalliimpaa, koska se vaatii rinnalleen toimivan videokamerajärjestelmän. Usein pyritään ottamaan mahdollisimman halpa sensori, jonka tunnistuskyky on vielä riittävän tehokas kunnolliseen tunnistukseen. Tästä johtuen pienet muutokset ympäristössä voivat aiheuttaa vakavia ongelmia sensorien toimintaan ja samalla myös datan eheys voi kärsiä. Koneoppimista ja tekoälyä pyritään hyödyntämään PIR-sensorien yhteydessä, koska niiden avulla datasta on mahdollista saada enemmän irti (Donovan ym. 2015).

Infrapunasensoreita käytettäessä on otettava huomioon häiriötekijät, joita ovat muun muassa

lämmitys- ja viilennyslaitteet sekä havainnointialueen ulkopuolelta tulevat heikot signaalit. Näiden lisäksi tulee myös huomioida se, miten erotetaan eläinten aiheuttamat lämpösignaalit ihmisten tekemistä. Mikäli sensorin avulla on tarkoitus kontrolloida jotakin, kuten esimerkiksi valaistusta, tulee huomioida se, että kontrolloitava alue on usein paljon isompi kuin yhden sensorin havainnointialue. Tästä johtuen usein käytetään useita infrapunasensoreita yhdessä, jotta saataisiin tarkempi kuva oikeasta tilanteesta. Sensorit saatetaan sijoittaa ympäri havainnoitavaa aluetta tai sitten niitä voidaan laittaa useita yhteen paikkaan, esimerkiksi torniin kiinni. Tällä on tarkoitus maksimoida havainnointialueen koko ja mahdollistaa sekä liikkuvien ihmisten reaaliaikainen seuraaminen että useamman suunnan yhtäaikainen huomioiminen (Dodier ym. 2005).

Infrapunakameroita käyttäessä tulee huomioida tasapainotus sensorien laadun ja resurssien kanssa. Koska laadukkaat kamerat tuottavat korkealaatuisia kuvia, kuluu niiden käsittelemiseen enemmän laskennallista kapasiteettia. Tämän lisäksi korkealaatuiset infrapunakamerat ovat huomattavasti kalliimpia kuin muut sensorit yleisesti. Yksi ominaisuus, joka täytyy erityisesti ottaa huomioon, on kameran resoluutio. Kameran resoluution vaikuttaa suoraan kuvassa näkyvään ihmiseen, joten mitä pienempi resoluutio on käytössä sitä pienemmältä ihminen näyttää kuvassa. Kuvissa olevat muodot ja tekstuurit voidaan kuvata Haar waveletin sekä staattisten ja histogrammisten ominaisuuksien avulla (Franz ym. 2010).

Brehar ym. käyttivät tutkimuksissaan infrapunakameraa yhdistettynä kahteen stereonäkösensoriin, joista käytetään yleisimmin nimitystä kamera. Stereonäkösensorit oli kiinnitetty auton sisäpuolelle tuulilasin taakse, jotta niillä olisi mahdollisimman hyvä näkymä eteenpäin. Infrapunakamera oli kiinnitetty auton katolle toisten kameroiden keskelle käyttäen auton pituusakselia kohdistukseen. Kameroista vain infrapuna oli koko ajan päällä ja kuvasi liikennettä ajon aikana. Tavalliset kamerat ottivat kuvia vain silloin kun ne tunnistivat ihmishahmon ja liittivät kuvaan myös aikaleiman. Ajon loputtua kuvat pyrittiin synkronisoimaan aina yhdeksi havainnoksi käyttäen apuna tavallisten kameroiden ottamia kuvia. Koska kuvaruutujen välillä on pientä epävarmuutta, yhteen tulokseen otetaan kaksi kuvaruutua, joissa on havaittu ihminen. Näin varmistetaan tuloksen oikeellisuus, koska kuvaruutujen välinen aika on niin pieni, ettei oikea havainto näy vain yhdessä kuvaruudussa. Jos jokin kameroista irroitetaan autosta, niin se täytyy kalibroida aina kun se kiinnitetään uudelleen autoon (Brehar

ym. 2015).

Tutkimuksessa käytetty havainnointiprosessi on kolmivaiheinen ja siinä vuorotellaan eri kameroiden käyttämistä vaiheessa tehtävien asioiden perusteella. Aluksi rajataan tavallisen kameran avulla kuvasta laatikko, joka voi mahdollisesti sisältää ihmisen. Seuraavaksi katsotaan infrapunakameralla otettua kuvaa samasta hetkestä ja koitetaan päätellä lämmön ja symmetrian avulla, onko edellisen vaiheen rajauksen sisällä ihminen. Viimeisessä vaiheessa pyritään käyttämään sekä muodon että pään tunnistusta jäljelle jääneisiin laatikoihin. Lopulta kun kaikki vaiheet on käyty läpi, saadaan selville, missä kuvissa esiintyy ihminen tai ihmisiä (Brehar ym. 2015).

PIR-sensorit ovat yleisesti halpoja ja pienen energiankulutuksen sensoreita, jotka ovat hyvin herkkiä ihmisistä aiheutuville infrapunasäteille. Tästä syystä useimmissa liiketunnistimissa hyödynnetään kyseistä teknologiaa. PIR-sensori on päällä vain silloin, kun se vastaanottaa infrapunasäteitä, muutoin se ei reagoi ympäristöön mitenkään. Kun tarkoituksena on seurata ihmisiä ja heidän liikettä, tarvitaan useampi PIR-sensori (Hao, Hu ja Lu 2010). Käytettäessä analogista ulostuloa täytyy sensorin liikkua, jotta se voi havaita staattiset objektit. Paikallaan oleva sensori havaitsee vain liikkeen. Analogista signaalia voidaan käyttää myös parantamaan objektien etäisyyden arvioimisen tarkkuutta, mutta vaatii erillisen kalibroinnin, ettei yleinen tarkkuus kärsi (Li ym. 2016). Signaalin avulla voidaan tunnistaa kohteen pituus, liikesuunta, tietyt eleet sekä erillisen kalibroinnin vaativa etäisyys sensorista. Näiden lisäksi infrapunan avulla voidaan hahmottaa yksittäisistä kohteista ei yksilöiviä ominaisuuksia, kuten askellus ja tavanomaiset liikeradat. Näitä tietoja voidaan hyödyntää mallien rakentamiseen, jotta havainnoista voitaisiin suodattaa muun muassa eläinten aiheuttavat virheelliset havainnot (Hao, Hu ja Xiao 2009).

PIR-sensorien virheen todennäköisyys on pienin kun etäisyys sensorista on viiden ja kuuden metrin välillä, mutta ympäristö vaikuttaa tähän hieman. Koska sensorit ovat erityisen herkkiä liikkeelle, täytyy havaintoja tehdessä ottaa huomioon mahdolliset ympäristötekijät, kuten esimerkiksi lehtien havina, tuuli ja lämpimät ilmavirrat. Yleisesti PIR-sensorien havainnointialue on pieni ja neliskulmainen, mutta sitä voidaan laajentaa käyttämällä Fresnel-linssiä. Fresnel-linssin avulla voidaan suurentaa havainnointialuetta, mutta ei kuitenkaan pysty tuottamaan korkealaatuista kuvaa. Infrapunasensorien tapauksessa tämä ei haittaa, koska lämpö

yleisesti säteilee hieman, joten kuvassa ei muutenkaan olisi täysin tarkkoja ääriviivoja tai muotoja. Jotta havainnoinneista voitaisiin luokitella pois eläimet, tulee luoda rajat ja luokat eläimiä varten. Esimerkiksi asettamalla erilaisia korkeusluokkia voidaan havainnot kategorisoida pituuden perusteella ja liian lyhyet objektit ottaa pois havainnoista. Sama toimenpide voidaan tehdä leveyssuunnassa, jolloin on helpompi havaita objektien liikkeet ja niiden perusteella luokitella havainto oikein. Jokaiselle alueelle tai luokalle annetaan oma sensori, joka seuraa vain kyseisen alueen liikettä eikä tee havaintoja laajemmalta alueelta. Virheiden välttämiseksi havainnointialueiden välinen ero niiden keskiosasta mitattuna tulisi olla vähintään metri (Narayana ym. 2015).

Usein PIR-sensoreita käyttävät havainnointijärjestelmät koostuvat kolmesta osasta, joista jokaisella on oma tehtävänsä. Havainnoiva moduuli koostuu pelkästään sensoreista ja sen ainoa tehtävä on ilmoittaa eteenpäin kun se havaitsee kohteen. Virheiden torjuntamoduulin tehtävänä on päätellä, onko havainto oikea vai häiriötekijän aiheuttama. Jotta havainto voidaan hyväksyä, tulee sen olla toistettavissa sekä erottua selkeästi ympäristöstä. Viimeinen osa on datan yhdistämisestä vastaava moduuli, joka kerää oikeat havainnot ja tekee mahdolliset laskentatoimet. Järjestelmässä on usein monta havainnoivaa moduulia sekä yksi tai useampi virheiden poistoon tarkoitettu moduuli. Datan yhdistämisestä vastaavia moduuleita on usein vain yksi (Hao ym. 2006).

Signaali voidaan prosessoida muutamilla tekniikoilla, joista yleisin on Kalman-suodatin. Tekniikkaa valittaessa tulee huomioida se, että niiden käyttöön vaadittavien laskennallisten resurssien lukumäärä vaihtelee huomattavasti. Esimerkiksi ikkunoidut tekniikat vaativat paljon laskennallisia resursseja verrattuna Kalman-suodattimeen, joka sopii parhaiten lineaarisiin ja Gaussin käyrän toteuttaviin prosesseihin (Hao ym. 2006).

2.1.3 Ultraääni

Ultraäänen perusteella voidaan arvioida ja laskea ihmisten lukumäärä tietyissä tiloissa. Tällöin käytetään aktiivista ultraäänisensoria, joka koostuu erillisestä kaiuttimesta ja mikrofonista. Kaiutin toistaa tietyn ennalta määrätyn äänen ja mikrofoni kuuntelee takaisin tulevaa ääntä. Äänen vaimentumisen avulla voidaan laskea algoritmien avulla, montako ihmistä ky-

seisessä tilassa on. Ihmiset vaimentavat ääntä, joten ihmisten laskemista varten tulee selvittää vaimennuskerroin eli kuinka paljon yksi ihminen vaimentaa ääntä. Näiden lisäksi on myös huomioitava joitain perustietoja tilasta, kuten esimerkiksi äänen voimakkuus tyhjässä huoneessa sekä kuinka huoneen muoto ja muut tavarat vaikuttavat äänenvoimakkuuteen (Shih ja Rowe 2015).

Ultraäänisensorissa käytetty havaintoääni ei saisi olla pituudeltaan alle 200 millisekuntia pitkä ja sen taajuusalue tulisi olla vähintään 5 kHz. Ääni voidaan laittaa sellaiselle taajuudelle, ettei ihminen kykene huomaamaan sitä. Normaalit kaiuttimet eivät kykene toistamaan korkeita ultraääniä vaan niitä varten on hankittava erillinen kyseiseen tarkoitukseen tehty kaiutin. Havaintoäänessä voidaan käyttää samanlaisia tekniikoita kuin käytetään nykyisissä tutkajärjestelmissä. Tekniikoiden tarkoituksena on parantaa kahden tai useamman kohteen erottamista toisistaan tietyllä alueella. Tekniikassa ultraäänen aiheuttama pulssi kokee kompression, jolloin sen volyymi pienenee ja paine kasvaa. Näitä tekniikoita ja järjestelmiä käyttäessä tulee huomioida se, että useamman samankaltaisen järjestelmän käyttäminen samassa tilassa mahdollistaa viestien sekoittumisen ja datan rikkoutumisen (Shih ja Rowe 2015).

Ultraäänisensoreita on kehitetty mittaamaan kohteen etäisyys havaitsevasta sensorista ja nämä sensorit ovat paljon yleisempiä liikenteen havainnoimisessa. Etäisyyssensorit ovat halpoja verrattuina kameroista koostuviin järjestelmiin verrattuna, eivätkä ne riko yksityisyyden suojaa. Sensorit asennetaan usein siten, että niiden säteet kulkevat tietyn rajatun alueen yli, kuten esimerkiksi kauppakadun. Kun sensorit on asennettu näin, ne kykenevät tunnistamaan ihmisen, mutta nopeus ja kulkusuunta jäävät peittoon. Jotta voitaisiin kunnolla havaita kohteen kulkusuunta ja nopeus, sensori tulee asentaa hieman vinoon. Näin ollen kun kohteen etäisyys sensorista vaihtuu, voidaan sen avulla laskea nopeus sekä päätellä kulkusuunta. Ultraääni ei ole riippuvainen valosta, joten se mahdollistaa tarkan havainnoinnin myös pimeällä. Sensoreiden tarkkuus on korkea silloin, kun havainnoitavan alueen koko pysyy kohtalaisena. Tarkimmat tulokset saadaan kun kohteen etäisyys sensorista on alle 7,5 metriä (Minomi ym. 2012).

Ultraäänisensoreita käytetään usein yhdessä muiden sensoreiden kanssa, joko parantamaan koko järjestelmän tarkkuutta tai auttamaan uuden sensorin kehityksessä. Esimerkiksi PDR

-menetelmässä (pedestrian dead reckoning) käytetään ultraääntä korjaamaan arvioita siitä, minne pelastajien tulisi oikeasti mennä (Fisher ym. 2008). Ulkona tapahtuvassa paikannuksessa saatetaan yhdistellä GPS:ää ja ultraääntä saadakseen tarkkoja tietoja, esimerkiksi maastossa kulkevista ihmisistä ja mahdollisista reiteistä. Hankalissa maastoissa ja tilanteissa on usein turvallisempaa luottaa järjestelmään, joka koostuu useammasta sensorista. Näin saadaan mahdollisimman tarkkoja tuloksia ja voidaan vähentää erilaisten tekijöiden aiheuttamia virheitä (Qiuying ym. 2018).

2.1.4 Konenäkö

Konenäössä käytetään usein erillistä stereokameraa tai jo valmiina olevaa valvontaan tarkoitettua videokamerajärjestelmää, jonka avulla saadaan videokuvaa halutusta alueesta. Toisissa malleissa video käsitellään Dynamic Stixel-World -mallin avulla, jolloin siitä saadaan kolmiulotteinen esitys. Stixel on vertikaalinen suorakulmio, jossa leveys pysyy aina ennalta määrätyn kokoisena, mutta korkeus vaihtelee. Operaation perusteella saadusta datasta voidaan alustavasti luokitella havaittuja objekteja omiin luokkiinsa. Tarkoituksena on poistaa kaikki vääristymät ja yhdistää havaitut objektit siten, että ne muodostavat kokonaisuuden, kuten esimerkiksi ihmisryhmän (Muffert, Pfeiffer ja Franke 2013).

Nykyaikaiset valvontaan tarkoitetut kamerat selviävät suhteellisen hyvin erilaisista sääolosuhteista ja valoisuuden muutoksista. Kameran liike tai tunnistettavan objektin vaihteleva muoto aiheuttavat ongelmia joillekin tunnistukseen käytetyille algoritmeille. Muodon vaihtelulla voidaan tarkoittaa joko fyysisen ulkomuodon muutoksia tai vaatetukseen liittyviä muutoksia. Tämän takia osassa kameroista on käytössä harmaaskaala, joka poistaa osan muun muassa värien aiheuttamista ongelmista. Vaikka käytössä on harmaaskaala, tulee kuvasta poistaa häiriötekijöitä, kuten valotaulut ja heijastavat pinnat. Nämä tekijät voivat vääristää lukemia, koska valotauluissa saattaa esiintyä ihmisiä ja heijastavat pinnat mahdollistavat saman ihmisen laskemisen useampaan kertaan. Objektien ja ihmisten tunnistamiseen voidaan käyttää hyödyksi ääriviivoja, joiden avulla voidaan rajata tietty alue kuvasta, jossa hyvin suurella todennäköisyydellä on ainakin yksi ihminen (Maurin, Masoud ja Papanikolopoulos 2005).

Konenäkö mahdollistaa erilaisten mittaustietojen keräämisen, kuten esimerkiksi nopeus ja liikesuunta, normaalin kokonaisluvun lisäksi. Videokuvasta poistetaan tausta ja jätetään jäljelle vain löytyneet objektit. Tietyissä prosesseissa tunnistetaan pikseleitä, jotka eivät sovi taustaan. Näitä pikseleitä pyritään yhdistelemään värien perusteella, jotta saataisiin esille koko objekti. Videon perusteella voidaan myös hahmottaa objektien liikettä, mutta sitä varten on huomioitava valoisuuden muutokset sekä toistuvat liikkeet ryhmässä. Liikettä havainnoidessa tulee huomioida ja suodattaa pois toistuvat liikkeet, jotka johtuvat useimmiten taustakohinasta tai taustalla olevista epäolennaisista osista (Shirazi ja Morris 2015).

Konenäössä hyödynnetään usein koneoppimista ja tekoälyä, jotta saataisiin parhaat havainnot esille. Järjestelmissä, joissa hyödynnetään kameroita, joiden resoluutio on matala, tarvitaan erityisesti erilaisia apukeinoja. Objektien ja taustan erottamiseen toisistaan on olemassa useita apukeinoja, kuten esimerkiksi neuroverkot ja Haar wavelet. Jotta ihmisten tunnistamisesta ja laskemisesta tulisi automatisoitua, tulee järjestelmä opettaa aluksi opetusvideon avulla. Opetusvideolla esiintyvät havainnot on vahvistettu manuaalisesti laskemalla. Järjestelmälle annetaan opetusvideo ja katsotaan, pystyykö se tiettyjen vaiheiden jälkeen tunnistamaan videolla esiintyvien ihmisten lukumäärän sekä tunnistamaan itse ihmiset (Raman, Sa ja Banshidhar 2016).

Raman, Sa ja Banshidhar (2016) käsittelevät useita artikkeleita, joissa on tutkittu kävelijöiden liikettä ja liikesuunnan havainnointia videokuvan avulla. Osassa menetelmistä on mahdollista havaita kameraa lähenevä ja siitä loitontuva liike, normaalin liikesuuntia havaitsemisen lisäksi. Videokuva prosessoidaan aluksi siten, että siitä poistetaan tausta ja suoritetaan muotojen tunnistamiseen liittyvät operaatiot. Tämän jälkeen poistetaan osat, joita ei haluta mukaan tunnistukseen. Jäljelle jääviin osiin laitetaan neliskulmainen rajaus, jonka sisällä on oletettu havainto. Rajaus elää aina objektin liikkeen mukaan ja niiden avulla voidaan laskea ihmiset ja liikesuunnat.

2.2 Pyöräilijät

Pyöräilijöiden havainnoiminen toteutetaan usein joko käyttämällä infrapunasäteitä, induktiosilmukoita tai pneumaattisia putkia. Manuaalinen screen-line laskenta on yksi perintei-

simmistä metodeista pyöräilijöiden laskemiseen. Siinä valittu laskenta-alue jaetaan erillisiin pienempiin alueisiin mielivaltaisten viivojen avulla. Tämän jälkeen lasketaan montako pyöräilijää käy kyseisellä alueella. Näiden lukemien perusteella muodostetaan suurempi kokonaiskuva alueen liikenteestä (Schasberger ym. 2012).

2.2.1 Pneumaattiset putket

Eco-Counterin TUBES on pneumaattisiin putkiin perustuva sensori tilapäiseen polkupyöräilijöiden laskemiseen. Sensorilla on pitkäkestoinen akku sekä sen asentaminen on yksinkertaista ("TUBES Mobile bike counter, designed with short-term bike volume studies in mind." 2019). Pneumaattiset putket tunnistavat kulkuvälineet paineen perusteella. Putken sisäinen paine muuttuu kun sen yli ajetaan ja paineen muutoksen aiheuttama ilmavirta voidaan mitata (Leduc 2008).

Pneumaattisia putkia on kahdenlaisia, vain polkupyörien havainnoimiseen tarkoitettuja sekä ajoneuvojen ja polkupyörien havainnoimiseen tarkoitettuja. Vain polkupyörien havainnointiin tarkoitettujen putkien tarkkuus on noin 20 prosenttiyksikköä suurempi kuin ajoneuvojen ja polkupyörien havainnointiin tarkoitettujen putkien. Kuitenkin molempien putkien tarkkuus laskee huomattavasti tien ollessa yli 8,5 metriä leveä (Hyde-Wright, Graham ja Nordback 2014).

2.2.2 Induktiosilmukat

Eco-Counterin kehittämä ZELT on induktiosilmukoihin perustuva sensori polkupyöräilijöiden laskemiseen. Kyseinen sensori on käytössä monessa suuressa kaupungissa ympäri maailmaa, kuten esimerkiksi Pariisissa, New Yorkissa, Oslossa ja Ottawassa. ZELT-sensorissa on 13 erilaista kriteeriä, joiden perusteella se analysoi polkupyörien elektromagneettisia tunnuksia. Sensori käyttää SIRIUS nimistä algoritmia polkupyörien tunnistamiseen samalla jättäen autot pois laskuista ("Urban ZELT: The world's most trusted permanent bike counter, designed for urban cycling" 2019).

Induktiosilmukoiden virhemarginaali vain polkupyöräilijöille tarkoitetuilla teillä on 3 prosenttia. Virhemarginaali muodostuu siitä, ettei sensori laske kaikkia polkupyöriä. Jaetuilla

kulkuväylillä induktiosilmukoiden virhemarginaali polkupyörien tunnistamiseen on 4 prosenttia. Jaetuilla teillä sensori laskee polkupyöriksi myös osan muusta liikenteestä. Havainnoinnissa tulee virheitä, vaikka pyöräilijöille olisi merkitty erillinen kaista autotien laitaan. Tällöin induktiosilmukka saattaa havaita myös osan vierellä ajavista autoista. Ongelma voidaan välttää siten, että pyörätien ja autotie eivät ole lähekkäin toisiaan. Niiden väliin tulisi jättää piennaralue, joka heikentää sensorin signaalia siten, että se ei havaitse autoja. Induktiosilmukoiden asentamisessa täytyy myös kiinnittää erityistä huomiota mahdollisiin sähköisiin häiriötekijöihin, kuten esimerkiksi voimalinjoihin ja suuriin maanalaisiin sähkökaapelien keskittymiin (Nordback ym. 2011).

Induktiosilmukoiden mittauksissa tulee usein esille usein kolmenlaisia virheitä, joista ensimmäisessä polkupyöriksi lasketaan muita kulkuneuvoja. Toinen yleinen virhe on se, että sama polkupyörä lasketaan useamman kerran. Kolmannessa virheessä ei lasketa polkupyörää, vaikka sellainen ylitti induktiosilmukan. Tämä virhe tapahtuu useimmiten silloin kun kaksi polkupyöräilijää ajaa rinnakkain sensorin yli tai pyöräilijät ajavat peräkkäin siten, että heidän välissä oleva tila on pienempi kuin polkupyörän pituus. Yksisilmukkainen sensori ei pysty erottamaan rinnakkain ajavia polkupyöräilijöitä useammaksi kuin yhdeksi pyöräksi, mutta peräkkäin minimaalisella välimatkalla ajavat pyöräilijät tunnistetaan hyvin. Monisilmukkainen sensori kykenee tunnistamaan rinnakkain ajavat polkupyöräilijät oikein, mutta ei pysty tunnistamaan polkupyöräilijöitä erilliseksi silloin, kun pyörien välimatka on minimalistinen. Yksisilmukkainen tai monisilmukkainen sensori ei kykene tunnistaa pieniä lasten polkupyöriä, joiden rengaskoko on alle 20 tuumaa (Nordback ym. 2011).

Induktiosilmukat sopivat hyvin polkupyörien laskemiseen, kunhan niiden silmukoiden lukumäärän ja asennuspaikan valintaan kiinnitetään huomiota. Kun lasketaan polkupyöriä täytyy miettiä, minkälaisia polkupyöriä halutaan laskea ja mitkä tapaukset ovat todennäköisempiä. Yhdelläkään sensorilla ei voida havaita jokaista polkupyöräilijää, koska pikkulasten polkupyöriä ei ole mahdollista havaita induktiosilmukoilla. Induktiosilmukka tulisi myös sijoittaa sellaiseen paikkaan, jossa polkupyöräilijät eivät pääse kiertämään sitä tai ajamaan vain osittain sen yli. Tällä tavoin varmistetaan, että mahdollisimman moni polkupyöräilijä tulee lasketuksi. Laskennan kannalta järkevimmät paikat sijoittaa induktiosilmukka on risteyksien jälkeen pöyrätielle siten, ettei autotie ole siinä kiinni tai paikan lähellä kulje voimalinjoja.

Kuvio 4. Polkupyöräilijöiden laskemiseen tarkoitettu induktiosilmukka.

2.2.3 Bluetooth

Polkupyöräilijöiden tunnistaminen Bluetoothin avulla perustuu MAC-osoitteiden havaitsemiseen ja laskemiseen. Tekniikan käyttäminen polkupyöräilijöiden laskemiseen on kuitenkin ongelmallista muutamasta syystä. Yksi suurimmista ongelmista on se, etteivät kaikki pyöräilijät omista älypuhelimia tai pidä Bluetooth-yhteyttä päällä. Tästä johtuen laskennoissa täytyy käyttää apuna erillisiä osuuksia, joiden laskemiseen tarvitaan toinen luotettava sensori tai manuaalinen laskenta Bluetooth-laskurin rinnalle. Laskennoissa täytyy huomioida myös se, että Bluetooth-laskuri saattaa havaita myös kävelijöiden ja autoilijoiden laitteita. Tästä syystä osuuksia varten tehtävät laskennat tulisi suorittaa paikoissa, joissa autotie ei ole heti pyörätien vieressä eikä jalankulkijoita ole paljoa. Toinen suuri ongelma osuuksien laskemisessa on se, ettei Bluetooth-laskennassa näy usein samanlaisia piikkejä kuin muilla sensoreilla suoritetuissa laskuissa. Aamuruuhkat näkyvät selkeästi piikkeinä, esimerkiksi induktiosilmukoissa, mutta Bluetooth-laskurien datassa ei ole havaittavissa samanlaisia piikkejä. Tästä johtuen osuudet saattavat vääristyä ja todellisten osuuksien laskemiseen täytyy tehdä muutoksia, jotka ottavat tällaiset tapaukset huomioon erityistapausten lisäksi (Jansen ym. 2014).

2.3 Autot

Autojen seuraamiseen käytettävät sensorit asennetaan usein liikennevalojen ja risteysten yhteyteen. Autoja seurataan yleisesti niiden liikenteen laskemisen kannalta sekä liikenteen sulavuuden kannalta. Sensoreista tulevien tietojen perusteella voidaan ohjailla liikennevalojen vaihtumista sekä teiden yleistä käyttöastetta. Yleisimmin autojen laskemiseen käytetään pneumaattisia putkia, induktiosilmukoita, videokuvasta laskemista, mikroaaltotutkia sekä manuaalista laskemista (Leduc 2008).

2.3.1 Mikroaaltotutka

Mikroaaltotutka on sensori, joka kykenee paikallisesti seuraamaan useaa kohdetta yhtäaikaisesti ilman suurempia ongelmia. Jotta tutka toimii parhaiten, tarvitaan joitain perustietoja tiestä, kuten esimerkiksi kaistojen leveys sekä lukumäärä. Jos seurattava tieosuus on kaartuva täytyy tutkan edessä oleva alue jakaa ruudukoksi, josta vain tietyistä ruuduista tulevat havainnot lasketaan. Kaartuvan osuuden seuraaminen on usein kuitenkin harvinaisempaa ja sitä käytetään lähinnä tieosuuksien kartoittamiseen sekä karttojen tarkkuuden parantamiseen tutkan avulla (Behrendt 2016).

Nykyään mikroaaltotutkat ovat alkaneet korvata perinteisiä induktiosilmukoita, vaikka suurimmassa osassa risteyksiä on molemmat asennettuina varoiksi. Mikroaaltotutkien etuna on pidetty sitä, että niiden avulla voidaan arvioida myös ajoneuvojen nopeutta. Usein tutkat asennetaan tien reunassa oleviin sähkötolppiin, joista tutkalla on paras mahdollisuus havainnoida alue koko tien leveydeltä. Tutkan toiminta perustuu taajuuden perusteella mukautettuihin mikroaaltoihin, joita lähetetään kohti tietä. Kun allot osuvat ajoneuvoihin ne kimpoavat takaisin tutkaa kohti. Tutka mittaa ajan, joka aalloilta menee edestakaiseen matkaan, jonka perusteella voidaan mitata tiellä olevien autojen lukumäärä. Aaltojen vastaanottamisessa tulee kiinnittää erityistä huomiota sekä kaistojen vaihtamisesta että kaistojen reunalla ajavista autoista aiheutuviin virheisiin. Aikaleimojen avulla on mahdollista suodattaa havaintoja siten, että otetaan vain yksi havainto yhtä aikaleimaa kohden. Aikaleimojen suodattaminen edellyttää myös tietoja tien ja kaistojen leveydestä, jotta voidaan hahmottaa mahdollisen ajoneuvon etäisyys sensorista (Ho ja Ching 2016).

2.3.2 Induktiosilmukka

Perinteiset induktiosilmukat ovat sidonnaisia kaistoihin eli ne eivät kykene laskemaan ajoneuvoja kaistattomissa olosuhteissa, kuten esimerkiksi Intian liikenteessä. Nämä sensorit sopeutuvat parhaiten liikenteelle, joka noudattaa kaistoja sekä koostuu pääsääntöisesti moottoriajoneuvoista. Perinteiset sensorit eivät kykene laskemaan sekä polkupyöriä että moottoriajoneuvoja. Induktiosilmukoiden toimintaperiaate perustuu magneetteihin ja siihen, miten auton ajaessa silmukan päälle silmukan induktio muuttuu. Induktion muutos aiheuttaa taajuuden muutoksia, jotka voidaan mitata (Ali, George ja Vanajaksi 2013).

Monisilmukkainen sensori mahdollistaa sekä erilaisten kulkuneuvojen havainnoinnin ja laskemisen että kaistattoman liikenteen seuraamisen. Sensorin asennus tapahtuu samalla tavalla kuin perinteisen induktiosilmukan asennus eli asvalttiin sahataan viivoja, joihin silmukan osat sijoitetaan. Erona perinteisen ja monisilmukkaisen välillä on sahattavien viivojen lukumäärä, koska silmukoita sijoitetaan koko tien leveydelle eikä yhdelle kaistalle vain yhtä silmukkaa. Monisilmukkaisessa sensorissa silmukat kytketään sarjaan, joka vähentää yhdistämiseen tarvittavien johtojen määrä sekä pienentää virheiden mahdollisuutta. Vaikka induktiosilmukat asennus on hieman vaativampaa kuin toisten sensorien, ne ovat halvempia ja vaativat harvemmin huoltoa kuin muut sensorit (Ali, George ja Vanajaksi 2013).

Sensoreita varten voidaan luoda erillisiä luokkia eri ajoneuvojen mukaan, kuten esimerkiksi autot, linja-autot, moottoripyörät ja skootterit. Jokaisella ajoneuvotyypillä on omanlaisensa signaali, jonka avulla se voidaan tunnistaa ja luokitella. Vertailuarvot signaalien analysointiin ja luokkien selvittämiseksi saadaan arvoista, joita silmukat antavat tyhjistä kohdista(Ali, George ja Vanajaksi 2011). Eri ajoneuvojen tunnistaminen helpottuu myös kun tiedetään, kuinka monta silmukkaa ilmoittaa havainnosta. Monisilmukkaisessa järjestelmissä silmukoilla on tietty ennalta määrätty koko, useimmiten noin puoli metriä. Jotta viereinen sensori ei virheellisesti ilmoittaisi havainnosta, vaikka sen päältä ei ole ajanut ajoneuvo, tulee silmukoiden väliin jättää noin 10 senttimetrin levyinen tila. Yleisesti kaistanleveys vaihtelee 2,5 metristä 3,25 metriin, jolloin yhdelle kaistalle mahtuu 4 tai 5 sensoria väliin jäävien tilojen kanssa (Ali, George ja Vanajaksi 2013).

Sensori voidaan kalibroida automaattisesti aikoina, jolloin ruuhkaa ei ole odotettavissa, ku-

ten esimerkiksi yöllä. Kalibrointi voidaan aloittaa myös manuaalisesti komentokeskuksesta. Kalibroinnin avulla saadaan helpommin huomattua epänormaalit taajuuden vaihtelut, jotka järjestelmän on tarkoitus raportoida takaisin komentokeskukseen. Induktiosilmukoita voidaan käyttää myös nopeuden mittaamiseen, koska tutkimusten mukaan suurin heitto laserilla tehtyjen mittausten ja induktiosilmukasta saatujen arvojen välillä on 1 km/h (Ali, George ja Vanajaksi 2013).

2.3.3 Magneettiset sensorit

Magneettisia sensoreita on mahdollista käyttää pitkäaikaiseen seurantaan, koska ne kestävät ajankulun aiheuttamia ilmiöitä hyvin. Sensori ei ole mitenkään riippuvainen säästä, joten erilaiset olosuhteet valoisuuden suhteen, eivät muuta havaintoja. Sensorit ovat myös langattomia ja ne asennetaan usein tien reunaan. Asennuspaikasta ja sensorin koosta johtuen, itse tielle aiheutuu hyvin vähän vahinkoa. Asennusta varten tiehen täytyy tehdä vain pieni sylinterimäinen reikä, jonne sensori asetetaan, kun taas induktiosilmukoita varten täytyy tehdä useita viiltoja. Sensorin käyttämisestä ruuhkaisilla teillä ruuhka-aikaan ei ole tarpeeksi tutkimusta, jotta voitaisiin tietää tarkalleen, kuinka data saataisiin vastaamaan totuutta (Bao ym. 2016).

Sensorin toiminta perustuu sähköisen resistanssin vaihtumiseen magneettikentän muutosten mukaan. Sensoreissa käytetään usein ferromagneettisia metalleja, jotka vetävät magneettejä puoleensa. Magneettiset sensorit ovat erittäin herkkiä magneettikentän vaihtelulle, joita esimerkiksi ohiajavat autot aiheuttavat. Yleisesti magneettisien sensoreiden kanssa käytetään kolmea yleistä magneettisuus-resistanssi vaikutusta, joista ordinary magneto-resistance (OMR) ja giant magneto-resistance (GMR) eivät ole olleet käytössä niin paljon kuin kolmas vaikutus. Anisotropic magneto-resistance (AMR) on yleisin käytössä oleva metodi, jota on myös tutkittu eniten. Anisotropic sanana tarkoittaa sitä, että jollain asialla on eri arvo eri kohdista mitattuna. Esimerkiksi puun lujuuskerroin on täysin eri mitattuna pitkittäissuunnasta kuin poikittaissuunnasta. Sensoria käytettäessä täytyy ottaa huomioon se, että sensoria ei yleensä huolleta vaan vanhan tilalle vaihdetaan usein uusi. Isot autot ja kaistaa vaihtavat autot aiheuttavat vääristymiä dataan samoin kuten ruuhka-aikana autojen seisominen paikallaan. Sensoreita ei ole vielä laajalti käytössä niihin liittyvien ongelmien takia (Bao ym. 2016).

3 Liikenteen mallintaminen

Liikennettä mallinnetaan useilla eri tavoilla, joista jokaisella on oma käyttötarkoituksensa. Yleisimmin liikenteestä julkaistaan erilaisia kuvaajia ja raportteja, jotka kokoavat yhteen kyselytutkimuksen ja manuaalisen laskennan tulokset. Usein tällaisia tutkimuksia tehdään kaupungeissa muutamien vuosien välein ja maanlaajuisesti aina silloin tällöin. Näiden tutkimusten tarkoituksena on pyrkiä katsomaan, mihin suuntaan liikenne on kehittymässä ja miten eri vuodet vertautuvat toisiinsa.

3.1 Avoimen datan laatu

Avoin data määritellään usein siten, että se on kaikkien vapaasti käytettävissä. Avoimen datan käyttöä rajoitetaan kuitenkin joissakin tilanteissa, kuten esimerkiksi lain puitteissa tulee toimia. Avoimen datan tarjoajien tulisi varmistaa datan laatu ja tarjota vain laadukasta dataa. Asiaa vaikeuttaa se, että laadulle on monta määritelmää ja myös datan määrä vaikuttaa laatuun. Avoimen datan tarjoajat voidaan jaotella kolmeen eri ryhmään. Ensimmäiseen ryhmään kuuluvat organisaatiot, jotka antavat ilmaisen pääsyn säännösteltyyn dataan. Toiseen ryhmään kuuluvat yksilöt, joiden tuottama avoin data hyödyntää yksityisiä yrityksiä. Kolmanteen ryhmään kuuluvat organisaatiot, jotka myyvät yrityksille pääsyä dataa sisältäviin tietokantoihin (Immonen 2017).

Avoin data itsessään ei ole arvokasta, vaan sen arvo tulee siitä, mitä kaikkea datan avulla voidaan tehdä ja selvittää. Avoin data jaotellaan usein ryhmiin, kuten esimerkiksi liikennedata tai ostokäyttäytyminen. Yleensä yhden ryhmän data keskitetään tiettyyn paikkaan, josta sitä tarjotaan muille ja yhdestä paikasta saatua dataa voidaan hyödyntää useassa eri paikassa. Jotta data olisi laadukasta tulee sen täyttää sille asetetut vaatimukset, kuten esimerkiksi mitä dataa tarjotaan, miten data on tarkoitettu hyödynnettäväksi, milloin dataa voidaan hyödyntää ja missä sitä voidaan hyödyntää. Yleinen käytäntö avoimen datan kanssa on se, että sitä validoidaan empiirisen datan avulla. Tästä johtuen sama avoin data ei ole kaikkien mielestä yhtä laadukasta, koska se ei sovellu jokaiseen tarkoitukseen (Immonen 2017).

Datan laatu voidaan joissain tilanteissa johtaa suoraan sensorien toiminnasta. Vialliset sen-

sorit tulee vaihtaa tai korjata, koska ne antavat vääriä arvoja, jotka laskevat kokonaisuuden laatua. Kiinteästä sensoriverkosta saatu data on usein laadukkaampaa kuin liikuteltavien sensorien avulla saatu data. Tämä johtuu siitä, että kiinteästä verkosta on helpompi huomata ja korjata puutteet ja virheet. Puutteelliset arvot heijastuvat suoraan datan laatuun, vaikka pienet puutteet eivät yleensä haittaa data-analyysissä. Mikäli puute on pieni ja hetkellinen, esimerkiksi yksi sensori ei ole lähettänyt yhtään dataa tunnin ajalta, se ei suuressa mittakaavassa vaikuta datan laatuun. Esimerkiksi data-analyysissä tällainen pieni puute ei aiheuta ongelmia, koska se voidaan paikata joko historiadatan avulla tai johtaa muiden sensorien datasta. Datan laatuun liittyy myös sen alkuperä eli onko data puhtaasti yhdestä lähteestä vai onko siihen sekoitettu muuta dataa. Usein data, joka ilmoittaa kokonaismääriä on sekoitettua dataa eli se sisältää dataa useammasta sensorista tai siihen on lisätty historiallista dataa parantamaan tarkkuutta (Syrjärinne 2016).

3.2 Tilastoiminen

Suomessa ei kerätä vuosittain mitään valtakunnallista tilastoa polkupyöräilijöistä tai jalankulkijoista. Valtakunnallisesti kerätään kuuden vuoden välein tietoa tehtyjen matkojen määrien ja pituuden kehityksestä, jonka avulla voidaan vain seurata kehitystä. Toiset kaupungit keräävät tietoa jalankulkijoiden ja polkupyöräilijöiden lukumääristä. Useimmiten tällaiset tiedot kerätään kyselytutkimuksen avulla, joka lähetetään syksyisin koulujen alettua. Viimeaikoina sensorit ovat yleistyneet kevyen liikenteen seuraamisessa ja monet kaupungit ovat asentaneet muutamia sensoreita seuraamaan myös kevyttä liikennettä. Autojen seurannassa sensorit ovat olleet yleisiä jo pitkään, koska useimpien liikennevalojen yhteydessä on induktiosilmukoita. Autojen lukumääristä pidetään yllä tilastoa ja siihen saadaan tietoa muun muassa liikennelaskennasta, kulutustutkimuksista, ilmakuvauksesta sekä polttoaineen myynnin ja autojen keskikulutuksen avulla tehdyistä laskennoista (Kalenoja ja Mäkelä 2001).

Viestintä- ja liikenneministeriön mukaan jokaisen kunnan, jossa on vähintään 100 kilometriä kevyen liikenteen väyliä, tulisi suorittaa tiettyjä laskentoja vuoden aikana. Kunnat on jaettu tiepiireihin, joiden sisällä laskennat tulisi suorittaa. Kerran vuodessa tulisi tehdä manuaalinen otoslaskenta, jota varten tarvitaan 50-100 laskentapistettä. Mikäli kunnassa on käytössä sensoreita, tulisi niiden suorittaa automaattista ja jatkuvaa seurantaa 28-74 laskentapisteel-

lä. Sensoreiden avulla tulisi ottaa myös otoslaskenta kaksi kertaa viikossa yhteensä 250-600 laskentapisteeltä. Kun tiepiiri täyttää edellä mainitut laskentapisteet, on se saavuttanut hyvän laskentatason. Maanlaajuisesti hyvä laskentataso saavutettiin arviolta vuonna 2012 ja taso on pysynyt samana siitä eteenpäin (Litmanen ym. 2006).

Jyväskylän kaupunki on vuonna 2015 tehnyt tutkimuksen kevyestä liikenteestä ja vuonna 2009 kokonaisvaltaisen tutkimuksen alueen liikenteestä, johon lukeutui myös kulkutapajakauma. Vuonna 2015 Jyväskylässä oli noin 470 kilometriä kevyen liikenteen väyliä ja koko kaupungin liikenteestä kävelijöiden osuus oli 21 prosenttia ja polkupyöräilijöiden 13 prosenttia. Kevyen liikenteen väylien käyttöaste on korkea, joka näkyy myös kaupungin keräämässä liikennedatassa. Aktiivisempana aikana pyöräilijöitä on kuukaudessa 50 000 - 60 000 ja kävelijöitä noin 40 000 (*Jalankulku ja pyöräily 2015* 2015).

3.3 Kulkutapajakauma

Viimeisten vuosien aikana älykkäät liikenteenohjauskeinot ovat yleistyneet kaikkialla. Jotta liikennettä voitaisiin ohjata älykkäästi, siitä tarvitaan valtavasti tietoa. Uusien sensoreiden avulla liikenteestä saadaan niin paljon dataa, että ollaan siirrytty jo Big datan puolelle. Ajankohtaisilla kuvauksilla liikenteestä voidaan mm. helpottaa oikeanlaisten reittivalintojen tekemistä sekä parantaa ruuhkaisia tieosuuksia (LV ym. 2014).

Liikenteen tila arvioidaan yleensä suhteellisen kattavasta sensoridatasta, manuaalisesta laskennasta tai niiden yhdistelmästä. Tila voidaan kuitenkin arvioida myös sellaisesta datasta, joka ei ole täysin kattava. Tällaisissa tilanteissa täytyy kuitenkin olla ennalta jonkinlaista tietoa teiden normaalista liikenteestä, kuten esimerkiksi vanhoja kulkutapajakaumia tai tieto siitä, että sensorittomilla teillä liikenne on hyvin samanlaista kuin sellaisilla teillä, joilla on sensoreita. Kalman-suodatin yhdessä makroskooppisen liikennedynamiikan mallin mahdollistaa liikenteen kokonaistilan arvioinnin. Tila voidaan myös muuntaa 3D-malliksi, jota voidaan tarkemmin tutkia virtuaalisen todellisuuden keinoilla, kuten muun muassa VR-laseilla (Wilkie, Sewall ja Lin 2013). Tilaa arvioidessa tulee huomioida sääolosuhteet sekä datan yhdistämisestä aiheutuvat ongelmat, kuten esimerkiksi sensorien eri herkkyydet havainnoinnissa sekä datassa esiintyvät virhearviot ja virhetilat. (Yuanhua ja Enhui 2011).

Vuosituhannen alkupuolella kehitettiin muutamia menetelmiä, joista osa on vielä osittain käytössä ja toisten pohjalta on kehitetty uusia menetelmiä. Näillä vanhoilla menetelmillä on jokaisella omat vahvuutensa ja heikkoutensa, mutta suurin ero nykyisiin menetelmiin on se, että nykyiset menetelmät kykenevät havainnoimaan paljon enemmän erilaisia parametreja liikenteestä. Nämä menetelmät kykenivät jonkinlaiseen alkeelliseen ennustamiseen, mutta vaativat muita järjestelmiä ja ihmisiä niiden toteuttamiseen. USA:ssa kehitetty WADS-menetelmä kykenee laskemaan liikenteen määrän sekä selvittämään liikenteessä olevien ajoneuvojen nopeuden. Englannissa kehitetty TULIP-järjestelmään pohjautuva menetelmä kykeni mittaamaan enemmän asioita liikenteestä WADS-menetelmään verrattuna. Menetelmä hyödynsi analysointimenetelmää, joka koostui kahdesta erillisestä metodista. Belgiassa kehitetty tietokoneavusteinen liikennesensori CATS kykeni laskemaan ajoneuvojen lukumäärän, yksittäisten ajoneuvojen nopeuden sekä tunnistamaan yksittäisten ajoneuvojen pituuksia. Analyytikot pystyivät näiden tietojen perusteella tekemään alustavia arvioita liikenteestä (Fathy ja Siyad 1998).

3.4 Ennustaminen

Liikenteen ennustaminen on mahdollista tekoälyn ja koneoppimisen avulla. Ennustaminen on hyödyllinen työkalu sekä liikenteen että liikenteeseen liittyvän infrastruktuurin parantamiseen. Nykyaikaisen liikenteen ja sen sujuvuuden kannalta on tärkeää muun muassa pystyä optimoimaan reittivalinnat siten, etteivät kaupungin pääväylät ruuhkaudu liikaa tai heikompia teitä kuormiteta niin paljon, että niitä joudutaan korjaamaan jatkuvasti. Tämän lisäksi ennustamisesta on hyöty liikennevalojen ohjauksessa siten, ettei ruuhkaisina aikoina odoteta turhaan pitkiä aikoja sellaisia kaistoja, joita tilastojen mukaan käytetään todella vähän tiettyinä aikoina. Ennustamiseen koneoppimisen avulla käytetään yleensä malleja, jotka voidaan luokitella kolmeen erilaiseen kategoriaan. Parametrillisia malleja ovat muun muassa Kalman-suodatin sekä aikasarja-metodit. Parametrittomia malleja ovat k-lähintä naapuria algoritmi (k-NN) sekä neuroverkot. Simulaatioihin kuuluvat mallit, joiden avulla on tarkoitus näyttää oikeaa tilannetta simuloivia tilanteita (LV ym. 2014).

Fuzzy-menetelmiä on yleisesti käytetty liikenteen ohjaamiseen, mutta se soveltuu myös ennustamiseen. Menetelmä pyrkii löytämään datasta tiettyjen ilmiöiden tunnistamisen kannalta

oleellisen rakenteen. Fuzzy-menetelmät sisältävät erilaisia malleja ryhmittelyyn, jotka keskittyvät hieman eri osiin datasta. Yleisimmät mallit ovat Fuzzy C-means, Fuzzy C-linear varieties ja Fuzzy C-elliptotypes. Malli valitaan usein ennakkotietojen perusteella, joita ovat muun muassa liikenteen oletettu määrä, erilaisten ajoneuvojen lukumäärä ja tiekohtaiset rajoitukset. Näiden tietojen perusteella voidaan valita malli, joka soveltuu parhaiten käytössä olevasta datasta rakenteen etsimiseen (Stutz ja Runkler 2002).

Stacked autoencoder -malli (SAE) mahdollistaa yleisten liikenteen kulkuun liittyvien ominaisuuksien opettamisen tietokoneelle. Liikenteen ennustamisessa käytetään usein aikasarjametodeihin perustuvia ratkaisuja, mutta hybridimallien suosio on kasvanut huomattavasti viime aikoina. Hybridimallien etu on selkeästi se, että siinä voidaan hyödyntää useiden mallien parhaimpia puolia ja ne mukautuvat sekä olosuhteiden että tapausten vaihtoihin. Kerroksittaista opetusta on myös hyödynnetty paljon liikenteen mallintamisessa, mutta aiemmin sen käytössä on ollut ongelmia. Näitä ongelmia on vähennetty siten, että hiotaan opetusta ensimmäisten karkeiden opetuskertojen jälkeen, jotta lopputulos vastaisi paremmin todellisuutta. Kerroksittaisen mallin perusidea on se, että edellisen kerroksen lopputulos toimii seuraavan kerroksen alkutilanteena. Kierroksia toistetaan niin monta kertaa kuin halutaan ja kun on saavutettu haluttu kierrosmäärä, ylimmät kerrokset hiovat lopputulosta mahdollisimman hyvin todellisuutta kuvaavaksi. Tällä metodilla saadut kuvaajat ovat yleensä muodoltansa samanlaisia kuin absoluuttisen laskennan avulla saadut, mutta pieniä eroja määrissä esiintyy. Metodi sopii parhaiten sellaisiin tapauksiin, joissa tiellä oleva liikenne on kohtalaista tai vilkasta, koska erot korostuvat ja kasvavat teillä, joilla liikenne on vähäistä (LV ym. 2014).

Kaikissa ennustamiseen liittyvissä malleissa ei käytetä historiallista dataa eli dataa, jonka keräämisestä on kulunut aikaa yli vuosi. Tällaisissa malleissa kyseistä dataa ei ryhmitellä yhteen nykyisen datan kanssa ja samalla pyritään välttämään ennen ennustamista tapahtuvaa ryhmittelyä kokonaan. Ennustus on tarkoitus toteuttaa funktionaalisesti mahdollisimman tarkasti käyttäen vain suhteellisen reaaliaikaista dataa, josta etsitään ennustamisen kannalta oleelliset tiedot. Funktionaalisen data-analyysin funktionaalisia komponentteja voidaan hyödyntää myös puuttuvan datan kanssa, koska analyysin tarkoituksena on löytää hallitsevat ilmiöt datasta. Nämä ilmiöt määrittelevät, millaista kerätty liikennedata on pohjimmiltaan ja mitä ominaisuuksia siitä voidaan hyödyntää jatkossa. Koska tärkeimmät ominaisuudet eivät

muutu päivien kuluessa, voidaan niiden avulla ennustaa seuraavan päivän liikenne, vaikka osa datasta puuttuisi (Wagner-Muns ym. 2018).

ARIMA:n kausittainen malli SARIMA ja PCA:n funktionaalinen malli FPCA ovat kaksi yleistä analysointimallia, joita käytetään liikenteen ennustamiseen. Molempien pohjalla olevat mallit ovat olleet jo pitkään käytössä ja niiden eri muotoja hyödynnetään paljon muissa koneoppimista ja tekoälyä vaativissa tehtävissä. Erityisesti FPCA on hyödyllinen liikenteen ennustamisessa, koska se jättää hyvin harvinaiset tapahtumat huomioimatta. Tästä johtuen mallilla on mahdollista saada tarkkoja yleisiä ennustuksia liikenteestä, joita yleensä hyödynnetään liikennesuunnittelussa. SARIMA:n avulla voidaan laskea seuraavan päivän FPCA-pisteet, joiden pohjalta voidaan laskea funktionaalinen ennuste seuraavan päivän liikenteestä. FPCA:lla tehdyt ennusteet ovat tutkimusten mukaan antaneet tarkempia ennustuksia kuin yksistään SARIMA:lla saadut ennustukset (Wagner-Muns ym. 2018).

Liikenteen ennustamisessa on otettava huomioon sään vaikutus ihmisten liikkumiseen ja kulkuneuvojen valintaan. Kevyellä sateella ihmiset valitsevat huomattavasti useammin kulkuneuvokseen joko oman auton tai linja-auton, samalla kävelijöiden ja polkupyöräilijöiden määrä tippuu huomattavasti. Tämä hankaloittaa huomattavasti liikenteen tarkkaa ennustamista, koska sateella liikenteessä olevien autojen lukumäärä kasvaa. Rankkasateet vaikeuttavat vielä enemmän liikenteen ennustamista, koska moni ei halua ajaa kovalla sateella vaan joko liikkuvat linja-autolla tai jäävät mahdollisuuksien salliessa kotiin. Säätiloista johtuen osa malleista käyttää kahta erilaista runkoa liikenteen ennustamiseen, joista valitaan aina seuraavan päivän sääennusteen mukaan sopivampi (Dunne ja Ghosh 2013).

Neuroverkot ovat selkeästi yksi suosituimmista parametrittomista malleista, joita käytetään liikenteen ennustamiseen. Ne soveltuvat hyvin ennustamiseen joko yksinään tai osana hybridimallia. Neuroverkko on staattinen työkalu, joka on tarkoitettu datan louhimiseen, esimerkiksi yhdessä erilaisten signaalien prosessointitekniikoiden kanssa. Niiden innoittajana on toiminut biologinen neuroverkko, kuten esimerkiksi ihmisen hermosto ja muistamiseen liittyvät synapsit aivoissa. Wavelet transform on yleinen signaalin prosessointiin tarkoitettu tekniikka, josta on kehitetty Discrete wavelet transform -tekniikka. Kyseinen tekniikka on käytössä data-analyyseissa, joissa on käytössä useampi resoluutio (Dunne ja Ghosh 2013).

4 Datan kerääminen ja analysointityökalut

Tässä luvussa kuvataan teoriapohjaa yleisimmille liikennedatan analysointiin tarkoitetuille menetelmille sekä kuvataan datan keräämiseen liittyviä asioita. Datan keräämiseen ja analysointiin liittyvät vahvasti tietyt perinteiset mallit, jotka on todettu toimiviksi. Tämän lisäksi analysoitavan datan tulisi olla mahdollisimman tarkkaa ja eheää, jotta siitä olisi hyötyä. Kappaleessa esitellään menetelmät, joita voidaan soveltaa tutkimuksen kohteena olevaan dataan sekä kuinka dataa voidaan hyödyntää jos siinä on puutteita.

4.1 Analysointimetodit

4.1.1 PCA

Lyhenne PCA tulee sanoista principal component analysis eli pääkomponenttianalyysi ja siitä on olemassa perinteisen metodin lisäksi muita erilaisia versioita. Yleisimmin liikennedatan ennustamisessa käytetään joko funktionaalista pääkomponenttianalyysia (FPCA) tai todennäköistä pääkomponenttianalyysia (PPCA). Jotta PCA voisi tunnistaa datasta pääkomponentit, tulee siitä aluksi suodattaa pois melu, jonkin toisen metodin avulla. PCA tunnistaa datasta kaavat ja korostaa niiden samankaltaisuudet ja erot datassa. Tästä johtuen PCA on hyvä työkalu analysoimaan raakadataa, erityisesti silloin kun datassa on laajoja ulottuvaisuuksia. Kun datasta on tunnistettu kaavat, metodi tiivistää datan. Tiivistämisessä vähennetään ulottuvuuksia siten, ettei mitään oleellisia tietoja menetetä (Meta ja Cindikici 2010).

PPCA eli todennäköinen pääkomponenttianalysointi mahdollistaa puuttuvien data-arvojen löytämisen. Tällainen tilanne tulee usein eteen silloin, kun jokin sensori ei lähetä mittaustuloksia tai mittaustuloksissa on jotain virheellistä. Tarkoituksena on eritellä datasta merkittävät ja dominoivat ilmiöt, jotta siitä voidaan suodattaa pois merkityksettömät osat sekä osat, joita ei voida mallintaa. PCA sopii independent component analysis -mallia eli itsenäistä komponenttianalyysiä paremmin liikennedatan kanssa käytettäväksi, mutta joissain hybridimenetelmissä se on otettu osaksi kokonaisuutta. Tämä johtuu siitä, että ICA on kehitetty käytettäväksi itsenäiselle datalle, joka ei noudata Gaussin jakaumaa (Qu ym. 2009).

PCA:ta käytetään datan ulottuvuuksien vähentämiseen sekä itse datan tiivistämiseen. Näiden lisäksi tekniikkaa käytetään ominaisuuksien löytämiseen datasta sekä kerrointen analysoimiseen. Metodilla löydettyjä kertoimia voidaan myöhemmin käyttää ennustamisessa. Menetelmä ei yleensä ota huomioon harvinaisempia tapahtumia, kuten esimerkiksi onnettomuuksia tai tietöitä. Menetelmää voidaan kuitenkin käyttää näiden harvinaisempien tapahtumien löytämiseen datasta. PCA:ta on käytetty paljon liikennedatan analysoimiseen ja ennustamiseen sekä siitä löytyy paljon erilaisia tutkimuksia vuosien varrelta. Toiset PCA-pohjaiset metodit osaavat myös huomioida koko datassa esiintyvät dominoivat ilmiöt ilman, että alussa olevat dominoivat vääristymät vaikuttavat lopputulokseen (Qu ym. 2009).

4.1.2 Kalman-suodatin

Kalman-suodatin on rekursiivinen suodatin, jonka avulla datasta voidaan poistaa melu. Suodatin yhdistää paikallisen liikennedatan malliin, joka mahdollistaa mallin tilan korjaamisen. Näin ollen menetelmä korjaa itse itseään, samalla parantaen ennustusta. Kalman-suodatinta käytetään todella usein induktiosilmukoista kerätyn datan kanssa, koska se sisältää paljon taustamelua, joka normaalisti vääristäisi ennustuksen tuloksia. Kalman-suodattimesta on olemassa useita erilaisia versioita, kuten esimerkiksi extended ja unscented. Jokainen versio muuttaa perusmallia hieman, jotta se sopisi paremmin erilaisiin tarkoituksiin ja päästäisiin heikkouksista eroon. Yleisimmin ennustamiseen käytetään kuitenkin vielä perinteistä mallia, jonka tuloksia käytetään tai tarkennetaan toisissa metodeissa (Hinsbergen ym. 2012).

Yleisesti perinteinen Kalman-suodatin on liian hidas toimiakseen reaaliaikaisesti isoissa verkostoissa. Suodattimen tekemät operaatiot ovat kalliita, joten niihin kuluu paljon laskennallista aikaa. Suodatin on ainakin teoriassa hyvin herkkä ei-lineaariselle liikenteelle, mutta soveltuu myös käytettäväksi kyseisen datan kanssa. Kalman-suodattimessa lineaarisen mallin tila arvioidaan käyttäen viimeisintä arviota tilasta hyödyntäen normaalijakauman havaintoja ja suodattimen koko toiminta perustuu ei-lineaariseen tila-paikka-yhtölään. Jotta Kalman-suodatinta voidaan käyttää ei-lineaarisen datan kanssa, data tulee aluksi muokata lineaariseen muotoon käyttäen apuna erilaisia työkaluja. Suodatinta käytetään yleensä datalla ja malleille, joiden jakautuminen perustuu Gaussin jakaumaan. Kalman-suodatinta on mahdollista käyttää myös datalle, joka ei jakaudu Gaussin jakauman mukaisesti (Hinsbergen ym. 2012).

Kalman-suodatin soveltuu hyvin lähitulevaisuuden ennustamiseen, mutta perinteisellä metodilla on tapana ampua yli ruuhka-aikojen liikenteen ennustamisen yhteydessä. Yliampumista voidaan hillitä ja tasoittaa käyttämällä ennustamiseen Kalmanin lisäksi toista metodia. Itse suodattimen muokkaus kahteen tasoon, korkeaan ja matalaan, parantaa myös tulosten luotettavuutta antaen tarkempia ennustuksia tulevasta (Zhou ym. 2019).

Kalman-suodatinta käytetään optimoimaan liikennemallien tuloksia rekursiivisesti. Data jaetaan intervalleihin, joiden aikana kerätty data syötetään Kalmanille. Suodattimen antama lopputulos on jokaisen intervallin tulos. Kalman-suodattimella saadaan aikaan optimaalisia tuloksia kun minimoidaan keskineliövirheen neliöjuuri. Koko prosessin tarkoituksena on saada paras arvio liikenteen tilasta yhdistämällä liikennemalleista saatuja tuloksia sensoreilta kerättyihin mittaustietoihin (Wong ym. 2018).

4.1.3 SARIMA

Seasonal autoregressive integrated moving average eli SARIMA on huippuluokan ennustusmalli, joka on todella tarkka ennustuksissaan ja sitä käytetään usein aikasarjadatan analysoimiseen. Mallin avulla voidaan ennustaa sekä lähitulevaisuutta että pidemmälle ajanjaksolle sijoittuvia asioita. Pidemmän ajan ennusteiden tekeminen on mahdollista muutossääntöjen luomisen avulla eli järjestelmä etsii säännöt, joiden perusteella liikenne muuttuu todennäköisimmin. Perinteinen ARIMA-malli ottaa huomioon muutaman edellisen tilan liikenteestä sekä tulevan tilan ja pyrkii muodostamaan näiden välille suhteen. SARIMA pyrkii parantamaan tätä ennustuksessa käytettyä suhdetta siten, että mukaan otetaan datan kausiluontoisuus tai jaksollisuus, joka parantaa huomattavasti ennustusten tarkkuutta (Xu ym. 2014).

SARIMA hyödyntää itsenäisiä muuttujia, jotka johdetaan sekä historiallisesta datasta että edellisten viikkojen liikennedatasta tietyillä intervalleilla. Nämä muuttujat vaikuttavat ennustamiseen, joten jos edellisellä viikolla on tiettynä ajankohtana tapahtunut jotain epänormaalia liikenteessä, heijastuu se ennustukseen. Tästä syystä kaikki SARIMA:lla tehdyt ennustukset eivät aina onnistu ja pidä paikkaansa, mutta yleisesti SARIMA on yksi tarkimmista ennustusmalleista (Xu ym. 2014).

Kaaviossa 4.1 esitetään SARIMA:n muoto, jota käytetään kaikissa SARIMA:lla tehdyissä

ennustuksissa . Funktiossa pienet kirjaimet p, d ja q ovat lyhyen ajan komponenttien parametreja. Isot kirjaimet P, D ja Q ovat taas kausiluontoisiin komponentteihin liittyviä parametreja. Pienellä s kirjaimella merkitään jaksollisen intervallin pituus. Liikennedatassa mittausten intervalli voi olla 15 minuuttia ja tutkittava jakso on rajoitettu arkipäiviin. Näillä tiedoilla voidaan laskea jaksollisen intervallin pituus, joka on 480 (Xu ym. 2014). Funktion muuttujalla p kuvataan autoregressiivisten termien järjestystä. Muuttujalla d kuvataan erottelevien termien järjestystä ja muuttujalla q kuvataan liukuvan keskiarvon termien järjestystä. Muuttujilla P, D ja q kuvataan samoja asioita kuin muuttujilla p, d ja q, mutta vain jaksollisena (Noureen ym. 2019).

$$SARIMA(p,d,q)(P,D,Q)_s \tag{4.1}$$

SARIMA-mallia käytetään usein muuttumattomalla datalla, mutta pienten muutosten jälkeen sitä voidaan hyödyntää myös muuttuvan datan analysointiin. Muuttuvasta datasta saadaan muuttumatonta muun muassa erottelemalla datasta oleelliset osat, pakkaamalla data tai keräämällä datasta tiettyjä ilmiöitä pidemmältä ajalta. SARIMA-mallin luomiseksi ensimmäisenä tulee tunnistaa kausiluontoiset ja trendejä muistuttavat kaavat. Tämän jälkeen data tulee erotella siten, että siinä on selkeästi esillä sekä kausiluontoisuus että löytyneet trendit. Seuraavaksi data analysoidaan käyttäen muita metodeita, jotta saadaan p:n ja q:n arvot selville sekä kausiluontoiselle että tavalliselle datalle. Seuraavaksi arvioidaan mallin kausiluontoisuus ja tarkistetaan, onko malli sopiva tilanteeseen (Noureen ym. 2019). Vaiheessa kolme eli datan analysoimisessa, käytetään usein metodeina autokorrelaatiofunktiota (ACF) ja osittaista autokorrelaatiofunktiota (PACF). Nämä funktiot muodostavat kuvaajan, josta voidaan lukea kausiluontoisesti toistuvan kaavan aikaväli. Kuvaajassa positiivisella puolella olevat piikit ilmestyvät tasaisin väliajoin, joten asteikolta voidaan suoraan lukea välimatkan pituus (Lie ym. 2013)

4.1.4 TensorFlow

TensorFlow on Googlen kehittämä avoimen koodin alusta, joka on tarkoitettu järjestelmille, jotka käyttävät tekoälyä. Alusta julkaistiin loppuvuodesta 2015 korvaamaan sen edeltäjä

Distblief, jossa oli toimintaan vaikeuttavia ongelmia. TensorFlow on saatavilla myös kirjastona, jota käytetään usein tekoälyllä ja Pythonilla tehdyissä sovelluksissa. TensorFlow:n avulla on mahdollista analysoida Big dataa reaaliaikaisesti, joka normaalisti on hyvin raskasta laskennallisille komponenteille. Alustan avulla on mahdollista suorittaa useita yhtäaikaisia laskutoimituksia hajautetuissa järjestelmissä, joten käytössä on paljon enemmän laskutehoa kuin tavallisissa järjestelmissä (Dehghan-Banadaki, Taufik ja Feliachi 2018).

Koska alusta ja kirjasto on toteutettu avoimen lähdekoodin periaatteiden mukaisesti, esimerkiksi Airbus, Lenovo ja muut isot yritykset käyttävät TensorFlow:ta omissa projekteissaan. Koska alusta skaalautuu erittäin hyvin usealle laitteelle ja käyttöjärjestelmälle, sen käyttäminen on helppoa ja tehokasta. Alusta ja kirjasto eivät myöskään ole sidottuja vain yhteen kieleen, vaikka usein niiden yhteydessä käytetään Pythonia. TensorFlow:ta hyödynnetään myös biometriikassa, konenäköön liittyvissä sovelluksissa sekä puheen tunnistamisen yhteydessä (Dehghan-Banadaki, Taufik ja Feliachi 2018).

TensorFlow on kehitetty yksittäin koodattujen pilvipohjaisten ratkaisujen perusteella, jotta Big datan käsittely olisi helpompaa ja järkevämpää. Yleisesti nämä yksittäiset pilvipohjaiset koodit ovat olleet perustana erilaisille neuroverkkoja kehittäville ohjelmistoille. Nimi TensorFlow tulee neuroverkkojen suorittamista operaatioista. Sen avulla on helpompi luoda parempaa oppimiskäyttäytymistä koneoppimisen parissa ja tiedot voidaan sijoittaa malleihin huomattavasti aiempaa nopeammin. Debuggausta eli virheiden jäljittämistä ja visualisointia varten on kehitetty TensorBoard, joka visualisoi koko rakennus- ja opetusprosessin (Fuente, Erazo ja Smith 2018).

4.2 Datan hyödyntäminen

Muutamia menetelmiä on kehitetty liikennemäärien arvioimiseen, vaikka osa liikenteestä jäisi havaitsematta sensorien puutteiden takia. Näillä menetelmillä voidaan arvioida liikenne, vaikka jotkin sensorit eivät lähettäisi kunnolla dataa tai olisivat vain väliaikaisesti tietyillä paikoilla. Menetelmät takaavat myös sen, että väliaikaisten sensorien avulla voidaan saada kattavampi kuva kaupungin liikenteestä ilman, että siihen pitäisi sijoittaa valtavia summia. Yleisimmät menetelmät puuttuvan datan arvioimiseen voidaan jakaa kolmeen kategoriaan:

interpolointi, ennustus ja staattinen oppiminen. Liikennevaloissa, joissa on käytössä monisilmukkainen sensori, voidaan datan täydentämiseen hyödyntää myös tensor-pohjaisia menetelmiä (Wang ja Mao 2018).

Yksi uusi malli on Optimum closed cut -malli, jossa hyödynnetään Kriging arvioijaa. Jotta tätä menetelmää voitaisiin käyttää jonkin tiealueen liikenteen arvioimiseen, tulee tien täyttää kolme ehtoa. Ensimmäinen ehto on se, että kohteena oleva tiealue risteää sellaisen alueen kanssa, josta saadaan tietoa sensoreiden avulla. Toinen ehto on se, että lähellä olevien sensoreilla varustettujen teiden liikenteellä on sama maksimi korrelaatiossa kuin kohteena olevalla tiellä. Kolmas ja viimeinen ehto on se, että kohteena oleva tien risteykset muiden teiden kanssa on minimoitu (Wang ja Mao 2018).

Yleisesti monessa järjestelmässä ei ole suunnitelmia tai algoritmeja, jotta kadonnut data voitaisiin takaisinmallintaa oikeaksi. Jos vastaan tulee tilanne, jossa sensori havaitsee väärin kohteitaan tai ei toimi ollenkaan, usein ratkaisuna on vaihtaa viallinen sensori tai korjata se. Harvemmin keräämätöntä dataa yritetään palauttaa vanhojen arvojen avulla, koska tähän kuluu paljon resursseja eikä useat järjestelmät hyödynnä dataa siinä määrin, että tällainen toiminta olisi kannattavaa. Virheellisen datan kohdalla menetellään usein kahdella eri tavalla joko kyseiset kirjaukset poistetaan täysin tai ne jätetään paikoilleen, mutta merkitään tietoihin virheelliseksi. Molemmissa tapauksissa datan eheys kärsii hetkellisesti, mutta virheen korjauksen jälkeen tilanne normalisoituu suhteellisen nopeasti (Tan ym. 2010).

5 Data-analyysi

Tässä luvussa kuvataan Kalman-suodattimella, PCA:lla, SARIMA:lla ja TensorFlow:lla tehtyjä data-analyyseja ja ennustuksia. Kaikissa menetelmissä käytetyt datat on haettu Power BI:n avulla Microsoft Azuren SQL-tietokannasta ja niitä on muokattu aina jokaisen menetelmää koskevan koodin vaatimalla tavalla. Useimmissa menetelmissä koronaviruksen vaikutukset on huomioitu joko mainitsemalla siitä tekstissä tai mahdollisuuksien salliessa käyttämällä sellaisia kuukausia datasta, johon rajoitukset eivät ole vielä vaikuttaneet.

5.1 Kalman-suodattimella saadut tulokset

Kalman-suodattimella pyrittiin poistamaan datasta kaikki melu, joka aiheutuu vääristä havainnoista ja sensoreiden ominaisuuksista. Data, joka on kerätty sekä induktiosilmukoista että WiFi-laitelaskureista, sisältää paljon taustamelua, joka vaikuttaa data-analyysien ja ennustuksien toteutumiseen. Liikenteestä kerätyn datan kanssa käytettiin perinteistä Kalman-suodatinta, koska siitä on tehty toteutus Pythonille. Pythonia käytetään useimmiten koneoppimisessa laajojen kirjastojensa takia ja datan jatkokäsittelyn kannalta koodit kannattaa toteuttaa samalla ohjelmointikielellä. Analysoinnissa käytetty koodi on kokonaisuudessaan esitetty liitteessä C.

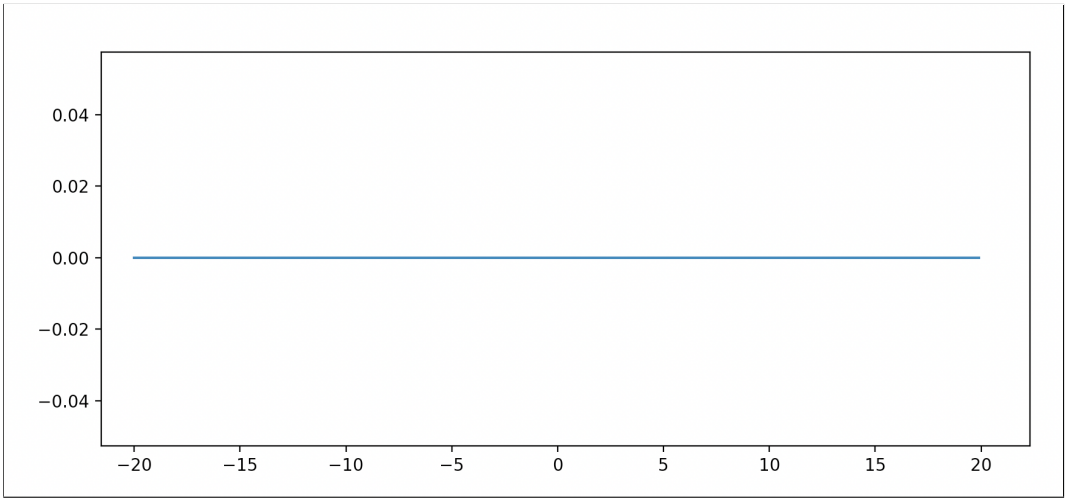
Koodia varten data täytyi muuttaa oikeanlaiseen muotoon, jotta sitä voidaan käyttää laskutoimituksissa. Kaikista mittauksista kerätty data tulee olla float-tyyppisiä, jotta niiden avulla voidaan tehdä laskelmia menetelmällä. Laskelmia varten jokaisesta mittauksesta tuli laskea askeleet seuraavaan mittaukseen eli mittausten erotus sekä laskea mittausten ja askelien sigma-arvot. Yleinen sigma-arvo eli epävarmuus laitettiin korkeaksi, mutta se laskee lähemmäs nollaa laskutoimitusten aikana. Mitä pienempi sigma-arvo, sitä tarkempi lopputulos on. Muuttujalla mu kuvataan tilan alkuarvoa, joka muuttui jokaisella iteraatiokerralla.

Koodi etenee aina yksi iteraatiokierros kerrallaan kunnes kaikki mitatut arvot on käyty läpi. Jokaisen kierroksen alussa päivitetään tilasta tehtyä arviota ja tulostetaan konsoliin uusi arvio sekä tilasta että sen epävarmuudesta. Tämän jälkeen tehdään ennustus tulevasta tilasta, jonka tiedot tulostetaan konsoliin. Sekä päivittämiseen että ennustamiseen käytetyissä funktioissa

lasketaan uusi keskiarvo ja uusi arvio tilan epävarmuudesta. Jokaiselta iteraatiokierrokselta konsoliin tulostetaan arvoja kunnes viimeisen kierroksen jälkeen tulostetaan lopullinen tila ja sen epävarmuus. Kuviossa 5 on koodin tekemää päivitystä ja ennustusta muutamalta iteraatiokierrokselta.

```
Update: [250550.50154111368, 8.902921800749363]
Predict: [266032.5015411137, 21.90292180074936]
Update: [211427.02328964323, 8.902921800749363]
Predict: [291464.0232896432, 21.90292180074936]
Update: [201894.68086984596, 8.902921800749363]
Predict: [235371.68086984596, 21.90292180074936]
```

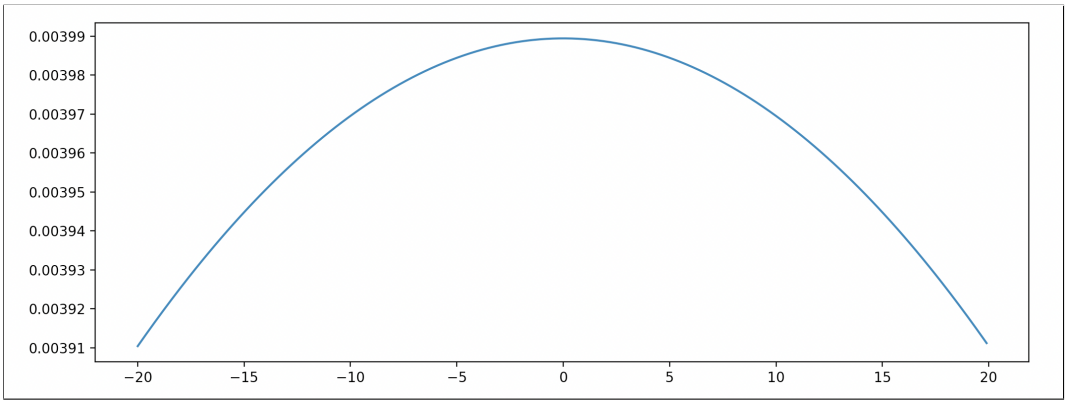
Kuvio 5. Kalman-suodattimella tehtyjen iteraatiokierrosten päivittämisen ja ennustamisen jälkeiset tilat ja epävarmuudet.



Kuvio 6. Kalman-suodattimella tehtyjen iteraatiokierrosten jälkeinen tulos.

Kuviosta 6 nähdään, että koodin tuottama kuvaaja on suora viiva eikä se muistuta yhtään Gaussin käyrää eli normaalijakaumaa. Usein Kalman-suodattimen lopputulos ei ole suora viiva vaan jonkinlainen käyrä kuvaaja, joka kuvastaa arvojen jakautumista. Kuvaaja voi olla tavallista kuvion 7 esittämää yleistä Gaussin käyrää kapeampi tai loivempi, jolloin data jakautuu eri tavoin. Kuvaajassa näkyvä käyrä voi myös sijoittua x-akselilla eri kohtaan, jonka avulla voidaan tulkita datan jakautumista. Optimaalisessa tilanteessa kuvaajan esittämä jakauma olisi muuttunut hieman, mutta siinä olisi selkeästi nähtävillä huippu ja leveys. Tämä

kuvastaisi, että data jakautuisi laajemmalle alueelle kuin yhden suoran muodostama kuvaaja, joka kuvastaa hyvin muuttumatonta dataa.



Kuvio 7. Kalman-suodattimella tehty perinteinen Gaussin käyrä, joka on iteraatiokierrosten lähtökohta.

Data, jota käytettiin Kalman-suodattimen kanssa ei noudata normaalijakaumaa, joten se vaikuttaa tuloksiin. Ennen koodin ajamista datalle oli pakko tehdä esikäsittely, jossa laskettiin tiettyjä arvoja valmiiksi, jotka tulevaisuudessa olisi hyvä saada tehtyä koodin avulla. Koska dataa oli alle 150 riviä, käsin laskemisesta ei aiheutunut suuria ongelmia, mutta on aina mahdollisuus inhimillisiin virheisiin. Mikäli tulevaisuudessa halutaan hyödyntää Kalman-suodatinta melun poistamiseen, tulee dataa olla huomattavasti enemmän ja sen tulisi noudattaa Gaussin käyrää. Kun suodatinta käytetään isommalle määrälle dataa, tulee data lukea jostain toisesta tiedostosta, kuten esimerkiksi data-tiedostosta. Tiedostossa kaikki luvut tulee olla float-tyyppisiä ja erotettuina pilkuilla, jotta koodi osaa tulkita ne oikein. Nykyinen koodi, jossa luvut on koodattu suoraan koodiin, ei ole järkevä vaihtoehto tulevaisuudessa vaan luvut tulisi lukea tiedostosta samalla tavalla kuin SARIMA:n, PCA:n tai TensorFlow:n tapauksissa tehdään. Kalman-suodatin ei sovellu isojen määrien reaaliaikaiseen laskemiseen, koska kyseessä on suhteellisen hidas menetelmä. Jos tulevaisuudessa käytetään Kalman-suodatinta, tulee varmistaa laskentaan käytettävän palvelimen laskentateho sekä varata riittävästi aikaa. Mieluiten suodatinta hyödynnettäisiin vain sellaisissa tapauksissa, joissa analysointi tehdään rauhassa eikä tuloksia tarvitse syntyä nopeasti.

5.2 SARIMA:lla saadut tulokset

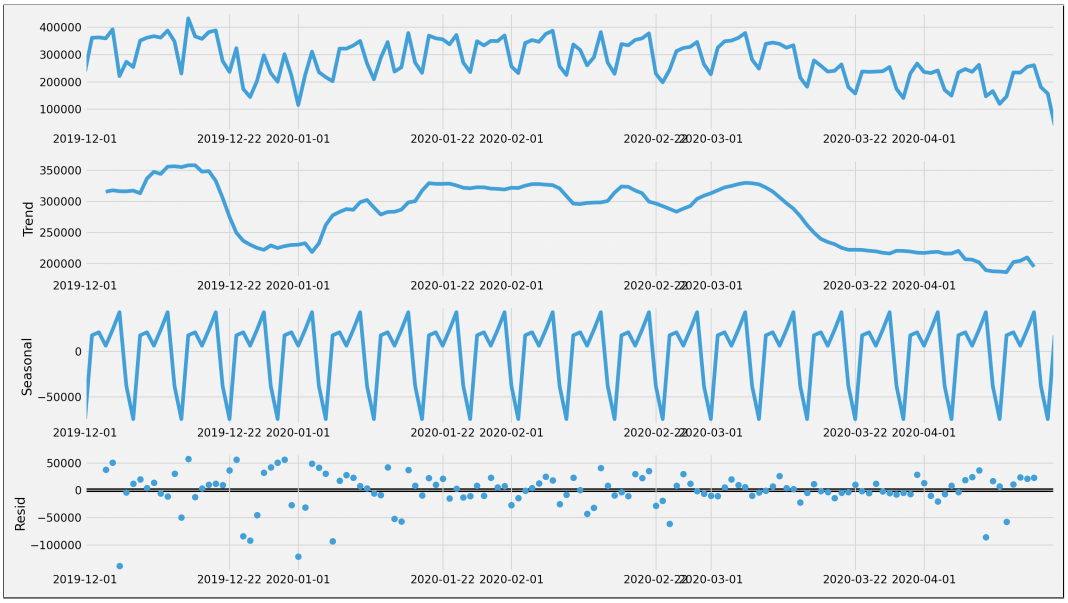
SARIMA:lla eli kausiluontoisella ARIMA:lla pyrittiin ennustamaan tulevaa liikennettä samalla ottaen huomioon siinä kausittain tapahtuvat muutokset. Kausiluontoisella ennustamisella pyritään saamaan tarkka ja todenmukainen kuva tulevasta liikenteestä. Koska liikenne ei ole yhdenmukaista päivästä toiseen vaan siihen vaikuttavat monet asiat, kuten esimerkiksi viikonloppu ja työmatkaliikenne, tulee liikenteestä erottaa kausia. Näiden kausien liikenne on keskenään erilaista, mutta yhden kauden sisällä liikenne noudattaa yleisesti samoja sääntöjä.

Liitteessä D esitetylle koodille annetaan tiedot Excel-tiedostona, mutta pienillä muutoksilla voidaan hyödyntää myös CSV-tiedostoa. Aluksi koodissa pyritään tarkastelemaan erilaisten kuvaajien avulla dataa ja selvittämään siitä kausiluontoisuus sekä muut ennustamisen kannalta oleelliset ominaisuudet. Seuraavaksi pyritään löytämään funktion parametrit, joiden avulla ennustuksesta saadaan mahdollisimman tarkka. Tämä tehdään Akaike informational criterian eli aic-arvon perusteella, jossa tarkoituksena on löytää vertailemalla malli, jossa kadotetaan mahdollisimman vähän tietoja (Lie ym. 2013). Vertailun perusteella käytössä olevasta datasta saadaan paras mahdollinen ennustus käyttämällä kaavion 5.1 mallia. Kyseisen mallin aic-arvo oli lähimpänä nollaa, mutta lukeman tulisi olla paljon pienempi ennustuksen luotettavuuden kannalta. Datan vähyyden takia aic-arvon lukema on yli 2400, mutta luotettavien tulosten kannalta lukeman tulisi olla alle 200.

$$SARIMA(1,1,1)(0,1,1,20)20. \tag{5.1}$$

Koodin seuraavassa osiossa käytetään kaavion 5.1 mallia dataan ja pyritään selvittämään siitä asioita, kuten esimerkiksi korrelaatio ja otoksen osuuspisteiden suhde teoreettisiin osuuspisteisiin, jonka avulla voidaan selvittää datan jakauma. Kuviossa 9 näkyvät funktion tekemät kuvaajat, joiden avulla datan ominaisuudet on esitetty. Käytössä oleva data noudattaa normaalijakaumaa, koska kuvaajassa näkyvät pisteet noudattavat lineaarisesti kasvavaa käyrää.

Ensimmäinen SARIMA:lla tehdyn ennustuksen tavoitteena oli ennustaa liikennemääriä mallilla, jossa aluksi ennustetaan yksi askel eli yksi päivä eteenpäin. Ennustamisen jälkeen ennen seuraavan askeleen ennustamista, datasta otetaan mukaan kyseisen päivän laskettu liikenne. Mallia toistetaan kunnes ennustamiseen käytettävää dataa ei ole. Mallissa voidaan määrit-

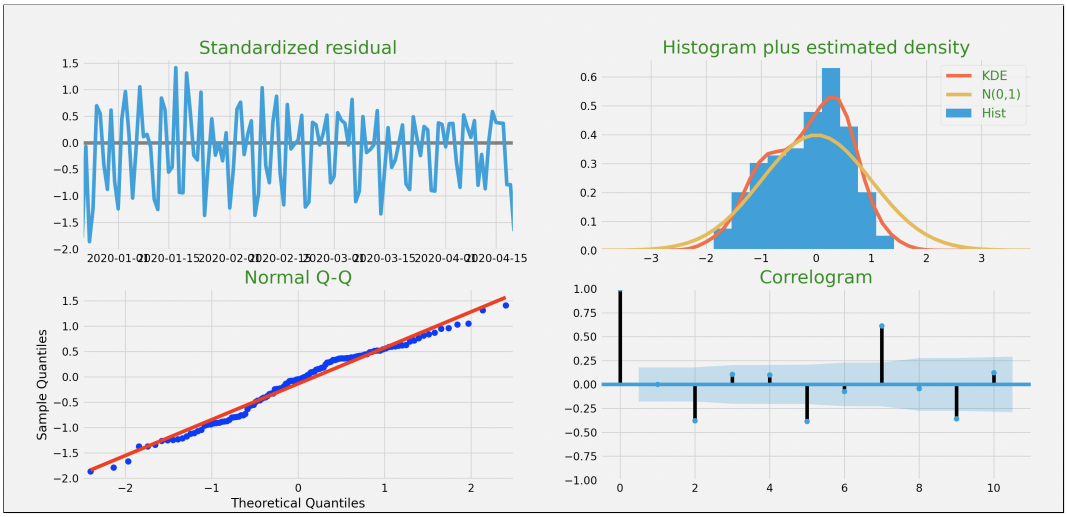


Kuvio 8. Liikenteestä kerätystä datasta ennustamisen ja kausiluontoisuuden kannalta oleelliset ominaisuudet.

tää aloituspäivä, jonka jälkeen tulevaa dataa pyritään ennustamaan ja ennustusta verrataan näiden päivien tietoihin. Käytettäessä yhden askeleen ennustamista tulee huomioida se, että aloituspäivän tiedot tulee olla datassa ja myös sen jälkeen tulee olla mittauksia. Mikäli aloituspäiväksi laitetaan jokin sellainen päivä, jolta ei ole dataa, ennustus epäonnistuu täysin.

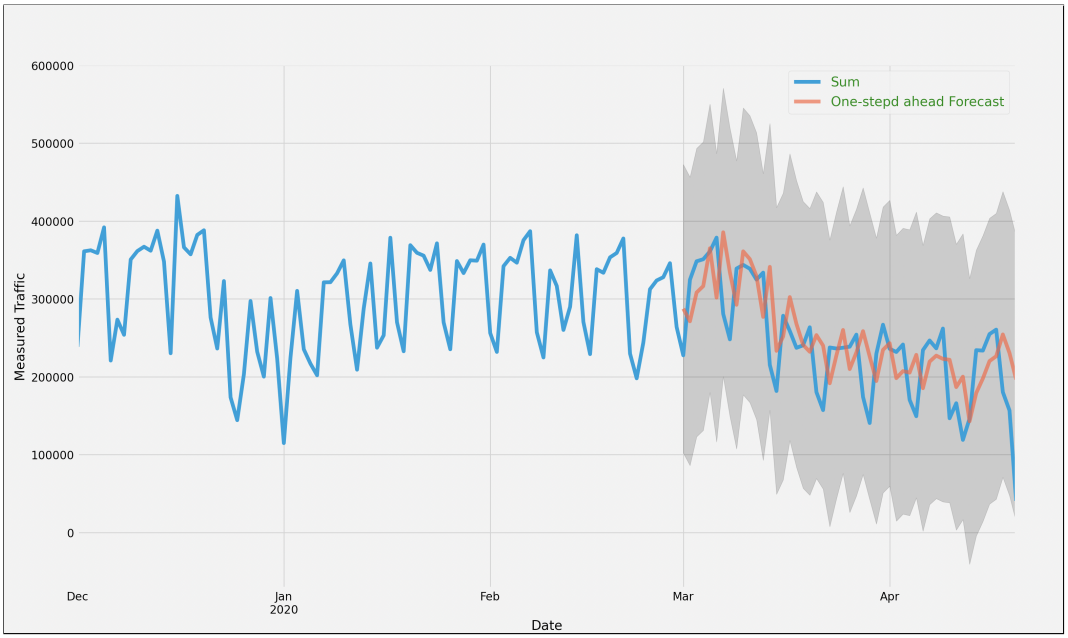
Kuviosta 10, jossa askel kerrallaan ennustus on kuvattuna punertavalla värillä, voidaan tulkita, ettei ennustus onnistunut kuvaamaan todellista tilannetta. Koodissa tutkittiin myös keskineliövirhettä ja keskineliövirheen neliöjuurta, joiden avulla voidaan arvioida ennustuksen tarkkuutta. Liikenteestä kerätyllä datalla, jota on käytettävissä joulukuun alusta alkaen, keskineliövirhe on 10-numeroinen luku ja keskineliövirheen neliöjuuri on 5-numeroinen luku. Mitä pienempiä keskineliövirhe ja sen neliöjuuri ovat, sitä tarkempia ennustuksen tulokset ovat. Koodin lopussa tehtiin myös pidempiaikainen ennustus, joka on pituudeltansa 20 päivää ja ulottuu huhtikuun loppupuolelta toukokuun puoleen väliin. Ennustuksen tulos on esitetty kuviossa 11, josta on nähtävissä samanlainen trendi kuin edellisten kuukausien kohdalla. Kuvaajasta on selkeästi luettavissa kohta, jossa kerätty data päättyy huomattavasti alemmas kuin ennustuksen ensimmäinen lukema.

SARIMA:lla tehdyt ennustukset eivät ole tarkkoja johtuen sekä datasta että Pythonilla teh-

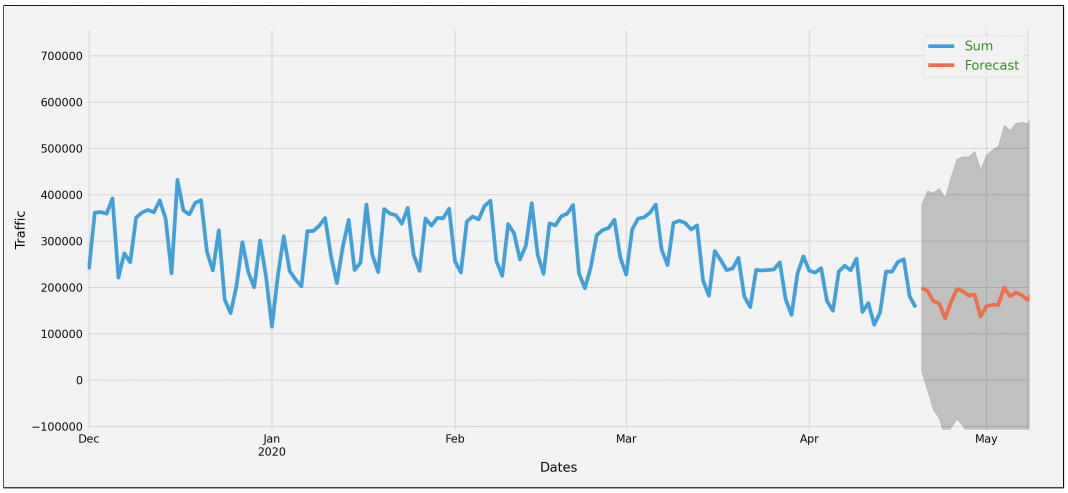


Kuvio 9. SARIMA:n avulla datasta löydetyt ominaisuudet, joita käytetään ennustamiseen.

dyn menetelmän rajoituksista. Ennustusta varten dataa tulisi olla useammasta mittauksesta ja pidemmältä aikaväliltä. Koodissa käytetty data sisältää päivittäin yhteen kerätyn yhteismäärän liikenteestä joulukuun ensimmäisestä päivästä hieman yli huhtikuun puoleen väliin. Mittauksia on oikeasti useita tunnissa, joten yhtä päivää kohden olisi saanut 96 mittausta, joka olisi huomattavasti parantanut ennustamisen tarkkuutta. Python kirjasto, jossa SARIMA-menetelmä sijaitsee ei mahdollistanut sellaisen datan käyttämistä, jossa yhtä päivää kohden on monta mittausta. Jotta menetelmällä saataisiin aikaan parempia ennustuksia, tulee dataa olla kerättynä paljon pidemmältä ajalta, mieluiten muutaman vuoden ajalta. Toinen vaihtoehto on yrittää löytää jokin korjaus menetelmän päiväkohtaisten mittausten rajoitukseen. Molemmilla tavoilla tarkoituksena on saada SARIMA-mallien aic-arvo sekä keskineliövirhe ja sen neliöjuuri mahdollisimman lähelle nollaa, jolloin ennustukset ovat tarkkoja. Tulevaisuudessa ennustaminen kannattaa tehdä erillisellä palvelimella, joka on tarkoitettu koneoppimiseen ja laskentaa, koska tavallinen kannettava tietokone ei ole optimoitu ennustuskoodien ajamiseen.



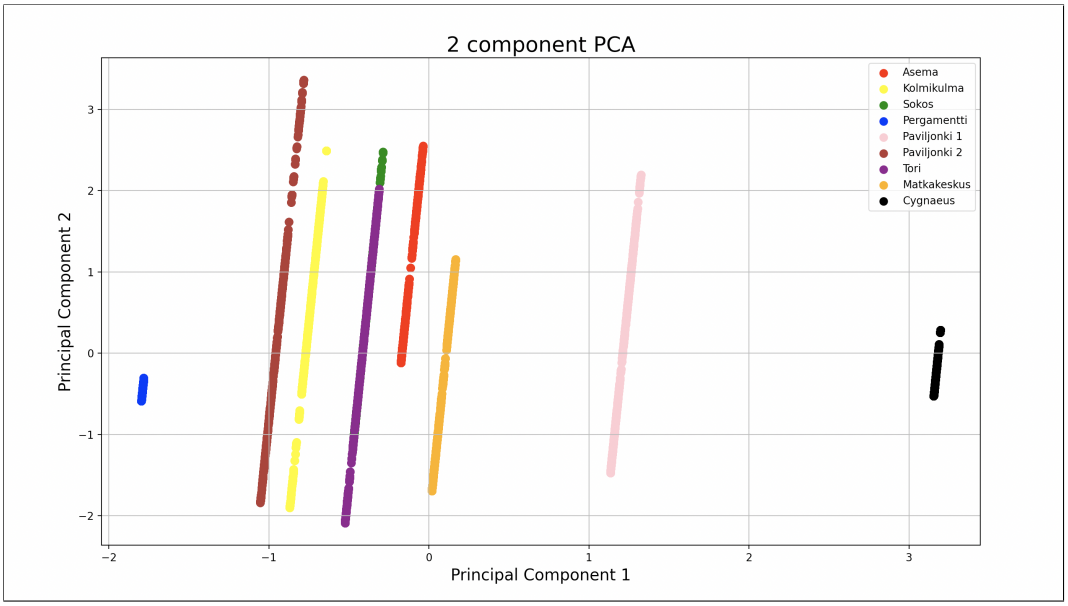
Kuvio 10. SARIMA:n avulla tehty ennustus, joka etenee askel kerrallaan.



Kuvio 11. SARIMA:n avulla tehty pidempi ennustus, joka ulottuu toukokuun puoleen väliin asti.

5.3 PCA:lla saadut tulokset

PCA:lla eli pääkomponenttianalyysilla pyrittiin analysoimaan liikennedataa ja katsomaan, voidaanko datasta tunnistaa eri parkkihallit tiettyjen ominaisuuksien perusteella. Analysointiin käytetty koodi kirjoitettiin Pythonilla ja siinä hyödynnettiin kirjastoja Pandas, Matplotlib ja Sklearn, joita käytetään koneoppimisessa visualisointiin ja opettamiseen. Data annetaan koodille data-tiedostona, joka muistuttaa CSV-tiedostoa, mutta siitä puuttuu otsikkorivi ja erottimena käytetään pilkkua puolipisteen sijasta. Käytetty data tulee aluksi standardoida sekä muuttaa kaksiulotteiseksi, jotta voidaan hyödyntää PCA:n menetelmiä. Koodissa täytyy jokaiselle tunnistettavalla luokalle, tässä tapauksessa parkkihallille, antaa yksilöivä nimi ja väri. Näiden tunnistetietojen avulla kuvaajasta on helpompi lukea, onko siinä päällekkäisiä tuloksia ja miten kohteet ryhmittyvät. Kokonaisuudessaan pääkomponenttianalyysissa käytetty koodi on nähtävissä liitteessä B.



Kuvio 12. PCA:lla saadut tulokset parkkihallien datasta.

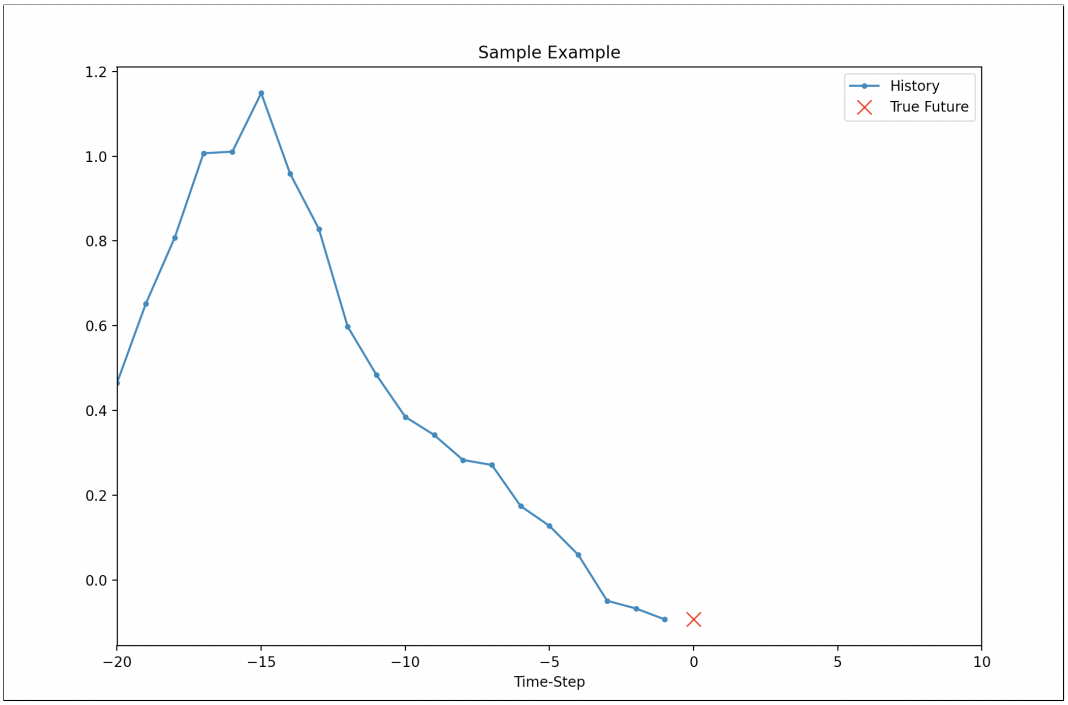
PCA:lla pystyttiin hyvin tunnistamaan eri parkkihallit toisistaan annetun datan perusteella, joka koostui parkkipaikkoihin liittyvistä lukemista. Kuusi parkkihallia on lähellä toisiaan, kaksi hieman erillään isommasta ryhmästä ja yksi parkkihalli on täysin erillään muista kuten kuviosta 12 voidaan tulkita. Parkkihalleista Sokos ja Tori ovat lähimpänä toisiansa, osa pisteistä on hieman päällekkäin. Tämä johtuu siitä, että datassa molemmilla parkkihalleilla

on yhtä paljon parkkipaikkoja, joka vaikuttaa datan arvoihin. PCA on vielä nykyäänkin hyvä vaihtoehto eri luokkien tunnistamiseen datasta, koska sen avulla liikennedatasta voidaan tunnistaa eri komponentteja ja luokitella ne omiin luokkiinsa.

PCA:ta käyttäessä on otettava muutama asia huomioon, jotta saadaan aikaan luotettavia tuloksia. Datassa tulee olla useampia ominaisuuksia tai sarakkeita, joita voidaan hyödyntää tunnistamisessa. Dataa tulisi olla myös mahdollisimman monta riviä, jotta menetelmällä on tarpeeksi materiaalia tunnistamisen rakentamiseksi. PCA:ta voidaan käyttää sekä luokitteluun että koneoppimisen nopeuttamiseen neuroverkkojen avulla, mutta siihen vaaditaan enemmän ominaisuuksia ja rivejä dataa, kuin liikennedatasta on tällä hetkellä saatavilla. Mieluiten yhdellä tunnistettavalla kohteella tulisi olla useita kymmeniä ominaisuuksia ja dataa tulisi olla kymmeniä tuhansia rivejä. PCA:ta voidaan tulevaisuudessa hyvin hyödyntää eri luokkien tunnistamiseen datasta sekä sen tuottamaa tietoa voidaan hyödyntää muissa ennustuksissa. Mikäli liikenteestä on mahdollista saada lisää ominaisuuksia, jotka auttavat tiettyjen luokkien tunnistamisessa, voidaan PCA:ta hyödyntää paljon laajemmin koneoppimisessa yhdessä tekoälyn kanssa.

5.4 TensorFlow:lla saadut tulokset

TensorFlow:lla tehtiin kaksi ennustusta erilaisilla menetelmillä, joiden avulla pyrittiin ennustamaan liikennettä muutaman tunnin päähän. Ennustamisessa käytetty koodi kirjoitettiin Pythonilla ja siinä hyödynnettiin TensorFlow:n lisäksi kirjastoja Numpy, Matplotlib ja Pandas, joita käytetään usein koneoppimisessa. Koodissa data otetaan vastaan CSV-tiedostona, josta haettu data pyritään standardoimaan analysointia varten. Koodissa täytyy määritellä, kuinka monta riviä tiedostosta käytetään mallin opettamiseen ja mistä kohtaa ennustettava data alkaa. Mallille annetaan myös tieto siitä, montako edellistä mittausta se saa käyttää ennustamiseen ja montako tulevaisuuden arvoa sen tulee ennustaa. Koodi tuottaa aluksi kuvaajan, josta nähdään sekä edelliset arvot että arvo, jonka mallin tulisi ennustaa. Tämän jälkeen koodi laskee yksinkertaisen mallin mukaisen ennustuksen ja tuottaa siitä kuvaajan. Koodissa käytetään myös neuroverkkoja monimutkaisemman mallin opettamisessa, jonka jälkeen malli ennustaa tulevan arvon ja tuottaa ennustuksesta kuvaajan. TensorFlow:lla ennustamisessa käytetty koodi on esitetty liitteessä A.

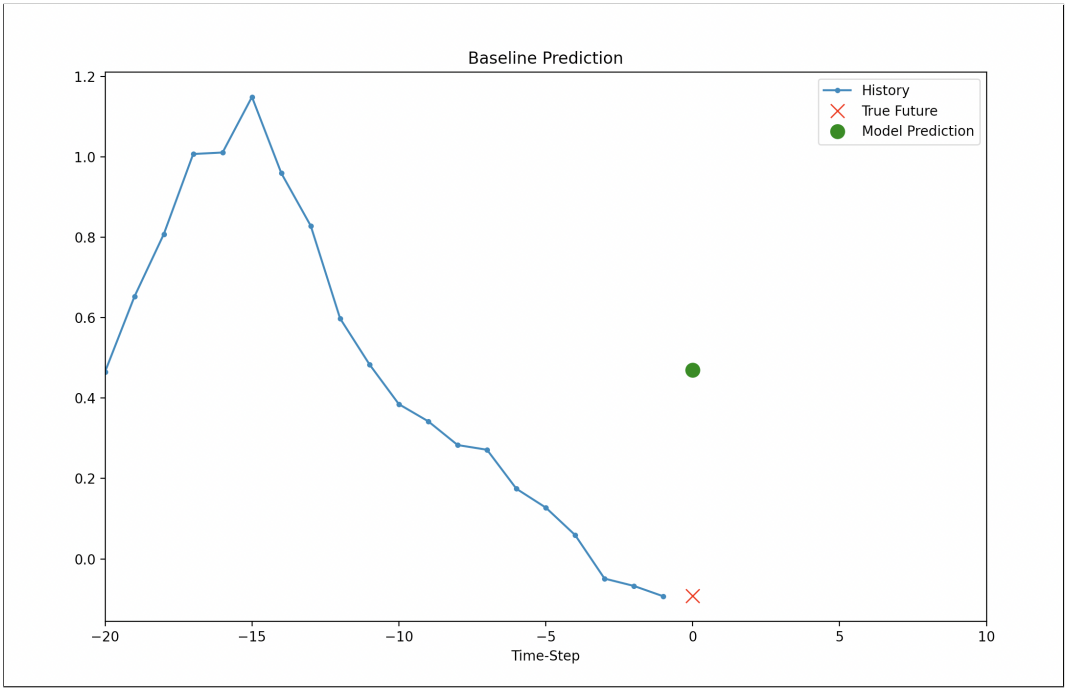


Kuvio 13. Arvo, jonka TensorFlow:lla tehdyn mallin tulee ennustaa.

TensorFlow:lla tehty ensimmäinen ennustus hyödyntää yksinkertaista mallia, joka ei vielä hyödynnä montaa koneoppimisen metodia. Mallilla tehdyn ennustuksen tulos ei ole lähellä oikeaa arvoa vaan poikkeaa siitä huomattavasti, kuten kuviosta 14 voidaan nähdä. Ihanteellisessa tapauksessa malli olisi osannut ennustaa oikean arvon, mutta todella usein yksinkertainen ennustusmalli ei kykene tuottamaan oikeaa arvoa. Tapaukset, joissa yksinkertainen malli kykenee ennustamaan oikean arvon, ovat usein sellaisia, että niissä arvojen vaihtelu eri indekseillä on lähes olematonta. Tällaisissa tapauksissa myös ihminen kykenee suhteellisen helposti ennustamaan oikean arvon, koska se ei vaadi monimutkaisia matemaattisia laskukaavoja tai analyyseja. Yksinkertainen malli ei yksinään kunnolla sovellu liikennedatasta tehtävään ennustukseen.

TensorFlow:lla voidaan tehdä monimutkaisempia ennustusmalleja, joissa hyödynnetään neuroverkkoja. Mallissa käytetään yksinkertaista LSTM-malli, joka hyödyntää Long Short-Term memory -kerrosta neuroverkkojen opettamiseen. Mallilla voidaan tuottaa useita peräkkäisiä kuvaajia, joissa jokaisesta on uusi ja paranneltu ennustus. Koodissa tuotetaan kolme kuvaajaa, mutta datan vähyyden takia kuvaajat eivät eroa toisistansa merkittävästi. Monimutkaisel-

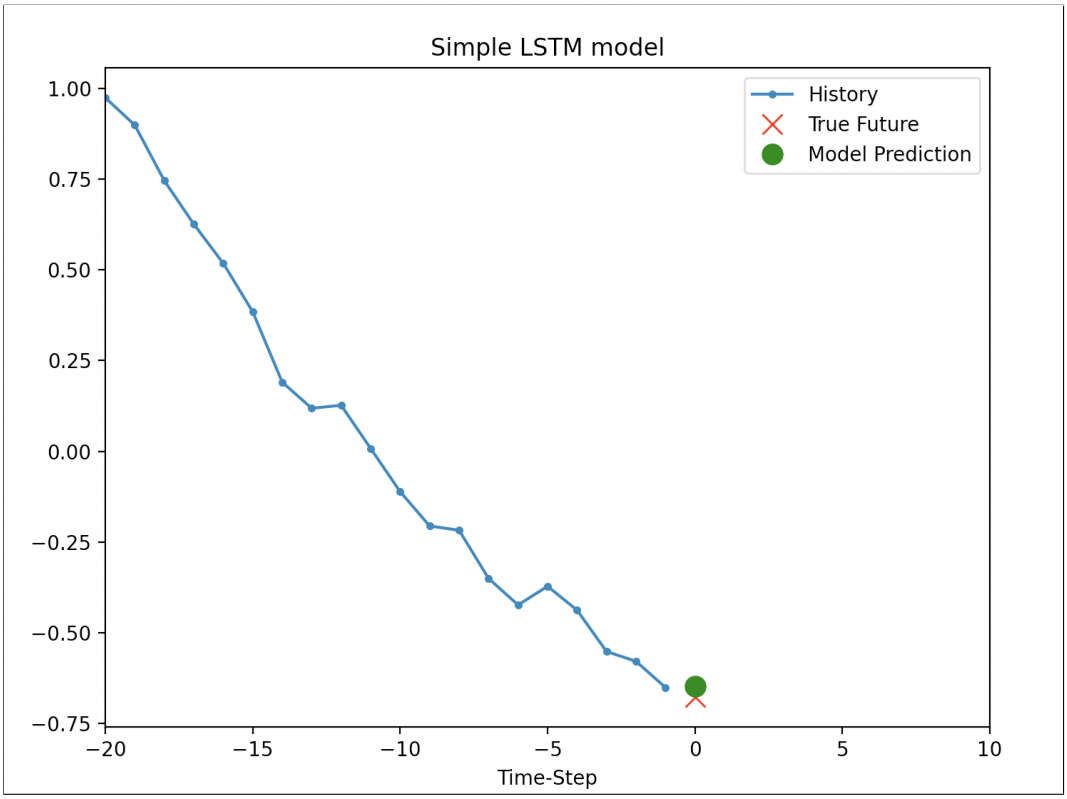
le mallille annettava harjoitusdatan koko sekä ennustettavien arvojen joukko, määrittelevät pitkälti sen, kuinka hyvin ennustus onnistuu. Liikennedatan tapauksessa mallille on annettava vähintään 8100 indeksiä harjoitusdataa ja ennustettavien arvojen joukko on oltava 2000 indeksiä, jotta ennustus onnistuisi. Monimutkainen malli pystyy ennustamaan tulevan arvon paljon paremmin kuin yksinkertainen malli. Kuviossa 15 nähtävä vihreä pallo, jolla merkitään ennustettua arvoa, on melkein punaisen rastin päällä. Yksinkertaisen mallin ja neuroverkkoja käyttävät mallin kuvaajat eroavat toisistansa sekä y-akselin asteikon puolesta että historiaa kuvaavan viivan muodosta.



Kuvio 14. TensorFlow:lla tehdyn yksinkertaisen mallin ennustuksen lopputulos.

TensorFlow:lla ennustusta kokeiltiin tehdä muutaman tunnin päähän, koska käytössä olevaa liikennedataa oli saatavilla vain 5 kuukauden ajalta ja siitä osa oli käyttökelvotonta koronaviruksen aiheuttamien poikkeusolojen takia. Pidemmän ajan ennustukset vaativat paljon laskentatehoa tietokoneelta, joten ne on järkevintä toteuttaa erillisillä laskentaan tarkoitetuilla palvelimilla eikä kannettavilla tietokoneilla. Kannettavalla tietokoneella tehdyt laskennat kuitenkin osoittavat, että pidemmän ajan ennustaminen liikenteestä kerättävällä datalla on täysin mahdollista. Käytössä täytyy olla tehokkaampi palvelin, joka on tarkoitettu dataanalyysin ja laskentaan sekä data tulee olla kerättynä vähintään vuoden ajanjaksolta. Ennus-

tamiseen käytettävässä datassa tulisi kiinnittää huomiota myös sen oikeellisuuten ja mahdolliseen poikkeamiin normaalista tilanteesta. Koska ennustamista voidaan tehdä TensorFlow:n avulla myös suhteellisen pienestä määrästä dataa, voidaan sitä hyödyntää myös erikoistilanteiden liikenteen ennustamiseen. Jos tulevaisuudessa halutaan ennustaa liikennemääriä vuoden 2020 liikenteestä kerätyllä datalla, tulee ottaa huomioon koronaviruksen vaikutukset, kuten esimerkiksi autoilijoiden ja joukkoliikenteen käyttäjämäärien raju laskeminen rajoitusten takia.



Kuvio 15. TensorFlow:lla tehdyn monimutkaisemman neuroverkkomallin ennustuksen lopputulos.

6 Power BI

Raportointityökalussa havaittiin virhe 29.10.2019, jonka takia raportointityökalun ulkonäkö ja sen tiedot eivät näkyneet kunnolla. Virhe johtui siitä, että kuukausi vaihtuu keskellä viikkoa ja raportointityökalussa koitettiin tarkastella viikkoa kuluvan kuun puolella. Raportointityökalu koittaa etsiä ElasticSearch-tietokannasta tietoja sellaiselta ajalta, jota ei ole vielä alustettu. Kuukaudet alustetaan aina silloin, kun sinne viedään ensimmäiset tiedot eli yleensä kuukauden ensimmäisenä päivänä muutama minuutti yli keskiyön. Virhe ilmenee vain silloin, kun kuukausi vaihtuu keskellä viikkoa ja raportointityökalua koetetaan käyttää silloin, kun viikon viimeisen päivän kuukausi ei ole vielä alkanut.

Edellä mainittu virhe on yksi syy siihen, miksi sivuston raportointityökalu päädyttiin vaihtamaan uudempaan versioon. Uudemmassa versiossa tullaan ottamaan huomioon myös uusia sensoreita ja liikennevaloista haettua dataa. Vanha versio käytti hyödykseen paljon vuonna 2009 manuaalisesti kerättyjä tietoja, joiden pohjalta laskettiin muun muassa autoilijoiden ja julkisen liikenteen lukumääriä erinäisten kertoimien avulla. Uudessa versiossa luvut tulevat pohjautumaan ainostaan kerättyyn dataan, joten autoilijoiden ja julkisen liikenteen luvut tulevat vaihtelemaan viikosta toiseen. Tämän lisäksi kyseiset luvut tulevat paremmin edustamaan todellisuutta mittaustulosten perusteella eikä vanhojen laskelmien perusteella tehtyjen oletusten kautta.

6.1 Power BI:llä raportointi

Raportointityökalun tekemistä aloitettaessa tietokannassa ei ollut vielä tietoa linja-autolla matkustavien henkilöiden määrästä. Jotta tietoja voitaisiin käyttää kunnolla raportointityökalussa, tulee tiedot aluksi viedä Azuren SQL-tietokantaan. Tätä varten tiedot täytyy aluksi saada Jyväskylän kaupungilta ja muokata muotoon, joka on helppo viedä tietokantaan. Tiedot toimitettiin Jyväskylän yliopistolle Excel-taulukossa, josta voidaan ottaa halutut arvot ylös. Helpoin tapa viedä suuri määrä dataa Azuren SQL-tietokantaan on SQL Management -työkalun avulla, joka osaa ottaa vastaan suoraan CSV-tiedoston.

Linja-autojen kävijämäärästä saatu raakadata täytyy aluksi käydä käsin läpi ja valita siel-

tä vain oleelliset tiedot. Ilman suodattamista tietokannasta tulee todella monimutkainen ja se sisältäisi paljon tietoa, josta ei ole hyötyä JKL-Openin kannalta. Saadusta raakadatasta halutaan ottaa huomioon vain jokaisen päivän matkustajien kokonaismäärä, joka kuvastaa hyvin päivän aikana tapahtuvaa linja-autojen käyttöä. Suodatettu data viedään tietokannassa erilliseen tauluun, joka on tehty vain linja-autoista kerättävälle datalle. Linja-autojen data tullaan lisäämään raportointityökaluun myöhemmin, koska tutkimuksen toteuttamishetkellä datan päivittäminen tehdään vielä käsin. Tulevaisuudessa tarkoituksena on, että nämä tiedot saataisiin automatisoitua aina keräämisestä tietokantaan asti, jolloin niiden hyödyntäminen helpottuu.

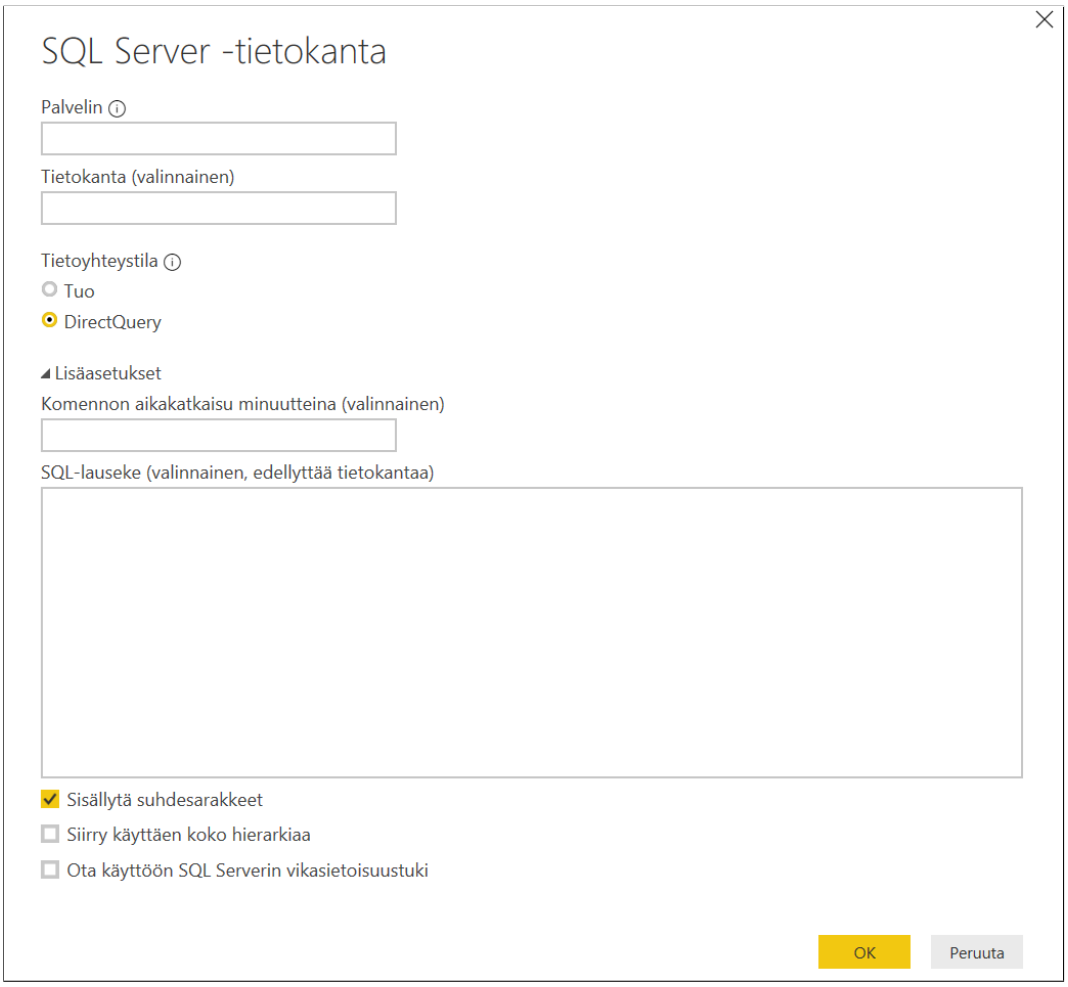
6.1.1 Tietojen hakeminen tietokannasta

Power BI työkaluun tiedot haetaan Azuren SQL-tietokannasta, jonne on viety Jyväskylän kaupungin alueen liikennetietoja. Tiedot täytyy hakea tietokannasta käyttäen SQL-kyselyitä, jotka täytyy muodostaa erikseen. Tietoja hakiessa täytyy vielä erikseen valita, mistä tietolähteestä tiedot haetaan. Koska tietokanta sisältää myös muuta kuin liikennedataa, tulee kyselyissä suodattaa epäolennaiset tiedot pois. Tietokantaan päätettiin tehdä kolme eri kyselyä, joista jokaisella haetaan tietynlaista liikennedataa. Kyselyiden kohteena ovat parkkihallit, liikennevalot ja kevyt liikenne.

Power BI sisältää valmiina jo työkalun tietojen hakemista varten, käyttäjän täytyy vain syöttää oikeat tiedot tietokannasta sekä vapaavalintaisesti erillisen SQL-kyselyn. Tarvittaviin tietoihin lukeutuvat tietokantapalvelimen nimi sekä tietokannan käyttämiseen oikeuttavat tunnukset. Näiden lisäksi pitää tietokantapalvelimen puolella varmistaa, että palomuuri sallii yhteyden ottamisen käytettävältä koneelta. Kuviossa 16 on näkyvillä työkalu, jolla tiedot haetaan Azuren SQL-tietokannasta.

Tietojen tuomista varten Power BI:ssä on kaksi erilaista tapaa. Ensimmäisessä tiedot tuodaan vain kerran haettavasta tietokannasta, jolloin päivittäminen täytyy tehdä aina käsin. Toisessa vaihtoehdossa tiedot haetaan suoran kyselyn avulla, joka mahdollistaa tietojen automaattisen päivittämisen tietyin väliajoin. Yleensä suositellaan käytettäväksi toista tapaa, koska se mahdollistaa paljon automatisoidumman visualisoinnin, joka ei vaadi käyttäjältä ja ylläpitäjältä

niin paljoa toimenpiteitä. Ensimmäiseen vaihtoehtoon päädytään yleensä silloin, kun tietokannassa on rajoitettu päivässä tehtävien kyselyiden lukumäärä. JKL-Openin tapauksessa päädyttiin käyttämään ensimmäistä tapaa, koska Power BI:n avulla tehty raportointityökalu on tarkoitus julkaista myöhemmin. Tietokannasta tuomalla haetut tiedot eivät kuormita tietokantaa jatkuvilla kyselyillä, joita julkaistussa versiossa tulisi aina kun joku henkilö avaa raportin. Koska kyseessä ei ole suora kysely, tiedot eivät päivity ilman manuaalista päivittämistä. Tästä syystä tietojen päivittäminen pyritään mahdollistamaan siten, että päivitetyt tiedot haetaan aina puolen tunnin välein ajastetusti.



Kuvio 16. Tietokannasta tietojen hakeminen SQL-kyselyn avulla.

Alla on esimerkki kyselystä, jonka avulla tietokannasta voidaan hakea parkkihallien tiedot. Nimet unit1, unit2 ja unit3 kuvaavat sarakkeiden nimiä ja value-alkuiset muuttujat kuvaavat jokaista saraketta vastaavaa arvoa. Samalla tavalla rakennetaan myös muut kyselyt, joilla

haetaan tietoja tietokannasta. Tärkeintä on muistaa antaa oikeat rajat where-ehdolle, jotta vain sellaiset rivit haetaan, jotka vastaavat toivottua lopputulosta.

```
SELECT SensorID, date, unit1, value1, unit2,
value2, unit3, value3
FROM IoTread
WHERE unit1 LIKE '%paikat';
```

6.1.2 Näkymän luominen

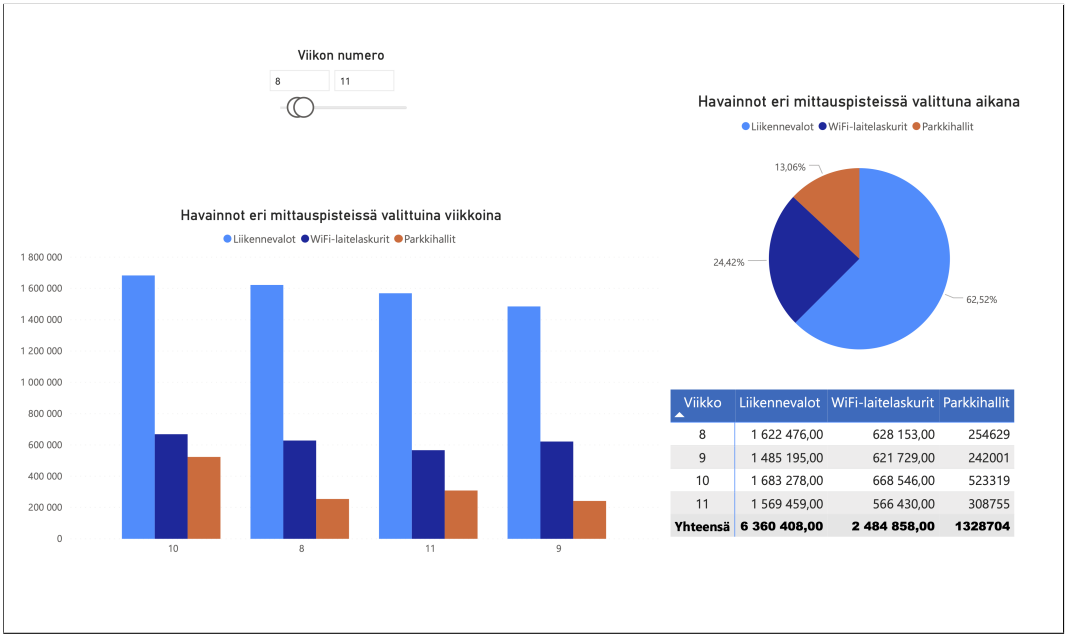
Näkymän luomista varten täytyy luoda kuvaajia, joihin tiedot otetaan tietokannasta tuodusta datasta. Yksinkertaisia kuvaajia voidaan luoda helposti Power BI:n perustoimintojen avulla. Perustoiminnot mahdollistavat yksinkertaisten suodattimien käyttämisen, kuten esimerkiksi näytetään kuvaajassa vain viimeisen viikon tiedot. Raportointityökalussa on tarkoitus käyttää dynaamisia kuvaajia, joista loppukäyttäjä itse voi vaihtaa aikaväliä, kuten esimerkiksi kalenteriviikkoja. Dynaamisten kuvaajien luomisessa ja niissä tietojen esittämisessä, täytyy käyttää monimutkaisempia suodattimia. Näitä suodattimia varten täytyy luoda uusia tauluja, joihin luodaan sarakkeet ja rivit suodattamista varten. Yksi taulu, joka suodattamista varten täytyy luoda, on aikamääreet, jossa on eritelty kuvaajassa käytettävät aikavälit.

Dynaamisten kuvaajien luomisessa käytetään usein suodattimia ja osittajia, joiden avulla visualisointi on helpompi järjestää pienempiin osiin. Osittajien avulla on mahdollista näyttää vain tietyt tiedot, kuten esimerkiksi yhden parkkihallin keskiarvo varatuista paikoista helmikuun viidentenä päivänä. Jos Power BI:hin tuodaan tiedot tuo-metodilla, niin ohjelma osaa itse muodostaa aikaleimasta hierarkian, jota voidaan hyödyntää osittajassa. Kun käytössä on suora kysely, hierarkia täytyy aluksi muodostaa manuaalisesti aikaleimoista, jotta sitä voidaan hyödyntää myöhemmissä vaiheissa.

Power BI:n avulla ei voi kunnolla tehdä aikamääreiden mukaan vaihtuvaa kuvaajaa, jos kuvaajien tiedot tuodaan useammasta taulusta. Ongelman ratkaisemiseksi päätettiin luoda uusi taulukko, jossa on kaikki raportointityökalun tarvitsemat tiedot. Taulukossa on sarakkeet päivämäärälle, liikennevalojen havainnoille, WiFi-laitelaskurien havainnoille ja parkkihallien

varatuille paikoille. Koska taulukossa on vain yksi sarake päivämäärälle ja muut taulukon sarakkeet käyttävät sitä tietojen hakemiseen toisista taulukoista, se ei hankaloita osittajan käyttämistä. Ongelman olisi voinut ratkaista myös luomalla suhteen taulukoiden välille päivämäärien avulla. Taulukkojen välisiä suhteita käytettäessä menetetään aikaleimasta muodostuneen hierarkian edut ellei aikaleimat ole molemmissa taulukoissa täysin samat.

Raportointityökalua varten luodussa näkymässä on pylväsdiagrammi, jossa pylväät on luokiteltu viikkojen mukaan ja jokaisessa ryhmässä on oma pylväänsä liikennevaloille, WiFi-laitelaskureille ja parkkihalleille. Pylväissä käytetään eri värejä, jotka ovat myös eri tummuusasteisia, jotta niiden erottaminen toisistaan olisi mahdollisimman helppoa. Parkkihalleista esitetään tiedot parkkitunteina, koska saatavilla olevassa datassa on vain tiedot varatuista ja vapaista paikoista sekä parkkihallin kokonaiskapasiteetti. Raportoinnin kannalta ideaalisempaa olisi saada lukumäärät parkkihalliin tulleista ja sieltä lähteneistä autoista, mutta nykyisillä sensoreilla ja sopimuksilla se ei ole mahdollista. Kuviossa on 17 on esitetty alustava kuva uudesta raportointityökalusta. Valmiissa versiossa pylväsdiagrammi on järjestetty viikkojen eikä mittausten mukaan. Myös kuvaajissa esiintyvä selite parkkihallit on muutettu muotoon parkkitunti, joka kuvaa paremmin todellisuutta mittauksien perusteella.



Kuvio 17. Alustava kuva uudesta raportointityökalusta.

6.1.3 Raportoinnin julkaiseminen

Valmis raportointityökalu voidaan julkaista suoraan Power BI:stä käsin hyödyntäen sen sisäänrakennettua julkaise-metodia. Ennen julkaisemista tulee kuitenkin varmistaa muutama asia tiedostosta, joka sisältää raportin ja itse raportista. Julkaisemiseen on olemassa kaksi erilaista vaihtoehtoa, ensimmäisessä julkaistaan vain raportti ja toisessa julkaistaan koko tiedosto, joka voi sisältää useita raportteja ("Tietojoukkojen ja raporttien julkaiseminen Power BI Desktopista" 2020). Molemmissa tapauksissa tulee tarkistaa näkyvät tiedot ja se, miten loppukäyttäjä voi vaikuttaa raporttiin. Kaikki taulukot ja mittarit kannattaa piilottaa sekä estää loppukäyttäjiä näkemästä niitä, koska niiden avulla voi paljastua tietoa tietokannan rakenteesta sekä mahdollisista taustajärjestelmistä. Jos raportissa haetaan tietoja useasta eri lähteestä, kannattaa jokainen SQL-kysely tarkistaa huolella. Koska kyselyt on laitettu samaan tiedostoon, voivat virheet johtaa kyselyiden sekoittumiseen ja väärien tietojen hakemiseen. Parhaassa tapauksessa virheellinen kysely palauttaa vain virheilmoituksen tai tyhjän tietueen, mutta pahimmassa tapauksessa palautettava tieto voi paljastaa mahdollisia heikkouksia tietokannasta. Kyselyiden huolellinen muotoilu ja tietojen hakeminen vain yhdestä tietokannasta on yksinkertainen tapa pitää huolta raportin tietoturvallisuudesta.

Raportissa olevien visualisointien suhteen on myös mahdollista rajata tietojen käyttämistä. Luodusta kuvaajasta on mahdollista viedä tietoja muualle CSV-tiedostona ellei tätä vaihtoehtoa ole poistettu loppukäyttäjiltä. Avoimen datan kohdalla vaihtoehdon voi hyvin jättää päälle, koska nimensä mukaisesta avoimen datan käyttäminen on kaikille sallittua. Kuvaajan avulla voidaan saada vain kyseisessä kuvaajassa esiintyviä tietoja, joten niiden avulla ei kaikkea dataa voi saada. Joissakin tapauksissa kuvaajissa oleva tieto saattaa sisältää päivämääriä, jotka on piilotettu päivämääristä koostuvan hierarkian taakse. Loppukäyttäjällä on usein kuitenkin mahdollisuus käyttää kuvaajien ominaisuutta, jossa hierarkiassa voidaan liikkua tasolta toiselle. Tällaisissa tilanteissa suuremman tason tiedot voivat olla julkisia, mutta päiväkohtaisia tietoja ei haluta luovuttaa eteenpäin esimerkiksi kilpailuetuun vedoten. Tällaisissa tilanteissa kannattaa muodostaa kuvaaja joko suoraan halutulle tasolle tai kieltää loppukäyttäjältä tietojen vieminen raportista. Yksi mahdollinen vaihtoehto on myös sallia tasoilla liikkuminen vain tiettyyn pisteeseen asti, jolloin loppukäyttäjä ei näe kaikkia tietoja.

Raportti voidaan julkaista uudestaan suhteellisen helposti ja yleensä tällainen tilanne tulee

eteen kun raporttia halutaan täydentää uusilla kuvaajilla. Muutamia asioita kannattaa huomioida ennen uudelleenjulkaisua, jotta vältytään suuremmilta ongelmilta. Tiedostojen ja raporttien nimeämisen kanssa kannattaa olla erityisen tarkka, koska julkaiseminen epäonnistuu jos samannimisiä julkaistuja tiedostoja on kaksi tai useampi. Kaikki muutokset, jotka on tehty uuteen julkaistavaan versioon tulevat voimaan, mikä voi johtaa joidenkin kuvaajien rikkoutumiseen. Tällainen tilanne johtuu yleensä sarakkeiden ja mittarien poistamisesta tai muokkaamisesta, joka aiheuttaa visualisoinnin ja tietojen välisten reittien muuttumista. Yleensä raportit on asetettu päivittymään tietyllä aikataululla, joka pysyy automaattisesti samana myös uudelleenjulkaisun jälkeen. Poikkeuksena tähän on sellaiset tapaukset, joissa raporttiin on lisätty uusia tietolähteitä. Näissä tapauksissa uudelleenjulkaisun jälkeen tulee raportista kirjautua tietokantoihin uudestaan ennen seuraavaa ajastettua päivitystä. Mikäli tietokantoihin kirjautuminen unohtuu, mahdollisesti vain osa tiedoista voidaan päivittää ja loppujen kohdalla vastaan tulee virheviestejä, jotka estävät raportoinnin kunnollisen näkymän. Tämä johtaa koko raportin rikkoutumiseen, joka ilmenee sillä, että kaikki kuvaajat avautuessaan näyttävät vain virhettä (“Tietojoukkojen ja raporttien julkaiseminen Power BI Desktopista” 2020).

JKL-Openin uusi raportointityökalu tullaan julkaisemaan Power BI:n avulla kevään tai kesän 2020 aikana. Raportointityökalun aikataulu tietojen päivittymisen osalta tullaan laatimaan siten, että tiedot päivittyvät kerran tai kaksi päivässä. Päivitys pyritään tekemään sellaisina aikoina, jolloin se aiheuttaa mahdollisimman vähän haittaa loppukäyttäjille. Alustavasti loppukäyttäjien sallitaan viedä kuvaajien tietoja CSV-tiedostoon, koska käytetty data on avointa kaikille. Tietojen vieminen voidaan helposti estää tulevaisuudessa muuttamalla raportin asetuksia ja julkaisemalla se uudestaan. Raportointityökalu tullaan hyvin todennäköisesti julkaisemaan uudestaan siinä vaiheessa, kun linja-autojen käyttäjämäärien vieminen tietokantaan on automatisoitu. Tämä mahdollistaisi näiden tietojen sisällyttämisen raporttiin, joka antaisi paljon kattavamman kuvan Jyväskylän alueen liikennemääristä ja siitä, kuinka käyttäjämäärät vaihtelevat eri kulkuneuvojen välillä.

6.2 Raportointityökalun päivittäminen

Uuden raportointityökalun julkaisemisen jälkeen JKL-Open-sivusto vanha raportointityökalu otetaan pois käytöstä ja korvataan uudella. Korvaus tullaan toteuttamaan joko integroimalla koko työkalu sivustolle hyödyntäen edelliselle raportille varattua tilaa tai korvaamalla nykyisen linkin osoite uudella. Lopullinen päätös riippuu pitkälti siitä, kuinka raportoinnin julkaiseminen sujuu ja onko julkaistu työkalu mahdollista siirtää pois Power BI:n ympäristöstä. Mikäli raportointityökalu julkaistaan siten, että sen käyttäminen on mahdollista vain Power BI:n ympäristössä, tullaan raportointityökalun siirtäminen tekemään linkityksen avulla. Työkalun päivittämisen yhteydessä tulee testata sen toimivuus eri selaimilla eri käyttöjärjestelmissä, koska Power BI:n työpöytäversio toimii vain tietokoneissa, joissa käyttöjärjestelmänä on Windows. Raportointityökalun tulisi toimia myös mobiililaitteilla, mutta ensisijaisesti sen on toimittava tietokoneilla, joissa voi olla muu käyttöjärjestelmä kuin Windows.

Tutkielmassa kehitetty raportointityökalu on alun perin suunniteltu ja tarkoitettu korvaamaan JKL-Open-sivuston vanha raportointityökalu. Tämä ei kuitenkaan estä sitä, että tulevaisuudessa sama raportointityökalu voitaisiin integroida osaksi useampaa sivustoa. Tässä tilanteessa integrointi tulee tehdä linkkien avulla ja kiinnittää erityistä huomiota tilanteisiin, joissa raportointityökalu tulee uudelleenjulkaistua Power BI:n avulla. Näissä tilanteissa on mahdollista, että raportin sijainti ja osoite vaihtuvat, joten linkki tulee vaihtaa sivustoille, joihin raportointityökalu on integroitu.

7 Yhteenveto

Pro gradu -tutkielman teoriaosassa tutkittiin erilaisia sensoreita, joita voidaan käyttää jalankulkijoiden, pyöräilijöiden ja autojen havaitsemiseen. Tämän lisäksi teoriaosassa perehdyttiin sensoreilta kerätyn datan esittämiseen ja analysointiin. Tutkielman konstruktiivisessa osassa rakennettiin Power BI:n avulla uusi raportointityökalu, jonka avulla kerättyä dataa voidaan esittää helpommin ymmärrettävässä muodossa. Rakennetulla raportointityökalulla tullaan korvaamaan JKL-Open-sivuston nykyinen raportointityökalu, joka on todettu puutteelliseksi. Konstruktiivisessa osassa datalle tehtiin erilaisia analyyseja, joiden avulla pyrittiin tunnistamaan, mistä lähteestä kyseinen data on kerätty sekä ennustamaan tulevaisuuden liikennemääriä. Datan alkuperän tunnistamisessa käytettiin pääkomponenttianalyysia ja ennustamisessa käytettiin kolmea menetelmää SARIMA:a, TensorFlow:ta ja Kalmansuodatinta.

Pro gradu -tutkielman ensimmäinen tutkimuskysymys on “Miten avointa liikennedataa voidaan hyödyntää eri kulkuvälinetyyppien käyttäjämäärien vertailemiseen sekä kuinka tieto tulisi esittää?” jonka vastaus saadaan luvun 3 teoriasta sekä lukujen 5 ja 6 tuloksista, painottuen toteutettuun raportointityökaluun. Koska kevyestä liikenteestä ei ole olemassa valtakunnallisia tilastoja ja useimmat tiedot saadaan kyselyiden ja manuaalisten laskentojen avulla, tulee aluksi tehdä verkosto, jonka avulla saadaan useammin ja luotettavammin tietoja kevyestä liikenteestä. Yleisin tapa esittää liikenteestä kerättyjä lukemia on kulkutapajakauma, joka koostetaan tiettynä aikavälinä kerätyistä tiedoista. Kulkutapajakaumassa tulokset esitetään yleensä prosentteina ja verrataan tuloksia edelliseen kulkutapajakaumaan. Tällä tavoin voidaan esittää pitkällä aikavälillä tapahtuneet muutokset, mutta lyhyen aikavälin muutokset jäävät usein taustalle. Raportointityökalussa tarkoituksena on esittää liikenteestä sensorien avulla kerättyjä lukumääriä siten, että niitä voidaan tarkastella ja vertailla viikkojen tasolla. Selkeyden vuoksi työkalussa on kolme erilaista kuvaajaa, joista voidaan lukea eri muotojen osuudet kokonaismäärästä sekä esitetään pylväsdiagrammissa kuvaajana ja taulukossa lukuina erilaisten sensorien viikon aikana havaitsemat lukumäärät.

Tutkielman toinen tutkimus kysymys on “Miten kerättyä liikennedataa voidaan tarkentaa paremmin vastaamaan todellisuutta?” johon vastaus saadaan pääsääntöisesti luvuista 2 ja 4, mutta osittain hyödynnetään luvun 5 tuloksia. Liikennedataa voidaan tarkentaa käyttämällä tilanteisiin parhaiten sopivia sensoreita sekä asentamalla lisää sensoreita, jotta ne kattavat suuremman osan tieverkosta. Tarkennusta voidaan saada myös analysoimalla dataa ja selvittämällä, minkälaista liikennettä normaalitilanteessa on, jotta virhetilanteet saataisiin paremmin kiinni. Esimerkiksi pääkomponenttijaon ja Kalman-suodattimen avulla voidaan selvittää datan ominaisuuksia sekä suodattaa mahdollisia taustamelun aiheuttamia tuloksia.

Parhaan sensorin valinta ei ole yksinkertaista, koska jokaisella sensorilla on omat vahvuutensa ja heikkoutensa, eikä yksikään sensori ole täysin ylivoimainen muihin verrattuna. Jalankulkijoiden havainnoimiseen kannattaa käyttää joko videokameroita ja konenäköä tai WiFi-laitelaskureita ja infrapunasensoreita. Valinta riippuu pitkälti siitä, kuinka paljon rahaa on mahdollista investoida sensoreihin ja kuinka yksityisyys tulee huomioida. Polkupyöräilijöiden havainnointiin kannattaa käyttää induktiosilmukoita, kunhan ne asennetaan omalla väylälle eikä lähistöllä ole voimalinjoja. Väliaikaiseen mittaukseen voidaan hyödyntää pneumaattisia putkia, mutta niitä ei suositella pitkäaikaiseen käyttöön. Autojen havainnointiin käytetään usein induktiosilmukoita ja ne vaikuttavat parhaalta vaihtoehdolta. Induktiosilmukat kestävät pitkään ja ne ovat yleensä suuressa osassa liikennevalojen automaattista ohjausta.

Tutkielmassa tehtiin myös ennustuksia tulevasta liikenteestä SARIMA:lla, TensorFlow:lla ja Kalman-suodattimella. Kalman-suodattimella tehdyt ennustukset eivät tuota kuvaajia, joista voisi suoraan katsoa tulevia liikennemääriä vaan pyritään ennustamaan tulevia tiloja. Näitä tiloja voidaan hyödyntää myöhemmin muissa menetelmissä ja niiden pääsääntöinen tarkoitus on poistaa vääristymät havainnoista. TensorFlow:n ja SARIMA:n avulla tehdyt ennustukset onnistuivat kohtalaisesti ennustamaan tulevia liikennemääriä, mutta datan vähyyden takia tulokset eivät ole täysin luotettavia. Tulosten luotettavuuteen vaikuttaa myös koronaviruksen aiheuttamat rajoitukset liikenteessä, jonka takia ennusteet eivät vastaa todellisuutta.

Liikennedatan analysointia ja siitä ennustamista tulisi tutkia jatkossa, kunhan dataa on kerätty huomattavasti enemmän. Uusien analysointien perusteella voidaan tarkemmin selvittää ongelmakohtia datassa ja itse sensoreissa, jotta tulevaisuudessa saataisiin luotettavampaa da-

taa. Kun itse datan luotettavuus on korkeampi ja sen määrä suurempi, pystytään tekemään sekä pidempiaikaisia että luotettavampia ennustuksia tulevasta liikenteestä. Liikenteen ennustamisesta voidaan tulevaisuudessa lisätä oma kuvaaja raportointityökaluun, jonka avulla on mahdollista nähdä alusta arvio tulevasta liikenteestä.

Lähteet

Ali, S. Sheik Mohammed, Boby George ja Lelitha Vanajaksi. 2011. “A Simple Multiple Loop Sensor Configuration for Vehicle Detection in an Undisciplined Traffic”. Teoksessa *International Conference on Sensing Technology,* 644–649.

———. 2013. “An Efficient Multi-Loop Sensor Configuration Applicable for Undisciplined Traffic”. *IEEE Transactions on Intelligent Transportation Systems* 14 (3): 1151–1161. doi:10.1109/TITS.2013.2255038.

Bao, Xu, Haijian Li, Dangwei Xu, Limin Jia, Bin Ran ja Jian Rong. 2016. “Traffic Vehicle Counting in Jam Flow Conditions Using Low-Cost and Energy Efficient Wireless Magnetic Sensors”. *Sensors* 16 (11): 1868–1882. doi:10.3390/s16111868.

Behrendt, Ramona. 2016. “Monitoring Radar for Road Map Calculation”. Teoksessa *International Radar Symposium,* 1–4.

Brehar, Raluca, Christian Vancea, Tiberius Marita, Ion Goisan ja Sergiu Nedevschi. 2015. “Pedestrian Detection in the Context of Multiple-Sensor Data Alignment for Far-Infrared and Stereo Vision Sensors”. Teoksessa *IEEE International Conference on Intelligent Computer Communication and Processing,* 385–392.

Dehghan-Banadaki, Ali, Taufik Taufik ja Ali Feliachi. 2018. “Big Data Analysis in a Day-Ahead Electricity Price Forecasting Using TensorFlow in Restructed Power Systems”. Teoksessa *International Conference on Computational Science and Computational Intelligence,* 1065–1069.

Dodier, Robert H., Gregor P. Henze, Dale Tiller ja Xin Guo. 2005. “Building Occupancy Detection Through Sensor Belief Networks”. *Energy and Buildings* 38 (9): 1033–1043. doi:10.1016/j.enbuild.2005.12.001.

Donovan, Brian, Yanning Li, Raphael Stern, Jiming Jiang, Christian Claudel ja Daniel Work. 2015. “Poster Abstract: Vehicle Detection and Speed Estimation with PIR Sensors”. Teoksessa *International Conference on Information Processing in Sensor Networks,* 142–153.

Dunne, Stephen, ja Bidisha Ghosh. 2013. “Weather Adaptive Traffic Prediction Using Neurowavelet Models”. *IEEE Transactions on Intelligent Transportation* 14 (1): 370–379. doi:10.1109/TITS.2012.2225049.

Farooq, Bilal, Alexandra Beaulieu, Marwan Ragab ja Viel Dang Ba. 2015. “Ubiquitous Monitoring of Pedestrian Dynamics Exploring wireless Ad Hoc Network of Multi-sensir Technologies”. Teoksessa *IEEE-Sensors.*

Fathy, M., ja M. Y. Siyad. 1998. “A Window-Based Image Processing Technique for Quantitative and Qualitative Analysis of Road Traffic Parameters”. *IEEE Transactions on Vehicular Technology* 47 (4): 1342–1349. doi:10.1109/25.728525.

Fisher, Carl, Kavitha Muthukrishnan, Mike Hazas ja Hans Gellersen. 2008. “Ultrasound-Aided Pedestrian Dead Reckoning for Indoor Navigation”. Teoksessa *ACM Interna Workshop on Mobile Entity Localization and Tracking in GPS-less Environments,* 31–36.

Franz, Stefan, Roland Schweiger, Otto Löhlein ja Kristian Kroschel. 2010. “Analysis and Assessment of Far Infrared Sensor Performance Parameters and Their Impact on Pedestrian Detection”. Teoksessa *IEEE Conference on Intelligent Transportation Systems,* 119–124.

Fuente, Rodrigo De la, Ignacio Erazo ja Raymond L. Smith. 2018. “Enabling Intelligetn Processes in Simulation Utilizing the TensorFlow Deep Learning Resources”. Teoksessa *Winter Simulation Conference,* 1108–1119.

Hao, Qi, David J. Brady, Bob D. Guenther, John B. Burchett, Mohari Shankar ja Steve Feller. 2006. “Human Tracking with Wireless Distributed Pyroelectric Sensors”. *IEEE Sensors* 6 (6): 1683–1696. doi:10.11109/JSEN.2006.884562.

Hao, Qi, Fei Hu ja Jiang Lu. 2010. “Distributed Multiple Human Tracking with Wireless Binary Pyroelectric Infrared (PIR) Sensor Networks”. Teoksessa *Sensors,* 946–950.

Hao, Qi, Fei Hu ja Yang Xiao. 2009. “IEEE Systems Journal”. *Multiple Human Tracking and Indentification with Wireless Distributed Pyroelectric Sensor Systems* 3 (4): 428–439. doi:10.1109/JSENS.2009.2035734.

Hinsbergen, Chris P. I. J. van, Thomas Schreiter, Frank S. Zuurbier, J. W. C. van Lint ja Henrik J. van Zuylen. 2012. “Localized Extended Kalman Filter for Scalable Real-Time Traffic State Estimation”. *IEEE Transactions on Intelligent Transportation Systems* 13 (1): 385–394. doi:10.1109/TITS.2011.2175728.

Ho, Tan-Jan, ja Meng-Ju Ching. 2016. “An Approach to Traffic Flow Detection Improvements of Non-Contact Microwave Radar Detectors”. Teoksessa *International Conference on Applied System Innovation,* 1–4.

Hyde-Wright, Alexander, Brian Graham ja Krista Nordback. 2014. “Counting Bicyclists with Pneumatic Tube Counters on Shared Roadways”. *ITE Journal* 84 (2): 32–37. https://researchgate.net/publication/297828311.

Immonen, Anne. 2017. “Quality in Open Data Based Digital Service Ecosystem”. Tohtorinväitöskirja, University of Oulu.

Jalankulku ja pyöraily 2015. 2015. Tilasto. Jyväskylä liikennejärjestelyt.

Jansen, Stephan, Dennis Höting, Jens Runge, Thomas Brinkhoff, Daniela Nicklas ja Jürgen Sauer. 2014. “9 Million Bicycles? Extending Induction Loops with Bluetooth Sensing”. Teoksessa *International Conference on Mobile Data Management,* 113–116.

Kalenoja, Hanna, ja Kari Mäkelä. 2001. *Henkilöliikennesuoritteet: taustatietoja ja suosituksia määritelmistä, tilastoinnista ja laskentatavoista.* Selvitys 26. Liikenne- ja viestintäministeriö.

“Kari Lukka: Konstruktiivinen tutkimusote”. 2014. Viitattu 18. marraskuuta 2019. https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/.

Kura, Satomi, Hirozumi Yamaguchi ja Yoh Shiraishi. 2018. “Low-Cost Pedestrian Counter Using Wi-Fi APs for Smart Building Applications”. Teoksessa *International Computer Software and Application Conference,* 640–645.

Leduc, Guillaume. 2008. *Road Traffic Data: Collection Methods and Applications.* Tekninen raportti. European Comission Joint Research Center.

Lesani, Asad, ja Luis Miranda-Moreno. 2018. “Development adn Testing of a Real-Time WiFi-Bluetooth System for Pedestrian Network Monitoring, Classification and Data Extrapolation”. *IEEE Transactions on Intelligent Transportation Systems* 20 (4): 1484–1496. doi:10.1109/TITS.2018.2854895.

Li, Yilong, Deng Li, Yun Cheng, Guo Liu, Jianwei Niu ja Lu Su. 2016. “Abstract: A Novel Human Tracking and Localization System Based on Pyroelectric Infrared Sensors”. Teoksessa *International Conference on Information Processing in Sensor Networks,* 3–4.

Lie, Mengchen, Cleas Sandels, Kun Zhu ja Lars Nordström. 2013. “A Seasonal ARIMA Model with Exogenous Variables for Elspot Electricity Prices in Sweden”. Teoksessa *International Conference on the European Energy Market,* 1–4.

Litmanen, Juha, Kimmo Saastamoinen, Minna Kemppinen, Hanna Horppila, Jutta-Leea Kärki ja Valtteri Rantala. 2006. *Kevyen liikenteen laskentatietojen hallinta- ja tietopalveluiden määrittely.* Raportti 7. Liikenne- ja viestintäministeriö.

Liu, Hongbo, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen ja Fan Ye. 2012. “Push the Limit of WiFi Based Localization for Smartphones”. Teoksessa *Annual International Conference on Mobile Computing and Networking,* 305–316.

LV, Yisheng, Yanjie Duan, Wenwen Kang, Zhengxi Li ja Fei-Yue Wang. 2014. “Traffic Flow with Big Data: A Deep Learning Approach”. *IEEE Transactions on Intelligent Transportation Systems* 16 (2): 865–873. doi:10.1109/TITS.2014.2345663.

Maurin, B., O. Masoud ja N. P. Papanikolopoulos. 2005. “Tracking All Traffic: Computer Vision Algorithms for Monitoring Vehicles, Individuals, and Crowds”. *IEEE Robotics and Automation Magazine* 12 (1): 29–36. doi:10.1109/MRA.2005.1411416.

Meta, Soner, ja Muhammed G. Cindikici. 2010. “Vehicle-Classification for Single-Loop Inductive Detector”. *IEEE Transactions on Vehicular Technology* 59 (6): 2795–2805. doi:10.1109/TVT.2010.2049756.

Minomi, Shohei, Hiroshi Yamamoto, Katsuichi Nakamura ja Katsuyuki Yamazaki. 2012. “A Study of Pedestrian Observation System with Ultrasonic Distance Sensor”. Teoksessa *International Conference on Advanced Communication Technology,* 251–256.

Muffert, MAximillian, David Pfeiffer ja Uwe Franke. 2013. “A Stereo Vision Based Object Tracking Approach at Roundabouts”. *IEEE Intelligent Transportation Systems Magazine* 5 (2): 22–32. doi:10.1109/MITS.2013.2244934.

Narayana, Sujay, R. Venkatesha Prasad, T. V. Prabhakar, Sripad Kowshik, Vijay S. Rao ja Madhuri Sheethala Lyer. 2015. “PIR Sensors: Characterization and Novel Localization Technique”. Teoksessa *International Conference on Information Processing in Sensor Networks,* 142–153.

Nordback, Krista, Daniel Piatkowski, Bruce Janson, W.E. Marshall, Kevin J. Krizek ja eborah S. Main. 2011. “ITE Journal: Institute of Transportation Engineers”. *Using Inductive Loops to Count Bicycles in Mixed Traffic* 2 (1): 35–56. https://www.researchgate.net/publication/285851002.

Noureen, Subrina, Sharif Atique, Vishwajit Roy ja Stephen Bayne. 2019. “Analysis and Application of Seasonal ARIMA Model in Energy Demand Forecasting: A Case Study of Small Scale Agricultural Load”. Teoksessa *IEEE International Midwest Symposium on Circuits and Systems,* 521–524.

Oliveira, Luis, Daniel Schneider, Jano De Souza ja Weiming Shen. 2019. “IEEE Access”. *Mobile Device Detection Through WiFi Probe Request Analysis* 7:98579–98588. doi:10.1109/ACCESS.2019.2925406.

Qiuying, Wang, Guo Zheng, Zhang Minghui, Cui Xufei, Wu Hui ja Jia Li. 2018. “Research on Pedestrian Location Based on Dual MIMU/Magnetometer/Ultrasonic Module”. Teoksessa *IEEE/ION Positioon, Localization and Navigation Symposium,* 565–570.

Qu, Li, Li Li, Yi Zhang ja Jianming Hu. 2009. “PPCA-Based Missi Data Imputation for Traffic Flow Volume: A Systematic Approach”. *IEEE Transactions on Intelligent Trasportation Systems* 10 (3): 512–522. doi:10.1109/TITS.2009.2026312.

Raman, Rahul, Pankaj Kumar Sa ja Bakshi Majhi Banshidhar. 2016. “Direction Estimation dor Pedestrian Monitoring System in Smart Cities: An HMM Based Approach”. *IEEE Access* 4 (1): 5788–5808. doi:10.1109/ACCESS.2016.2608844.

Sandrasegaran, Kumbesan, Xiaoying Kong, Zhu Xinnig, Jingbin Zhao, Bin Hu, Cheng-Chung Lin ja Zhuliang Xu. 2013. “FPedestrian Monitoring System Using Wi-Fi Technology and RSSI Based Localization”. *International Journal of Wireless and Mobile Networks* 5 (4): 17–34. doi:10.5121/ijwmn.2013.5402.

Schasberger, Michele G., Jessica Raczkowski, Lawrence Newman ja Michael F. Polgar. 2012. “Using a Bicycle-Pedestrian Count ot Assess Active Living in Downtown Wilkes-Barre”. *American Journal of Preventive Medicine* 43 (544): 399–402. doi:10.1016/j.amepre.2012.06.029.

Schauer, Lorenz, Martin Werner ja Philipp Marcus. 2014. “Estimating Crowd Densities and Pedestrian Flow Using Wi-Fi and Bluetooth”. Teoksessa *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services,* 171–177.

Shih, Oliver, ja Anthony Rowe. 2015. “Occypancy Estimation Using Ultrasonic Chirps”. Teoksessa *ACM/IEEE Sixth International Conference on Cyber-Physical Systems,* 149–158.

Shirazi, Mohammed Shokrolah, ja Brendan Tran Morris. 2015. “Vision-Based Turning Movement Monitoring: Count, Speed and Waiting Time Estimation”. *IEEE Intelligent Transportation Systems Magazine* 8 (1): 23–34. doi:10.1109/MITS.2015.2477474.

Shlayan, Neveen, Abdullah Kurkcu ja Kaan Ozbay. 2016. “Exploring Pedestrian Bluetooth and WiFi Detection at Public Transportation Terminals”. Teoksessa *International Conference on Intelligent Transportation Systems,* 230–234.

Stutz, C., ja T. A. Runkler. 2002. “Classification and Prediction of Road Traffic Using Application-Spicific Fuzzy Clustering”. *IEEE Transactions on Fuzzy Systems* 10 (3): 297–308. doi:10.1109/TFUZZ.2002.1006433.

Syrjärinne, Paula. 2016. “Urban Traffic Analysiswith Bus Location Data”. Tohtorinväitöskirja, University of Tampere.

Tan, Huachun, Guangdong Feng, Jianshuai Feng, Wuhong Wang, Yu-Jin Zhang ja Feng Li. 2010. “A Tensor-Based Method for Missing Traffic Data Completion”. *Transportation Research Part C* 28 (1): 15–27. doi:10.1016/j.trc.2012.12.007.

"Tietojoukkojen ja raporttien julkaiseminen Power BI Desktopista". 2020. Viitattu 24. huhtikuuta 2020. https://docs.microsoft.com/fi-fi/power-bi/desktop-upload-desktop-files.

"TUBES Mobile bike counter, designed with short-term bike volume studies in mind." 2019. Viitattu 24. lokakuuta 2019. https://www.eco-compteur.com/en/produits/tubes-en/tubes-2/.

"Urban ZELT: The world's most trusted permanent bike counter, designed for urban cycling". 2019. Viitattu 5. joulukuuta 2019. https://www.eco-compteur.com/en/produits/zelt-range/urban-zelt/.

Wagner-Muns, Isaac Michael, Ivan G. Guardiola, V. A. Samaranayke ja Wasim Ishad Kayani. 2018. "A Functional Data Analysis Approach to Traffic Volume Forecasting". *IEEE Transactions on Intelligent Transportation Systems* 19 (3): 878–888. doi:10.1109/TITS.2017.2706143.

Wang, Shangboo, ja Guoqiang Mao. 2018. "IEEE Transactions on Intelligent Transportation Systems". *Missing Data Estimation for Traffic Volume by Searching an Optimum Closed Cut in Urban Networks* 29 (1): 75–86. doi:10.1109/TITS.2018.2801808.

Wilkie, David, Jason Sewall ja Ming Lin. 2013. "Flow Reconstruction for Data-Driven Traffic Animation". *ACM Transactions on Graphics* 32 (4): 1–10. doi:10.1145/2461912.2462021.

Wong, Chong, Bim Ran, Han Yang, Jian Zhang ja Xu Qu. 2018. "A Novel Aprroach to Estimate Freeway Traffic State: Parallel Computing and Improved Kalman Filter". *IEEE Intelligent Transportation Systems Magazine* 10 (2): 180–193. doi:10.1109/MITS.2018.2806627.

Xu, Yanyan, Qing-Jie Kong, Reinhard Kiette ja Yuncai Liu. 2014. "Accurate and Interpretable Bayesian MARS for Traffic Flow Prediction". *IEEE Transactions on Intelligent Transportation Systems* 15 (6): 2457–2469. doi:10.1109/TITS.2014.2315794.

Yuanhua, Jia, ja Xing Enhui. 2011. “The Application of Traffic Flow Detection Technology on Characteristics of Freeway Traffic Flow”. Teoksessa *International Conference on Remote Sensing, Environment and Transportation,* 838–840.

Zhou, Teng, Dazhi Jiang, Zhizhe Lin, Guoqiang Han, Xuemia Xu ja Jing Qin. 2019. “Hybrid Dual Kalman Filtering Model for Short-Term Traffic Flow Forecasting”. *IET Intelligent Traansport Systems* 13 (6): 1023–1032. doi:10.1049/iet-its.2018.5385.

Liitteet

A TensorFlow:lla ennustamisen koodi

```
# Modeled after TensorFlow time series forecasting tutorial
# that is on tensorflow.org.

import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import tensorflow as tf



mpl.rcParams['figure.figsize'] = (8, 6)
mpl.rcParams['axes.grid'] = False
file_to_open = "koko_data.csv"
df = pd.read_csv(file_to_open)

def univariate_data(dataset, start_index, end_index,
                    history_size, target_size):
    data = []
    labels = []

    start_index = start_index + history_size
    if end_index is None:
        end_index = len(dataset) - target_size

    for i in range(start_index, end_index):
```

```
        indices = range(i-history_size, i)
        data.append(np.reshape(dataset[indices],
                    (history_size, 1)))
        labels.append(dataset[i+target_size])
        return np.array(data), np.array(labels)


TRAIN_SPLIT = 8100  # This gives the best result
tf.random.set_seed(13)

uni_data = df['Value']
uni_data.index = df['date']
uni_data.plot(subplots=True)
uni_data = uni_data.values
uni_train_mean = uni_data[:TRAIN_SPLIT].mean()
uni_train_std = uni_data[:TRAIN_SPLIT].std()
uni_data = (uni_data-uni_train_mean)/uni_train_std


# On a normal laptop don't use large number like 2000
# because it takes time and GPU power
univariate_past_history = 20
univariate_future_target = 0

# The start index of training data should be 0 or
# the index that starts the first whole day
x_train_uni, y_train_uni = univariate_data(
    uni_data, 35, TRAIN_SPLIT, univariate_past_history,
    univariate_future_target)

# Index 10380 is a good place to stop,
```

```
# indexes after that have been affected by Corona virus
x_val_uni, y_val_uni = univariate_data(
    uni_data, TRAIN_SPLIT, 10380, univariate_past_history,
    univariate_future_target)

print('Single window of past history')
print(x_train_uni[0])
print('\n Target value to predict')
print(y_train_uni[0])


def create_time_steps(length):
    return list(range(-length, 0))


def show_plot(plot_data, delta, title):
    labels = ['History', 'True Future', 'Model Prediction']
    marker = ['.-', 'rx', 'go']
    time_steps = create_time_steps(plot_data[0].shape[0])
    if delta:
        future = delta
    else:
        future = 0

    plt.title(title)
    for i, x in enumerate(plot_data):
        if i:
            plt.plot(future, plot_data[i], marker[i],
                     markersize=10, label=labels[i])
        else:
            plt.plot(time_steps, plot_data[i].flatten(
```

```
            ), marker[i], label=labels[i])
    plt.legend()
    plt.xlim([time_steps[0], (future+5)*2])
    plt.xlabel('Time-Step')
    return plt


show_plot([x_train_uni[0], y_train_uni[0]], 0,
          'Sample Example')


def baseline(history):
    return np.mean(history)


show_plot([x_train_uni[0], y_train_uni[0], baseline(
    x_train_uni[0])], 0, 'Baseline Prediction')


BATCH_SIZE = 256
BUFFER_SIZE = 10000

train_univariate = tf.data.Dataset.from_tensor_slices(
    (x_train_uni, y_train_uni))
train_univariate = train_univariate.cache().shuffle(
    BUFFER_SIZE).batch(BATCH_SIZE).repeat()

val_univariate = tf.data.Dataset.from_tensor_slices(
  (x_val_uni, y_val_uni))
val_univariate = val_univariate.batch(BATCH_SIZE).repeat()
```

```
simple_lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(8,
      input_shape=x_train_uni.shape[-2:]),
    tf.keras.layers.Dense(1)
])
simple_lstm_model.compile(optimizer='adam', loss='mae')

for x, y in val_univariate.take(1):
    print(simple_lstm_model.predict(x).shape)

EVALUATION_INTERVAL = 200
EPOCHS = 10

simple_lstm_model.fit(train_univariate, epochs=EPOCHS,
steps_per_epoch=EVALUATION_INTERVAL,
                      validation_data=val_univariate,
                      validation_steps=50)

for x, y in val_univariate.take(3):
    plot = show_plot([x[0].numpy(), y[0].numpy(),
                      simple_lstm_model.predict(x)[0]], 0,
                      'Simple LSTM model')
    plot.show()
```

B PCA:n koodi

```
  # Modeled after tutorial on https://towardsdatascience.com.

  import pandas as pd
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

file = 'parkkihallit.data'

df = pd.read_csv(file, names=['All Places', 'Reserved',
                              'Free', 'Target'])

features = ['All Places', 'Reserved', 'Free']

# Separating out the features
x = df.loc[:, features].values
y = df.loc[:, ['Target']].values

# Standardizing the features
x = StandardScaler().fit_transform(x)

pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data=principalComponents,
                  columns=['principal component 1',
                           'principal component 2'])
finalDf = pd.concat([principalDf, df[['Target']]],
                    axis=1)

fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(1, 1, 1)
ax.set_xlabel('Principal Component 1', fontsize=15)
ax.set_ylabel('Principal Component 2', fontsize=15)
ax.set_title('2 component PCA', fontsize=20)
```

```
  targets = ['Asema', 'Kolmikulma', 'Sokos', 'Pergamentti',
             'Paviljonki 1', 'Paviljonki 2', 'Tori',
             'Matkakeskus', 'Cygnaeus']
  colors = ['red', 'yellow', 'green', 'blue',
            'pink', 'brown', 'purple', 'orange', 'black']

  for target, color in zip(targets, colors):
      indicesToKeep = finalDf['Target'] == target
      ax.scatter(finalDf.loc[indicesToKeep,
                 'principal component 1'],
                 finalDf.loc[indicesToKeep,
                 'principal component 2'],
                 c=color, s=50)

  ax.legend(targets)
  ax.grid()

  plt.show()
```

C Kalman-suodattimen koodi

```
  # Modeled after example on https://towardsdatascience.com.

from math import *
import matplotlib.pyplot as plt
import numpy as np

# The values are taken from a file that has all the data.
  Sigma values are calculated from the data.
```

```
# Add correct values from data, these are just an example,
  a snippet from the actual data.
measurements = [238422., 361153., 362466., 358966., 391932.,
                220798., 273341., ..., 156696.]

motions = [49976., 122731., 1313., 3500., 32966.,
           171134., ..., 23579.]

measurement_sig = 15.
motion_sig = 13.
mu = 0.
sig = 10000.


# Default Gaussian function
def f(mu, sigma2, x):
    coefficient = 1.0/sqrt(2.0 * pi * sigma2)
    exponential = exp(-0.5 * (x - mu) ** 2 / sigma2)
    return coefficient * exponential


# Function used to update the data
def update(mean1, var1, mean2, var2):
    new_mean = (var2 * mean1 + var1 * mean2) / (var2 + var1)
    new_var = 1 / (1 / var2 + 1 / var1)
    return [new_mean, new_var]


# The implementation of Kalman-filter and predict function
def predict(mean1, var1, mean2, var2):
    pred_mean = mean1 + mean2
```

```
    pred_var = var1 + var2
    return[pred_mean, pred_var]


for i in range(len(measurements)):
    mu, sig = update(mu, sig, measurements[i],
                     measurement_sig)
    print('Update: [{}, {}]'.format(mu, sig))
    mu, sig = predict(mu, sig, motions[i], motion_sig)
    print('Predict: [{}, {}]'.format(mu, sig))


print('\n')
print('Final result: [{}, {}]'.format(mu, sig))

# Plot the calculated figures into a visualization
mu = mu
sigma2 = sig

x_axis = np.arange(-20, 20, 0.1)
g = []
for i in x_axis:
    g.append(f(mu, sigma2, i))


plt.plot(x_axis, g)

# Plot for regular looking Gaussian function
mu = 0
sigma2 = 10000
```

```
g = []
for i in x_axis:
    g.append(f(mu, sigma2, i))

plt.plot(x_axis, g)
plt.show()
```

D SARIMA:lla ennustamisen koodi

```
  # Modeled after tutorial on https://towardsdatascience.com.

from pylab import rcParams
import statsmodels.api as sm
import pandas as pd
import warnings
import itertools
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import xlrd
from sklearn.metrics import mean_squared_error
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
mpl.rcParams['axes.labelsize'] = 14
mpl.rcParams['xtick.labelsize'] = 12
mpl.rcParams['ytick.labelsize'] = 12
mpl.rcParams['text.color'] = 'G'

df = pd.read_excel('SARIMA_data.xlsx')
y = df.set_index(['Date'])
```

```
y.plot(figsize=(19, 4))

rcParams['figure.figsize'] = 16, 9
decomposition = sm.tsa.seasonal_decompose(y,
                  model='additive')
fig = decomposition.plot()

# There are 20 weeks in the data and those are the seasons.
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 20)
                for x in list(itertools.product(p, d, q))]
print('Examples of parameter for SARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[3], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[4], seasonal_pdq[4]))

print('AIC scores for the data')

for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(
                y, order=param,
                seasonal_order=param_seasonal,
                enforce_stationarity=False,
                enforce_invertibility=False)
            results = mod.fit()
            print('ARIMA{}x{}20 - AIC:{}'.format
                 (param, param_seasonal, results.aic))
```

```
        except:
            continue

# Best option from results: ARIMA(1, 1, 1)x(0, 1, 1, 20)20

mod = sm.tsa.statespace.SARIMAX(y, order=(1, 1, 1),
         seasonal_order=(0, 1, 1, 20),
         enforece_stationarity=False,
         enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
results.plot_diagnostics(figsize=(18, 8))

pred = results.get_prediction(
    start=pd.to_datetime('2020-03-01'), dynamic=False)
pred_ci = pred.conf_int()
ax = y['2019':].plot(label='observed')
pred.predicted_mean.plot(ax=ax, label='One-step ahead',
                         alpha=.7, figsize=(14, 4))
ax.fill_between(
    pred_ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1],
    color='k', alpha=.2)
ax.set_xlabel('Date')
ax.set_ylabel('Measured Traffic')
plt.legend()

y_forecasted = pred.predicted_mean
y_truth = y['2020-03-01':]

mse = mean_squared_error(y_truth, y_forecasted)
print('The Mean Squared Error is {}'.format(round(mse, 2)))
```

```
print('The Root Mean Squared Error is {}'.format(round(
      np.sqrt(mse), 2)))

pred_uc = results.get_forecast(steps=20)
pred_ci = pred_uc.conf_int()
ax = y.plot(label='observed', figsize=(14, 4))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
ax.fill_between(
    pred_ci.index, pred_ci.iloc[:, 0], pred_ci.iloc[:, 1],
    color='k', alpha=.25)
ax.set_xlabel('Dates')
ax.set_ylabel('Traffic')
plt.legend()
plt.show()
```