Tampereen yliopisto

Juho Taavetinkangas

JOUKKOLIIKENTEEN VAIHTOYHTEYKSIEN TUNNISTAMINEN JA ANALYSOINTI

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastajat: Tarmo Lipping, Jari Turunen
Marraskuu 2022

TIIVISTELMÄ

Juho Taavetinkangas: Joukkoliikenteen vaihtoyhteyksien tunnistaminen ja analysointi
Diplomityö
Tampereen yliopisto
Johtaminen ja tietotekniikka
Marraskuu 2022

Runkolinjaston käyttöönoton myötä vaihtomatkojen määrän on ennustettu kasvavan. Moni kokee vaihdolliset bussiyhteydet vähemmän miellyttäviksi kuin suorat vaihdottomat yhteydet. Joukkoliikenteen kulkutapaosuuden kasvattamiseksi vaihtoyhteyksiin liittyvää analytiikkaa on syytä kehittää.

Tässä diplomityössä kehitettiin algoritmi, jolla voidaan tunnistaa vaihtomatkaketjut Turun seudun joukkoliikenteen matkustusdatan perusteella. Algoritmin seurauksena saatiin näkyvyys toteutuneiden vaihtomatkaketjujen sisältöön ja samankaltaiset matkaketjut voitiin yhdistää klustereihin.

Tutkimuksessa käytettiin syötedatana lippujärjestelmän tallentamaa matkakorttidataa, josta ilmenivät mm. nousutapahtuman aikaleima, käytetty pysäkki ja käytetty linja. Dataa rikastettiin yhdistämällä siihen nousuhetkellä voimassaolevaa reittidataa. Vaihtomatkaketjut muodostettiin päättelemällä poistumispysäkki seuraavasta nousupysäkistä. Lopputuloksena saadut merkkijonojen joukot klusteroitiin locality-sensitive hashing -menetelmällä niiden samankaltaisuuteen perustuen.

Työ on jaettu johdannon lisäksi neljään osaan. Kaksi ensimmäistä osaa käsittelevät pääasiassa aiheeseen liittyvää teoriaviitekehystä, joista ensimmäisessä keskitytään joukkoliikenneteemoihin ja toisessa samankaltaisuuden löytämiseen. Kolmannessa osassa kuvaillaan tutkimuksessa käytetyt menetelmät ja viimeisessä osassa menetelmän avulla saadut tulokset pohdintoineen.

Tutkimuksessa havaittiin, että selkeää tarvetta aikataulujen synkronoimiselle ei ole, koska vaihtoyhteyksien käyttö kohdistuu linjoille, joilla vuoroväli on jo nykyisellään tiheä. Vaihtomatkat jakautuvat ajallisesti pääasiassa samoin, kuin matkustustapahtumatkin. Menetelmänä locality-sensitive hashing on nopea ja tarjoaa hyvin samankaltaisen lopputuloksen kuin perinteinen iterointiin perustuva raakamenetelmä.

Avainsanat: joukkoliikenne, data mining, data-analyysi, minhash, locality-sensitive hashing

ABSTRACT

Juho Taavetinkangas: Identifying and analysis of transfer connections in public transport
Masters' Thesis
Tampere University
Degree Programme in Management and Information Technology
November 2022

With the launch of the trunk bus line network it is predicted that the amount of the trips with transfers will increase. Many people find journeys with a transfer less pleasant than direct non-transfer connections. In order to improve customer experience and to get public transport more attractive it is important to develop new data analysis methods.

In this master's thesis, an algorithm was developed to identify journey patterns containing one or more transfer connections. As a result of the algorithm, visibility was achieved to the content of calculated journey patterns. Similar journey patterns could then be grouped to clusters.

The study used travel card data recorded by the ticketing system as input data. This transaction data included the time stamp of the boarding event, the stop where the boarding took place and the line used. The data was enriched by combining route data valid at the time of the boarding. The journey patterns were formed by inferring the exit stop from the next boarding stop. The resulting sets of strings were clustered using the locality-sensitive hashing method based on their similarity.

The work is divided into four chapters in addition to the introduction. The first two chapters deal mainly with the theoretical framework related to the topic, the first focusing on public transport themes and the second on finding similarity. The third chapter describes the methods used in the study and, in the last chapter, the results obtained by the method are presented with reflections.

The study found that there is no obvious need for timetable synchronization because the use of transfers occurs at lines where a scheduled headway is already dense. Transfer connections are distributed by a time in mainly the same way as boarding events in general. Locality-sensitive hashing provides fast clustering and very similar result as the more traditional brute force method.

Keywords: public transport, data mining, data analysis, minhash, locality-sensitive hashing

ALKUSANAT

Haluan ilmaista kiitokseni professori Tarmo Lippingille sekä lehtori Jari Turuselle, jotka toimivat työn ohjaajina ja antoivat hyödyllisiä suuntaviivoja koko projektin ajan.

Kiitän myös joukkoliikennepalvelujohtaja Sirpa Kortetta mielenkiintoisen ja ammattitaitoa kartuttavan työn mahdollistamisesta.

Lopuksi haluan kiittää perhettäni kärsivällisyydestä ja joustavuudesta jo muutoinkin työntäyteisen vauva-arjen keskellä.

Liedossa, 1. marraskuuta 2022

Juho Taavetinkangas

SISÄLLYSLUETTELO

1. JOHDANTO

Joukkoliikenne on järjestetty Turun seudulla kuuden kunnan alueella seudullisen joukkoliikenneviranomaisen, Fölin, toimesta. Föli on Turun kaupungin alainen yksikkö, jota johtaa poliittisista päättäjistä koostuva seudullinen joukkoliikennelautakunta. Föli kilpailuttaa järjestämänsä liikenteen tilaaja-tuottaja -mallilla. Vuonna 2022 kilpailutettua linja-autoliikennettä harjoittaa seitsemän liikennöitsijää sekä vesibussiliikennettä kaksi liikennöitsijää. [1]

Turun seudun joukkoliikenteessä on suunniteltu runkolinjastototeutusta yli kymmenen vuoden ajan. Runkolinjaston yhtenä tavoitteena on joukkoliikenteen kilpailukyvyn parantaminen henkilöautoliikenteeseen nähden ja sen toteutuminen edellyttäisi joukkoliikenteen sujuvoittamista sekä matka-aikojen pienentämistä. Suunnitelmaan liittyvä linjastouudistus perustuu korkean palvelutason runkolinjoihin sekä niitä tukeviin täydentäviin linjoihin [2]. Mikäli toteutuksen yhteydessä keskustan kautta ajavien linjojen määrä vähenee, voi se tarkoittaa vaihtoyhteyksien käytön lisääntymistä [3].

Vaihtoyhteydet voidaan jakaa suunniteltuihin ja suunnittelemattomiin vaihtoyhteyksiin. Suunnittelemattomat vaihtoyhteydet ovat yksilön tekemiä valintoja, jotka voivat pohjautua esimerkiksi reittipalveluiden tarjoamiin ehdotuksiin. Suunnittellut vaihtoyhteydet ovat liikenteen tilaajan tai liikennöitsijän tiettyihin solmukohtiin asettamia vaihtoyhteyksiä, joiden tarkoituksena on tarjota sujuva yhteys joukkoliikenneverkon eri puolille. [4]

Sujuvat vaihtoyhteydet ovat oleellinen osa toimivaa linjastoa. Hyvin suunniteltuina ne mahdollistavat houkuttelevammat matkaketjut ja nopeammat matka-ajat ilman merkittäviä vaikutuksia kustannuksiin. Tarve aikataulujen synkronoinnille korostuu etenkin tapauksissa, joissa sekä saapuvan että lähtevän yhteyden vuoroväli on harva. [5]

Trafix Oy:n ja WSP Finland Oy:n laatimassa esitelmässä [6] todetaan vuonna 2017 vaihdollisia matkoja olevan 12 % kaikista Fölin matkoista. Niiden ennustetaan kasvavan runkolinjaston myötä jopa 18 prosenttiin toteutusmallista riippuen.

Linna on tutkinut pro gradu -tutkielmassaan [3] joukkoliikenteen käyttöön vaikuttavia tekijöitä Turun seudulla. Vaihtoyhteyksien olemassaolo vaihdottomien suorien yhteyksien sijaan rajoittaa etenkin turkulaisten työssäkäyvien ja opiskelijoiden joukkoliikenteen käyttöä. Tietyissä seutukunnissa edellä mainittu ryhmä kokee myös matka-ajan pituuden yleisesti rajoittavana muuttujana. Sujuvat vaihtoyhteydet muuallakin kuin keskustassa voisi-

vat lyhentää kokonaismatka-aikoja. Vaihtoihin liittyen rajoittavana tekijänä koetaan myös liian pitkä tai lyhyt vaihtoaika.

Runkobussilinjaston kehittämisohjelmaraportissa [2] on julkaistu vuoden 2010 asukaskyselyn tuloksia, jossa lähes 90 % vastaajista ilmoittaa sujuvat paikallisliikenteen vaihtoyhteydet erittäin tärkeäksi palvelutasotekijäksi. Myös kokonaan vaihdottomia yhteyksiä pitää erittäin tärkeänä lähes 70 % vastaajista. Matkustajanäkökulman suhteen raportissa todetaan, että vaihdon bussista toiseen on onnistuttava helposti, kun asiakas valitsee runkolinjan.

Tämän työn tavoitteena on rakentaa algoritmi vaihtomatkaketjujen generointiin ja löytää tehokas menetelmä samankaltaisten vaihtomatkaketjujen klusterointiin kysynnän tunnistamiseksi. Tutkimuskysymyksenä on, voiko samankaltaisia matkaketjuja järkevästi klusteroida siten, että tuloksia voi edelleen hyödyntää joukkoliikennesuunnittelussa. Lähtötilanteessa tiedetään ainoastaan pysäkki, josta asiakas on noussut kyytiin. Rakennettava algoritmi tarjoaa vastauksen myös siihen, voidaanko vaihtoyhteydet löytää pelkkien nousutapahtumien perusteella.

2. JOUKKOLIIKENNE JA VAIHTOYHTEYDET

Tässä luvussa käydään läpi työn joukkoliikenteeseen liittyvää käsitteistöä. Se koostuu liikennöintiin liittyvistä käsitteistä, kuten pysäkeistä, ajoneuvoista ja linjoista, sekä tuotteisiin liittyvistä käsitteistä, kuten tuotetunniste ja nousutapahtuma. Lisäksi esitellään kirjallisuudessa tunnistettuja matkakorttidatan hyödyntämistapauksia.

2.1 Pysäkit

Fölin liikennöintialue käsittää Turun, Raision, Naantalin, Ruskon, Liedon ja Kaarinan kunnat. Alueella on noin 3400 tietokantaan tallennettua pysäkkiä, joista kullakin on pysäkin yksilöivä tunniste ja nimi. Eri puolilla tietä sijaitsevat saman nimiset, mutta eri tunnisteiset pysäkit muodostavat pysäkkiparin. Pysäkkitunniste on tyypillisesti merkkijonona esitettävä numerosarja, mutta etenkin terminaalialueilla, kuten Kauppatorilla ja Puutorilla, voidaan käyttää kirjaintunnistetta numeron edessä. [7]

Vaikka useissa tapauksissa pysäkkitunnisteiden numero voi kasvaa juoksevasti tieosuuden edetessä, tai samalla alueella voi tunnistaa samansuuruisia tunnisteita, ei pelkän tunnisteen perusteella voi tehdä varmoja johtopäätöksiä pysäkin ominaisuuksista. Seutukunnissa pysäkin kuntalaisuus on kuitenkin pääteltävissä sen tunnisteesta, tunnisteen sijoittuessa tiettyyn tuhatsarjaan kunnasta riippuen. [7]

2.2 Linjat ja reittivariaatiot

Turun seudulla, kuten kaupunkiliikenteessä yleensä, reitit erotellaan toisistaan linjatunnuksilla. Kirjoitushetkellä linjasto on toteutettu siten, että linjatunnus tarjoaa informaatiota reitin suunnasta, mutta samalla linjatunnuksella voi silti olla useita eri reittivariaatioita. Toisaalta joillain linjoilla linjatunnuksen muodostaa selkeä kirjain-numeroyhdistelmä, jolloin poikkeavat variaatiot ovat pääteltävissä linjatunnuksista. [8]

Reittivariaatiolla tarkoitetaan käytännössä, että samalla linjatunnuksella ajetaan toisistaan poikkeavia reittiosuuksia, jolloin myös variaatioiden pysäkkisarjat ovat erit. Saman linjatunnuksen alla olevat variaatiot erotellaan järjestelmätasolla pysäkkisarjatunnisteella (engl. *pattern code*), joka muodostuu linjatunnuksesta, valinnaisesta variaatiotunnisteesta sekä suunnasta. Tilannetta on havainnollistettu taulukossa 2.1. Reittivariaatioiden pysäk-

kisarjat voivat vaihdella tilanteen mukaan esimerkiksi tilapäisten reittimuutosten vuoksi. Tästä syystä variaatioon liittyvän pysäkkisarjadatan voimassaolo tulee tarkastaa ennen sen hyödyntämistä.

Taulukko 2.1. *Esimerkkejä reittivariaatioiden ilmentymistä. [8]*

Linja-tunnus	*Reitti*	*Pysäkkisarja-tunniste*	*Variaatio linjatunnuk-sessa*
2B	Littoinen - Keskusta - Länsinummi	2B1	Kyllä
2C	Littoinen - Keskusta - Liljalaakso	2C1	Kyllä
2B	Länsinummi - Keskusta - Littoinen	2B2	Kyllä
2C	Liljalaakso - Keskusta - Littoinen	2C2	Kyllä
99	Uittamo - Länsikeskus - Perno	99UT1	Ei
99	Ilpoinen - Länsikeskus - Pansio	99UP1	Ei
99	Perno - Länsikeskus - Ilpoinen	99IT2	Ei
99	Pansio - Länsikeskus - Uittamo	99UP2	Ei

Vuoro (engl. *trip*) on tietyn linjan aikataulun mukainen lähtö ja sitä seuraava matka, johon liittyy linjatunnus, variaatio, pysäkkisarja ja pysäkkisarjan aikataulu. Normaalitilanteessa yhden vuoron ajaa sama ajoneuvo, ellei se esimerkiksi rikkoudu matkalla. Autokierto (engl. *block*) on kokoelma vuoroja, jotka yksi ajoneuvo voi päivän aikana ajaa. Autokierto on yleensä suunnitelma automäärän määrittämiseksi ja sitä voidaan käyttää myös laskutusperusteena tilaajan liikennöitsijälle maksamissa liikennöintikorvauksissa.

2.3 Linjatyypit

Linjaverkko koostuu linjoista, jotka kohdistuvat tietyllä tavalla kaupunkirakenteeseen. Saavuttaessaan riittävän hyvän palvelutason, voidaan linjaa kutsua runkolinjaksi. Runkobussilinjaston kehittämisohjelman raportissa [2] on määritelty runkobussilinjan palvelutaso siten, että pysäkit ovat helposti löydettävissä ja saavutettavissa, linjan tulee liikennöidä varhaisesta aamusta myöhäisiltaan ja linjan vuorovälin tulee olla päiväsaikaan suurimmillaan 10 minuuttia. Vuorovälillä tarkoitetaan kahden peräkkäisen vuoron väliin jäävää aikaa. Runkolinjalla myös matkustajainformaation tulee olla asianmukaista, kaluston tulee olla modernia ja vaihtoyhteyksien on oltava sujuvia.

Millä tahansa linjamallilla voidaan toteuttaa eri asteisia palvelutasoja. Meyer esittelee [5] linjamallien perustyyppejä jakamalla ne neljään pääryhmään: heiluri-, säteis-, poikittais- ja rengaslinjoihin.

Lävistäjälinjoille, tai niin kutsutuille *heilurilinjoille* on ominaista liikennöidä kaupungin kah-

den reuna-alueen välillä kulkemalla jonkin merkittävän terminaalin kautta. Turun seudun joukkoliikenteessä linjastorakenne toteuttaa vahvasti heilurimallia. Turussa paikallisliikenteen terminaalina toimii Kauppatori ja sitä ympäröivät korttelit. Heilurimallia tullaan toteuttamaan myös runkolinjastossa, jolloin vaihtoyhteyksien sujuvuus korostuu joukkoliikenteen kulkutapaosuuden kasvattamiseksi.

Säteislinjat operoivat terminaalin ja taajaman ulkopuolisen kohteen välillä. Heilurilinjasta poiketen toisena päätepisteenä on siis keskusterminaali. Säteislinjalle on ominaista, että matkustajakuormitus kasvaa terminaalia lähestyttäessä ja vastaavasti pienenee terminaalista poispäin ajettaessa. Kaupunkialueen ulkopuolelle ulottuvien maaseutumaisten linjojen linjapituus on usein suuri ja heilurimallin toteuttaminen näillä linjoilla voi edellyttää jopa ajopiirturin käyttöä. Tällöin on tarkoituksenmukaista asettaa keskusterminaali linjan päätepisteeksi.

Tangentti-, eli *poikittaislinjoiksi* voidaan kutsua niitä linjoja, jotka eivät aja keskusterminaalin kautta. Turun sisällä tällaisia linjoja ovat monet työmatka- ja koululaislinjat.

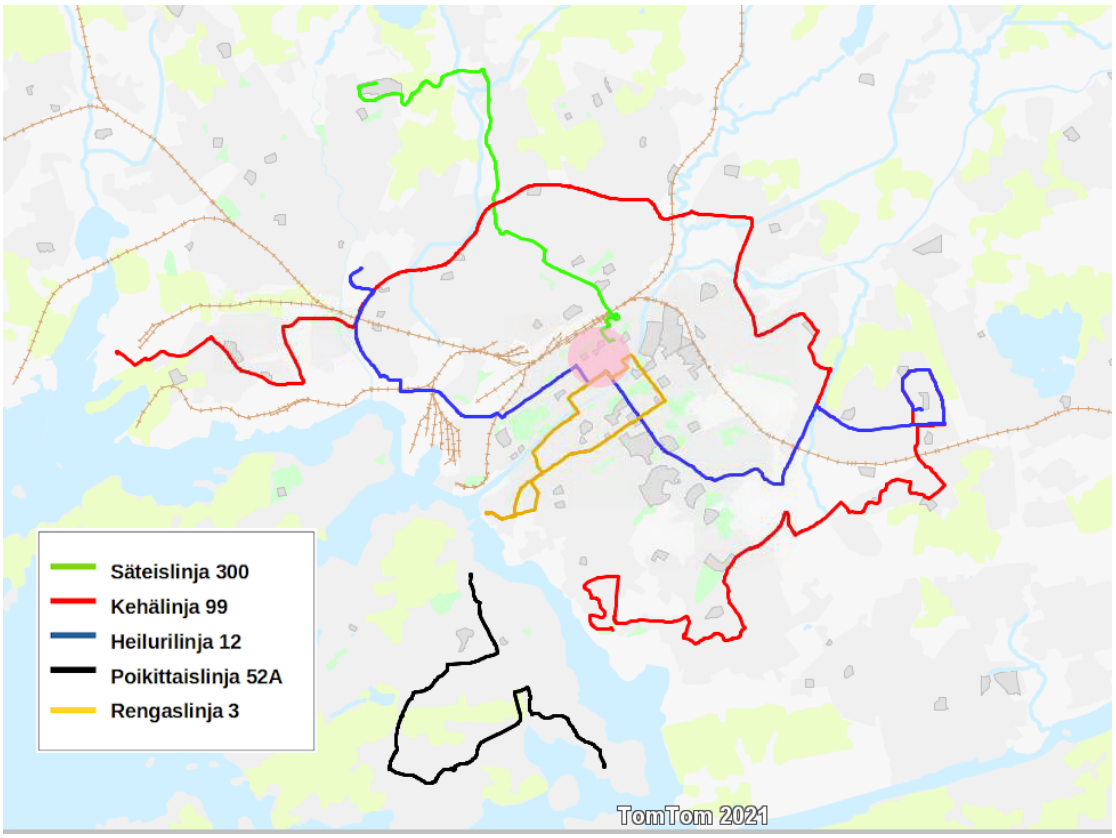
Rengaslinja määritellään linjana, jonka reitti muodostaa suljetun kehän. Turun seudun linjastossa rengaslinjoilla on tyypillisesti jokin suljetusta kehästä erkaantuva haara.

Näiden pääryhmien lisäksi Turussa liikennöidään *kehälinjaa*. Määritelmällisesti myös kehälinja on poikittaislinja, koska reitti ei kulje keskusterminaalin kautta, eikä kehä ole suljettu. Linjaston noudattaessa pääosin heilurimallia, mahdollistaa kehälinja matka-ajan lyhentymisen etenkin silloin, kun matkaketjuun kuuluu kaksi heilurilinjaa, mutta varsinaista tarvetta keskusterminaaliin matkustamiselle ei ole. Kehälinja on huomioitu myös runkolinjastosuunnitelmassa, jolloin se tullaan palvelutason nostamisen myötä määrittelemään yhdeksi runkolinjoista. Vaihtoaikojen lyhentäminen aikatauluja synkronoimalla tällöin luonnollisesti korostuu.

Kuvassa 2.1 on esimerkkejä linjatyyppien ilmentymistä Turun seudulla. Turun ydinkeskusta on merkittynä punaisella täytetyllä ympyrällä. Linja 300 on Turun keskusta-alueella sijaitsevan terminaalin ja Raisiossa sijaitsevan Kauppakeskus Myllyn välillä liikennöivä säteislinja. Linja 99 on pitkähkö, Uittamon ja Pernon välillä liikennöivä, keskustan kiertävä kehälinja. Linja 12 on Turun keskusta-alueen lävistävä heilurilinja, jonka päätepisteet ovat idässä Varissuo ja lännessä Härkämäki. Poikittaislinja 52A on Hirvensalon saarella liikennöivä koululaislinja, joka sopii poikittaislinjan määritelmään. Linja 3 on Turun itäistä keskustaa myötäpäivään kiertävä rengaslinja, joka tekee piston Majakkarantaan. Linjaa liikennöidään vastapäivään linjatunnuksella 30. [8]

2.4 Tunnistepohjainen lippujärjestelmä

Fölissä on käytössä tunnistepohjainen lippujärjestelmä. Tunnistepohjaisuudella tarkoitetaan, että varsinaisessa lipputuotteessa, kuten matkakortissa, ei ole muuta järjestel-

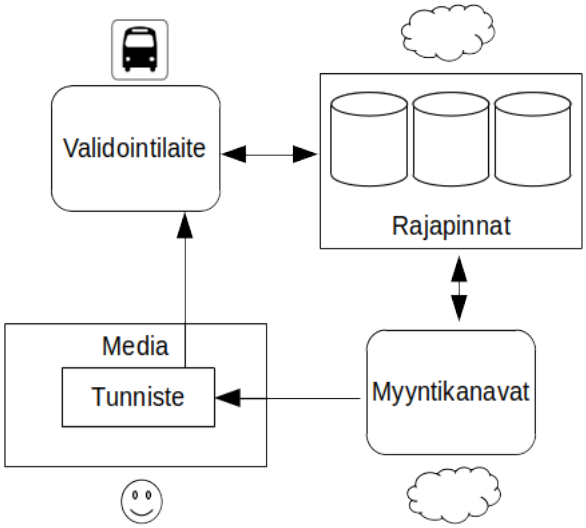


Kuva 2.1. *Linjatyyppien ilmentymiä Turun seudulla viidellä eri linjatunnuksella.*

män tarvitsemaa informaatiota, kuin tunnistenumero. Ajoneuvolaitteet päivittävät jatkuvasti taustajärjestelmän palvelimelta listaa, joka sisältää tiedon kunkin tunnisteen tilasta. Näin ajoneuvolaitteet voivat palauttaa asiakkaalle kortin tilan ja tiedon matkustusoikeudesta sen jälkeen, kun korttia on näytetty lukijalaitteelle. [9]

Koska tunnistenumero voi olla periaatteessa mikä tahansa järjestelmään tallennettu merkkijono, mahdollistaa tunnistepohjainen järjestelmä laajasti eri medioiden käyttämisen maksuvälineenä. Konkreettisia esimerkkejä sovelluksista ovat lähimaksuominaisuudella varustetut pankkikortit ja teatterilippujen viivakoodit, jotka molemmat ovat käytössä maksutapoina Turun seudun joukkoliikenteessä [1]. Tuotejärjestelmän päälle on rakennettu avoin rajapinta, jonka avulla kolmannen osapuolen toimijat voivat myydä lipputuotteita omissa kanavissaan. Tunnistepohjaisen järjestelmän toimintaperiaatetta on havainnollistettu kuvassa 2.2.

Lastenvaunujen kanssa kulkevia, päiväsaikaan rollaattoria käyttäviä sekä maksutta kulkevia lapsia lukuunottamatta jokainen matkustaja ostaa lipputuotteen kuljettajalta tai validoi sen lukijalaitteella. Tämän seurauksena taustajärjestelmän tietokantaan tallennetaan tieto tuotteen tunnistenumerosta ja siitä voidaan muuta dataa yhdistelemällä muodostaa kattava tapahtumarivi sisältäen esimerkiksi ajoneuvonumeron, pysäkkinumeron ja tapahtuman aikaleiman. Näin järjestelmää voidaan hyödyntää myös erilaisissa matkamäärätilastoissa ja nousutapahtumiin liittyvässä analytiikassa.



Kuva 2.2. *Tunnistepohjaisen järjestelmän vuorovaikutus asiakkaan matkustusmedian, ajoneuvon validointilaitteen, sekä taustajärjestelmän välillä.*

Föli ei liikenteen tilaajana omista itse ajoneuvoja, vaan liikenteen tuottava liikennöitsijä järjestää tarvittavan ajoneuvokannan. Liikennöitsijä vastaa itse myös ajoneuvojen järjestysnumeroinnista, jolloin useammalla liikennöitsijällä voi olla samalla järjestysnumerolla liikennöivä auto. Järjestelmätasolla ajoneuvo yksilöidään liikennöitsijätunnisteen ja ajoneuvotunnisteen yhdistelmällä. Tällöin uniikiksi ajoneuvotunnisteeksi muodostuu 5-6 merkkiä pitkä kokonaisluku.

Fölin liikennettä ajava ajoneuvo on lippujärjestelmän osalta varustettu ajoneuvopäätteellä (engl. *main PC*), sekä kortinlukijalaitteella. Päätteeseen on määritetty se ajoneuvotunniste, johon laite on asennettu. Tämän perusteella tapahtumat osataan tietokannassa yhdistää oikeaan autoon. Kuljettaja kirjautuu päätteelle henkilökohtaisilla tunnuksillaan, jolloin työvuoron aikana tehdyt nousutapahtumat ja myynnit kohdistetaan oikealle kuljettajalle ja oikealle liikennöitsijälle.

Kuljettaja valitsee ajamansa vuoron, jolloin auton lähettämät tapahtumat, kuten nousutapahtumat ja toteutuneet pysäkkiajat, voidaan yhdistää vuoron suunniteltuun aineistoon. Ajoneuvopääte lähettää reaaliaikadataa tietyin intervallein, tyypillisesti kolmen sekunnin välein taustajärjestelmään. Reaaliaikadataa voidaan näin hyödyntää myös matkustajainformaatiopalveluissa, kuten pysäkkien aikataulunäytöillä ja karttasovelluksissa.

Kortinlukijalaitteita on joitain poikkeuksia lukuunottamatta ajoneuvossa yksi kappale ja se on sijoitettu auton etuosaan, kuljettajan näkökenttään. Matkustajat validoivat käytännössä kaikki lipputuotteet lukijalla. Kortinlukija on suoraan yhteydessä ajoneuvopäätteeseen ja tunnistepohjaiset nousutapahtumat lähetetään lähes reaaliajassa taustajärjestelmään.

2.5 Matkakorttidatan hyödyntäminen

Lippu- ja maksujärjestelmien toimiessa yhteydessä muihin joukkoliikenteen järjestelmiin, on matkakorttitunnisteiden käyttö perusteltua analytiikkatyössä. Monet alan toimijat hyödyntävät dataa esimerkiksi raportoinnissa ja tilastoinnissa, mutta datan suora hyödynnettävyys esimerkiksi suunnittelutyössä ei kuitenkaan ole itsestäänselvää. Hyödyntämistä voi hankaloittaa käsiteltävän data-aineiston kasvaminen liian suureksi, jolloin halutun tiedon louhinta ei onnistu tavanomaisilla menetelmillä. Matkustusdatan tietosisältö voi myös olla vajaavainen, jolloin dataa joudutaan yhdistelemään eri lähteistä riittävän kattavan aineiston saamiseksi. [10]

Matkakorttitunnisteet ovat usein myös yhdistettävissä henkilöön, jolloin tietojen käyttämiseen voi olla juridisia rajoitteita. Föli toteaa rekisteriselosteessaan tallennettavien tietojen sisällön ja niiden luovutusperusteet. Matkakorttidataa voidaan käyttää esimerkiksi joukkoliikenteen nimiin tehtävään tutkimustyöhön. [11]

Matkustusdatasta voidaan louhia tietoa asiakkaan matkustamasta reitistä. Etenkin tunnistepohjaisuus mahdollistaa datalähteiden yhdistämisen, joka helpottaa esimerkiksi sijaintidatan yhdistämistä. Usein tunnetaan ainoastaan nousupysäkki, jolloin matkan määränpää on jollain menetelmällä pääteltävä yhdistetyn tiedon avulla. [12]

Yksi yleisimmistä keinoista poistumispysäkkien päättelyyn on matkaketjujen löytäminen. Tällöin esimerkiksi kahden matkan ketjussa oletetaan, että ensimmäisen matkaosuuden poistumispysäkki olisi toisen matkaosuuden nousupysäkkiä lähinnä oleva pysäkki. Menetelmä olettaa, että tietystä lähtöpaikasta alkava matkaketju lopulta päättyy samaan paikkaan ja että siirtymät pysäkkien välillä on tehty mahdollisimman lyhyinä. [12]

Vaihtotapahtuma voi syntyä todellisen vaihtoyhteyden lisäksi myös asioinnin, tai aktiviteetin seurauksena. Näissä tapauksissa esimerkiksi kahden matkan ketjussa suoritetaan matkojen välillä jokin aktiviteetti, ja palataan lopuksi lähtöpisteeseen. Nämä tapaukset eivät usein ole relevantteja analysoinnin kannalta. Todellisen vaihtoyhteyden perusteena on käytetty esimerkiksi aikarajaa yhteyksien välillä, eri linjojen käyttöä kullakin matkaosuudella sekä asettamalla ehtoja pysäkkien välisille etäisyyksille. [12]

Matkustajavirtojen klusterointi erilaisilla louhintamenetelmillä on ajankohtainen tutkimuskohde. Matkaketjuja on pyritty klusteroimaan ajan ja paikkatiedon perusteella. K-keskiarvon menetelmällä on tunnistettu poikkeavaa matkustuskäyttäytymistä ajan ja paikkatiedon perusteella tehdyn klusteroinnin pohjalta. Sekä k-means- että Naïve Bayesin menetelmää on käytetty myös matkustajien luokitteluun niiden tekemien matkustusajankohtien perusteella. [12]

3. SAMANKALTAISUUS JA LUOKITTELU

Linja-autolla toteutunut matkaketju voidaan ajatella kuljettujen pysäkkien järjestettynä joukkona, jonona. Koska kunkin pysäkin yksilöi sen merkkijonotunniste, saadaan matkaketjusta näin muodostettua merkkijonolista. Keskenään samankaltaiset matkaketjut ovat näin ollen keskenään samankaltaisia merkkijonolistoja, jolloin niiden vertailu tekstuaalisen sisällön samankaltaisuuden vertailuun tarkoitetuilla menetelmillä on perusteltua.

3.1 Joukon vektorisointi

Yleisiä tapoja tekstidokumentin muuttamiseksi kokonaislukuvektoriksi ovat määrä- ja esiintyvyystiheysmatriisit. Yksinkertaisimmillaan tämä tarkoittaa kaksiulotteista taulukkorakennetta, jossa rivi-indekseinä ovat kaikki kokoelmassa esiintyvät termit ja sarakeindekseinä kaikki kokoelman dokumentit. Sarakkeet ovat laskennan seurauksena keskenään samanpituisia kokonaislukuvektoreita. [13]

Esimerkiksi analysoitaessa samankaltaisia tekstiartikkeleita on syytä huomioida, että laskettaessa yksittäisiä sanoja (engl. *unigram*), niiden semanttinen yhteys voi kadota. Tällöin rivi-indekseinä voidaan käyttää useamman sanan ryhmiä (n-gram), jolloin myös sanojen järjestykseen liittyvä merkitys voidaan ottaa huomioon. [14]

3.2 Samankaltaisuuden määrittely

Tekstin samankaltaisuuden vertailuun soveltuvia menetelmiä on tutkittu Gomaan ja Fahmyn artikkelissa [15], jossa menetelmät luokiteltiin analysoitavan aineiston mukaan merkkijonoperusteisiin, kokoelmaperusteisiin ja tietoperusteisiin menetelmiin, joista merkkijonoperusteiset menetelmät edelleen merkki- ja termiperusteisiin menetelmiin. Esitellyt kokoelmaperusteiset menetelmät liittyvät pitkälti merkitykseltään samankaltaisten sanojen löytämiseen kokoelmatasolla. Tietopohjaiset menetelmät laajentavat vertailuaineistoa esimerkiksi synonyymejä sisältävillä tietokannoilla, mutta kyse on edelleen pääasiassa semanttisen samankaltaisuuden löytämisestä. Merkkitason menetelmät ovat kiinnostuneita merkkijonojen sisällöstä.

Pysäkkitunnisteet ovat samanlaiset vain, jos merkkijonon merkkien järjestys ja sisältö ovat samat. Tunnisteen sisältö ei kuitenkaan kerro mitään pysäkin sijainnista, vaan läheisesti

samankaltainen tunniste voi sijaita toisella puolen kaupunkia. Näin ollen tässä työssä perehdytään artikkelissa esitettyihin termiperusteisiin menetelmiin, joiden voidaan olettaa soveltuvat useasta merkkijonosta koostuvien joukkojen vertailuun.

Kosinisamankaltaisuus

Kosinisamankaltaisuus, kosinietäisyys, on funktio, joka palauttaa arvon väliltä 0–1, jolloin arvolla 1 vektorit ovat samat. Laskenta perustuu vektorien välisen kulman kosiniin. Kahta vektoria vertailtaessa kosinietäisyys saadaan kaavalla

$$cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{(\sum_{i=1}^{n} A_i^2)}\sqrt{(\sum_{i=1}^{n} B_i^2)}}$$

jossa A ja B ovat vertailtavat vektorit. [16]

Dicen kerroin

Dicen kerroin (engl. *Dice's coefficient*) ilmoittaa kahden joukon kaksinkertaisen alkioiden leikkauksen suhteessa molempien joukkojen alkioiden lukumäärään. Vastauksena tulee arvo väliltä 0–1, jonka perusteella samankaltaisuutta voidaan arvioida halutulla kynnysarvolla [17]. Dicen kerroin lasketaan kaavalla

$$DC(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

jossa A ja B ovat vertailtavat joukot.

Yhteensopivuuskerroin ja päällekkäisyyskerroin

Yhteensopivuuskerroin (engl. *matching coefficient*) eli yksinkertainen yhteensopivuuskerroin (engl. *simple matching coeffiecient*) on kahden vertailtavan vektorin alkioiden lukumäärän, jossa molemmilla alkioilla on sama arvo, suhde vertailtavien alkioiden kokonaismäärään.

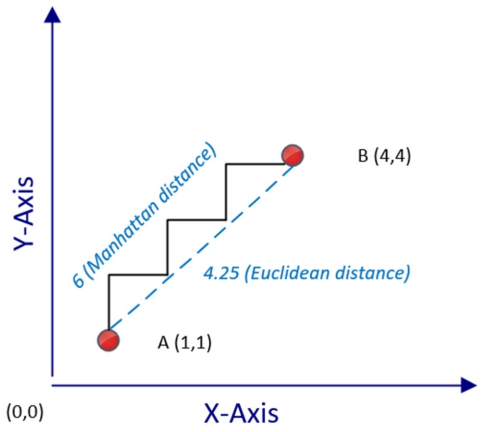
Päällekkäisyyskerroin (engl. *overlap coefficient*) palauttaa arvon 1, kun vertailtava merkkijono on toisen merkkijonon osajoukko. [15]

Euklidinen etäisyys ja Manhattan-etäisyys

Euklidinen etäisyys (engl. *euclidean distance*) on eräs yksinkertaisimmista samankaltaisuusalgoritmeista. Algoritmin palauttama etäisyys on kahden pisteen, tai vektorin välisen suoran pituus. Moniulotteisen vektorin tapauksessa lasketaan vastaavien alkioiden etäisyyksien neliöiden summan neliöjuuri. [18]

Euklidisen etäisyyden ollessa lyhin mahdollinen etäisyys kahden pisteen välillä, on Manhattan-etäisyys (engl. *Manhattan distance, Block distance*) pisin mahdollinen. Tilannetta on havainnollistettu kuvassa 3.1. Algoritmin käyttötapauksiin lukeutuvat juurikin etäisyyslasken-

ta kartalla, jolloin matka on pidempi kuin linnuntietä laskettaessa. Tästä johtuen Manhattan-etäisyys on aina suurempi tai yhtäsuuri kuin euklidinen etäisyys. Etäisyys voidaan laskea kahden vektorin vastaavien alkioiden erotusten summana. [15, 18]



Kuva 3.1. *Manhattan-etäisyyden ja euklidisen etäisyyden vertailu [18].*

Jaccard-samankaltaisuus

Jaccard-samankaltaisuus (engl. *Jaccard similarity*), tai Jaccard-indeksi (engl. *Jaccard index*), pyrkii mittaamaan kahden joukon samankaltaisuutta niiden samanlaisten alkioiden suhteella alkioiden kokonaismäärään. Toisinsanoen samankaltaisuus on joukkojen leikkauksen suhde niiden unioniin. Unioni käsittää joukkojen kaikki yksilölliset alkiot ja leikkaukseen kuuluvat ne alkiot, jotka kuuluvat molempiin joukkoihin. Mittarina voidaan käyttää myös joukkojen erilaisuutta kuvaavaa Jaccard-etäisyyttä (engl. *Jaccard distance*), joka on Jaccard-indeksin komplementti. [19]

Kuvassa 3.2 on esimerkki joukkojen A ja B Venn-diagrammista, jolla samankaltaisuutta voidaan havainnollistaa. Joukot A ja B ovat datasta tunnistettuja vaihtomatkaketjuja. Joukkojen unioniin kuuluu $11 + 22 + 13 = 46$ alkiota ja joukkojen leikkaukseen 22 alkiota.

Jaccard-samankaltaisuus lasketaan kaavalla

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{22}{46} = 0.48$$

jossa A ja B ovat aiemmin mainittuja pysäkkijoukkoja. Jaccard-indeksiksi saadaan näin ollen 0,48.

3.3 Klusterointimenetelmät

Tyypillisesti klusterointimenetelmät jaetaan osittaiseen ja hierarkkiseen klusterointiin. Osittainen klusterointi perustuu käyttäjäparametrin perusteella tehtävään iteroivaan luokitteluun, jolloin tavoitteena on klusterin laadun parantaminen jokaisella iteraatiolla. Osittai-

```
A = {'1048', '1042', '1711', '856', '1047', '65', '220', '1486', '1721', '1044', '1472',
'66', '1911', '1045', '635', '1485', '1046', '219', '996', '637', '505', '1712', '1983',
'854', '507', '636', '1719', '1487', '1488', '506', '1489', '1722', '1717'}

B = {'1711', '856', '10', '4', '1721', '109', '1486', '1472', '7', '12', '9', '635', '6',
'1485', '2', '996', '3', '637', '505', '1902', '1712', '1983', '854', '11', '507', '636',
'5', '8', '1719', '1487', '1488', '506', '1489', '1722', '1717'}
```

Kuva 3.2. *Kahden pysäkkijoukon VENN-diagrammi.*

sessa klusteroinnissa käytetään usein ns. virtuaalista samankaltaisuusmetriikkaa, kun taas hierarkkinen klusterointi perustuu aineiston absoluuttisiin arvoihin. [20]

Yksi yleisimmin käytetyistä osittaisen klusteroinnin menetelmistä on euklidista etäisyyttä hyödyntävä K-means -menetelmä, jonka iteraatioissa liitetään klusteri keskiarvoltaan lähimpään toiseen klusteriin ja uudelle klusterille lasketaan keskiarvo. Iteraatiota toistetaan, kunnes käyttäjäparametrina annettu k määrä klustereita on syntynyt. [21]

Hierarkkiset klusterointimenetelmät jaetaan alhaalta ylöspäin eteneviin kokoaviin ja ylhäältä alaspäin eteneviin jakaviin menetelmiin. Kokoavissa menetelmissä yksittäiset datapisteet tai ilmentymät yhdistetään uudeksi klusteriksi jonkin etäisyysarvon perusteella. Jakavissa menetelmissä prosessi etenee päinvastaiseen suuntaan. Iteraatiota toistetaan, kunnes jokin ennalta määritetty ehto on saavutettu. Kunkin iteraation lopputulos on peruuttamaton eikä samaan klusteriin päätyneitä datapisteitä voida hyödyntää sellaisenaan seuraavassa iteraatiossa, vaikka ne olisivatkin soveltuneet siihen edellistä paremmin. Hierarkisten menetelmien yksinkertaisuus on selkeä etu, mutta huomioitava seikka on, että eri menetelmillä toteutettu kokoaminen voi tuottaa erisisältöiset klusterit. [22]

3.4 Locality-sensitive hashing

Locality-sensitive hashing (LSH) on funktioperhe lähekkäisten datapisteiden löytämiseen lähimmän naapurin (engl. *nearest neighbour*) periaatteella [23]. Locality-sensitive voi olla mikä tahansa funktioperhe, joka toteuttaa seuraavat ehdot [19]:

1. Funktioiden tulee löytää ensisijaisesti samankaltaisia pareja.

2. Funktioiden tulee olla tilastollisesti riippumattomia, jotta voidaan arvioida todennäköisyyttä sille, että kaksi tai useampi funktio antaa tietyn vastineen todennäköisyyden tulosäännön mukaisesti.
3. Funktioiden tulee olla tehokkaita:
   (a) Samankaltaiset parit tulee löytää nopeammin kuin jokaista dokumenttia vertailemalla, ns. brute force menetelmällä.
   (b) Funktiot tulee olla yhdistettävissä siten, että virheelliset positiiviset ja virheelliset negatiiviset tilanteet voidaan välttää suoritusajan ollessa kuitenkin nopeampi kuin brute force -menetelmällä.

3.4.1 Minhash

Blekanov ja Korelin ovat päätyneet [24] tekstiaineistojen klusterointia käsittelevässä työssään hyödyntämään Jaccard-indeksiä. Ongelmana on nähty dokumenttien vertailuun edellytettävän totuustaulun väljyys, jonka generointi ja jolla tehtävät operaatiot käyttävät resursseja tarpeettoman tehottomasti. Tiivistämällä matriisi edellämainitut ehdot täyttävän minhash-funktion avulla, matriisista saadaan tiiviimpi ja Jaccard-indeksiä voidaan soveltaa suuriinkin data-aineistoihin [25]. Väljän matriisin ongelmaa on pyritty havainnollistamaan kuvassa 3.3.

	$Matka_1$	$Matka_2$	$Matka_3$	…	$Matka_{82354}$
'1'	0	0	0	…	0
'2'	0	0	0	…	0
'3'	0	1	0	…	0
'4'	0	1	0	…	0
'5'	0	0	0	…	0
'6'	0	0	0	…	0
'7'	1	0	1	…	0
'8'	0	0	1	…	0
'9'	0	0	1	…	0
'10'	0	0	0	…	1
'11'	0	0	0	…	0
⋮	⋮	⋮	⋮	…	⋮
'3404'	0	0	0	…	0
'3405'	0	0	0	…	0

Kuva 3.3. *Esimerkki väljästä matriisista, jossa sarakkeina ovat kuvitteelliset matkaketjut ja riveinä unigram-tarkkuisista pysäkkitunnuksista muodostuvat päreet.*

LSH:ta on käytetty useissa tutkimuksissa [24, 26, 27] minhash-algoritmin tuottamalla matriisilla, kun kahden joukon samankaltaisuuksia on haluttu löytää. Klusteroinnin lopputulos

LSH:ta hyödyntäen on ollut sama kuin perinteisemmillä klusterointimenetelmillä, mutta nopeushyöty on LSH:ta sovellettaessa merkittävä [24].

Tekstidokumenttien tapauksessa minhash-prosessi alkaa aineiston jakamisella päreiksi (engl. *shingle*). Tekstiaineistosta muodostetaan *k* pituisia peräkkäisiä merkkijonoja, jolloin lyhyidenkin samojen ilmaisujen esiintyvyys molemmissa aineistoissa voidaan huomata. Myös riippuvuus sanojen järjestyksestä poistuu. Valitun päreen pituus vaikuttaa luonnollisesti siihen, miten samankaltaisina joukot nähdään. Pitkille dokumenteille on syytä valita suurempi päreen pituus kuin pienille. [19]

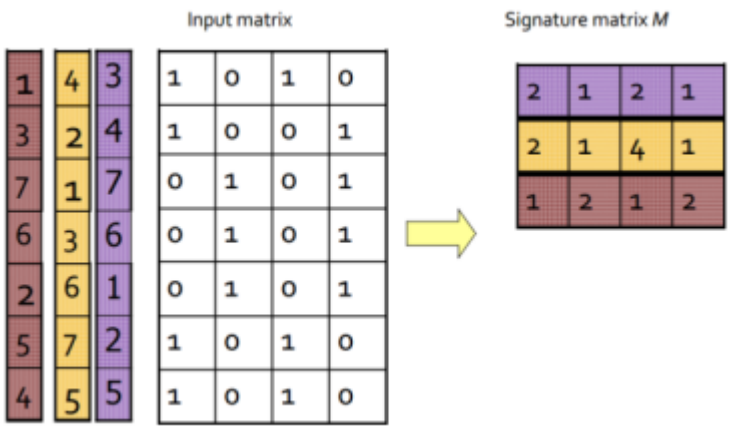
Lähtötilanteessa käytössä on väljä totuustaulu, jossa rivi-indekseinä ovat kaikki aineistokokoelmassa esiintyvät päreet ja sarakkeina dokumentteja kuvaavat vektorit. Vektorin alkio on 1 silloin, kun sitä vastaavan indeksin päre esiintyy dokumentissa. Muussa tapauksessa alkio saa arvon 0.

Minhash-menetelmällä rakennetaan uusi tiiviimpi matriisi, johon voidaan tehokkaammin kohdistaa operaatioita. Matriisin englanninkielinen nimitys on *signature matrix*, jonka vapaa käännös tässä työssä on tiivistematriisi. Tiivistematriisi on m x n matriisi, jossa m on suoritettavien sekoitusfunktioiden, eli permutaatioiden h lukumäärä, ja n on mukana olevien dokumenttien tai joukkojen lukumäärä. Käytännön toteutuksissa voidaan nähdä myös matriisin transpoosi, jossa matriisin rivi m vastaa dokumenttia ja sarakkeet n permutaation tulosta.

Perusajatuksena on, että vektoreiden rivi-indeksit sekoitetaan m kertaa ja kunkin sekoituskerran jälkeen poimitaan pienin rivi-indeksi, jonka alkiossa on arvo 1. Indeksin numero siirretään tiivistematriisiin. Koska menetelmä on raskas etenkin aineistokoon kasvaessa, voidaan samaan lopputulokseen päästä käyttämällä sekoitusfunktiona h esimerkiksi hajautusfunktiota, joka antaa kullekin riville satunnaisen hajautuskoodin.

Kuvassa 3.4 on havainnollistettu prosessia kolmella permutaatiolla. Jokaisella permutaatiolla annetaan satunnaiset rivinumerot. Tiivistematriisiin siirretään kunkin vektorin pienin indeksi, jonka alkion arvo on 1.

Minhash-menetelmän toimivuus perustuu väitteeseen, että kahden joukon A, B Jaccard-indeksi on sama kuin todennäköisyys sille, että sekoitusfunktio palauttaa saman indeksin joukoille A ja B. Tämän todistaa oikeaksi se, että Jaccard-indeksi on joukkojen leikkauksen koko suhteessa unioniin, ja vain jos alkiot kuuluvat leikkaukseen, on niiden molempien arvo totuustaulussa 1. Näin ollen todennäköisyys sille, että indeksien numerot ovat samat, on sama kuin leikkaukseen kuuluvien alkioiden lukumäärä suhteessa kaikkiin alkioihin. [19]



Kuva 3.4. *Esimerkki minhash-prosessista, jossa suoritetaan kolme permutaatiota [24].*

3.4.2 Samankaltaisten tulosten etsiminen

Vaikka samankaltaisten parien löytäminen tiivistematriisista on jo lähtökohtaisesti tehokkaampaa kuin väljästä matriisista, voidaan operaatiota edelleen tehostaa LSH-menetelmällä. Tehokkuuden hintana on lopputuloksen luotettavuus, mutta ajansäästö voi olla aineiston koosta riippuen erittäin merkittävä.

LSH-menetelmä voidaan toteuttaa esimerkiksi jakamalla minhash-algoritmin tuottama matriisi b ryhmään, siten että kussakin ryhmässä b_i on r riviä. Jokaiselle ryhmälle suoritetaan algoritmi, jonka seurauksena ryhmän jokainen vektori saa hajautuskoodin. Hajautuskoodi voi olla esimerkiksi binäärikoodi, joka rakennetaan toistamalla algoritmia siten, että jokaisella toistokerralla koodiin lisätään luku 1, tai 0. Tällöin toistomäärä on sama, kuin koodin pituus k.

Kullekin ryhmälle b_i luodaan oma hajautustaulu H_i, jossa saman hajautuskoodin saaneet vektorit ovat samassa korissa. Jos vektori on samassa korissa yhdessä tai useammassa hajautustaulussa, voidaan puhua kandidaattipareista, jolloin vektorit ovat suurella todennäköisyydellä samat. [23]

Muuttujat b ja r vaikuttavat lopputuloksen tarkkuuteen. Kaavassa

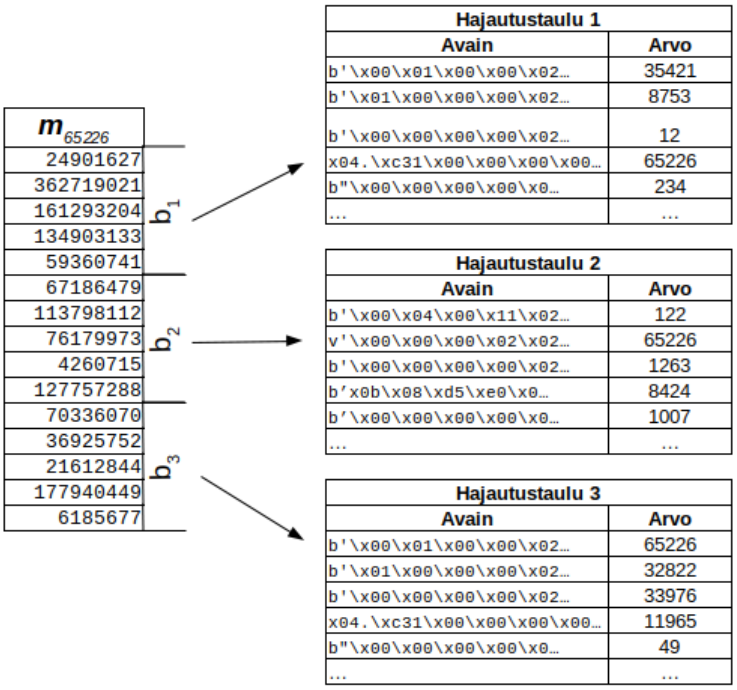
$$t = \left(\frac{1}{b}\right)^{\frac{1}{r}} = \left(\frac{1}{5}\right)^{\frac{1}{25}} = 0,938$$

on esitetty ryhmien määrän b ja ryhmässä olevien rivien tai alkioiden määrän r vaikutuksen samankaltaisuuteen t. Kaavaan on esimerkkinä sijoitettu 128-rivisen tiivistematriisin jako, jolla saadaan haluttu kynnysarvo t lähelle arvoa 0,9. [19]

Eric Zhu on toteuttanut menetelmän käytännössä Python-ohjelmointikielellä. Parametreina annetaan haluttu Jaccard-indeksin kynnysarvo t sekä aineistosta h määrällä permu-

taatioita rakennettu tiivistematriisi M, jonka rivejä m ovat aineiston dokumentit tai muut objektit. Algoritmi laskee LSH-parametrit eli ryhmien määrän b ja rivien määrän r kynnysarvon t perusteella. [28]

Prosessissa rakennetaan LSH-indeksi, josta samankaltaisuuksien hakeminen on tehokasta. Rakennusperiaate on havainnollistettu kuvassa 3.5. Indeksi rakennetaan antamalla syötteenä vektori, joka on yksittäinen tiivistematriisin rivi m. Vektori jaetaan b osaan ja jokainen osa tiivistetään tavukoodiksi Pythonin pickle-ohjelmointikirjastolla. Tavukoodi viedään avaimeksi ryhmää vastaavaan hajatutustauluun aiemmin esitetyn teorian mukaisesti. Avainta vastaava arvo on syötetyn minhash-rivin indeksi, joka on myös alkuperäisen aineiston vastaavan objektin indeksi. [28]



Kuva 3.5. *LSH-indeksin rakennusmenetelmä.*

Kun indeksiin kohdistetaan hakuoperaatio, annetaan hakuparametrina objektin minhash-vektori. Vektori jaetaan rakennusvaihetta vastaavalla tavalla b osaan, jotka muutetaan pickle-tavukoodeiksi. Hajautustaulusta H_i voidaan nyt hakea koodia b_i vastaava avain, joka lisätään kandidaattijoukkoon. Muuttujien b ja r arvot ovat siis lopputuloksen kannalta oleellisia, koska kandidaattiobjekti lisätään joukkoon sen löytyessä yhdestäkin hajautustaulusta. [28]

3.4.3 Kokoava klusterointi

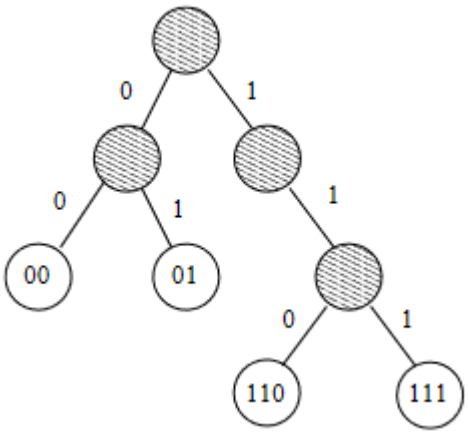
Koga, Ishibashi ja Watanabe [23] esittelivät vuonna 2005 LSH-link-algoritmin, joka toimii kokoojamenetelmänä LSH-funktioita toteuttaen. Kussakin iteraatiossa etsitään etäisyydellä r olevat klusterit ja kootaan ne edelleen yhdeksi klusteriksi. Etäisyysmitta r sää-

detään ensimmäisellä iteraatiolla k:n mukaan. LSH-hajautuksen jälkeen etsitään koreista ne pisteet, jotka ovat r etäisyyttä pienemmällä etäisyydellä tarkastelupisteestä ja yhdistetään ne uudeksi klusteriksi. Vastaavankaltaiset klusteriparit muista koreista yhdistetään. Jos iteraation päätyttyä jäljellä on enemmän kuin yksi klusteri, aloitetaan alusta pienemmällä k-arvolla, ja sen myötä suuremmalla r-arvolla.

3.4.4 Vakiokokoisen lähimpien naapureiden joukon löytäminen

Bawa, Condie ja Ganasan esittelivät [29] vuonna 2005 LSH-metsä (engl. *LSH-forest*) -algoritmin, jossa LSH-hajautuskoodin pituus k on mukautuva ja joka tarjoaa perinteisiä LSH-menetelmiä paremman tarkkuuden heikentämättä laskentatehon hyötysuhdetta, tai lopputuloksen virheettömyystasoa.

LSH-metsä on l kokoinen kokoelma LSH-puita, joiden lehtiä ovat ainestossa esiintyvät datapisteet eli esimerkiksi dokumentit. Perinteisemmissä LSH-menetelmissä korin nimi on k-pituinen koodi, mutta LSH-metsässä k on muuttuja. Jokainen lehti saa nimekseen yksilöllisen muuttujan k pituisen koodin, joka on myös juuresta solmujen kautta lehteen kuljettu polku. Kuvassa 3.6 on yksittäinen neljä pistettä sisältävä LSH-puu.



Kuva 3.6. *Neljä pistettä sisältävä LSH-puu.*

Hakumenetelmä perustuu m lähimmän naapurin löytämiselle. Kaksiosaisen menetelmän top-down-algoritmi alkaa juuresta ja etenee solmuittain alaspäin. Jokaisella tasolla tarkastetaan,onko solmu lehti ja jos on, tallennetaan solmun polku ja syvyys. Muutoin kasvatetaan syvyyttä ja laskeudutaan edelleen tarkastelemaan seuraavia lapsisolmuja, kunnes vertailupistettä vastaava solmu löytyy.

Bottom-up-vaiheessa tavoitteena on kerätä metsästä M pistettä iteroimalla top-down-vaiheessa tallennetut arvot. Bottom-up-vaihetta toistetaan, kunnes saavutetaan juuri tai kunnes m pistettä on kerätty. M:n on oltava dynaaminen muuttuja, jotta voidaan varmistua, että m lähintä naapuria saadaan täyteen.

4. VAIHTOMATKAKETJUJEN KÄSITTELY

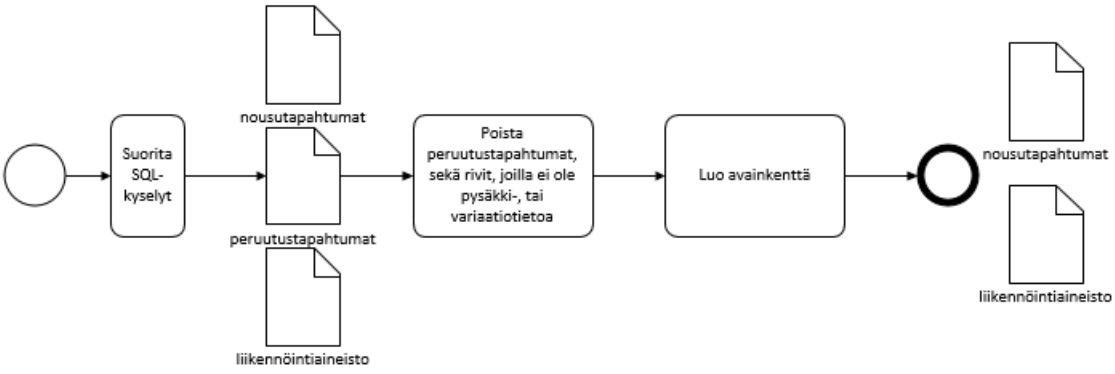
Työn tarkoituksena on yhdistää kahdesta eri tietolähteestä saatavaa dataa vaihtomatkaketjuja kuvaavien pysäkkijoukkojen muodostamiseksi. Soveltamalla samankaltaisuuden vertailuun käytettäviä menetelmiä pyritään tunnistamaan keskenään samankaltaiset matkaketjut ja ryhmittelemään ne samankaltaisuuden perusteella klustereihin.

Tutkimus tehtiin pääasiassa suorittamalla Python-ohjelmointikielellä kirjoitettua ohjelmakoodia, josta muodostettiin VPN-tunnelia käyttämällä suora yhteys joukkoliikenteen raportointitietokantoihin. Datan prosessointi tapahtui koodissa paikallisesti, jonka jälkeen prosessoitu aineisto tallennettiin JSON-tiedostoihin. Tuotantokäytössä data voidaan näin tallentaa esimerkiksi NoSQL-tietokantaan.

Tutkimuksessa yhdistetään yhdellä suorituskerralla yhden vuorokauden aineisto. Tällöin prosessi voidaan ajaa tuotantokäytössä ajastetusti päivittäin ja kulloinkin voimassaolevan reittidatan yhdistäminen on varmempaa.

4.1 Aineiston kerääminen

Aineisto haettiin joukkoliikenteen raportointitietokannasta SQL-kyselyillä. Eri segmenteistä noudettu data valmisteltiin poistamalla tutkimusta haittaavat rivit sekä yhdistämällä osaaineistot luotujen avainkenttien perusteella. Prosessi on kuvattu kuvassa 4.1.



Kuva 4.1. *Aineiston kerääminen ja esivalmistelu.*

4.1.1 Nousutapahtuma-aineisto

Nousutapahtumia varten on tietokannassa valmiiksi koottu näkymä, johon raportoidaan kaikki järjestelmässä tapahtuvat lipputuotetapahtumat. Tehokkuuden optimoimiseksi prosessointiin haetaan ainoastaan tarvittavat kentät, jotka on listattu taulukossa 4.1.

Taulukko 4.1. *Nousutapahtumiin liittyvät kentät ja niiden selitykset.*

Kentän nimi	*Selite*	*Tietotyyppi*
TRANSACTIONJOURNALID	Tapahtuman yksilöivä tunnus	int
FAREMEDIATYPE	Lipputuotemedian tyyppi	int
DEVICETIME	Leimauslaitteen aikaleima	datetime
LINE	Linjatunnus	string
DIRECTION	Suunta	int
ROUTE_CODE	Mahdollinen reittivariaatiotunnus	string
STOPNUMBER	Pysäkkinumero	int
FAREMEDIAID	Lipputuotemedian yksilöivä tunnus	string

Varsinaiset matkustustapahtumat erotellaan asettamalla ehdoiksi, että tapahtumalla on oltava linjatunnus ja tapahtumatyypin on oltava nousutapahtumalle ominainen. Kaikissa kyselyissä käytetään ehtona samoja aikaleimoja, jotka on tallennettu muuttujiin *datefrom* ja *dateto*.

Osa tapahtumista voi olla peruutettu jälkeenpäin. Peruutustapahtuma raportoidaan omana rivinään ja rivin kenttä *cancellationreferenceid* viittaa peruutetun tapahtuman *transactionjournalid*-kenttään. Peruutustapahtumien poistamisen katsottiin olevan helpompaa vasta ohjelmakoodissa, joten peruutetut tapahtumat haettiin kannasta omalla kyselyllä ja poistettiin aineistosta. Tämä mahdollisti myös peruutustapahtumien määrän suoraviivaisen analysoinnin: suurin osa peruutustapahtumista on pankkikortilla tehtyjä lähimaksuleimauksia, joissa rahaa ei ole saatu perittyä asiakkaalta.

Nousutapahtumiin liittyvä metatieto, kuten pysäkki ja linja, ovat täysin riippuvaisia auton laitteiston moitteettomasta toiminnasta. Jos auton paikannustieto on puutteellista, ei myöskään pysäkkitietoa saada ja tällöin kantaan raportoidaan pysäkiksi ’0’. Mobiililaitteilla tehtyjen leimausten raportoinnissa huomattiin virhe *routecode*-kentässä, sillä mobiilitapahtumien yhteydessä tätä tietoa ei raportoida lainkaan. Molemmat edellämainitut tiedot ovat oleellisia datan yhdistämisessä, joten virheelliset rivit poistettiin aineistosta.

4.1.2 Liikennöintiaineisto

Lipputuotetapahtumiin ja rahaliikenteeseen keskittyvään tietokantasegmenttiin ei raportoida reittidataa sisältävää liikennöintiaineistoa, joten tiedon yhdistäminen ei onnistu samassa kyselyssä. Reittidata haetaan siis erillisellä kyselyllä omasta segmentistään. Liikennöintiaineisto koostuu suunnitellusta datasta (NOM) sisältäen tietoa mm. aikatauluista, reiteistä ja pysäkeistä; ajoneuvodatasta (VEH) sisältäen tietoa mm. toteutuneista ajoajoista ja ajoneuvossa tapahtuneista toiminnoista kuten ovien sulkeutumisesta; liikennöintipäivädatasta (OPD) sisältäen tietoa mm. liikennöintikalenterista; sekä automaattiseen ajoneuvopaikannukseen (AVL) liittyvästä datasta sisältäen tietoa pääasiassa tähän liittyvien komponenttien, kuten ovien ja GPS-antennien toiminnasta. Tässä tutkimuksessa käytettiin ainoastaan suunniteltua liikennöintidataa, jonka sisältö on listattu taulukossa 4.2.

Taulukko 4.2. *Liikennöintiaineistoon liittyvät kentät ja niiden selitykset.*

Kentän nimi	*Selite*	*Tietotyyppi*
PATTERN_CODE	Reittivariaation pysäkkisarja, sisältäen linjatunnuksen, variaatiotunnuksen ja suunnan	string
STOP_ID	Pysäkkinumero	int
INDEX_NO	Pysäkin järjestysnumero pysäkkisarjassa	int
GPS_LONGITUDE	Koordinaatti	float
GPS_LATITUDE	Koordinaatti	float

Aineistoa varten yhdistellään dataa kolmesta taulusta, jotka sisältävät tietoa pysäkkisarjoista, nimenomaiseen pysäkkisarjaan kuuluvasta pysäkkipisteestä sekä pysäkkipisteistä yleisesti. Käyttämällä globaaleja aikaleimamuuttujia, on käytössä nousutapahtuman aikaan voimassaoleva aineisto, joka tekee datan yhdistämisestä varmempaa mm. aiemmin mainittujen pysäkkisarjojen osalta.

4.1.3 Aineistojen yhdistäminen

Jotta liikennöintiaineistoa voidaan hyödyntää, tulee nousutapahtumadatasta löytyä soveltuva avainkenttä. Nousutapahtuman yhteydessä raportoidaan linjatunnus, variaatiotunnus ja suunta. Yhdistämällä nämä saadaan liikennöintiaineiston *pattern_code*-tieto eli pysäkkisarjatunnusta vastaava merkkijono. Nousutapahtuma-aineistoa laajennettiin luomalla *pattern_code*-kenttä, jolloin sitä voidaan käyttää avaimena.

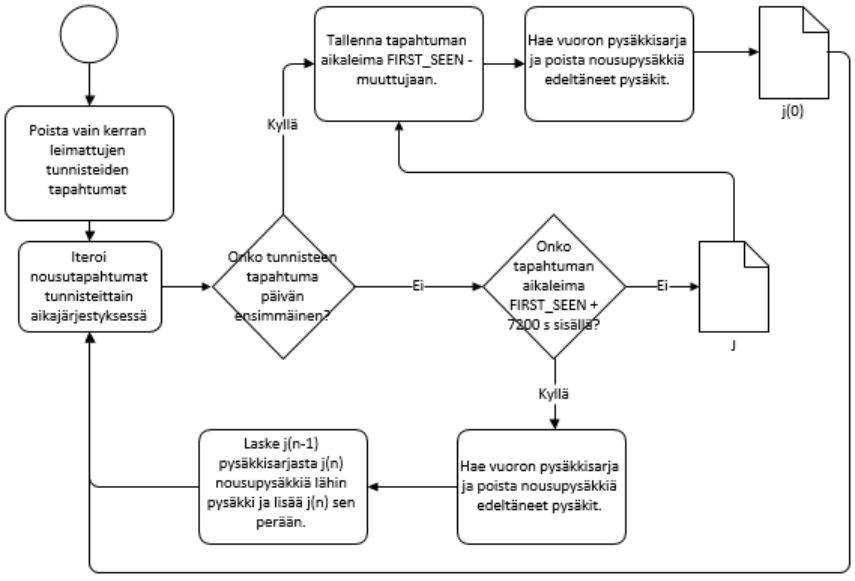
Pysäkkisarjatunnus koostuu linjatunnuksesta, valinnaisesta variaatiotunnuksesta ja suunnasta. Esimerkiksi linjan 6 pysäkkisarjatunnus ilman täsmentävää variaatiotunnusta voi

olla ’61’, jolloin kyseessä on perusmuotoinen linjan 6 lähtö Liedosta Naantaliin. Jos vuoro kiertää ruuhka-aikana reitin varrella olevan Suovuoren alueen kautta, on sille määritetty eri pysäkkisarja, jonka tunnus on ’6S1’ sisältäen myös Suovuoren kierroksen pysäkit.

4.2 Vaihtomatkaketjun tunnistaminen ja listan muodostaminen

Tässä työssä vaihtomatkaketjulla tarkoitetaan yhden asiakkaan linja-autolla tekemiä matkustuksia, joiden leimaustapahtumat ajoittuvat 7200 sekunnin aikaikkunaan ensimmäisestä leimauksesta laskettuna. Aikaikkuna on valittu kertalipuille ja arvotuotteille määritetyn kahden tunnin maksuttoman vaihto-oikeuden perusteella. Asiakas yksilöidään matkustusmedian, kuten matkakortin, yksilöllisellä tunnistenumerolla. Varsinaista matkaketjua ei synny, jos asiakas on matkustanut päivän aikana vain kerran, joten jokaista tunnistenumeroa kohden laskettiin tapahtumarivien määrä ja aineistosta poistettiin tunnisteet, joita on käytetty päivän aikana vain yhden kerran.

Kuvassa 4.2 on esitetty vaihtomatkaketjun muodostamisen logiikkaa. Kullakin tunnisteella aikaikkunassa tapahtuneet nousutapahtumat järjestetään aikajärjestykseen. Ensimmäiseen tapahtumaan liittyvän pysäkkisarjatunnisteen perusteella haetaan sinä päivänä voimassa oleva pysäkkisarja, joka leikataan nousupysäkin kohdalta siten, että sarjan alkuosa jää ketjusta pois. Vastaavasti haetaan seuraavaan nousutapahtumaan liittyvä pysäkkisarja ja mahdollinen alkuosa jätetään pois. Edeltävästä sarjasta päätellään poistumispysäkki tarkastamalla kunkin pysäkin GPS-koordinaattien etäisyys linnuntietä seuraavalle nousupysäkille. Toisaalta, jos seuraavan tapahtuman nousupysäkki on myös edellisessä sarjassa, oletetaan tämän olevan poistumispysäkki. Edeltävästä ketjusta poistetaan poistumispysäkin jälkeinen osa ja se yhdistetään vuorossa olevaan ketjuun.



Kuva 4.2. *Vaihtomatkaketjun muodostamisen periaate. Matkaosuuksista j koostuu valmis vaihtomatkaketju J.*

Prosessia jatketaan, kunnes matkoja ei aikaikkunassa enää löydy. Ketjun viimeinen matka lisätään nousupysäkistä eteenpäin sellaisenaan ketjuun, koska poistumispysäkkiä ei voida päätellä. Mikäli kyseessä ei ole päivän ensimmäinen matka, eikä nousu ole ensimmäistä matkaa seuraavassa aikaikkunassa, aloitetaan uusi vaihtomatkaketju ja menetellään kuten päivän ensimmäisen ketjun tapauksessa.

Tarkastelemalla saman tunnisteen seuraavan vaihtomatkaketjun ensimmäistä nousupysäkkiä, oltaisiin voitu tehdä päätelmiä myös viimeisen reittiosuuden poistumispysäkistä. Niille matkoille, joille tulevaa tapahtumaa ei samalla tunnisteella löydy, jouduttaisiin kuitenkin lisäämään viimeinen reittiosuus kokonaisuudessaan. Selkeyden vuoksi tätä tarkastelua ei tässä työssä tehty.

Varsinainen toteutus tehtiin selkeämmän ohjelmakoodin vuoksi kahdessa osassa. Ketjun ensimmäisen tunnistetun matkan yhteydessä generoitiin ketjulle yksilöllinen tunniste, joka annettiin kaikille ketjun matkoille. Nousutapahtumarivien pysäkkisarjatunnisteen perusteella haettiin liikennöintiaineistosta vuoron pysäkkisarjat, ja vaihtomatkaketjun pysäkkisarja rakennettiin tapahtumien nousupysäkin perusteella.

Valmis ketju lisätään Python-dictionaryyn, eli hajautustaulutyyppiseen tietorakenteeseen, johon tallennetaan myös metatietoa matkoista. Ohjelmassa 4.1 on tästä esimerkki. Dictionary on suhteellisen vaivaton tallentaa ja edelleen hyödyntää yleisesti tunnetussa JSON-muodossa. Tällöin tuotantokäytössä data voidaan tallentaa esimerkiksi NoSQL-tietokantaan.

Tässä työssä metatietona tallennettiin ketjun aloitustapahtuman päivämäärä, käytetyt linjatunnukset sekä nousu- ja oletetut poistumispysäkit. Dataa tarkastellessa huomattiin, että jos asiakas on epähuomiossa tai tarkoituksella leimannut matkakorttiaan useamman kerran nousupysäkillä, muodostuu siitä vaihtomatkaketju, vaikkei vaihtoa olisikaan myöhemmin tehty. Ongelma on ratkaistu poistamalla aineistosta ne matkat, joissa on vähemmän kuin kaksi uniikkia nousupysäkkiä, sekä matkat, joille ei ole voitu päätellä yhtään poistumispysäkkiä.

```
'12':
{'journey_id': '070322-12',
 'enter_stops': ['280', '6016'],
 'exit_stops': ['219'],
 'lines': ['7', '42'],
 'pattern': ['6016', '6014', '6012', '6126', '6010', '6008',
 '6006', '6004', '6002', '6128', '792', '793', '795', '467',
 '468', '469', '470', '162', '163', '219', '280', '114',
 '866', '867', '869', '870', '1669', '871', '872', '873',
 '874', '876', '1338', '879', '880', '881', '645', '883',
 '884', '885', '886', '887', '888', '1664']}
```

Ohjelma 4.1. *Esimerkki vaihtomatkaketjun dictionary-rakenteesta.*

4.3 Samankaltaisten matkaketjujen klusterointi

Työssä noudatetaan Theodoridiksen ja Koutroumbasin esittelemää perusvaiheistusta klusterointitehtävän suorittamiseksi [30]:

Ominaisuusvalintana käytetään pysäkkijoukkoja. Vaihtoehtoisesti samankaltaisuutta voisi arvioida esimerkiksi sijaintikoordinaattien perusteella. Maantieteelliset esteet, kuten vesistöt, tai infrastruktuuriset esteet, kuten moottoritiet, voivat tehdä kahdesta koordinaateiltaan varsin samankaltaisesta joukosta joukkoliikenteen näkökulmasta täysin erilaiset. Kuten aiemminkin on tullut ilmi, uniikki pysäkkitunniste kapseloi aina myös sijaintikoordinaatit. Aineisto esikäsitellään siten, että data on mahdollisimman yksiselitteistä.

Läheisyysmittarina käytetään Jaccard-indeksiä. Koska matkaketjut koostuvat selkeistä pysäkkitunnisteista, joilla kullakin on itsenäiset ominaisuudet, eikä tunnisteen merkkisisällön perusteella voida tehdä oletuksia, soveltuvat joukkovertailumenetelmät hyvin matkaketjujen vertailuun. Jaccard-samankaltaisuus ottaa esimerkiksi Dicen kerrointa huonommin huomioon eripituisten matkaketjujen samankaltaisuuden. Tämä voidaan katsoa tässä työssä mittarin eduksi, sillä lyhyet reittiosuudet tulevat joka tapauksessa esiin, kun samankaltaisuus on vahva saman suuruusluokan ketjuissa.

Klusterointikriteerinä käytetään joukkojen samankaltaisuutta.

Klusterointialgoritmina käytetään LSH-indeksiä, jonka voidaan katsoa kuuluvan Locality-Sensitive Hashing -algoritmiperheeseen. LSH on yleisesti käytetty [24, 26, 27] menetelmä yhdessä minhash-algoritmin kanssa. Myös vaihtelevien kokoisten data-aineistojen käsittelyyn soveltuvaa, ja perinteisiä LSH-menetelmiä tehokkaammaksi todettua [29] LSH-metsä -algoritmia arvioidaan. Jaccard-samankaltaisuuden laskentaa tehostetaan minhash-algoritmilla, jonka lopputuotteena saatavan tiivistematriisin perusteella klusterointi suoritetaan.

4.3.1 Aineiston valmistelu

Tässä työssä saman matkaketjun sisällä olevilla vierekkäisillä pysäkeillä ei ole semanttista yhteyttä, joten unigram-tarkkuus on riittävä. Päreenä käytetään siis pysäkkitunnusta. Data on alussa JSON-muotoisissa dictionary-rakenteissa, joten aineisto muodostetaan keräämällä *pattern*-kentissä olevat pysäkkijoukot listaksi.

Alustavassa tarkastelussa huomattiin, että moni vaihtomatka on ns. asiointikäynti, jossa ketjussa on kaksi samaa reittiä kuljettua erisuuntaista matkaa. Nämä tapaukset ovat työn kannalta epäoleellisia, joten listalle kerättiin pysäkkijoukot ainoastaan niistä objekteista, joissa käytettyjen linjojen joukon koko on suurempi kuin yksi. Joissain tapauksissa samaa reittiosuutta ajavat eri linjaosuudet ja asiointikäynnit jäävät tällöin mukaan aineistoon.

Nousu- ja vaihtotapahtumien määrässä tulee huomioida ryhmätuotteet, kuten opettajien luotolliset kortit. Tällöin opettaja voi leimata kokonaisen koululuokan samalla pysäkillä ja vaihtotapahtumien määrä tässä yhteydessä kasvaa leimatun ryhmän verran suuremmaksi kuin mitä kortilla todennäköisesti on tehty. Tästä syystä skripteissä on käytetty vaihtomatkaketjussa olevien nousujen määränä uniikkien nousupysäkkien lukumäärää ja vaihtomatkojen määränä nousupysäkkien lukumäärää vähennettynä yhdellä. Saman nousupysäkin toistuva käyttö yhdessä vaihtomatkaketjussa on harvinaista.

4.3.2 Toteutus

Työssä käytetään Datasketch-kirjastoa [28] locality-sensitive hashing -toteutukseen. Aineisto tiivistetään tiivistematriisiksi minhash-funktiolla ja klusterointi suoritetaan indeksoimalla matriisi LSH-indeksiin.

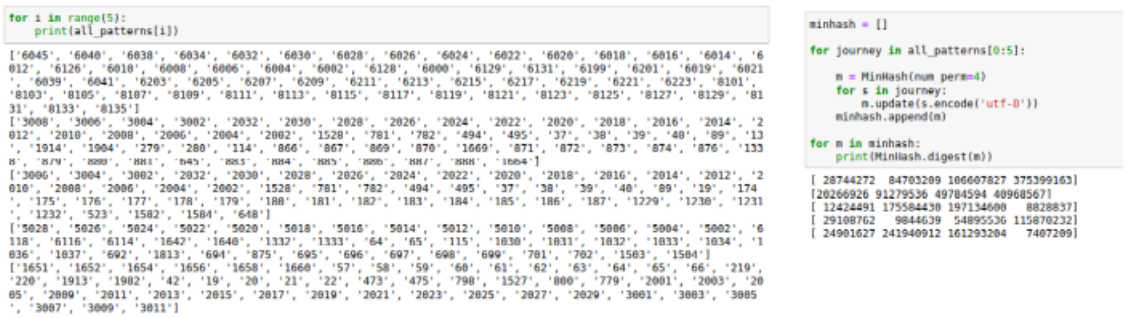
Minhash-funktiolle annetaan parametrina haluttu sekoitusfunktioiden toistomäärä, mahdollisten satunnaisten sekoitusfunktiojoukkojen ohjausparametri sekä haluttu sekoitukseen käytettävä hajautusfunktio.

Sekoitusfunktioiden toistomäärän kasvattaminen parantaa lopputuloksen tarkkuutta. Koska lopputuotteena syntyvässä m x n -matriisissa n on suoritettavien sekoitusfunktioiden toistomäärä, kasvattaa se myös lopputuotteena muodostettavan matriisin kokoa. Tämä on verrannollinen myös vaadittavaan laskentatehoon.

Toistomäärän parametreina testattiin arvoja 8, 16, 32, 64, 128, 256 ja 512. Huomattiin että suoritusaika kasvaa lineaarisesti toistomäärän kasvaessa. Klustereiden suuruudet olivat pienemmillä toistomäärillä samaa suuruusluokkaa, mutta klustereiden sisäistä pysäkkihajontaa oli huomattavasti enemmän, eikä lopputulos näin ollen ollut käyttökelpoinen. Myös duplikaattiklustereita syntyi pienemmillä arvoilla enemmän. Tutkimuksessa päädyttiin käyttämään oletusarvoa 128, koska aikavaatimuksia läpimenoajalle ei ollut. Duplikaattiklusterit poistettiin aineistosta.

Datasketch käyttää oletusarvoisesti sekoitusfunktiona SHA-1 algoritmiin perustuvaa hajautusta, joka toteutetaan Pythonin hashlib-kirjastolla. Algoritmi ei perustu satunnaiseen sekoittamiseen, joten samalla datasyötteellä ja samoilla parametreilla saadaan lopputuloksena aina sama matriisi. Oletusparametrina annettava SHA-1 sopii tarkoitukseen hyvin, vaikka Kyberturvallisuuskeskuksen mukaan [31] varsinaisessa salauskäytössä sitä ei enää voi pitää tietoturvallisena. Minhash-funktion toimintaa on havainnollistettu kuvassa 4.3, jossa viisi matkaketjua tiivistetään.

Klusterointia varten vertailtiin perinteisempää LSH-indeksiin perustuvaa toteutusta, sekä LSH-metsä -algoritmia. Tavoitteena on samankaltaisten vaihtomatkaketjujen klusterointi, joten dynaaminen klusterikoko on toivottavaa. LSH-metsää käytetään lähtökohtaisesti suositusten eli lähimpien naapureiden löytämiseen, joten m naapurin löytämisen jälkeen



Kuva 4.3. *Minhash -funktion tuottama 5x4 matriisi, kun syötteenä on 5 vaihtomatkaketjua ja suoritetaan neljä sekoitusfunktiota.*

tulee tehdä erillinen suodatusprosessi jollain etäisyysmittarilla. Suoritukseen vaaditaan myös käyttäjäsyötteenä haluttu naapureiden määrä M.

LSH-metsän etuina on mainittu tehokkuus ja mahdollisuus olla määrittelemättä hajautuksessa käytettävää k-arvoa. Kaksivaiheinen prosessi kasvattaa kuitenkin suoritusaikaa, eikä k-arvoa vaadita myöskään LSH-indeksin tapauksessa, joten työssä päädyttiin käyttämään suoraviivaisempaa LSH-indeksi -toteutusta.

Työssä käytettiin Datasketch MinHashLSH-kirjaston funktioita klustereiden luomiseen. Prosessi alkaa minhash-matriisin luomisella, jonka jälkeen rakennetaan LSH-indeksi. Indeksifunktio saa oleellisina parametreina Jaccard-indeksin kynnysarvon, sekoitusfunktioiden lukumäärän, painotussuhteen virheellisille negatiivisille ja virheellisille positiivisille arvoille, halutut LSH-parametrit (b ryhmää ja r riviä), sekä valinnaisen hajautusfunktion määrittelyn. Jokainen minhash-matriisin rivi lisätään tavukoodina LSH-indeksiin ja indeksinumerona annetaan minhash-matriisin rivinumero.

LSH-parametrien syöttäminen ohittaa Jaccard-indeksi -kynnysarvon ja painotusparametrin, joten niitä ei käytetty. Kynnysarvoksi sen sijaan valittiin 0,9, jolla saatiin riittävän tarkkoja tuloksia. Algoritmi laski b-arvoksi 5 ryhmää, jolloin r-arvoksi tuli 25 riviä. Virheellisten tulosten painotuksessa käytettiin oletusarvoa 0,5, jolloin painotus virheellisille negatiivisille ja virheellisille positiivisille tuloksille on sama.

Indeksin luomisen jälkeen iteroidaan minhash-matriisi ja vuorossa olevalla rivillä haetaan LSH-indeksistä hakuparametria vastaavat indeksinumerot, jotka vastaavat vaihtomatkaketjujen indeksejä. Algoritmi lisää vastauksena saatavat indeksinumerot kandidaattijoukkoon. Lopputuloksena saatu joukko viedään omaan hajautustauluun käyttäen avaimena minhash-indeksiä, joka vastaa syötedatan indeksiä. Näin klustereihin kerättyjen indeksien perusteella voidaan poimia todelliset vaihtomatkaketjut syötedatasta.

Lopputuloksessa ilmeni päällekkäisyyttä eli samojen vaihtomatkaketjujen esiintymistä useissa klustereissa. Tulosta voisi yksinkertaistaa jatkamalla prosessia esimerkiksi kokoamalla klustereita LSH-link-menetelmällä. Päällekkäisyyden ehkäiseminen edellyttäisi tässä-

kin tapauksessa vaihtomatkaketjujen tai klustereiden merkitsemistä niin, etteivät ne voisi päätyä toiseen klusteriin. Tämä lisäisi todennäköisyyttä sille, ettei kaksi samankaltaista vaihtomatkaketjua päätyisi samaan klusteriin. Kokoaminen vaatisi myös halutun klusterien määrän antamisen käyttäjäsyötteenä. Näistä syistä prosessia ei tässä sovelluksessa jatkettu.

5. TULOKSET JA POHDINTAA

Tässä luvussa esitellään toteutetun algoritmin, sekä valittujen menetelmien kanssa saadut tulokset. Toteutettu algoritmi tarjoaa oivallisen näkyvyyden vaihtomatkoihin, joten ensin tarkastellaan vaihtomatkaketjujen ominaisuuksia. Sen jälkeen arvioidaan valittujen menetelmien tehokkuutta ja lopuksi analysoidaan vaihtomatkaklustereiden sisältöä.

5.1 Vaihtomatkaketjuaineiston tarkastelu

Taulukossa 5.1 on esitetty löydettyjen vaihtotapahtumien lukumäärä suhteessa vastaavan aikajakson kokonaismatkamäärään. Kokonaismatkamäärä on poimittu Turun seudun joukkoliikenteen matkamäärätilastosta.

Vaihtotapahtumalla tarkoitetaan muita kahden tunnin aikaikkunassa tehtyjä leimauksia, kuin ensimmäistä leimausta. Vaihtomatkojen osuus on hieman suurempi arkipäivinä, kuin viikonloppuna.

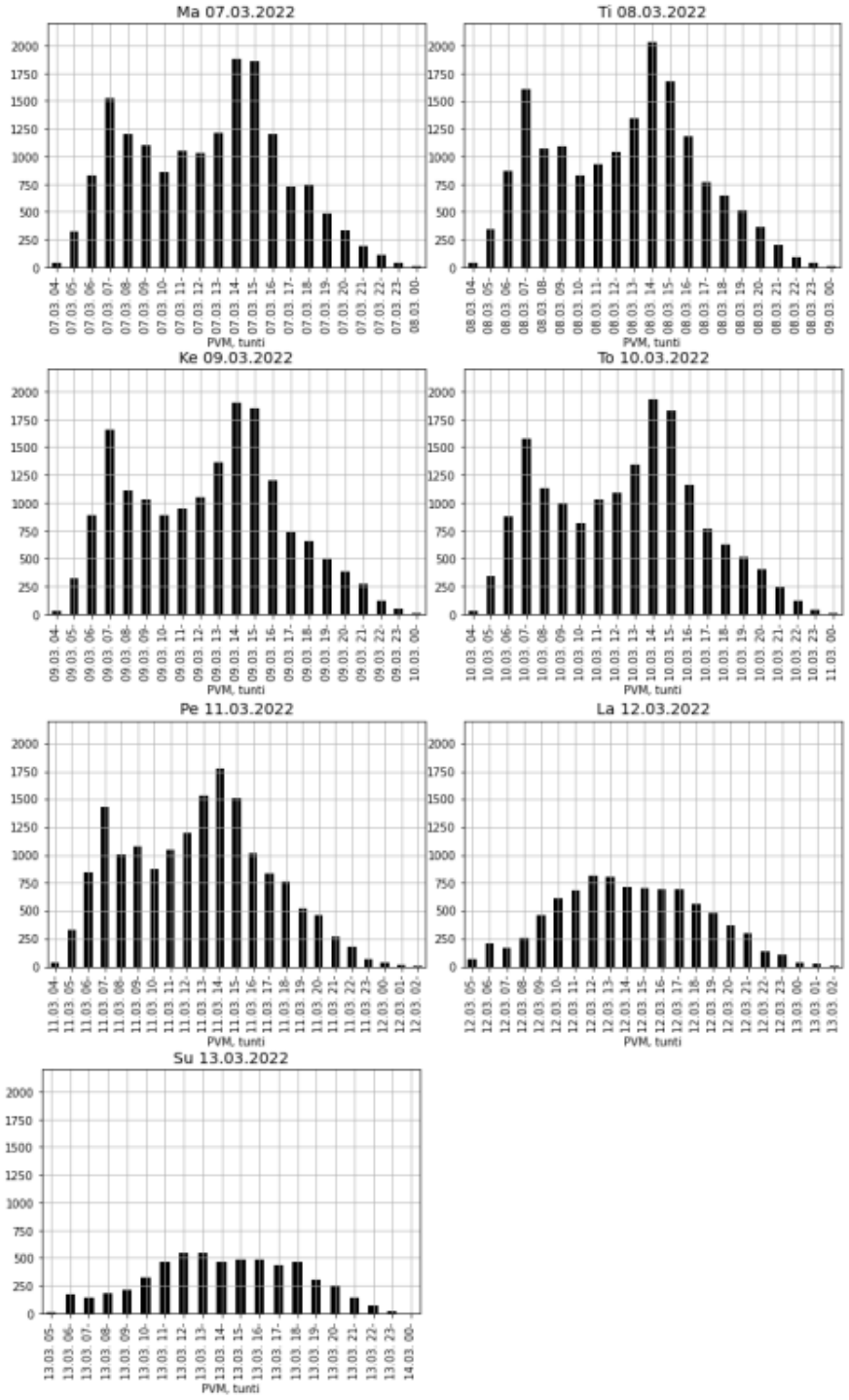
Taulukko 5.1. *Vaihtotapahtumien määrä suhteessa kaikkiin tapahtumiin.*

Pvm	*Vaihtoja*	*Kokonaisnousut*	*Suhde*
07.03.22	19333 kpl	78821 kpl	24,53 %
08.03.22	19120 kpl	80205 kpl	23,84 %
09.03.22	19440 kpl	80540 kpl	24,14 %
10.03.22	19289 kpl	80765 kpl	23,88 %
11.03.22	19543 kpl	82108 kpl	23,80 %
12.03.22	9758 kpl	44122 kpl	22,12 %
13.03.22	6189 kpl	26464 kpl	23,39 %
Koko vko	112672 kpl	473025 kpl	23,82 %

Kuvassa 5.1 on esitetty vaihtomatkaketjujen ajallista jakaantumista niiden aloitusmäärinä tunneittain. Jakaumien muoto ei arkipäivien tapauksessa juurikaan poikkea eri viikonpäivien kohdalla ja mukailee tyypillistä matkamääräjakaumaa.

Eniten vaihtomatkaketjuja aloitetaan aikavälillä 14-15, josta voi päätellä useiden koulu-

matkojen sisältävän vaihtoyhteyksiä. Vaihdollisten matkojen määrä pysyy korkeana edelleen kello 17:ään asti, jonka jälkeen ne vähenevät vuorokauden päättymistä lähestyttäessä. Keskipäivällä vaihtomatkoja tehdään enemmän kuin illalla. Viikonloppuna vastaavia piikkejä ei nähdä, vaan vaihtomatkat ajoittuvat suhteellisen tasaisesti ns. valoisaan aikaan.



Kuva 5.1. *Vaihtomatkaketjujen alkamisaikojen jakauma tunneittain ja viikonpäivittäin.*

5.1.1 Nousutapahtumien määrä matkaketjussa

Taulukossa 5.2 on esitetty yksittäisten vaihtomatkaketjujen sisältämien nousutapahtumien lukumääriä. Odotetusti kahden nousutapahtuman ketjuja on selkeä enemmistö. Tämä on odotettua, koska Turun linjasto perustuu kirjoitushetkellä yhteen vaihtoterminaaliin ja sen kautta kulkeviin heilurilinjoihin. Kolmen nousutapahtuman ketjuja on alle kymmenen prosenttia, mutta määrä on suhteellisen merkityksellinen, sillä kolmen matkan sovittaminen kahden tunnin aikaikkunaan viittaa suunnitelmalliseen joukkoliikenteen käyttöön.

Taulukko 5.2. *Nousutapahtumien määrä vaihtomatkaketjuissa.*

Nousuja	*Vaihtomatkaketjuja*	*Suhde*
2 kpl	84963 kpl	87,31 %
3 kpl	9676 kpl	9,94 %
4 kpl	2395 kpl	2,46 %
5 kpl	237 kpl	0,24 %
6 kpl	38 kpl	0,04 %
7 kpl	3 kpl	0,003 %
9 kpl	2 kpl	0,002 %

Neljän nousutapahtuman matkoihin liittyvissä linjaryhmissä korostuvat Raision sisäiset R-linjat, kuten taulukosta 5.3 voi huomata. R-linjaa voi käyttää liityntälinjana Turkuun suuntautuvilla päälinjoilla (6, 7, 7A), josta aikaikkunan puitteissa voi vaihtaa edelleen seuraaville linjoille. Useampien nousujen ketjut ovat harvinaisia ja niissä linjaryhmien hajontaa on runsaasti.

Taulukko 5.3. *Usein toistuvat linjaryhmät neljän nousutapahtuman vaihtomatkaketjuissa.*

Käytetyt linjat	*Määrä*
’R2’, ’7A’, ’32’, ’42’	25
’R2’, ’300’, ’300’, ’7’	21
’R1’, ’R1’, ’7A’, ’801’	11
’R1’, ’R1’, ’6’, ’61’	10
’R1’, ’7A’, ’18’, ’801’	9
’R1’, ’7’, ’6’, ’220’	9
’R1’, ’7’, ’18’, ’18’	8

5.1.2 Suosituimmat vaihtoalueet

Vaihtomatkaketjun viimeinen reittiosuus lisätään ketjuun kokonaisuudessaan, josta johtuen jokaista laskettua poistumista on seurannut uusi nousu. Vaihtoalueina käsitetään

tästä syystä vaihtomatkaketjun laskennalliset poistumispysäkit. Koko aineiston 20 suosituinta poistumispysäkkiä on listattu taulukossa 5.4. Lähes kaikki suosituimmat vaihtoalueet sijoittuvat Turun keskusterminaalina toimivan Kauppatorin ympäristöön.

Taulukko 5.4. *Suosituimmat poistumispysäkit tunnistetuissa vaihtomatkaketjuissa.*

Pysäkkitunnus	*Nimi*	*Poistumisia*
1983	Keskusta	5767
1904	Kauppatori	4730
1903	Kauppatori	4428
1984	Keskusta	4091
1959	Puutori	3683
1913	Keskusta	3394
1942	Keskusta	2855
1914	Keskusta	2845
13	Puutori	2664
1941	Keskusta	2567
1985	Keskusta	2501
42	Puutori	2365
1911	Keskusta	2132
1986	Keskusta	2123
68	Brahenkatu	2022
219	Kåren	2019
1901	Kauppatori	1978
220	Kaskenkatu	1513
1912	Keskusta	1490
1989	Keskusta	1458

Taulukkoon 5.5 on listattu suosituimmat vaihtopysäkit, kun *19*-alkuiset keskustan pysäkit on jätetty pois. Suosituimmat pysäkit ovat edelleen keskustaa ja Kauppatoria lähellä sijaitsevia pysäkkejä.

Keskustaa lähellä sijaitseva Autistenaukio tarjoaa vaihtomahdollisuuden Satakunnantietä ja Tampereentietä kulkevien linjojen välillä. Tällöin vaihdon voi tehokkaasti suorittaa ilman käyntiä keskustassa. Pysäkki *203* on poistumispysäkki Tampereentietä saavuttaessa ja *40* Satakunnantietä saavuttaessa. Pysäkki *219 Kåren* tarjoaa saman mahdollisuuden Hämeenkadun ja Uudenmaankadun linjojen välillä, jossa molemmista suunnista saavuttaessa poistutaan pysäkillä *219*.

Selkeästi keskustan ulkopuolelta listalle nousevat Itäkeskuksen pysäkit *645* ja *644*, jotka

sijaitsevat Varissuon kauppakeskuksen alla. Kahden tunnin aikaikkunassa on mahdollista asioida kauppakeskuksessa ja jatkaa matkaa samalta pysäkiltä. Aineiston muodostamisen jälkeen Itäkeskuksen tarjontaa on lisätty kahdella linjalla, joka on saattanut kasvattaa Itäkeskuksen roolia myös linjojen välisenä vaihtopaikkana. Toisaalta, pysäkki tarjoaa vaihtoyhteyden kehälinjalle, kuten myös Länsikeskuksessa sijaitseva pysäkki *839 Viilarinkatu*.

Pysäkki *79 Turun Satama (Viking)* on Sataman poistumispysäkki ja sen nouseminen listaan on yllättävää, sillä esimerkiksi laivamatkustajien ei voida olettaa jatkavan matkaa aikaikkunassa Satamasta pois päin. Ilmiön selittänevät laivasiivoojat, jotka ehtivät tekemään paluumatkan leimauksen kahden tunnin sisällä.

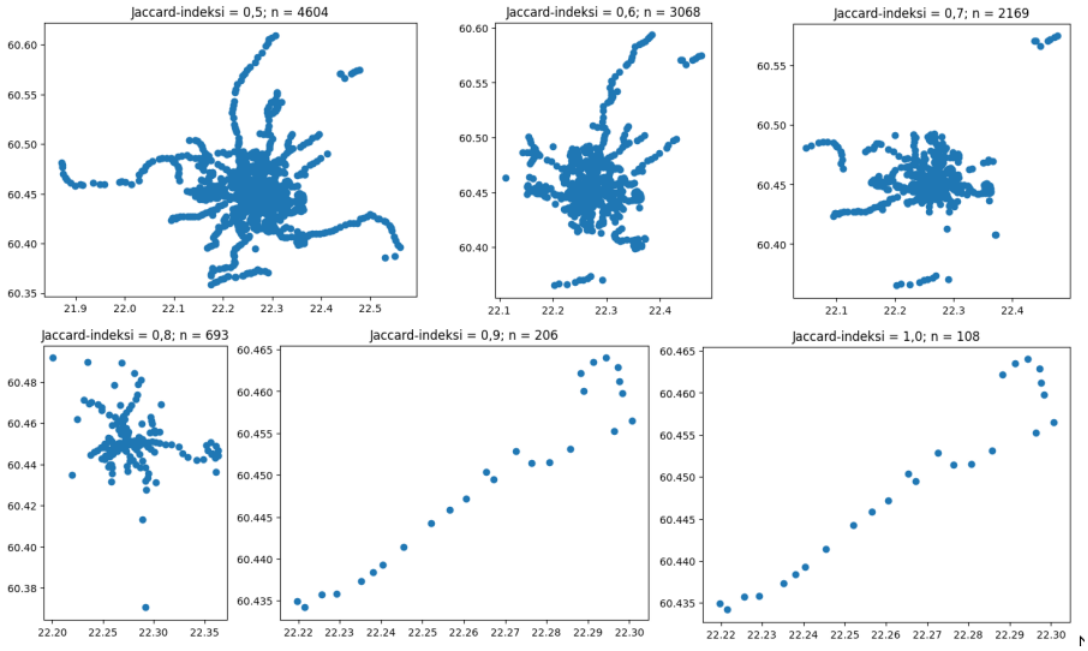
Taulukko 5.5. *Suosituimmat poistumispysäkit tunnistetuissa vaihtomatkaketjuissa, kun keskustan, Kauppatorin ja Puutorin pysäkit on poistettu tuloksista.*

Pysäkkitunnus	*Nimi*	*Poistumisia*
68	Brahenkatu	2022
219	Kåren	2019
220	Kaskenkatu	1513
109	Posti	1446
89	Turun linja-autoasema	1441
65	Yliopisto (TYKS U-sairaala)	1168
164	Tuomiokirkkotori	1057
203	Autistenaukio	923
19	Turun linja-autoasema	905
280	Kåren	865
16	Brahenkatu	862
66	Akatemiantalo	738
645	Itäkeskus	653
40	Autistenaukio	623
148	Posti	595
79	Turun Satama (Viking)	567
839	Viilarinkatu	480
115	Yliopisto (TYKS U-sairaala)	469
467	Kivikartiontie	466
644	Itäkeskus	454

5.2 Valitun menetelmän parametrit ja ominaisuudet

Haluttu samankaltaisuuden kynnysarvo asetettiin Jaccard-indeksinä. Kuvassa 5.2 on esitetty suuriman klusterin pysäkkijoukon koordinaatit eri kynnysarvoilla toteutettuna. Klusterin koko on ilmoitettu muuttujassa n.

Kuvista nähdään selvästi, että vielä kynnysarvolla 0,8 klusteriin ajautuu selvästi keskenään eri suuntaisia reittimuotoja. Kynnysarvojen 0,9 ja 1,0 välillä ei ole merkittävää eroa. Kynnysarvolla 1,0 klusteriin tulee kuitenkin vain täysin samanlaisia ketjuja ja tässä työssä tarkoituksenmukaista on löytää myös samankaltaiset saman muodon omaavat ketjut.

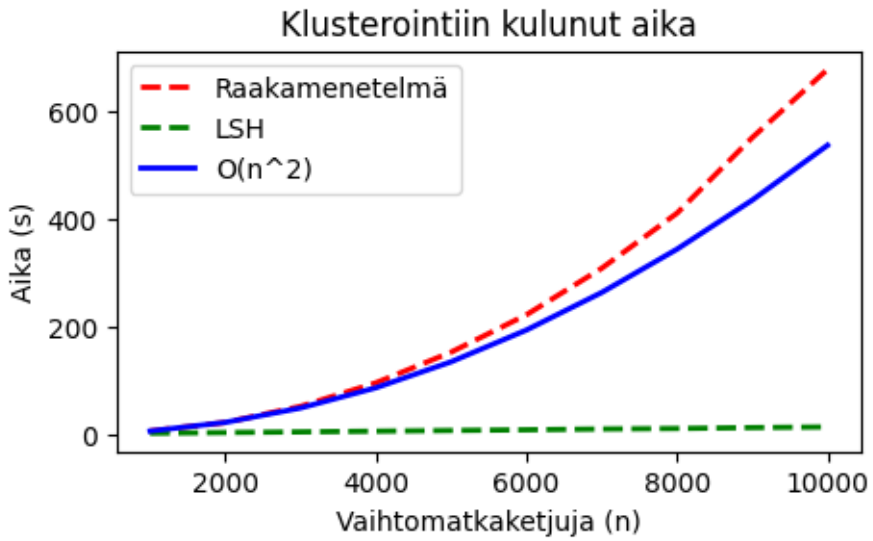


Kuva 5.2. *Jaccard-indeksin vaikutus lopputulokseen.*

Nopeusvertailua varten toteutettiin raakamenetelmä (engl. *brute force*) vertailemalla jokaista vaihtomatkaketjua iteroimalla ne sisäkkäisissä toistolauseissa. Sisäkkäisen iteraation toistovuorossa oleva vaihtomatkaketju lisättiin ulomman iteraation vaihtomatkaketjua vastaavaan klusteriin, jos niiden Jaccard-indeksi ylitti kynnysarvon 0,9. Koko vaihtomatkaketjuaineiston (*n = 81832*) käsittelyyn kului raakamenetelmällä 32859,55 sekuntia. Vastaavan aineiston käsittelyyn LSH-menetelmällä kului 93,19 sekuntia.

Nopeusvertailun kuvaajaesitys on kuvassa 5.3. Varsinainen vertailu toteutettiin pienemmillä aineistoilla valitsemalla koko aineiston alusta kymmenen otosta kokovälillä *n = 1000* ja *n = 10000*. Testiaineisto sisältää siis ainoastaan maanantain 7.3.2022 ajalta tunnistettuja vaihtelevan mittaisia vaihtomatkaketjuja. Jokaisesta otoksesta muodostettiin klusterit käyttäen sekä raakamenetelmää että LSH-menetelmää. Kulunut aika on Pythonin datetime-kirjastolla tallennettujen lopetus- ja aloitusaikojen erotus. Sininen kuvaaja on

vertailukäyrä aikakompleksisuudelle $O(n^2)$.



Kuva 5.3. *Klusterointiin kuluneen ajan vertailu raakamenetelmällä sekä Locality Sensitive Hashing -menetelmällä.*

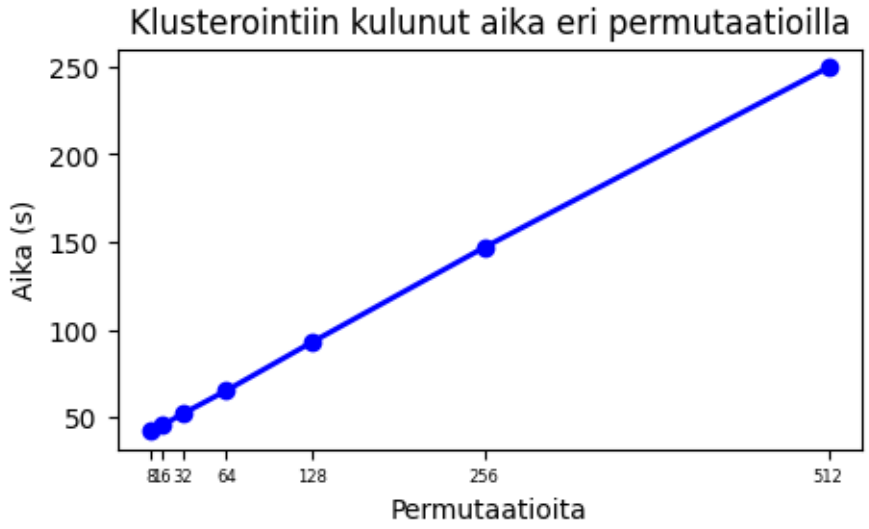
Kuvan 5.3 ajat on taulukoituna taulukossa 5.6. Luvuista huomataan, että LSH-menetelmällä muutos kuluneessa ajassa on lähes suoraan verrannollinen muutokseen aineiston koossa. Tämä tukee kirjallisuudessa esitettyä [23, 24] LSH:n aikakompleksisuuskaavaa $O(n)$, jossa n on aineiston koko.

Raakamenetelmällä klusteroinnille on kirjallisuudessa esitetty aikakompleksisuuskaava $O(n^2)$. Vertailussa kulunut aika alkaa merkittävästi eroamaan tästä noin *n = 3000* kohdalla. Kasvavaan eroon vaikuttanevat vaihtomatkaketjujen vaihtelevat koot.

Taulukko 5.6. *Klusterointiin kulunut aika eri menetelmillä eri kokoisilla aineistoilla.*

Aineiston koko	*Raakamenetelmä*	*LSH*
1000	5,40 s	1,31 s
2000	21,43 s	2,61 s
3000	51,44 s	3,83 s
4000	95,28 s	5,11 s
5000	152,51 s	6,47 s
6000	222,19 s	7,74 s
7000	308,93 s	9,18 s
8000	411,58 s	10,21 s
9000	552,63 s	11,79 s
10000	679,70 s	13,00 s

Permutaatioiden eli käytettyjen hajautusfunktioiden lukumäärän vaikutus aikaan on esitetty kuvassa 5.4. Prosessiin kuluva aika kasvaa lineaarisesti suhteessa suoritettaviin permutaatioihin. Tulokset on taulukoitu taulukossa 5.7.



Kuva 5.4. *Klusterointiin kuluneen ajan vertailu eri permutaatioilla.*

Taulukko 5.7. *Klusterointiin kulunut aika eri permutaatioilla.*

Permutaatioita	*Aika*
512	249,51 s
256	147,04 s
128	93,19 s
64	65,56 s
32	52,51 s
16	45,64 s
8	42,57 s

5.3 Samankaltaiset vaihtomatkaketjut

Taulukossa 5.8 on sadan suurimman klusterin joukosta poimitut uniikit reittikuvaukset raakamenetelmällä sekä LSH-menetelmällä. Taulukossa on lihavoitu ne muodot, jotka löytyvät molemmista tuloksista. LSH-menetelmä onnistui löytämään Varissuo-Satama-reittiä lukuunottamatta kaikki raakamenetelmällä löydetyt reitit. Reittimuodot on todettu visuaalisesti asettamalla klusterin pysäkkijoukko karttapohjalle.

Taulukko 5.8. *Uniikit reittikuvaukset sadan suurimman klusterin joukosta.*

Raakamenetelmä			*Locality-sensitive Hashing*		
Järj.	*Koko*	*Kuvaus*	*Järj.*	*Koko*	*Kuvaus*
1	246	**Runosmäki-Varissuo**	1	206	**YO-kylä-Satama**
5	206	**YO-kylä-Satama**	2	189	**YO-kylä-Keskusta-YO-kylä**
8	185	**Länsinummi-Harittu**	3	187	**Runosmäki-Varissuo**
9	185	**Varissuo-Keskusta-Varissuo**	4	179	**Räntämäki-Keskusta-Räntämäki**
10	184	**YO-kylä-Keskusta-YO-kylä**	5	179	**Satama-Varissuo**
30	154	**L:nummi-Keskusta-L:nummi**	12	169	**Länsinummi-Harittu**
36	148	**Satama-Varissuo**	20	154	**Keskusta-Varissuo**
39	145	Varissuo-Satama	24	148	**Keskusta-Naantali**
56	135	**Räntämäki-Keskusta-Räntämäki**	27	144	Kastu-Varissuo
57	135	**Runosmäki-Satama**	34	141	**Varissuo-Keskusta-Varissuo**
58	135	**Varissuo-Runosmäki**	45	133	**Kohmo-Keskusta-Kohmo (Littoinen)**
62	128	**Varissuo-Itäkeskus-Varissuo**	47	132	**Runosmäki-Naantali**
63	127	**Keskusta-Naantali**	54	124	Mäntymäki/Kurjenmäki-Naantali
70	123	**Kohmo-Keskusta-Kohmo (Littoinen)**	63	114	**Runosmäki-Satama**
75	123	**Runosmäki-Naantali**	71	109	**Varissuo-Runosmäki**
81	121	**Runosmäki-Kohmo (Littoinen)**	74	108	Varissuo-Naantali
85	120	**Keskusta-Varissuo**	76	108	**Varissuo-Itäkeskus-Varissuo**
100	117	**Runosmäki-Perno**	82	107	Runosmäki-Pansio
			84	106	**Runosmäki-Kohmo (Littoinen)**
			87	106	Harittu-Naantali
			91	105	**Runosmäki-Perno**
			96	105	Satama-Vaala
			99	104	Satama-Harittu
			100	104	**L:nummi-Keskusta-L:nummi**

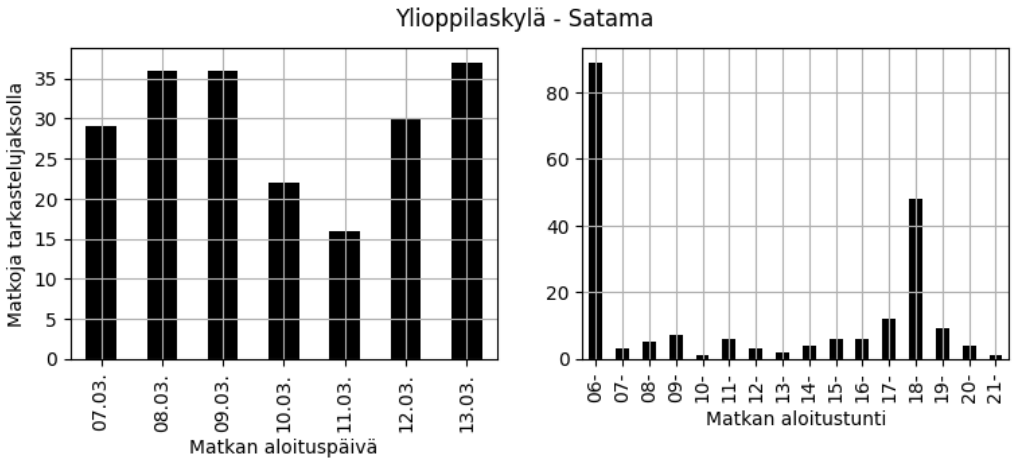
Vaikka aineistosta on poistettu duplikaatti-, eli keskenään täysin identtiset klusterit, nousevat samat reittimuodot useasti suurimpienkin klustereiden listalle. Koska tavoitteena ei ole ryhmitellä vain täysin samanlaisia vaihtomatkaketjuja, kerätään esimerkiksi klusteriin A yhtä pysäkkiä vertailtavaa matkaa a aiemmin aloitettu matka b ja vastaavasti yhtä pysäkkiä myöhemmin aloitettu matka c. Kun samankaltaisuutta vertaillaan c:n mukaan, otetaan klusteriin C matkat a ja b, sekä edelleen c:tä seuraavalta pysäkiltä aloitettu matka. Näin matkan reitti on lähes sama, mutta klustereiden A ja C sisältö on toisistaan poikkeava.

Ilmiö johtuu siis lähimpien naapureiden löytämisestä syntyvästä päällekkäisyydestä. Klusterien lukumäärää voisi rajoittaa jatkamalla hierarkista klusterointia kokoojamenetelmällä, mutta päällekkäisyys ei katoa, ellei klusteriin liitettyjä alkioita estetä liittymästä muihin klustereihin.

Seuraavassa analysoidaan LSH-menetelmällä löydettyjen suurimpien klustereiden uniikkeja reittejä. Koordinaattiesitysten pohjakarttana käytetään OpenStreetMap-aineistoa.

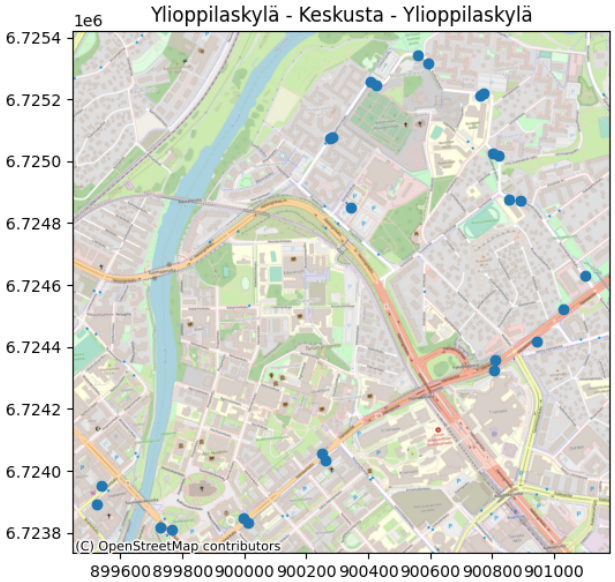
Ylioppilaskylä-Satama, Ylioppilaskylä-Keskusta-Ylioppilaskylä, Räntämäki-Keskusta-Räntämäki

Suurin klusteri on reittimuoto Ylioppilaskylästä Satamaan. Ylioppilaskylä on turkulainen pääasiassa opiskelija-asuntoja käsittävä lähiö. Matkojen jakaantumista viikonpäiville ja eri tunneille on kuvattu kuvassa 5.5. Koska matkojen aloitukset painottuvat erityisesti aamuaikaan, eikä viikonpäiväpainotusta ole esimerkiksi lauantaille, voidaan kysynnän katsoa olevan säännöllistä. Asiakasryhmänä voivat olla laivojen rantasiivoojat. Havainto voi nostaa esiin tarpeen vaihdottoman nopean yhteyden perustamiseksi Ylioppilaskylän ja Sataman välille työmatkaliikenteen palvelutason parantamiseksi.



Kuva 5.5. *Ylioppilaskylä-Satama-matkojen jakaantuminen viikonpäiville ja tunneille.*

Ylioppilaskylä-Keskusta-Ylioppilaskylä-reitin edestakaiset matkat nousevat klusteriin, koska Ylioppilaskylä-Keskusta-reittiosuutta ajetaan säännöllisesti neljällä eri (50, 51, 53, 54) linjatunnuksella. Meno- ja paluumatkat voi näin tehdä eri linjatunnuksilla, eikä samojen linjatunnusten filtteröinti poista niitä. Matkat jakaantuvat normaalin kysynnän mukaan valoisalle ajalle ja muodostaen huiput ruuhka-aikoina. Kuvasta 5.6 huomataan, että matkojen vaihtoalue ei ole keskustan terminaalialueella, vaan pysäkkiä aikaisemmin. Alueella sijaitsee mm. Lidl Eerikinkatu, joka on opiskelijoiden suosiossa.



Kuva 5.6. *Ylioppilaskylä-Keskusta-Ylioppilaskylä-klusterin pysäkkijoukko kartalla.*

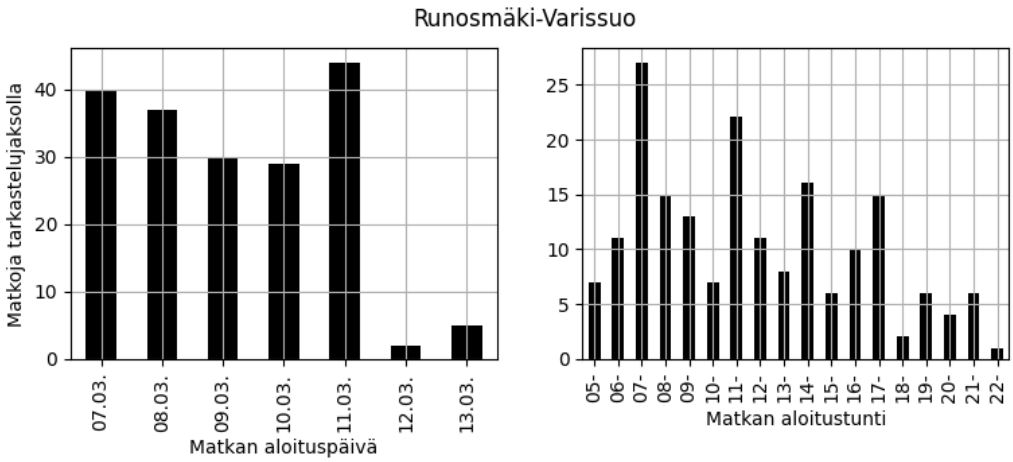
Räntämäki-Keskusta-Räntämäki on niin ikään linjojen 55, 55A, 56 jakama reittiosuus, jolloin ns. asiointikäynnit ovat aineistossa mukana. Reitti jakaa osuutta myös Ylioppilaskylästä lähtevien linjojen kanssa. Matkojen aloitusajat jakaantuvat normaalin kysynnän mukaan, painottuen kuitenkin hieman voimakkaammin kello yhdeksän tunnille.

Runosmäki-Varissuo, Varissuo-Runosmäki

Runosmäki ja Varissuo ovat Turun suurimmat lähiöt, eikä niiden välillä ole vaihdotonta yhteysvaihtoehtoa. Matkojen aloitusaikojen jakauma on esitetty kuvassa 5.7. Matkat painottuvat vahvasti arkipäiviin ja aloitusajoissa on huomattavissa muutama selkeämpi piikki. Lähiöiden välillä oli vuoteen 2014 asti vaihdoton ruuhka-aikana liikennöity linja, mutta se typistettiin Varissuon ja Keskustan välille. Toisaalta sekä Runosmäkeä palvelevan linjan 18 että Varissuon linjojen 32 ja 42 vuorovälit ovat erityisen tiheät, jolloin vaihtoyhteys on lähtökohtaisesti saumaton.

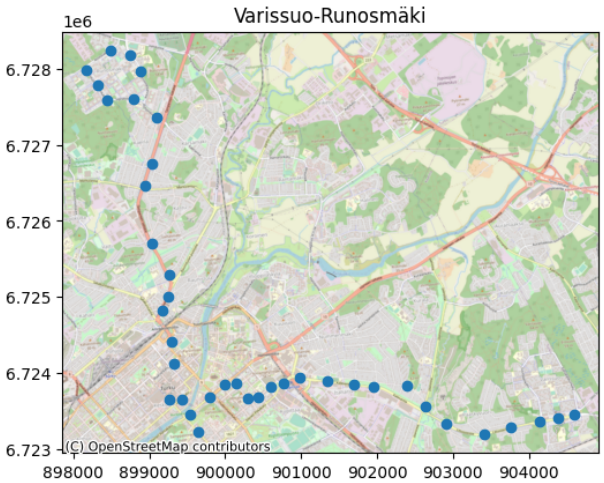
Keskusta-Varissuo-reittiosuus käsittää Varissuon lisäksi joitain suosittuja kohteita, kuten Kupittaan aseman, Kupittaan kampusalueen, TYKS T-sairaalan ja Itäharjun teollisuusalueen. On siis mahdollista, että suurikin osa klusterin matkoista päättyy todellisuudes-

sa ennen Varissuota, mutta reitin suurehkon pysäkkimäärän vuoksi ne saavat korkean Jaccard-indeksin.



Kuva 5.7. *Runosmäki-Varissuo -matkojen jakaantuminen viikonpäiville ja tunneille.*

Kuvassa 5.8 on vastakkaisen suunnan, Varissuo-Runosmäki -klusterin pysäkkijoukko. Ehkä osin edellämainitusta pohdinnasta johtuen Varissuo-Runosmäki-reitin suurimpaan klusteriin nousevat ainoastaan Itäkeskusta edeltävältä pysäkiltä ja sitä seuraavilta pysäkeiltä aloitettuja matkoja. Varissuolta Runosmäen suuntaan matkat jakautuvat hieman tasaisemmin myös viikonlopuille.

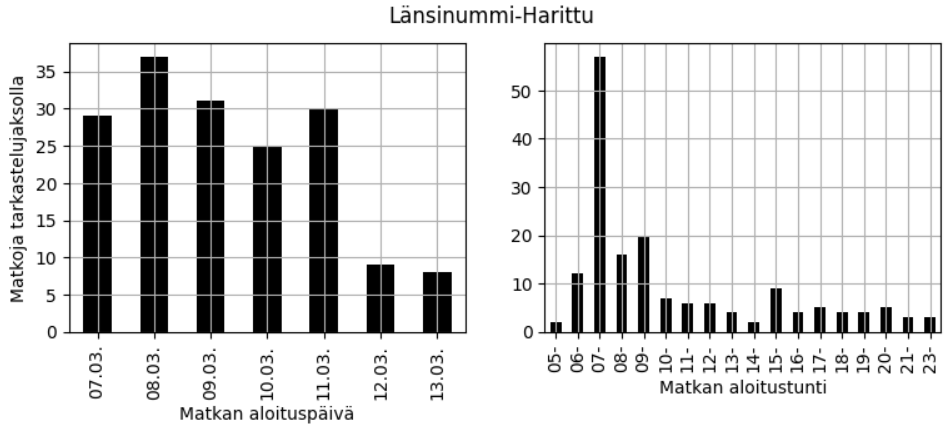


Kuva 5.8. *Varissuo-Runosmäki -klusterin pysäkkijoukko kartalla.*

Länsinummi-Harittu, Länsinummi-Keskusta-Länsinummi

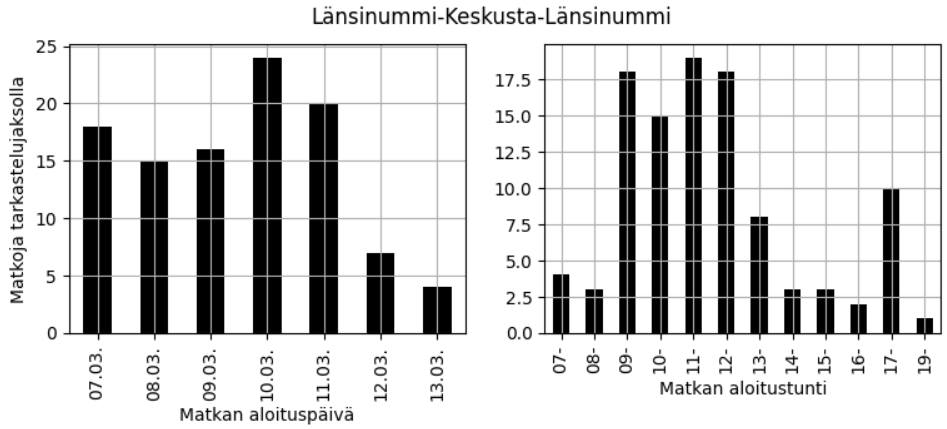
Länsinummi muodostaa yhtenäisen alueryhmittymän yhdessä Liljalaakson, Nättinummen ja Teräsrautelan kanssa. Länsinummen ja keskustan välisen osuuden varrelle jäävät myös

Hepokullan lähiö, Impivaaran liikuntapalvelut, sekä Satakunnantien varrella olevat palvelut. Kuten kuvasta 5.9 huomataan, Länsinummen ja Haritun välisen klusterin matkat painottuvat voimakkaasti arkiaamuihin muodostaen selkeän piikin kello seitsemän tunnille. Kaikki klusterin vaihdot on tehty linjalle 18. Keskustasta Harittuun kulkevan linjan 18 varrelle jäävät mm. TYKS Orto kirurginen sairaala sekä Turun Ammatti-instituutin tekniikan alan toimipiste, Peltolan koulutalo. Selkeästi suurin osa (141 kpl) vaihdoista on tehty laskennallisesti keskustan alueella, mutta myös Linja-autoasema nousee suosituksi (19 kpl) laskennalliseksi poistumispysäkiksi. Linja-autoasema on ensimmäinen linjojen yhteinen pysäkki, joten se mahdollistaa vaihdon ilman siirtymää.



Kuva 5.9. *Länsinummi-Harittu -matkojen jakaantuminen viikonpäiville ja tunneille.*

Länsinummi-Keskusta-Länsinummi-asiointimatkat jakautuvat selkeästi arkipäivien aamupäiviin. Jakauma on esitetty kuvassa 5.10. Länsinummen suunnasta liikennöidään neljällä eri linjatunnuksella siten, että linjat 2, 2B lähtevät Länsinummesta ja linjat 2A, 2C Liljalaaksosta. Reitti poikkeaa vain kahden pysäkit osalta.

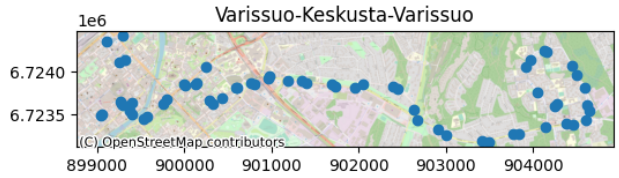


Kuva 5.10. *Länsinummi-Keskusta-Länsinummi-matkojen jakaantuminen viikonpäiville ja tunneille.*

Varissuo-Keskusta-Varissuo, Varissuo-Itäkeskus-Varissuo, Keskusta-Varissuo

Varissuon ja keskustan välillä ajetaan samaa reittiosuutta linjoilla 32 ja 42, sekä ruuhka-aikoina linjatunnuksella 92. Tämä selittää asiointimatkojen nousemisen listalle. Matkat jakaantuvat tasaisesti arkipäiville painottuen hieman perjantaille ja keskipäivän tunneille.

Kuvassa 5.11 nähdään, että joukossa on myös satunnaisia em. linjojen ulkopuolisia pysäkkejä, mutta valtaosalla matkoista (105 kpl) on silti käytetty ainoastaan linjoja 32 ja 42. Edestakaisen matkan pysäkkijoukko on niin pitkä, että lyhyiden keskusta-alueella tehtyjen vaihtomatkojen pysäkit eivät vaikuta matkan klusteriin päätymiseen.



Kuva 5.11. *Varissuo-Keskusta-Varissuo-klusterin pysäkkijoukko kartalla.*

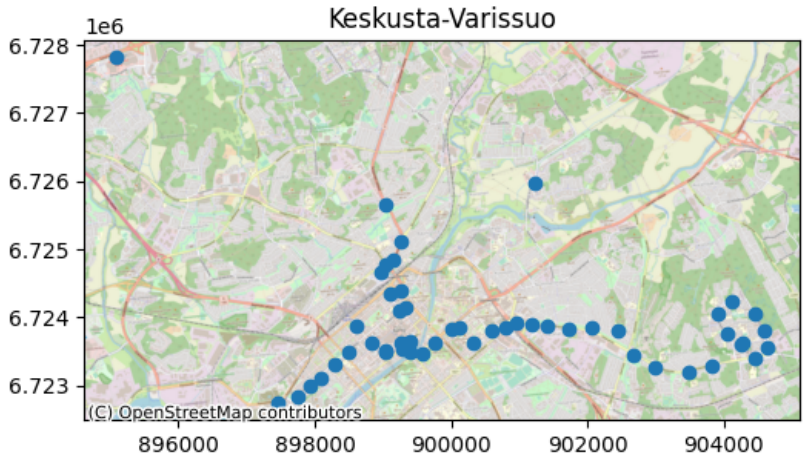
Kahden tunnin aikaikkuna mahdollistaa hyvin Varissuon sisäisten kauppamatkojen tekemisen. Kuten aiemmin mainittu, Varissuolle liikennöidään useammalla eri linjatunnuksella ja tästä syystä myös lyhyemmät edestakaiset matkat huomioidaan. Taulukossa 5.9 on statistiikkaa Varissuon sisäisten Itäkeskuksen asiointimatkojen nousupysäkeistä. Selkeästi suosituimpia ovat Varissuon pohjoisen alueen pysäkit 830, 831 ja 832, joiden palvelemalta alueelta on myös pisin matka ostoskeskukseen. Pysäkki 645 on Itäkeskus.

Taulukko 5.9. *Itäkeskukseen suuntautuneiden asiointimatkojen käytetyimmät nousupysäkkiryhmät.*

Matkaketjun nousupysäkit	*Määrä*
645, 832	44 kpl
645, 831	35 kpl
645, 830	25 kpl
645, 833	2 kpl
645, 828, 832	1 kpl
645, 831, 883	1 kpl

Keskusta-Varissuo-klusterin pysäkkijoukko on esitetty kuvassa 5.12. Matkojen alkuosuudet lähtevät keskusta-alueen reunoilta tai lähempää ydinkeskustaa. Joukossa on myös virheellisiä pisteitä, joista toinen kohdistuu Kauppakeskus Myllyyn, linjan 300 päätepisteelle. Ilmiö johtuu siitä, että nousutapahtuma on tehty päätepisteellä ennen, kuin kuljettaja on valinnut seuraavan lähdön suuntaa. Tästä syystä ensimmäiseen pysäkkisarjaan

rekisteröidään ainoastaan reitin päätepysäkki. Toinen Halisissa oleva virhepiste aiheutuu todennäköisimmin ajoneuvon paikannusongelmasta.



Kuva 5.12. *Keskusta-Varissuo -klusterin pysäkkijoukko kartalla.*

Taulukossa 5.10 on Keskusta-Varissuo-klusterin viisi suosituinta linjaryhmää. Huomataan, että vaihtoja tehdään paljon linjalta 12 linjoille 32 ja 42, vaikka linja 12 itsessäänkin kulkisi Varissuolle. Linjojen reittiosuudet poikkeavat siten, että linjat 32 ja 42 ajavat Varissuolle Kupittaan kautta, linjan 12 ajaessa Vasaramäen kautta. Tämän perusteella voidaan päätellä, että kysyntä kohdistuu Port Arthurin ja Kupittaan välille.

Taulukko 5.10. *Keskusta-alueelta Varissuolle suuntautuneiden vaihtomatkaketjujen käytetyimmät linjaryhmät.*

Linjaryhmä	*Määrä*
12, 42	39 kpl
12, 32	17 kpl
32, 42	13 kpl
42, 32	9 kpl
1, 32	9 kpl

Satama-Varissuo, Satama-Vaala, Satama-Harittu

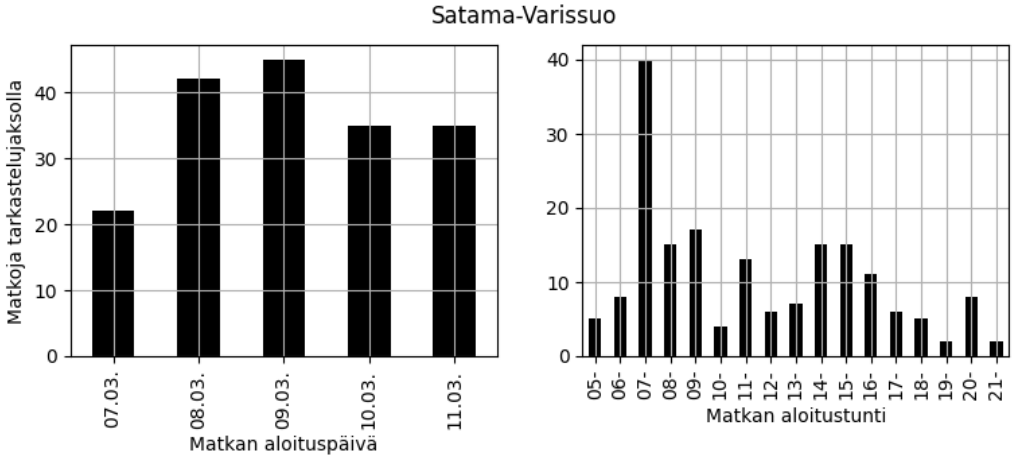
Taulukossa 5.11 on Satamasta alkavien vaihtomatkaketjujen viisi käytetyintä nousupysäkkiryhmää klustereittain. Turussa pysäkkien numerointi alkaa Satamasta numerolla yksi ja pysäkkinumerot kasvavat keskustaa kohden mentäessä.

Taulukko 5.11. *Satamasta alkavien vaihtomatkaketjujen käytetyimmät nousupysäkkiryhmät klustereittain.*

Satama-Varissuo		*Satama-Vaala*		*Satama-Harittu*	
Nousupysäkit	*Määrä*	*Nousupysäkit*	*Määrä*	*Nousupysäkit*	*Määrä*
5, 1904	35 kpl	7, 1912	34 kpl	7, 1983	17 kpl
7, 1904	27 kpl	2, 1912	12 kpl	6, 1983	16 kpl
4, 1904	24 kpl	6, 1912	12 kpl	9, 1983	8 kpl
9, 1904	23 kpl	5, 1912	11 kpl	8, 1983	6 kpl
6, 1904	22 kpl	9, 1912	11 kpl	219, 1983	6 kpl

Pysäkki 7 sijaitsee Turun taideakatemian välittömässä läheisyydessä. Turun taideakatemia on Turun Ammattikorkeakoulun toimipiste, jonka pääkampus sijaitsee Vaalaan menevän linjan 60 reitillä. Pääkampuksen tavoittaa hyvin myös Varissuon linjoilla. Pysäkki 5 sijaitsee uuden Linnanfältin kaupunginosan läheisyydessä. Laivaterminaalien pysäkit ovat 1 ja 2, joista ainoastaan pysäkki 2 nousee käytetyimpien nousupysäkkien listaan klusterissa Satama-Vaala. Yleisesti voitaneen siis olettaa, että Satamasta lähiöihin suuntautuvat klusterit käsittävät muiden kuin laivamatkustajien tekemiä matkoja.

Kuvassa 5.13 on esitetty Satama-Varissuo-klusterin matkojen aloitusaikojen jakauma. Satama-Varissuo ja Satama-Vaala-klustereiden kaikki matkat on tehty arkipäivinä. Jakaumat ovat hyvin samanmuotoisia ja selkeä huippu on nähtävissä kello seitsemän tunnilla. Satama-Harittu välillä on matkoja myös viikonloppuna. Satama-Keskusta-reittiosuuden lyhyydestä johtuen pysäkkisarjoissa on hieman hajontaa siten, että myös keskusta-alueella aloitetut matkat nousevat klusteriin.

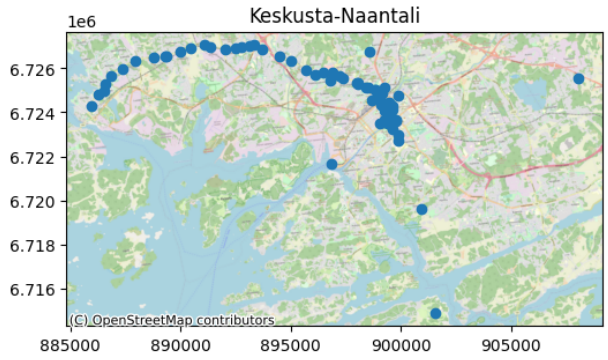


Kuva 5.13. *Satama-Varissuo-matkojen jakaantuminen viikonpäiville ja tunneille.*

Keskusta-Naantali, Varissuo-Naantali, Harittu-Naantali, Runosmäki-Naantali

Naantaliin liikennöivät linjat 6 Lieto-Turun keskusta-Naantali ja 7 Kaarina-Turun keskusta-Naantali. Linjat ovat Fölin käytetyimpiä linjoja ja niillä tehtiin vuonna 2021 lähes 12 % kaikista Fölin matkoista. Näiden linjojen lisäksi reittiosuutta jakaa ruuhka-aikana linja 7A Turun keskusta-Raisio-Vuorenpää.

Keskustan ja Naantalin välisen reittiosuuden pysäkkisarja on pitkä ja reitti käsittää Naantalin lisäksi huomattavan määrän tärkeitä kohteita, kuten Raision keskustan, Raision tehtaiden alueen, Turun Länsikeskuksen ja Satakunnantien palvelut. Pitkästä pysäkkisarjasta johtuen, myös Keskusta-Naantali -klusteriin tulee runsaasti erilaisia keskusta-alueella tehtyjä liityntämatkoja. Klusterin pysäkkijoukko on esitetty kuvassa 5.14. Myös tässä joukossa on päätepysäkeillä tehtyjä leimauksia, joilla kuljettaja ei ole valinnut paluusunnan reittiä ajoneuvopäätteeltä. Tästä johtuen kartalle piirtyy hajanaisia pysäkkipisteitä joidenkin linjojen pääteasemille.



Kuva 5.14. *Keskusta-Naantali-klusterin pysäkkijoukko kartalla.*

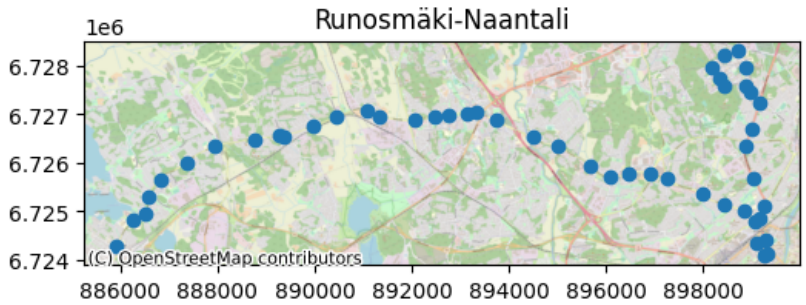
Taulukossa 5.12 on Keskusta-Naantali-reitin käytetyimmät linjaryhmät. Suhteellisesti eniten vaihtoja on tehty linjalta 18, mutta myös linjojen 6 ja 7 välillä on tehty runsaasti vaihtoja. Linjapari mahdollistaa hyvin poistumisen Raisiossa tai Länsikeskuksessa kahden tunnin aikaikkunassa.

Taulukko 5.12. *Keskusta-alueelta Naantaliin suuntautuneiden vaihtomatkaketjujen käytetyimmät linjaryhmät.*

Linjaryhmä	*Määrä*
18, 6	19 kpl
18, 7	25 kpl
7, 6	20 kpl
6, 7	19 kpl
7A, 6	8 kpl

Kaikille Naantaliin päättyville klustereille yhteinen piirre on se, että matkan alkuosuus on tehty suositulla linjalla asukasluvultaan suuresta lähiöstä. Koska Naantalin reitin varrella on monia kohteita, on näiden klustereiden analysointi erityisen haastavaa, kun lopullista poistumispysäkkiä ei tiedetä. Kaikissa klustereissa myös aloitusaikojen jakauma noudattaa normaalia, mm. kuvassa 5.1 näkyvää kysyntää.

Haritun ja Varissuon suunnasta Naantaliin mentäessä on käytännössä pakko tehdä vaihto keskustassa. Kuvassa 5.15 esitetystä Runosmäki-Naantali-klusterin pysäkkijoukosta huomataan, että yhtään vaihtoa ei ole tehty keskustassa. Tämän perusteella voitaneen olettaa, että matkustajat pyrkivät optimoimaan liikkumistaan niin, että matka-ajat ovat mahdollisimman lyhyet. Suosituin poistumispysäkki on 203 Autistenaukio, jota on käytetty 83 prosentissa klusterin matkoista.



Kuva 5.15. *Runosmäki-Naantali-klusterin pysäkkijoukko kartalla.*

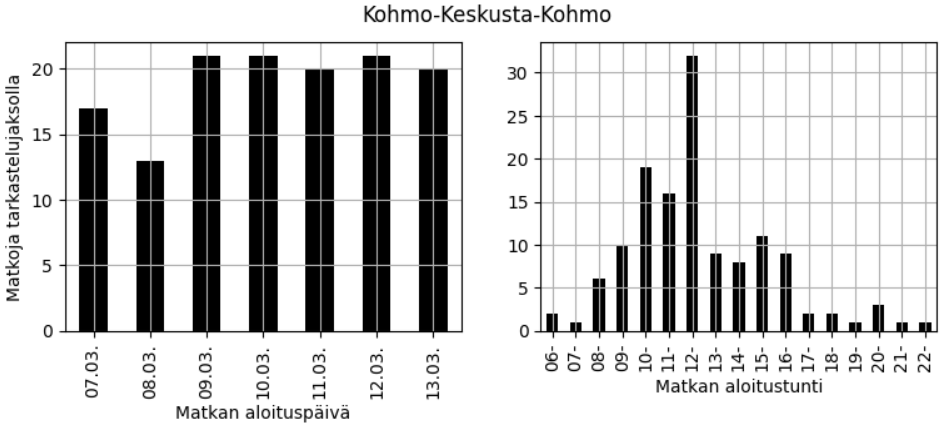
Kohmo-Keskusta-Kohmo, Runosmäki-Kohmo

Kohmo-Keskusta-reittiosuutta liikennöidään neljällä linjatunnuksella: 2, 2A, 2B ja 2C. Näistä 2B ja 2C lähtevät Liedon Littoisista, mutta jakavat saman reittiosuuden Kohmosta Keskustaan. Taulukossa 5.13 nähdään klusterin käytetyimmät nousupysäkit. Suurin osa nousuista tehdään Kuralan alueella olevilla pysäkeillä, joten mikäli matkaketjujen viimeiset poistumispysäkit tunnettaisiin, supistuisi klusteri todennäköisesti kattamaan ainoastaan Kurala-Keskusta-Kurala-reitin. Reitin varrelle jää myös alueellinen keskussairaala, TYKS T-sairaala.

Kuvasta 5.16 nähdään matkojen jakautuvan erittäin tasaisesti viikonpäiville. Kello kahdentoista tunnilla on selkeä huippu, joka voisi viitata esimerkiksi eläkeläisten tekemiin asiointimatkoihin.

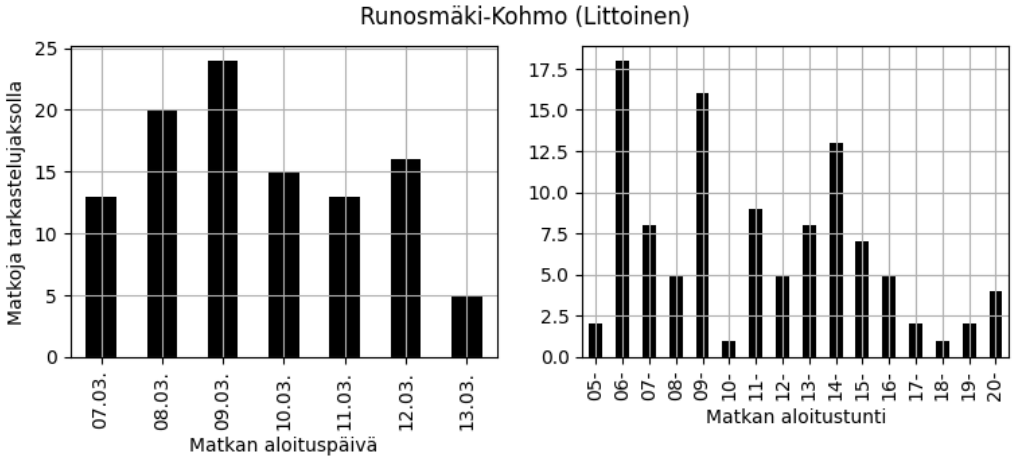
Taulukko 5.13. *Kohmo-Keskusta-Kohmo-vaihtomatkaketjujen käytetyimmät nousupysäkit.*

Nousupysäkit	*Määrä*
58, 1983	45 kpl
57, 1983	25 kpl
61, 1983	8 kpl
1656, 1983	8 kpl
59, 1983	6 kpl



Kuva 5.16. *Kohmo-Keskusta-Kohmo-matkojen jakaantuminen viikonpäiville ja tunneille.*

Runosmäki-Kohmo (Littoinen) -klusterin kaikki vaihdot on tehty linjalta 18 linjoille 2B ja 2C, jotka jatkavat Kohmosta Littoisiin. Kohmon ja Littoisten välinen osuus käsittää pääasiassa asuinalueita, joten tiettyä selittävää tekijää ilmiölle on vaikeaa löytää. Kuvasta 5.17 kuitenkin nähdään tiettyjä huippuja kello kuuden, yhdeksän ja 14:n tunneilla, joka voisi viitata työssä käyntiin.

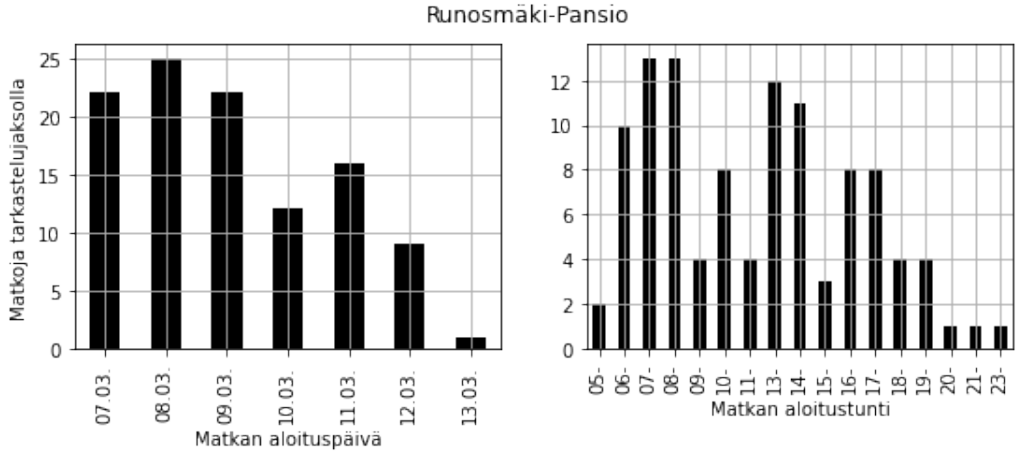


Kuva 5.17. *Runosmäki-Kohmo (Littoinen) -matkojen jakaantuminen viikonpäiville ja tunneille.*

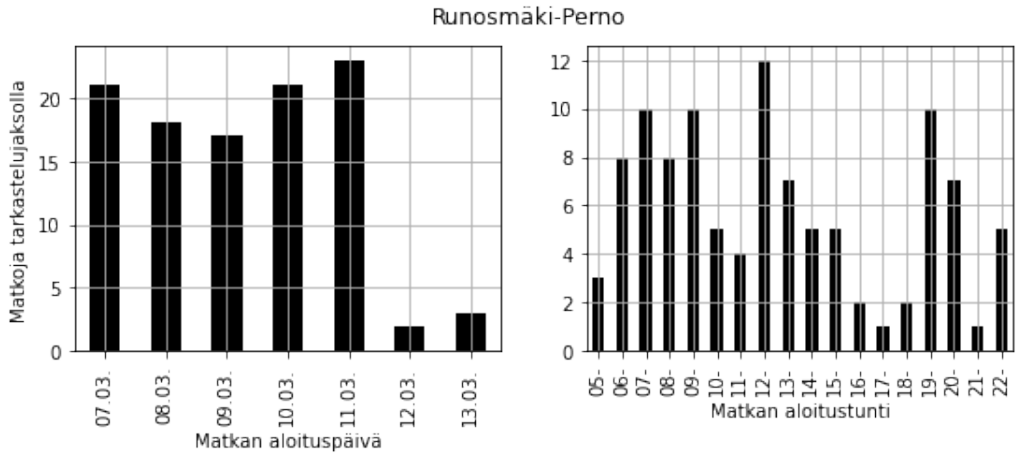
Runosmäki-Perno, Runosmäki-Pansio

Runosmäki-Perno ja Runosmäki-Pansio-reitit muodostavat yhteyden Turun suurimman lähiön ja Perno/Pansio-teollisuuskeskittymän välille. Alueella sijaitsee mm. Meyer Turun telakka-alue ja Pansion varuskunta. Vaihdot on tehty linjalta 18 linjoille 32, 42. Yksittäisillä matkoilla on käytetty myös muita linjoja näiden linjojen välillä.

Kuvissa 5.18 ja 5.19 nähdään matkojen aloitusaikojen jakautuminen viikonpäiville ja tunneille. Kysyntä jakautuu varsin tasaisesti ja huipputuntienkin kohdalla kyse on lähinnä parin matkan erosta. Poikkeavan suurta kysyntää esimerkiksi työmatkaruuhkan aikana ei ole havaittavissa.



Kuva 5.18. *Runosmäki-Pansio-matkojen jakaantuminen viikonpäiville ja tunneille.*



Kuva 5.19. *Runosmäki-Perno-matkojen jakaantuminen viikonpäiville ja tunneille.*

6. YHTEENVETO

Tämän työn tavoitteina oli rakentaa algoritmi vaihtomatkaketjujen tunnistamiseen matkustustapahtumadatan perusteella sekä klusteroida laskennalliset matkustajavirrat tunnistettujen ketjujen samankaltaisuuden perusteella.

Vaihtomatkaketjut rakennettiin tarkastelemalla yksittäisen matkatuotetunnisteen käyttöjä kahden tunnin aikaikkunan sisällä. Tapahtumiin yhdistettiin nousutapahtuman tietoja vastaava tapahtumahetkellä voimassaollut reittiaineisto. Ketju rakennettiin matkaosuuksista siten, että edellinen ketju päätettiin seuraavan matkan nousupysäkkiä lähimmälle pysäkille. Koska lopullista poistumispysäkkiä ei tiedetty, ketjun viimeinen matka sisältää pysäkit nousupysäkiltä reitin päätepysäkille asti. Muodostetut vaihtomatkaketjut olivat pysäkkikoodien merkkijonoesityksiä, joten klusterointi toteutettiin tekstiaineistojen vertailuun yleisesti käytetyllä Locality-sensitive hashing (LSH) -menetelmällä.

Aineisto tiivistettiin minhash-algoritmilla, jotta samankaltaisuuksien löytäminen olisi tehokkaampaa. Samankaltaisuuden mittariksi valittiin Jaccard-indeksi. Klusterointitehtävän toteutukseen vertailtiin kahta kirjallisuudessa esiteltyä menetelmää. LSH-indeksiin perustuva menetelmä sijoittaa todennäköisesti samankaltaiset objektit samoihin koreihin. LSH-metsä etsii kullekin objektille vakiomäärän mahdollisimman samankaltaisia naapureita. Työssä tavoiteltiin dynaamista klusterointia siten, että kaikki tietyn Jaccard-indeksin kynnysarvon ylittävät objektit pääsisivät samaan koriin ja sen alittavat objektit eivät. Tästä syystä valittiin LSH-indeksiin perustuva menetelmä.

Valitun menetelmän laatua arvioitiin vertailemalla sillä toteutettujen klustereiden sisältöä raakamenetelmällä, eli koko aineisto iteroimalla, saatujen klustereiden sisältöön. Huomattiin, että sadan suurimman klusterin joukosta löytyneet reittimuodot löytyivät pääsääntöisesti molemmilla menetelmillä saaduista klustereista. Locality-sensitive hashing -toteutuksen aikahyöty on kuitenkin merkittävä raakamenetelmään nähden.

Klustereiden sisältöä tarkasteltiin visuaalisesti asettamalla pysäkkijoukkojen GPS-koordinaatit kartalle. Samojen reittimuotojen toistoa oli sadan suurimmankin klusterin joukossa runsaasti. Kokoavaa klusterointia jatkamalla oltaisi saatu yhdistettyä samankaltaisia klustereita edelleen suuremmiksi joukoiksi, mutta päällekkäisyyden välttämiseksi vaihtomatkaketjuja, tai klustereita olisi siinäkin tapauksessa pitänyt estää päätymästä useampaan kuin yhteen ryhmään. Tämä lisäisi todennäköisyyttä sille, että kaksi samankaltaista

vaihtomatkaketjua tai klusteria, eivät koskaan päätyisi samaan ryhmään.

Matkustajavirtojen analysointi klustereiden perusteella onnistui hyvin. Vaihtomatkaketjujen määrät klustereissa olivat odotettuja pienempiä. Tästä voi päätellä ettei selkeitä kysyntäpiikkejä ole havaittavissa ainakaan tutkimuksessa käytetyllä tarkastelujaksolla. Missään suurimmista klustereista ei ole tilannetta, jossa aikatauluja olisi syytä entisestään synkronoida, koska käytettyjen linjojen vuoroväli on nykyiselläänkin tiheä. Huomattavaa on, että vaihtojen ollessa mahdollisia keskustan ulkopuolella, niitä myös hyödynnetään.

Vaihtomatkaketjujen tunnistamiseen kehitetty algoritmi löytää vaihtomatkaketjut ja voisi soveltua tuotantokäyttöön. Algoritmin ajo yhden päivän aineistolla vie useita minuutteja, joten analysointia varten vaihtomatkaketjun metatietoineen sisältävät JSON-dokumentit on syytä generoida valmiiksi. Tarkkuuden parantamiseksi on syytä harkita myös täydellisen matkaketjun, eli lähtöpisteestä lähtöpisteeseen etsimistä, jolloin lopulliset poistumispysäkit voidaan päätellä.

LSH-menetelmällä toteutettu klusterointi soveltuu tuotantokäyttöön erityisesti sen nopeuden puolesta ja sen avulla voidaan tunnistaa kysyntää. Algoritmia hyödyntävä sovellus tulee kuitenkin rakentaa siten, että käyttäjän on mahdollista antaa haluttu samankaltaisuuden kynnysarvo, tai LSH-parametrit syötteenä. Käyttäjälle on myös syytä esittää statistiikkaa kunkin klusterin todellisesta samankaltaisuuden toteutumisesta, koska menetelmän nopeuden hintana on sen perustuminen todennäköisyyteen.

LÄHTEET

[1] Föli. *Tietoa Fölistä*. 2022. URL: https://www.foli.fi/fi/etsitk%5C%C3%5C%B6n%5C%C3%5C%A4it%5C%C3%5C%A4/tietoa-f%5C%C3%5C%B6list%5C%C3%5C%A4 (viitattu 25. 10. 2022).

[2] Trafix Oy, Liidea Oy ja Reform Oy. *Runkobussilinjaston kehittämisohjelma vuosille 2012–2020*. Tekninen raportti. 2012. URL: https://cms.foli.fi/sites/default/files/documents-2021-06/Runkobussilinjaston%5C%20kehitt%5C%C3%5C%A4misohjelma%5C%20vuosille%5C%202012%5C%E2%5C%80%5C%932020.pdf.

[3] Mari Linna. ”Liikkumista rajoittavat tekijät Turun seudun joukkoliikenteessä – aikamaantieteellinen näkökulma tulevaan linjastouudistukseen”. Tutkielma. Turun yliopisto, elokuu 2017.

[4] Subeh Chowdhury ja Avishai Ceder. ”Definition of Planned and Unplanned Transfer of Public Transport Service and User Decisions to Use Routes with Transfers”. *Journal of Public Transportation* 16 (2013).

[5] Michael D. Meyer. *Transportation planning handbook*. John Wiley Sons, Inc., 2016.

[6] Atte Supponen, Arttu Kosonen ja Ruut Haapamäki. *Runkolinjaston lyhyen aikavälin matkustajamääräennusteet*. Tekninen raportti. 2018. URL: https://cms.foli.fi/sites/default/files/documents-2021-06/Runkolinjaston%5C%20lyhyen%5C%20aikav%5C%C3%5C%A4lin%5C%20matkustajam%5C%C3%5C%A4%5C%C3%5C%A4r%5C%C3%5C%A4ennusteet.pdf.

[7] Föli. *General Transit Feed Specification*. 2022. URL: https://data.foli.fi/doc/gtfs/v0/index (viitattu 02. 08. 2022).

[8] Föli. *Joukkoliikenteen suunnitteluaineisto*. 2022.

[9] INIT GmbH. *ID-/Account-based Ticketing, Whitepaper*. 2017.

[10] J. D. Schmöcker, F. Kurauchi ja Shimamoto H. ”An Overview on Opportunities and Challenges of Smart Card Data Analysis”. Teoksessa: *Public Transport Planning with Smart Card Data*. Taylor Francis Group, 2017, s. 1–11.

[11] Turun kaupunki. *Turun kaupungin käyttämät henkilörekisterit, Fölin henkilörekisteri*. 2022. URL: https://rekisteri.turku.fi/webSivut/pdfKirjasto/F%5C%C3%5C%B6lin%5C%20Asiakastietorekisteri.pdf (viitattu 13. 10. 2022).

[12] M. Hickman. ”Transit Origin-Destination Estimation”. Teoksessa: *Public Transport Planning with Smart Card Data*. Taylor Francis Group, 2017, s. 15–36.

[13] Giuseppe Bonaccorso. *Machine Learning Algorithms – Second Edition*. Packt Publishing, 2018.

[14] Alice Zheng ja Amanda Casari. *Feature engineering for Machine Learning*. O'Reilly Media, 2018.

[15] Wael H Gomaa and Aly A Fahmy. "A Survey of Text Similarity Approaches". In: *International Journal of Computer Applications* 68.13 (2013), pp. 13–18.

[16] Pratap Dangeti. *Statistics for Machine Learning*. Packt Publishing, 2017.

[17] Nurhilyana Anuar ja Abu Bakar Md Sultan. "Validate conference paper using dice coefficient". *Computer and Information Science* 3.3 (2010), s. 139.

[18] Ahmad Imran. *40 Algorithms Every Programmer Should Know*. Packt Publishing, 2020.

[19] Anand Rajaraman and Jeffrey Ullman. *Mining of massive datasets*. 2011. 315 pp.

[20] Charu Aggarwal ja Chandan K. Reddy. *Data Clustering*. Taylor Francis Group, 2015.

[21] Teemu Holopainen. "Klusterointi hierarkkisilla ja kombinatorisilla menetelmillä - sovelluksena tilastomenetelmien peruskurssiaineisto". Tutkielma. Jyväskylän yliopisto, maaliskuu 2012.

[22] Kari Lehmussaari. "Hierarkkinen klusterointi". Teoksessa: Klusterointimenetelmät-seminaari, Raportti C-2002-54. Helsingin yliopisto, tietojenkäsittelytieteen laitos, 2002.

[23] Koga Hisashi, Ishibashi Tetsuo ja Watanabe Toshinori. "Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing". Teoksessa: Lecture Notes in Computer Science · October 2004. Springer-Verlag London Ltd., 2006.

[24] Blekanov ja Vasilii Korelin. "Hierarchical clustering of large text datasets using Locality-Sensitive Hashing" (2015).

[25] Henry Garner. *Clojure for Data Science*. Packt Publishing, 2015.

[26] Francisco Javier Moreno Arboleda, Felipe Cortés Noreña ja Benjamín Cruz Álvarez. "On the Use of Minhash and Locality Sensitive Hashing for Detecting Similar Lyrics." *Engineering Letters* 30.1 (2022), s. 1–16. URL: http://libproxy.tuni.fi/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,uid&db=asn&AN=155423493&site=ehost-live&scope=site.

[27] Ercan Canhasi. "Fast document summarization using locality sensitive hashing and memory access efficient node ranking". *International Journal of Electrical and Computer Engineering (IJECE)* 6 (kesäkuu 2016), s. 945.

[28] Datasketch Eric Zhu. *DataSketch API Documentation*. 2022. URL: http://ekzhu.com/datasketch/documentation.html (viitattu 10. 09. 2022).

[29] Bawa Mayank, Condie Tyson ja Ganesan Prasanna. "LSH Forest: Self-Tuning Indexes for Similarity Search" (2015).

[30] Sergios Theodoridis ja Koutroumbas Konstantinos. "Clustering: Basic Concepts". Teoksessa: *Pattern Recognition - Fourth Edition*. San Diego, California: Elsevier Inc., 2009, s. 596–597.

[31] Kyberturvallisuuskeskus. *SHA-1-tiivistefunktio on lopullisesti murrettu*. 2020. URL: https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/sha-1-tiivistefunktio-lopullisesti-murrettu (viitattu 13. 09. 2022).