**Jari-Matti Kankaanpää**

# Klusterointi Radiotaajuuspaikannuksessa

Tietotekniikan pro gradu -tutkielma

18. toukokuuta 2017

Jyväskylän yliopisto

Tietotekniikan laitos

**Tekijä:** Jari-Matti Kankaanpää

**Yhteystiedot:** jajokank@gmail.com

**Ohjaajat:** Tapani Ristaniemi ja Riaz Uddin Mondal

**Työn nimi:** Klusterointi Radiotaajuuspaikannuksessa

**Title in English:** Clustering in Radio Frequency Positioning

**Työ:** Pro gradu -tutkielma

**Suuntautumisvaihtoehto:** Tietoliikennetekniikka

**Sivumäärä:** 53+0

**Tiivistelmä:** Mobiililaitteiden tarkalla paikantamisella on tärkeä rooli nykyisissä ja tulevissa langattomiin verkkoihin perustuvissa sovelluksissa. GPS-paikannus on yleisesti käytetty, mutta se suoriutuu kehnosti sisätiloissa ja saattaa ruuhkahuippujen aikana olla kykenemätön suorittamaan kaikkia paikannuspyyntöjä tehokkaasti. Mobiiliverkossa paikannus voidaan suorittaa kolmiomittauksen avulla, joka vaatii kuitenkin aktiivisen puhelun. Paikkatieto mahdollistaa verkko-operaattoreille ajantasaisen statistiikan keräämisen muuttuvassa radioverkkoympäristössä. Se auttaa myös parantamaan tilannetietoisuutta onnettomuuksien ja pelastusoperaatioiden aikana. Langattomien verkkojen massiivista dataa ei vielä ole käytetty kunnolla paikannusongelmien ratkaisemiseen. Tämän työn tarkoituksena on selvittää tunnettujen algoritmien paikannustarkkuus klusteroimalla signaalidataa, joka on kerätty empiirisesti WLAN- ja mobiiliverkon tukiasemista. Klusterointiin käytetään K-means-, K-medoids- ja Knn-algoritmeja, joista parhaat paikannustulokset saatiin Knn-algoritmilla. Se saavutti noin 17 metrin paikannustarkkuuden, joka on suunnilleen 20% parempi kuin K-means- tai K-medoids-algoritmien saavuttamat paikannustarkkuudet.

**Avainsanat:** klusterointi, LTE, WLAN, radiotaajuuspaikannus

**Abstract:** Positioning of user-equipments (UE) is a vital part of recent and future applications based on wireless networks. Global Positioning System (GPS) is popular and widely used, but it performs poorly indoors and might be too busy to handle all requests properly

during rush hours. In mobile networks, positioning can be done by triangulation, though it requires that a phone call is going on. Location data allows e.g. network operators to get updated measurements from UEs in a changing radio network environment. Situational awareness, e.g. during disasters or rescue operations, could be improved with precise positioning of UEs. Massive data of wireless networks are yet to be used for solving positioning issues. The aim of this work is to use known algorithms to cluster signal data collected empirically from mobile base stations and WLAN access points and determine the positioning error of each algorithm. Results will be compared between the used clustering algorithms. The algorithms that are used for clustering are: K-means, K-medoids and Knn, from which Knn achieved best results, approximately 17 metres, having 20% better positioning accuracy than K-means and K-medoids.

**Keywords:** clustering, LTE, WLAN, radio frequency positioning

# Kuviot



# Taulukot

# Sisältö

# 1 Johdanto

Tarkalla paikkatiedolla on nykyään tärkeä asema etenkin mobiilisovelluksien toiminnassa ja sen tarve tulee epäilemättä kasvamaan entisestään uusien sovelluksien myötä. Nykyään laajalti käytetty GPS-paikannus (*Global Positioning System*) toimii huonosti sisätiloissa, joissa paikannus voidaan toteuttaa myös WLAN-tukiasemien (*Wireless Local Access Network*) signaaleja mittaamalla (Bahl ja Padmanabhan 2000). Yhdistämällä sekä WLAN-tukiasemien että mobiiliverkon tukiasemien signaalinvoimakkuuksia, voidaan mobiililaite paikantaa myös idle-tilassa (puhelu ei käynnissä) (Mondal, Turkka ja Ristaniemi 2015).

Paikkatietoa voidaan käyttää esimerkiksi viranomais- ja sisällönjakopalveluissa, sosiaalisessa mediassa sekä kaupallisissa sovelluksissa. Viranomaiset voivat hätätapauksissa tarkan paikkatiedon avulla kohdistaa resursseja nopeasti ja tehokkaasti. Sisällönjakoon perustuvissa palveluissa ja sosiaalisessa mediassa on nykyään jo laajalti käytössä paikkatiedon liittäminen käyttäjän tekemän julkaisun yhteyteen. Kaupallisissa tarkoituksessa paikkatieto mahdollistaa kohdennetumman kuluttajamainonnan ja sitä voidaan käyttää esimerkiksi liikekeskussuunnitteluun.

Tämän työn tarkoituksena on tutkia paikannusmenetelmää, jossa hyödynnetään WLAN- ja mobiiliverkon tukiasemien signaaleja mobiililaitteen paikantamiseen. Menetelmä mahdollistaa paikantamisen myös sisätiloissa ja se on saatavilla myös silloin, kun GPS:n paikannussatelliitti ei ole. Erityisesti tiheään asutuilla alueilla ja kaupunkikeskuksissa, missä mobiiliverkon kate on hyvä ja useita WLAN-tukiasemia on saatavilla, radiotaajuuspaikannuksen hyödyt nousevat esiin.

Tässä tutkimuksessa määritellään mobiililaitteen paikannustarkkuus klusteroimalla empiirisesti kerättyä radiotaajuusdataa. Klusteroinnin tavoitteena on ryhmitellä niin sanottu opetusdata (radiotaajuusmittaukset, joiden GPS-koordinaatit tiedetään) yhdessä testausdatan (GPS-koordinaatit eivät tiedossa) kanssa ja arvioida paikannustarkkuutta sen perusteella, mihin klusteriin testausdata sijoittui. Zekavat ja Buehrer 2011 määrittelevät radiotaajuusdatan koostuvan niiden WLAN- ja mobiiliverkon tukiasemien signaaliparametreista, joihin mobiililaite on yhdistetty. Nämä parametrit muodostavat sijaintiriippuvaisia radiotaajuussormenjälkiä,

joista jokainen on sidottu johonkin maantieteelliseen lokaatioon.

Työssä opetus- ja testausdataa klusteroidaan K-means-, K-medoids- ja Knn- (*K-nearest neighbour*) algoritmeilla. Ennen varsinaista klusterointia radiotaajuusdata muokataan sopivampaan ja helpommin käsiteltävään muotoon. Sekä datan esikäsittely että sen klusterointi toteutetaan Matlab-ohjelmistolla, joka tarjoaa valmiit funktiot edellä mainituille algoritmeille. Varsinaisen klusteroinnin jälkeen tuloksista eritellään paikannustarkkuus, josta esitetään sekä 68%:n että 95%:n osuus.

Tutkielman luvussa 2 kerrotaan radiotaajuuspaikannuksen pääperiaatteista sekä selitetään paikannusprosessissa oleellisten käsitteiden, kuten radiotaajuussormenjälki ja korrelaatiotietokanta, toiminta. Tämän jälkeen avataan itse paikannusprosessia ja sen koulutus- ja testausvaihetta. Luvun lopussa on lisäksi osio, joka kertoo aihepiirin aiemmasta tutkimuksesta. Luvussa 3 käydään yleisesti läpi klusterointia tiedonlouhintatekniikkana, sen yleisimpiä käyttötarkoituksia sekä klusteroinnin määritelmää ja periaatetta. Lisäksi kerrotaan yhdestä klusteroinnin luokittelutavasta ja klusteroinnin pääpiirteisistä askeleista. Klusteroinnin käsitystä, sen askeleita ja klusterointikriteerin valintaa on lisäksi havainnollistettu kuvilla.

Luvussa 4 käydään läpi K-means-, K-medoids- ja Knn-algoritmien toimintaa. Algoritmeista on erityisesti kerrottu niiden toimintaan oleellisesti vaikuttavien parametrien valinnasta. K-means- ja K-medoids-algoritmien tapauksessa tällä tarkoitetaan klusterimäärän valintaa ja Knn-algoritmin kohdalla naapuriobjektien määrän valintaa. 5. luvussa kerrotaan klusteroitavan datan luonteesta, sen määrästä ja keräystavasta. Luvussa kerrotaan myös tutkielman analysointiprosessista. Luvussa 6 on käytetyillä algoritmeilla saadut klusterointitulokset sekä tulosanalyysi omina kappaleinaan. Tulosanalyysissa ei ole tarkoitus perehtyä syvällisemmin käytettyjen algoritmien toimintaan tai niiden tehokkuuteen vaan keskittyä algoritmien vertailuun saatujen tulosten perusteella. Viimeisenä lukuna on tutkimuksen yhteenveto.

# 2 Radiotaajuuspaikannus

Tässä luvussa paneudutaan radiotaajuuspaikannuksen periaatteisiin ja sen keskeisiin käsitteisiin. Radiotaajuussormenjäljen rakenteesta ja korrelaatiotietokannan toiminnasta kerrotaan omissa kappaleissaan. Lisäksi paikannusprosessia sekä koulutus- ja testausvaihetta on avattu myös omissa kappaleissaan. Viimeisessä kappaleessa käsitellään aihepiirin aiempaa tutkimista, joka on jaettu paikannuskohteen perusteella sisä- ja ulkotilamenetelmiin. Radiotaajuussormenjäljen muodostusta sekä koulutus- ja testausvaiheen toimintaa on havainnollistettu kuvilla.

## 2.1 Radiotaajuuspaikannuksen periaate

Mondal ym. 2013 mukaan käyttäjiensä sijaintitietoa hyödyntävät palvelut ja sovellukset ovat viime vuosina kasvattaneet entisestään suosiotaan monissa langattomissa verkoissa, kuten GSM- (*Global System for Mobile Communications*), UMTS- (*Universal Mobile Telephony System*), LTE- (*Long Term Evolution*) ja WLAN-verkoissa. Tämä on rohkaissut tutkijoita kehittämään laajalti erilaisia paikantamistekniikoita käyttäjien mobiililaitteiden sijainnin arvioimiseen. Paikannuksen mahdollistavilla radioverkoilla on monia käyttötarkoituksia esimerkiksi lainvalvonnassa, liikenneturvallisuudessa ja ensiaputilanteissa (Zekavat ja Buehrer 2011).

Laitinen, Lähteenmäki ja Nordström 2001 määrittelevät radiotaajuuspaikannuksen (*engl. radio frequency positioning*) ja DCM-metodit (*Database Correlation Method*) paikannustekniikoiksi, joita voidaan soveltaa mihin tahansa langattomaan verkkoon. Näiden tekniikoiden pääperiaatteena on varastoida mobiililaitteen havaitsema signaali-informaatio koko kuuluvuusalueelta tietokantaan paikannuspalvelimen (*engl. location server*) käytettäväksi. Tietokannan sisältämiä signaalinäytteitä kutsutaan sormenjäljiksi (*engl. fingerprint*). Radiotaajuuspaikannukseen perustuvat paikannustekniikat ovat kaikkein tehokkaimpia tekniikoita mobiililaitteiden paikantamiseen, koska ne perustuvat mittauksiin, jotka tehdään mobiililaitteen ja tukiaseman välillä kulkevan signaalin fyysisistä ominaisuuksista (Porretta ym. 2008).

Tarkassa sijaintiarviossa on tärkeää, että paikannusprosessin yksityiskohdat ja teoreettiset ra-

jat on ymmärretty hyvin. Langattoman paikantamisen on toimittava myös haastavissa ja realistissa tilanteissa, joissa on monia virheitä aiheuttavia tekijöitä. Yleisimmät virheenaiheuttajat ovat monitie- (*engl. multipath propagation*) ja NLOS-eteneminen (*non-line-of-sight propagation*). (Gezici ym. 2005) Sijainninarviointi voidaan määritellä prosessiksi, jossa arvioidaan kohdeobjektin sijainti langattomassa verkossa vaihtamalla signaaleja kohdeobjektin ja viiteobjektien välillä (Gezici 2008).

## 2.2 Radiotaajuussormenjälki

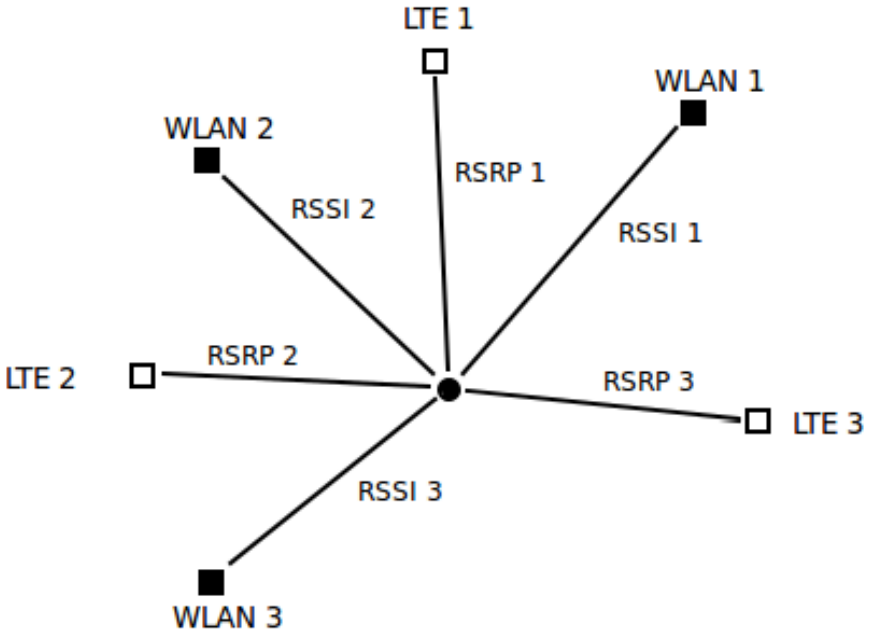
Ellei toisin mainita, tämän kappaleen teksti perustuu teokseen Zekavat ja Buehrer 2011.

Mikä tahansa maantieteellinen sijainti on tunnistettavissa radiotaajuussormenjäljen perusteella samalla tavalla, kuin ihminen on tunnistettavissa sormenjälkensä perusteella. Jotta sormenjäljen muodostaminen onnistuisi, sen signaaliparametrien määrä on oltava tarpeeksi suuri. Valituilla parametreilla, tai vähintään niiden keskiarvoilla, on oltava pieni vaihtelevuus aikaan nähden. On kuitenkin selvä, että parametrit eivät pysy täysin samoina ajan kuluessa. Vaikka keskiarvojen käyttäminen vähentää pieniä vaihteluita, muutokset radioverkossa, kuten uuden solun lisääminen, lähetys- tai vastaanottoantennin muuttaminen tai lähetystehon muuttaminen saattavat rikkoa siteen sormenjäljen ja sijainnin välillä. Tässä tapauksessa on hankittava uusia sormenjälkiä.

Radiotaajuussormenjäljet ovat joukko paikkariippuvaisia signaaliparametreja, joita on saatavilla radioverkossa. Koska parametrit ovat paikkariippuvaisia, jokainen sormenjälki liittyy johonkin tiettyyn sijaintiin. Sormenjälki on sitä yksilöllisempi, mitä enemmän siinä on signaaleja, tai mitä enemmän parametreja signaalit sisältävät. Tällöin myös paikannustarkkuus on parempi. Sormenjälki voi sisältää paljon erilaisia signaaliparametreja, esimerkiksi RSS- (*Received Signal Strength*) sekä RTD-arvoja (*Round Trip Delay*) ja PD-profiililleja (*Power Delay*) (Campos ja Lovisolo 2008, 2009; Ahonen ja Laitinen 2003). Nämä parametrit mitataan joko yhdestä tai useammasta ankkurisolusta (solu, jonka koordinaatit tiedetään).

Sormenjäljet voidaan luokitella joko kohde- tai viitesormenjälkiin. Kohdesormenjälki on sen mobiililaitteen sormenjälki, joka halutaan paikantaa. Se sisältää siis mobiililaitteen tai ankkurisolun mittaamat signaaliparametrit. Viitesormenjäljet kerätään tai luodaan koulutusvai-

heessa ja säilötään korrelaatiotietokantaan. Jokainen viitesormenjälki sisältää yksilöllisen sijaintitiedon. Ideaalissa tilanteessa kaikki kohdesormenjäljen parametrit löytyvät myös viitesormenjäljestä. Matriisi 2.1 havainnollistaa kuvan 1 esittämästä tilanteesta muodostettua kohdesormenjälkeä (sijainti ei tiedossa). Kuvassa musta ympyrä kuvaa sijaintia, joka halutaan määrittää ympärillä olevien LTE- ja WLAN-tukiasemien avulla. Tukiasemista on kerätty tunnustiedot ja signaalinvoimakkuudet. Rivien määrä matriisissa riippuu siitä, kuinka monta LTE- ja/tai WLAN-tukiasemaa paikannettavassa kohdassa havaitaan (esimerkin tapauksessa on havaittu kuusi tukiasemaa).




Kuvio 1. Radiotaajuussormenjäljen muodostus

$$M = \begin{bmatrix} LTEID_1 & RSRP_1 \\ LTEID_2 & RSRP_2 \\ LTEID_3 & RSRP_3 \\ WLANID_1 & RSSI_1 \\ WLANID_2 & RSSI_2 \\ WLANID_3 & RSSI_3 \end{bmatrix} \tag{2.1}$$

Ideaalissa tilanteessa mitattavat signaaliparametrit ovat valmiina saatavilla radioverkossa. Esimerkiksi puhelun sisältämien parametrien käyttö on edullista, koska se ei aiheuta radioverkkoon ylimääräistä kuormaa eikä vaadi mobiililaitteeseen mitään muutoksia. Siksi RSS- ja RTD-arvot ovat yleisimmin käytettyjä parametreja. Mobiililaitteet mittaavat määräajoin hallintakanavan RSS-arvoa, jotta ne pystyvät valitsemaan parhaimmin palvelevan solun sekä tarvittaessa vaihtamaan palvelevaa solua. CDM- (*Code Division Multiplexing*) tai TDM- (*Time Division Multiplexing*) tekniikkaa käyttävissä radioverkoissa RTD-arvoa mittaavat säännöllisesti joko ankkurisolu tai mobiililaite itse.

## 2.3 Korrelaatiotietokanta

Korrelaatiotietokanta (*engl. correlation database*) on kokoelma viitesormenjälkiä, joista jokainen on yhdistetty yksilölliseen maantieteelliseen koordinaattiin. Korrelaatiotietokanta muodostetaan paikannusprosessin koulutusvaiheessa. Tietokannan elementtien koordinaateista muodostuva tasojakauma määrittelee tietokannan rakenteen alueella, jolla paikannuspalvelua halutaan tarjota. Tietokannan rakenne voi olla tasainen ruudukko (*engl. uniform grid*) tai järjestetty lista (*engl. indexed list*). Tasaisessa ruudukossa kaikki viitekoordinaatit sijaitsevat tasaisin välimatkoin toisiinsa nähden. Koordinaattien välinen etäisyys kertoo ruudukon välityksen (*engl. grid spacing*) tai tason tarkkuuden (*engl. planar resolution*). Järjestetyssä listassa viitekoordinaatit eivät noudata mitään tiettyä kaavaa. Esimerkiksi kenttämittauksilla toteutettu tietokanta on usein järjestetty lista, koska katuverkon epäsäännöllisyys estää viitekoordinaattien saamisen tasaisin välimatkoin. (Zekavat ja Buehrer 2011)

Yksi suurimpia vaatimuksia radiotaajuuspaikannuksen käytössä on suuren korrelaatiotietokannan luominen ja ylläpitäminen (Laitinen, Lähteenmäki ja Nordström 2001; Mondal, Turkka ja Ristaniemi 2015). Tietokanta muodostetaan keräämällä tietoa kentältä tai luomalla vastaavat tiedot simuloimalla radiosignaalin etenemistä (*engl. signal propagation*) (Zekavat ja Buehrer 2011). Kentällä suoritettavat mittaukset ovat työläämpiä, mutta tuottavat tarkemman sormenjäljen. Myös kenttämittauksien ja simuloitujen sormenjälkien yhdistelmää voidaan käyttää. (Laitinen, Lähteenmäki ja Nordström 2001) Radiotie-etenemisen mallintamisella korrelaatiotietokannan rakentaminen on nopeampaa ja halvempaa, kuin kenttämittauksien tekeminen. Mallintamisella rakennetut tietokannat pystyvät nopeasti mukautumaan ra-

dioverkon muutoksiin. Virheellisistä radiosignaalin etenemisennusteista johtuvia paikannustarkkuuden huononemisia voidaan vähentää kalibroimalla etenemismallia. (Campos ja Lovisolo 2008) Mallintamisella luotua tietokantaa voidaan hienosäätää käyttäen hyväksi kenttämittauksilla saatuja tietoja (Zekavat ja Buehrer 2011).

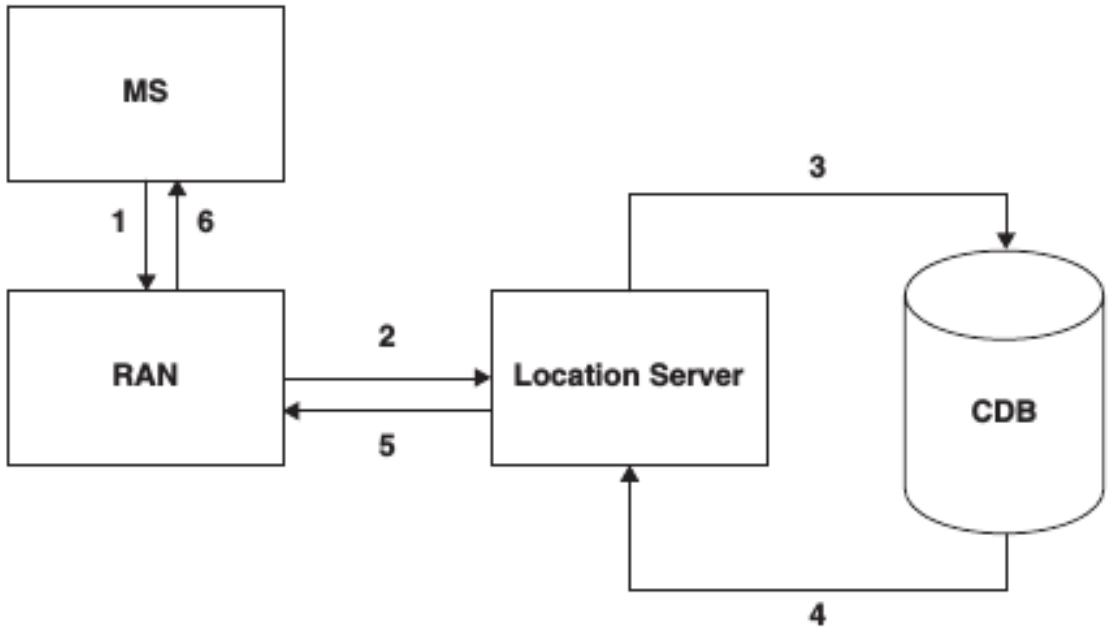
Korrelaatiotietokannan luomisella saavutetaan optimaalinen paikannustarkkuus ympäristöissä, joissa suora näköyhteys tukiaseman ja mobiililaitteen välillä ei ole mahdollinen. Tällaisia ympäristöjä ovat esimerkiksi tiheät kaupunkialueet ja monet sisätilat. Ainoa vaatimus on, että tietokanta on ajan tasalla. Jos tietokantaa ei ole päivitetty, pienet muutokset radioverkossa tai signaalin etenemisympäristössä, esimerkiksi uudet rakennukset, näkyvät heikentyneenä paikannustarkkuutena. Korrelaatiotietokannan luominen tukee myös verkon suunnittelua, koska siinä tarvitaan samanlaista tietoa, kuin mitä korrelaatiotietokanta sisältää. (Laitinen, Lähteenmäki ja Nordström 2001)

Kun radioverkkoympäristössä tapahtuu rakenteellisia muutoksia, esimerkiksi uuden tukiaseman lisääminen tai poistaminen, myös korrelaatiotietokantaa pitää päivittää. Operaattorit suorittavat yleensä määräajoin kattavia ja kalliita testiajoja (*engl. drive test*) päivittääkseen tietokantojaan. (Mondal, Turkka ja Ristaniemi 2015) 3GPP (*Third Generation Partnership Project*) Rel-10 (*Release 10*) esittelee MDT-tekniikan (*Minimization of Drive Tests*), joka mahdollistaa operaattoreille käyttäjien mobiililaitteiden valjastamisen radioverkkomittauksiin ja niihin liittyvien sijaintitietojen keräämiseen. MDT on standardoitu sekä UMTS- että LTE-tekniikoille. (Johansson ym. 2012) Mondal, Turkka ja Ristaniemi 2015 esittelevät GMDT-tekniikan (*Generalized Minimization of Drive Tests*), joka mahdollistaa sekä WLAN- että LTE-signaalivoimakkuuksien keräämisen mobiililaitteelta yhdessä sijaintitiedon kanssa. Kun käytetään mobiiliverkon signaaliparametrien lisäksi WLAN-signaalien voimakkuuksia, radiotaajuuspaikannus tarjoaa erittäin hyvän paikannustarkkuuden varsinkin tiheillä kaupunkialueilla.

## 2.4 Paikannusprosessi

Paikannusprosessissa on kaksi vaihetta: koulutus- ja testausvaihe. Koulutusvaiheessa (*engl. training phase*) luodaan korrelaatiotietokanta ja testausvaiheessa (*engl. testing phase*) tuo-

tetaan mobiililaitteen sijaintiarviot. (Chen ym. 2006) Kuva 2 havainnollistaa paikannusprosessin vaiheita, ja on luotu ETSI 2004-standardin pohjalta. Vaiheessa 1 mobiililaite (MS) lähettää radioverkon (RAN) kautta paikannuspyynnön paikannuspalvelimelle. Vaiheessa 2 radioverkko kommunikoi paikannuspalvelimen kanssa. Paikannuspalvelin saa paikannuspyynnön, joka sisältää mobiililaitteelta saadun sormenjäljen. Kohdassa 3 palvelin kysyy korrelaatiotietokannalta (CDB) vertailusormenjälkiä, jotka se saa kohdassa 4. Paikannuspalvelin vertaa tietokannasta saatuja sormenjälkiä mobiililaitteelta saatuun sormenjälkeen ja arvioi sen perusteella mobiililaitteen paikan. Kohdassa 5 ja 6 arvioitu sijaintitieto lähetetään takaisin mobiililaitteelle. (Zekavat ja Buehrer 2011)




Kuvio 2. Paikannusprosessin vaiheet (Zekavat ja Buehrer 2011)

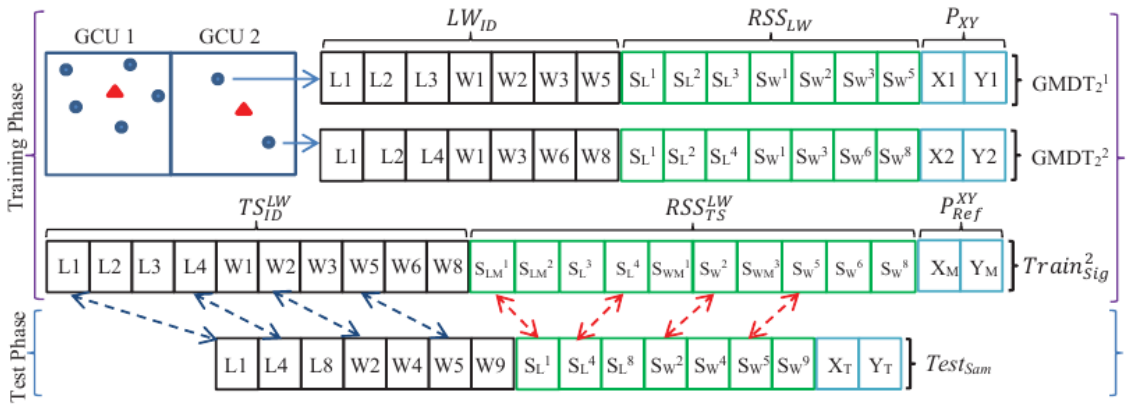
Vaatimuksena DCM-metodeille on, että mobiililaite pystyy lähettämään paikkariippuvaisen sormenjäljen paikannuspalvelimelle. Sormenjälki voi sisältää myös GSM-, UMTS-, WLAN- tai GPS-signaaleja. Paikannuspalvelimen on oltava myös tarpeeksi tehokas, jotta se suoriutuu kaikista paikannuspyynnöistä kohtuullisessa ajassa. Suuren mittakaavan sovelluksissa tämä voi vaatia hajautettua käsittelyä. (Laitinen, Lähteenmäki ja Nordström 2001)

Radiotaajuuspaikannus ei luo karttaa arvioiduista sijainneista eikä se mallinna radiotie-etenemistä (*engl. radio propagation*), vaan se luo hakemiston maantieteellisille koordinaateille radio-

taajuussormenjälkien avulla. Paikannustarkkuus riippuu suuresti kerättyjen sormenjälkien tiheydestä. Mitä tiheämmin sormenjälkiä on kerätty, sitä tarkempi on paikannusarvio. (Chen ym. 2006)

## 2.5 Koulutus- ja testausvaihe

Jos ei toisin mainita, tämän kappaleen teksti perustuu artikkeliin Mondal, Turkka ja Ristaniemi 2015. Kuvassa 3 havainnollistetaan ruudukkopohjaista radiotaajuuspaikannusta (*GRFFP, engl. Grid-based RF Fingerprinting*) jakamalla koko maantieteellinen alue neliönmuotoisiin soluyksiköihin (*GCU, engl. grid-cell unit*). Kuvassa siniset pisteet esittävät GMDT-näytteitä ja punaiset kolmiot vastaavia keskiarvosijainteja. Perinteisessä GRFFP-paikannuksessa yksi GCU sisältää monia koulutusnäytteitä (*engl. training signatures*) (Mondal ym. 2014, 2013). Näistä yhden GCU:n koulutusnäytteistä muodostetaan yksittäinen näyte ($Train_{Sig}$), jolloin vähennetään laskennallista taakkaa sekä nopeutetaan parhaan koulutusnäytteen yhdistämistä testausnäytteeseen. Kaikista yhden GCU:n näytteistä muodostuva ($Train_{Sig}$) määritellään seuraavasti:




Kuvio 3. Koulutus- ja testausvaihe ruudukkopohjaisessa radiotaajuspaikannuksessa (Mondal, Turkka ja Ristaniemi 2015)

$$Train_{Sig}^{i} = \{TS_{ID}^{LW}, RSS_{TS}^{LW}, P_{Ref}^{XY}\} \tag{2.2}$$

missä $TS_{ID}^{LW}$ sisältää kaikki LTE- ja WLAN-tukiasemien tunnukset, $RSS_{TS}^{LW}$ vastaa LTE- ja

WLAN-signaalien voimakkuuksia ja $P_{Ref}^{XY}$ on kaikkien näytteiden keskiarvokoordinaatti.

Kuvasta 3 nähdään, että GCU 2 sisältää kaksi GMDT-näytettä: $GMDT_2^1$ ja $GMDT_2^2$, joiden sisältö on esitetty kahdella rivivektorilla. Mustat neliöt tarkoittavat tukiasemien tunnuksia (L1 LTE-tukiasemalle 1 ja W1 WLAN-tukiasemalle 1), vastaavien tukiasemien signaalivoimakkuudet esitetään vihreillä neliöillä ($S_L^1$ on LTE-tukiasema 1:n RSRP-arvo ja $S_W^1$ on WLAN-tukiasema 1:n RSSI-arvo) ja lopuksi näytteen x- sekä y-koordinaatti on merkitty sinisellä.

$Train_{Sig}^2$ on GCU 2:n GMDT-näytteistä muodostettu koulutusnäyte, jolla on kolme osaa: ensimmäinen osa sisältää kaikki yksilölliset tukiasemien tunnukset, toinen sisältää yhteisien tukiasemien keskiarvolliset signaalien voimakkuudet (jos tietty tukiasema löytyy vain toisesta GMDT-näytteestä, arvo kopioidaan koulutusnäytteeseen) ja kolmas on GMDT-näytteiden keskiarvokoordinaatit.

Paikannusprosessin testausvaiheessa testattavan GMDT-näytteen LTE- ja WLAN-tunnuksia verrataan kaikkiin saatavilla oleviin koulutusnäytteisiin. Koulutusnäytteistä valitaan ne, jotka täyttävät raja-arvon yhteisien tunnuksien määrälle. Esimerkiksi kuvassa 3 oleva $Train_{Sig}^2$ sisältää neljä samaa tukiasematunnusta: L1, L4, W2 ja W5, jotka ovat yhteisiä testinäytteen ($Test_{Sam}$) kanssa. Kuvan testi- ja koulutusnäyte ovat siis 57% samanlaisia, jos raja-arvo asetettaisiin 50%:iin, tulisi $Train_{Sig}^2$ valittua etäisyyden arviointiin. Kuten kuva 3 näyttää (punaiset katkonuolet), vain yhteisiä RSRP- ja RSSI-arvoja käytetään euklidisen etäisyyden (*engl. euclidian distance*, 2.3) määrittelemiseen $Train_{Sig}^2$:n ja $Test_{Sam}$:n välillä.

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \tag{2.3}$$

## 2.6 Aiempi tutkimus

Vaikka erilaisia paikannustekniikoita on olemassa lukuisia, sisä- ja ulkotilamenetelmille ei ole olemassa selkeää luokittelua eri paikannuspalveluiden (*engl. location services*) vaatimuksiin perustuen. Paikannustekniikat voitaisiin jakaa myös itsepaikantaviin (*engl. self-positioning*) ja etäpaikantaviin tekniikoihin riippuen siitä käyttääkö mobiililaite antennien ja

yhdyskäytävien (*engl. gateways*) lähettämiä signaaleja oman sijaintinsa määrittämiseen, vai käyttääkö se erilaisten vastaanottimien välillä kulkevia signaaleja laskeakseen oman sijaintinsa signaalien voimakkuuksiin ja/tai suuntiin perustuen. (Zeimpekis, Giaglis ja Lekakos 2002) Edellä on kuitenkin listattu muutamia paikannusmenetelmiä niiden käyttökohteiden perusteella.

### 2.6.1 Ulkotilamenetelmät

Triangulaatio (*engl. triangulation*) hyödyntää kolmion geometrisia ominaisuuksia kohteen paikantamiseen. Triangulaatiosta on kaksi versiota: lateraatio (*engl. lateration*) ja angulaatio (engl. angulation). Lateraatiossa arvioidaan kohteen sijainti laskemalla sen etäisyys useammasta viitepisteestä, jotka ovat yleensä radioverkon tukiasemia. Kohteen sijainti viitepisteistä voidaan laskea esimerkiksi TOA- (*Time of Arrival*) tai TDOA- (*Time Difference of Arrival*) tekniikoilla. Angulaatiossa kohteen sijainti saadaan kohteen ja viitepisteiden välisen ympyräsäteiden leikkauskohdasta. Angulaatio tarvitsee vähintään kaksi viitepistettä (tukiasemaa) ja kaksi ympyräsäteen kulmaa, jotta halutun kohteen sijainti voidaan määrittää kaksiulotteisessa ympäristössä. (Liu ym. 2007)

Ahonen ja Laitinen 2003 käyttävät DCM-metodia yhdessä PD-profiilin kanssa määritelläkseen mobiililaitteen sijainnin kaupunkiympäristön UMTS-verkossa. Menetelmän suorituskykyä vertaillaan OTDOA- (*Observed Time Difference of Arrival*, (Spirito 2001)) ja Cell ID- (*Cell Identification*) tekniikoihin. DCM-metodia käyttäen, tutkimuksessa päästin 25 metrin paikannustarkkuuteen 67%:ssa tuloksista. Tulos on riittävä monille paikannussovelluksille. Vastaavasti OTDOA-tekniikan tarkkuus oli 97 metriä (67%), ja Cell ID-tekniikalla päästiin 152 metrin tarkkuuteen, kun vain yksi tukiasema oli kuultavissa.

Laitinen, Lähteenmäki ja Nordström 2001 käyttivät tutkimuksessaan DCM-metodia paikantaakseen mobiililaitteita GSM-verkossa sekä kaupunki- että esikaupunkialueella. Paikannustuloksia vertaillaan E-OTD- (*Enhanced Observed Time Difference*) ja AOA- (*Angle of Arrival*) tekniikoihin. Kaupunkialueella DCM-metodi yltää 44 metrin tarkkuuteen 67%:ssa tuloksista ja 90 metriin 90%:ssa tuloksista. Vastaavasti esikaupunkialueella DCM pääsee 74 metrin ja 190 metrin tarkkuuksiin. E-OTD- tekniikka yltää kaupunkialueella 141 metrin ja

237 metrin tarkkuuksiin ja esikaupungissa 125 metrin ja 231 metrin tarkkuuksiin. Esikaupunkialueella AOA-tekniikalla päästään 45 metrin tarkkuuteen 67%:ssa tuloksista ja 89 metriin 90% tuloksista.

Trevisani ja Vitaletti 2004 tutkivat Cell-ID -tekniikan tarkkuutta mobiililaitteen paikannuksessa. Tekniikka perustuu GPS:n määrittelemään mobiililaitteen tarkkaan sijaintiin sekä tukiaseman sijaintitietoihin. Paikannustarkkuus saadaan näiden kahden välisistä etäisyyksien keskiarvoista. Tekniikka on siis sitä epätarkempi, mitä suurempia mobiilisolut ovat. Parhaimpiin tuloksiin päästään siis kaupunkialueilla. Tutkimustuloksia esitetään Yhdysvalloista sekä Italiasta kaupunki-, esikaupunki- sekä päätiealueilla. Kaupunkialueella paikannustarkkuus oli Italiassa noin 500 metriä ja Yhdysvalloissa noin 800 metriä. Esikaupunkialueella päästiin Italiassa noin 750 metrin tarkkuuteen ja Yhdysvalloissa noin 490 metrin tarkkuuteen. Lopuksi päätiealueiden tarkkuudet olivat Italiassa noin kilometri ja Yhdysvalloissa noin 2,9 kilometriä.

Vidal, Brooks ym. 2002 käyttämä AOA-tekniikka perustuu tukiasemalle tulevan signaalin kulmasta tukiaseman vastaanottoantenniin nähden. Mobiililaitteen sijainti voidaan määritellä, jos signaalin tulokulma saadaan kahdesta tai useammasta tukiasemasta. 70% paikannustuloksista antoi tarkemman tuloksen, kuin 10 metriä ja 85%:ssa tuloksista tarkkuus oli alle 25 metriä.

### 2.6.2 Sisätilamenetelmät

RADAR on radiotaajuuspaikannukseen perustuva tekniikka käyttäjien paikantamiseen sisätiloissa. Se perustuu tukiasemilta saataviin signaalinvoimakkuustietoihin, joita yhdistetään mittaustietoihin sekä radiosignaalin etenemismallinnuksiin. RADAR-tekniikka mahdollistaa käyttäjien paikantamisen 2-3 kolmen metrin tarkkuudella, mikä on suunnilleen samaa suuruusluokkaa, kuin tyypillinen toimistohuone. (Bahl ja Padmanabhan 2000) Artikkeli Bahl, Padmanabhan ja Balachandran 2000 käsittelee RADAR-tekniikan ongelmia ja ehdottaa niihin konkreettisia parannuksia, jotka mahdollistavat tarkemman paikannuksen ja tekevät järjestelmästä muutoskykyisemmän.

Varshavsky ym. 2007 esittelee paikannustekniikan, joka hyödyntää GSM-signaaleista muo-

dostettuja sormenjälkiä. Sen perustana on koulutusvaihe, jossa ympäristöstä muodostetaan radiotaajuuskartta suorittamalla mittauksia useissa kohteissa. Menetelmän tarkkuuden mahdollistaa laajan sormenjäljen käyttö. Sormenjälki sisältää kuuden voimakkaimman GSM-signaalin lisäksi lukemia jopa 29 GSM-kanavasta, jotka ovat tarpeeksi voimakkaita tullakseen havaituksi, mutta liian heikkoja käytettäväksi tehokkaaseen kommunikointiin. Menetelmää testattiin kolmessa eri rakennuksessa: keskusta-alueen hotellissa (9-kerrosta), yliopiston hotellissa (12-kerrosta) sekä Tartu-rakennuksessa (16-kerrosta). Paikannustarkkuus vaihteli 1,94 metrin ja 4,07 metrin välillä, oikea kerros löytyi 60%:ssa tapauksista ja kerroksen päähän menetelmä päätyi 98%:ssa tapauksista.

Youssef ja Agrawala 2005 esittelevät artikkelissaan Horus-järjestelmän, joka käyttää eri vaiheita käyttäjän sijainnin määrittämiseen. Offline-vaiheessa rakennetaan radiotaajuuskartta tukiasemien signaalivoimakkuuksista ja esikäsitellään signaalitietoa. Online-vaiheessa arvioidaan käyttäjän sijainti kuultavissa olevien tukiasemien ja radiotaajuuskartan avulla. Horus-järjestelmä ottaa huomioon tekijät, jotka aiheuttavat vaihteluita langattomaan signaaliin ja käyttää tekniikoita, jotka vähentävät näitä vaihteluita. Horus-järjestelmä oli ensimmäisessä testissä 89% tarkempi kuin RADAR ja toisessakin testissä 82% parempi.

Gwon ja Jain 2004 esittelevät kalibroimattoman TIX-algoritmin (*Triangular Interpolation and extrapolation*) ja analysoivat kalibrointipohjaisten algoritmien virheominaisuuksia. TIX-algoritmi saavuttaa 5,4 metrin keskiarvotarkkuuden, joka on kilpailukykyinen kalibroitavien algoritmien 4,7 metrin tarkkuuteen verrattuna. Vertailukohteena on kalibrointipohjaisista algoritmeista triangulointi, Knn-algoritmi ja SMP-menetelmä (*Smallest M-vertex polygon*). Kalibroimattomilla paikannusalgoritmeilla voidaan eliminoida työläät offline-/koulutusvaiheen signaalivoimakkuusmittaukset.

Myöskään Lim ym. 2005 luoma paikannustekniikka ei vaadi esikonfigurointia. Tämän mahdollistaa täysin automaattinen kalibrointimekanismi, joka luo riippuvuuden signaalivoimakkuuksien ja maantieteellisten etäisyyksien välille. Paikannusalgoritmi käyttää syötteenään reaaliaikaisia RSS-mittauksia sekä eri tukiasemien välillä että mobiililaitteen ja sen lähimpien tukiasemien välillä. Menetelmällä on päästy noin kolmen metrin tarkkuuksiin.

# 3 Klusterointi

Tässä luvussa käydään yleisesti läpi klusterointia tiedonlouhintatekniikkana, sen yleisimpiä käyttötarkoituksia, klusteroinnin määritelmiä ja periaatetta. Lisäksi kerrotaan klusteroinnin yleisestä jaottelusta sekä klusteroinnin pääpiirteisistä askeleista. Klusteroinnin käsitystä, sen askeleita ja klusterointikriteerin valintaa on lisäksi havainnollistettu kuvilla.

## 3.1 Käyttötarkoitus

Digitaalisen datan tuottamiseen tarkoitettujen laitteiden yleinen saatavuus, varastointiteknologioiden kehitys, digitaalinen kuvantaminen sekä internethakujen määrän nousu ovat luoneet perustan saatavilla olevan tiedon räjähdysmäiselle lisääntymiselle. Tätä digitaalista informaatiota säilytetään monissa erittäin suurissa tietokannoissa, jotka luovat suuren potentiaalin automattisen data-analyysin, luokittelun ja tiedonhaun kehittämiselle. Informaation määrän lisäksi myös sen monimuotoisuus (ääni, video ja kuva) on kasvanut. (Jain 2010)

Klusteroinnin tarkoituksena on löytää ryhmiä ja tunnistaa mielenkiintoisia jakaumia sekä malleja tarkasteltavana olevasta datasta, ja se on tiedonlouhintaprosessissa yksi hyödyllisimmistä keinoista tähän tarkoitukseen (Halkidi, Batistakis ja Vazirgiannis 2001). Klusterointi myös auttaa ihmisiä analysoimaan, kuvailemaan ja hyödyntämään ryhmiin piilotettua arvokasta tietoa. Lisäksi sillä voidaan luoda prototyyppejä ja tyypillisiä objekteja kuvaamaan kaikkia yhden klusterin objekteja (Wu 2012).

Monesti klusterointi toimii johdantona muille tiedonlouhinta- ja mallinnusmenetelmille. Esimerkiksi markkinoiden segmentoinnissa voidaan koko asiakaspohja ensi jakaa ostotottumuksien perusteella klustereihin, joihin jokaiseen voidaan kohdistaa erilainen ja parhaiten toimiva markkinointistrategia. (Berry ja Linoff 2004) Klusterointi toimii myös vieraiden havaintojen (*engl. outliers*) löytämiseen tapauksissa, joissa ne voivat olla mielenkiintoisempia, kuin tavalliset havainnot. Klusteroinnissa vieraat havainnot ovat arvoja, jotka ovat "kaukana"kaikista klustereista. Tällaisia tapauksia voi esiintyä esimerkiksi luottokorttipetoksien havaitsemisessa ja rikoksien ennaltaehkäisyssä sähköisessä kaupankäynnissä. (Han ja Kamber 2001)

Theodoridis ja Koutroumbas 2006 tiivistävät teoksesta Everitt, Landau ja Leese 2009 seuraavat tärkeät käyttötarkoitukset klusteroinnille:

- Tiedon vähentäminen. Monessa tapauksessa käsiteltävän datan määrä on suuri, jolloin sen prosessointi on erittäin vaativaa. Klusteroinnilla voidaan vähentää käsiteltävän datan määrää analysoimalla kunkin klusterin edustajaobjektia.
- Hypoteesin luominen. Klusterointia voidaan käyttää hyödyksi, kun halutaan tehdä johtopäätöksiä käsiteltävästä datasta.
- Hypoteesin testaus. Klusteroinnin avulla voidaan testata hypoteesien paikkansapitävyyttä. Esimerkiksi hypoteesia "isot yritykset investoivat ulkomaille"voidaan testata klusteroimalla yrityksiä käsittelevää dataa. Jos klusteroinnin tuloksena saadaan klusteri, joka korreloi isojen ja ulkomaisia investointeja tekevien yrityksien kanssa, klusterointi tukee hypoteesia.
- Ryhmiin perustuva ennustaminen. Kunkin klusterin sisältämät objektit määrittelevät itse klusteria. Tämän avulla voidaan tuntemattomalle objektille määrätä todennäköisin klusteri, jolloin tuntematonta objektia voidaan kuvailla sille määrätyn klusterin avulla.

## 3.2 Määritelmä ja periaate

Klusterointi (*engl. clustering*) eli ohjaamaton luokittelu (*engl. unsupervised classification*) on ihmisille lajityypillinen toimintatapa, jonka ansiosta opimme jo varhaisessa lapsuudessa erottamaan kissat koirista ja eläimet kasveista. Tämä tapahtuu kehittämällä jatkuvasti meidän alitajuntaista klusterointijärjestelmäämme. Lisäksi automaattisella klusteroinnilla voidaan tunnistaa kohdeavaruudesta tiheät ja väljät alueet, joiden avulla on mahdollista löytää yleisiä jakaumia ja mielenkiintoisia korrelaatioita eri attribuuttien välillä. (Han ja Kamber 2001) Tiedon klusterointia voidaan hyödyntää muun muassa tiedonhaussa, biologiassa, ilmastotutkimuksessa, psykologiassa, lääketieteessä ja liiketoiminnassa. (Pang-Ning, Steinbach, Kumar ym. 2006)

Termille ’klusteri’ on ehdotettu useita eri määritelmiä vuosien saatossa (Everitt, Landau ja Leese 2009; Wallace ja Boulton 1968). Suurin osa näistä määritelmistä perustuu kuitenkin termiin ’samankaltainen’, joka on melko löyhästi määritelty, tai ne sopivat vain tietynlaisil-

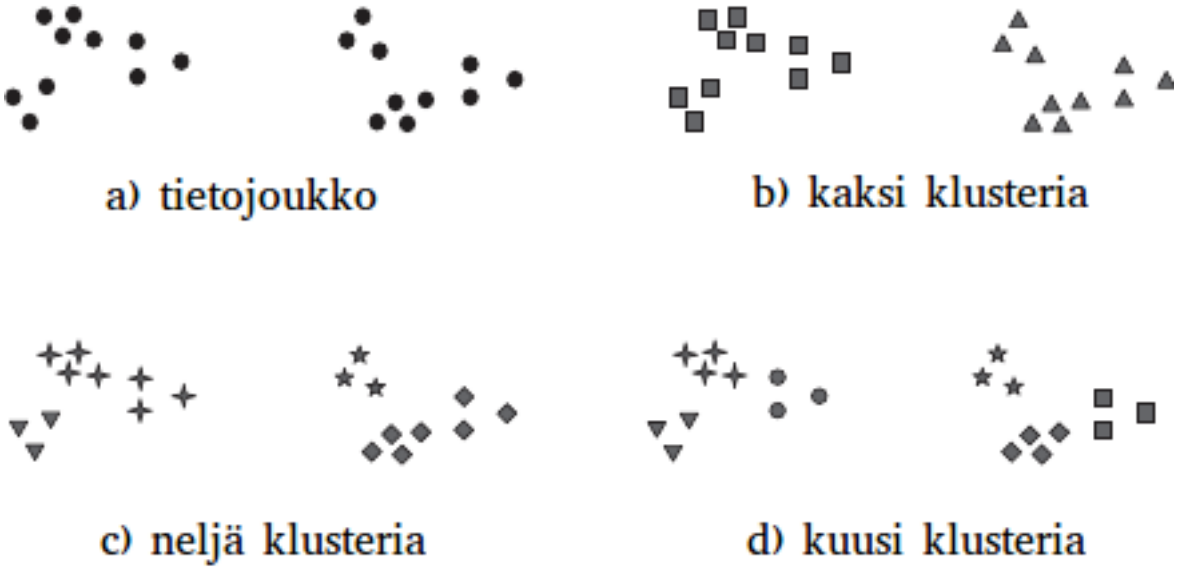
le klustereille (Theodoridis ja Koutroumbas 2006). Romesburg 2004 määrittelee klusterin seuraavalla tavalla: klusteri on joukko, joka sisältää yhden tai useamman objektin, joita haluamme pitää samanlaisina keskenään. Sen mukaan klusteri voi myös koostua vain yhdestä objektista, jos emme halua pitää muita objekteja samanlaisina. Voimme pitää myös kaikkia objekteja samanlaisia, jolloin klusteri koostuu kaikista käsiteltävistä objekteista. Artikkelissa mainitaan myös sanan 'haluta' käyttämisestä tässä yhteydessä, mikä voi kuulostaa oudolta, mutta se on juurikin oikea sana. Lopuksi siinä painotetaan, että kutsuaksemme kahta tai useampaa objektia samanlaisiksi, meidän täytyy jättää huomioimatta joitakin ominaisuuksia, mitkä tekevät niistä erilaisia.

Klusteroinnin periaate voidaan määritellä seuraavasti: jaa annettu data *d* klustereihin *k* siten, että jokainen datapiste klusterin sisällä muistuttaa enemmän toisia pisteitä saman klusterin sisällä, kuin eri klusterin sisällä. (Guha, Rastogi ja Shim 2001; Wallace ja Boulton 1968) Mitä enemmän saman klusterin sisältämät datapisteet muistuttavat toisiaan, eli mitä homogeenisempia ne ovat, ja mitä enemmän ne eroavat muiden klustereiden datapisteistä, sitä parempi tai selkeämpi klusterointi. (Pang-Ning, Steinbach, Kumar ym. 2006) Klusterointi eroaa luokittelusta siten, että se ei riipu ennaltamääritellyistä luokista. Luokittelussa jokaiselle objektille asetetaan ennaltamääritelty luokka perustuen malliin, joka on luotu esiluokiteltujen esimerkkien avulla. (Berry ja Linoff 2004)

Klusteroinnin käsitys voi olla monitulkintainen, kuten kuva 4 havainnollistaa. Kuviossa samanmuotoiset kuuluvat samaan klusteriin. Kuvion tarkoituksena on painottaa klusterin määritelmän epätarkkuutta ja sitä, että paras määritelmä klusterille riippuu käsiteltävän datan luonteesta sekä halutuista tuloksista. (Pang-Ning, Steinbach, Kumar ym. 2006) Klusterointimenetelmien kirjo on kattava ja usein myös sekava, koska datan esittämiselle, dataobjektien samankaltaisuuden arvioinnille ja itse datan klusteroinnille on olemassa erittäin monia eri tekniikoita. (Jain, Murty ja Flynn 1999)

## 3.3 Jaottelu

Klusterointi voidaan yleisesti jakaa kahteen osaan: hierarkkiseen ja osittavaan. Hierarkkisessa klusteroinnissa (*engl. hierarchical clustering*) luodaan rekursiivisesti sisäkkäisiä klusterei-




Kuvio 4. Eri tapoja klusteroida sama joukko dataa (Pang-Ning, Steinbach, Kumar ym. 2006)

ta joko yhdistelemällä (*agglomeratiivinen, engl. agglomerative*) samanlaisia datapisteitä tai jakamalla (*erotteleva, engl. divisive*) isompia klustereita pienempiin. Osittavassa klusteroinnissa (*engl. partitional clustering*) puolestaan kaikki klusterit muodostuvat samanaikaisesti ja jokainen klusteri esittää tiettyä osaa käsitellystä datasta, eivätkä siten sisällä hierarkkisia rakenteita. (Jain 2010) Osittava klusterointi pyrkii joko maksimoimaan tai minimoimaan ennaltamäärätyn klusterointikriteerin, esimerkiksi klusterikeskuksen ja muiden klusteriobjektien välisen neliövirheen minimointi. Hierarkkisessa klusteroinnissa sekä agglomeratiiviset että erottelevat metodit klusteroivat dataa läheisyysmatriisin mukaan. Tuloksena on yleensä binääripuu tai dendrogrammi. (Xu ja Wunsch 2009)

Lisäksi voidaan puhua myös eksklusiivisesta, päällekkäisestä sekä sekavasta klusteroinnista (*engl. fuzzy clustering*). Eksklusiivinen klusterointi määrää jokaisen objektin yhteen klusteriin. Joskus on kuitenkin mielekästä, että tiettyjä objekteja voidaan määrätä useampaan klusteriin, tällöin puhutaan päällekkäisestä klusteroinnista. Sekavassa klusteroinnissa jokainen objekti kuuluu jokaiseen klusteriin, mutta objekteilla on kullakin oma painoarvonsa klusterissa. Painoarvo vaihtelee välillä 0 (ei missään nimessä kuulu klusteriin) - 1 (kuuluu ehdottomasti klusteriin). Täydellisessä klusteroinnissa jokainen objekti määrätään johonkin klusteriin, kun taas osittaisessa klusteroinnissa ei. Osittaisen klusteroinnin tarkoituksena on jät-

tää klusteroimatta "huonot"objektit, jotka voivat vääristää klusterointiprosessia. (Pang-Ning, Steinbach, Kumar ym. 2006)

## 3.4 Klusterointiprosessi

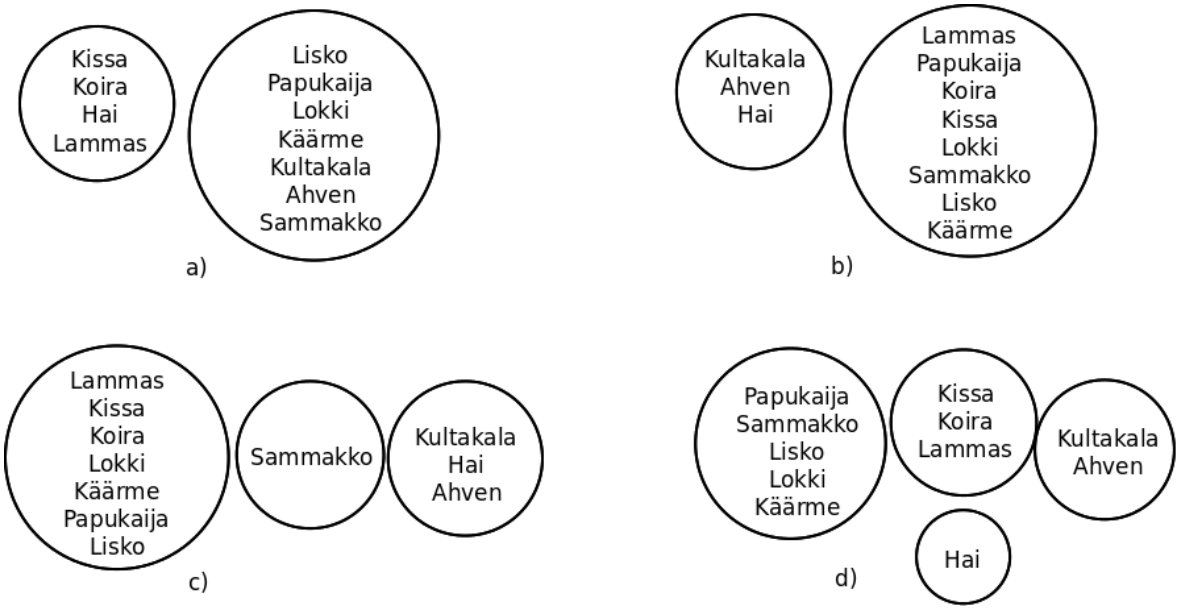
Jain, Murty ja Flynn 1999 jakavat yleisesti klusterointiprosessin viiteen osaan. Kuva 5 havainnollistaa prosessia osittain:

1. Käsiteltävän tiedon esittäminen
2. Klusterointikriteerin määrittely
3. Klusterointi
4. Tiedon yleistäminen (vaihtoehtoinen)
5. Tulosteen arviointi (vaihtoehtoinen)


Kuvio 5. Klusteroinnin vaiheet

Ennen käsiteltävän tiedon vertailukriteerien määrittelyä ja klusterointia, on se esitettävä käsiteltävässä muodossa. Tämä voi tarkoittaa esimerkiksi tiedon tyypin ja/tai koon määrittelemistä. Tässä kohdassa voidaan myös määritellä, että datasta valitaan käsiteltäväksi vain tietyt attribuutit. Seuraavaksi valitaan käsiteltävälle datalle sopiva klusterointikriteeri, jonka perusteella objektien samankaltaisuutta arvioidaan. Klusterointi voidaan tämän jälkeen toteuttaa monella tavalla riippuen käsiteltävän tiedon luonteesta. Sopivan klusterointitavan valinta on tärkeää, sillä eri menetelmillä saadaan samasta datasta ulos erilaisia klustereita. (Jain, Murty ja Flynn 1999)

Klusterointikriteerin valinta riippuu siitä, mitä klusteroinnilla halutaan saavuttaa. Erilaiset kriteerit johtavat erilaisiin klusterointituloksiin. Kuva 6 havainnollistaa kriteerin valinnan vaikutusta. A-kohdassa klusteroinnin kriteerinä on synnytystapa, b-kohdassa keuhkojen olemassaolo, c-kohdassa elinympäristö (meri) ja d-kohdassa synnytystapa sekä keuhkojen olemassaolo. (Theodoridis ja Koutroumbas 2006)




Kuvio 6. Klusterointikriteerin valinta vaikuttaa klusterointitulokseen

Tyypillisin mitta eri objektien samankaltaisuuden arvioinnille on niiden välinen etäisyys. Yksi vaihe klusterointiprosessissa voi olla esimerkiksi sopivan etäisyysfunktion määrittteleminen ja kaikkien objektiparien välisen etäisyyden laskeminen. Jos etäisyys on hyvä samankaltaisuuden mitta (*engl. similarity measure*), saman klusterin objektit ovat lähellä toisiaan ja kaukana muiden klustereiden objekteista. Kahden objektin vertaamiseen voidaan käyttää myös yhtäläisyysfunktiota, jonka antama suuri arvo ilmaisee, että vertailtavat objektit ovat jokseenkin samanlaisia. Esimerkiksi objektien välisen kulman kosini voisi olla sopiva yhtäläisyysfunktio tapauksessa, jossa objektein väliset kulmat ovat mielekkäitä. Lisäksi kahden objektin välistä yhteisten attribuuttien suhdetta attribuuttien määrään voidaan pitää yhtenä samankaltaisuuden mittana. (Duda, Hart ja Stork 2000)

Klusteroinnin jälkeen voidaan saatujen klustereiden sisältämää tietoa yleistää helpommin käsiteltävään ja havainnollistettavampaan muotoon. Tyypillisesti yleistäminen toteutetaan joko klusteriprototyypillä tai klusterille tyypillisellä dataobjektilla, mitkä niin sanotusti edustavat yhden klusterin kaikkia objekteja. Klusterointituloksien arvioiminen voi olla hankalaa, sillä "hyvän"klusterin määritelmä on usein monitulkintainen ja riippuu yleensä datan luonteesta ja sen käyttötavasta. Kaikki klusterointialgoritmit löytävät annetusta datasta klustereita riippumatta siitä onko niitä datassa jo valmiina vai ei. Jos annettu data sisältää klustereita, toiset

algoritmit antavat "parempia"tuloksia kuin toiset. On myös järkevää pohtia, että kannattaako dataa, joka ei sisällä klustereita, prosessoida klusterointialgoritmeilla. (Jain, Murty ja Flynn 1999)

# 4 Klusterointialgoritmit

Tässä luvussa on tarkoitus kertoa tarkemmin tässä työssä käytetyistä klusterointialgoritmeista, joita ovat K-means, Knn ja K-medoids. Jokaiselle algoritmille on oma kappaleensa. Lisäksi kahdesta ensin mainitusta algoritmista perehdytään niiden toimintaan oleellisesti vaikuttavan attribuutin valintaan sekä niiden toiminnassa ilmeneviin ongelmiin.

## 4.1 K-means

K-means-algoritmi (MacQueen ym. 1967; Forgy 1965) on yksi vanhimmista ja laajimmin käytössä olevista klusterointialgoritmeista. Se on yksinkertainen ja prototyyppipohjainen osittava klusterointialgoritmi, joka yrittää muodostaa $k$ ei-päällekkäistä klusteria. Jokaista muodostunutta klusteria edustaa sentroidi (*engl. centroid*). Sentroidi on tyypillisesti keskiarvo klusterin objekteista. (Wu 2012) K-means pyrkii minimoimaan sekä euklidisen etäisyyden lähimpään sentroidiin (neliövirhe) että kaikkien klustereiden neliövirheen summan (Yhtälö 4.1). (Jain 2010; Duda, Hart ja Stork 2000; Pang-Ning, Steinbach, Kumar ym. 2006)

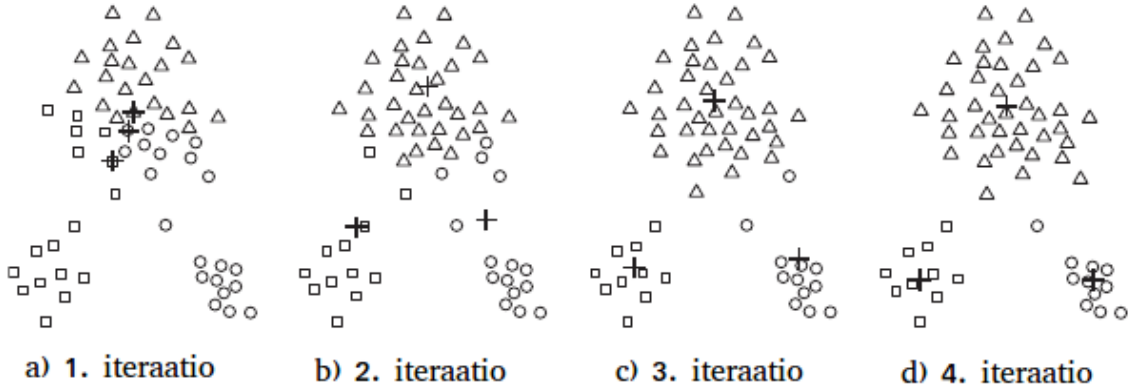
$$J_e = \sum_{i=1}^{c} \sum_{x \in D_i} ||x - m_i||^2 \tag{4.1}$$

Yhtälössä 4.1 klusterin $D_i$ keskiarvovektori $m_i$ edustaa kaikkia klusterin $D_i$ objekteja minimoimalla virhevektorien $x - m_i$ neliön summan klusterissa $D_i$. $J_e$ mittaa siis neliövirheen summaa esitettäessä $n$ objektia $x_1, ..., x_n$ clusterikeskuksien $m_1, ..., m_c$ avulla. $J_e$:n arvo riippuu objektien klusteroinnista ja klustereiden lukumäärästä. $J_e$:n minimoiva klusterointi on optimaalisin. (Duda, Hart ja Stork 2000)

Jain, Murty ja Flynn 1999, MacQueen ym. 1967 ja Ishioka 2000 määrittelevät K-means-algoritmin vaiheet (7) seuraavasti:

1. Valitaan (yleensä satunnaisesti) $k$ pistettä tai objektia toimimaan ensimmäisinä sentroideina.
2. Määrätään jokainen objekti lähimmälle sentroidille.

3. Lasketaan uudet sentroidit vallitsevien klusteriobjektien perusteella.
4. Jos konvergenssiehto ei täyty, mene kohtaan 2. Tyypillisiä konvergenssiehtoja ovat esimerkiksi: objektit eivät siirry klusterista toiseen uusien sentroidien laskemisen jälkeen, tai neliövirheen väheneminen on minimaalista.




Kuvio 7. K-means algoritmin iteraatiot (Pang-Ning, Steinbach, Kumar ym. 2006)

K-means-algoritmissa objektien samankaltaisuutta mitataan yleensä niiden etäisyyden perusteella. Usein käytetään euklidista etäisyyttä (2.3), kun taas kahden vektorin välinen kulma (*engl. cosine similarity*) toimii paremmin asiakirjoille. Erilaiselle datalle saattaa kuitenkin löytyä useampi metodi objektien välisen samankaltaisuuden mittaamiseen. Esimerkiksi Manhattan-etäisyyttä voidaan käyttää euklidiselle tiedolle ja Jaccard-mittaa voidaan myös käyttää asiakirjoille. (Pang-Ning, Steinbach, Kumar ym. 2006)

K-means toimii hyvin moniin käytännön ongelmiin. Erityisen hyvin se suoriutuu tapauksessa, jossa saadut klusterit ovat kompakteja ja muodoltaan hyperpallomaisia (*engl. hyperspherical*). (Xu ja Wunsch 2009; Duda, Hart ja Stork 2000) Se on suosittu algoritmi, sillä se on helppo implementoida ja sen aikavaativuus on O($n$), missä $n$ on käsiteltävien objektien määrä. Suuri ongelma tämän algoritmin kanssa on kuitenkin se, että se on herkkä 1. vaiheen suhteen ja voi lähestyä vain paikallista minimiä, jos ensimmäiset sentroidit on valittu huonosti. (Jain, Murty ja Flynn 1999) K-means-algoritmin kanssa voidaan käyttää toistomenetelmää, jolloin sen 1. vaihe toistetaan siten, että klusterointiparametrit säilyvät. Näistä toistoista voidaan siten valita sopivin asetelma ensimmäisiksi sentroideiksi. (Lamrous ja TaÏleb

2006; Oughdi ym. 2006)

K-means-algoritmi vaatii kolme käyttäjän määrittelemää parametria: klusterien määrä $K$, jota pohjustetaan kappaleessa 4.1.1, klusterialustuksen ja etäisyysmitan. Vaikkei $K$:n valintaan ole olemassa täydellistä matemaattista kriteeriä, useita heuristiikkoja sen valitsemiseen on. Tyypillisesti K-means ajetaan itsenäisesti useilla $K$:n arvoilla, jonka jälkeen parhaimmalta vaikuttava klusterointi valitaan. (Jain 2010)

### 4.1.1 Klusterimäärän valinta

Klustereiden määrää pohtiessa, on hyvä kiinnittää huomiota seuraaviin asioihin. Kun halutaan määrittää $K$:lle useita arvoja, on tärkeää, että arvoja määritellään verrattain paljon. Näin käsiteltävän tiedon eri ominaisuudet käyvät helpommin ilmi. Lisäksi valittujen arvojen määrän pitäisi olla selkeästi pienempi, kuin käsiteltävän tiedon sisältämien arvojen määrä, jotta klusteroinnin perimmäinen tarkoitus säilyisi. (Pham, Dimov ja Nguyen 2005) Erilaisia tekniikoita klusterimäärän valitsemiseksi on olemassa erittäin paljon, tässä kappaleessa niistä mainitaan vain muutamia.

Monet K-means-algoritmin implementoinnit ja niitä käyttävät ohjelmistot (The Mathworks, Inc. 2015) vaativat, että käyttäjä itse määrittelee klustereiden määrän. Löytääkseen tyydyttävän tuloksen, on käyttäjän yleensä ajettava algoritmia $K$:n eri arvoilla. Klusterointituloksen validointi suoritetaan yleensä silmämääräisesti ilman formaaleja menetelmiä. Tämä aiheuttaa hankaluuksia etenkin moniulotteisten tietojoukkojen arvioinnissa. (Pham, Dimov ja Nguyen 2005)

Kun K-means-algoritmia käytetään tiedon esikäsittelytyökaluna, varsinainen tiedonlouhinta-algoritmi määrittelee klustereiden määrän (Hansen ja Larsen 1996). Tällöin klusterointituloksiin pääalgoritmin tehokkuuden kannalta ei kiinnitetä huomiota. Synteettiset tietojoukot, joita käytetään klusterointialgoritmien testaamiseen, on yleensä luotu käyttäen tasajakaumageneraattoreita. Tällöin klustereiden määrä on sama kuin synteettisen datan luomiseen käytettyjen generaattoreiden määrä. Oletuksena on, että jokainen saatu klusteri kattaa jonkun tietyn generaattorin luomat objektit, jolloin klusteroinnin tehokkuuden määrää klustereiden kattamien objektien ja generaattoreiden luomien objektien erotus. Valitettavasti tätä metodia

ei kuitenkaan voida käyttää käytännön ongelmien ratkaisemiseen, koska yleensä jakauma ei ole tiedossa eikä generaattoreiden lukumäärää voida määritellä. (Pham, Dimov ja Nguyen 2005)

On olemassa myös muutamia tilastollisia keinoja klustereiden määrän ($K$) valitsemiseen. Esimerkiksi X-means- algoritmi (Pelleg, Moore ym. 2000; Ishioka 2000), joka puolittaa saatuja klustereita ja suorittaa näille puolikkaille paikallisen K-means-algoritmin. Tämän jälkeen saatuja puolikkaita ja alkuperäistä klusteria vertaillaan keskenään BIC- (textitBayesian information criterion) tai AIC- (textitAkeike's information criterion) arvojen mukaan. Isomman BIC- tai AIC-arvon saanut klusterointi valitaan lopputulokseen. Hardy 1996 tutkii seitsemän eri metodin suoriutumista "oikeaa"klusterimäärää laskettaessa käyttäen jokaisessa metodissa samaa dataa. Samalla kehottaen käyttämään useita klusterointitekniikoita määrittelemään sopivaa klusterimäärää. Lisäksi hän kannustaa analysoimaan kaikkia tuloksia, jotta klustereista saataisiin mahdollisimman paljon tietoa irti.

Kothari ja Pitts 1999 loivat naapurustoasteikkoon perustuvan menetelmän, jossa sentroidit, jotka sattuvat "voittavan"klusterin läheisyyteen, päivitetään lähestymään tätä "voittavaa"klusteria yhdistäen näin molemmat klusterit. Pienellä naapurustoasteikon arvolla tuloksena on enemmän erillisiä klustereita ja suurella arvolla saadaan pienempi määrä klustereita. Laskennallisesti tämä keino on hieman intensiivisempi kuin perinteinen K-means-algoritmi, mutta sietää paremmin vieraita havaintoja. Algoritmi toimii tehokkaasti sekä synteettisillä että luonnollisilla tietojoukoilla.

### 4.1.2 Ongelmia

Pena, Lozano ja Larranaga 1999 määrittelevät K-means-algoritmin keskeisimmiksi ongelmiksi seuraavat asiat.

- Monien muiden klusterointialgoritmien tapaan K-means olettaa, että käsiteltävän datan sisältämä klustereiden määrä on tiedossa etukäteen. Tämä ei tietenkään ole mahdollista monessakaan käytännön sovelluksessa.
- Iteratiivisena tekniikkana K-means on erityisen herkkä alkuasetelman suhteen (ensimmäisten sentroidien valinta).

- K-means lähestyy paikallista minimiä. Algoritmin suorittaminen luo deterministisen kartoituksen alkuasetelmasta lopputulokseen.

Ennakkotiedon puutetta voidaan korvata karkealla, mutta melko yleisellä tavalla suorittaen algoritmi $K$:n eri arvoilla. Tämä on laskennallisesti kuitenkin erittäin vaativa toimenpide, koska tiedonlouhintaprosesseissa käsitellään yleensä suuria tietomääriä (Davidson ja Ravi 2005). Alkuasetelmaongelma ei ole yksinään K-means-algoritmin ongelma, vaan yleinen mäennousualgoritmien (*engl. hill-climbing*) keskuudessa. Niiden deterministinen käyttäytyminen johtaa paikalliseen minimiin riippuen alkuasetelmasta. (Pena, Lozano ja Larranaga 1999)

## 4.2 Knn

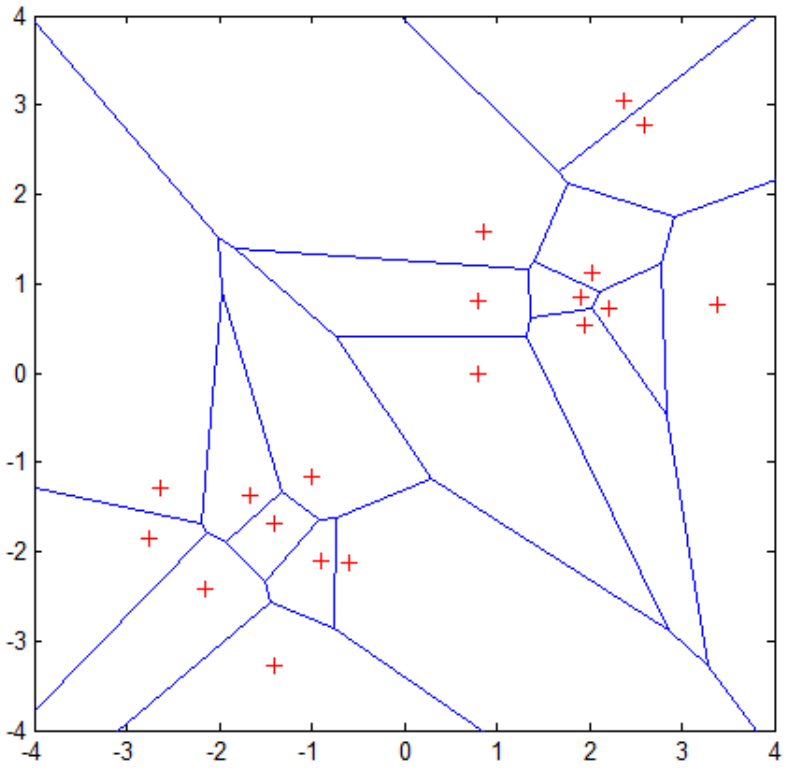
Knn-algoritmi on yksi vanhimmista ja yksinkertaisimmista metodeista objektien luokitteluun. Siitä huolimatta sillä saavutetaan kilpailukykyisiä tuloksia, ja ennakkotietoon viisaasti yhdistettynä se on edistänyt myös tutkimustyötä. (Weinberger, Blitzer ja Saul 2005) Sitä voidaan käyttää luokitteluun, arviointiin sekä ennustamiseen. Uuden objektin luokittelu tapahtuu vertaamalla sitä koulutusdatasta löytyviin samankaltaisiin objekteihin. Myös Knn-algoritmissa kahden objektin välistä samankaltaisuutta mitataan usein niiden euklidisella etäisyydellä 2.3 toisistaan, sillä se edustaa hyvin ihmisten tavallista tapaa ajatella etäisyyksiä käytännön tilanteissa. (Larose 2014) Toisaalta sen suoriutuminen on erittäin riippuvainen valitusta etäisyysmitasta (*engl. distance metric*) ja se tulisikin valita aina tilanteesta riippuen (Weinberger, Blitzer ja Saul 2005).

Knn-algoritmin esiasteena voidaan pitää artikkelia Fix ja Hodges Jr 1951, joka esittelee eiparametrisen (*engl. non-parametric*) luokittelumenetelmän. (Peterson 2009) Knn on laskennallisesti työläs algoritmi eikä saavuttanut suurta suosiota ennen 1960-lukua, jolloin tietokoneiden laskentateho kasvoi tarpeeksi suureksi (Han ja Kamber 2001). Knn-algoritmin periaatteena on, että objekti tulisi määrätä samaan klusteriin, missä sen lähin naapurikin on. Jos kaksi objektia jakavat naapuriobjektin, tulisi niitä pitää samanlaisina. (Jain ja Dubes 1988) On perusteltua ottaa huomioon luokittelemattoman objektin naapurit, sillä on järkevää olettaa, että lähellä toisiaan olevilla (jollakin sopivalla mitalla) objekteilla on samanlaisia omi-

naisuuksia (Cover ja Hart 1967; Dudani 1976).

Theodoridis ja Koutroumbas 2006 määrittelevät Knn-algoritmille seuraavat vaiheet:

- Objektille $x$ määritellään muista objekteista $k$ lähintä naapuria.
- Näistä $k$:sta naapurista määritellään klusterit, joihin ne kuuluvat.
- Määrätään objekti $x$ siihen klusteriin, jonka edustajia on eniten lähimpien naapurien joukossa $k$.



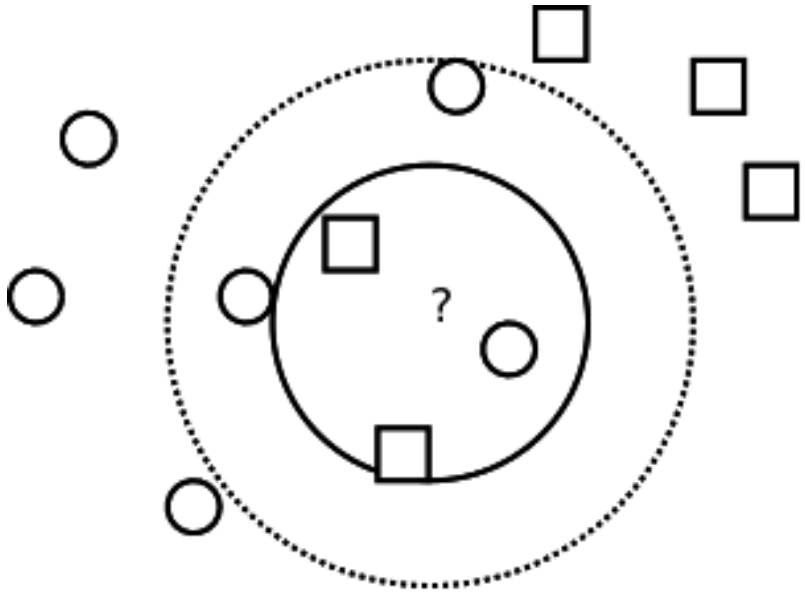

Kuvio 8. Graafinen esitys Knn-algoritmin periaatteesta. (Peterson 2009)

Kuvan 8 Voronoi-tesselaatio (Voronoï 1908) esittää graafisesti Knn-algoritmin toimintaperiaatetta. 19 objektia on merkitty plusmerkeillä ja viivoilla on esitetty Voronoin solut. Voronoin solut sisältävät kaikki pisteet, jotka ovat lähinnä yhtä tiettyä objektia. (Peterson 2009) Knn-algoritmin tapauksessa tämä tarkoittaa sitä, että jokainen saman Voronoin solun sisällä oleva objekti kuuluu samaan klusteriin.

### 4.2.1 Naapurimäärän valinta

Kun käsiteltävän tiedon määrä on suuri, on järkevää käyttää yhden lähimmän naapurin sijasta useampaa ($k$) lähintä naapuria. Lähimpien naapureiden joukko on oltava tarpeeksi suuri, jotta ei-bayesilaisen päätöksen todennäköisyys voidaan minimoida, mutta tarpeeksi pieni (käsiteltävän tiedon määrään suhteutettuna), jotta naapurijoukko olisi tarpeeksi lähellä antaakseen tarkan arvion. (Cover ja Hart 1967) Kahden luokan tilanteessa $k$:n arvoksi valitaan yleensä pariton arvo, jotta vältetään tasatilanteet naapurijoukossa (Peterson 2009). Useamman kuin kahden luokan tapauksessa pelkkä pariton $k$:n arvo ei riitä ratkaisemaan tasatilanteita, esimerkiksi, jos $k = 5$ ja lähimpien naapurien luokat ovat (2,2 ja 1) (Dougherty 2012).

Kuva 9 havainnollistaa lähimpien naapureiden lukumäärän merkitystä uuden objektin luokittelussa. Kuvassa uuden objektin luokaksi tulee joko ympyrä tai neliö. Tapauksessa $k = 1$, uuden objektin luokaksi tulee ympyrä. Jos naapureiden määrää kasvatetaan kolmeen ($k = 3$), uuden objektin luokka on neliö. Viiden naapurin tapauksessa ($k = 5$) uusi luokka on jälleen ympyrä.




Kuvio 9. Lähimpien naapureiden lukumäärä vaikuttaa uuden objektin luokitteluun

### 4.2.2 Ongelmia

Yksi Knn-algoritmin suurista ongelmista on lähimpien naapureiden etsimisen kompleksisuus. Raa'alla voimalla (*brute force* suoritettu etsintä kasvattaa operaation aikavaativuuden luokaksi $kN(O(kN))^2$. Ongelma kasvaa erittäin suureksi korkeaulottuvuuksisissa tietojoukoissa. (Theodoridis ja Koutroumbas 2006) Knn-algoritmin tuottamat arviot ovat myös alttiita taustamelulle, ja tarvittavan koulutusdatan määrä kasvaa nopeasti, kun käsiteltävien objektien sisältämät ominaisuudet (*engl. dimensionality*) kasvavat. Nyrkkisääntönä voidaan pitää 10-kertaista koulutusdatan määrää verrattuna objektien ominaisuuksiin. (Dougherty 2012)

Guo ym. 2003 avaavat Knn-algoritmin olevan K-means:n tapaan erittäin riippuvainen valittujen naapureiden määrästä ($k$). Sen valintaan on olemassa monia keinoja, joista yksinkertaisin lienee algoritmin ajaminen eri ($k$):n arvoilla valiten näistä arvoista parhaiten suoriutuva. Knn on lisäksi erittäin kallis algoritmi uuden objektin luokittelemiseen, sillä kaikki sen suorittama laskenta menee objektien luokitteluun koulutusdatan määrittelyn sijasta.

## 4.3 K-medoids

K-medoids toimii samaan tapaan kuin K-means, mutta klusterin objektien keskiarvojen sijaan K-medoids valitsee itse objektin edustamaan koko klusteria. Jokaista klusteria siis edustaa yksi edustajaobjekti, medoidi. Loput objektit määrätään klustereihin sen mukaan, mikä medoideista on kaikkein samanlaisin klusteroitavaan objektiin verrattuna. K-medoids-algoritmin periaatteena on minimoida erilaisuuksien (*engl. dissimilarity*) summa jokaisen objektin ja sen medoidin välillä. Erilaisuuden summan minimoimisessa käytetään absoluuttisen virheen kaavaa 4.2. (Han ja Kamber 2001)

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|, \tag{4.2}$$

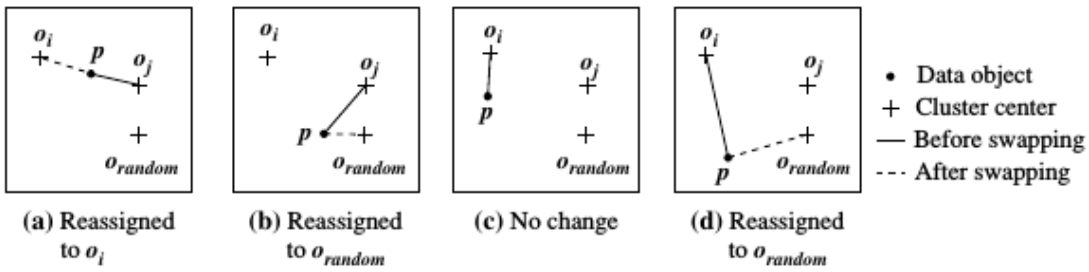
missä $E$ on kaikkien objektien absoluuttisen virheen summa, ja $p$ on objekti, joka kuuluu klusteriin $C_j$. $o_j$ on klusterin $C_j$ medoidi. Yleensä algoritmi iteroi, kunnes jokainen medoidi on oman klusterinsa keskeisin objekti. (Han ja Kamber 2001)

K-medoids-algoritmi minimoi keskimääräisen erilaisuuden klusterin objektien ja sen medoidin välillä. Se ei ole altis vieraiden havaintojen vaikutukselle, koska se perustuu keskeisimpään objektiin klusterissa eikä klusterin keskiarvokoordinaatteihin (Park ja Jun 2009). Lisäksi se mahdollistaa muodostettujen klustereiden luonnehdinnan tehokkaasti, kunhan ne eivät ole liian venyneitä. Useimmissa tapauksissa myös vieraiden havaintojen tunnistaminen on mahdollista. Monissa yksi- ja moniulotteisen tilastoinnin tapauksissa keskimääräisen erilaisuuden minimointiin perustuvat metodit ovat paljon vakaampia kuin neliöiden summaan perustuvat metodit. Jälkimmäisten laskennallinen tehokkuus ei riitä, sillä ne ovat erittäin herkkiä vieraiden havaintojen vaikutukselle. (Kaufman ja Rousseeuw 1987)

Han ja Kamber 2001 määrittelevät K-medoids-algoritmin vaiheet seuraavalla tavalla:

1. Valitaan, yleensä satunnaisesti, $k$ objektia toimimaan ensimmäisinä edustajaobjekteina eli medoideina.
2. Määrätään jokainen objekti lähimmälle medoidille.
3. Valitaan satunnaisesti objekti $o_{random}$, joka ei ole medoidi.
4. Lasketaan hinta $S$, jos medoidi $o_j$ korvataan objektilla $o_{random}$.
5. Jos $S < 0$, korvataan $o_j$ objektilla $o_{random}$, jolloin siitä tulee uusi medoidi.

Algoritmin kohtia 2-5 toistetaan, kunnes muutoksia medoideissa ei enää tapahdu. Kohdassa 4 mainitulla kokonaiskustannuksella tarkoitaan sitä, että parantuuko klusteroinnin laatu, jos uusi medoidi valitaan. Yleensä tällä laadulla tarkoitetaan hintafunktiota (*engl. cost function*), kuten keskimääräistä erilaisuutta objektin ja sen lähimmän medoidin välillä. (Han ja Kamber 2001)




Kuvio 10. Hintafunktion neljä eri tapausta. (Han ja Kamber 2001)

Medoidin $o_j$ korvaaminen toisella objektilla $o_{random}$ tapahtuu laskemalla jokaisen objektin $p$ etäisyys lähimmästä medoidista ja päivittämällä tämä etäisyys hintafunktioon. Itse medoidin korvaaminen on yksinkertaista. Oletetaan, että objekti $p$ on määrätty klusteriin, jota edustaa medoidi $o_j$ (Kuva 10 a- ja b-kohdat). Jos medoidi $o_{random}$ korvaa objektin $o_j$ medoidina, objekti $p$ määrätään siihen klusteriin, jonka medoidi on lähimpänä. Kuvan 10 a-kohdassa medoidi $o_i$ on vaihdon jälkeen lähempänä ja b-kohdassa uusi medoidi $o_{random}$ on lähempänä. C-kohdassa ei tapahdu muutosta, koska $o_i$ on edelleen lähempänä medoidin korvauksen jälkeen. D-kohdassa objekti $p$ määrätään uuteen medoidiin $o_{random}$, koska se on korvauksen jälkeen lähempänä objektia $p$ kuin $o_i$. (Han ja Kamber 2001)

Kun medoidi vaihtuu, absoluuttisen virheen $E$ erotus lisätään hintafunktioon. Näin ollen hintafunktio laskee erotuksen absoluuttisessa virheessä, jos medoidi vaihtuu. Vaihdon kokonaishinta on kaikissa objekteissa tapahtuvan muutoksen yhteishinta. Jos kokonaishinta on negatiivinen, medoidi vaihdetaan, koska absoluuttinen virhe $E$ pienenee. Vaihdosta ei tapahdu kokonaishinnan ollessa postiviinen. K-medoids-algoritmissa jokaisen iteraation aikavaativuus on luokkaa $O(k(n-k)^2)$, joten suurilla $n$:n ja $k$:n arvoilla algoritmista tulee laskennallisesti todella haastava, paljon haastavampi kuin K-means-algoritmista. (Han ja Kamber 2001)

Park ja Jun 2009 listaavat vaihtoehtoja ensimmäisten medoidien valinnalle:

- Valitaan ensimmäiset medoidit kaikkien objektien joukosta täysin satunnaisesti.
- Järjestetään objektit jonkin arvon mukaan, jaetaan järjestetty joukko $k$:hon yhtäsuureen osaan ja valitaan satunnaisesti jokaisesta osasta yksi objekti.
- Satunnaisotanta, joka on kooltaan 10% alkuperäisestä objektien määrästä. Otannalle suoritetaan alustava klusterointi, jonka tuloksena saadut medoidit toimivat ensimmäisinä medoideina varsinaiselle klusteroinnille.
- Valitaan $k$ objektia, jotka ovat kauimpana keskustasta.

Myöskään K-medoids-algoritmia käytettäessä klustereiden määrä $k$ ei ole ennalta tiedossa, vaan yleensä algoritmia joudutaan suorittamaan useilla $k$:n arvoilla ja valitsemaan tuloksien perusteella sopivin. (Kaufman ja Rousseeuw 1987)

# 5 Mittausdatan klusterointi

Tässä luvussa kerrotaan tähän tutkimukseen käytetyn klusteroitavan datan luonteesta ja tutkimuksen analysointiprosessista. Tiedonkeruupaikan ja -ajankohdan lisäksi ensimmäisessä kappaleessa selitetään, mitä attribuutteja käsiteltävä data sisältää. Siinä kerrotaan tarkasti myös laitteisto sekä sen asetukset tiedonkeruun hetkellä, käytetyt tekniikat ja taajuudet sekä kerättyjen näytteiden määrä. Tutkimuksen analysointiprosessia on avattu vaiheittain omassa kappaleessaan.

## 5.1 Klusteroitava data

Tässä työssä on tarkoituksena klusteroida empiirisesti kerättyä dataa edellisessä luvussa esitetyillä algoritmeilla. Klusterointi suoritetaan Matlab-ohjelmistolla. Mittausasetelma on sama kuin artikkeleissa (Mondal, Turkka ja Ristaniemi 2015; Turkka ym. 2015; Hiltunen ym. 2015). Klusteroitava data sisältää LTE-tukiasemien RSRP-näytteitä (*engl. Reference Signal Received Power*) ja WLAN-tukiasemien RSSI-arvoja (*Received Signal Strength Indicator*), jotka on kerätty Tampereen Kalevan alueelta kahdessa osassa: vuoden 2014 syyskuussa ja vuoden 2015 toukokuussa.

Mittaukset on suoritettu sekä 800 MHz että 1800 MHz LTE-taajuuksilla. Mittausympäristö on tyypillinen suomalainen kaupunkiympäristö koostuen erikorkuisista rakennuksista, aukioista ja puistoista. Reitti suoritettiin useampaan kertaan, jotta tarpeellinen määrä näytteitä saatiin kerättyä ja näytteisiin saatiin lisää satunnaisuutta esimerkiksi kehohävikin (*engl. body loss*) vuoksi. Näytteitä kerättiin yli 150 kilometrin matkalta kävellen, polkupyörällä ja autolla kattaen 0,33 neliökilometrin alueen. Kuva 11 havainnollistaa mittausreittiä, jonka sijainti on saatu GNSS-satelliittivastaanottimen (*engl. Global Navigation Satellite System*) avulla. Kuvassa ympyrät merkkaavat lähtö- ja loppupaikkaa sekä kuljettua matkaa kilometreissä.

Mittalaitteena toimi Samsung Galaxy S3 LTE-puhelin ohjelmistolla, joka mahdollistaa 2G-, 3G-, 4G- ja WLAN-radiotaajuusmittauksien kirjaamisen yhdessä solutunnisteiden ja sijaintitietojen kanssa. Mittausten ajan laite sijaitsi takin etutaskussa. Laitteen RRC-tila (*radio resource control*) oli jatkuvasti päällä ja lukittuna tietylle taajuuskaistalle välttääkseen solun-


Kuvio 11. Mittausreitti on merkitty karttaan sinisellä

vaihdot (*engl. handover*) taajuuksien välillä. LTE RSRP-näytteitä on yli 210 000 ja WLAN RSSI-arvoja yli 147 000, mikä vastaa noin 4500 LTE- ja 2600 WLAN-näytettä per hehtaari 1800 MHz taajuudella sekä 2200 LTE- ja 1900 WLAN-näytettä per hehtaari 800 MHz taajuudella. 800 MHz taajuudella mitattuja näytteitä on vähemmän, koska 1800 MHz taajuudella näytteet koostuivat sekä taajuuksien sisäisistä että taajuuksien välisistä mittauksista, missä 800 MHz näytteet koostuivat ainoastaan taajuuksien sisäisistä näytteistä. 1800 MHz taajuudella yksi LTE-näyte sisältää keskimäärin 4,8 havaittua solua ja 800 MHz taajuudella vain 3,2 havaittua solua. LTE-näytteet sisältävät 800 MHz taajuudella 502 yksilöllistä PCI-arvoa (*Physical Cell Identifications*) ja 1800 MHz taajuudella vastaava luku on 399. Havaittuja tukiasemia on yhteensä 3161.

## 5.2 Analysointiprosessi

1. Klusteroitavan tiedon kerääminen
2. Tiedon esikäsittely
3. Esikäsitellyn tiedon klusterointi
4. Saatujen tuloksien koonti
5. Tuloksien analysointi
6. Yhteenveto

Analysointiprosessin ensimmäinen vaihe on käsiteltävän tiedon kerääminen, ja se on kuvattu edellisessä kappaleessa. Tiedon esikäsittelyllä tarkoitetaan kerätyn tiedon muokkaamista helpommin käsiteltävään muotoon, mikä tarkoittaa tässä tapauksessa helpommin klusteroitavaa muotoa. Esikäsittelyssä kenttämittauksilla kerätyt sormenjäljet ryhmitellään ja niistä poistetaan tyhjät arvot. Tässä työssä tiedon esikäsittelyyn ja klusterointiin käytetään Matlab-ohjelmistoa, josta löytyy valmiit funktiot K-means-, K-medoids- ja Knn-algoritmeille. K-means- ja K-medoids- algoritmit suoritetaan erikseen 10-kertaista toistoa käyttäen. Kuten kappaleessa 4.1 (s. 22-23) mainittiin, 10-kertaisella toistolla tarkoitetaan algoritmin 1. vaiheen toistamista klusterointiparametrit säilyttäen. Toistokerroista valitaan sopivin alkuasetelma klusteroinnin seuraavia vaiheita varten.

Kerätty sormenjälkitieto on lisäksi jaoteltu kymmeneen yhtä suureen osaan, joista klusterointivaiheessa yksi osa toimii vuorollaan testausdatana ja loput yhdeksän osaa koulutusdatana. Jokainen klusterointialgoritmi suoritetaan siis kymmenen kertaa samansuuruisella, mutta eri sormenjälkiä sisältävällä datalla. Tämän lisäksi 10-kertaisessa toistossa klusteroinnin 1. vaihe on suoritettu 10 kertaa.

Jokaisen eri suorituskerran tulokset tallennetaan automaattisesti erilliseen tiedostoon. Nämä tiedostot sisältävät jokaisella suorituskerralla käytetyt muuttujat ja niiden lopulliset arvot. Lisäksi tulostiedostoista löytyy sekä 68%:n että 95%:n klusterointitulos ja klusterointiprosentti. Ohjelmistolla lasketaan myös jokaisen algoritmin suoritusaika. Yhden algoritmin muodostamista tulostiedostoista muodostetaan taulukko, joka ilmoittaa klusterointitulokset (68% ja 95%), klusterointituloksien keskiarvot ja algoritmin suoritusajan. Kootut tulokset analysoidaan sen perusteella, mitä tuloksilta odotettiin ja miksi mitäkin tuloksia saatiin. Lopuksi esi-

tetään vielä tutkielman yhteenveto, jossa todetaan lyhyesti, mitä tutkielmassa esitetyn nojalla voidaan sanoa johdannon väitteen totuudesta tai tutkimuskysymyksen vastauksesta.

# 6 Klusterointitulokset

Tässä luvussa käsitellään klusteroinnin varsinaisia tuloksia eli sitä, kuinka tarkasti mobiililaite voidaan paikantaa milläkin algoritmilla. Luku on jaettu kahteen kappaleeseen, joista ensimmäisessä tulokset esitetään ja selitetään auki. Toinen kappale sisältää varsinaisen tulosanalyysin, jossa pohditaan, mitä tuloksilta odotettiin ja miksi mitäkin tuloksia saatiin.

## 6.1 Tulosten esittely

Taulukko 1 näyttää metreissä saadut klusterointitulokset K-means-algoritmilla ja taulukossa 2 on tulokset, jotka on saatu K-means-algoritmilla ja 10-kertaisella toistolla. Kummassakin algoritmissa klustereiden määrä on kolme. Molemmista tuloksista on esitetty sekä 68%:t että 95%:t tulokset, tuloksien keskiarvot sekä suoritusajat. Lisäksi taulukoihin on koostettu jokaisen suorituskerran klusterointiprosentti, joka kertoo kuinka suuri osa objekteista on klusteroitu. Taulukkoon 2 on merkitty punaisella tulokset, jotka ovat huonompia kuin vastaavat tulokset taulukossa 1.

Taulukko 1. K-means-algoritmi, kolme klusteria

| | 68% (m) | 95% (m) | Klusterointi (%) |
|---|---|---|---|
| 1 | 21,848 | 65,364 | 99,922 |
| 2 | 24,504 | 68,540 | 99,727 |
| 3 | 23,269 | 60,262 | 99,064 |
| 4 | 21,343 | 65,904 | 99,571 |
| 5 | 21,273 | 63,305 | 99,337 |
| 6 | 22,954 | 63,567 | 99,922 |
| 7 | 22,157 | 59,852 | 99,922 |
| 8 | 22,408 | 55,538 | 99,610 |
| 9 | 23,326 | 62,268 | 99,961 |
| 10 | 22,430 | 59,492 | 99,961 |
| Keskiarvo | 22,551 | 62,409 | 99,700 |
| Kesto | 69,386 s | 19,274 h | |

Taulukko 2. K-means-algoritmi 10-kertaisella toistolla, kolme klusteria

| | 68% (m) | 95% (m) | Klusterointi (%) |
|---|---|---|---|
| 1 | 19,156 | 55,617 | 99,961 |
| 2 | 22,927 | 60,846 | 99,727 |
| 3 | 22,681 | 63,490 | 99,376 |
| 4 | 19,407 | 60,718 | 99,610 |
| 5 | 20,150 | 60,822 | 99,454 |
| 6 | 21,690 | 63,437 | 100,000 |
| 7 | 20,795 | 53,533 | 99,961 |
| 8 | 22,227 | 50,333 | 99,805 |
| 9 | 22,411 | 62,784 | 99,961 |
| 10 | 21,478 | 53,033 | 100,000 |
| Keskiarvo | 21,292 | 58,461 | 99,785 |
| Kesto | 70 403,302 s | 19,556 h | |

Taulukoissa 3 ja 4 on tulokset, jotka on saatu klusteroimalla paikannusdata K-medoids-algoritmilla ilman 10-kertaista toistoa sekä 10-kertaisen toiston kanssa. Molemmissa klusteroinneissa on jälleen kolme klusteria, jotta eri algoritmien välisiä tuloksia voidaan vertailla paremmin. Molemmista on esitetty sekä 68%:t että 95%:t tulokset. Taulukkoon 4 on lisäksi merkitty punaisella tulokset, jotka ovat huonompia kuin vastaavat tulokset taulukossa 3. Myös molemmista K-medoids-ajoista on kerätty niiden klusterointiprosentti sekä suoritusaika.

Kuvat 12 ja 13 havainnollistavat Knn-algoritmin paikannustarkkuuden keskiarvon kehitystä, kun lähimpien naapureiden määrää kasvatetaan.

## 6.2 Tulosanalyysi

Taulukoista 1 ja 2 nähdään, että verrattuna pelkkään K-means-algoritmiin, 10-kertaisella toistolla suoritettu K-means antaa parempia tuloksia kaikissa kohdissa, paitsi 95%:n kohdissa 3 ja 9 (merkitty punaisella taulukkoon 2). Keskiarvollisesti 10-kertainen toisto antaa

Taulukko 3. K-medoids-algoritmi, kolme klusteria

| | 68% (m) | 95% (m) | Klusterointi (%) |
|---|---|---|---|
| 1 | 19,963 | 55,588 | 99,961 |
| 2 | 22,312 | 62,129 | 99,727 |
| 3 | 22,537 | 62,129 | 99,376 |
| 4 | 19,100 | 60,254 | 99,610 |
| 5 | 20,064 | 61,705 | 99,454 |
| 6 | 22,663 | 65,520 | 100,000 |
| 7 | 21,013 | 57,175 | 99,961 |
| 8 | 21,709 | 52,255 | 99,805 |
| 9 | 22,043 | 59,905 | 99,961 |
| 10 | 22,038 | 56,902 | 100,000 |
| Keskiarvo | 21,344 | 59,356 | 99,785 |
| Kesto | 67 677,793 s | 18,799 h | |

parempia tuloksia sekä 68%:ssa että 95%:ssa tuloksista. Taulukoista 3 ja 4 voidaan päätellä, että keskinäinen ero K-medoids -suorituksien välillä on pienempi kuin keskinäisten K-means-suorituksien välillä. Myöskään keskiarvollisesti 10-kertainen toisto ei tarjoa suurta parannusta K-medoids-algoritmin paikannustarkkuuteen, joten sen käyttäminen ei välttämättä ole mielekästä sen kuitenkin pidentäessä algoritmin suoritusaikaa. Kuvan 12 mukaan paras paikannustulos Knn-algoritmilla on 16,9638 metriä, joka saadaan, kun lähimpien naapureiden määrä on 9. Vastaava tulos kuvassa 13 on 50,2590 metriä. 95 %:ssa tuloksista paras paikannustarkkuus, 49,6171 metriä, saadaan lähimpien naapureiden lukumäärän ollessa 13.

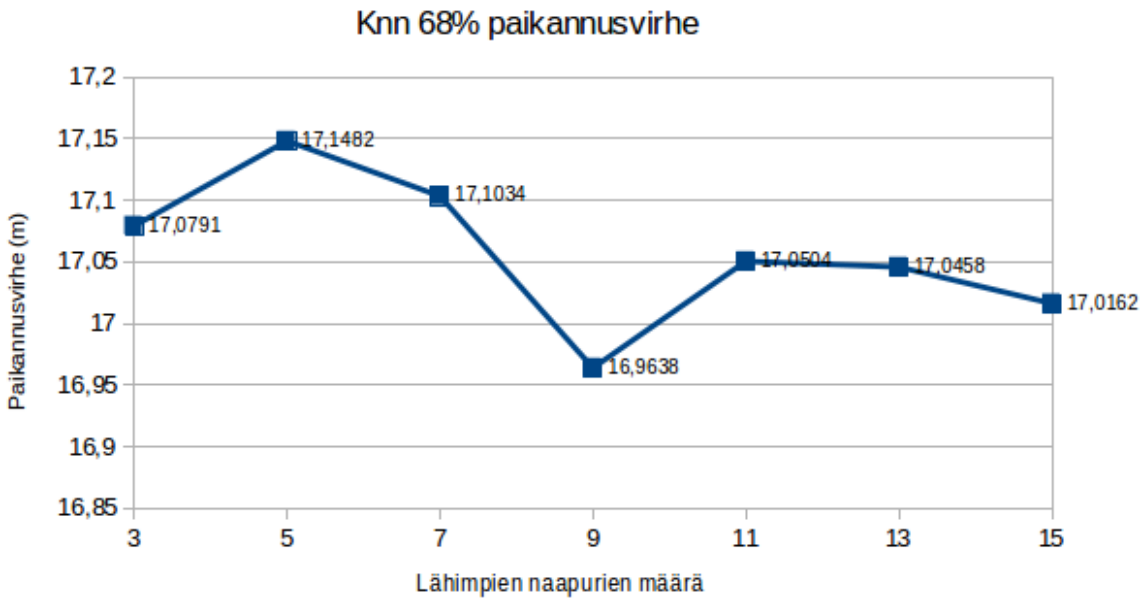
Taulukosta 14 nähdään, että Knn-algoritmi saavuttaa tarkimman paikannustuloksen molemmissa tuloskategorioissa. Ero K-means- ja K-medoids-algoritmeihin on selkeä, kun taas niiden keskinäinen ero on selvästi vähäisempi.

Tuloserot K-means:n ja K-medoids:n välillä ovat odotetusti pieniä, sillä molempien toimintaperiaate on jokseenkin samanlainen; ne pyrkivät jakamaan annetun datan *k*:n klusteriin siten, että etäisyyksien summa sentroidin/medoidin ja muiden objektien välillä on mahdollisimman pieni. Lisäksi molemmat algoritmit käyttävät vakiona ensimmäisten klusterikeskuk-

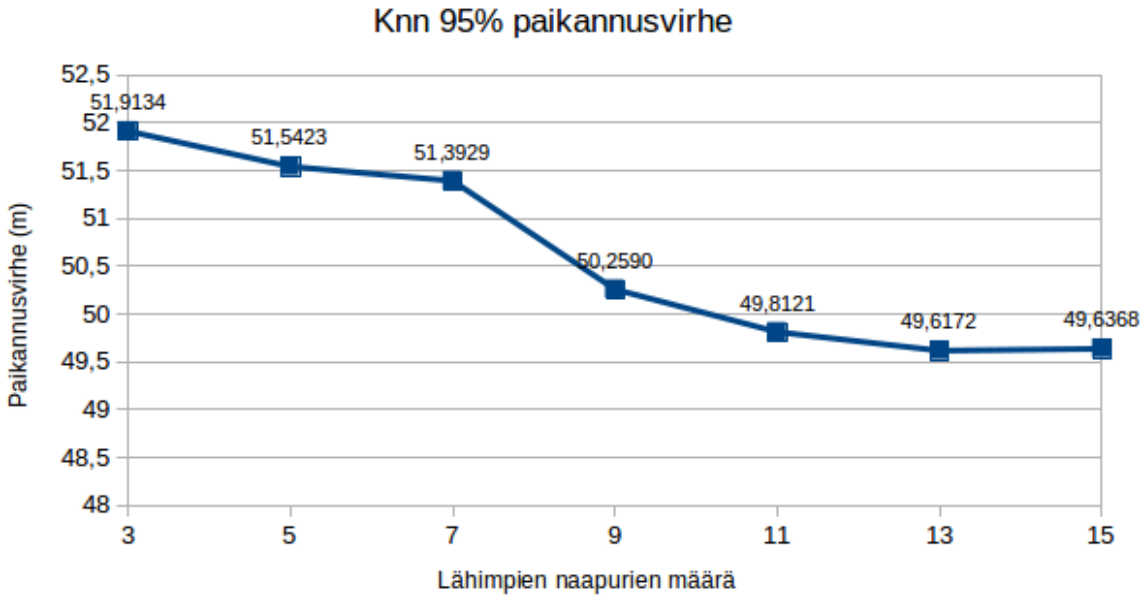
Taulukko 4. K-medoids-algoritmi, 10-kertaisella toistolla, kolme klusteria

| | 68% (m) | 95% (m) | Klusterointi (%) |
|---|---|---|---|
| 1 | 19,952 | 55,460 | 99,961 |
| 2 | 22,501 | 62,129 | 99,727 |
| 3 | 22,393 | 61,553 | 99,376 |
| 4 | 19,270 | 60,529 | 99,610 |
| 5 | 19,948 | 61,024 | 99,454 |
| 6 | 22,052 | 65,520 | 100,000 |
| 7 | 20,886 | 57,417 | 99,961 |
| 8 | 21,803 | 52,615 | 99,805 |
| 9 | 22,043 | 59,905 | 99,961 |
| 10 | 22,305 | 56,898 | 100,000 |
| Keskiarvo | 21,315 | 59,305 | 99,785 |
| Kesto | 70 787,586 s | 19,663 h | |

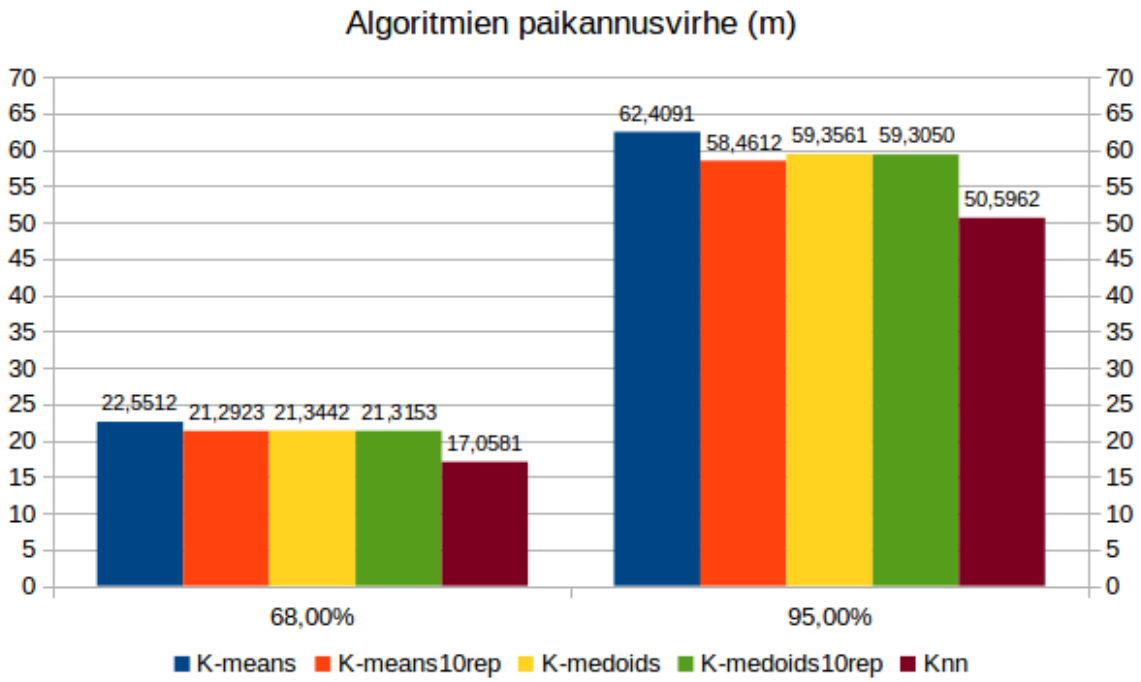
sien valintaan K-means++ -algoritmia. K-means++ valitsee ensimmäiset klusterikeskukset sen perusteella, mikä edistää parhaimman klusterointituloksen saavuttamista eniten (Arthur ja Vassilvitskii 2007). Knn-algoritmilla ei ole samaa klustereiden alustamiseen liittyvää ongelmaa kuin K-means- ja K-medoids-algoritmeilla, koska sen klusterointi perustuu naapuriobjektien tiedossa oleviin ominaisuuksiin. Tämä saattaa olla yksi selittävä tekijä saaduille paikannustuloksille.




Kuvio 12. Knn-algoritmin paikannusvirhe 68%:ssa tuloksista.




Kuvio 13. Knn-algoritmin paikannusvirhe 95%:ssa tuloksista.




Kuvio 14. Algoritmien paikannusvirheet.

# 7 Yhteenveto

Tämän tutkimuksen keskeisenä tavoitteena oli tutkia mobiililaitteiden paikannusmenetelmää, jossa hyödynnetään WLAN- ja mobiiliverkon tukiasemien signaaleja mobiililaitteen paikantamiseen. Klusteroinnin ja radiotaajuuspaikannuksen esittelyn lisäksi tarkoituksena oli määrittää mobiililaitteiden paikannustarkkuus klusteroimalla empiirisesti kerättyä radiotaajuusdataa. Klusterointiin käytettiin K-means-, K-medoids- ja Knn-algoritmeja ja se toteutettiin Matlab-ohjelmistolla. Lisäksi K-means- ja K-medoids-algoritmeja suoritettiin 10-kertaisella toistolla, jossa klusterialustus suoritettiin satunnaisilla arvoilla kymmenen kertaa. Satunnaistoistoista paras alustus valittiin varsinaiseen klusterointiin.

Edellisessä luvussa esitettyjen tuloksien perusteella voidaan todeta, että kolmesta edellä mainitusta algoritmista Knn saavutti parhaimman paikannustarkkuuden. Tarkin tulos, 16,96 metriä, saatiin lähimpien naapureiden lukumäärän ollessa 9. Knn-algoritmin saavuttamien tuloksien keskiarvo oli 17,05 metriä. Vastaavasti K-means:n saavuttama keskiarvotarkkuus oli 10-kertaista toistoa käyttäen 21,29 metriä (22,55 metriä ilman toistoa) ja K-medoids-algoritmin tulos oli 21,32 metriä (21,34 metriä ilman toistoa). Radiotaajuuspaikannuksen etuna on sen toiminta ulko- ja sisätiloissa. Se toimii myös silloin, kun GPS-järjestelmän paikannussatelliitti ei ole saatavilla tai se ei ruuhkan vuoksi pysty käsittelemään kaikkia paikannuspyyntöjä tehokkaasti. Toisaalta radiotaajuuspaikannus toimii tehokkaasti vain, jos saatavilla on riittävästi WLAN- tai mobiiliverkon tukiasemia. Tämä aiheuttaa ongelmia harvaan asutuilla alueilla, mutta tiheästi rakennetuissa kaupunkikeskustoissa tukiasemien signaaleja on yleensä riittävästi.

Tutkimuksessa ei ollut tarkoitus selvittää tarkasti parasta klusterialustusta käytetyille algoritmeille. Erityisesti K-means- ja K-medoids-algoritmien suorituksiin vaikuttaa käsiteltävän datan luonteen lisäksi se, kuinka monta klusteria halutaan. Parhaimman paikannustarkkuuden saavuttama klusterimäärä olisikin hyvä aihe jatkotutkimukselle. Ensimmäisten sentroidien ja medoidien valinta on myös hyvin tärkeä osa edellä mainittujen algoritmien suoritusta. Niiden valintaan on olemassa lukuisia tekniikoita, joiden vaikutus algoritmien paikannustarkkuuteen voisi myös olla yksi tämän tutkimuksen seuraavista vaiheista.

# Lähteet


Ahonen, S., ja H. Laitinen. 2003. “Database correlation method for UMTS location”. Teoksessa *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual,* nide 4, 2696–2700 vol.4. Huhtikuu. doi:10.1109/VETECS.2003.1208882.

Arthur, David, ja Sergei Vassilvitskii. 2007. “k-means++: The advantages of careful seeding”. Teoksessa *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms,* 1027–1035. Society for Industrial ja Applied Mathematics.

Bahl, Paramvir, ja Venkata N Padmanabhan. 2000. “RADAR: An in-building RF-based user location and tracking system”. Teoksessa *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE,* 2:775–784. Ieee.

Bahl, Paramvir, Venkata N Padmanabhan ja Anand Balachandran. 2000. *Enhancements to the RADAR user location and tracking system.* Tekninen raportti. technical report, Microsoft Research.

Berry, Michael JA, ja Gordon S Linoff. 2004. *Data mining techniques: for marketing, sales, and customer relationship management.* John Wiley & Sons.

Campos, Ricardo Silva, ja Lisandro Lovisolo. 2008. “Location methods for legacy GSM handsets using coverage prediction”. Teoksessa *Signal Processing Advances in Wireless Communications, 2008. SPAWC 2008. IEEE 9th Workshop on,* 21–25. IEEE.

———. 2009. “A Fast Database Correlation Algorithm for Localization of Wireless Network Mobile Nodes using Coverage Prediction and Round Trip Delay”. Teoksessa *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th,* 1–5. Huhtikuu. doi:10.1109/VETECS.2009.5073292.

Chen, Mike Y, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith ja Alex Varshavsky. 2006. “Practical metropolitan-scale positioning for gsm phones”. Teoksessa *UbiComp 2006: Ubiquitous Computing,* 225–242. Springer.

Cover, Thomas M, ja Peter E Hart. 1967. “Nearest neighbor pattern classification”. *Information Theory, IEEE Transactions on* 13 (1): 21–27.

Davidson, Ian, ja SS Ravi. 2005. “Clustering with Constraints: Feasibility Issues and the k-Means Algorithm.” Teoksessa *SDM,* 5:201–211. SIAM.

Dougherty, Geoff. 2012. *Pattern recognition and classification: an introduction.* Springer Science & Business Media.

Duda, Richard O., Peter E. Hart ja David G. Stork. 2000. *Pattern Classification (2Nd Edition).* Wiley-Interscience. ISBN: 0471056693.

Dudani, Sahibsingh A. 1976. “The distance-weighted k-nearest-neighbor rule”. *Systems, Man and Cybernetics, IEEE Transactions on,* numero 4:325–327.

ETSI. 2004. *Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Functional description; Stage 2 (3GPP TS 03.71 version 8.9.0 Release 1999).* TS 101 724. European Telecommunications Standard Institute.

Everitt, Brian S., Sabine Landau ja Morven Leese. 2009. *Cluster Analysis.* 4th. Wiley Publishing. ISBN: 0340761199, 9780340761199.

Fix, Evelyn, ja Joseph L Hodges Jr. 1951. *Discriminatory analysis-nonparametric discrimination: consistency properties.* Tekninen raportti. DTIC Document.

Forgy, Edward W. 1965. “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”. *Biometrics* 21:768–769.

Gezici, Sinan. 2008. “A survey on wireless position estimation”. *Wireless personal communications* 44 (3): 263–282.

Gezici, Sinan, Zhi Tian, Georgios B Giannakis, Hisashi Kobayashi, Andreas F Molisch, H Vincent Poor ja Zafer Sahinoglu. 2005. “Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks”. *Signal Processing Magazine, IEEE* 22 (4): 70–84.

Guha, Sudipto, Rajeev Rastogi ja Kyuseok Shim. 2001. “Cure: an efficient clustering algorithm for large databases”. *Information Systems* 26 (1): 35–58.

Guo, Gongde, Hui Wang, David Bell, Yaxin Bi ja Kieran Greer. 2003. “KNN model-based approach in classification”. Teoksessa *OTM Confederated International Conferences"On the Move to Meaningful Internet Systems",* 986–996. Springer.

Gwon, Youngjune, ja Ravi Jain. 2004. “Error characteristics and calibration-free techniques for wireless LAN-based location estimation”. Teoksessa *Proceedings of the second international workshop on Mobility management & wireless access protocols,* 2–9. ACM.

Halkidi, Maria, Yannis Batistakis ja Michalis Vazirgiannis. 2001. “Clustering algorithms and validity measures”. Teoksessa *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on,* 3–22. IEEE.

Han, Jiawei, ja Micheline Kamber. 2001. *Data mining: concepts and techniques.*

Hansen, L.K., ja J. Larsen. 1996. “Unsupervised learning and generalization”. Teoksessa *Neural Networks, 1996., IEEE International Conference on,* nide 1, 25–30 vol.1. Kesäkuu. doi:10.1109/ICNN.1996.548861.

Hardy, André. 1996. “On the number of clusters”. *Computational Statistics & Data Analysis* 23 (1): 83–96.

Hiltunen, Tuomas, Jussi Turkka, Riaz Mondal ja Tapani Ristaniemi. 2015. “Performance evaluation of LTE radio fingerprint positioning with timing advancing”. Teoksessa *Information, Communications and Signal Processing (ICICS), 2015 10th International Conference on,* 1–5. IEEE.

Ishioka, Tsunenori. 2000. “Extended K-means with an efficient estimation of the number of clusters”. Teoksessa *Intelligent Data Engineering and Automated Learning—IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents,* 17–22. Springer.

Jain, A. K., M. N. Murty ja P. J. Flynn. 1999. “Data Clustering: A Review”. *ACM Comput. Surv.* (New York, NY, USA) 31, numero 3 (syyskuu): 264–323. ISSN: 0360-0300. doi:10.1145/331499.331504. http://doi.acm.org/10.1145/331499.331504.

Jain, Anil K. 2010. “Data clustering: 50 years beyond K-means”. *Pattern recognition letters* 31 (8): 651–666.

Jain, Anil K, ja Richard C Dubes. 1988. *Algorithms for clustering data.* Prentice-Hall, Inc.

Johansson, Johan, Wuri A Hapsari, Sean Kelley ja Gyula Bodog. 2012. “Minimization of drive tests in 3GPP release 11”. *Communications Magazine, IEEE* 50 (11): 36–43.

Kaufman, Leonard, ja Peter Rousseeuw. 1987. *Clustering by means of medoids.* North-Holland.

Kothari, Ravi, ja Dax Pitts. 1999. “On finding the number of clusters”. *Pattern Recognition Letters* 20 (4): 405–416.

Laitinen, Heikki, Jaakko Lähteenmäki ja Tero Nordström. 2001. “Database correlation method for GSM location”. Teoksessa *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd,* 4:2504–2508. IEEE.

Lamrous, Sid, ja Mounira TaÏleb. 2006. “Divisive hierarchical k-means”. Teoksessa *Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on,* 18–18. IEEE.

Larose, Daniel T. 2014. *Discovering knowledge in data: an introduction to data mining.* John Wiley & Sons.

Lim, Hyuk, Lu-Chuan Kung, Jennifer C Hou ja Haiyun Luo. 2005. “Zero-configuration, robust indoor localization: Theory and experimentation”.

Liu, H., H. Darabi, P. Banerjee ja J. Liu. 2007. “Survey of Wireless Indoor Positioning Techniques and Systems”. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37, numero 6 (marraskuu): 1067–1080. ISSN: 1094-6977. doi:10.1109/TSMCC.2007.905750.

MacQueen, James, ym. 1967. “Some methods for classification and analysis of multivariate observations”. Teoksessa *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability,* 1:281–297. 14. Oakland, CA, USA.

Mondal, Riaz Uddin, Jussi Turkka ja Tapani Ristaniemi. 2015. “An efficient cluster-based outdoor user positioning using LTE and WLAN signal strengths”. Teoksessa *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on,* 2182–2186. IEEE.

Mondal, Riaz, Jussi Turkka, Tapani Ristaniemi ja Tero Henttonen. 2013. “Positioning in heterogeneous small cell networks using MDT RF fingerprints”. Teoksessa *Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on,* 127–131. IEEE.

———. 2014. “Performance evaluation of MDT assisted LTE RF fingerprint framework”. Teoksessa *Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference on,* 33–37. IEEE.

Oughdi, M., S. Lamrous, A. Caminada ja B. Morin. 2006. “Time-Clustering of Load in Mobile Networks”. Teoksessa *Service Systems and Service Management, 2006 International Conference on,* 2:1483–1488. Lokakuu. doi:10.1109/ICSSSM.2006.320743.

Pang-Ning, Tan, Michael Steinbach, Vipin Kumar ym. 2006. “Introduction to data mining”. Teoksessa *Library of Congress,* 74.

Park, Hae-Sang, ja Chi-Hyuck Jun. 2009. “A simple and fast algorithm for K-medoids clustering”. *Expert Systems with Applications* 36 (2): 3336–3341.

Pelleg, Dan, Andrew W Moore ym. 2000. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters.” Teoksessa *ICML,* 727–734.

Pena, José Manuel, Jose Antonio Lozano ja Pedro Larranaga. 1999. “An empirical comparison of four initialization methods for the k-means algorithm”. *Pattern recognition letters* 20 (10): 1027–1040.

Peterson, Leif E. 2009. “K-nearest neighbor”. *Scholarpedia* 4 (2): 1883.

Pham, Duc Truong, Stefan S Dimov ja CD Nguyen. 2005. “Selection of K in K-means clustering”. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 219 (1): 103–119.

Porretta, M., P. Nepa, G. Manara ja F. Giannetti. 2008. “Location, Location, Location”. *Vehicular Technology Magazine, IEEE* 3, numero 2 (kesäkuu): 20–29. ISSN: 1556-6072. doi:10.1109/MVT.2008.923969.

Romesburg, Charles. 2004. *Cluster analysis for researchers.* Lulu. com.

Spirito, M. 2001. “Accuracy of hyperbolic mobile station location in cellular networks”. *Electron. Lett* 37 (11): 708–710.

The Mathworks, Inc. 2015. *MATLAB version 8.5.0.197613 (R2015a).* Natick, Massachusetts: The Mathworks, Inc.

Theodoridis, Sergios, ja Konstantinos Koutroumbas. 2006. *Pattern Recognition.* Academic Press. ISBN: 9780123695314. `http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=230861&site=ehost-live`.

Trevisani, Emiliano, ja Andrea Vitaletti. 2004. “Cell-ID location technique, limits and benefits: an experimental study”. Teoksessa *Mobile computing systems and applications, 2004. WMCSA 2004. Sixth IEEE workshop on,* 51–60. IEEE.

Turkka, Jussi, Tuomas Hiltunen, Riaz Uddin Mondal ja Tapani Ristaniemi. 2015. “Performance evaluation of LTE radio fingerprinting using field measurements”. Teoksessa *Wireless Communication Systems (ISWCS), 2015 International Symposium on,* 466–470. IEEE.

Wallace, Christopher S, ja David M Boulton. 1968. “An information measure for classification”. *The Computer Journal* 11 (2): 185–194.

Varshavsky, Alex, Eyal De Lara, Jeffrey Hightower, Anthony LaMarca ja Veljo Otsason. 2007. “GSM indoor localization”. *Pervasive and Mobile Computing* 3 (6): 698–720.

Weinberger, Kilian Q, John Blitzer ja Lawrence K Saul. 2005. “Distance metric learning for large margin nearest neighbor classification”. Teoksessa *Advances in neural information processing systems,* 1473–1480.

Vidal, Josep, Dana H Brooks ym. 2002. “Closed-form solution for positioning based on angle of arrival measurements”. Teoksessa *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on,* 4:1522–1526. IEEE.

Voronoï, Georges. 1908. “Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs”. *Journal für die reine und angewandte Mathematik* 134:198–287.

Wu, Junjie. 2012. *Advances in K-means Clustering.* Springer Berling Heidelberg.

Xu, Rui, ja Donald C. Wunsch. 2009. *Clustering.* IEEE Series on Computational Intelligence. Wiley-IEEE Press. ISBN: 9780470276808. http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=254099&site=ehost-live.

Youssef, Moustafa, ja Ashok Agrawala. 2005. "The Horus WLAN location determination system". Teoksessa *Proceedings of the 3rd international conference on Mobile systems, applications, and services,* 205–218. ACM.

Zeimpekis, Vasileios, George M Giaglis ja George Lekakos. 2002. "A taxonomy of indoor and outdoor positioning techniques for mobile location services". *ACM SIGecom Exchanges* 3 (4): 19–27.

Zekavat, Reza, ja R Michael Buehrer. 2011. *Handbook of position location: Theory, practice and advances.* Nide 27. John Wiley & Sons.