Tampereen ammattikorkeakoulu

Datan käsittelymenetelmien hyödyntäminen sähkönkulutuksen seurannassa

Paavo Yrtti

OPINNÄYTETYÖ
Huhtikuu 2025

Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Automaatiotekniikka

YRTTI, PAAVO:
Datan käsittelymenetelmien hyödyntäminen sähkönkulutuksen seurannassa

Opinnäytetyö 66 sivua, joista liitteitä 9 sivua
Huhtikuu 2025

Sähkönkulutuksen seuranta on tärkeää, koska se auttaa muun muassa sähköyhtiöitä optimoimaan järjestelmien tuottavuutta sekä ennakoimaan sähkönkulutusta. Lisäksi sähköyhtiöiden on tärkeää tietää sähköntuotannon ja kulutuksen suhde kysynnän ja tarjonnan tasapainottamiseksi.

Tässä opinnäytetyössä tutkittiin, miten dataa käsitellään ja analysoidaan sekä miten eri datan käsittelymenetelmiä hyödynnetään sähkönkulutuksen seurannassa. Lisäksi opinnäytetyössä tarkasteltiin tekoälyä ja koneoppimista osana nykypäivän datan käsittelyä sekä muun muassa kartoitettiin datan hyödyntämisen haasteita teollisuudessa.

Työn tuloksena luotiin aikasarjadatan käsittelyyn ja sähkönkulutuksen ennakointiin ohjelmakoodi, jota TAMK pystyy hyödyntämään osana AIKO-tekoälyhanketta. Hankkeessa sovelletaan tekoälyn hyödyntämistä teollisuusyritysten erilaisissa automaatioratkaisuissa. Työn ohjelmakoodissa käsiteltiin ja analysoitiin olemassa olevaa sähkönkulutukseen liittyvää aikasarjadataa sekä kuvattiin, miten koneoppimista voidaan hyödyntää sähkönkulutuksen ennakoinnissa.

Opinnäytetyön aihealueet ja toiminnallinen osuus toimivat johdantona koneoppimisen hyödyntämiselle teollisuudessa sekä aikasarjadatan käsittelylle. Työn teoriaosuudessa määritetään datan käsittelyn eri menetelmät ja tekoälyn rooli nykypäivän datan käsittelyssä. Ohjelmakoodissa määritettiin sähkönkulutuksen kanssa korreloivat piirteet data-aineistosta ja ennakoitiin niiden avulla onnistuneesti sähkönkulutusta käyttämällä koneoppimisen muotona ohjattua oppimista.

Ohjelmakoodin pohjalta sähkönkulutuksen ennakointia voidaan jatkokehittää tarkempien koneoppimismallien luomiseksi ja sitä kautta parempien ennusteiden tuottamiseksi. Lisäksi ohjelmakoodia ja siinä käytettyjä menetelmiä voidaan myös soveltaa laajemmin muihin teollisuuden aikasarjapohjaisiin data-aineistoihin.

Asiasanat: datan käsittely, sähkönkulutus, tekoäly, koneoppiminen

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical and Automation Engineering
Automation Engineering

YRTTI, PAAVO:
Utilising Data Processing Methods in Power Consumption Monitoring

Bachelor's thesis 66 pages, appendices 9 pages
April 2025

This thesis studied how data is processed and analysed as well as how various data processing methods are utilised in power consumption monitoring. The thesis also explored artificial intelligence and machine learning as part of modern data utilisation.

As a result of this work, a script was developed for processing time-series data and predicting power consumption. The script can be used by TAMK as part of AIKO artificial intelligence project where artificial intelligence is applied in different automation solutions for businesses. The script involved processing and analysing a dataset related to power consumption and demonstrated how supervised learning can be used to predict energy usage.

This thesis serves as an introduction to the use of machine learning in different industrial applications and the processing of time-series data. Building on the script, power consumption prediction can be further developed by creating more accurate machine learning models, leading to more reliable forecasts. Additionally, the script and its methods can also be applied to the processing and analysis of other time-series data sets within the industrial sector.

Key words: data processing, power consumption, artificial intelligence, machine learning

SISÄLLYS

LYHENTEET JA TERMIT

csv	Comma-Separated Values, tiedostopääte, jossa datan arvot erotetaan toisistaan pilkulla
cv	Cross-Validation, datan jakaminen osiin koneoppimismallin tarkkuuden arvioimiseksi
df	Data Frame, ohjelmoinnissa yleinen tapa kuvata taulukkomuotoista dataa
ensemble	termi koneoppimismenetelmille, joissa käytetään useita oppimisalgoritmeja tarkemman mallin luomiseksi
GPU	Graphics Processing Unit, grafiikkasuoritin
IoT	Internet of Things, esineiden internet
IF-THEN	JOS-NIIN, looginen implikaatio matemaattisessa logiikassa
MATLAB	ohjelmointikieli sekä ohjelmisto numeeriseen laskentaan
mdl	Model, ohjelmoinnissa yleinen nimeämiskäytäntö koneoppimismalleille
R^2	kerroin, joka kuvaa koneoppimismallin hyvyyttä riippuvan muuttujan selityksessä
regressioanalyysi	tilastollinen menetelmä, jossa määritetään selitettävän muuttujan riippuvuus selittävään muuttujaan
RMSE	Root Mean Squared Error, koneoppimismallin ennusteiden keskimääräisten virheiden neliöt verrattuna alkuperäisiin arvoihin
SCADA	Supervisory Control and Data Acquisition, teollisuuden valvonta- ja tiedonkeruujärjestelmä
SSE	Sum of Squared Error, kokonaisvirheiden neliösumma
std	Standard Deviation, keskihajonta
sähkönjakeluasema	sähkönjakeluverkon liitoskohta jännitteen muuntamiseksi matalamman jännitteen jakeluverkoksi
sähkönjakeluverkko	infrastruktuuri sähköenergian jakamiselle sen kuluttajille

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on selvittää, mitä eri vaiheita datan käsittelyyn sisältyy ja miten dataa pystytään hyödyntämään sähkönkulutuksen seurannassa. Tavoitteena on muodostaa kattava yleiskuva siitä, miten datasta saadaan hyödyllistä tietoa ja miksi data-analytiikka on tärkeä osa päätöksentekoa yrityksissä. Opinnäytetyön tuloksena luodaan ohjelmakoodi aikasarjadatan käsittelyyn ja analysointiin sekä koneoppimisen hyödyntämiseen sähkönkulutuksen ennakoinnissa.

Opinnäytetyön aluksi keskitytään siihen, miten datasta saadaan hyödyllistä informaatiota. Osiossa käsitellään datan louhimista, data-analytiikkaa, datan analysointimenetelmiä sekä datan visualisoinnin merkitystä. Seuraavassa luvussa tutkitaan tekoälyä osana datan käsittelyä ja selvitetään, millä eri tavoin koneoppimista hyödynnetään datan käsittelyssä. Sen jälkeen perehdytään sähkönkulutukseen ja siihen vaikuttaviin tekijöihin, teollisuuden datan keruuseen sekä siihen, mitä haasteita datan hyödyntämiseen liittyy teollisuusympäristössä. Opinnäytetyön lopuksi käsitellään olemassa olevaa data-aineistoa liittyen sähkönkulutuksen seurantaan ja ennakoidaan sähkönkulutusta hyödyntäen koneoppimisen muotona ohjattua oppimista.

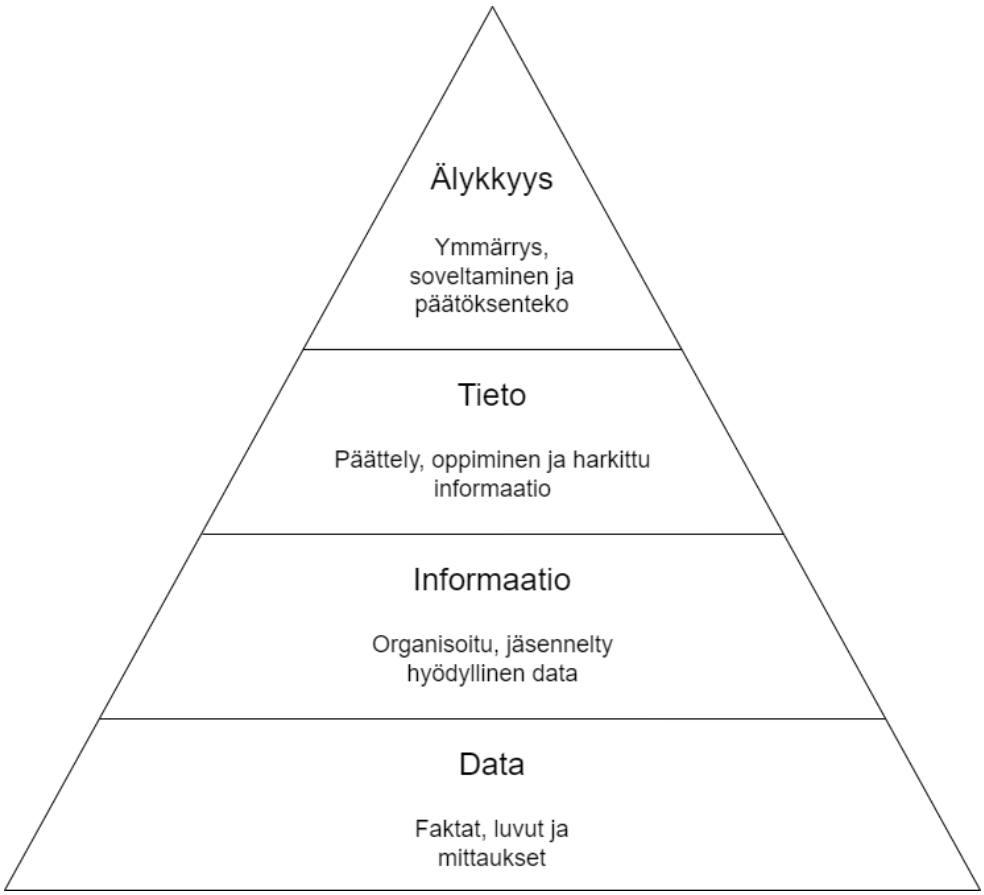
Kaikki esitellyt kuviot ja ajatuskartat ovat itsetehtyjä, mukaillen kussakin kuviossa viitattuun lähdemateriaaliin. Olemassa olevien data-aineistojen avulla havainnoitujen esimerkkien koodipohjat on lisätty liitteisiin. Koko ohjelmakoodi, johon työn viimeinen pääluku ja sen kaikki kuvat ja kuviot perustuu, on lisätty viimeiseksi liitteeksi. Tekoälyä on hyödynnetty apuna englanninkielisten lähdemateriaalien termien ja lauserakenteiden suomentamisessa sekä ohjelmakoodin ongelmanratkaisussa, erityisesti pienempien yksityiskohtien ja toimintojen osalta.

2 DATAN KÄSITTELY

2.1 Tiedon tasot

Data on prosessoimatonta faktatietoa, jolla ei ole itsessään merkitystä tai tarkoitusta (Annansingh & Bon Sesay 2022, 2). Data voi olla esimerkiksi bittejä, numeroita, tekstiä, ääntä tai kuvia, ja sijaita esimerkiksi laitteissa (IoT), sensoreissa, fyysisissä arkistoissa, verkossa tai organisaatioiden tietokannoissa. Datan määrän ja dataa hyödyntävien menetelmien lisääntyessä on yhä tärkeämpää huomioida datan laatu ja ymmärtää, miten dataa jalostamalla saadaan informaatiota ja edelleen informaatiosta organisaatioita hyödyntävää tietoa. (Annansingh & Bon Sesay 2022, 2.)

Tieto on laaja käsite, mutta sitä pystytään kuitenkin jäsentelemään eri menetelmillä. Yksi yleinen jäsentelytapa tiedon kuvaamiseen on käyttää kolmea eri käsitettä: data, informaatio ja tietämys. (Laihonen ym. 2013, 18.) Näiden kolmen käsitteen lisäksi puhutaan akateemisessa kirjallisuudessa myös korkeammasta tasosta, jota kutsutaan viisaudeksi tai älykkyydeksi (Laihonen ym. 2013, 18). Kuviossa 1 ovat nämä neljä tiedon eri tasoa mukaillen Annansinghin ja Bon Sesayn (2022, 4) hahmotelmaa.



KUVIO 1. Tiedon tasot (mukaillen Annansingh & Bon Sesay 2022, 4).

Data on usein käsittelemätöntä ja sitä kutsutaan siksi yleisesti raakadataksi. Datatasolla on tärkeää kyetä keräämään oikeellista ja luotettavaa dataa, mutta haasteena on nykyaikana saatavilla olevan datan määrän eksponentiaalinen kasvu sekä eri datatyyppien yleistyminen. (Annansingh & Bon Sesay 2022, 2.)

Informaatio on rakenteellista dataa, jota on siivottu ja organisoitu, jotta sitä pystytään käyttämään data-analyysissä hyödyllisen tiedon löytämiseksi. Informaatiolle voidaan soveltaa esimerkiksi datan louhintaa ja data-analytiikan eri osa-alueita, joiden avulla datasta tuotetaan tietoa tai tietämystä (Annansingh & Bon Sesay 2022, 3). Datan louhintaa käsitellään myöhemmin luvussa 2.3 sekä data-analytiikan osa-alueita luvussa 2.4.1.

Tieto on hyödyllistä ja harkittua informaatiota (Annansingh & Bon Sesay 2022, 3). Toinen tiedon tason erottelu voidaan tehdä hiljaisen ja eksplisiittisen tiedon (eng. implicite and explicit knowledge) välillä (Laihonen ym. 2013, 18). Hiljaista tietoa kutsutaan osaamiseksi tai intuitioksi ja se on yleisesti kokemuksen kautta siirtyvää tietämystä. Hiljaisen tiedon siirtäminen henkilöltä toiselle henkilölle on

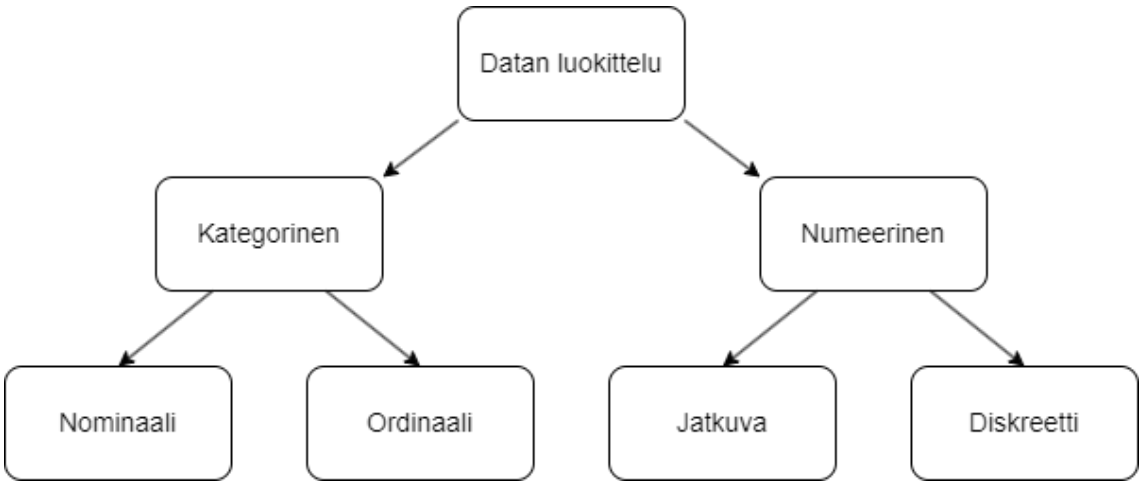
yleisesti haastavaa. Eksplisiittinen tieto vastaavasti on usein helposti tunnistettavaa, esimerkiksi kirjalliseen muotoon puettua tietoa. Tämä tieto on helpompi siirtää henkilöiden välillä sekä tallettaa, kuten muun muassa eri kielet. (Laihonen ym. 2013, 18.)

Annansinghin ja Bon Sesayn (2021, 5) mukaan tiedon ylin taso eli älykkyys, viisaus tai totuus (eng. intelligence) on kyky soveltaa ja kontrolloida jotain toimintaa. Ylimmän tason piirteitä ovat syy- ja seuraussuhteiden syvällinen ymmärtäminen sekä toimiminen mahdollisimman järkevästi ja taloudellisesti suosiollisella tavalla monissa eri tilanteissa. Älykkyyden tasoon voidaan liittää myös tekoäly, koneäly, kognitiiviset robotit ja tietokoneet, sekä muut erilaiset älykkäät systeemit. (Annansingh & Bon Sesay 2021, 5.)

2.2 Datan luokittelu

Data voidaan yleisesti jakaa kategoriseen ja kvantitatiiviseen luokkaan (Watkins 2016, 3). Kategorisella datalla tarkoitetaan sellaista dataa, jonka voi luokitella itsessään jo johonkin ryhmään, luokkaan tai kategoriaan. Kategorinen data voi olla joko nominaalista (järjestäytymätöntä), kuten sukupuoli, tai ordinaalista (järjestyksellistä), kuten koulutustaso, jonka sisällä vallitsee tietty hierarkia eli järjestys (Watkins 2016, 3).

Numeerinen eli kvantitatiivinen data on jokin mitattavissa oleva arvo tai havainto, ja se voi olla tyypiltään joko diskreettiä tai jatkuvaa (Nelli 2023a). Diskreettejä arvoja pystytään laskemaan yhteen, ja ne ovat loogisesti eteneviä, keskenään eriarvoisia, kuten esimerkiksi aikasarjadatan arvot. Jatkuvat arvot voivat olla mitä tahansa määritetyllä arvovälillä, kuten paino tai ikä, ja mitä ei yleisesti kannata laskea yhteen. (Nelli 2023a.) Kuviossa 2 ovat koottuna datan eri luokat mukaillen Watkinsin (2016, 3) ja Nellin (2023a) määritelmiä.



KUVIO 2. Datan luokittelu (mukaillen Watkins 2016, 3 ja Nelli 2023a).

Ennen kun alkaa käsittelemään ja analysoimaan dataa, on tärkeää määritellä minkälaista dataa tulee käsittelemään, mistä data on peräisin, onko se luotettavaa ja mikä data on kyseisen tutkimuksen tai kokeen kannalta merkityksellistä. Datan oikea luokittelu auttaa valitsemaan tilanteeseen sopivat datan analysointimenetelmät, mikä vastaavasti auttaa hyödyllisen tiedon löytämisessä ja oikeiden johtopäätöksien muodostamisessa. (Watkins 2016, 4).

2.3 Datan louhinta

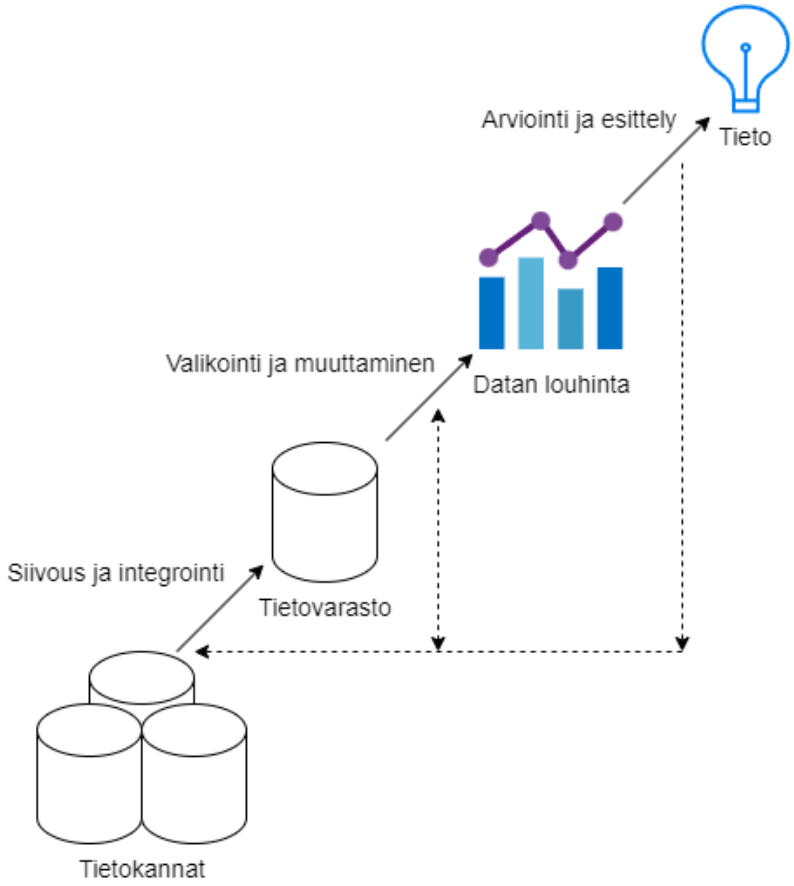
Jotta datasta saadaan hyödyllistä tietoa, sitä pitää käsitellä ja analysoida. Tärkeänä osana datan analysointia ja sen kehittyneempänä, nykyaikaisena versiona voidaan pitää datan louhintaa (eng. data mining) (Han, Pei & Kamber 2012, 3). Datan louhinta on poikkitieteellinen termi, joka voidaan määrittää monella eri tavalla. Yleisesti voidaan kuitenkin todeta sen tarkoittamista datan jalostamista tiedoksi. (Han ym. 2012, 2.)

Annansinghin ja Bon Sesayn (2022, 13) mukaan datan louhinnan pääasiallisia ominaisuuksia ovat

- mallien ja yhtymäkohtien luominen trendeihin ja käytökseen perustuen
- todennäköisten tapahtumien ennustaminen ja niihin reagointi
- päätöksentekoon suuntautuneen tiedon ja informaation lisääminen
- keskittyminen suurien datajoukkojen tutkimiseen sekä
- visuaalinen dokumentointi ja uusien faktojen havainnollistaminen.

Datan louhinta muun muassa auttaa yrityksiä ja organisaatioita oppimaan enemmän asiakkaiden tottumuksista kehittääkseen tehokkaampaa markkinointia ja parantaakseen asiakassuhteita. Lisäksi se auttaa tuotannossa ja teollisuudessa vähentämään riskejä ja kustannuksia, sekä lisäämään kannattavuutta. (Annansingh & Bon Sesay 2022, 13.)

Datan louhinnan merkitys kasvaa datan lisääntyessä, koska myös informaation ja datan välimatka kasvaa. Tästä voidaan päätellä, että informaation saamiseen kuluu yhä enemmän aikaa ja energiaa. (Annansingh & Bon Sesay 2022, 13.) Datan louhintaa pidetään yhtenä tiedon löytymisen prosessin (eng. knowledge discovery) osa-alueena, jossa sovelletaan älykkäitä menetelmiä hyödyllisen tiedon löytämiseksi (Han ym. 2012, 7). Kuviossa 3 on datan louhinta osana tiedon löytymisen prosessia mukaillen Hanin ym. (2012, 7) hahmotelmaa.



KUVIO 3. Datan louhinta osana tiedon löytämisen prosessia (mukaillen Han ym. 2012, 7).

Tietokannat (eng. databases) sisältävät yleisesti kaiken halutun datan jostain tietystä lähteestä (Han ym. 2012, 6). Yleinen menettely on kohdistaa

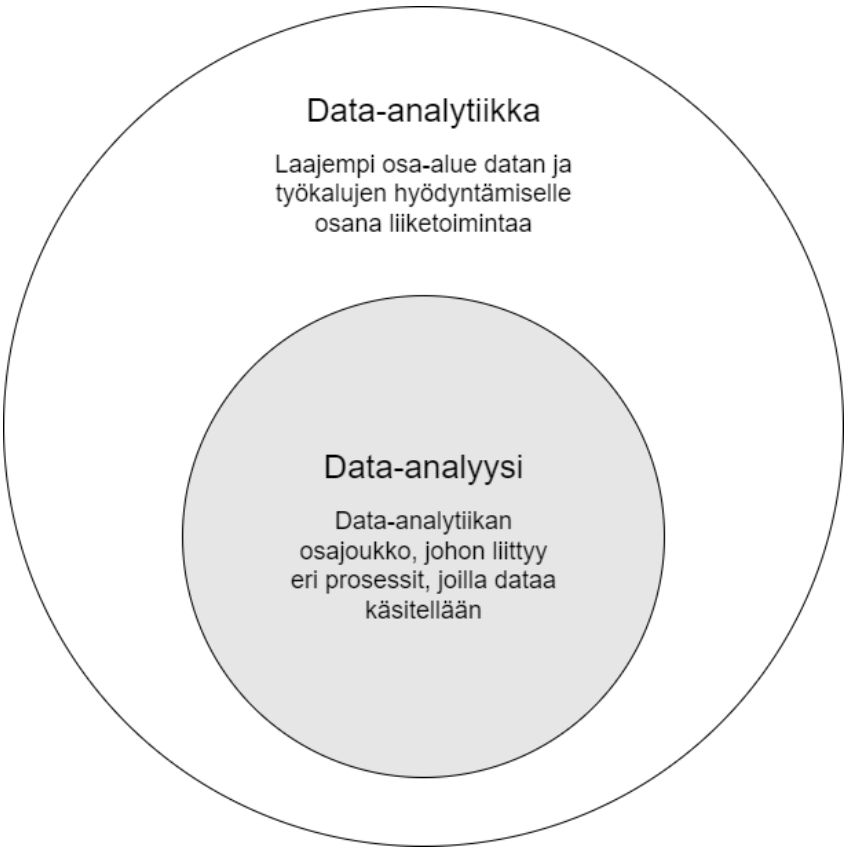
tietokannassa olevaan dataan sen siivous ja integrointi (kuvio 3), jossa siitä erotetaan kohina (eng. noise) ja epäjohdonmukainen tai käyttökelvoton data sekä mahdollisesti yhdistetään eri datan lähteitä yhteen (Han ym. 2012, 6).

Tietovarastossa (eng. data warehouse) on koottuna käyttökelpoinen data, josta tietoa on myöhemmin mahdollista louhia ja siten tuoda esille uutta hyödyllistä informaatiota (Han ym. 2012, 8). Tietovarastossa oleva data valikoidaan ja tarvittaessa muutetaan (kuvio 3) datan louhintaan sopivaksi, kuten esimerkiksi taulukoksi tai matriisiksi. Siivousta ja integrointia sekä valikointia ja muuttamista pidetään yleisesti datan esikäsittelynä. (Han ym. (2012, 8.)

Datan louhinnan avulla saadusta informaatiosta voidaan tunnistaa malleja (eng. patterns), tehdä arviointeja ja esitellä malleja, luoden niistä tietoa. Kuvion katkoviivat kuvaavat mahdollisuutta hyödyntää jo opittua (eng. feedback), kerätä uutta dataa tai käsitellä samaa dataa toisella menetelmällä. (Han ym. 2012, 8.) Datan louhintaa sovelletaan monella eri osaamisalueella informaatioteknologiassa, kuten tekoälyssä ja koneoppimisessa (eng. machine learning). Tekoälyn tarkoitus datan louhinnassa on sen automatisointi, mukautuminen uuteen dataan sekä louhinnan soveltaminen eri datatyypeille, kuten kuville ja äänelle (Han ym. 2012, 24).

2.4 Data- ja visuaalinen analytiikka

Dataan, datan analysointiin ja tekoälyyn liittyy yleisesti paljon eri käsitteitä ja termejä, joiden määritelmät saattavat vaihdella. On tyypillistä, että kaksi henkilöä puhuu samoista asioista, kuten esimerkiksi data-analytiikasta ja data-analyysista, mutta eri nimillä. (Pöyry 2024.) Kuviossa 4 erotellaan nämä kaksi määritelmää toisistaan mukaillen Kiddin ja Hornayn (2021) hahmotelmaa.



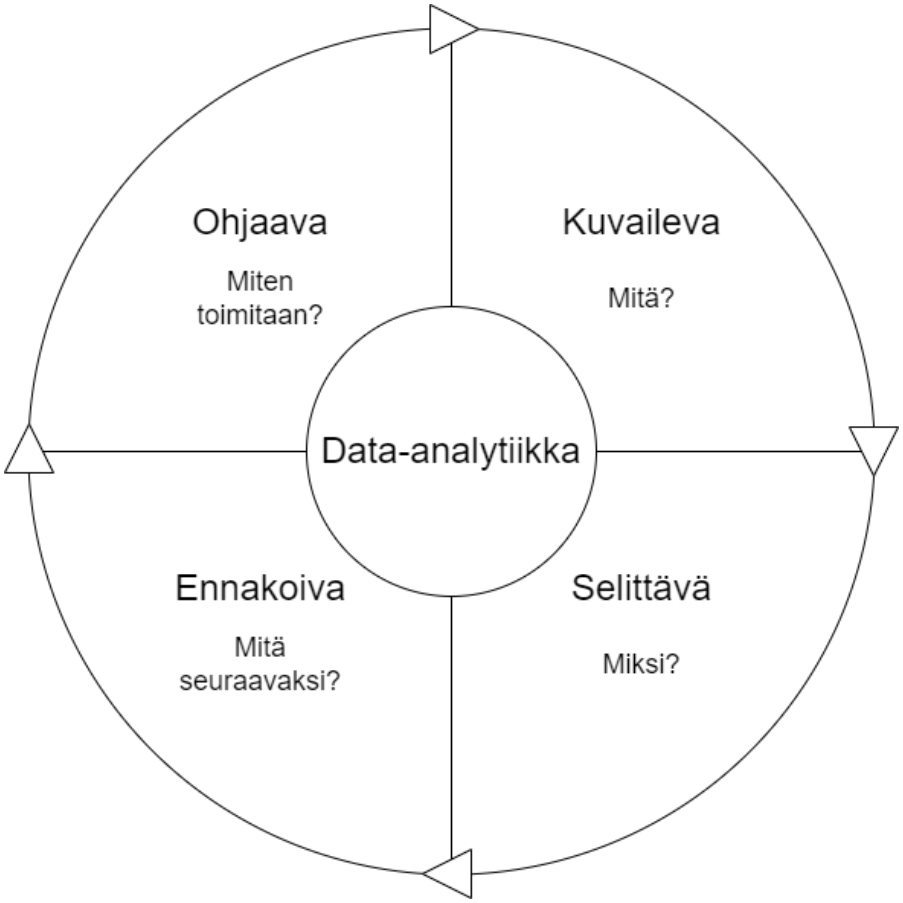
KUVIO 4. Data-analytiikan ja data-analyysin erottelu (mukaillen Kidd & Hornay 2021).

Data-analytiikka määritetään yleisesti laajempana kenttänä datan ja työkalujen käytölle osana liiketoimintaa ja sisältää kaikki ne vaiheet, jotka löytävät, tulkitsevat ja esittävät dataa (Kidd & Hornay 2021). Vastaavasti data-analyysilla tarkoitetaan data-analytiikan osajoukkoa, jonka alle sisältyy vain ne yksityiskohtaiset prosessit, joilla tiettyä data-aineistoa esikäsitellään ja analysoidaan hyödyllisen informaation löytämiseksi. Data-analytiikka on vahvasti osana päätöksen tekoa organisaatioissa ja sen avulla vaikutetaan esimerkiksi prosessien tehokkuuteen, ennakoivaan suunnitteluun, riskienhallintaan sekä asiakaskokemuksen parantamiseen. (Kidd & Hornay 2021.)

2.4.1 Data-analytiikka

Liebowitzin (2021, 23) mukaan data analytiikkaan kuuluu neljä osa-aluetta: kuvaileva (eng. descriptive), selittävä (eng. diagnostic), ennakoiva (eng. predictive) ja ohjaava (eng. presciptive). Jokaisella näistä osa-alueista on eri

tavoite ja paikka data-analytiikan prosessissa (Liebowitz 2021, 23). Kuviossa 5 ovat esiteltynä data-analytiikan osa-alueet mukaillen Liebowitzin (2021, 24) hahmotelmaa.



KUVIO 5. Data-analytiikan osa-alueet (mukaillen Liebowitz 2021, 24).

Kuvaileva analytiikka analysoi raakadataa antaakseen arvokasta tietoa vertaamalla menneitä tapahtumia. Se vastaa kysymykseen mitä on tapahtunut, sekä mahdollisesti mikä ongelma on todettu. (Liebowitz 2021, 23). Sähkönkulutuksen seurannassa voidaan huomata esimerkiksi sähkönkulutuksen nouseminen ja laskeminen tiettyinä vuodenaikoina.

Vuodenaikojen vaihtelu vaikuttaa esimerkiksi lämpötilaan, mikä selittää sähkönkulutuksen muutoksen esimerkiksi kuluttajatasolla lisääntyvän kotien lämmityksen tai jäähdytyksen tarpeena. Selittävä analytiikka antaa siis lisätietoa ongelmasta ja pyrkii vastaamaan miksi jotain on tapahtunut (Liebowitz 2021, 23). Selitettävässä analytiikassa on vähintään kaksi muuttujaa: selitettävä muuttuja ja

selittävä muuttuja, ja ne voivat datatyypin mukaan olla joko kategorisia tai numeerisia (kuvio 2) (Pöyry 2024).

Ennakoiva analytiikka kertoo, mitä seuraavaksi todennäköisesti tapahtuu ja käyttää siinä apuna sekä kuvailevan että selittävän menetelmän löydöksiä. Lisäksi apuna käytetään yleensä koneoppimista ja datan analysointimenetelmiä, kuten luokittelua, klusterointia ja poikkeuksien huomaamista tulevaisuuden ennakoimisessa. (Liebowitz 2021, 23.) Eri datan analysointimenetelmiä kuvataan tarkemmin luvussa 2.5. Esimerkiksi lämpötilan laskiessa sähkönkulutus kasvaa lisääntyneen lämmitystarpeen myötä, jolloin voidaan havaita yhteys sähkönkulutuksen ja lämpötilan välillä.

Lämpötilan ollessa alhainen suuren osan vuodesta voidaan kehittää keinoja, joilla parannetaan esimerkiksi sähköntuotannon tehokkuutta ja kustannuksia tai parannetaan rakennusinfrastruktuuria. Viimeinen menettely eli ohjaava analytiikka kertookin, miten ongelman kanssa toimitaan, ja se hyödyntää kaikkia kolmea muuta menetelmää määrittääkseen järkevimmän menettelytavan kyseisen ongelman ratkaisemiseksi. Ohjaava analytiikka sisältää myös ratkaisun esittelyn sekä mahdollisen visualisoinnin, ja ongelmanratkaisussa käytetään nykyään poikkeuksetta tekoälyä ja koneoppimista. (Liebowitz 2021, 23.)

2.4.2 Visuaalinen analytiikka

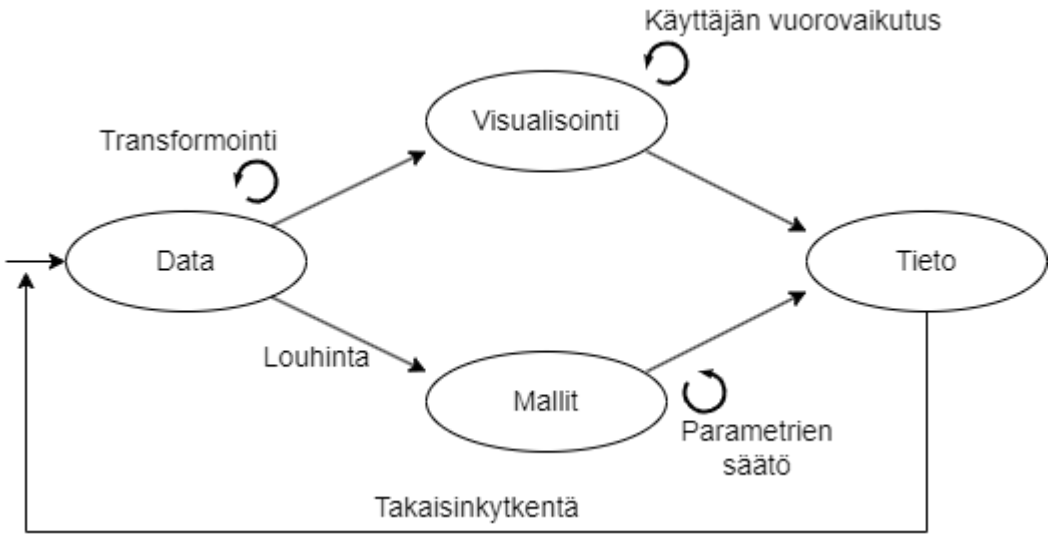
Visuaalisuus ja datan visuaalinen esittämismuoto on monesti helpompi ymmärtää kuin perinteinen numeerinen tai laskennallinen muoto. Lisäksi datan visuaalisuus tuo esiin uusia näkökulmia ja myös virheitä datassa, jotka saattaisivat muuten jäädä huomaamatta. (Keim ym. 2010, 2–3.)

Visuaalinen analytiikka tarkoittaa eri analysointimenetelmien ja visualisoinnin yhdistämistä osana päättelyä, ymmärtämistä ja päätöksen tekoa kompleksisessa ja suuressa määrässä dataa (Keim ym. 2010, 7). Se mahdollistaa ihmisen

- havaitsemaan odotettu tieto ja löytämään odottamaton tieto
- yhdistämään informaatiota ja tuottamaan tietoa suuresta määrästä dataa
- tuottamaan ajankohtaisia, päteviä ja ymmärrettäviä arvioita sekä

• kommunikoimaan näitä arvioita tehokkaasti toiminnan aikaansaamiseksi (Keim ym. 2010, 7).

Datan visualisointi voidaan nähdä myös osana tiedon löytämisen prosessia (kuvio 3). Jotta datasta saadaan tietoa, visuaalisen analytiikan prosessi yhdistää automaattiset datan louhintamenetelmät ja visuaaliset analyysimenetelmät ihmisen vuorovaikutuksen kautta. Kuviossa 6 on esiteltynä prosessi mukaillen Keimin ym. (2010, 10) hahmotelmaa.



KUVIO 6. Visuaalisen analytiikan prosessi (mukaillen Keim ym. 2010, 10).

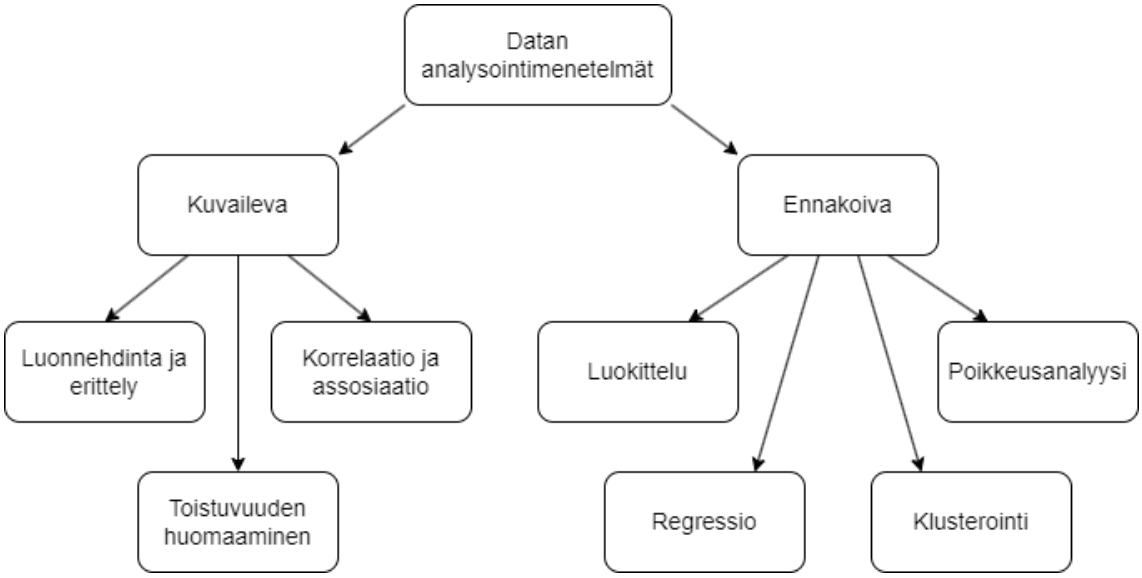
Kuviossa kuvataan datan haarautumista datan visualisointiin sekä datan louhinnan avulla saatuihin malleihin. Käyttäjällä on tällöin mahdollisuus valita etenemismenetelmä tiedon tuottamiseen ja huomioida takaisinkytkennän kautta (eng. feedback loop) jo tuotettua tietoa, ja tämän avulla tuottaa uutta tietoa toisella menetelmällä. (Keim ym. 2010, 11.)

Datan louhinta on yleisesti automaattinen, tekoälyä kuten koneoppimista hyödyntävä tapa, kun taas visualisointi vaatii enemmän käyttäjän vuorovaikutusta (eng. user interaction) tiedon löytämiseksi. Yksi etenemistapa on ensin analysoida dataa, visualisoida saadut tärkeimmät havainnot, jonka jälkeen analysoida näiden pohjalta yksityiskohtia ja luoda niitä tukevia visualisointeja. Visuaalinen analytiikka keskittyy ihmisen vuorovaikutuksen kautta myös enemmän syy-seuraussuhteisiin kuin datan louhinta tekoälyn avulla. (Keim ym. 2010, 11.)

2.5 Datan analysointimenetelmät

Data-analyysiin sisältyy sekä datan esikäsittely että datan analysointi, mutta data on rajoitettu yleensä yhteen tiettyyn tarkasteltavaan data-aineistoon (Kidd & Hornay 2021). Datan analysointimenetelmiä on monia ja ne voidaan yleisesti jakaa kuvailevaan ja ennakoivaan (eng. descriptive and predictive) menetelmään (kuvio 5) (Han ym. 2012, 15).

Kuvailevat menetelmät luonnehtivat data-aineiston eri olemassaolevia ominaisuuksia, kun taas ennakoivat menetelmät käyttävät koneoppimisen ehtoja ja sääntöjä (eng. rule induction) ennustuksien määrittämiseen datasta. (Han ym. 2012, 15) Kuviossa 7 ovat luokiteltuna eri datan analysointimenetelmät mukaillen Hanin ym. (2012, 15–21) määritelmiä.



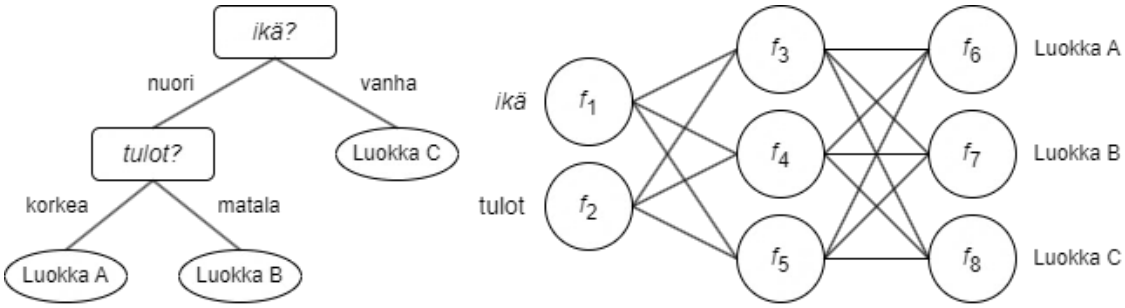
KUVIO 7. Datan analysointimenetelmät (mukaillen Han ym. 2012, 15–21).

Kuvailevat menetelmät ovat esimerkiksi datan luonnehdintaa (eng. characterisation) ja erittelyä (eng. discrimination). Esimerkiksi voidaan huomata jokin tietty henkilöprofiili esimerkiksi ostokäyttäytymisen perusteella ja vastaavasti erotella hyvin erilaiset profiilit toisistaan ja tehdä päätelmiä niiden avulla. (Han ym. 2012, 17.)

Kuvailevia menetelmiä voivat olla myös toistuvuuden huomaaminen (eng. frequent patterns) sekä korrelaatio ja assosiaatio. Esimerkiksi tiettyjen tuotteiden

ostaminen samalla kertaa luo riippuvuussuhteen eli korrelaation näiden tuotteiden välille, ja ne voidaan assosioida silloin yhdessä toistuvana mallina (eng. pattern). (Han ym. 2012, 17.)

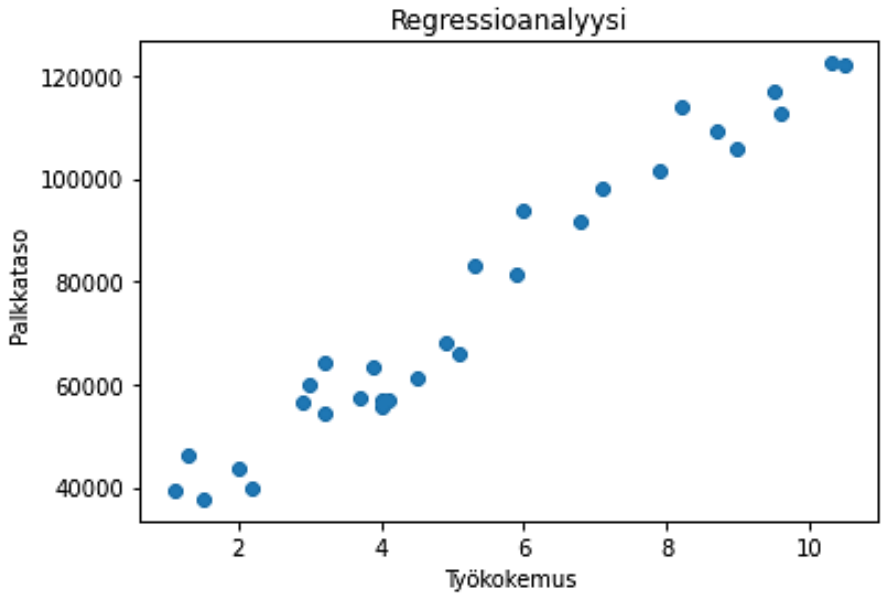
Luokittelu (eng. classification), regressio (eng. regression), klusterointi (eng. cluster analysis) sekä poikkeusanalyysi (eng. outlier analysis) ovat esimerkkejä yleisistä ennakoivista menetelmistä. Luokittelun ideana on organisoida kategorista ja nominaalista dataa (kuvio 2) ryhmiin käyttäen loogisia sääntöjä ja ehtoja, kuten JOS-NIIN operaatioita, päätöspuita (eng. decision tree) tai neuroverkkoja (eng. neural network). (Han ym. 2012, 18). Kuviossa 8 ovat kuvattuna päätöspuun ja neuroverkon rakenne mukaillen Hanin ym. (2012, 18) hahmotelmaa.



KUVIO 8. Päätöspuun (vas.) ja neuroverkon rakenne (mukaillen Han ym. 2012, 18).

Päätöspuu käyttää päätöksenteossa haarautuvia sääntöjä, kun taas neuroverkko koostuu kerroksista, jotka määrittelevät kaikki mahdollisesti vaihtoehdot ihmisten aivojen neuronien toimintaa mukaillen. Satunnaismetsä (eng. random forest) vastaavasti koostuu monista päätöspuista. (Han ym. 2012, 19.) Ennakointimenetelmä on tärkeä osa muun muassa sähkönkulutuksen seurantaa, jotta organisaatiot pystyvät reagoimaan ajoissa kulutukseen vaikuttavien tekijöiden muuttuessa (Salam & El Hibaoui 2018a).

Regressiossa tutkitaan yleensä numeerista ja datana jatkuvaa, yhden tai useamman selittävän muuttujan yhteyttä selitettävään muuttujaan (Han ym. 2012, 19). Tekemässäni regressioanalyysin esimerkissä (liite 1) verrataan työkokemuksen kertymistä palkkatasoon (kuvio 9).



KUVIO 9. Esimerkki regressioanalyysista (liite 1).

Kuvion simuloidussa data-aineistossa ennakoidaan työkokemuksen karttumisen vaikuttavan lineaarisesti palkkatason nousuun (lineaarinen regressio), jolloin voidaan todeta suora yhteys näiden muuttujien välillä. Lineaarista regressiota käytetäänkin apuna kausaliteetin eli syy-seuraussuhteen todentamisessa eri muuttujien välillä (Watkins 2016, 33.)

Klusteroinnissa pyritään muodostamaan kategorioita tai ryhmiä, joita ei välttämättä tiedetä etukäteen (Pöyry 2024). Klusteroinnin seurauksena havaitaan, että tietyt havainnot tai datapisteet ovat todella samankaltaisia ja lähellä toisiaan, kun taas vastaavasti tietyt havainnot ovat kauempina muista keskittymistä luoden oman keskittymän eli klusterin. Tätä menetelmää käytetään esimerkiksi markkinoinnin kohdistamiseen tiettyyn asiakasryhmään. (Han ym. 2012, 20.) Klusterointia käsitellään vielä tarkemmin osana koneoppimista luvussa 3.2.2.

Poikkeusanalyysissa tunnistetaan muusta datasta huomattavasti poikkeavia havaintoja eli anomalioita. Monet datan käsittelymenetelmät suodattavat näitä anomalioita pois kohinana tai viallisena datana osana datan esikäsittelyä. (Han ym. 2012, 20.) Kuitenkin esimerkiksi mittalaitteen mittausdatassa tai petoksen havaitsemisessa (eng. fraud detection) nämä poikkeukset voivat antaa arvokasta tietoa laitteen kunnosta tai luottokorttien huijauskäytöstä normaalista poikkeavien tai muun muassa suurien ostotapahtumien vuoksi. (Han ym. 2012, 21.)

3 TEKOÄLY JA KONEOPPIMINEN

3.1 Tekoäly

Tekoäly (eng. Artificial Intelligence) on termi ja tieteenala, millä ei ole yhtä yksiselitteistä määritelmää (Nelli 2023b). Ertelin ja Blackin (2011, 2) lainauksen mukaan “tekoäly on tieteenala, joka tutkii miten tietokoneet saadaan tekemään asioita, joissa ihmiset ovat tällä hetkellä parempia”. Yleisesti voidaan todeta, että tekoälyn olemassaolon tarkoitukseen ja kehitykseen liittyy vahvasti autonomisuus sekä adaptiivisuus. Autonomialla tarkoitetaan jonkin systeemin toimimista itsenäisesti, ja adaptiivisuudella systeemin kykyä oppia ja ottaa vastaan uutta informaatiota. (Helsingin Yliopisto & MinnaLearn 2025.)

Tekoälyä sovelletaan yhä laajemmin tietokoneiden prosessointikyvyn kasvaessa ja algoritmien kehittyessä. Tämä mahdollistaa nykyaikana tekoälyn hyödyntämisen muun muassa visuaalisissa ja auditiivissa operaatioissa, joita ennen pidettiin pelkästään ihmiselle eksklusiivisina, kuten

- kuvan luokittelu
- kohteiden tunnistaminen kuvasta
- kohteiden erottelu kuvasta
- kielen kääntäminen ja tulkkaus
- luonnollisen kielen ymmärtäminen sekä
- puheentunnistus. (Nelli 2023b.)

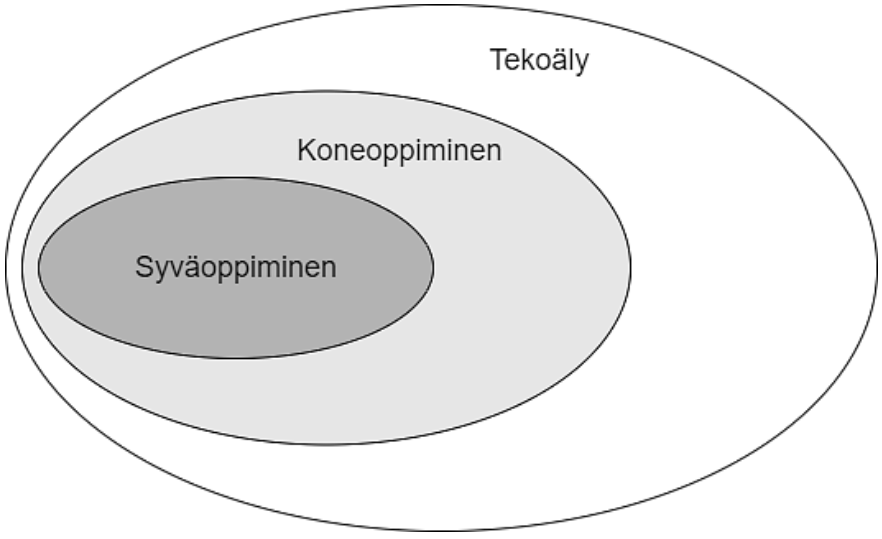
3.1.1 Data-analytiikka ja tekoäly

Nykyaikana datan määrän tulviessa (eng. data deluge) on kehittynyt suuri tarve automatisoida datan käsittelyä ja analysointia. Tämän takia tekoälyä ja tarkemmin sanottuna koneoppimista, on alettu hyödyntämään data-analytiikassa. Tämän ansiosta organisaatiot pystyvät edelleen keskittymään data-analytiikassa strategiaan ja datan hyödyntämiseen liiketoiminnassa, eikä ihmisen tarvitse olla niin aktiivisesti osana datan tulkintaa ja tiedon löytämistä. (Murphy 2012, 1.)

Liebowitzin (2021, 25) näkemyksen mukaan analyytikko voi tarvita hypoteesien ja päätelmien muodostamiseen tunteja, päiviä tai jopa viikkoja, kun taas koneoppiminen mahdollistaa näiden sekä korrelaation erottamisen kausaliteetista alle sekunneissa. Näin saadaan paljon ajankohtaisempaa tietoa datasta, millä on suuri hyöty nykypäivän liiketoiminnassa, kuten tuotantoprosesseissa, mittauksissa ja muussa datankeruussa. Suurien tietoaineistojen (eng. big data) aikakaudella ajantasaisuus ja nopea reagointi on edelletys yritysten, kuten sähköyhtiöiden pysymiseen edellä dynaamisessa markkinatilanteessa. (Liebowitz 2021, 25.)

3.1.2 Tekoälyn kerrokset

Tekoäly kehittyy jatkuvasti datan ja tietokoneiden grafiikkasuorittimien (GPU) prosessointikyvyn kasvaessa. Viime vuosikymmeninä tekoälyn alle on siten muodostunut koneoppimisen haara ja viime vuosikymmenen aikana koneoppimisen alle edelleen syväoppimisen haara (eng. deep learning). (Nelli 2023b.) Kuviossa 10 ovat kuvattuna näiden kolmen suhde mukaillen Nellin (2023b) hahmotelmaa.



KUVIO 10. Tekoälyn, koneoppimisen ja syväoppimisen suhde (mukaillen Nelli 2023b).

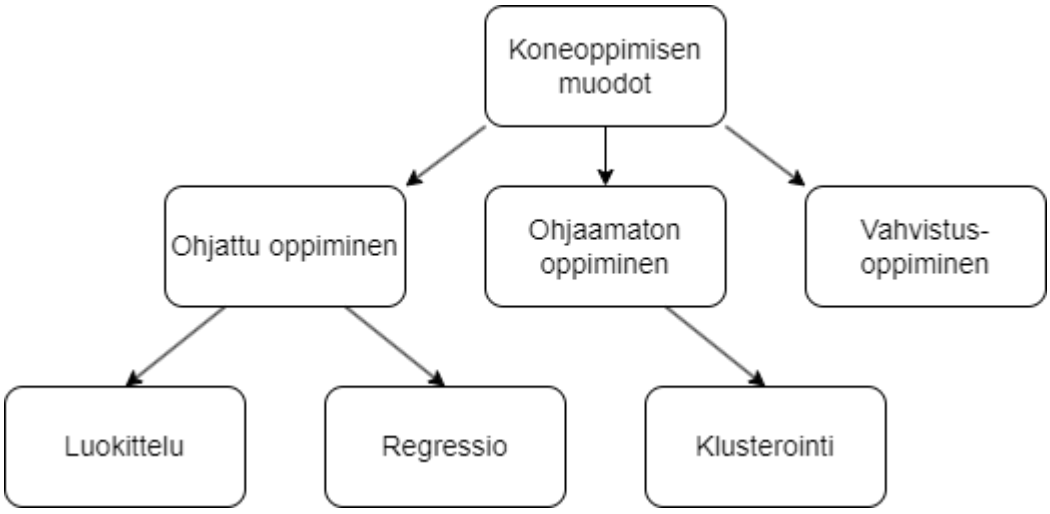
Yleisesti koneoppiminen ja syväoppiminen liittyvät vahvasti nimenomaan oppimiseen ja adaptiivisuuteen. Syväoppiminen on edistänyt kielten sekä kuva-

ja äänidatan käyttämisen koneoppimisessa ja tämä haara on jatkuvassa kehityksessä. (Nelli 2023b.)

3.2 Koneoppiminen

Koneoppiminen (eng. machine learning) on tekoälyn alaluokka, jonka tarkoitus on jalostaa datasta tietoa käsittelemällä ja analysoimalla dataa automaattisesti (Murphy 2012, 1). Koneoppimisen syntyyn on vaikuttanut yhtälailla sekä datamäärän kasvu, että tietokoneiden prosessointikyvyn kehittyminen. Nykyaikana koneoppimisen funktio on käytännössä suurien tietoaineistojen (eng. big data) louhinta hyvin tehokkaasti sekä louhinnasta saadun tiedon hyödyntäminen. (Murphy 2012, 1.)

Koneoppiminen voidaan jakaa kahteen yleisimpään oppimisen muotoon sekä niiden yleisemmin käytettyihin, osin aiemmin mainittuihin analysointimenetelmiin. Nämä oppimismuodot ovat nimeltään ohjattu oppiminen (eng. supervised learning) ja ohjaamaton oppiminen (eng. unsupervised learning). Näiden lisäksi on olemassa myös vähemmän käytetty muoto, vahvistusoppiminen (eng. reinforced learning). Kuviossa 11 ovat esiteltynä kolme yleisintä koneoppimisen muotoa mukailen Murphyn (2012, 2) ja Pöyryn (2024) määritelmiä.



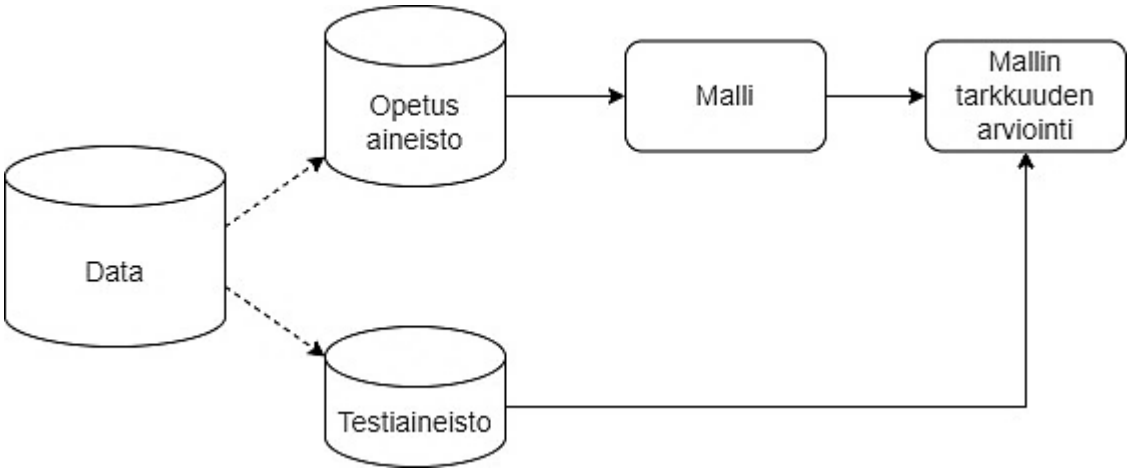
KUVIO 11. Koneoppimisen muodot (mukaillen Murphy 2012, 2 ja Pöyry 2024).

Oikean oppimismuodon valinta on tärkeää oman päämäärän ja kerätyn datan perusteella, koska jokaisella näistä on erilainen lähestymistapa datan

analysointiin, mallin luomiseen sekä tiedon löytämiseen. Ohjatussa oppimisessa mallille annetaan syöte eli millä opetetaan ja haluttu tulos eli mitä opetetaan, ohjaamattomassa oppimisessa mallille annetaan vain syöte ilman erillistä valvontaa ja vahvistusoppimisessa annetaan syöte ja erillinen algoritmi ohjaa mallin toimivuutta palautteen mukaan. (Murphy 2012, 2.)

3.2.1 Ohjattu oppiminen

Sekä ohjatussa että ohjaamattomassa oppimisessa data jaetaan ensin opetus- ja testiaineistoon (eng. training and test data). Mallin opetus tapahtuu opetusaineistolla ja sen tarkkuutta voidaan vastaavasti arvioida testiaineistolla. (Han ym. 2012, 370.) Kuviossa 12 on esiteltynä datan käsittelyprosessi koneoppimisessa mukaillen Hanin ym. (2012, 370) hahmotelmaa.



KUVIO 12. Datan käsittelyprosessi koneoppimisessa. (mukaillen Han ym. 2012, 370).

Opetusaineiston luomaa algoritmia voidaan pitää luonteeltaan pessimistisenä, koska vain osaa data-aineistosta käytetään mallin luomiseen. Data-aineiston jako tehdään yleensä opetusaineiston osuuden ollessa selvästi suurempi, noin kaksi-kolmasosaa koko datasta, jotta itse mallin opettamiseen olisi mahdollisimman paljon resursseja eli dataa käytettävissä. Jako voi olla radikaalimpikin, mikäli data-aineiston koko on huomattavan suuri, jolloin testiaineistoon jää silti tarvittava määrä dataa mallin tarkkuuden arviointia varten. Testiaineisto ei saa kuitenkaan

olla myöskään liian pieni, jolloin mallin tarkkuuden arviointi kärsii. (Han ym. 2012, 370.)

Ohjatun oppimisen tai ennakoivan oppimisen tavoite on muuntaa syötteet (eng. inputs) tuloksiksi (eng. outputs) opetusaineiston eli syöte-tulos parien avulla (Murphy 2012, 2). Opetusalgoritmia voidaan havainnollistaa kaavalla 1,

$$D = \{(x_i, y_i)\}_{i=1}^{N}, \quad (1)$$

jossa D on opetusaineisto, x_i on syöte, y_i on tulos ja N on aineiston koko eli syöte-tulos parien määrä. Yksinkertaisimmassa muodossa syöte x_i on *D*-ulotteisten lukujen vektori esimerkiksi henkilön painosta ja pituudesta, ja sitä kutsutaan piirteeksi tai ominaisuudeksi (eng. feature or attribute). (Murphy 2012, 2) Usean vektorin muodostamaa datajoukkoa eli taulukkoa käsitellään tarkemmin luvuissa 5.1.1 ja 5.1.2.

Tulos y_i voi olla joko kategorinen muuttuja, kuten henkilön sukupuoli, tai numeerinen muuttuja, kuten henkilön paino (kuvio 2), jota halutaan mallin avulla opettaa. Tuloksen y_i ollessa kategorinen, ongelma tunnistetaan luokittelevana ja jos se on numeerinen, ongelma tunnistetaan regressiona. (Murphy 2012, 2.) Ohjattu oppiminen esimerkiksi pyrkii luokittelemaan jonkin henkilön sukupuolen käyttämällä syötteenä esimerkiksi painoa ja pituutta. Opetusdatassa mallille kerrotaan oikea sukupuoli ja sen jälkeen voidaan antaa mallille lisää dataa ja malli luokittelee opetusdatan perusteella henkilön sukupuolen uudesta datasta. (Nelli 2023b)

Yleisimpiä luokittelumenetelmiä ovat muun muassa päätöspuut (kuvio 8), satunnaismetsät (eng. random forests) sekä k-lähin naapuri (k-Nearest Neighbour). KNN luokittelee tietyn datapisteen sitä lähimpinä olevilla datapisteillä (naapureilla) k:n ollessa haluttu tarkasteltavien naapurien määrä. (Nelli 2023b.) Satunnaismetsät voivat olla myös regressiomenetelmiä, jos data-aineisto on numeerista. Regressiomenetelmiä ovat lisäksi esimerkiksi lineaarinen- (kuvio 9), polynominen- (usean muuttujan) tai logistinen- eli binäärinen (joko-tai) regressio (Han ym. 2012, 636).

3.2.2 Ohjaamaton oppiminen

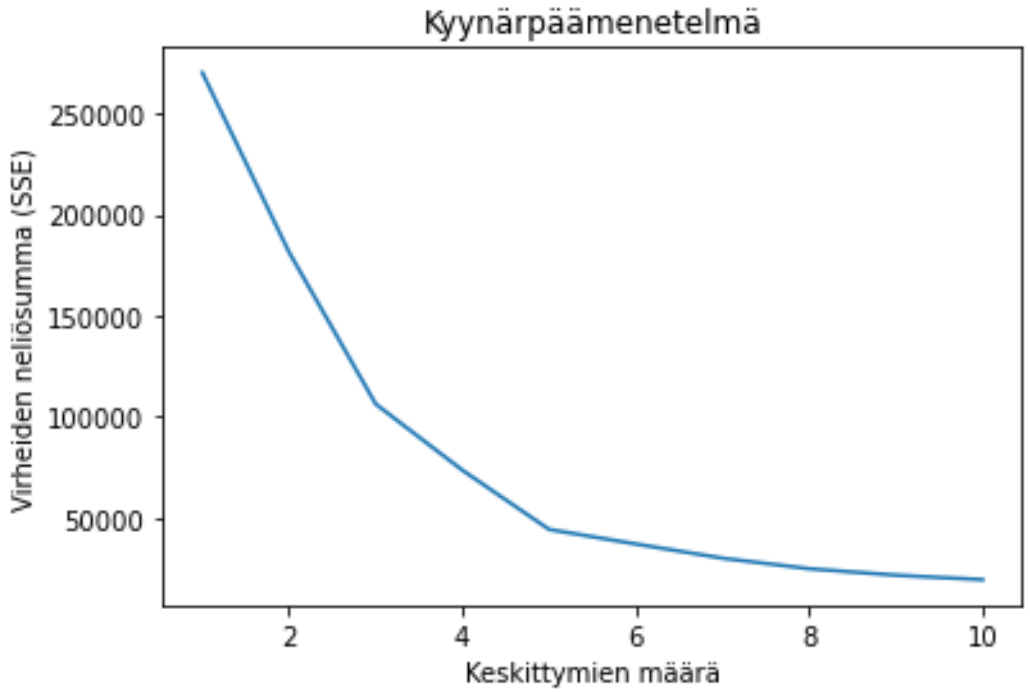
Ohjaamattoman oppimisen ero ohjattuun oppimiseen on ennaltamäärätyn tuloksen puuttuminen. Oppimismallin tavoitteena on ei-merkittyjen syötteiden uudelleenorganisointi samankaltaisten objektien ryhmäksi (Pöyry 2024). Mallia voidaan verrata kaavaan (1), mutta muodossa

$$D = \{(x_i)\}_{i=1}^{N},$$

jossa tulos y_i on poistettu, jolloin analysoidaan pelkästään datan eri piirteitä x_i (Murphy 2012, 2).

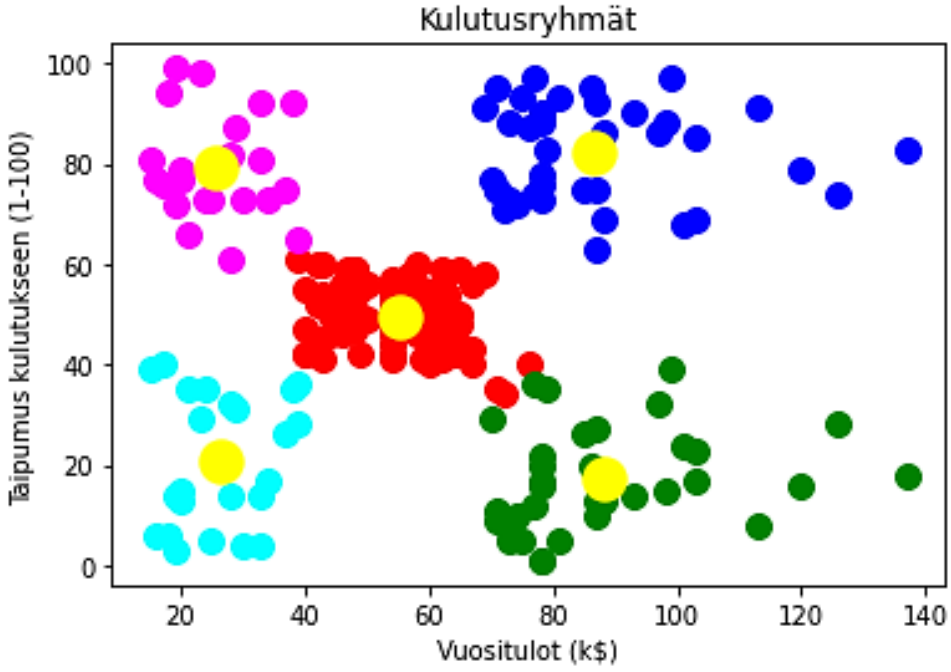
Klusterointi on yleisin ohjaamattoman oppimisen menetelmä, jossa kerätään datasta samanlaisia kohteita yhteen. Koska malli ei tiedä haluttua tulosta eikä sitä määritetä, klusteroinnin tarkoitus on selvittää tulos pelkkien syötteiden avulla. (Harrington 2012, 205.) Klusteroinnin esimerkkinä voidaan muun muassa käyttää *k-means*-klusterointia, jossa määritetään haluttu keskittymien määrä *k*, jonka jälkeen malli laskee kunkin klusterin sisällä olevista datapisteistä niiden välisen keskiarvon (*k-mean*) (Harrington 2012, 208). *K*:n optimaalisen määrän laskemiseen voidaan käyttää muun muassa kyynärpäämenetelmää (eng. elbow method), jossa kyseisessä datassa mitataan kunkin datapisteen etäisyys klusterin keskipisteestä ja lasketaan näiden etäisyyksien kokonaisvirheiden neliösumma (SSE). (Pöyry 2024 ja Harrington 2012, 214–216.)

Tekemässäni klusteroinnin esimerkissä (liite 2) on simuloituna henkilöitä sekä niiden vuositulot ja taipumus kulutukseen (1–100). Kuviossa 13 muodostetaan liitteen 2 datasta kyynerpäämenetelmällä klustereiden eli keskittymien määrä ja kuviossa 14 käytetään koneoppimisen menetelmänä *k-means*-klusterointia eri kulutusryhmien määrittämiseen datasta.



KUVIO 13. Esimerkki kyynärpäämenetelmästä (liite 2).

Kyynärpäämenetelmä saa tässä esimerkissä viisi potentiaalisinta keskittymää kuvion 13 mukaisesti, koska silloin SSE pienenee enää vain marginaalisesti (kyynärpäänivelen kohta kädessä). Kuviossa 14 on klusteroinnin tulokset viidellä eri kulutusryhmällä.



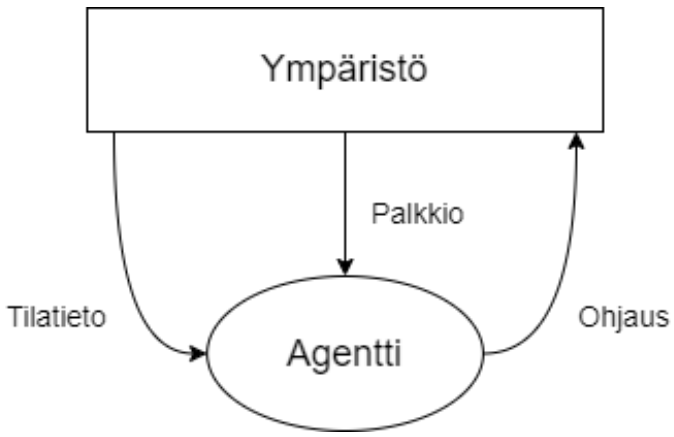
KUVIO 14. Esimerkki k-means-klusteroinnista (liite 2).

Tästä kyseisestä simuloidusta datasta saatuja tuloksia voidaan analysoida niin, että keskituloiset ihmiset ovat varsin säästeliäitä ja valveentuneita rahankäyttäjiä. Vähätuloiset sortuvat taasen usein liialliseen kulutukseen ja toisaalta osa suurituloisista kuluttaa huomattavasti vähemmän vuosituloihinsa suhteutettuna. Ohjaamaton oppiminen pyrkii siis löytämään vastaavanlaisesta datasta hyödyllistä tietoa ilman ennakkokäsitystä halutusta tuloksesta (tässä esimerkissä kulutusryhmistä).

3.2.3 Vahvistusoppiminen

Vahvistusoppiminen on laskennallinen lähestymistapa automatisoituun sekä tavoiteohjattuun (eng. goal-directed) oppimiseen ja päätöksentekoon (Sutton & Barto 2015, 13). Vahvistusoppimisessa oppiminen tapahtuu niin kutsutun agentin avulla ympäristössä, ja jonka toimintaa ohjaa palkitseminen (tai rangaistus) ongelman ratkaisemiseksi.

Oppimismalli toimii yritys-erehdys (eng. trial and error) periaatteella, jota voidaan verrata osaltaan ihmiselle tai eläimelle tyypilliseen oppimistapaan, missä oppimista ohjataan positiivisen tai negatiivisen palautteen perusteella. (Sutton & Barto 2015, 13.) Kuviossa 15 on kuvattuna vahvistusoppimisen toimintamalli Alpaydin (2020, 570) hahmotelman mukaisesti.



KUVIO 15. Agentin ja ympäristön vuorovaikutus (mukaillen Alpaydin 2020, 570).

Tilatieto on yksittäinen datapiste, jonka perusteella agentti ohjaa ja muuttaa ympäristön tilaa. Tästä muutoksesta agentti saa palautteen, jonka avulla se

suorittaa seuraavan ohjauksen pyrkien mahdollisimman isoon kumulatiiviseen palkkioon. (Alpaydin 2020, 570.)

Esimerkiksi robotin ohjaaminen sokkelon läpi tapahtuu vahvistusoppimisen mukaisesti niin, että agentti eli päätöksen tekijä on robotin ”aivot” ja ympäristö on sokkelo. Agentti liikuttaa robottia ja saa palautteen siitä onko edessä esteitä, jonka avulla robotti saadaan yritys- ja erehdysmenetelmällä tutkimaan sokkeloa niin kauan, kunnes se pääsee pois sokkelosta ja saa huomattavasti suuremman palkinnon. (Alpaydin 2020, 571.)

Vahvistusoppiminen tunnetaan muun muassa erilaisten sääntösidonnaisten pelien oppimisympäristönä ja sen avulla muun muassa mahdollistettiin tekoälyn kouluttaminen tekoälyllä (syvävahvistusoppiminen) (Hassabis 2021, 19). Vahvistusoppimista hyödynnetään nykyään muun muassa erilaisissa dynaamisessa sovelluksissa, kuten itseohjautuvien ajoneuvojen ja robottien suunnittelussa, mikä vaatii uudenlaista adaptiivisuutta tekoälyltä. (Brunton 2022, 424.)

4 DATAN HYÖDYNTÄMINEN SÄHKÖNKULUTUKSESSA

4.1 Datan keruu ja korreloivat tekijät

Sähkönkulutuksen seuranta sekä ennustaminen on tärkeää ja auttaa muun muassa sähköyhtiöitä optimoimaan järjestelmien suorituskykyä ja tuottavuutta. Lisäksi sähköyhtiöiden on hyvä tietää sähköntuotannon ja kulutuksen suhde kysynnän ja tarjonnan tasapainottamiseksi. (Salam & El Hibaoui 2018a.)

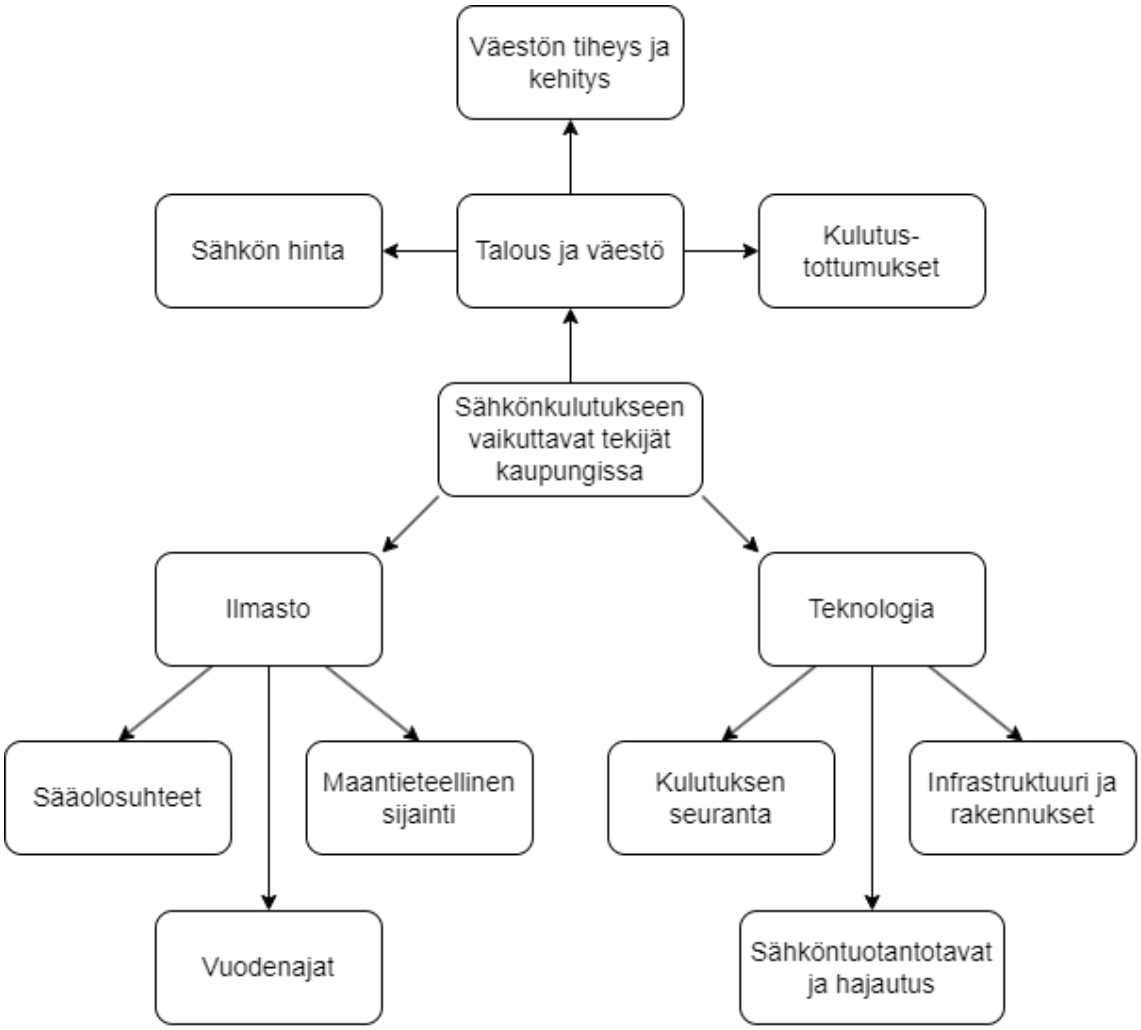
Datan hyödyntämiseksi tarvitsee kuitenkin tuottaa relevanttia raakadataa. Watkinsin (2016, 65) mukaan datan tuottamiseksi voidaan kysyä:

- Mitä voidaan mitata?
- Mitä kannattaa mitata?
- Miten se mitataan?
- Kuinka usein sitä kannattaa mitata?
- Mitä haasteita luotettavan datan mittaamiseen liittyy?

Verkot, jotka ohjaavat ja valvovat eri järjestelmiä, kuten sähkönjakelua tai sähköntuotantoa, kutsutaan OT-verkoiksi (Jurvanen 2025). Sähkönkulutuksen mittausdata on silloin niin sanottua OT- (eng. Operational Technology) dataa eli infrastrukuurin ja eri teollisuusprosessien valvontaan ja ohjaukseen tarkoitettua dataa. Se eroaa IT- (eng. Information Technology) datasta siten, että OT-dataa keräävät laitteet toimivat yleensä itsenäisesti ilman ihmistä, kun taas IT-laitteet, kuten tietokoneet eivät ole yleisesti käytettävissä ilman ihmiskontaktia. (Jurvanen 2025.)

OT-verkkoihin liittyy tietoa kerääviä järjestelmiä, kuten SCADA, joka valvoo ja kerää nimenomaan reaaliaikaista dataa teollisuuden prosesseista ja mittauksista. Datan keruun kannalta on tärkeää ymmärtää minkälaisessa ympäristössä dataa kerätään, ja mikä data on siinä ympäristössä mitattavissa sekä helposti saavutettavissa. (Jurvanen 2025.) Datan hyödyntämisen haasteita teollisuudessa käsitellään luvussa 4.3.

Sähkönkulutuksen seurannan ja ennakoinnin näkökulmasta tärkeitä asioita ovat muun muassa sähkönkulutusdatan aikajatkuvuus sekä pitkä aikajänne, jonka avulla huomataan helpommin toistuvuutta sekä yhteyksiä muihin tekijöihin (Salam & El Hibaoui 2018a). Kuvioon 16 on kerätty sähkönkulutukseen vaikuttavia tekijöitä kaupunkitasolla sekä Nguyenin ym. (2021) että Salamin ja El Hibaouin (2018a) tekemien tutkimuksen perusteella.



KUVIO 16. Sähkönkulutukseen vaikuttavat tekijät kaupungissa (mukaillen muun muassa Nguyen ym. 2021).

Sähkönkulutuksen seuraamisen jakeluverkosta lisäksi kannattaa siis mitata kulutuksen kanssa mahdollisesti korreloivia tekijöitä. Kuviossa 16 mainituista, tärkeitä muuttujia on esimerkiksi sääolosuhteet, väestönkehitys, sähkönhinta tai kulutustottumukset (eng. consumer behaviour). Näiden mainittujen tekijöiden vaihtelevuus kuitenkin tekee kulutuksen ennakoinnista vaikeampaa. Muun

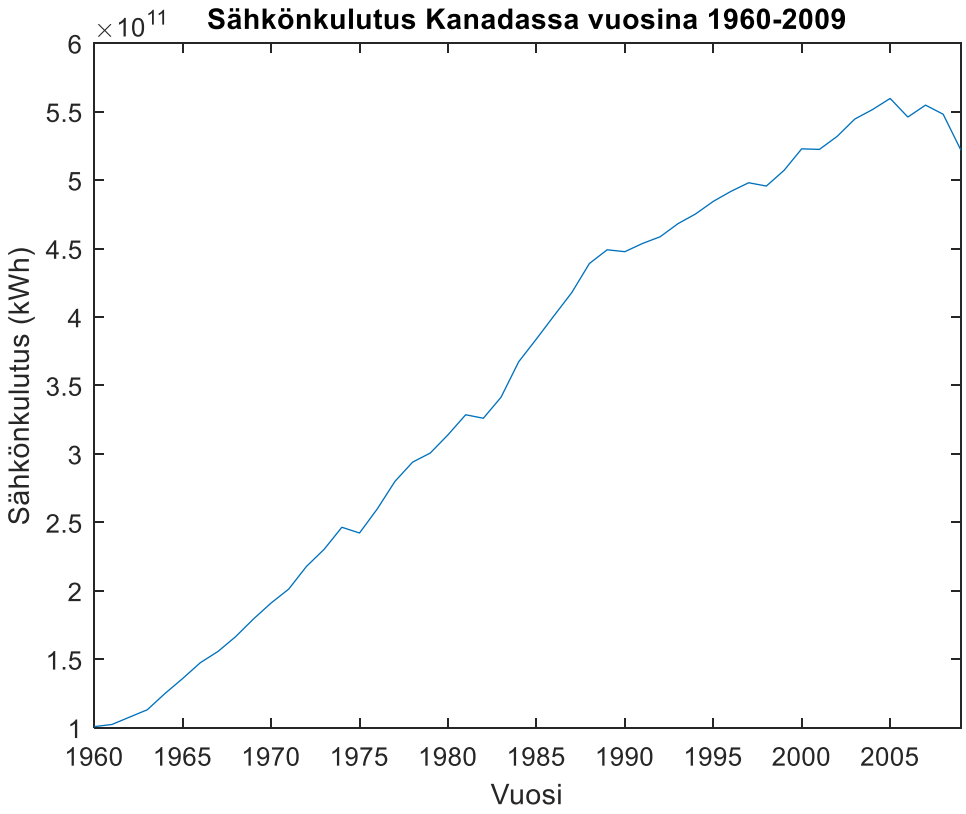
muassa mitattava data on usein aikasarjadataa ja vaatii usein pitkän ajanjakson ennen hyödyllisen tiedon tulemista esiin. (Salam & El Hibaoui 2018a.)

Kuvion 16 teknologiahaaraan ja sen alla olevaan kulutuksen seurantaan voidaan nykyään vahvasti liittää älykäs sähköverkko (eng. smart grid), jonka tarkoitus on tuottaa realistista tietoa sähkönkulutuksesta ja siten pyritään vastaamaan energiamurrokseen sekä muihin nykypäivän sähköverkojen haasteisiin (Pekkarinen 2024, 7). Älykkäässä sähköverkossa sähkö ja tieto kulkevat myös takaisin käyttäjiltä tuottajille, mikä antaa kuluttajalle aktiivisemman roolin sähkömarkkinoilla, parantaen muun muassa sähkönsiirron tehokkuutta sekä markkinoiden luotettavuutta (Hietamies 2022, 10).

4.2 Aikasarjadata

Aikasarjalla tarkoitetaan ajassa *t* mitattuja perättäisiä havaintoja (Palma 2016, 35). Aikasarjaa voidaan merkitä esimerkiksi $Y = Y_1, Y_2, \ldots, Y_n$, missä Y:n arvot ovat havaintoja (Dickey & Fuller 1979). Palman (2016, 35) mukaan havainnon tallennushetki *t* voi olla säännöllinen tai epäsäännöllinen, ja jatkuva tai diskreetti (kuvio 2). Jatkuva aikasarja on esimerkiksi lämpötilan mittaus ajan *t* suhteen, missä sen arvo voi olla käytännössä mikä tahansa lämpömittarin arvovälillä. Diskreetillä aikasarjalla voidaan viitata esimerkiksi kävijöiden määrään museossa yhden päivän aikana. Diskreettisyydellä viitataan myös ajan jaksollisuuteen kellonajan tapaan. (Palma 2016, 35.)

Airikan (2024a) mukaan monet teolliset valmistusprosessit ja automaatiojärjestelmät tuottavat aikasarjadataa, eli dataa, jonka mukana on aikaleima. Tämä pätee myös sähkönkulutukseen sekä monien siihen vaikuttavien tekijöiden mittaamiseen. Tämä data on yleensä säännöllistä eli tasavälistä, jatkuvaa dataa (Airikka 2024b). Aikaleima on datassa usein omana piirteenään ja esimerkiksi muodossa pp/kk/vv + tt:mm:ss eli tarkan vuoden, kuukauden, päivän sekä kellonajan kuvaus jokaisen datapisteen kohdalla (Airikka 2024a). Kuviossa 17 on tekemäni esimerkki aikasarjadatan mallinnuksesta, jossa on seurattu Kanadan sähkönkulutuksen kehitystä vuosina 1960–2009 (liite 3).



KUVIO 17. Esimerkki aikasarjadatasta (liite 3).

Pitkän aikavälin aikasarja-analyysia voidaan kutsua myös trendianalyysiksi tai trendiksi. Trendi tarkoittaa pitkän aikavälin muutosta (eng. long-term movement), jossa käy ilmi mihin suuntaan käyrä liikkuu ajan kuluessa. (Han ym. 2012, 588.) Aikasarjadatasta voi käydä ilmi esimerkiksi sykliset muutokset (eng. cyclic movements) eli signaalin tai piirteen luontainen oskillointi mukaillen esimerkiksi taloussuhdanteita (Han ym. 2012, 588).

Aikasarjadatasta voi käydä ilmi myös kausivaihtelut eli identtiset kuviot, jotka toistuvat tasaisin väliajoin, kuten esimerkiksi vuodenajat tai juhlapyhät. Lisäksi on niin sanottuja satunnaismuuttujia (eng. random movements), jotka johtuvat hyvin odottomattomista tapahtumista, kuten kuvion 17 äkillinen piikki vuonna 2008 tapahtuneesta globaalista talouskriisistä. (Han ym. 2012, 588.)

4.3 Haasteet datan hyödyntämisessä

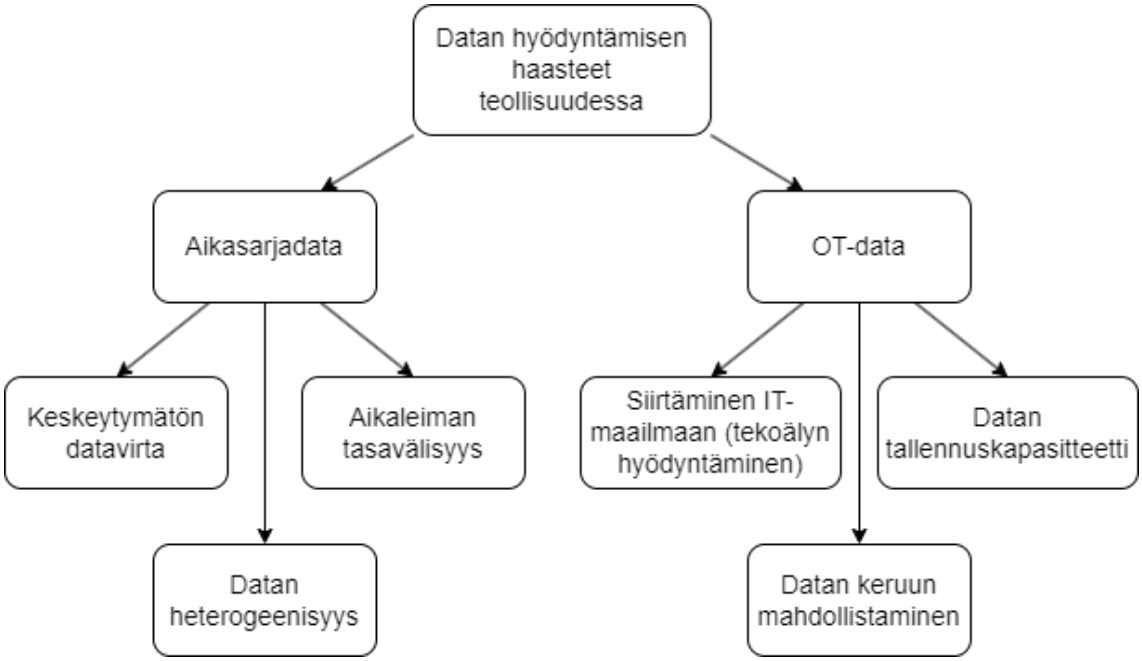
Teollisuudessa käytettyjä OT-järjestelmiä ei ole lähtökohtaisesti suunniteltu datan hyödyntämiseen (Joutsijoki 2024). Datan hyödyntämiseksi täytyy usein räätälöidä sovelluskohtaisesti se ympäristö, jossa datan keruu on ylipäätään mahdollista ja ratkaisut voivat olla usein kalliita ja aikaavieviä. Isot infrastruktuuriin kohdistuvat muutokset ovat myös vaikeita toteuttaa, jos kriittisiä prosesseja joudutaan keskeyttämään datan keruun mahdollistamisen takia. (Joutsijoki 2024.)

Datan keruun jälkeen haasteena on muun muassa saada OT-järjestelmien kuten SCADA:n tallentama tieto liikkumaan OT-maailmasta IT-maailmaan, kuten eri pilvialustoille (Joutsijoki 2024). Koska monet teollisuuden prosessit ovat keskeytymättömiä, uutta dataa tulee kellon ympäri. OT-järjestelmien datan tallennuskapasiteetti on usein rajoitettu, jolloin uusi dataa päällekirjoitetaan (eng. overwrite) vanhan datan päälle, jolloin sitä ei voida enää hyödyntää. Lisäksi OT-verkkojen tietoturva on IT-verkkoa alttiimpi haavoittuvuuksille, koska sen laitteet ovat vanhempia ja niiden tietoturvaa ei ole taloudellisesti niin kannattavaa päivittää prosessien keskeytymisen takia. (Jurvanen 2025.)

Aikasarjadata voidaan mieltää kompleksiseksi dataksi (Han ym. 2012, 585). Lisäksi aikasarjadatan kompleksisuutta lisää datan keruun keskeytymättömyys (eng. data stream), joka vaatii usein datan varastoinnilta erityistoimenpiteitä (Han ym. 2012, 598). Dataa voi olla useissa eri tietokannoissa ja se voi olla luonteeltaan heterogeenistä eli epäyhtenäistä, jolloin ne pitää integroida yhteen tietovarastoon datan louhintaa varten (kuvio 3) (Han ym. 2012, 126).

Aikasarjadatan hyödyntämisen kannalta aikaleiman tasavälisyys on vaatimus, jotta jokaisella datapisteellä on vertailukelpoinen arvo toisiinsa nähden. Tämän lisäksi aikasarjadataa voi joutua myös harventamaan tai interpoloimaan. (Airikka 2024b.) Datan harventamisessa datasta poistetaan ylimääräisiä näytteitä halutun intervallin saavuttamiseksi datapisteiden välillä. Aikasarjadatan interpoloinnissa kahden yksittäisen datapisteen väliin lisätään näytteitä jollakin sovitusmenetelmällä käyttäen hyödyksi ympäröiviä datapisteitä. (Airikka 2024b.) Kuviossa 18 ovat koottuna datan hyödyntämisen eri haasteita teollisuudessa,

kuten sähkönkulutuksen mittaamisessa, mukaillen Jurvasen (2025), Joutsijoen (2024) ja Airikan (2024b) aineistoja.



KUVIO 18. Datan hyödyntämisen haasteet teollisuudessa (mukaillen muun muassa Jurvanen 2025).

Joutsijoen (2024) mukaan tekoälyn ja koneoppimisen hyödyntämisen OT-maailmassa on huomioitava datan keräämiseen kuluva aika, joka on monesti pitempi kuin IT-maailmassa. Lisäksi datan siivoamiseen kuluu yleensä enemmän aikaa, minkä takia on tärkeää ymmärtää kohde, ympäristö ja mittalaitteet sekä mitä datalla halutaan saavuttaa ennen prosessin aloittamista. Koska aikasarjadatan keruu ja louhinta voi olla pitkäkin prosessi, koneoppimismallin rakentamista ja hienosäätöä ei voi jatkaa ikuisesti. Vaikka malli teoriassa voisi parantuakin uuden datan myötä, käytettävissä oleva aika ja resurssit ovat todellisuudessa kuitenkin rajallisia. (Joutsijoki 2024.)

5 DATAN KÄSITTELYPROSESSI

5.1 Datan hankinta ja tausta

Käsiteltäväksi dataksi valikoitui Tétouanin kaupungin sähkönkulutus (Salam & El Hibaoui 2018b). Data-aineisto on peräisin Californian yliopiston, Irvinen kampuksen (UCI) tekemästä tietokannasta, jossa säilytetään koneoppimiseen ja datan analysointiin tarkoitettua oppimisdataa (Kelly, Longjohn & Nottingham 2023).

Data-aineistossa seurataan vuoden 2017 ajan Tétouanin kaupungin sähkönkulutusta (eng. power consumption) sekä eri ilmastoon liittyviä mittauksia, kuten lämpötilaa ja ilmankosteutta. Tétouanin kaupunki sijaitsee Pohjois-Marokossa, Välimeren läheisyydessä. (Salam & El Hibaoui 2018a.) Ilmastoon liittyviä mittauksia on kerätty eri lähteistä ja sensoreista, muun muassa kaupungin lentokentältä sekä keskustasta viiden minuutin välein, jonka jälkeen näiden keskiarvot näytteistettiin uudelleen (eng. resample) kymmenen minuutin välein. Data on csv-muodossa talletettua tasavälistä aikasarjadataa, jonka mittausväli on kaikilla muuttujilla kymmenen minuuttia. (Salam & El Hibaoui 2018a.)

Sähkönkulutusta mitattiin kaupungin kolmesta eri sähkönjakeluasemasta (eng. substation): Quadsista, Smirista ja Boussafousta, jotka jakavat sähköä eri puolille kaupunkia. Sähkö on ensin muunnettu 64 kV suurjännitteestä 20 kV keskijännitteeksi, jotta sitä voidaan käyttää raskaassa teollisuudessa ja infrastuktuurissa, sekä edelleen jakaa pienjännitteeksi esimerkiksi asuinalueita varten. Sähkönkulutuksen mittaukset ovat peräisin Amendis-nimisen yrityksen SCADA-järjestelmästä, mikä vastaa muun muassa juomaveden- ja sähkönjakelusta Pohjois-Marokon alueella. (Salam & El Hibaoui 2018a.)

Datan käsittelyyn ja analysointiin käytetään MATLAB 2024b versiota sekä sen lisäosaa, Statistics and Machine Learning Toolboxia. Liitteessä 4 on tehty koodi MATLAB-livescriptinä, jonka lähteenä on käytetty suurimmaksi osaksi MATLABin tekijöiden eli MathWorksin tekemää dokumentaatiota, josta löytyy kattavat selitykset ja esimerkit jokaiselle funktiolle ja niiden eri parametreille.

5.1.1 Datan esikäsittely

Datan hankinnan jälkeen tutustutaan dataan ja valmistellaan sitä tarkempaa analysointia varten. Raakadataan oli jo kohdistettu paljon prosessointia, joten datan puhdistusta tai integrointia (kuvio 3) ei tarvinnut enää suorittaa. MATLAB-koodissa 1 luetaan data timetable-muuttujaan ja talletetaan koskematon data omaan muuttujaansa sekä haetaan aineiston perustiedot.

```
ogdata = readtimetable('Tetouan.csv','VariableNamingRule',"preserve");
df = ogdata; % käsiteltävä data

% Perustiedot datasta
summary(df)
```

MATLAB-KOODI 1. Datan lukeminen ja perustiedot.

MATLABin *readtimetable*-funktio on aikasarjadatalle suunniteltu muoto, ja sen avulla datan käsittely on huomattavasti käytännöllisempää verrattuna esimerkiksi matriisi- tai vektorimuotoon. Datan perustiedot saadaan yhdellä funktiolla *summary*, joka antaa muun muassa datan koon, 52416x8 eli 52 416 havaintoa (riviä) ja kahdeksan piirrettä (saraketta). Lisäksi komento antaa kaikki datan muuttujat, niiden datatyypin, puuttuvat arvot sekä tilastolliset tunnusluvut. Lisätään piirteenä kolmen lähdeaseman yhdistetty sähkönkulutus datan analysointia varten sekä muutetaan tuulen nopeus Suomessa enemmän käytettyyn muotoon (MATLAB-koodi 2).

```
% Kolmen jakeluaseman yhdistetty sähkönkulutus
df.TotalPowerConsumption = df.Quads + df.Smir + df.Boussafou;

% Tuulen nopeus km/h -> m/s
muuntokerroin = 1/3.6;
df.WindSpeed = df.WindSpeed * muuntokerroin;
```

MATLAB-KOODI 2. Muuttujien lisääminen ja transformointi.

Osana esikäsittelyä yhtenäistettiin myös muuttujien nimeämistä: Muun muassa jakeluasemat nimettiin niiden aiemmin mainituilla oikeilla nimillä, *zone 1*, *zone 2,* … -nimien sijaan. Muuttujien lisääminen sekä niiden transformointi on tärkeä osa

esikäsittelyä, koska siten dataa on helpompi ymmärtää ja hyödyntää. Data-aineiston alkuperäisiä muuttujia esitellään tarkemmin luvussa 5.1.2.

Koska havaintoja on vuoden ajalta, on tarpeellista tarkastella dataa myös kapeammin, kuten kuukausittain tai yhden vuorokauden aikana. Tätä varten luodaan apumuuttujia osaksi datajoukkoa käyttäen MATLABin *datetime*-oliota, joka erottaa eri aikakomponentteja datan aikaleimasta (The MathWorks, Inc. 2024a). MATLAB-koodissa 3 erotetaan data tunneittain, viikonpäivittäin, kuukauden ajan päivittäin, kuukausittain sekä kvartaaleittain.

```
% Apumuuttujien luominen eri ajanjaksojen tarkasteluun
df.Hour = hour(df.DateTime);
df.Weekday = weekday(df.DateTime);
df.Monthday = day(df.DateTime);
df.Month = month(df.DateTime);
df.Quarter = quarter(df.DateTime);
```

MATLAB-KOODI 3. Aikasarjadatan aikakomponentit.

Esimerkiksi tuntikomponentti tunnistaa *DateTime*-muuttujan formaatin perusteella aikaleiman tunnit ja indeksoi niitä 0–23 välillä eli vuorokausi kerrallaan. Jokainen tuntikomponentti käyttää silti jokaista mittauspistettä, koska yhdellä indeksillä on sama kymmenen minuutin intervalli (kuva 1).

DateTime	Hour
01/01/2017 00:00	0
01/01/2017 00:10	0
01/01/2017 00:20	0
01/01/2017 00:30	0
01/01/2017 00:40	0
01/01/2017 00:50	0
01/01/2017 01:00	1
01/01/2017 01:10	1

KUVA 1. Aikasarjadatan tuntikomponentti.

Yhdellä tuntikomponentin indeksillä on siis yhteensä kuusi mittausta. Eri ajanjaksot auttavat muodostamaan kokonaisvaltaisemman kuvan

aikasarjadatasta, mikä vastaavasti auttaa datan analysoinnissa ja johtopäätelmien muodostamisessa. Muuttujien käyttäytymistä eri aikaikkunoissa esitellään tarkemmin luvussa 5.2.

5.1.2 Datan yleiskuva

Esikäsittelyn jälkeen voidaan muodostaa hyvä peruskäsitys data-aineistosta ja sen eri piirteistä sekä alustavasti miettiä, mitkä niistä ovat potentiaalisimpia ja tärkeimpiä hyödyntämiseen koneoppimisessa. Kuvassa 2 on ote esikäsitellyn datan perusrakenteesta, jossa on nähtävillä puolet kaikista piirteistä.

DateTime	Temperature	Humidity	WindSpeed	GeneralDiffuseFlows
01/01/2017 00:00	6.559	73.8	0.023056	0.051
01/01/2017 00:10	6.414	74.5	0.023056	0.07
01/01/2017 00:20	6.313	74.5	0.022222	0.062
01/01/2017 00:30	6.121	75	0.023056	0.091
01/01/2017 00:40	5.921	75.7	0.0225	0.048
01/01/2017 00:50	5.853	76.9	0.0225	0.059
01/01/2017 01:00	5.641	77.7	0.022222	0.048
01/01/2017 01:10	5.496	78.2	0.023611	0.055

KUVA 2. Ote esikäsitellystä datasta.

Timetable-taulukossa piirteet ovat ylimpänä ja niitä ei Matblabissa lasketa riveiksi. Rivin muodostaa siis jokaisen datan piirteen arvo samalla ajanhetkellä. *DateTime*-muuttuja eli aikaleima muodostaa taulukon rungon ensimmäisenä piirteenä ja muita piirteitä luetaan sen mukaisesti.

Datan yleiskuvan määrittämiseksi on vielä tarpeellista tutkia data-aineiston eri piirteitä tarkemmin. Kerätään kaikkien mitattujen piirteiden nimet, yksiköt sekä tilastolliset tunnusluvut taulukkoon 1. Datassa on esikäsittelyn myötä myös lisättyjä piirteitä, kuten kokonaissähkönkulutus ja eri aikakomponentit, mutta nämä eivät ole alkuperäisiä mitattuja piirteitä. Tilastolliset tunnusluvut sai tulostettua jo aiemmin mainitulla MATLABin *summary*-funktiolla.

TAULUKKO 1. Mitattujen piirteiden tilastolliset tunnusluvut.

Piirre	Yksikkö	Min	Mediaani	Max	Keskiarvo	Std
Lämpötila	°C	3,25	18,78	40,01	18,81	5,82
Ilmankosteus	%	11,34	69,86	94,80	68,26	15,55
Tuulen nopeus	m/s	0,01	0,02	1,80	0,54	0,65
Yl. Diffuusio	-	0,00	5,04	1163	182,70	264,40
Diffuusio	-	0,01	4,46	936	75,03	124,21
Quads	kWh	13 896	32 266	52 204	32 345	7 131
Smir	kWh	8 560	20 823	37 409	21 043	5 202
Boussafou	kWh	5 935	16 415	47 598	17 835	6 622

Taulukosta huomataan, että tuulen nopeus on odotettua selvästi alhaisempi ottaen huomioon kaupungin maantieteellisen sijainnin, jolloin datan keruussa tai integroinnissa on pitänyt tapahtua jokin virhe. Diffuusioita ei selitetty tarkemmin aineistossa eikä niille myöskään annettu yksikköä, joten näiden piirteiden tarkoitusta data-aineistossa ei tiedetä.

Taulukon mediaani- ja keskiarvot ovat suhteellisen lähellä toisiaan, sekä min- ja max-arvot ovat järkevällä arvovälillä, eli suuria poikkeamia tai anomalioita ei ole suoraan nähtävissä. Keskihajontaan vaikuttaa tässä tapauksessa vuodenajat, jolloin esimerkiksi sähkönkulutus ja ilmasto ovat erilaiset esimerkiksi kesällä ja talvella, jolloin keskiarvon ympärille syntyy luontaista hajontaa.

5.2 Datan analysointi ja visualisointi

Datan esikäsittelyn ja yleiskuvan muodostamisen jälkeen ryhdytään louhimaan data-aineistoa tarkemmin tiedon löytämiseksi. Tässä opinnäytetyössä analysoidaan taulukon 1 muuttujista tarkemmin eri sähkönkulutusasemia sekä lämpötilaa ja ilmankosteutta, koska niillä voisi olla havaittavissa riippuvuuksia toisiinsa nähden.

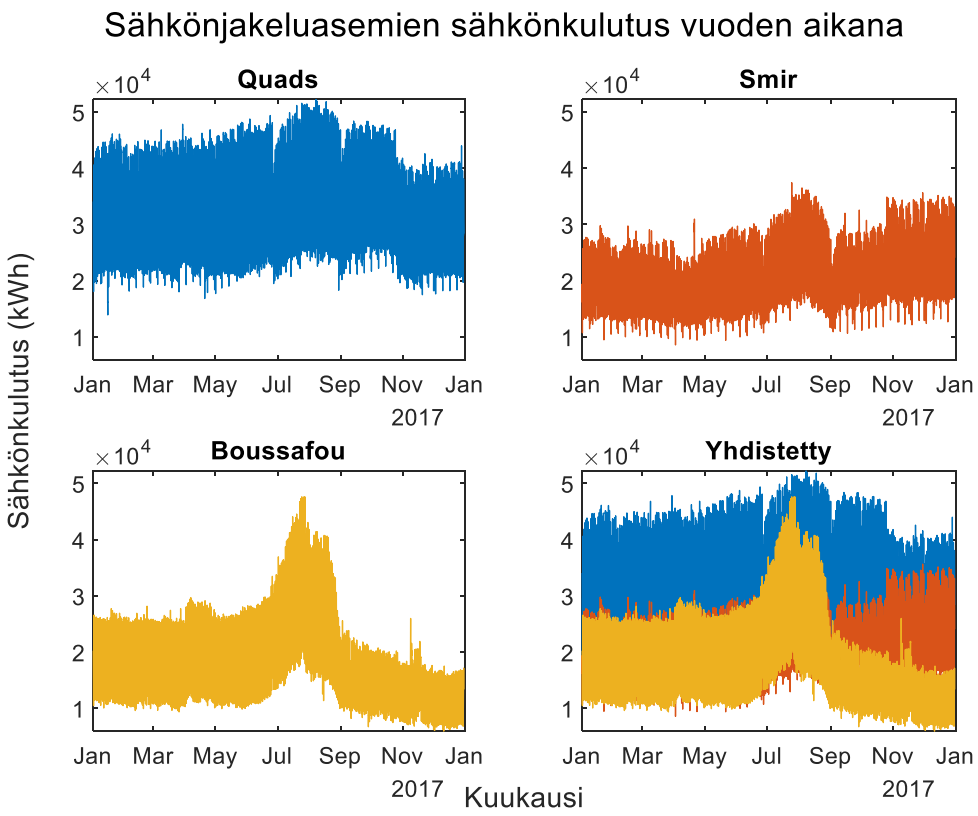
Analyysissä hyödynnetään myös luotuja aikakomponentteja (MATLAB-koodi 3), jotka auttavat paremmin tunnistamaan toistuvia malleja sekä uusia havaintoja.

Näiden aikakomponenttien avulla voidaan tarkastella ilmiöitä tarkemmin kuin pelkästään koko vuoden havaintoja analysoimista kerralla. Eri aikakomponenttien analysointi liittyy myös vahvasti visuaaliseen analytiikkaan, missä tarkentamalla aikaikkunaa luodaan uudenlaisia havainnollisia kuvioita, mutta käyttämällä silti samaa dataa.

Taulukosta 1 nähdään, että kaikki data on pelkästään numeerista ja diskreettiä mittausdataa, eli yhden piirteen arvoja ei kannata laskea yhteen (kuvio 2). Tämä rajaa jo huomattavasti käytettävissä olevia analysointimenetelmiä (kuvio 7). Aikasarjadatan analysoinnissa kuvailevat menetelmät eivät voi olla esimerkiksi piirakkakaavioita tai muita kategorisoivia menetelmiä vaan enemmän numeerisia arvoja kuvaavia, kuten laatikkokuvaajia tai histogrammeja, jotka esittelevät arvojen jakautumista tietylle mittausvälille. Kuvion 7 ennakoivista menetelmistä potentiaalisin juuri tälle data-aineistolle on lähtökohtaisesti regressioanalyysi, koska se on suurelle määrälle numeerista dataa yleensä sopiva menetelmä. Opinnäytetyön viimeisessä luvussa 5.3.3 data-aineistolle sovelletaan kyseistä ennakoivaa menetelmää osana ohjattua oppimista.

5.2.1 Sähkönkulutus

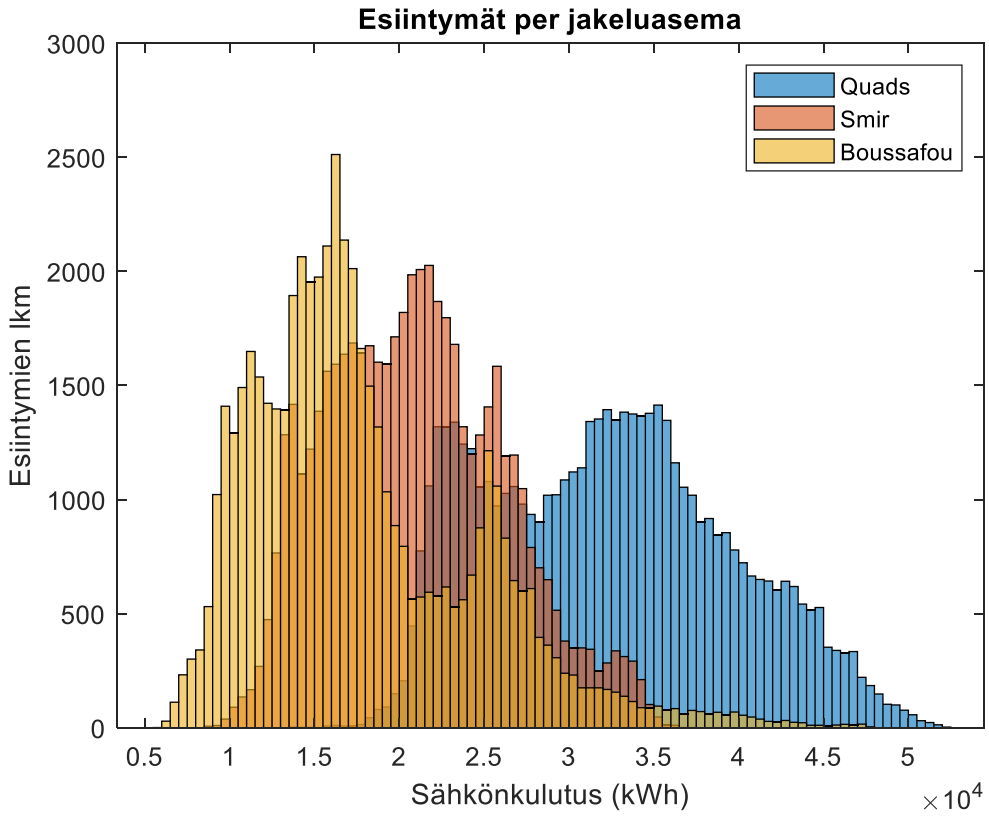
Sähkönkulutusta on hyvä analysoida sekä jakeluasemakohtaisesti että yhdistettynä sähkönkulutuksena. Lisäksi on hyvä käyttää eri visualisointimenetelmiä, jotta analysointi ei olisi liian yksiulotteista, sekä hyödyntää luotuja aikakomponentteja uusien havaintojen löytämiseksi. Havainnollistetaan ensimmäiseksi jokaisen jakeluaseman sähkönkulutus koko vuoden ajalta (kuvio 19).



KUVIO 19. Sähkönjakeluasemien sähkönkulutus vuoden aikana.

Quadsin sähkönkulutus on yleisesti kaikkein korkeinta ja se lisäksi nousee kesällä ja laskee talvella. Kesän korkeammat lämpötilat, turismi ja maatalous ovat osasyitä tähän ilmiöön (Salam & El Hibaoui 2018a). Smirin ja Boussafoun sähkönkulutukset ovat kesään asti identtisiä, mutta Smirin sähkönkulutus nousee talvella, kun taas Boussafoun nousee vahvemmin kesällä ja laskee talvella.

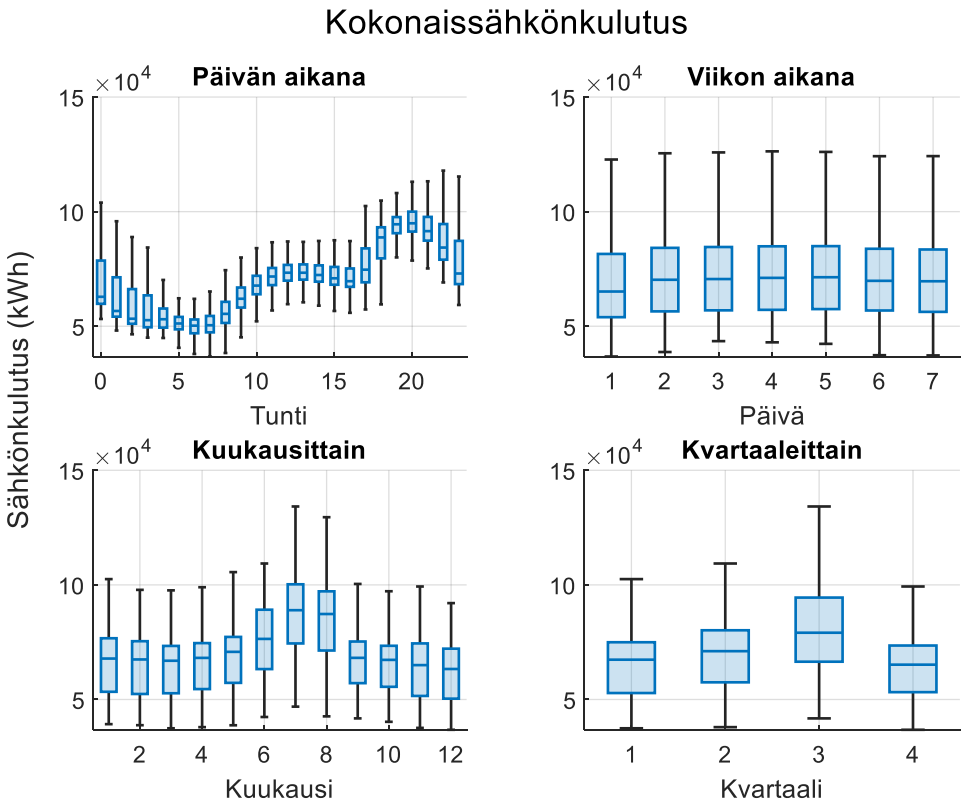
Sähkönjakeluasemien maantieteellisiä sijainteja ei kerrottu, mutta vaikuttaa siltä, että asemat ovat eri tarkoituksiin luotuja ja jakavat sähköä keskenään erilaisiin ympäristöihin. Kuviossa 20 on histogrammi kulutusesiintymistä eri jakeluasemilla. Histogrammin avulla havainnollistetaan, mitkä sähkönkulutuksen arvot ovat kaikkein yleisimpiä.



KUVIO 20. Histogrammi jakeluasemien kulutuksesta.

Quadsin sähkönkulutus on jakautunut tasaisimmin, kun taas Boussafoun mittaama kulutus on keskittyneintä. Sähkönkulutuksen ennakoinnin näkökulmasta Quads voisi olla potentiaalisin, koska sen kokonaiskulutus on suurinta (kuvio 19), mutta arvovälin hajonta on silti tasaisinta (kuvio 20).

Jos käyttää luotua kokonaissähkönkulutusta sekä eri aikakomponentteja, voidaan tutkia koko kaupungin sähkönkulutusta sekä yleistä kulutusta tarkemmin. Kuviossa 21 ovat koottuna kokonaissähkönkulutus päivän aikana, viikon aikana, kuukausittain sekä kvartaaleittain. Kvartaali ei juurikaan eroa kuukausittaisesta mittauksesta, mutta se kuvaa vuodenaikojen vaikutusta sähkönkulutukseen paremmin.



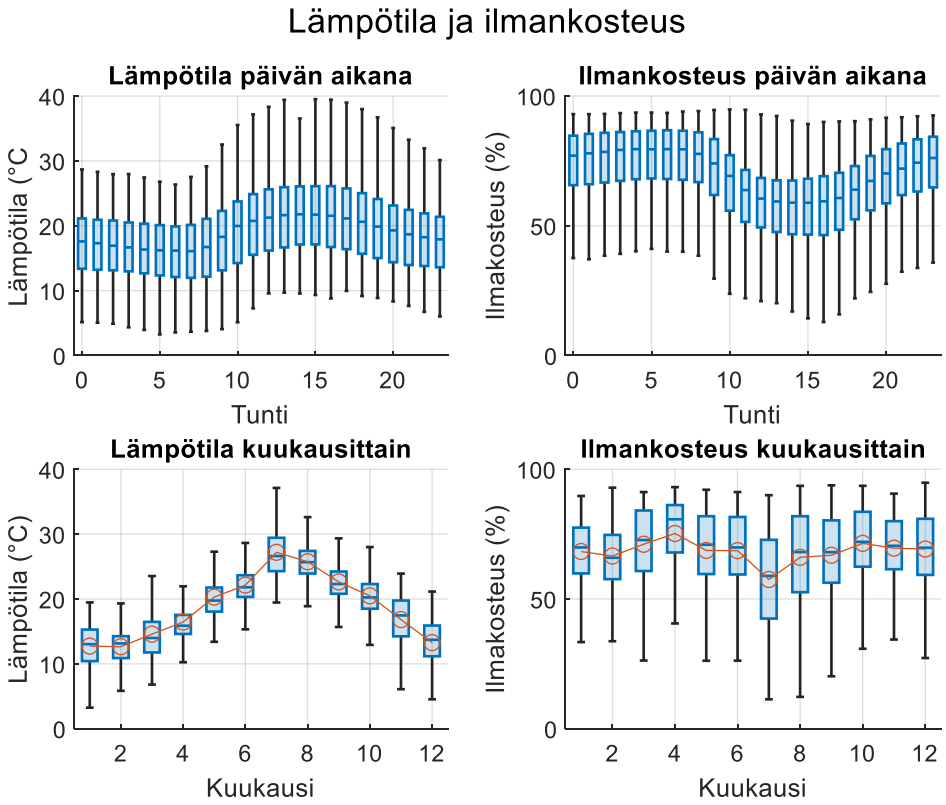
KUVIO 21. Kokonaissähkönkulutus.

Kuvaajien havainnollistamisessa on käytetty MATLABin *boxchart*-funktiota eli laatikkokuvaajaa, jonka avulla pystytään visualisoimaan tilastollisia tunnuslukuja. Laatikkokuvaajan sininen vaakasuuntainen viiva on kunkin näytevälin mediaani eli keskimmäinen arvo. Laatikon pinta-ala kuvaa sitä aluetta, jossa puolet arvoista sijaitsee, ja sen ylä- ja alareuna kuvaavat näytevälin kvartaaleita eli vastaavasti 75 % rajaa ja 25 % rajaa. Laatikon ulkopuolella olevat ”viikset” ovat min- ja max-arvoja kullakin näytevälillä. (The MathWorks, Inc. 2024b.)

Päivän aikana-kuviosta huomataan toistuva malli, jossa ensimmäiset tunnit ovat alhaisimpia, minkä jälkeen kulutus lähtee nousuun kulutushuipun ollessa illalla. Viikon aikana, jossa päivä 1 on sunnuntai ja 7 on lauantai, huomataan, että sunnuntaina sähkönkulutus on vähäisempää kuin muina päivinä. Kuvion alemmissa kuvaajissa huomataan sama kuvion 19 trendi, jossa sähkönkulutus kasvaa kesällä edellä mainituista syistä, kuten lämpötilasta, joka vaikuttaa muun muassa kaupungin turismin lisääntymiseen (Salam & El Hibaoui 2018a).

5.2.2 Lämpötila ja ilmankosteus

Tutkitaan samaan tyyliin lämpötilaa eri ajanjaksoilla, mutta koska lämpötila ei riipu viikonpäivistä ja kvartaalit eivät tuo uutta tietoa verrattuna kuukausittaiseen tarkasteluun, analysoidaan samassa kuviossa sen suhdetta ilmankosteuteen. Kuviossa 22 ovat lämpötila ja ilmankosteus päivän aikana sekä kuukausittain. Alemmissa kuvaajissa on lisättynä keskiarvojakauma näiden riippuvuuden korostamiseksi.



KUVIO 22. Lämpötila ja ilmankosteus eri ajanjaksoilla.

Verrattaessa lämpötilaa kuvioon 21, huomataan kuvaajissa jonkin verran samankaltaisuutta. Lämpötila lähtee päivän aikana nousuun, mutta laskee jo illalla toisin kuin kokonaissähkönkulutus. Kuukausittainen lämpötila on myös kesä-heinäkuun aikana korkeimmillaan, mutta nousu ja lasku on tasaisempaa kuin sähkönkulutuksessa.

Lämpötilan ja ilmankosteuden välillä huomataan selkeä käänteinen riippuvuus varsinkin päiväkohtaisesti, jossa lämpötilan noustessa ilmankosteus laskee. Tämä vaikuttaa lievästi myös sähkönkulutuksen ja ilmankosteuden

riippuvuuteen, vaikkakin riippuvuus on negatiivinen. Muuttujien riippuvuuksien määrittämistä tilastollisesti käsitellään luvussa 5.3.1.

5.3 Koneoppimisen hyödyntäminen

Opinnäytetyön ja datan käsittelyprosessin lopuksi tutkitaan, miten sähkönkulutusta voidaan ennakoida koneoppimisen avulla. Tämän opinnäytetyön tarkoituksena on esimerkkien avulla havainnollistaa koneoppimisen hyödyntämistä sähkönkulutuksen ennakoinnissa, eikä niinkään täydellisen mallin jalostamista lukuisten iterointien seurauksena.

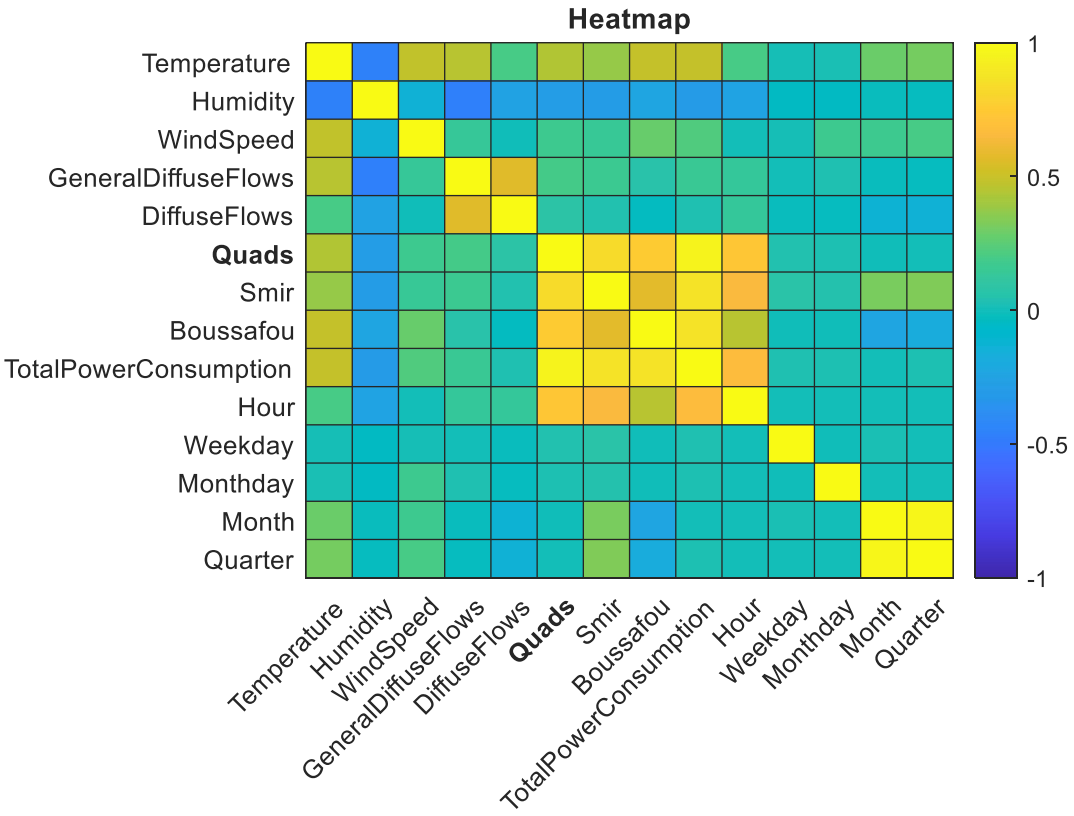
Koneoppimisen muodoksi valitaan ohjattu oppiminen (kuvio 11), jossa opetetaan malli olemassa olevan datan perusteella. Ennakoinnissa käytetään analysointimenetelmänä regressiota (kuvio 7), koska kuten aiemmin mainittu, käytettävissä oleva data on pelkästään numeerista mittausdataa. Tavoitteena on valita yksi sähkönkulutusasema, selvittää tämän kanssa korreloivat tekijät, sekä luoda niiden avulla malli, jolla jakeluaseman sähkönkulutusta pystytään ennustamaan jollain määritetyllä ajanjaksolla.

Regressiossa selitettäväksi muuttujaksi valitaan tässä työssä Quadsin sähkönjakeluasema, koska sen kulutus oli suurinta ja tasaisinta koko arvovälillä (kuvio 19 ja 20). Laajemmassa tutkimuksessa voisi esimerkiksi tutkia kaikkia kolmea jakeluasemaa ja eri vuodenaikoina, mutta tässä kohtaa valitaan vain yksi ja käytetään koko vuoden mittausdataa mallin luomisessa. Selittävät muuttujat määritetään korrelaatioanalyysin avulla seuraavassa luvussa.

Sähkönkulutuksen ennakoinnissa verrataan kahta regressiomenetelmää: lineaarista regressiota sekä satunnaismetsää, joka on kehittyneempi regressioanalyysin muoto. Menetelmät valikoituivat niiden yleisyyden takia numeerisen datan hyödyntämiseksi, sekä tarkoituksena on myös demonstroida eri mallien suorituskykyä, vaikka tulos ei olisikaan riittävällä tasolla.

5.3.1 Korrelaatioanalyysi

Sähkönkulutuksen ennakoinnin onnistumisen kannalta on oleellista tunnistaa ne muuttujat, jotka vaikuttavat selitettävään muuttujaan, joko positiivisesti tai negatiivisesti. Tämän takia luodaan kaikkien data-aineiston piirteistä korrelaatiomatriisi käyttämällä MATLABin *corr*-funktiota (Pearsonin lineaarinen korrelaatiokerroin). Funktio luo automaattisesti kaikkien muuttujien välille korrelaatiokertoimen välllä -1–1, missä 1 on täydellinen positiivinen korrelaatio ja -1 täydellinen negatiivinen korrelaatio (The MathWorks, Inc. 2024c). Kuviossa 23 on korrelaatiomatriisi visualisoituna heatmapiksi, jossa värit kertovat korrelaation vahvuuden piirteiden välillä numeeristen kertoimien sijaan.

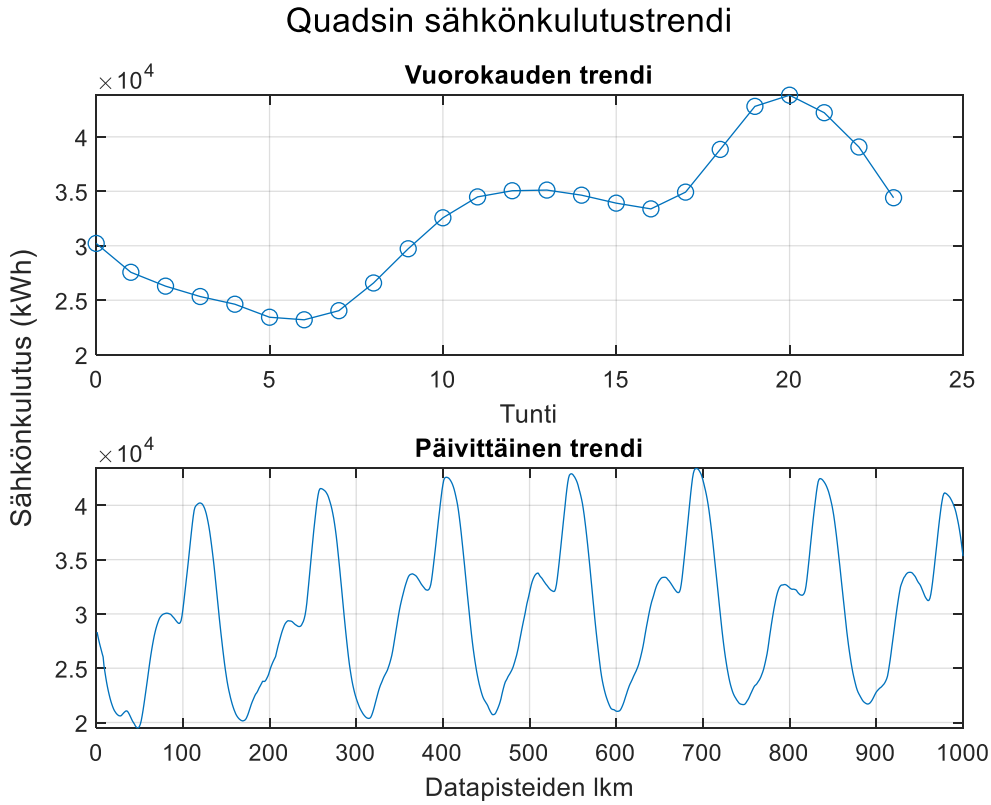


KUVIO 23. Korrelaatiomatriisi piirrettynä heatmappina.

Tummennettuna piirteenä on Quads eli selitettävä muuttuja. Korreloivia piirteitä ovat lämpötila, tosin vain noin 0,5 kertoimella toisin kuin Boussafoun kanssa, jonka kulutus nousi vahvemmin kesällä ja laski talvella. Muita korreloivia muuttujia ovat ilmankosteus (eng. humidity), jakeluasemat Smir ja Boussafou sekä aikaleiman tuntikomponentti. Kokonaissähkönkulutusta ei oteta huomioon, koska se koostuu osaltaan valmiiksi jo selitettävästä muuttujasta. Smiriä ja

Boussafouta voidaan käyttää, koska ne ovat keskenään erilaisia (kuvio 19) ja osa alkuperäistä dataa.

Tuntikomponentilla on siis vahva korrelaatio Quadsin sähkönkulutuksen kanssa tässä data-aineistossa. Vaikka tuntikomponentti ei ole mittausdataa, sitä voidaan silti käyttää ennakoinnissa, koska se käytännössä kuvaa miten kellonaika vaikuttaa sähkönkulutukseen. Havainnollistetaan Quadsin sähkönkulutustrendiä vielä sekä yhden päivän tasolla että päivittäisellä tasolla, paremman käsityksen saamiseksi ilmiölle (kuvio 24).



KUVIO 24. Quadsin sähkönkulutustrendi.

Ylemmässä kuvaajassa on esitetty Quadsin sähkönkulutuksen keskiarvo vuorokauden ajalta. Tuntikomponentti on käytännössä siis vain indeksi, joka laskee vuorokaudessa olevia tunteja (kuva 1). Koska sähkönkulutus on alhaisimmillaan päivän ensimmäisinä tunteina ja korkeimmillaan päivän viimeisinä tunteina, näiden kahden muuttujan välille syntyy luonnollinen korrelaatio.

Alemmassa kuvaajassa on vuoden ensimmäiset seitsemän päivää, joissa pääosin toistuu sama päiväkohtainen trendi pieniä vaihteluita lukuunottamatta. Kaksi ensimmäistä päivää ovat myös vuoden ensimmäiset päivät, joista oletettavasti ensimmäinen on uudenvuodenpäivä, joka vaikuttaa vähintään sunnuntain tapaan sähkönkulutuksessa (kuvio 21). Tällaisten syklisten mallien tai muiden kausivaihteluiden huomioiminen sähkönkulutuksen ennakoinnissa edellyttäisi luonnollisesti vähintään kahden vuoden mittaista datan keruuta.

5.3.2 Datan jakaminen opetus- ja testiaineistoon

Korrelaatioanalyysin jälkeen on tiedossa selitettävä muuttuja eli Quadsin sähkönjakeluasema, sekä selittävät muuttujat: lämpötila, ilmankosteus, jakeluasemat Smir ja Boussafou sekä ajan tuntikomponentti. Ennen ennakointimallin luomista jaetaan data opetus- ja testiaineistoon (kuvio 12) (MATLAB-koodi 4).

```
% Datan jakaminen selittäviin ja selitettävään muuttujaan
X = df{:,{'Temperature','Humidity','Smir','Boussafou','Hour'}};
y = df{:,'Quads'};

% Datan jakaminen 80/20
cv = cvpartition(height(df), 'HoldOut', 0.2); % randomoi datapisteet

% Datan jakaminen opetus- ja testiaineistoihin
X_train = X(training(cv),:);
y_train = y(training(cv),:);
X_test = X(test(cv),:);
y_test = y(test(cv),:);
```

MATLAB-KOODI 4. Datan jakaminen opetus- ja testidataan.

Aineistot nimetään yleensä *X*- ja *y*-matriiseiksi. *X*-matriisissa on datan aiemmin mainittujen selittävien piirteiden kaikki havainnot eli rivit. *Y*-matriisissa on vastaavasti kaikki selitettävän piirteen havainnot. MATLABin *cvpartition*-funktio ristiinvalidoi kaikki havainnot, jolloin ne valikoidaan sattumanvaraisesti datasta *HoldOut*-parametrin avulla, eivätkä ne enää silloin ole aikaleiman mukaisessa järjestyksessä (The MathWorks, Inc. 2024d).

Tässä työssä data jaetaan 80 % opetus- ja 20 % testiaineistoksi. Testiaineiston koko tällöin on yli 10 000, eli siinä on noin kahden ja puolen kuukauden verran randomoituja datapisteitä. Jaon jälkeen luodaan opetus- ja testiaineistomatriisit mallin luomista ja testaamista varten.

5.3.3 Sähkönkulutuksen ennakointi

Datan jakamisen opetus- ja testiaineistoksi jälkeen voidaan luoda koneoppimismallit sekä tehdä ennustukset sähkönkulutuksesta. Ennakointi tehdään koko mittaushistorian ajalta, mutta sitä visualisoidaan noin viikon ajalta, jotta ennustuksen tarkkuus tulee paremmin ilmi. MATLAB-koodissa 5 sovitetaan molemmat regressiomallit sekä luodaan ennustukset.

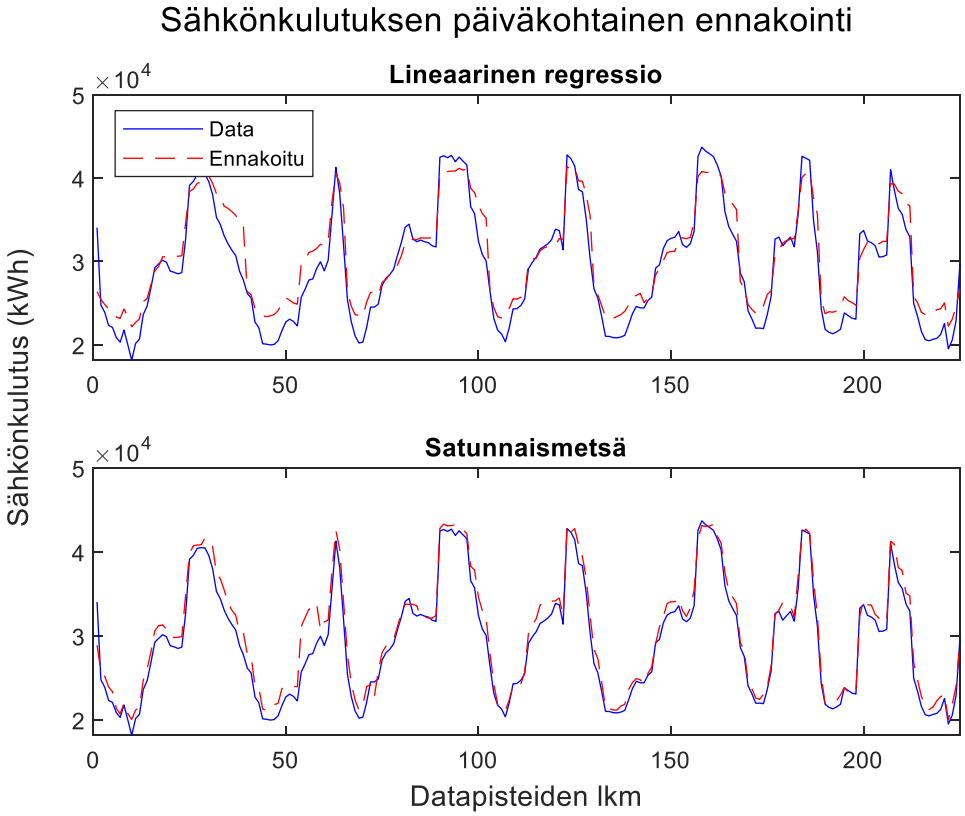
```
% Sovitetaan lineaarinen regressiomalli
mdllinear = fitlm(X_train,y_train);
y_pred = predict(mdllinear,X_test);

% Sovitetaan satunnaismetsämalli
mdlRF = fitrensemble(X_train,y_train,'Method','Bag');
y_pred = predict(mdlRF,X_test);
```

MATLAB-KOODI 5. Regressiomallien luominen ja ennustus.

Esimerkkinä käydään läpi satunnaismetsämallin luominen, joka on osa *ensemble*-algoritmia MATLABissa. Malli käyttää molempia opetusaineistoja sekä parametrinaan *bag*-metodia (eng. Bootstrap aggregation), joka kouluttaa sata päätöspuuta (kuvio 8) samanaikaisesti ja vie myös muutaman sekunnin prosessointiaikaa mallin luomiseksi. Päätöspuiden satunnaisista datapisteistä luomien arvioiden avulla funktio rakentaa mallin. (The MathWorks, Inc. 2024e.)

Ennusteen tekemisessä käytetään sekä luotua mallia että *X_test*-matriisimuuttujaa, jossa on testiaineisto selittävistä muuttujista. Näiden avulla voidaan pirtää kuvaajaan sekä alkuperäinen data *y_test* että ennustus *y_pred*. Kuviossa 25 ovat kuvattuna sekä lineaarisen regression että satunnaismetsän tuottama ennustus verratuna alkuperäiseen dataan Quadsin

sähkönkulutuksesta. Aikaväli on sama kuvion 24 kanssa eli vuoden ensimmäiset seitsemän päivää.



KUVIO 25. Sähkönkulutuksen päiväkohtainen ennakointi.

Kuviosta voidaan päätellä, että lineaarinen regressio ei kykene niin hyviin ennustamaan sähkönkulutuksen ääripäitä kuin satunnaismetsän tuottama malli. Kahden ensimmäisen päivän kulutus jää normaalia alemmas, eikä kummallakaan mallilla ole tarpeeksi dataa ennustamaan tämäntyyppistä kausivaihtelua, jota jo aiemminkin havaittiin. Satunnaismetsä onnistuu ennustamaan päiväkohtaista sähkönkulutusta yleisesti varsin hyvin. Kerätään vielä lasketut mallien tunnusluvut taulukkoon 2, joista ilmenee mallien tilastollinen suorituskyky.

TAULUKKO 2. Mallien tunnusluvut.

Tunnusluku	Lineaarinen regressio	Satunnaismetsä
R^2	0,8478	0,9762
RMSE (x10^3)	2,7859	1,0929

R^2-kerroin kuvaa mallin hyvyyttä selitettävän muuttujan arvioinnissa, missä arvo on sitä parempi, mitä lähempänä se on yhtä (1). Arvon ollessa 0,5 niin keskimäärin puolet havainnoista voidaan selittää mallin avulla. RMSE-arvo kuvaa mallin ennusteiden keskivirheiden neliöiden suhdetta alkuperäisiin arvoihin, jolloin sen arvo on parempi, mitä pienempi se on. (Pöyry 2024). Taulukosta voidaan todeta, että satunnaismetsän tuottaman mallin tunnusluvut ovat varsin hyviä ja sähkönkulutuksen ennakointi on luotettavalla tasolla tämän kyseisen data-aineiston kohdalla ja käytetyillä parametreilla.

6 POHDINTA

Tämä opinnäytetyö toimii johdatuksena koneoppimisen hyödyntämiseen oikean maailman teollisuusprosesseissa. Teoriaosuudessa määritetään datan käsittelyn eri menetelmät ja tekoälyn rooli nykypäivän datan käsittelyssä. Tominnallisessa osuudessa eli ohjelmakoodissa vastaavasti määritettiin sähkönkulutuksen kanssa korreloivat piirteet data-aineistosta ja ennakoitiin niiden avulla onnistuneesti sähkönkulutusta MATLABilla käyttämällä koneoppimisen muotona ohjattua oppimista.

Työn teoriaosuudessa käsiteltiin aihealueita varsin yleisellä tasolla, mutta aiheet ovat luonteeltaan yleishyödyllisiä ja antavat hyvän peruskuvan laajemmasta aihepiiristä. Työn aihepiiriä olisi voinut rajata tarkemmin teorian osalta, sillä sähkönkulutuksen ennakointi olisi voinut jo kattaa koko opinnäytetyön sisällön. Toisaalta laaja perustietämys ja termien ja taustojen ymmärtäminen aina tiedon tasoista data-analytiikkaan on työelämän ja jatko-opiskeluiden kannalta hyödyllistä, koska se helpottaa uuden tiedon omaksumista.

Työn toiminnallinen osuus eli ohjelmakoodi esitteli yleisellä tasolla datan käsittelyn ja analysoinnin prosessia. Esikäsittelyn osuus oli lyhyt, koska data-aineisto oli valmiiksi varsin prosessoitua. Datasta löydettiin toistuvia malleja ja rakenteita yhdistelmällä eri visuaalisia menetelmiä ja datan analysointimenetelmiä, joista suurin osa tosin oli odotettuja ilmiöitä. Data-aineistossa oli kaksi diffuusiomittausta, joiden tarkoitus osana aineistoa jäi epäselväksi, ja lisäksi tuulen nopeuden mittaukset olivat liian alhaisia ollakseen uskottavia, joten näitä piirteitä ei hyödynnetty tässä työssä.

Koneoppimismallien käsittelystä juuri MATLABilla minulla ei ollut kokemusta aiemmin, joten samojen perusteiden oppiminen uudessa ympäristössä opetti paljon. Toiminnallisen osuuden rakenteesta tuli myös loogisempi ja visuaalisesti yhtenäisempi, kun datan käsittelyssä käytti vain yhtä ympäristöä. Toisena haasteena oli aikasarjadatan käsittely, koska siitäkään minulla ei ollut juurikaan kokemusta. Vaatikin jonkin aikaa dokumentaation ja datan luonteen sisäistämistä, jotta käsittely alkoi sujumaan.

Aikasarjadatan ja yleisesti numeerisen datan luonteen ymmärtäminen datan luokittelun kontekstissa ja analysointimenetelmien valinnassa oli tärkeä hahmottaa etukäteen, jotta ei yrittänyt esimerkiksi luokitella numeerista dataa tai käyttää visualisointimenetelmiä, jotka eivät ole tarkoitettu juuri tämän tyyppiselle datalle. Tässä auttoi teoriaosuuden eri ajatuskartat ja kuviot, joihin viitattiin varsin paljon jälkeenpäin tekstissä. Tämä osaltaan kertoo, että suurin osa teorian aihealueista ainakin jollain tasolla kytkeytyvät toisiinsa.

Ohjelmakoodin pohjalta sähkönkulutuksen ennakointia voi jatkokehittää tarkempien koneoppimismallien luomiseksi ja sitä kautta parempien ennusteiden tuottamiseksi. Mahdollista on myös ennustaa muun muassa jotain tiettyä vuodenaikaa tai ajanjaksoa, jolloin saadaan enemmän hyödyllistä tietoa mittausajankohtana tapahtuneista ilmiöistä. Lopullinen TAMKille luovutettu ohjelmakoodi on myös pyritty rakentamaan ja kommentoimaan niin, että sen rakennetta ja käytettyjä menetelmiä pystyy soveltamaan laajemmin myös muihin teollisuuden aikasarjapohjaisiin data-aineistoihin.

Kokonaisuutena opinnäytetyö antaa hyvän pohjan datan analysoinnille MATLABilla sekä yleisesti koneoppimisen soveltamiselle teollisuudessa. Työn käsitellyistä aihealueista sekä eri datan käsittelyyn liittyvien ohjelmistojen käytöstä on hyötyä tulevaisuuden insinööritehtävissä, kun eri teollisuusprosessien ja automaatiojärjestelmien tuottamaa dataa pyritään hyödyntämään yhä enemmän yrityksissä.

LÄHTEET

Airikka, P. 2024a. Data- ja signaalianalyysi: Datan esikäsittely. Data- ja signaalianalyysi, sähkö- ja automaatiotekniikka. TAMK. Luentomateriaali. Viitattu 10.3.2025. https://moodle.tuni.fi/mod/resource/view.php?id=3160035

Airikka, P. 2024b. Data- ja signaalianalyysi: Aikasarjadata. Data- ja signaalianalyysi, sähkö- ja automaatiotekniikka. TAMK. Luentomateriaali. Viitattu 11.3.2025. https://moodle.tuni.fi/mod/resource/view.php?id=3160045

Alpaydin, E. 2020. Introduction to Machine Learning, Fourth Edition. E-kirja. 4. painos. Cambridge: MIT Press. Viitattu 7.3.2025. https://ebookcentral.proquest.com/lib/tampere/reader.action?docID=6676810&ppg=1

Annansingh, F & Bon Sesay, J. 2022. Data Analytics for Business: Foundations and Industry Applications. E-kirja. Oxford: Routledge. Viitattu 28.2.2025. https://doi.org/10.4324/9781003129356

Brunton, S. L. 2022. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. E-kirja. 2. painos. Cambridge: University Press. Viitattu 7.3.2025. https://doi.org/10.1017/9781009089517

Dickey, D & Fuller, W. 1979. Distribution of the Estimators for Autoregressive Time-Series with a Unit Root. Journal of the American Statistical Association, 74(366), 427–431. Viitattu 10.3.2025. https://doi.org/10.2307/2286348

Ertel, W. & Black, N. T. 2011. Introduction to Artificial Intelligence. E-kirja. London: Springer London, Limited. Viitattu 5.3.2025. https://doi-org.libproxy.tuni.fi/10.1007/978-0-85729-299-5

Han, J., Pei, J. & Kamber, M. 2012. Data mining: Concepts and Techniques. E-kirja. 2. painos. Morgan Kaufmann. Viitattu 28.2.2025. https://doi.org/10.1016/C2009-0-61819-5

Harrington, P. 2012. Machine Learning in Action. E-kirja. New York: Manning Publications Co. LLC. Viitattu 6.3.2025. https://learning.oreilly.com/library/view/machine-learning-in/9781617290183/?sso_link=yes&sso_link_from=tampere-university

Hassabis, D. 2021. DeepMind: From Games to Scientific Discovery. Research Technology Management, 64(6), 18–23. Viitattu 7.3.2025. https://doi.org/10.1080/08956308.2021.1972390

Helsingin yliopisto & MinnaLearn. Elements of AI. Päivitetty 2025. Verkkosivu. Viitattu 11.3.2025. https://www.elementsofai.com/

Hietamies, T. 2022. Energiamurroksen vaikutukset Elenian sähköverkossa. Sähkö- ja automaatiotekniikan tutkinto-ohjelma. Tampereen ammattikorkeakoulu. Opinnäytetyö. Viitattu 10.3.2025. https://urn.fi/URN:NBN:fi:amk-202204286332

Joutsijoki, H. 2024. Insta, edistynyt analytiikka ja automaatio. Insta Automation Oy. Vierailuluento. Koneoppiminen, sähkö- ja automaatiotekniikka. TAMK.

Jurvanen, L. Päivitetty 2025. Mikä on OT-verkko? Opas tuotantoverkkojen maailmaan! Savelan. Verkkosivu. Viitattu 12.3.2025. https://www.savelan.fi/mika-on-ot-verkko/#

Keim, D. A., Kohlhammer, J., Mansmann, F., May, T., & Wanner, F. 2010. Introduction. Mastering the Information Age. Solving Problems with Visual Analytics. Goslar: Eurographics Association. Viitattu 1.3.2025. https://www.researchgate.net/publication/277007765_Mastering_The_Information_Age_-_Solving_Problems_with_Visual_Analytics

Kelly, M., Longjohn, R. & Nottingham, K. 2023. The UCI Machine Learning Repository. Verkkosivu. Viitattu 17.3.2025. https://archive.ics.uci.edu

Kidd, C. & Hornay, R. 2021. Data Analytics vs Data Analysis: What's The Difference? Verkkosivu. Viitattu 27.2.2025. https://www.bmc.com/blogs/data-analytics-vs-data-analysis/

Laihonen, H., Hannula, M., Helander, N., Ilvonen, I., Jussila, J., Kukko, M., Kärkkäinen, H., Lönnqvist, A., Myllärniemi, J., Pekkola, S., Virtanen, P., Vuori, V., Yliniemi, T., & Tietojohtamisen tutkimuskeskus Novi. 2013. Tietojohtaminen. Tampereen teknillinen yliopisto - Tiedonhallinnan ja logistiikan laitos. Viitattu 2.3.2025. https://urn.fi/URN:ISBN:978-952-15-3058-6

Liebowitz, J. 2021. Data Analytics and AI. E-kirja. Auerbach Publications. Viitattu 1.3.2025. https://doi.org/10.1201/9781003019855

Murphy, K. P. 2012. Machine Learning: A Probabilistic Perspective. E-kirja. Cambridge: MIT Press. Viitattu 5.3.2025. https://ebookcentral.proquest.com/lib/tampere/detail.action?docID=3339490

Nelli, F. 2023a. An Introduction to Data Analysis. In Python Data Analytics. E-kirja. Apress L. P. Viitattu 27.2.2025. https://learning.oreilly.com/library/view/python-data-analytics/9781484239131/

Nelli, F. 2023b. Deep Learning with TensorFlow. In Python Data Analytics. E-kirja. Apress L. P. Viitattu 5.3.2025. https://learning.oreilly.com/library/view/python-data-analytics/9781484239131/?sso_link=yes&sso_link_from=tampere-university

Nguyen, C. T., Nguyen D. T. H., & Phan, D. K. 2021. Factors affecting urban electricity consumption: A case study in the bangkok metropolitan area using an integrated approach of earth observation data and data analysis. Environmental Science and Pollution Research, 28(10), 12056–12066. Viitattu 10.3.2025. https://doi.org/10.1007/s11356-020-09157-6

Palma, W. 2016. Time series analysis. E-kirja. New York: Wiley. Viitattu 10.3.2025. https://ebookcentral.proquest.com/lib/tampere/reader.action?docID=7103907&ppg=35

Pekkarinen, K. 2024. Älykkäät sähköjärjestelmät. Sähkö- ja automaatiotekniikan tutkinto-ohjelma. Oulun ammattikorkeakoulu. Opinnäytetyö. Viitattu 10.3.2025. https://urn.fi/URN:NBN:fi:amk-202402142917

Pöyry, P. 2024. Data-analytiikka ja koneoppiminen. Luentomateriaali. Koneoppiminen, sähkö- ja automaatiotekniikka. TAMK. Viitattu 27.2.2025. https://moodle.tuni.fi/pluginfile.php/4149735/mod_resource/content/0/koneoppiminen.pdf

Salam, A & El Hibaoui, A. 2018a. Comparison of Machine Learning Algorithms for the Power Consumption Prediction: - Case Study of Tetouan city. 2018 6th International Renewable and Sustainable Energy Conference (IRSEC), 1–5. Viitattu 10.3.2025. https://doi.org/10.1109/IRSEC.2018.8703007

Salam, A. & El Hibaoui, A. 2018b. Power Consumption of Tetouan City. UCI Machine Learning Repository. Data-aineisto. Verkkosivu. Viitattu 17.3.2025. https://doi.org/10.24432/C5B034

Sutton, R. S & Barto, A. G. 2015. Reinforcement Learning: An Introduction. Second edition, in progress. University of Stanford. PDF. Viitattu 6.3.2025 https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRL-Book2ndEd.pdf

The MathWorks, Inc. 2024a. Datetime. Verkkosivu. Viitattu 17.3.2025. https://se.mathworks.com/help/matlab/ref/datetime.html

The MathWorks, Inc. 2024b. Boxchart. Verkkosivu. Viitattu 19.3.2025. https://se.mathworks.com/help/matlab/ref/boxchart.html

The MathWorks, Inc. 2024c. Corr. Verkkosivu. Viitattu 2.4.2025. https://se.mathworks.com/help/stats/corr.html

The MathWorks, Inc. 2024d. Cvpartition. Verkkosivu. Viitattu 21.3.2025. https://se.mathworks.com/help/stats/cvpartition.html

The MathWorks, Inc. 2024e. Fitrensemble. Verkkosivu. Viitattu 2.4.2025. https://se.mathworks.com/help/stats/fitrensemble.html

Watkins, J. 2016. An Introduction to the Science of Statistics: From Theory to Implementation. University of Arizona. PDF. Viitattu 27.2.2025. https://math.arizona.edu/~jwatkins/statbook.pdf

LIITTEET

Liite 1. Esimerkki regressioanalyysistä

Työkokemuksen vaikutus palkkatasoon. Spyder. https://www.kaggle.com/datasets/ravitejakotharu/salary-datacsv

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 11 13:47:38 2025

@author: Paavo
"""

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('salary.csv')

plt.scatter(df.YearsExperience, df.Salary)
plt.xlabel('Työkokemus')
plt.ylabel('Palkkataso')
plt.title('Regressioanalyysi')
plt.show()
```

Liite 2. Esimerkki k-means-klusteroinnista

Henkilöiden taipumus kulutukseen vuositulojen perusteella. Spyder.

https://www.kaggle.com/datasets/shrutimechlearn/customer-data

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 11 13:46:06 2025

@author: Paavo
"""
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('customers.csv')
X = df.iloc[:, [3, 4]].values

# Hyödynnetään kyynärpäämenetelmää
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    model = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    model.fit(X)
    wcss.append(model.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Kyynärpäämenetelmä')
plt.xlabel('Keskittymien määrä')
plt.ylabel('Virheiden neliösumma (SSE)')
plt.show()

# Mallin opettaminen
model = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = model.fit_predict(X)

# Klustereiden visualisointi
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100,
            c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100,
            c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100,
            c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100,
            c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100,
            c = 'magenta', label = 'Cluster 5')
plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:, 1],
            s = 300, c = 'yellow', label = 'Centroids')
plt.title('Kulutusryhmät')
plt.xlabel('Vuositulot (k$)')
plt.ylabel('Taipumus kulutukseen (1-100)')
plt.show()
```

Liite 3. Esimerkki aikasarjadatasta

Energian kulutus Kanadassa vuosina 1960–2009. MATLAB 2024b (Economics Toolbox). https://se.mathworks.com/help/econ/data-sets-and-examples.html

```
load Data_PowerConsumption

df = DataTable;
plot(df.Time, df.consump)

xlabel('Vuosi')
ylabel('Sähkönkulutus (kWh)')
title('Sähkönkulutus Kanadassa vuosina 1960-2009')
```

Liite 4. Ohjelmakoodi

CASE: Power Consumption of Tétuan City. Aikasarjadatan käsittely ja analysointi MATLABilla. MATLAB 2024b (Statistics and Machine Learning Toolbox).

CASE: Power Consumption of Tétouan City.

Aikasarjadatan käsittely ja analysointi MATLABilla. Paavo Yrtti

Datasetti: Power Consumption of Tetouan City - UCI Machine Learning Repository

Käytetty versio:

```
ver;
```

Alustus:

```
close all
clear
clc
```

Esikäsittely ja yleiskuva

Esikäsittely:

```
% Säilytetään alkuperäinen data
ogdata = readtimetable('Tetuan City power consumption.csv','VariableNamingRule',"preserve");
df = ogdata; % käsiteltävä data

% Perustiedot datasta
summary(df);

% Yhtenäistetään muuttujien nimeämiset
df.Properties.DimensionNames;
df.Properties.VariableNames;
oldvars = {'Wind Speed','general diffuse flows','diffuse flows','Zone 1 Power Consumption',...
           'Zone 2  Power Consumption','Zone 3  Power Consumption'};
newvars = {'WindSpeed','GeneralDiffuseFlows','DiffuseFlows','Quads','Smir','Boussafou'};
df = renamevars(df,oldvars,newvars);

% Lisätään kolmen jakeluaseman yhdistetty sähkönkulutus
df.TotalPowerConsumption = df.Quads + df.Smir + df.Boussafou;

% Tuulen nopeus km/h -> m/s
muuntokerroin = 1/3.6;
df.WindSpeed = df.WindSpeed * muuntokerroin;

% Luodaan apumuuttujat eri ajanjaksojen tarkasteluun:
df.Hour = hour(df.DateTime);
df.Weekday = weekday(df.DateTime);
df.Monthday = day(df.DateTime);
df.Month = month(df.DateTime);
df.Quarter = quarter(df.DateTime);
```

Yleiskuva:

```
% Ote esikäsitellystä datasetistä
disp(head(df));
summary(df);
```

(jatkuu)

Analysointi ja visualisointi

Sähkönkulutus:

Jakeluasemien sähkönkulutus vuoden aikana:

```
figure;

% Sama asteikko kaikille kuvioille
y_min = min([min(df.Quads),min(df.Smir),min(df.Boussafou)]);
y_max = max([max(df.Quads),max(df.Smir),max(df.Boussafou)]);

% Quads
subplot(2,2,1);
plot(df.DateTime,df.Quads);
ylim([y_min y_max]);
title('Quads');

% Smir
subplot(2,2,2);
plot(df.DateTime,df.Smir,'color',"#D95319");
ylim([y_min y_max]);
title('Smir');

% Boussafou
subplot(2,2,3);
plot(df.DateTime,df.Boussafou,'Color',"#EDB120");
ylim([y_min y_max]);
title('Boussafou');

% Yhdistetty
subplot(2,2,4);
plot(df.DateTime,df.Quads);
hold on;
plot(df.DateTime,df.Smir);
hold on;
plot(df.DateTime,df.Boussafou);
hold off;
ylim([y_min y_max]);
title('Yhdistetty');

% Yhteinen akseleiden nimeäminen
sgtitle('Sähkönjakeluasemien sähkönkulutus vuoden aikana');
han=axes('visible','off');
han.XLabel.Visible='on';
han.YLabel.Visible='on';
ylabel(han,'Sähkönkulutus (kWh)');
xlabel(han,'Kuukausi');
```

Histogrammi jakeluasemista:

```
figure;

histogram(df.Quads);
hold on;
histogram(df.Smir);
hold on;
histogram(df.Boussafou);
hold off;
legend('Quads','Smir','Boussafou');
title('Esiintymät per jakeluasema');
xlabel('Sähkönkulutus (kWh)');
ylabel('Esiintymien lkm');
```

Kokonaissähkönkulutus:

```
figure;

% Kokonaissähkönkulutus päivän aikana
subplot(2,2,1);
boxchart(df.Hour,df.TotalPowerConsumption,"Markerstyle","none");
title('Päivän aikana');
xlabel('Tunti');
grid on;

% Kokonaissähkönkulutus viikon aikana
subplot(2,2,2);
boxchart(df.Weekday,df.TotalPowerConsumption,"Markerstyle","none"); % päivä 1 on sunnuntai
title('Viikon aikana');
xlabel('Päivä');
grid on;
xticks(1:1:7);

% Kokonaissähkönkulutus kuukausittain
subplot(2,2,3);
boxchart(df.Month,df.TotalPowerConsumption);
title('Kuukausittain');
xlabel('Kuukausi');
grid on;

% Kokonaissähkönkulutus kvartaaleittain
subplot(2,2,4);
boxchart(df.Quarter,df.TotalPowerConsumption,"Markerstyle","none");
title('Kvartaaleittain');
xlabel('Kvartaali');
grid on;

sgtitle('Kokonaissähkönkulutus');
han=axes('visible','off');
han.YLabel.Visible='on';
ylabel(han,'Sähkönkulutus (kWh)');
```

Lämpötila ja ilmankosteus:

```
figure;

sgtitle('Lämpötila ja ilmankosteus')

% Lämpötila vuorokauden aikana + keskiarvojakauma
subplot(2,2,1)
boxchart(df.Hour,df.Temperature,"Markerstyle","none");
title('Lämpötila päivän aikana');
xlabel('Tunti');
ylabel('Lämpötila (°C')
grid on;

% Lämpötila kuukausittain + keskiarvojakauma
subplot(2,2,3)
meanTemp = groupsummary(df,'Month','mean','Temperature');
boxchart(df.Month,df.Temperature,"Markerstyle","none");
hold on;
plot(meanTemp.Month,meanTemp.mean_Temperature,'-o');
hold off;
title('Lämpötila kuukausittain');
xlabel('Kuukausi');
ylabel('Lämpötila (°C)');
grid on;

% Ilmankosteus vuorokauden aikana
subplot(2,2,2)
boxchart(df.Hour,df.Humidity,"MarkerStyle","none");
title('Ilmankosteus päivän aikana');
xlabel('Tunti');
ylabel('Ilmakosteus (%)');
grid on;

% Ilmankosteus kuukausittain + keskiarvojakauma
subplot(2,2,4)
meanHumidity = groupsummary(df,'Month','mean','Humidity');
boxchart(df.Month,df.Humidity,"MarkerStyle","none");
hold on;
plot(meanHumidity.Month,meanHumidity.mean_Humidity,'-o');
hold off;
title('Ilmankosteus kuukausittain');
xlabel('Kuukausi');
ylabel('Ilmakosteus (%)');
grid on;

% Histogrammi lämpötiloista
figure;
histogram(df.Temperature);
title('Lämpötilaesiintymät');
xlabel('Lämpötila (°C)');
ylabel('Esiintymien lkm');

% Histogrammi ilmankosteuksista
histogram(df.Humidity)
title('Ilmankosteusesiintymät');
xlabel('Ilmankosteus (%)');
ylabel('Esiintymien lkm');
```

Soveltaminen koneoppimiseen

Tehtävä: Quadsin sähkönkulutuksen ennustaminen päivätasolla

Korrelaatio:

```
figure;

% Luodaan korrelaatiomatriisi
X = df.Variables;
corrMatrix = corr(X);

vars = df.Properties.VariableNames;
vars = strrep(vars,'Quads','\bf Quads'); % boldataan tarkasteltava muuttuja

% Luodaan heatmap
h = heatmap(vars,vars,corrMatrix,'colormap',parula,'ColorLimits',[-1, 1]);
h.XDisplayLabels = vars;
h.YDisplayLabels = vars;
title("Heatmap");figure;
```

Tarkastellaan tuntikomponenttia tarkemmin:

```
hourlyConsumption = groupsummary(df,"Hour","mean","Quads"); % luodaan keskiarvomuuttuja

subplot(2,1,1);
plot(hourlyConsumption.Hour,hourlyConsumption.mean_Quads,'-o');
title('Vuorokauden trendi');
xlabel('Tunti')
grid on;

subplot(2,1,2);
plot(movmean(df.Quads,15)) % liikkuva keskiarvo kuvaajan pehmentämiseen
title('Päivittäinen trendi');
xlabel('Datapisteiden lkm');
grid on;
xlim([0 1000]); % tuhat ensimmäistä datapistettä

sgtitle('Quadsin sähkönkulutustrendi');
han=axes('visible','off');
han.YLabel.Visible='on';
ylabel(han,'Sähkönkulutus (kWh)');
```

Datan jakaminen opetus- ja testidataan:

```
% Datan jakaminen opetus- ja testiaineistoon
X = df{:,{'Temperature','Humidity','Smir','Boussafou','Hour'}};
y = df{:,'Quads'};

% 80/20 opetusdataa, koska dataa on silti tarpeeksi testaamiseen
cv = cvpartition(height(df), 'HoldOut', 0.2); % randomoi datapisteet

X_train = X(training(cv),:);
y_train = y(training(cv),:);
X_test = X(test(cv),:);
y_test = y(test(cv),:);
```

Mallin luominen ja testaaminen:

Lineaarinen regressio:

```
% Sovitetaan lineaarinen regressiomalli
mdllinear = fitlm(X_train,y_train);
y_pred = predict(mdllinear,X_test);

figure;
subplot(2,1,1)
plot(y_test,'b');
hold on;
plot(y_pred,'r--');
hold off;
legend('Data','Ennakoitu','Location','northwest');
title('Lineaarinen regressio');
xlim([0 225]);
```

Satunnaismetsä:

```
% Luodaan satunnaismetsämalli
mdlRF = fitrensemble(X_train,y_train,'Method','Bag');
y_pred = predict(mdlRF,X_test);

% Tunnuslukuja
R2_LR = mdllinear.Rsquared.Ordinary % mitä lähempänä (1) sen parempi
RMSE_LR = mdllinear.RMSE % mitä pienempi sen parempi

SST = sum((y_test - mean(y_test)).^2); % kokonaisneliösumma
SSE = sum((y_test - y_pred).^2); % virheiden neliösumma
R2_RF = 1 - (SSE / SST)
RMSE_RF = sqrt(mean((y_test - y_pred).^2))
```

Ennusteiden piirtäminen ja vertailu:

```
subplot(2,1,2)
plot(y_test,'b');
hold on;
plot(y_pred,'r--');
hold off;
title('Satunnaismetsä');
xlim([0 225]);

sgtitle("Sähkönkulutuksen päiväkohtainen ennakointi")
han=axes('visible','off');
han.XLabel.Visible='on';
han.YLabel.Visible='on';
xlabel(han,'Datapisteiden lkm');
ylabel(han,'Sähkönkulutus (kWh)');
```