Tampereen yliopisto

Paula Jyrkönen

POIKKEAMIEN TUNNISTAMINEN JÄTE-VESIPUMPPAAMON MITTAUSDATASTA

Diplomityö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastajat:
professori (emeritus) Hannu Koivisto ja
professori Matti Vilkko
Huhtikuu 2022

TIIVISTELMÄ

Paula Jyrkönen: Poikkeamien tunnistaminen jätevesipumppaamon mittausdatasta
Diplomityö
Tampereen yliopisto
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Huhtikuu 2022

Tässä työssä tutkittiin koneoppimisen käyttöä häiriötilanteiden tunnistamisessa jätevesipumppaamoiden mittausdatasta. Työn tavoitteena oli selvittää, miten mittausdatassa esiintyviä poikkeamia voidaan tunnistaa ja mikä olisi tähän tarkoitukseen sopivin menetelmä. Poikkeamien tunnistamisen algoritmeista tuotettuja malleja voitaisiin käyttää apuna jätevedenpumppaamoiden kunnonvalvonnassa. Malli tunnistaisi pumppaamolla esiintyvän poikkeavan toiminnan ja siitä tehtävän ilmoituksen perusteella operaattori voisi tarkistaa, onko kyseessä todellinen häiriö ja päättää mahdollisista jatkotoimenpiteistä. Tällä hetkellä häiriöiden löytäminen vaatii operaattorien aktiivista mittausarvojen ja niistä tehtyjen graafisten esitysten tarkkailua.

Menetelmien testaamiseen ja arviointiin oli käytettävissä jätevesipumppaamoiden mittausten historiadataa. Pumppaamoilta ei kuitenkaan ollut saatavilla tarkkoja tietoja mahdollisista tapahtuneista häiriötilanteista, joten työssä päädyttiin käyttämään ohjaamattoman oppimisen poikkeamien tunnistamisen menetelmiä. Datan tuomien rajoitusten, mallin lopullisen käyttötarkoituksen sekä eri menetelmien tutkimisen jälkeen päädyttiin kokeilemaan ja arvioimaan neljää eri poikkeamien tunnistamiseen käytettävää algoritmia. DBSCAN-klusteroinnin, K-means klusteroinnin, isolation forestin sekä local outlier factorin algoritmeista opetettiin erilaisia malleja, joiden tunnistamia poikkeamia vertailtiin ja analysoitiin tarkoitukseen sopivimman algoritmin löytämiseksi.

Poikkeamien tunnistamisen mallien luominen algoritmien avulla on melko suoraviivaista. Mallin luomiseen käytettävä opetusdata on esikäsiteltävä, käytettävät piirteet valittava ja mallille annettavat parametrit määriteltävä, jonka jälkeen algoritmi suorittaa mallin opettamisen. Malli luokittelee jokaisen opetusdatajoukon pisteen joko poikkeamaksi tai normaaliksi pisteeksi. Haasteena työssä oli kunkin mallin antamien tulosten oikeellisuuden ja tunnistettujen poikkeamien oleellisuuden arvioiminen, sillä ohjaamattomille menetelmille ei voida laskea selkeitä mallien suorituskykyä kuvaavia tunnuslukuja. Eri algoritmien löytämiä poikkeamia arvioitiin ja analysoitiin muodostamalla samoista opetusdatajoukoista useita eri malleja ja vertailemalla niiden saamia tuloksia. Tämän lisäksi käytettiin erilaisia graafisia esityksiä mallien saamien tulosten tarkastelemiseen.

Algoritmien ja mallien arvioinnin lisäksi työssä pohdittiin erilaisia koneoppimisen algoritmien käyttämisen sekä työkalun toteuttamisen ja sen toimimisen tuomia vaatimuksia ja haasteita. Yksi selkeimmistä ongelmista on jokaisen koneoppimismallia käyttävän tapauksen yksilöllisyys, minkä takia jokainen tapaus on käsiteltävä erikseen. Kaikki käytettävä data on esikäsiteltävä erikseen, parametrit on optimoitava käytettävään dataan ja tapaukseen sopivaksi sekä työkalun toimintaa tulisi tarkastella aina käyttökohteessa tapahtuvien muutosten jälkeen uudestaan.

Tutkimus osoittaa, että ohjaamattoman koneoppimisen poikkeamien tunnistamisen menetelmiä voidaan käyttää apuna jätevedenpumppaamoiden kunnonvalvonnassa. Työssä käytetyt algoritmit löysivät pumppaamon mittausdatan seasta selkeitä poikkeamia. Tutkituista algoritmeista DBSCAN sekä iForest toimivat työssä tehtyjen analyysien perusteella parhaiten, joten niitä voidaan suositella harkittavaksi lopullisessa pumppaamoiden häiriöiden tunnistamiseen tarkoitetussa työkalussa käytettäväksi.

Avainsanat: kunnonvalvonta, poikkeamien tunnistaminen, koneoppiminen, ohjaamaton oppiminen, DBSCAN, K-means klusterointi, iForest, LOF

ABSTRACT

Paula Jyrkönen: Anomaly detection in measurement data of sewage pumping station
Master of science thesis
Tampere University
Master's Degree Programme in Automation Technology
April 2022

This thesis explored the use of machine learning to observe the faults of sewage pumping stations from measurement data. The aim of the thesis was to determine how anomalies in measurement data can be detected and what would be the most appropriate method for this purpose. Models created from algorithms for anomaly detection could be used as a tool in condition monitoring of sewage pumping stations. The model would identify anomalous activity occurring at the measurement data of the pumping station and based on the notification the system gives, the operator would be able to check whether there is a real fault and decide on any further actions. Currently, finding faults requires operators to actively observe measurement values and graphical charts.

Measurement data of wastewater pumping stations were available to test and evaluate the methods. However, accurate information about previous faults of pumping stations was not available, so the thesis ended up using unsupervised anomaly detection. Various methods were studied to choose suitable algorithms to be tested and estimated. Also, the constraints the data brought, and the purpose of use were considered when choosing the algorithms. Based on the requirements, DBSCAN clustering, K-means clustering, isolation forest and local outlier factor were selected as algorithms. Various models from the algorithms were trained and anomalies detected by the models were compared and analyzed to find the most suitable algorithm.

Creating models to detect anomalies using algorithms is quite straightforward. The training data used to create the model must be preprocessed, the features to be used must be selected and the parameters to be given to the model must be defined, after which the algorithm performs the training of the model. The model defines each point of the training data set as either an anomaly or a normal point. The challenge in the thesis was to assess the validity of the results given by each model and the relevance of detected anomalies, since clear indicators describing model performance cannot be calculated for unsupervised methods. Anomalies found by different algorithms were estimated and analyzed by forming several different models of the same training data sets and comparing the results they received. In addition to this, various graphic charts were used to examine the results obtained by the models.

In addition to evaluating the algorithms and the models, the work presented various requirements and challenges caused by using machine learning algorithms and implementing the tool. One of the clearest problems using these machine learning models is the individuality of each case, which is why each case must be handled separately. All data used must be preprocessed separately, the parameters must be optimized to fit the case and used data, and the operation of the tool should always be reviewed after any change in the system.

The study shows that the methods of anomaly detection with unsupervised machine learning can be used as a tool in the condition monitoring of sewage pumping stations. The algorithms used in the work found clear anomalies among the pumping station measurement data. Based on performed analyses it was found out that DBSCAN and iForest were the most suitable of four algorithms, so they can be recommended for consideration for use in the final tool to identify pumping station faults.

Keywords: condition monitoring, anomaly detection, machine learning, unsupervised learning, DBSCAN, K-means clustering, iForest, LOF

ALKUSANAT

Ensinnäkin suuret kiitokset diplomityöni tarkastajille Hannu Koivistolle ja Matti Vilkolle. Teiltä saadut neuvot, vinkit ja kannustavat sanat olivat korvaamattomia tämän työn kanssa painiessa. Kiitokset myös Instan puolelle Arttu Hanhelalle, joka tarjosi työn aiheen sekä sen suorittamiseen tarvittavat materiaalit.

On myönnettävä, että yliopistoon hakiessani olin yksi niistä tyypeistä, joka kuvitteli opiskelun olevan vain nopea viisivuotinen vaihe elämässä, joka olisi vain saatava äkkiä pois alta. Voi kuinka väärässä voi ihminen olla. Vaikka onkin vähän kliseisesti sanottu, niin sen kaiken hauskanpidon lisäksi ovat opiskeluvuodet myös opettaneet ja kasvattaneet ihmisenä. Mikään määrä luentoja, kirjoja tai Youtube-videoita ei voisi ikinä tarjota kaikkia näitä oheistoiminnan tuomia tietoja ja taitoja.

Kiitos kaikille ystäville, Puoliautomaattiselle, kuuneuvostolle, auteklaisille sekä muille, jotka olette olleet tässä matkassa mukana. Vaikka välillä on saanutkin kärsiä ja hävetä ihan kunnolla, on opiskelijaelämä ollut erittäin antoisaa ja näitä aikoja tulen aina muistelemaan lämmöllä.

Erityisen ison kiitoksen haluan esittää perheelleni, joka on aina ollut suurena tukena niin opinnoissa kuin elämässä ylipäänsä. Erityismaininta on annettava myös yliopiston ulkopuolisille ystäville, jotka ovat olleet muistuttamassa Herwantakuplan ulkopuolisesta elämästä. Ehkä juuri teidän ansiostanne olen edelleen niin järjissään kuin näiden vuosien jälkeen on edes mahdollista olla.

Kangasalla, 2.4.2022

Paula Jyrkönen

SISÄLLYSLUETTELO

KUVALUETTELO

LYHENTEET JA MERKINNÄT

AD	Anomaly detection, poikkeamien tunnistaminen
BST	Binary search tree, binäärinen hakupuu
DBSCAN	Density-Based Spatial Clustering of Applications with Noise, tiheyspohjainen klusterointimenetelmä
EDA	Exploratory data analysis, eksploratiivinen data-analyysi
FN	False negative, väärä negatiivinen arvo
FP	False positive, väärä positiivinen arvo
iForest	Isolation Forest, eristysmetsä
iTree	Isolation tree, eristyspuu
lkm.	lukumäärä
LOF	Local outlier factor, poikkeamien tunnistamisen menetelmä
LRD	Local reachability density, paikalinen saavutettavuustiheys
PCA	Principal component analysis, pääkomponenttianalyysi
RD	Reachability distance, saavutettavuusetäisyys
SSE	Sum of squared errors, jäännösneliösumma

1. JOHDANTO

1.1 Aiheen kuvaus

Toimiva vesihuolto on yksi ihmisen hyvinvoinnin ja laadukkaan elämän perusedellytyksistä. Vesihuolto voidaan jakaa puhtaan veden tarjoamisesta huolehtivaan vesijohtoverkostoon ja likaisesta vedestä huolehtivaan jäteviemäriverkostoon. Jätevesiverkostojen luotettava toiminta ja toimintavarmuuden ylläpito tulee olla huolehdittu kaikissa käyttöolosuhteissa. Toimivilla jätevedenpumppaamoilla ovat tärkeä osuus jätevedensiirrossa ja häiriöt niiden toiminnassa voivat käyttökatkosten lisäksi aiheuttaa jäteveden vuotoriskin vesistöihin tai muualle ympäristöön. Jäteveden sekä muiden haitallisten vesien keräyksestä, käsittelystä ja poisjohtamisesta huolehtivat paikalliset viemärilaitokset [37].

Kaukovalvonnan ansiosta viemäriverkostoon kuuluvia jätevedenpumppaamoita voidaan seurata ja mahdollisesti myös ohjata etänä [70]. Pumppaamoilta saatava mittausdata kerätään usein keskitetysti talteen raportointityökaluun, jonka kautta operaattorit ja valvomotyöntekijät pääsevät tarkastelemaan pumppujen ja pumppaamoiden toimintaa ja mittauksia sekä tarvittaessa tekemään syvällisempiä analyysejä niiden perusteella. Pumpuilta ei aina ole saatavilla suoria vika- tai häiriötietoja, eikä pumppaamon raportointijärjestelmässä välttämättä ole käytössä minkäänlaista hälytys- tai varoitusjärjestelmää, joiden antamien tietojen perusteella operaattorit tietäisivät kiinnittää huomiota häiriössä olevien tai muuten viallisesti toimivien pumppujen tai pumppaamoiden toimintaan. Löytääkseen poikkeuksellisesti käyttäytyvät pumppaamot on operaattorin aktiivisesti tarkkailtava pumppaamoiden tietoja järjestelmässä ja itse huomattava niiden poikkeuksellinen käyttäytyminen. Vikaantuminen ja kunnossapidon tarve näkyy laitteissa ja niihin liittyvissä mittauksissa yleensä muutoksina niiden normaaliin toimintaan verrattuna. Mittauksista voi olla rakennettu graafisia esityksiä, kuten kuvaajia ja diagrammeja, helpottamaan pumppaamoiden toiminnan seuraamista ja mittauksissa tapahtuvien muutoksien huomaamista.

Käytössä olevissa pumppaamoissa voi lähtökohtaisesti uskoa häiriötilanteiden olevan huomattavasti harvinaisempia tapauksia kuin niiden tarkoituksenmukainen toimiminen. Tästä syystä häiriötilanteet ovat poikkeamia kohteen normaalissa toiminnassa ja niiden löytämiseksi voidaan käyttää poikkeamien tunnistamista. Poikkeamien tunnistaminen on yleinen haaste, jota on tutkittu ja pyritty ratkaisemaan etenkin tilastomenetelmien sekä

koneoppimisen ja hahmontunnistuksen avulla [5]. Poikkeamien tunnistamisen tekniikoita on käytetty laajasti eri aloilla erilaisiin tarkoituksiin, kuten ohjelmistojen toiminnan tarkastelussa [44], verkkohyökkäysten [29, 40] ja luottokorttipetosten tunnistamisessa [3] sekä teollisuuden ja koneiden kunnonvalvonnassa [33].

1.2 Ongelma

Pumppujen toimittajilla on tarjolla omia palveluita pumppujen seuraamiseen ja kunnonvalvontaan, mutta nämä ratkaisut usein toimivat vain valmistajien omille pumpuille ja saattavat sijaita valmistajan omassa pilvipalvelussa. Viemäriverkostossa ja yksittäisessä pumppaamossa voi olla käytössä useamman eri valmistajan pumppuja, jolloin useiden eri kunnonvalvontajärjestelmien ylläpito nostaa kustannuksia ja erilaiset toimintamallit hankaloittavat operaattorien töitä. Kustannusten, selkeyden ja yhtenäisyyden takia halutaan rakentaa kaikille pumpuille ja pumppaamoille samalla tavalla toimiva kunnonvalvontaratkaisu.

Häiriöt pumppaamoiden toiminnassa voivat näkyä esimerkiksi erilaisina piikkeinä mittausten trendeissä sekä vuorottelukäytössä olevien pumppujen käyntiaikojen suhteiden tai muiden mittausarvojen muutoksina. Mittausten valvontaa voisi hoitaa asettamalla mittauksille ja niiden suhteille raja-arvoja sekä muita ehtoja. Sopivien raja-arvojen löytäminen, ylläpito ja päivitys jokaiselle yksilölliselle tapaukselle ilman koneoppimista voi olla työlästä ja tiukasti määriteltyjen raja-arvojen käyttö yleensä aiheuttaa suuren määrän virheellisiä hälytyksiä [59]. Toisaalta taas kaikista häiriöistä ei aiheudu hälytystä, jos ne eivät aiheuta raja-arvojen ylityksiä tai niiden vaikutuksia mittauksiin ei ole huomioitu määrittelyä tehtäessä. Staattiset raja-arvot eivät myöskään ole sopiva vaihtoehto monessa tilanteessa. Systeemien toiminnassa voi olla esimerkiksi olosuhteiden muutosten aiheuttamaa nopeaa vaihtelua sekä systeemin muuttumisen tai kehittymisen aikaansaamaa pidemmän ajan kuluessa tapahtuvaa muutosta [31].

Kun oletetaan, että pumppaamot toimivat suurimman osan ajasta niiltä vaaditulla tavalla, ovat erilaiset häiriötilanteet poikkeamia normaalin toiminnan seassa. Poikkeamien tunnistamisen menetelmien avulla voidaan tunnistaa juuri tällaisia epätavallisia tiloja tai käyttäytymismalleja tarkasteltavan systeemin ja siitä saatujen tietojen, kuten mittausten tai tilatietojen, perusteella. Poikkeamien tunnistamista varten on olemassa erilaisia tekniikoita [12]. Tässä työssä yhtenä lähtökohtana oli juuri koneoppimisen hyödyntäminen, joten erilaiset statistiset sekä informaatioteoriaan perustuvat poikkeamien tunnistamisen menetelmät on jätetty työstä pois.

Pumppaamoilta saatavaa mittausdataa voidaan käyttää koneoppimisessa opettamaan algoritmi tunnistamaan normaalista poikkeavia tilanteita. Useissa koneoppimista hyödyntävissä, vastaavanlaisen ongelman ratkaisuissa on käytetty ohjattua oppimista tai normaalin käyttäytymisen opettamista häiriöttömästä datasta [3, 51, 66]. Tämän työn tarkastelukohteilta on saatavilla vain merkitsemätöntä (engl. unlabeled data) ja häiriötilanteet sisältävää opetusdataa, minkä takia on käytettävä ohjaamattoman oppimisen tapoja poikkeamien tunnistamiseen.

1.3 Tavoitteet ja rajaukset

Työssä tutkitaan pumppujen vikaantumisen heijastumista pumppaamon mittauksiin ja pyritään pumppaamoiden historiadatan avulla luomaan malli, joka tunnistaa pumppujen vikatilanteita reaaliaikaisesti mittausdatan perusteella. Työn tavoitteet on tiivistetty seuraaviin tutkimuskysymyksiin:

1. Mikä on vikaantuneelle pumpulle tunnusomaista pumppaamon mittausdatassa?
2. Miten ohjaamattoman koneoppimisen avulla voidaan tunnistaa poikkeamia pumppaamoiden toiminnassa?
3. Mikä tutkituista poikkeamien tunnistamisen metodeista sopii parhaiten työn tapaukseen?

Laitteen todelliseen kuntoon perustuva kunnossapitostrategia sisältää kolme päävaihetta: datan hankinta, datan käsittely ja kunnossapitopäätösten tekeminen [17]. Tässä työssä mallin luomisessa käytetty pumppaamoiden historiadata on saatu suoraan raportointijärjestelmästä, eikä raakadataan ole voitu vaikuttaa työtä tehtäessä. Tästä syystä datan hankintaa ei käsitellä työssä kovin tarkasti. On kuitenkin tärkeää pystyä tunnistamaan ja valitsemaan käytettäväksi oleelliset pumppujen ja pumppaamoiden tilaa heijastavat mittaukset ja tilatiedot. Käytössä olevaa dataa voidaan esikäsitellä, kuten yhdistellä, poistaa ja skaalata, joten on hyödyllistä tuntea pumppaamoiden toimintaperiaate ja erilaisten vikatilanteiden heijastuminen pumppaamoiden mittausdataan.

Myös kunnossapitopäätösten tekeminen rajataan työn ulkopuolelle. Työn tavoitteena on tarjota mahdollisimman hyvä työkalu operaattorille, joka voi itse tarkastaa tilanteen ja tehdä lopullisen kunnossapitopäätöksen.

1.4 Metodit ja materiaalit

Työssä käytetään koneoppimisen algoritmeja, joiden avulla muodostetaan pumppaamoiden kunnonvalvontaan osallistuvia malleja. Mallien opettaminen tapahtuu käyttämällä

oikeaa pumppaamoilta saatua historiadataa. Historiadata on saatu Instawahti-raportointijärjestelmästä ja opetuksessa on käytetty aikavälillä 1.1.-31.12.2020 kerättyä mittausdataa.

Pumppaamoilta ei ole saatavilla kattavaa tietoa aiemmista vikatilanteista tai tehdyistä huoltotoimenpiteistä, joten on käytettävä ohjaamattoman oppimisen tapoja. Käytännön osuudessa luodaan 4 erilaista mallia tunnistamaan pumppausprosessissa tapahtuvia poikkeus- ja häiriötilanteita. Käytössä olevan historiadatan, tutkimuksen luonteen ja aiempien aiheeseen liittyvien tutkimusten perusteella on pyritty valitsemaan työn tapaukseen sopivat mallit.

Mallit luodaan Python-ohjelmointikielellä käyttäen valmiina olevia koneoppimiseen tarkoitettuja kirjastoja. Kirjastot sisältävät valmiit funktiot eri koneoppimisalgoritmien luomiseen. Kyseiset kirjastot ovat avoimeen lähdekoodiin perustuvia, joten ne ovat vapaasti kaikkien saatavilla ja käytettävissä.

Aiempia häiriötilanteita koskevan tiedon puuttumisen takia ei mallien toimivuutta tai tarkkuutta pystytä suoraan osoittamaan tämän työn puitteissa. Useamman tutkittavan pumppaamon ja eri algoritmien käyttö kuitenkin mahdollistavat tehtyjen mallien vertailun toisiinsa, esimerkiksi sen perusteella, kuinka suuri osa niiden löytämistä poikkeamista on samoja ja kuinka paljon mallien väliset tulokset vaihtelevat eri pumppaamoille. Mallien löytämiä poikkeamia voidaan myös tarkastella erilaisten graafisten esitysten avulla.

1.5 Työn rakenne

Työ jakautuu kuuteen lukuun, joista ensimmäinen johdattelee aiheeseen esittelemällä työn ongelman, tavoitteet, rajaukset sekä toteutuksen. Luvuissa 2 ja 3 käydään läpi aiheeseen liittyviä teorioita yleisellä tasolla. Toisessa luvussa esitellään jätevedenpumppaamot osana viemärijärjestelmää, tutustutaan jätevedenpumppaamoiden häiriöihin ja yleisesti kunnonvalvontaan sekä poikkeamien tunnistamiseen. Luvussa 3 käydään läpi työssä käytettävät koneoppimisen algoritmit ja muita työssä apuna käytettäviä metodeja. Tämän jälkeen luvussa 4 sovelletaan edellisessä luvussa läpi käytyjen poikkeamien tunnistamisen metodeja luvussa 2 esiteltyyn järjestelmään ja käydään läpi mallien luomisen vaiheita. Viidennessä luvussa esitellään saatuja tuloksia, vertaillaan käytettyjä malleja ja käydään läpi tulosten perusteella tehdyt päätelmät. Viimeisessä luvussa vedetään yhteen työn suoritus.

2. TEORIA

2.1 Jätevesiverkosto ja jätevedenpumppaamot

Viemärilaitoksen tehtävänä on kerätä ihmisten käyttämä jätevesi, kuljettaa se jätevedenpuhdistamolle sekä käsitellä ja palauttaa luonnonympäristöön aiheuttamatta vaaraa tai suurempaa haittaa ympäristölle tai ihmisille. Jätevesijärjestelmään johdetaan asumis- ja teollisuusjäteveden lisäksi sateesta ja lumen sulamisesta sekä perustusten kuivatusvesistä muodostuvaa hulevettä ja sinne päätyy myös esimerkiksi putkirikoista ja vuotavista putkiliitoksista verkostoon pääsevää vuotovettä. Hulevesi voi kulkea joko omassa viemärissään, jolloin kyseessä on erillisviemäröinti tai talous- ja teollisuusjäteveden kanssa samassa viemärissä, jolloin puhutaan sekaviemäröinnistä. Kaikki uudet viemärijärjestelmät ja vanhojen viemäreiden saneeraukset toteutetaan Suomessa erillisviemäröinteinä. [37, 38]

Jätevesien siirtolinjat koostuvat yleensä paine- ja viettoviemäreistä sekä jätevedenpumppaamoista. Kaltevissa viettoviemäreissä jätevesi kulkee painovoiman avulla, kun taas paineviemäreissä käytetään jätevedenpumppaamoiden aikaansaamaa painetta jäteveden liikuttamiseen paineputkissa. Paineviemärit ovat käytössä viemäriverkoston osissa, joissa viemäriputkille ei saada riittävää kaltevuutta tai viettoviemärin rakentaminen olisi esimerkiksi maaperän olosuhteista johtuen kallista. Paineviemäreiden ja pumppauksen käyttö jäteveden kuljettamiseen on välttämätöntä siis esimerkiksi siirtolinjan ollessa pitkä, ylitettävän maaston ollessa tasainen tai kun jätevesi halutaan kuljettaa vedenjakajan tai mäen yli tai vesistön ali. Usein paine- ja viettoviemäreitä käytetään vuorotellen. Paineviemärin avulla jätevesi voidaan pumpata ylempään korkeustasoon, jolloin sen painovoimainen eteenpäin kuljettaminen on mahdollista tai paineviemäriä voidaan käyttää vakiosyvyydessä, jolloin viemärin korkeustaso ei muutu [35]. Jätevettä voidaan joutua pumppaamaan siirtolinjan pituudesta riippuen useita kertoja sen siirron aikana. Normaaleissa olosuhteissa viettoviemärit ovat tällä hetkellä paineviemäreitä yleisempiä. [37, 38, 36]

Jätevedenpumppaamo koostuu yleensä ainakin pumpuista, imualtaista, imuputkistosta, paineputkistosta sekä venttiileistä [53]. Jätevesipumput voidaan asentaa niin sanotulla märkäasennuksella, jolloin pumppu on jäteveden kanssa samassa säiliössä tai kuivaasennuksella, jolloin pumppu sijaitsee omassa säiliössä. Pumppaamossa on usein ainakin kaksi pumppua toimintavarmuuden takaamiseksi. Pumppujen määrä riippuu usein pumppaamon kriittisyydestä ja pumppaustarpeesta. Pumput ovat normaalisti

vuorottelukäytössä, jollei pumppaustarve vaadi useamman pumpun käyttöä yhtäaikaisesti. Yhden pumpun vikaantuessa muut pumput korvaavat sen toiminnan.

Paineviemäreistä jätevesi tulee pumppaamoon etukaivon (tulokaivo) kautta, josta se tuodaan pumppaamokaivoon yhtä putkea pitkin [60]. Etukaivon käytön ansiosta jäteveden johtaminen pumppaamokaivoon on hallitumpaa ja mahdollista suorittaa tasaisempana virtana. Pumppaamoiden toimintaa ohjataan usein pumppaamokaivon pinnansäädöllä. Pintakytkimen tai paineanturin havaitessa jätevedenpinnan nousevan kaivossa sille määriteltyyn ylärajaan eli käynnistysrajaan, käynnistää ohjaus pumpun. Pumppu pysähtyy jätevedenpinnan saavuttaessa sille asetetun alarajan eli pysäytysrajan.

2.2 Jätevedenpumppaamoiden häiriöt

Viemärijärjestelmän vikaantuminen voi aiheuttaa laajoja taloudellisia ja ympäristöllisiä vahinkoja, joten järjestelmän toimintavarmuuteen tulee kiinnittää erityistä huomiota [39]. Jätevedenpumppaamot ovat jätevesijärjestelmän kriittisiä komponentteja, joten niiden toiminta on suoraan vaikutuksessa koko järjestelmän toimivuuteen. Miszta-Krukin viemäriverkostoja käsittelevän tutkimuksen [47] mukaan yli 90 % tutkituista paineviemärien vikatilanteista johtui jätevedenpumppaamoiden pumppujen häiriöistä.

PSK-standardissa [71] määritellään vikaantumiseen liittyviä termejä, joita käytetään usein hieman epätarkasti ja osin päällekkäin. Vikaantumiseksi kutsutaan tapahtumaa, ”jonka seurauksena kohteen kyky suorittaa vaadittu toiminto päättyy”. Vika taas määritellään tilaksi, ”jossa kohde ei kykene suorittamaan vaadittua toimintoa täydellisesti pois lukien ehkäisevän kunnossapidon, jonkin muun suunnitellun toimenpiteen tai ulkoisten resurssien puutteesta johtuvan toimintakyvyttömyyden takia”. Häiriöksi taas luokitellaan tilanne, joka ”aiheuttaa tuotannon menetyksiä ja välittömän korjaustarpeen”. [71]

Pumppujen tapauksessa vika tarkoittaa tilaa, jossa se ei pysty suorittamaan sille annettua tehtävää eli pumppaamaan jätevettä vaaditulla kapasiteetilla. Pumppujen vikatyyppejä ovat erilaiset lämpö- ja sähköviat, mekaaniset viat sekä ohjausyksikön virheelliseen ohjelmointiin tai pumppujen virheelliseen operointiin liittyvät viat. Korving et al. tekemässä jätevesipumppuja ja niiden vikaantumista käsittelevässä tutkimuksessa [39] todetaan huomattavasti suurimman osa tutkittavien pumppujen vikatilanteista olevan mekaanisia vikoja, jotka johtuivat pumppujen tukkeutumisesta. Syitä tukoksille voivat olla esimerkiksi jäteveden koostumus, sopimattomien jätteiden joutuminen viemäriin tai liian pieni paine ja virtaus viemärissä. Muita mahdollisia vikojen aiheuttajia voivat olla komponenttien kuluminen, ympäristöolosuhteet, puutteellinen tai väärä kunnossapito sekä

virheet pumppaamon tai muun viemärijärjestelmän suunnittelussa, kokoonpanossa tai asennuksessa [10, 61, 47].

Viemäriverkoston kunnon ja toiminnan ylläpitäminen vaatii aktiivisempaa valvontaa vedenjakeluverkoston kunnonvalvontaan verrattuna. Käyttäjät raportoivat herkästi vedenjakelussa ilmenevistä ongelmista, kuten heikosta paineesta, veden huonosta laadusta tai riittämättömästä määrästä. Viemärijärjestelmän osalta käyttäjien voi olla hyvin vaikea havainnoida järjestelmän kuntoa ja huomata puutteita sen toiminnassa. Jätevesiverkoston kunnonvalvonta ja sen toiminnan ongelmien havaitseminen jäävät viemärilaitoksen vastuulle, jotta puutteet toiminnassa voitaisiin huomata ja korjata ajoissa. Ilman asianmukaista kunnonvalvontaa viemäriverkoston toiminnan ongelmat saattavat jäädä huomaamatta, ennen kuin ne ovat jo aiheuttaneet vahinkoa ympäristölle tai ovat työläitä korjata. Ylivuodot ovat mahdollisia seurauksia pumppaamoiden häiriöistä. Suurissa määrissä ne voivat aiheuttaa vahinkoa pumppaamon omille laitteille sekä haittaa ympäristölle ja läheisille vesistöille. [38]

2.3 Kunnonvalvonta

Kunnonvalvonta on tärkeänä ohjaavana osana kunnossapitoa, sillä kuntoon perustuvan tai ennustavan kunnossapidon toteuttamiseksi on kunnossapidettävän kohteen kunnosta oltava todenmukaista ja ajantasaista tietoa. PSK-standardien värähtelymittausten kunnonvalvontaa koskevan standardin [72] mukaan kunnonvalvonta määritellään olevan ”koneen kuntoa ilmaisevien tietojen ja datan selville saanti ja keruu”. Kunnonvalvonnan tarkoituksena on pyrkiä havaitsemaan häiriöt ajoissa, jotta niihin voidaan reagoida ennen suurempia vahinkoja tai vikoja, voidaan varata varaosia ja henkilöresursseja sekä ajoittaa tarvittavia kunnossapitotöitä suunniteltuihin huoltoseisokeihin. [30, 61]

Kunnonvalvontaa voidaan suorittaa etänä tai paikallisesti tarkasteltavan kohteen luona. Etävalvonta suoritetaan kauempana kohteen välittömästä läheisyydestä. Valvonta voidaan suorittaa itsenäisenä toimena tai esimerkiksi osana kohteen ohjausta hoitavaa automaatiojärjestelmää. Seuranta ja diagnosointi voivat olla joko automaattisesti tai manuaalisesti suoritettavia. Aktiivista kohteen kunnon tarkastelua voidaan joko suorittaa jatkuvasti, tietyin väliajoin tai käyttäjän määrätessä. Etäluettavat anturit, langattomat tiedonsiirtomenetelmät, mobiilisovellukset ja kaukovalvotut prosessit mahdollistavat kunnonvalvonnan suorittamisen lähes kokonaan etänä. Paikallisesti toteutetusta kunnonvalvonnasta huolehtii yleensä kunnossapitoinsinööri, teknikko tai operaattori. Paikallisesti suoritettava kunnonvalvontaprosessi sisältää haluttujen kunnonvalvontatietojen ja mittausten keräämisen ja tallentamisen valvottavan kohteen kunnon määrittämiseksi.

Näiden tietojen perusteella tulee arvioida, onko komponentin kunto hyväksyttävä vai vaatiiko kohde kunnossapitotoimia. [26]

Kunnonvalvonta on tutkittavan kohteen jatkuvaa tai ajoittaista seurantaa ja analysointia. Kunnonvalvonta sisältää 3 vaihetta, jotka ovat datan kerääminen, datan käsittely ja päätöksenteko. Kerättävä data voi sisältää normaalia prosessista ja laitteista saatavaa mittaus- ja tilatietodataa sekä erikseen kunnonvalvontaa varten asetettujen anturien ja kerättävien mittausten dataa. Kunnonvalvonnassa käytettyjä antureita ovat esimerkiksi erilaiset värähtelymittauksiin käytetyt kiihtyvyys-, nopeus- ja siirtymäanturit, ultraäänianturit sekä akustisen emission mittaamiseen tarkoitetut anturit [50, 30].

Raa'an mittausdatan käsittely voidaan hoitaa mittalaitteessa tai keskitetysti automaatiojärjestelmän ylemmillä tasoilla. Älykkäät kenttälaitteet hoitavat osin omaa diagnosointiaan ja tunnistavat häiriöiden tai vikaantumisten merkkejä itsessään. Kenttälaitteet ilmoittavat vikatilanteista omassa paneelissaan tai lähettävät automaatiojärjestelmään tiedot tekemistään havainnoista. Joissakin automaatiojärjestelmissä on tarjolla omia diagnostiikkatyökaluja, joiden avulla voidaan analysoida mittausdataa, seurata antureita ja laitteita sekä tehdä omia hälytyksiä. Mittausdataa voidaan käsitellä myös automaatiojärjestelmän ulkopuolella, käyttäen joko omia tai muiden valmistajien kehittämiä data-analytiikkatyökaluja. [25, 1, 73]

Usein lopullinen vastuu kunnossapitopäätöksen teosta on ihmisellä. Erilaisten kunnonvalvontajärjestelmien tarkoituksena on toimia mahdollisimman tehokkaana apuvälineenä päätöksen tekemiseen. Kunnossapitopäätöksen tekemistä varten voidaan kunnonvalvonnan avulla saada tietoa hälytyksinä, diagnooseina ja prognooseina. Hälytysten tarkoituksena on ilmoittaa kohteen nopeasti kehittyvästä viasta, joka saattaa aiheuttaa keskeytyksiä tai muuta vahinkoa kohteen toimintaan. Diagnoosi kertoo kehittyvästä viasta ja sen tyypistä ja prognoosi ennustaa vian kehittymistä ja arvioi kohteen jäljellä olevaa käyttöikää. [50, 34]

2.4 Koneoppiminen kunnonvalvonnassa

Kunnonvalvonnan ja kunnossapidon onnistumisen kannalta on tarkasteltavan kohteen tehtävä ja siltä vaadittava suorituskyky tunnettava. Yhtenä haasteena kunnonvalvonnassa on löytää juuri ne tiedot, toiminnot ja mittaukset, joihin häiriöiden vaikutukset heijastuvat sekä näiden mittausten raja-arvot, joiden ylityttyä voidaan ennustaa häiriön tarpeeksi suurella todennäköisyydellä tapahtuvan [63]. Koneoppimisen avulla voidaan reaaliaikaisesti löytää datasta merkkejä ja erikoisia käyttäytymismalleja, jotka johtuvat tulevasta häiriöstä. Näiden varhaisen tunnistamisen ansiosta pystytään minimoimaan

häiriön prosessiin vaikuttava aika [6]. Koneoppimisen avulla pystytään käymään läpi suuri määrä dataa tehokkaasti ja löytämään sellaisia merkkejä häiriöistä, joita ihmisen voi olla vaikea tai jopa mahdotonta löytää dataa tarkastelemalla. Koneoppivan kunnonvalvonnan käyttämisen ansiosta voidaan myös ainakin osin poistaa jatkuvan aktiivisen tarkkailun tarve, kun siihen liitetään ilmoitus- tai varoitusjärjestelmä, joka viestittää halutulle taholle tekemistään havainnoista.

Koneoppiminen on tekoälyn osa-alue, jossa ohjelma pystyy oppimaan itsenäisesti opetusdatan perusteella säännönmukaisuuksia ja luomaan yleistäviä malleja, jotka osaavat käsitellä analysoitavaa dataa. Mallien oppiminen tapahtuu ilman erillistä ohjelmointia tai toimintaohjeita. Koneoppimisen perimmäisenä tavoitteena on, että malli pystyisi lisää opetusdataa saadessaan kehittämään itseään ja parantamaan suorituskykyään. Tämä kuitenkin vaatii mallin säännöllistä opetusta uuden datan avulla, mikä käytännön sovelluksissa toteutuu todella harvoin. [20, 7]

Opetetun mallin avulla pystytään esimerkiksi ennustamaan syötteiden tai nykytilan perusteella lopputulemaa, opettelemaan toimintoketjuja, luokittelemaan asioita tai löytämään piilotettuja riippuvuuksia ja syy-seuraussuhteita. Pohjana koneoppimiselle on algoritmeja, jotka erilaisten tilastotieteen ja informaatioteorian menetelmien avulla muodostavat malleja datan rakenteeseen perustuen. Tavallisia koneoppimisen käyttökohteita ovat erilaiset luokittelu-, regressio- ja klusterointitehtävät. [20, 7, 6]

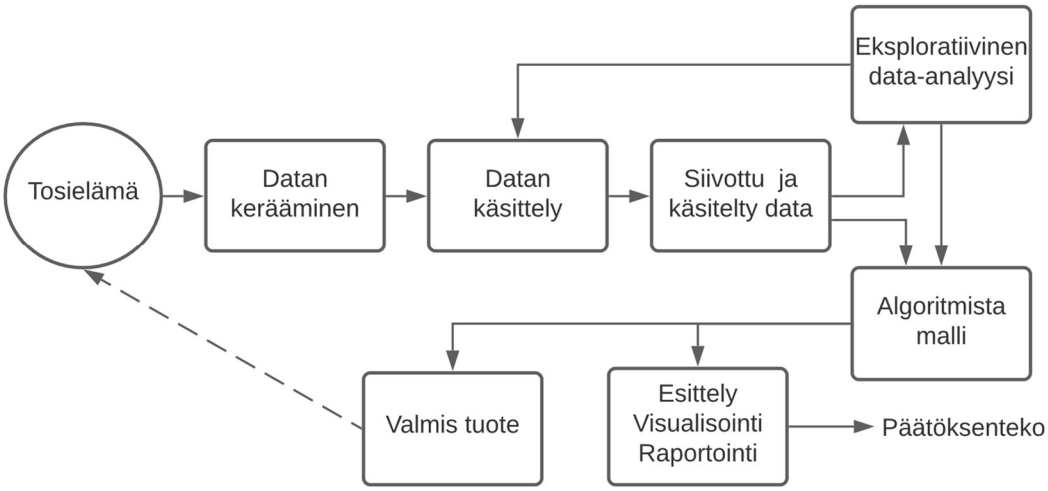
Koneoppimista ja sen algoritmeja voidaan käyttää osin samoihin tarkoituksiin kuin ihmisten käsin asettamia raja-arvotarkasteluja sekä loogisia ehtolauseita. Itse tehtyjen sääntöjen suuria heikkouksia ovat niiden vaatima spesifisyys ja syvä tietämys käsiteltävästä tapauksesta ja sen toiminnasta. Jokaiselle tapaukselle tulee laatia omat säännöt sekä päätöksentekoketjut ja pienikin tapauksessa tapahtuva muutos saattaa vaatia ongelman uudelleentarkastelua ja sääntöjen muutosta. Manuaalinen tapa vaatii ihmiseltä tarkkaa tietämystä kaikista eri vaihtoehdoista, raja-arvoista ja ymmärrystä erilaisista syy-seuraussuhteista. Koneoppimisen avulla voidaan välttää näitä ongelmia ainakin osin. [48]

Kunnonvalvontaan liittyvät ongelmatilanteet ovat melko yksilöllisiä, joten yhdestä kohteesta kerättyä dataa ei voida usein yleistää koskemaan kaikkia samantyyppisiä kohteita tai käyttää muuten apuna muissa kohteissa käytettävien mallien opettamisessa. Kohteiden käyttöympäristöt, käyttötavat ja kuormitukset voivat vaihdella ja erojen takia algoritmin opettamiseen tulisi käyttää valvottavasta kohteesta kerättyä dataa. Optimitilanteessa opetusta varten on saatavilla riittävästi dataa, data edustaa mahdollisimman kattavasti erilaisia tilanteita ja sisältää mahdollisimman vähän kohinaa.

Käytettävän datan ja ratkaistavan tehtävän tunteminen ovat tärkeitä koneoppimisen onnistumisen kannalta, jotta osataan valita kyseiselle tapaukselle sopiva koneoppimisen tapa, esikäsitellä data ja tunnistaa tapauksen kannalta olennaiset piirteet. Kunnonvalvontaan osallistuvan koneoppivan tavan valinta riippuu käyttötapauksesta sekä saatavissa olevasta datasta ja muista tiedoista. Käytettyjä tapoja ovat olleet esimerkiksi luokittelu, regressio, kohteen jäljellä olevan eliniän laskeminen ja poikkeamien tunnistaminen [31, 16, 62].

2.5 Datatiede

Koneoppivan järjestelmän luomiseen liittyvää prosessia voidaan pitää osana datatiedettä. Datatiede (engl. data science) tieteenalana käsittää datan hyödyntämisen jonkin lopputuotteen rakentamisessa. Nykyään käsiteltävän datan määrä on usein niin suuri, että sen käsittely ja analysointi ihmisvoimin on erittäin kallista tai jopa mahdotonta. Datatieteen prosessi pitää sisällään tieteelliset metodit, algoritmit ja muut analysointimenetelmät, joiden avulla datan sisältämää tietoa voidaan hyödyntää. Datatieteen prosessiin kuuluu tietyt vaiheet, jotka liittyvät ongelman ja datan valmisteluun, datan esikäsittelyyn, analyysiin sekä jälkikäsittelyyn ja mahdollisiin jatkotoimenpiteisiin. Datatieteen alle kuuluvien alojen ja niin myös koneoppimisen ratkaiseminen tapahtuu pääosin samalla kaavalla. Käytännössä prosessi ei kuitenkaan ikinä ole suoraviivainen, vaan kehitystyö tapahtuu iteratiivisesti ja hyvän lopputuloksen saamiseksi voidaan prosessissa joutua palaamaan useasti takaisin aikaisempiin vaiheisiin. Datatieteen prosessin kulku ja sen vaiheet on esitetty kuvassa 1.



Kuva 1. *Datatieteen prosessi, muokattu lähteestä* [58].

Ensimmäisenä vaiheena olisi hyvä olla kunnollinen valmistelu. Koko prosessin lähtökohtana tulisi aina olla jokin tapaus tai ongelma, joka pyritään ratkaisemaan datatieteen

avulla. Määritelty ongelma ja tavoite ohjaavat koko prosessia ja niiden perusteella pitäisi tapahtua kaikki dataan ja malliin liittyvät toimet ja päätökset. Tässä vaiheessa on kuitenkin hyvä tiedostaa tarjolla olevat resurssit ja niiden puutteiden tuomat rajoitukset, jotka vaikuttavat esimerkiksi valittavaan tekniikkaan ja saatavaan lopputulokseen. Usein käytettävä raakadata voidaan hakea esimerkiksi jonkin valmiin järjestelmän mittausten historiatietokannasta tai data on muuten valmiina. Näissä tapauksissa dataan voidaan usein vaikuttaa korkeintaan tarkemman käsittelykohteen tai tutkittavan ajanjakson valinnalla. Joissain tapauksissa data ei ole valmiina, jolloin datan hankinnan tai keräyksen suunnittelu ja toteutus sisältyvät prosessin valmisteluvaiheeseen. Tällöin voidaan jo suunnitteluvaiheessa määritellä, minkälaista dataa parhaan lopputuloksen saamiseksi tarvitaan. Toisaalta erikseen tapahtuva datan keräys tuo prosessiin yhden työvaiheen lisää ja kasvattaa koko prosessin työmäärää sekä toteuttamiseen kuluvaa aikaa. [13, 21, 4]

Valmistelun jälkeen tapahtuu datan esikäsittely. Esikäsittely voi sisältää kaikenlaisen datan puhdistuksen, suodatuksen, skaalauksen, täydentämisen, korjauksen ja muuntamisen sekä uusien piirteiden luomisen ja mallin muodostamisessa käytettävien piirteiden valinnan [58]. Koneoppimisessa piirteillä tarkoitetaan lähdeaineistosta koottuja tai muodostettuja ominaisuusjoukkoja. Piirteet voivat olla esimerkiksi erilaisia attribuutteja, diskreettejä muuttujia, jatkuvia mittausarvoja tai tunnuslukuja, jotka kuvaavat jollain tavalla data-aineiston yksittäisiä pisteitä [43].

Raakaa datajoukkoa tulee käsitellä puuttuvien tai virheellisten havaintojen täydentämiseksi, poistamiseksi ja korjaamiseksi. Tarvittaessa datan seasta voidaan poistaa tai korjata muista pisteistä merkittävästi eroavat poikkeamat. Muodostettavia malleja käyttämällä voidaan päättyä virheellisiin lopputuloksiin, jos datan puutteita ja ongelmia ei huomioida ja käsitellä asianmukaisesti ennen datan käyttöä mallien opetusmateriaalina. [8]

Datan eri ominaisuuksia kuvaavien piirteiden arvoalueet voivat vaihdella paljon ja erilaiset suuruusluokat voivat aiheuttaa joidenkin piirteiden korostumista toisiin verrattuna. Skaalaamalla eri piirteet samalle alueelle voidaan yhdenmukaistaa niiden vaikutukset mallien laskennassa [55]. Mallit käsittelevät pääasiassa pelkästään lukuja, joten sanallisia attribuutteja ja muuttujia sisältävät piirteet tulee muuntaa jollain tavoin numeerisiksi arvoiksi. Mallin muodostamista varten voidaan muodostaa myös uusia piirteitä alkuperäisten tietojen pohjalta esimerkiksi erilaisin matemaattisin keinoin.

Piirteiden valinta on mallin luomiseen käytettävän ominaisuusjoukon määrittämistä kaikkien alkuperäisten, muokattujen ja luotujen ominaisuuksien joukosta. Mallien ja

selvitettävän ongelman kannalta merkitykselliset piirteet valitaan tähän alijoukkoon, jota voidaan käyttää mallien opettamiseen koko alkuperäisen piirrejoukon sijasta. Kaikkien datan piirteiden käyttäminen voi laskea monien mallien tarkkuutta, sillä suuri piirrejoukko sisältää todennäköisesti mallin kannalta merkityksetöntä informaatiota, joka lopullisessa mallissa vaikuttaa häiritsevästi mallin toimintaan. Turhien piirteiden karsiminen on hyödyllistä erityisesti tapauksissa, joissa datapisteitä tai näytteitä on suhteellisen vähän ja piirteitä paljon. Tarkkuuden parantamisen lisäksi piirteiden valinta auttaa tunnistamaan tärkeimpiä ominaisuuksia ja vähentämään mallin opetusaikaa.

Datan käsittelyn jälkeen tapahtuu mallin luominen. Ongelman ratkaisemiseksi valitaan jokin koneoppimisen algoritmi tavoitteen ja käytettävän datan perusteella. Mallin muodostaminen tapahtuu iteratiivisesti, kun mallille määritellään parametrit, suoritetaan mallin opetus ja arvioidaan sen suorituskykyä. Mallin ensimmäisten versioiden luomisen jälkeen voidaan myös palata datan esikäsittelyyn ja kokeilla erilaisten piirrejoukkojen käyttämisen vaikutusta malliin. Hyvänä tapana voi olla toteuttaa useita malleja erilaisilla tekniikoilla tai piirrejoukoilla, vertailla niiden suoriutumista tehtävästä ja käyttää valittavien kriteerien perusteella tapaukseen parhaiten sopivaa mallia. Mallien ja niiden toiminnan analysointi ja vertailu riippuvat tarjolla olevasta datasta ja käytetyistä metodeista. Esimerkiksi merkitty data ja ohjattu koneoppiminen mahdollistavat erilaisten suorituskykyä kuvaavien tunnuslukujen, kuten tarkkuuden ja täsmällisyyden laskemisen [68]. Ohjaamattoman koneoppimisen algoritmien toimimisen arvioimiseen ei taas ole suoraan laskettavia tunnuslukuja, vaan arviointi on suoritettava muilla tavoin. [13]

Lopulta saaduista tuloksista voidaan tehdä omat tulkinnat ja tavoitteen kannalta oleelliset johtopäätökset sekä arvioida lopullisia tuloksia. Saaduista tuloksista voidaan luoda esimerkiksi erilaisia graafisia esityksiä auttamaan tulosten havainnollistamisessa. Vaiheiden suorituksen, tehtyjen toimenpiteiden ja saatujen tulosten dokumentointi voidaan suorittaa prosessin myöhempää tarkastelua ja mahdollista käyttöä varten. Malli saattaa olla luotuna testiympäristöön, joten sen käyttäminen voi vaatia mallin siirtämisen oikeaan toimintaympäristöön ja malli on erikseen muokattava ja käyttöönotettava toimiakseen myös siellä. Järjestelmien välille voi myös joutua luomaan rajapintoja tiedon liikkumisen toteuttamiseksi. Mallin tuottama lopputulos ei myöskään välttämättä ole suoraan määritellyn tavoitteen lopullinen ratkaisu tai käytettävissä ongelman ratkaisemiseksi, vaan mallilla saatu tulos voi toimia apuvälineenä tai osana lopullista ratkaisua.

2.6 Poikkeamien tunnistaminen

Poikkeamien tunnistaminen (engl. anomaly detection, AD) on prosessi, jossa etsitään annetusta datajoukosta merkittävästi muusta datasta poikkeavia arvoja tai tapahtumia.

Poikkeamat voivat olla yksittäisiä datapisteitä tai useamman datapisteen muodostamia joukkoja, jotka eivät käyttäydy odotetulla tavalla tai joiden ominaisuudet eroavat normaaleista tapauksista [41, 12].

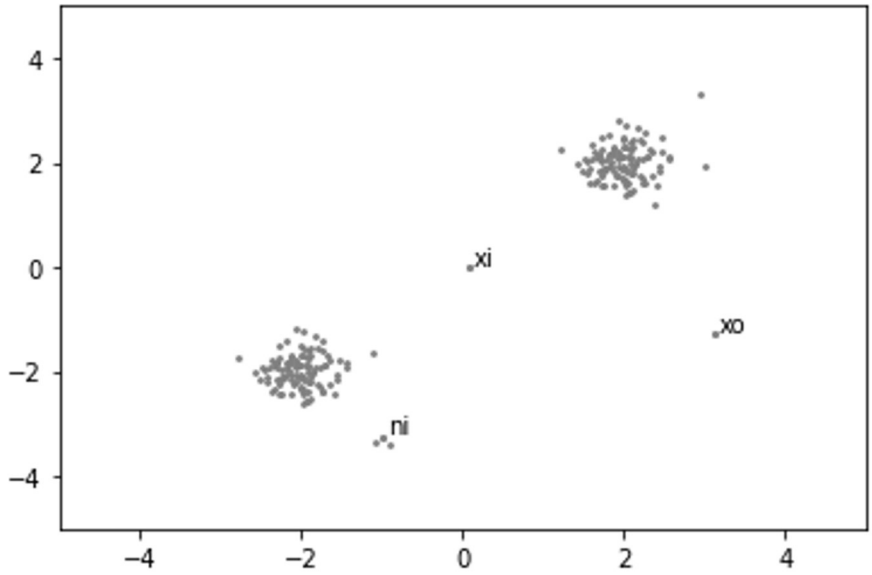
Poikkeamien tunnistamisella voi olla kaksi eri lähtökohtaa riippuen siitä, onko mielenkiinnon kohteena poikkeamat vai normaali data. Ensimmäisessä tapauksessa on kiinnostuttu poikkeamista itsestään ja niiden taustasyistä, sillä epänormaalit arvot mittausdatassa viittaavat mitattavan prosessin poikkeukselliseen käyttäytymiseen. Poikkeavan käyttäytymisen ja poikkeamien muodostumisen taustalla on syitä, joiden löytämisestä ja tunnistamisesta on kiinnostuttu. Tähän perustuu esimerkiksi poikkeamien tunnistamisen käyttäminen kunnonvalvonnassa ja luottokorttipetosten etsinnässä. Toisessa tapauksessa tavoitteena on yleensä datan puhdistaminen kohinasta sekä virheellisistä tai muuten poikkeavista arvoista. Poikkeamat itsessään eivät kiinnosta, mutta ne on löydettävä, jotta ne voidaan poistaa tai muuten käsitellä. Näin saadaan datasta puhtaampaa ja laadukkaampaa esimerkiksi tiedonlouhinta- tai koneoppimisalgoritmeja varten. [12, 2]

Poikkeamien osuus koko datajoukosta on pieni ja poikkeamat ovat huomattavasti harvinaisempia tapauksia normaaliin dataan verrattuna. Niiden määrittely tapahtuu aina verrattuna datajoukon muihin tapauksiin. Poikkeamien tunnistamisen menetelmät antavat lopputulokseksi yleensä joko binäärisen vastauksen (engl. label), luokitellen pisteen poikkeamaksi tai normaaliksi, tai poikkeavuusarvon (engl. anomaly score), jonka suuruus kuvaa pisteen poikkeavuutta muuhun aineistoon verrattuna [12]. Jotkin poikkeamien tunnistamisen tekniikat arvioivat itse datajoukon sisältämien poikkeamien määrän omien havaintojensa perusteella, mutta monet algoritmit vaativat lähtötiedokseen poikkeamien osuuden opetusdatasta. Poikkeavuusarvon antavat menetelmät mahdollistavat myös yksittäisten pisteiden poikkeavuuden tarkemman tarkastelun.

Käytössä olevasta datasta riippuen poikkeamien tunnistamisen mallit voidaan toteuttaa ohjatuilla, puoliohjatuilla tai ohjaamattomilla menetelmillä. Ohjatussa poikkeamien tunnistamisessa (engl. supervised anomaly detection) käytetään opetusdatana aineistoa, jossa on valmiiksi tiedossa ja merkittynä normaalit ja poikkeavat datapisteet. Merkittyjen datapisteiden avulla rakennetaan ennustava malli. Tätä mallia hyödyntävä menetelmä luokittelee tarkasteltavat pisteet normaaleiksi tai poikkeaviksi sen mukaan, kumpaa luokkaa piste enemmän muistuttaa. Puoliohjatussa tavassa (engl. semisupervised anomaly detection) mallille syötetään datajoukko, joka sisältää mallin normaalia käyttäytymistä ja luokkaa edustavia pisteitä. Tällöin opetusdatan avulla koulutetaan malli kuvaamaan datan normaalia käyttäytymistä ja tämän avulla pyritään tunnistamaan poikkeamat. Ohjaamattoman poikkeamien tunnistamisen (engl. unsupervised anomaly detection) tapoja käytetään tapauksissa, joissa datassa ei ole merkittynä valmiiksi poikkeavia ja

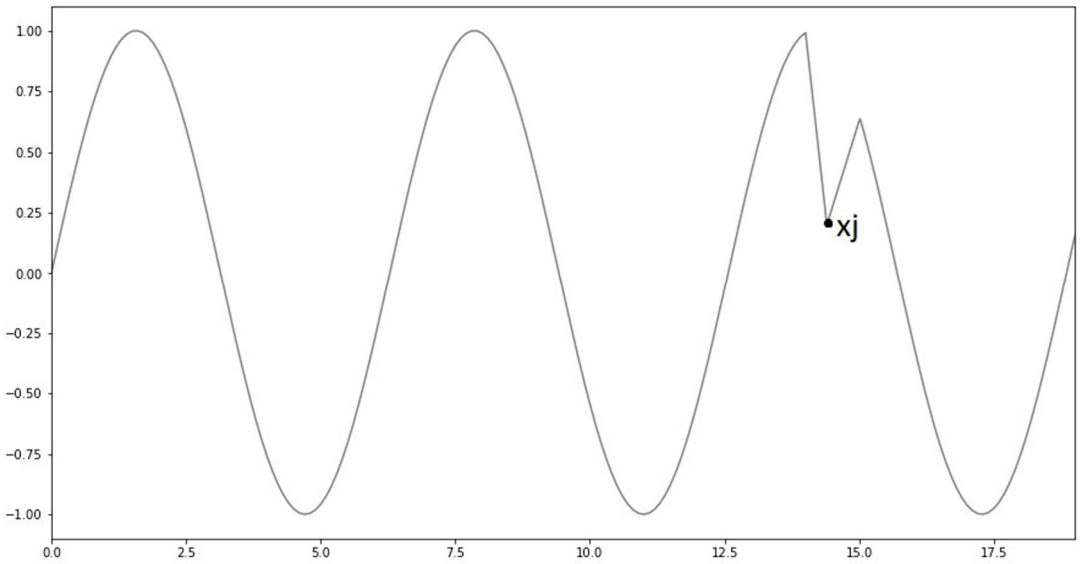
normaaleja datapisteitä. Algoritmien tehtävänä on mallintaa ja tunnistaa datapisteistä eniten muusta aineistosta poikkeavat pisteet. Ohjaamattoman oppimisen mallit olettavat poikkeamien olevan paljon harvinaisempia kuin normaalit tapaukset. [12, 14]

Poikkeamat voidaan jakaa kolmeen kategoriaan, joita ovat pistepoikkeamat, kontekstuaaliset poikkeamat ja kollektiiviset poikkeamat. Pistepoikkeamia (engl. point anomalies) edustavat pisteet tai pienet pistejoukot, jotka eroavat kaikista aineiston muista pisteistä ja eivät noudata mitään aineiston tuntemaa toimintatapaa. Esimerkkeinä kuvassa 2 on esitetty kaksiulotteisessa datajoukossa olevat yksittäiset pistepoikkeamat xi ja xo sekä joukko ni.



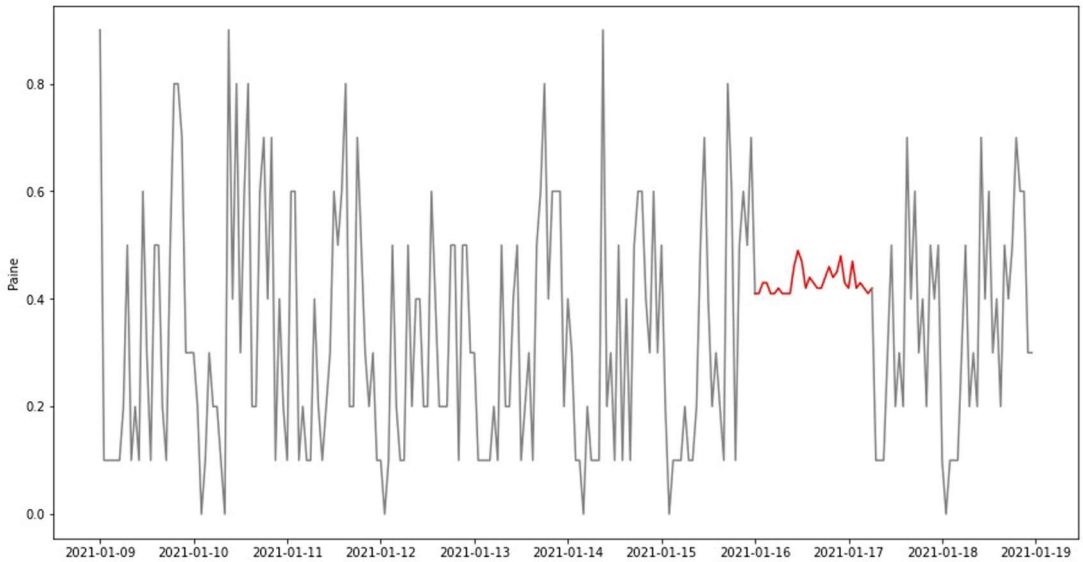
Kuva 2: *Pistepoikkeamat xi, xo ja ni esitettynä kaksiulotteisessa datajoukossa.*

Kontekstuaaliset poikkeamat (engl. contextual anomalies) edustavat pistepoikkeamien tavoin yksittäisiä pisteitä tai datajoukkoja, jotka tässä tapauksessa poikkeavat muusta datasta vain tässä lokaalissa kontekstissa. Globaalisti näitä pisteitä tai joukkoja ei luokiteltaisi poikkeamiksi tai epänormaaleiksi. Kuvassa 3 on esimerkki kontekstuaalisesta poikkeamasta xj, jonka y-arvo kuuluu datan globaaliin vaihteluväliin, mutta on selvästi poikkeama sen lokaalissa kontekstissa. [12]



Kuva 3: *Kontekstuaalinen poikkeama xj.*

Kollektiivinen poikkeama (engl. collective anomalies) koostuu toisiinsa liittyvän datan joukosta, joka yhdessä poikkeaa muusta datajoukosta. Poikkeamia ei voi laskea pistepoikkeamiksi, sillä peräkkäiset pisteet eivät ole poikkeavia verrattuna aikaisempiin pisteisiin. Pistepoikkeamia voi esiintyä kaikenlaisissa datajoukoissa, mutta kollektiivisia poikkeamia vain datajoukoissa, joissa yksittäiset datapisteet liittyvät toisiinsa, kuten esimerkiksi aikasarjadatassa. Kuvassa 4 on esimerkki kollektiivisesta poikkeamasta kuvattuna punaisella. [24]



Kuva 4: *Aikasarjadatassa merkitty kollektiivinen poikkeama punaisella.*

Lähteestä riippuen poikkeamien tunnistamisen tekniikat voidaan jakaa erilaisiin luokkiin ja Chandola et al. jakaa poikkeamien tekniikat kuuteen eri kategoriaan [12]. Näitä ovat luokitteluun, lähimpään naapuriin, klusterointiin, informaatioteoriaan ja spektriteoriaa

perustuvat sekä statistiset menetelmät. Menetelmän ja mahdollisen algoritmin valintaan ja käyttöön vaikuttavat taas käytössä oleva data ja käsiteltävä tapaus. Osa poikkeamien tunnistamisen tekniikoista on tarkoitettu tietyn tyyppisille tapauksille, kun taas toiset ovat yleispätevämpiä.

Työn tapauksessa käytetään koneoppimista apuna löytämään poikkeamia kohteen mittausdatasta. Koneoppimista käytettäessä hyvät algoritmit eivät pelkästään tarkastele ja analysoi yksittäisten mittausten muutosta, vaan osaavat löytää myös malleja ja mittausten suhteiden välisiä poikkeavuuksia. Poikkeamien tunnistamisen käyttö kunnonvalvonnassa olettaa mittausdatassa esiintyvien poikkeamien kertovan epätavallisesta tilanteesta kohteen toiminnassa, joka saattaa olla esimerkiksi jokin vika kohteessa tai rikkoutunut mittausanturi. Operaattorien ja kunnossapitohenkilökunnan tehtävänä on tarkastaa mallin löytämä poikkeama ja tehdä päätökset jatkotoimenpiteistä. [46]

2.7 Mahdollisia haasteita

Koneoppimista ja poikkeamien tunnistamista käytettäessä saattaa tulla vastaan joitakin hyvin yleisiä haasteita. Haasteet usein koskevat joko käytettävää dataa tai valittua poikkeamien tunnistamisen algoritmia. Nämä ongelmat on hyvä ottaa huomioon jo mallin luomisvaiheessa, jotta niiltä voitaisiin mahdollisuuksien mukaan välttyä.

Ensimmäisenä ongelmana voi olla datan määrä. Koneoppimisen näkökulmasta yleensä ajatellaan suuremman opetusdatamäärän saavan aikaan paremman mallin. Liian vähäinen opetusdatan määrä johtaa mallin suppeaan kokemukseen mahdollisista tapahtumista ja syy-seuraussuhteista, jolloin sen yleistyskyky ei pääse kehittymään tarpeeksi ja kohinan (engl. noise) vaikutus malliin on suurempaa. Kohinalla tarkoitetaan tässä tapauksessa satunnaisia häiriöitä mittaussignaaleissa, joita esiintyy väistämättä kaikessa mittausdatassa. Kohinaa voidaan yrittää vähentää datasta esimerkiksi erilaisilla signaalinkäsittelyn menetelmillä, mutta se vaatii tarkempaa tuntemusta kohinan luonteesta. Poikkeamien tunnistamisen tapauksessa kohina usein sattuu olemaan samanlaista kuin oikeat poikkeamat, jolloin kohina vaikuttaa poikkeamien tunnistustarkkuuteen ja sitä on vaikea poistaa [12].

Data voi myös olla tutkittavan tapauksen kannalta epärelevanttia, epätasapainoista tai muuten huonolaatuista. Datan tulisi kuvastaa tarpeeksi hyvin ja laajasti lopullisessa käyttökohteessa esiintyvää dataa, jotta malli tuntee kohteen normaalin toiminnan ja on oppinut huomioimaan esimerkiksi mahdolliset kausittaiset vaihtelut. Epärelevantteja piirteitä käytettäessä opetuksessa saattaa malli oppia näkemään oikeasti merkityksettömiä yhteyksiä piirteiden ja tulosten välillä ja tästä syystä johtaa vääriin johtopäätöksiin.

Poikkeamien tunnistamisessa oletetaan poikkeamien olevan huomattavasti harvinaisempia tapauksia normaaliin dataan verrattuna, joten jos poikkeamia on liikaa suhteessa muuhun dataan, ei poikkeamien tunnistamiseen erikoistuneet algoritmit välttämättä reagoi kaikkiin haluttuihin tapauksiin. [12, 17]

Jaottelu normaaleihin ja poikkeaviin tapauksiin ei aina ole selkeää. Normaalin ja epänormaalin käyttäytymisen välinen raja ei välttämättä ole kovin tarkka ja datassa esiintyvä kohina saattaa entisestään vaikeuttaa rajan asettamista. Nämä vaikuttavat erityisesti lähellä rajaa olevien tapausten luokitteluun ja luokittelun tarkkuuteen. Jotkin algoritmit vaativat tietoa opetusdatan sisältämien poikkeavuuksien osuudesta, mikä vaikuttaa suoraan rajanvetoon poikkeavuuden ja normaaliuden välillä. Mallin tekijällä ei välttämättä ole tarkkaa tai edes minkäänlaista tietoa poikkeamien osuudesta. [12, 31]

Osin epäselvien rajojen takia poikkeamien tunnistamisen voi sanoa olevan tasapainottelua virheellisten positiivisten (engl. false positive, FP) ja virheellisten negatiivisten (engl. false negative, FN) arvojen suhteen välillä. Tiukat mallit saavat aikaan enemmän FP-arvoja, eli virheellisesti poikkeamaksi merkittyjä normaaleja arvoja. Löysemmät mallit taas aiheuttavat enemmän FN-arvoja, jotka kuvaavat poikkeamia, joita malli ei ole tunnistanut. [22]

Poikkeamien tunnistamisen käyttäminen kunnonvalvonnassa tuo myös siihen liittyviä haasteita. Kunnonvalvottavilta kohteilta harvoin on saatavilla tarpeeksi luotettavaa ja kattavaa valmista dataa kohteiden todellisesta kunnosta kullakin hetkellä. Jos historiadataa on kohtuullinen määrä, voidaan sitä yrittää merkitä käsin. Suuremman datamäärän kanssa tämä on käytännössä mahdotonta ja taloudellisesti kannattamatonta. Merkitsemättömän datan käyttö koneoppimisessa tuo joitakin rajoituksia ja huomioitavia asioita kunnonvalvonnan kannalta. Poikkeamien tunnistamiseen tarkoitetut mallit eivät suoraan voi todeta, että järjestelmässä on jotain poikkeavaa, vaan ne tutkivat siitä saadun datan käyttäytymistä ja etsivät poikkeavuuksia tarkkailemistaan mittausjoukoista [31]. Kaikki poikkeamat datassa eivät aiheudu oikeista vioista tai häiriöistä, vaan esimerkiksi sateet tai muut olosuhteiden tai käytön vaihtelut saattavat aiheuttaa normaalista poikkeavia arvoja tai vaihtelua pumppaamon mittausdatassa. Poikkeamien tunnistamisen tavat eivät myöskään kerro poikkeaman syntymisen juurisyitä. Mallin osuus kunnonvalvonnasta on vain osoittaa tilanteet, mistä operaattorit saattaisivat olla kiinnostuneita. Poikkeamien tunnistamisen malli ei siis ole hyvä lopullisen kunnossapitopäätöksen tekijä, vaan se vaatii ihmisen tekemään lopullisen päätöksen.

Työssä mallit käsittelevät vain valmiiksi kerättyä dataa, mutta lopullisessa käyttökohteessa niitä on tarkoitus käyttää suoraan tarkasteltavasta pumppaamosta saatavan

reaaliaikaisen mittausdatan tarkastelemiseen. Tällöin malleja voidaan käyttää pumppaamoiden tilojen arvioinnissa ja niiden avulla voidaan tunnistaa vikatilassa olevia pumppuja. Koneoppimisen käyttäminen reaaliaikaisen järjestelmän työkaluna tuo kuitenkin omia haasteita. Oikeaa järjestelmää analysoitaessa on hyvä muistaa, että sen toiminta ja ympäristö saattavat muuttua ajan kuluessa. Mallia ja sen toimivuutta on hyvä tarkastella säännöllisin väliajoin ja suorittaa mallin opettaminen uudelleen tuoreemman datan avulla. Pienetkin muutokset järjestelmässä saattavat vaikuttaa radikaalisti mittauksiin ja sitä kautta mallin toimintaan. Tehtäessä muutoksia järjestelmään olisi myös hyvä huomioida niiden vaikutus poikkeamien tunnistamisen malliin ja sen toimintaan.

3. POIKKEAMIEN TUNNISTAMISEN METODIT

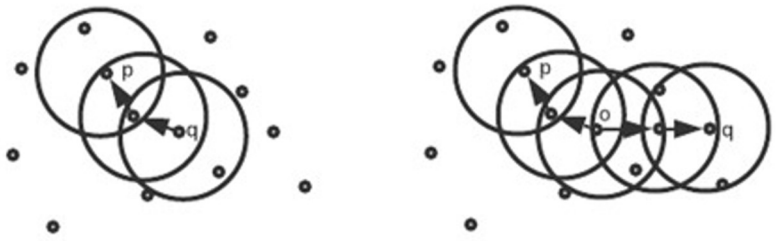
3.1 DBSCAN-klusterointi

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) on pisteiden esiintymistiheyteen perustuva klusterointimenetelmä. DBSCAN jakaa pisteet mielivaltaisen muotoisiin klustereihin sen perusteella, sijaitsevatko ne pisteiden keskittymässä vai erillään muista tarkasteltavista pisteistä. Toimintatapansa ansiosta menetelmä sopii kohinaisen datan klusterointiin sekä poikkeamien tunnistamiseen. [18]

Algoritmille annetaan parametreina minPts, joka määrittää kuinka monta pistettä ydinpisteen naapurustoon tulee kuulua sekä epsilon ε, joka määrittelee maksimin kahden pisteen väliselle etäisyydelle, jotta ne voivat kuulua toistensa naapurustoon. Etäisyyden laskemiseen voidaan käyttää mitä tahansa etäisyyden laskemiseen tarkoitettua kaavaa, mutta yleisimmin käytetty on euklidinen etäisyys (engl. Euclidean distance). Parametrit määrittelevät klustereille vaadittavan pistejoukon tiheyden. Suuremmalla ε:n arvolla useampi piste kuuluu klustereihin ja pienemmän minPts arvon seurauksena klusterin muodostumiseen vaaditaan pienempi määrä pisteitä, jolloin useampi piste luokitellaan normaaliksi, joten poikkeamien määrä vähenee. [69, 18]

Kukin datapiste luokitellaan joko ydin- (engl. core point) tai reunapisteeksi (engl. border point) tai poikkeavuudeksi. Tarkasteltava piste on ydinpiste, jos sen ε-naapurustossa on vähintään minPts määrä muita datapisteitä. Reunapisteitä ovat muut ydinpisteen naapurustossa sijaitsevat pisteet. [18]

Klustereiden määrittelyyn liittyvät myös tiheys-saavutettavuus (engl. density-reachability) ja tiheys-liitännäisyys (engl. density-connectivity). Kuvassa 5 on vasemmalla esitetty pisteen p tiheys-saavutettavuus pisteestä q ja oikealla pisteiden p ja q tiheys-liitännäisyys pisteen o kautta.



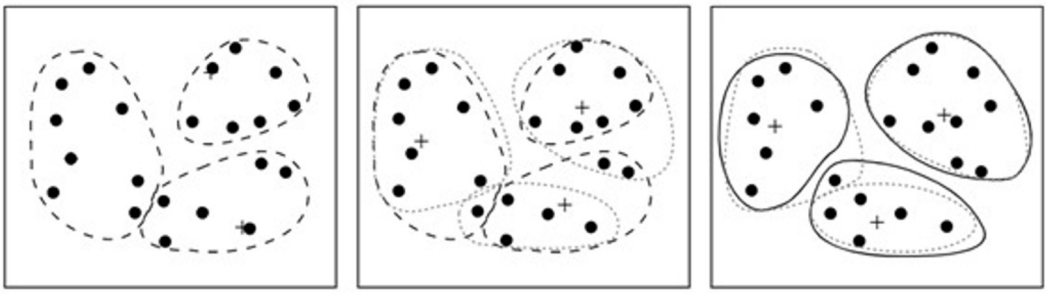
Kuva 5: *Tiheys-saavutettavuus ja tiheys-liitännäisyys.*

Pisteiden tiheys-saavutettavuus määräytyy sen mukaan, voidaanko niiden välille muodostaa ketju pisteistä, joista jokainen piste kuuluu edeltävän pisteen naapurustoon ja edeltävä piste on määritelty ydinpisteeksi. Tiheys-liitännäisyys toteutuu kahden pisteen välillä silloin, kun on olemassa piste o, jonka kanssa molemmat tutkittavat pisteet p ja q ovat tiheys-saavutettavia. [18]

Yhden klusterin muodostaa suurin mahdollinen määrä ydin- ja reunapisteitä, jotka ovat tiheys-liitännäisiä toisiinsa nähden. Poikkeamia tai kohinaksi määriteltyjä pisteitä ovat sellaiset pisteet, jotka eivät kuulu mihinkään klusteriin. Klustereiden lukumäärää ei tarvitse määritellä etukäteen, vaan menetelmä määrittelee dataan sopivimman klustereiden määrän.

3.2 K-means klusterointi

Klusterointimenetelmät etsivät datasta luonnollisia ryhmittymiä (engl. natural groups) datajoukon ominaisuuksien ja piirteiden perusteella. K-means klusteroinnissa datajoukko jaetaan valittuun määrään *k* ryhmiä eli klustereita samankaltaisuuden perusteella. Algoritmi valitsee datajoukosta satunnaisesti *k* määrän datapisteitä ensimmäisiksi klustereiden keskipisteiksi eli sentroideiksi (engl. centroid). Jäljelle jääneet pisteet jaetaan klustereihin niiden ja valittujen sentroidien välisten etäisyyksien mukaan niin, että pisteen ja sentroidin etäisyys on mahdollisimman pieni. Tämän jälkeen jokaisen klusterin uusi sentroidi lasketaan ja se valitaan klusterin sentroidiksi. Mallin lopullinen muoto haetaan iteratiivisesti, eli opetusta jatketaan, kunnes klusterit eivät enää muutu tai on suoritettu ennalta määrätty maksimimäärä iterointeja. Kuvassa 6 on esitetty k-means klusteroinnin iterointi. [23, 45]



Kuva 6: *K-means klusteroinnin iterointi* [23].

Kuvan vasemmassa laidassa on esitetty alkutilanne ensimmäisen klustereiden jaon jälkeen. Sentroidit ovat kuvassa esitetty merkillä +. Keskellä on uudet klusterit merkitty erilaisella pisteviivalla ja oikealla lopulliset iteroinnin päätöksenä saadut klusterit yhtenäisellä viivalla.

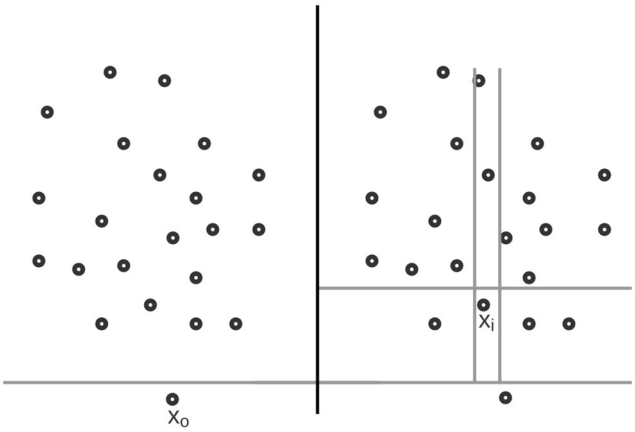
Iteroinnista apuna käytetään usein jäännösneliösummaa SSE (engl. sum of squared errors), jossa lasketaan kaikkien klustereiden sentroidien ja muiden klustereissa olevien pisteiden välisten etäisyyksien neliöiden summa käyttäen kaavaa 3.1.

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2 \qquad (3.1)$$

Yksi yleisimmin käytetyistä laskukaavoista klustereiden pisteiden *p* ja sentroidien c_i etäisyyden $dist(p,c_i)$ laskemiseen on euklidinen etäisyys, mutta myös esimerkiksi korttelietäisyyttä (engl. Manhattan distance) tai kosinietäisyyttä (engl. Cosine distance) voidaan käyttää. [23, 45]

3.3 Isolation Forest

Isolation Forestin (iForest) toiminta perustuu yksittäisten tapausten erottelemiseen tai eristämiseen muusta datajoukosta. Poikkeamat eroavat normaalista datasta ja niiden määrä on yleensä vähäinen, joten ne ovat alttiimpia eristämiselle. iForestissa havaintoja erotellaan toisistaan tekemällä jakoja datajoukkoon. Poikkeavuuksien määrittämiseen käytetään polun pituutta. Alkiokohtainen polun pituus määrittyy alkion eristämiseen vaadittavien jakojen lukumäärän mukaan. Lyhyempi polku kuvastaa datan olevan suuremmalla todennäköisyydellä poikkeama. Kuvasta 7 näkee, että normaali datapiste x_i vaatii enemmän jakoja tullakseen eristetyksi, kuin poikkeama x_o. [42]



Kuva 7: *Poikkeavan datapisteen x_o eristäminen vaatii vähemmän jakoja kuin normaalin datapisteen x_i eristäminen.*

iForest koostuu binääristen eristyspuiden joukosta (engl. isolation tree, iTree). Puut generoidaan satunnaisesti opetusdatan osajoukoista ja puun solmut kuvaavat satunnaisesti tehtyjä jakoja. Jokaisen solmun kohdalla valitaan satunnaisesti yksi datajoukon piirre (engl. attribute), joka jaetaan satunnaisesti valitun jakoarvon perusteella. Puun

kasvatus jatkuu, kunnes kaikki käytetyn datajoukon pisteet on saatu eristettyä toisistaan. Lopulliseksi kunkin pisteen polun pituudeksi tulee sen eristystä kuvaavien polkujen keskiarvo. [42]

Polun pituudesta lasketaan pisteelle oma poikkeavuusarvo, jonka suuruuden perusteella piste lopulta luokitellaan poikkeamaksi tai normaaliksi. Koko pistejoukosta valitaan opetukseen *ψ* kokoinen osajoukko, johon kuuluvan pisteen *x* poikkeavuusarvo *s(x,ψ)* lasketaan kaavalla 3.2. [42]

$$s(x,\psi) = 2^{-\frac{E(h(x))}{c(\psi)}} \tag{3.2}$$

Yksittäisten polkujen pituuksista *h(x)* lasketaan niille keskiarvo *E(h(x))*, joka normalisoidaan jakamalla epäonnistuneiden hakujen polkujen pituuksien keskiarvolla *c(ψ)*. Yksittäisen eristyspuun rakenne vastaa binäärisen hakupuun (engl. binary search tree, BST) rakennetta, joten tässä tapauksessa käytämme binääristen hakupuiden epäonnistuneiden hakujen pituuden määritelmää. Preiss [54] käyttää binäärisen hakupuun epäonnistuneen haun keskimääräiselle polun pituudelle kaavaa 3.3.

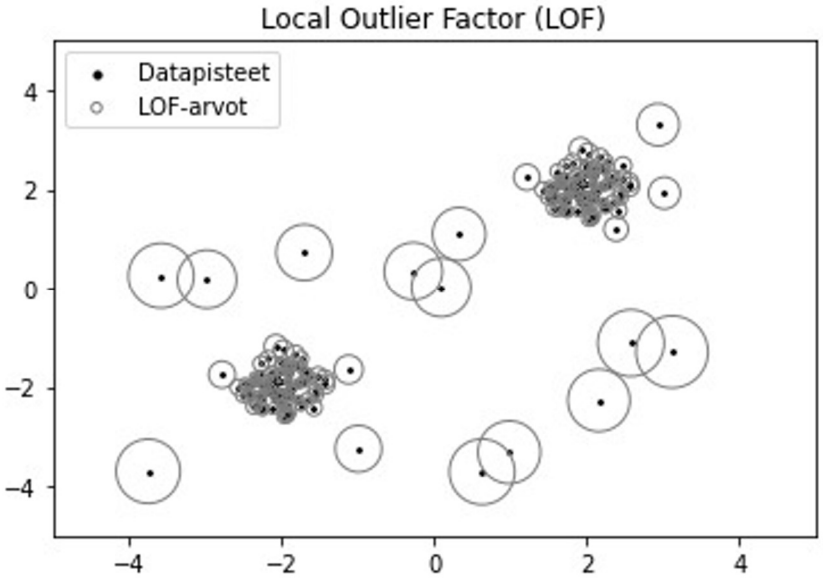
$$c(\psi) = \begin{cases} 2H(\psi-1) - 2(\psi-1)/\psi, & kun\ \psi > 2 \\ 1, & kun\ \psi = 2 \\ 0, & muulloin. \end{cases} \tag{3.3}$$

Kaavan harmoninen numero *H* voidaan laskea datapisteiden määrän *i* perusteella *H(i)=ln(i)+0,5772156649.*

Pisteet, joiden poikkeavuusarvo on lähellä arvoa 1 ovat suurella todennäköisyydellä poikkeamia ja reilusti alle 0,5 jäävät pisteet ovat todennäköisesti normaaleja arvoja. Jos tutkittavan datajoukon kaikkien pisteiden poikkeavuusarvot ovat lähellä arvoa 0,5, ei joukossa ole selkeitä poikkeamia. [41, 42]

3.4 Local outlier factor

Local outlier factor (LOF) perustuu pisteiden tiheyden laskemiseen ja yksittäisen pisteen tiheyden vertailuun pisteen k:n lähimmän naapurin kanssa. Pisteen LOF-arvo määrittyy sen naapureiden mukaan, joten LOF on toimiva tapa lokaalien poikkeamien tunnistamiseen. Muista poikkeavilla pisteillä on niiden lokaaleita naapureita matalampi tiheys, jolloin niiden LOF saa suuremman arvon. Local outlier factor ei luokittele pisteitä binäärisesti normaaleihin ja poikkeaviin, vaan määrittelee jokaiselle pisteelle sen poikkeavuuden asteen. Kuvassa 8 esitettyjen pisteitä ympäröivien renkaiden halkaisijat kuvastavat pisteiden LOF-arvon suuruutta. [11]



Kuva 8: *Esimerkki LOF:ista satunnaisella datalla* [74].

Pisteen *o* LOF-arvo määritetään laskemalla sen keskiarvoinen etäisyys k:n lähimmän naapurin kanssa ja etäisyys normalisoidaan laskemalla näiden k:n lähimmän pisteen keskiarvoinen etäisisyys niiden omista k lähimmästä naapurista [1]. Tutkittavien naapureiden määrä k on itse valittavissa. Pisteen o LOF-arvon laskeminen tapahtuu Breunig et al. [11] määrittelemien seuraavien askeleiden mukaan:

1. Pisteelle *o* tulee laskea K-etäisyys (engl. K-distance) $dist_k(o)$, joka on o:n ja sen k:nnen naapurin etäisyys.
2. Pisteen *o* K-etäisyydellä oleva naapurusto $N_k(o)$ koostuu kaikista siitä korkeintaan K-etäisyydellä olevista pisteistä. Näitä pisteitä voi siis olla enemmän kuin k, jos useammalla pisteellä on sama etäisyys (K-etäisyys) pisteestä *o*.
3. Pisteille *o* ja *o'* lasketaan saavutettavuusetäisyys (engl. reachability distance, RD) kaavalla 3.4

$$reach\_dist_k(o', o) = \max\{dist_k(o), dist(o', o)\} \qquad (3.4)$$

Naapurustoon $N_k(o)$ kuuluville saavutettavuusetäisyydeksi tulee siis K-etäisyys ja kauempana oleville pisteiden välinen todellinen etäisyys $dist(o',o)$.

4. Paikallinen saavutettavuustiheys (engl. local reachability density, lrd) pisteen *o* k:lle lähimmälle pisteelle saadaan käänteisesti kaikkien pisteiden saavutettavuusetäisyyksien keskiarvosta kaavalla (3.5)

$$lrd_k(o) = 1/\frac{\sum_{o' \in N_k(o)} reach\_dist_k(o', o)}{|N_k(o)|} \quad (3.5)$$

5. Lopulta pisteelle *o* voidaan laskea LOF-arvo laskemalla pisteen ja sen lähimpien naapureiden paikallisten saavutettavuustiheyksien keskimääräinen suhde kaavalla 3.6

$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrd(o')}{lrd(o)}}{|N_k(o)|} \quad (3.6)$$

Kaavan mukaan tarkasteltavan pisteen ollessa poikkeama, sen tiheys naapureiden tiheyksiin verrattuna on pienempi, jolloin sen LOF>>1. Tyypillisesti voidaan arvon LOF≈1 tarkoittavan normaalia pistettä. [11]

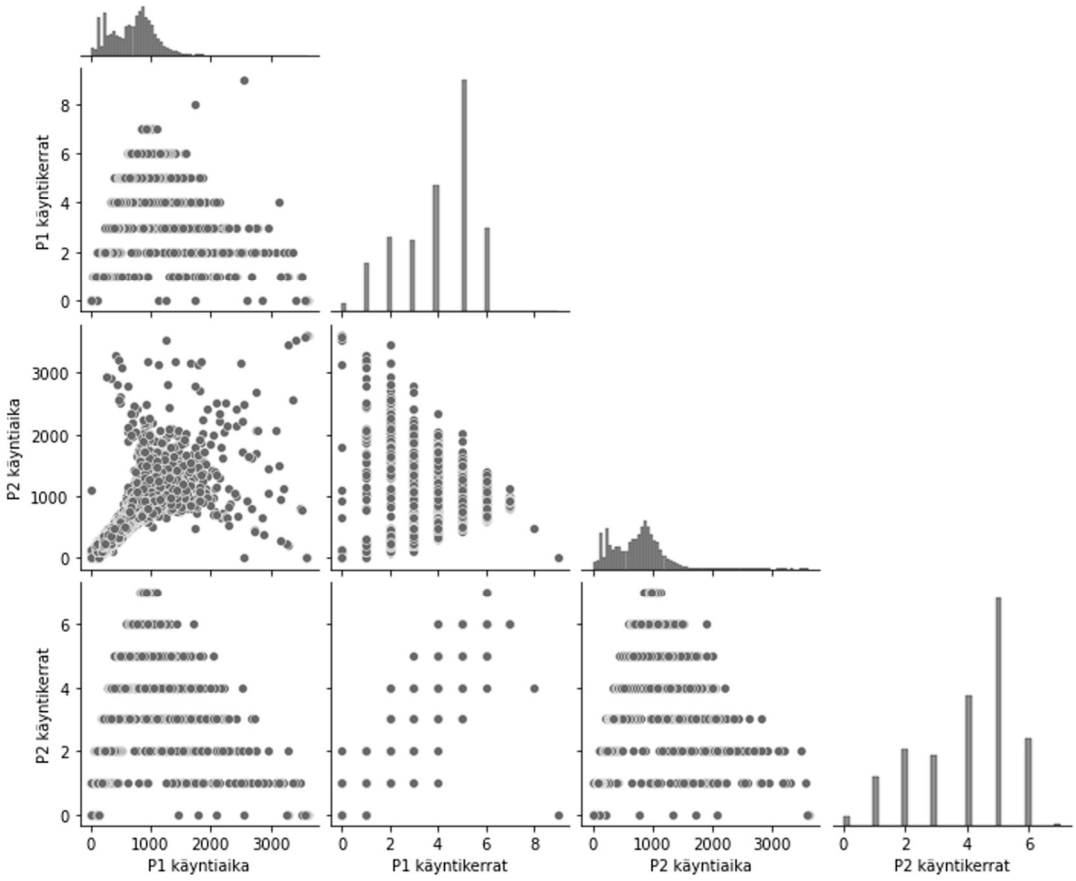
3.5 Apumetodit

3.5.1 Eksploratiivinen data-analyysi

Käytettävän datan tunteminen ja ymmärtäminen on tärkeää, jotta käsiteltävää ongelmaa varten pystytään luomaan toimiva malli järkevällä tavalla. Eksploratiivinen data-analyysi (engl. exploratory data analysis, EDA) tarjoaa työkaluja ja tapoja esimerkiksi rakenteiden, kuvioiden, ryhmittymien ja poikkeamien löytämiseen isosta datajoukosta sekä mahdollisten hypoteesien ja oletusten testaamiseen. EDA:n avulla pyritään selvittämään sisältääkö tutkittu data hyödyllistä tietoa ja miten eri muuttujat vaikuttavat systeemiin ja sen toimintaan. EDA:n käyttämisen tarkoituksena voi olla datan yhteenveto, analysointi ja visualisointi. EDA:a voidaan käyttää yhtenä osana datatieteen prosessia ja sen rooli osana datatieteen prosessia on esitetty kappaleesta 2.5 löytyvässä kuvassa 1. [64, 4]

Pistekaavio (engl. scatter plot) on EDA:ssa käytetty visuaalinen työkalu avustamaan datan jakautumisen esittämisessä sekä rakenteiden löytämisessä. Yhdessä kuvaajassa voidaan esittää helposti vain kaksi tai kolme ulottuvuutta, joten yksittäiset kuvaajat eivät ole kovin tehokas tapa visualisoimaan korkeadimensioista dataa. Pistekaaviomatriisi on tapa kerätä useampia dimensioita sisältävien datajoukkojen kaikkien muuttujaparien kaksiulotteiset pistekaaviot yhteen ja esittää ne matriisimuodossa. Jokaiselle riville ja sarakkeeseen on kerätty aina tietyn piirteen taulukot. Matriisin diagonaalilla voidaan esittää esimerkiksi muuttujien jakaumat tiheyskaavioiden ja histogrammien avulla. Lisäksi eri luokkiin kuuluvia pisteitä voidaan esittää esimerkiksi eri väreillä havainnollistamaan eri

luokkien yksilöiden sijoittumista kaksiulotteisessa kaaviossa. Kuvassa 9 on esitetty eräs pistekaaviomatriisi, jossa on koottuna neljän eri piirteen väliset pistekaaviot 4x4 matriisiin. Pistekaaviomatriisista voidaan kuvan tavoin jättää diagonaalin yläpuolinen osuus pois, sillä samat pistekaaviot on esitetty diagonaalin alapuolella.



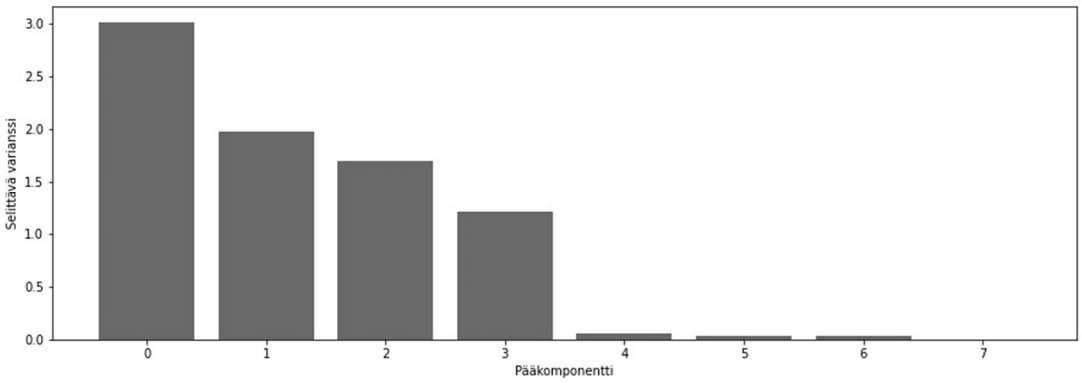
Kuva 9: *Pistekaaviomatriisi.*

Muuttujaparien pistekaaviot ja niistä koottu pistekaaviomatriisi eivät välttämättä suoraan kerro eri piirteiden välisistä riippuvuuksista, mutta se tarjoaa visuaalisen työkalun esimerkiksi erilaisten analyysien toteuttamiseen, piirteiden välisten korrelaatioiden löytämiseen sekä esimerkiksi luokittelussa tai poikkeamien tunnistamisessa saatujen tulosten analysoimiseen.

Datan piirteiden määrän kasvaessa matriisi kuitenkin kasvaa ja sen tulkittavuus hankaloituu. Visuaalisointiin liittyvien ongelmien lisäksi korkeadimensioinen data voi aiheuttaa ongelmia etäisyyksiin perustuvien menetelmien käyttämisessä. Piirteiden lisääminen yleensä parantaa luokittelijoiden tarkkuutta vain tiettyyn määrään asti. Tämän jälkeen piirreavaruus alkaa harvenemaan ja etäisyysmittojen eroavaisuudet heiketä, jolloin dataa ei pystytä erottelemaan yhtä tarkasti. Dimensioiden lisäämisen tuomia ongelmia kutsutaan dimensionaalisuuden kiroukseksi (engl. curse of dimensionality).

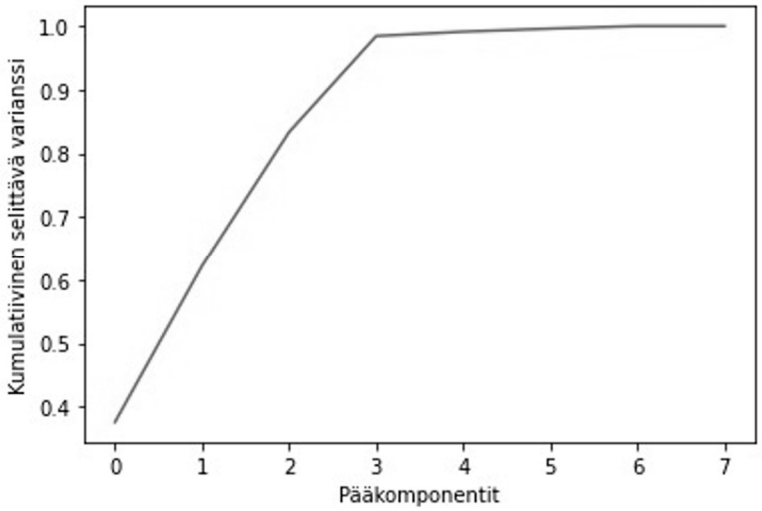
EDA tarjoaa erilaisia dimensioiden vähentämisen menetelmiä auttamaan dimensioiden korkean määrän tuomiin ongelmiin. Dimensioiden vähentämisen keinojen avulla pyritään kuvaamaan data vähemmillä piirteillä säilyttäen mahdollisimman suuri osa alkuperäisestä informaatiosta. Dimensioiden vähentäminen yksinkertaistaa malleja, pienentää mallien luomisen ja laskennan vaatimaa aikaa ja kustannuksia.

Pääkomponenttianalyysi (engl. principal component analysis, PCA) on yksi EDA:n dimensioiden vähentämisen menetelmistä, jossa suuresta joukosta muuttujia pyritään muodostamaan pienempi joukko kuvaamaan alkuperäisessä datassa esiintyvää vaihtelua. Dimensioiden vähentämisen lisäksi sitä voidaan käyttää apuna dataan sekä pisteiden ja piirteiden välisiin suhteisiin tutustumisessa sekä esimerkiksi jonkin mallin antamien tulosten analysoimisessa. PCA on lineaarinen menetelmä ja se perustuu toistensa kanssa korreloimattomien ortogonaalisten kantavektorien eli pääkomponenttien löytämiseen. Pääkomponentit pyritään muodostamaan minimoiden uudelleen rakennetun datan jäännösneliösumma. Ensimmäinen pääkomponentti pyrkii kuvaamaan mahdollisimman suuren osan aineiston vaihtelusta ja sitä seuraavat aina jäljelle jäävästä vaihtelusta. PCA järjestelee datan vaihtelua kuvaavat komponentit järjestykseen. Kuvassa 10 on esitetty eräästä työn pumppaamon mittausdatasta muodostettujen pääkomponenttien sisältämät selittävät varianssit. [28, 32, 9]



Kuva 10: *Selittävän varianssin jakautuminen pääkomponenttien välillä.*

PCA:n luomista pääkomponenteista on valittava käytettävien pääkomponenttien määrä yksittäisten komponenttien merkittävyyden perusteella. Pääkomponenteista voidaan hylätä vain vähän informaatiota lisäävät komponentit. Päätös valittavasta komponenttien määrästä voidaan tehdä kumulatiivisen selittävän varianssin avulla. Kuvassa 11 on esitetty edellisen kuvan pääkomponenttien kumulatiivinen selittävä varianssi.



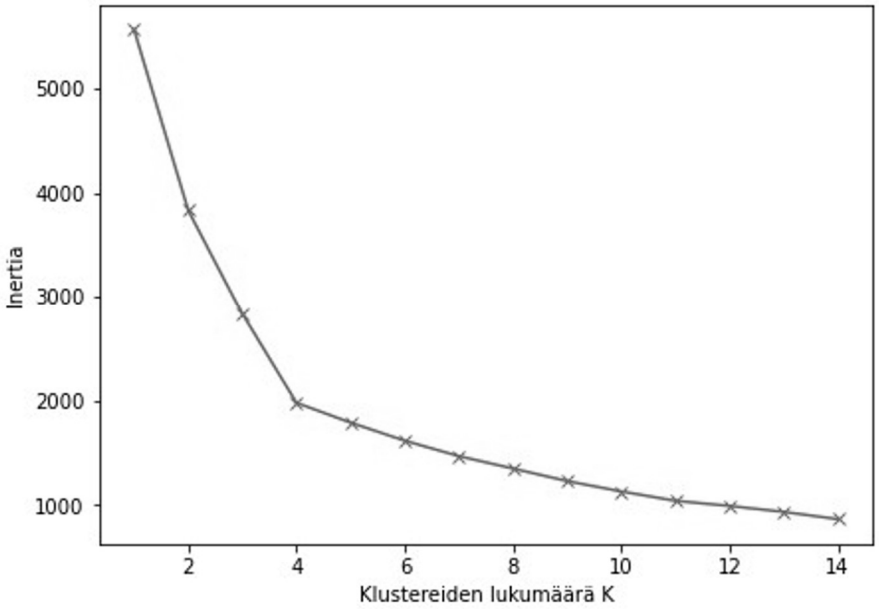
Kuva 11: *Pääkomponenttien kumulatiivinen selittävä varianssi.*

Kuvasta nähdään, että neljä ensimmäistä pääkomponenttia sisältävät lähes kaiken pääkomponenttien varianssista. Loput pääkomponenteista voidaan jättää käyttämättä menettämättä merkittävästi alkuperäisen datan sisältämää informaatiota.

Monidimensioisen datan tiedetään voivan aiheuttaa ongelmia erityisesti klusterointimenetelmiä käytettäessä [67] ja PCA onkin yleinen apuväline vähentämään dimensioiden määrän tuomia ongelmia. PCA:n sekä K-means klusteroinnin käyttöä yhdessä ovat käsitelleet artikkeleissaan muun muassa Honda et al. [27] sekä Ding ja He [15].

3.5.2 Kyynärpäämetodi

Joissain tapauksissa koneoppimisella ratkaistava ongelma määrittelee tietyt vaatimukset joillekin parametreille, kuten klustereiden määrälle, tai optimaalisista parametreista on jotain muuta tietoa etukäteen. Tällöin parametrit voidaan valita näiden vaatimusten ja tietojen perusteella, mutta näiden puuttuessa voidaan esimerkiksi klusterointimenetelmissä käyttää apuna kyynärpäämetodia sopivien parametrien valitsemiseen. Kyynärpäämetodin avulla voidaan esimerkiksi K-means klusteroinnissa löytää optimaalinen arvo klustereiden määrälle K laskemalla inertian arvo eri klustereiden määrällä [65, 19]. Inertia lasketaan klusterin sentroidin ja klusterin sisältämien pisteiden etäisyyksien neliösummana SSE. Kyynärpääkuvaajan y-akselilla on esitettynä inertia ja x-akselilla käytettyjen klustereiden määrä. Kuvassa 12 on esitetty erään tapauksen kyynärpääkuvaaja.



Kuva 12: *Kyynärpäämetodia varten piirretty kuvaaja.*

Jyrkempi käyrä kuvaa klustereiden määrän lisäämisen suurempaa vaikutusta inertian pienentymiseen. Kyynärpäämetodi perustuu siihen, että tietyn klustereiden määrän jälkeen ei niiden määrän lisääminen enää merkittävästi pienennä inertiaa. Tämä piste näkyy kuvaajassa selkeänä taitoksena eli kyynärpäänä. Useamman klusterin käyttäminen lisää mallin koulutusaikaa parantamatta mallin toimintaa merkittävästi. Kuvan 12 tapauksessa kyynärpääkuvaajan mukaan optimaalinen klustereiden määrä on 4.

4. TOTEUTUS

Tämä kappale esittelee, miten edellisten kappaleiden teorioita pyrittiin hyödyntämään johdannossa esitetyn ongelman ratkaisemiseksi ja asetettuihin tutkimuskysymyksiin vastaamiseksi. Työssä tuotettiin koneoppimisen avulla malleja, joiden tavoitteena oli tunnistaa pumppaamoilta saadusta mittausdatasta poikkeamia. Mallit voisivat toimia apuvälineinä toteamaan pumppaamon poikkeuksellista käyttäytymistä ja mahdollisia häiriötilanteita. Malleista pyrittiin tekemään mahdollisimman yleiskäyttöiset ja helposti muokattavat. Työssä tuotetut koodit pyrittiin kommentoimaan selkeästi sekä riittävästi. Tällöin työkalu olisi mahdollisimman pienellä vaivalla kehitettävissä sekä muunnettavissa erilaisten opetusdatajoukkojen käyttämistä varten.

Suurin osa opetukseen käytettävän datan esikäsittelystä sekä käytettävien piirteiden valinta tulee suorittaa erikseen ennen mallin luomista ja tiedoston on oltava rakenteeltaan määritellyn muotoinen, jotta muodostettu koodi käsittelee sen oikein. Pythoniin määritellään käytettävä datatiedosto ja parametrit, joiden perusteella koodi opettaa mallit ja muodostaa jokaiselle mallille erilliset DataFrame-datarakenteet, jotka sisältävät löydetyt poikkeavat pisteet. Saatujen tulosten perusteella luodaan graafisia esityksiä havainnoimaan mallien löytämiä poikkeamia sekä vertailemaan eri mallien saamia tuloksia. Työkalun koodissa on erilaisten mallien luomista ja testausta varten omia käytettävään dataan ja mallien luomiseen liittyviä parametreja. Nämä omat parametrit määrittelevät esimerkiksi käytettävän laitoksen, tutkittavan ajanjakson tai osuuden, tai määrittelevät kuvien tulostuksen. Näin pyrittiin mahdollistamaan erilaisten mallien luominen ja testien tekeminen peräkkäin mahdollisimman vähillä ja yksinkertaisilla muutoksilla.

Algoritmin valinnan lisäksi tärkeimpiä asioita mallin toimimisen kannalta ovat käytössä oleva data ja sen esikäsittely. Malleja luotaessa niiden toimintaan ja opettamisen kestoon voidaan vaikuttaa kuitenkin vielä esimerkiksi poikkeamien osuuden ja erilaisten parametrien kautta. Käytetystä tekniikasta ja mallista riippuen, voidaan sille valita esimerkiksi käytettävä etäisyyden laskukaava, klustereiden tai luokkien määrä sekä laskennassa huomioon otettavien naapureiden määrä. Joidenkin parametrien osalta voidaan sopivan arvon valitsemiseen käyttää apuna yleisiä dataan, piirteisiin, niiden määrään tai käytettävään kohteeseen liittyviä ohjearvoja tai kyseisen parametrin optimointiin sopivaa apumetodia.

4.1 Tutkittavat pumppaamot

Työssä käsiteltävien pumppaamoiden automaatiojärjestelmät keräävät useita niiden toimintaa kuvastavia mittauksia, kuten virtaus, lämpötila, pinnankorkeus, pumpun moottorin virta sekä pumpun käyntiaika ja käyntikerrat. Mittausdatat tuodaan yhteiseen Instawahti-raportointityökaluun, josta operaattorit ja valvomotyöntekijät voivat tarkastella mittauksia ja niiden historiatietoja. Pumppaamoiden valvonta tapahtuu keskitetysti keskusvalvomoilta.

Käytössä ollut data oli noin 150 jätevedenpumppaamolta, joista jokaisesta oli saatavilla eri mittauksia. Pumppaamoista valittiin tutkittavaksi 7 kriittisimmäksi luokiteltua pumppaamoa. Valitut pumppaamot sekä niiden pumppujen ja mittausten lukumäärät on kerätty taulukkoon 1.

Taulukko 1: *Työhön valitut pumppaamot.*

Laitos	Pumppujen lkm.	Mittausten lkm.
Roope	4	18
Pähkinäkallio	4	17
Kankkula	4	19
Lähtösaha	2	22
Taivallammi	2	15
Suntinmäki	3	20
Saarenmaa	3	20

Vaikka kaikkien pumppaamoiden perustoiminta on samanlaista, tulee jokaista pumppaamoa käsitellä yksitellen ja pumppaamon kunnonvalvonnassa tulee käyttää vain sen oman historiadatan avulla opetettua poikkeamien tunnistamisen mallia. Valituissa pumppaamoissa on kussakin 2–4 pumppua ja 15–22 mittausta. Kaikki näistä mittauksista eivät kuitenkaan ole oleellisia pumppaamoiden häiriöiden kannalta, joten datan esikäsittelyn yhteydessä tulee tutustua tarkemmin valittaviin mittauksiin.

4.2 Käytetyt ohjelmistot

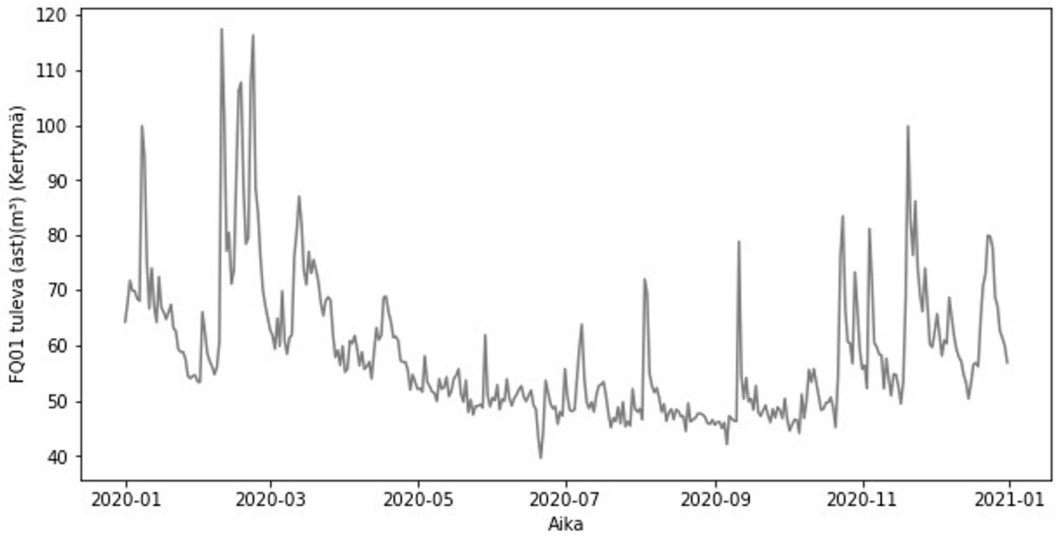
Työssä tuotetut mallit on rakennettu Python-ohjelmointikielellä hyödyntäen sen Pandas ja scikit-learn (sklearn) -kirjastoja. Pandas-kirjasto tarjoaa helppokäyttöisiä ja tehokkaita työkaluja datan käsittelemiseen, kuten datarakenteiden muodostamiseen,

muokkaamiseen ja analysointiin [49]. Sklearn-kirjasto taas sisältää yleisimmät koneoppimisalgoritmit ja muita koneoppimisessa käytettäviä työkaluja [52]. Kirjastot ja Spyder-ohjelmointiympäristö saatiin Anaconda-jakelupaketin kautta. Datan tarkasteluun ja käsittelyyn käytettiin myös Microsoft Excel -sovellusta.

4.3 Käytettävä data

Pumppaamoilla on käytössä eri suureita mittaavia antureita, joiden avulla voidaan valvoa pumppauksen tilaa. Poikkeamien tunnistamisen mallien tekemiseen käytettiin näiden mittausten historiadataa. Data saatiin Instawahti-järjestelmästä, jossa on tallennettuna jokaisesta mittauksesta joko tuntikeskiarvot tai tuntikohtaiset kertymät. Historiadata saatiin järjestelmästä CSV-muodossa, joka muunnettiin Excelissä helposti käytettävään xml-muotoon. Excelissä suoritettiin datan manuaalinen läpikäynti, järjestely ja korjaus ennen mittausdatan lataamista ohjelmointiympäristöön. Taulukossa oli riveittäin jokaisen tunnin aikaleima eli jokainen yksittäinen piste ja sarakkeissa aikaleimaa vastaavan tunnin mittaukset eli piirteet. Ensimmäisessä sarakkeessa tuli siis olla pisteitä vastaavat aikaleimat ja ensimmäisellä rivillä vapaavalintaisesti kunkin piirteen nimi tai muu tunnus.

Tutkittavaksi ajanjaksoksi valittiin 1.1.-31.12.2020. Valitsemalla tarkasteltavaksi ajanjaksoksi kokonainen vuosi saadaan mallin luomisessa huomioitua pumppaamoiden toiminnan ja datan normaali vaihtelu ja kausiluonteisuus. Eri vuodenajat ja sääolosuhteet, kuten sateiset kaudet, vaikuttavat pumppaamoiden kuormitukseen ja pumppaustarpeeseen. Kuvassa 13 on esitetty Kankkulan pumppaamolle tulevan jäteveden määrän päiväkohtaiset keskiarvot.



Kuva 13: *Kankkulan pumppaamolle tulevan jäteveden päiväkeskiarvot.*

Kuvaajasta voidaan huomata, että vuoden keskivaiheilla tulevan jäteveden määrä on tasaisempaa ja määrät ovat keskimäärin pienempiä kuin alku- ja loppuvuodesta. Sama ilmiö on selkeästi havaittavissa muillakin pumppaamoilla. Yhtenä syynä näin selkeään vaihteluun on sääolosuhteista johtuva sade- ja sulamisvesien vaihteleva määrä viemäriverkostossa ja sitä kautta pumppaamoilla.

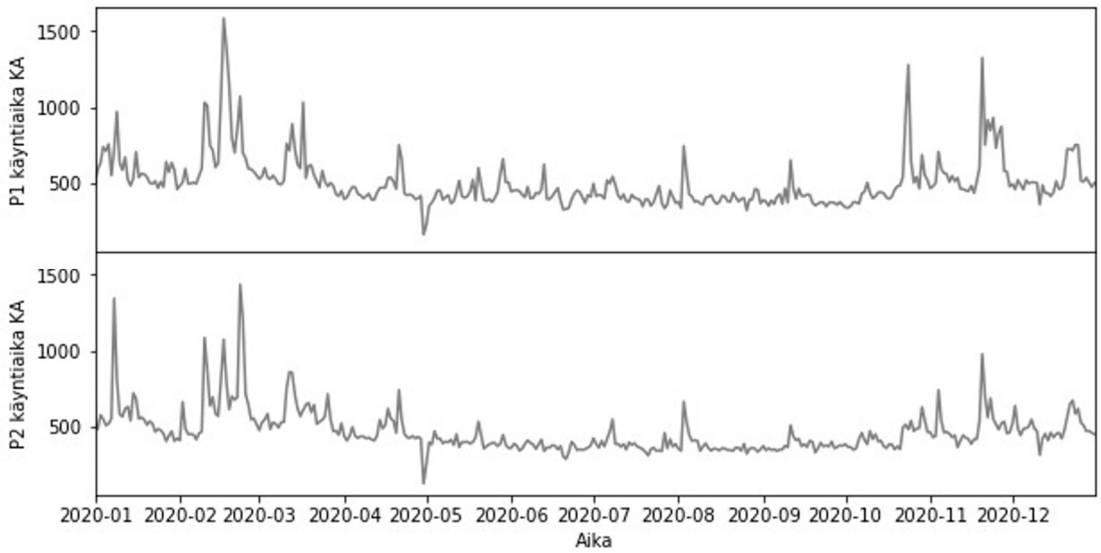
Ohjaamattoman oppimisen metodeja käytettäessä ei käytettävää dataa tarvitse jakaa erikseen opetusta, mahdollista validointia ja testausta varten, kuten ohjattua oppimista käytettäessä täytyy. Opetettaville algoritmeille voidaan siis syöttää yhden pumppaamon koko datajoukko kerralla tai vaihtoehtoisesti jakaa data pienempiin ajanjaksoihin ja opettaa kullekin ajanjaksolle oma malli. Kaikille algoritmeille käytetään samoja datajoukkoja mallien luomiseen, jotta saatuja tuloksia voidaan vertailla toisiinsa.

Mittausdatassa voi esiintyä erilaisia systemaattisia ja satunnaisia mittavirheitä. Systemaattisia mittavirheitä voivat aiheuttaa mitta-antureiden kalibrointien erot, niiden vaihto kesken mittausjakson sekä ääripäiden olosuhteet, jotka voivat vaikuttaa mitta-antureiden tarkkuuteen. Satunnaisia mittavirheitä aiheuttavat antureiden vikaantumiset, erilaiset katkokset mittauksissa sekä muut antureiden antamat nollalukemat. Mittavirheet tulisi pyrkiä mahdollisuuksien mukaan huomioimaan datan esikäsittelyn aikana, jotta ne eivät pääsisi vaikuttamaan mallien opetukseen.

4.4 Datan tunnusomaiset piirteet kunnonvalvonnan näkökulmasta

Vaikka pumppaamoiden aiemmista häiriötilanteista ei ole saatavilla tarkkaa dataa, tiedetään joidenkin häiriötilanteiden vaikuttavan erityisesti tiettyjen mittausten käyttäytymiseen. Tietoja häiriötilanteiden vaikutuksista mittausdataan voidaan käyttää apuna esimerkiksi mallien opettamiseen käytettävien piirteiden valinnassa ja analysoitaessa toteutettujen mallien toimivuutta tutkimalla mittausten arvoja mallien poikkeamiksi tunnistamissa pisteissä.

Normaalissa tilanteessa pumput ovat vuorottelukäytössä, joten niiden käyntikertojen ja käyntiaikojen tulisi olla normaalissa tilanteessa suunnilleen samansuuruisia. Kuvasta 14 näkee, että Kankkulan pumppujen 1 ja 2 käyntiaikojen päiväkohtaisten keskiarvojen kuvaajat mukailevat pääosin toisiaan.



Kuva 14: *Kankkulan pumppujen käyntiaikojen päiväkeskiarvot.*

Jonkin pumppaamon pumpun toiminnan häiriintyessä korvaavat muut pumput vajaatoimintaisen pumpun toiminnan. Tällöin muiden vuorottelukäytössä olevien pumppujen suhteelliset osuudet pumppujen kokonaiskäyntiajasta kasvavat. Muutokset pumppujen käyntiaikojen suhteissa ovat siis selkeitä merkkejä pumpuissa tai pumppauksessa esiintyvistä ongelmista tai häiriöistä.

Edellisessä alaluvussa todettiin pumppaamon toiminnassa ja mittausdatassa esiintyvän kausiluonteista vaihtelua. Vaikka sade, lumen sulaminen tai jokin muu pumppaamosta johtumaton syy voi aiheuttaa lisääntynyttä pumppauksen tarvetta ja sen myötä muutoksia ja piikkejä pumppujen käyntiajoissa sekä pumppaamoiden muissa mittauksissa, ovat nämä muutokset pumppaamon normaalia toimintaa. Samanlaiset piikit ja muutokset vähemmän sateisina aikoina voisivat olla seurausta häiriötilanteesta. Tästä syystä tulee malleja luotaessa pyrkiä huomioimaan ja arvioimaan myös erilaisten olosuhteiden ja kausittaisten vaihteluiden muodostettavalle työkalulle luomia haasteita.

Yksi selkeä pumppaamon tilaa kuvaava tarkastelukohde on ylivuotomittaus. Ihanteellisissa tilanteissa pumppaamoissa ei tapahdu ollenkaan ylivuotoa, joten ylivuodon mittauksen nollasta poikkeavat arvot ovat mahdollisia merkkejä pumppaamoissa esiintyvistä häiriöistä tai muista epätoivotuista tilanteista. Ylivuotoa voi tapahtua suoraan pumppaamon toimintaan liittymättömistä syistä, kuten pumppaamon maksimikapasiteetin ylittävästä jäteveden tulovirtausten kertymästä, tai pumppaamossa tapahtuvan häiriön aiheuttamasta pumppauskapasiteetin alenemisesta ja tätä kautta etukaivon ylitäyttymisestä. Vaikka ensimmäisen tapauksen syyt eivät liity suoraan pumppaamon toimimiseen ja ole siksi kunnossapidollisesta näkökulmasta olennaisia tietoja, saattavat nekin olla

operaattoria kiinnostavia tietoja. Jo yksittäiset jäteveden ylivuodot voivat olla ympäristölle haitallisia, joten jäteveden määrän äkillisen kasvamisen tai ylivuotoja aiheuttavien pumppujen häiriöiden syyt tulisi selvittää niiden aiheuttamien vahinkojen välttämiseksi tulevaisuudessa.

Pidempään jatkunut ylivuotojen esiintyminen tai selvät muiden mittausten arvojen pysyvät muutokset voivat olla pumppauksen häiriöiden sijaan merkkejä pumppaamon vaikutusalueella tapahtuneista muutoksista. Erityisesti suuret muutokset alueella voivat vaikuttaa pumppaamon pumppaustarpeeseen, jolloin pumppaamon kapasiteettia tulisi tarkastella uudestaan ja tarvittaessa nostaa. Esimerkiksi uusien asuinalueiden rakentaminen tai tehtaan lopettaminen alueella voivat vaikuttaa merkittävästi pumppaamon läpi virtaavan jäteveden määrään. Tällaisten ympäristön muutosten aiheuttamat muutokset mittausdatassa ja sitä kautta työkalun ilmoittamat poikkeavuudet aiempaan datan käyttäytymiseen verrattuna voivat olla hyödyllisiä merkkejä muuttuneesta toiminnasta. Poikkeavuuden aikaansaama operaattorin huomion kiinnittyminen muuttuneisiin datoihin ja muutosten juurisyyn selvitystyö voivat paljastaa sellaisten tekijöiden muutoksen, joka vaikuttaa myös pumppaamoon ja sen toimintaan. Esimerkiksi merkittävästi kasvanut pumppauskapasiteetin tarve voi aikaansaada tarpeen myös pumppaamon kapasiteetin kasvattamiselle. Ympäristössä tapahtuneet muutokset saattavat vaikuttaa mittausdataan sen verran, että mallien opettaminen voi olla hyvä suorittaa uudestaan uudella datalla.

4.5 Datan esikäsittely ja piirteiden valinta

Dataa tarkasteltiin ja esikäsiteltiin aluksi Excelissä. Datan mittaustulokset ovat tuntikohtaisia, joten aikaleimoja ja jokaista mittausta pitäisi raakadatassa olla 8784 kappaletta. Jokaisen pumppaamon datan joukosta poistettiin kaikki sellaiset aikaleimat, joiden jostakin mittauksesta puuttui arvo. Poistettavia aikaleimoja kaikilta seitsemältä laitokselta löytyi yhteensä 189 kappaletta, joka on 0,3 % kaikesta datamäärästä, joten voidaan olettaa näiden puuttuvien arvojen vaikutuksen mallien koulutukseen olevan olematon.

Mallien opettamiseen käytettävistä piirteistä jätettiin pois mittaukset, jotka kytkennän puuttumisen tai muun syyn takia näyttivät arvoa 0 tai olivat koko tarkastelujakson ajan vakio. Tällaisilla muuttujilla ei poikkeamien tunnistamisen kannalta ole merkitystä. Opetusdatan seasta poistettiin myös sellaiset mittaukset, jotka eivät todennäköisesti suoraan vaikuta häiriötilanteiden syntyyn tai joiden arvoon häiriötilanteet eivät vaikuta. Tällaisia mittauksia olivat esimerkiksi ulkolämpötila ja tuulen nopeus. Lopullisissa opetusdatajoukoissa piirteiden määrä oli pumppaamosta riippuen 11–18.

Mallien luomiseen käytetyt piirteet on esitetty taulukossa 2. Taulukon lista on yksinkertaistettu koko piirrejoukosta niin, että esimerkiksi eri pumppujen vastaavia mittauksia ei ole mainittu yksitellen. Kaikkien piirteiden mittausarvot ovat tuntikohtaisia keskiarvoja tai kertymiä. Jokaiselle mittauspisteelle on myös datetime-datatyyppinen aikaleima. Eri laitokset sisältävät erilaisia variaatioita taulukossa esitetyistä mittauksista.

Taulukko 2: *Valittujen piirteiden esittely.*

Piirre	Yksikkö	Kertymä/keskiarvo	Datatyyppi
Pumppu käyntiaika	s	Kertymä	Integer
Pumput kokonaiskäyntiaika	s	Kertymä	Integer
Pumppu käyntikerrat	kpl	Kertymä	Integer
Pumppu tuotto	m³/h	Keskiarvo	Float
Pumput kokonaistuotto	m³/h	Keskiarvo	Float
Pumppu virta	A	Keskiarvo	Integer
Tuleva	m³	Kertymä	Integer
Lähtevä	m³	Kertymä	Integer
Ylivuoto	s	Kertymä	Integer
Säiliön pinta	m	Keskiarvo	Float
Sakovirtaama	m³	Kertymä	Float
Sademäärä	mm	Keskiarvo	Integer

Valmiiden mittausten lisäksi luotiin omiksi piirteiksi kunkin pumpun käyntiajan osuus pumppaamon kaikkien pumppujen yhteenlasketusta käyntiajasta. Tämä mahdollistaa osuuksien muuttumisen suoran tarkkailun.

Kaikki käytettävä data oli annettu joko jatkuvina tai diskreetteinä numeroarvoina, joten datatyyppeinä ne olivat suoraan koneoppimisalgoritmin käytettävissä. Absoluuttisesti suurempia arvoja sisältävien piirteiden vaikutuksen korostumisen estämiseksi data skaalattiin ennen sen syöttämistä algoritmeille. Mallien opetukseen käytetty data skaalattiin pythonin valmiilla StandardScaler -funktiolla, joka käyttää skaalaukseen kaavaa 4.1 [75].

$$z = \frac{x - u}{s} \tag{4.1}$$

Kaavassa skaalattu arvo *z* lasketaan mittauksen arvon *x*, mittausten keskiarvon *u* ja keskihajonnan *s* avulla. Funktio skaalaa koko datajoukon nollakeskiarvoon ja yksikkövarianssiin.

4.6 Mallien toteutus

Työssä luotiin ohjelmarunko, josta muokattiin kaksi erillistä python-tiedostoa suorittamaan myöhemmin tässä kappaleessa esitettävät mallit. Ohjelmarungon rakenne on seuraavanlainen:

1. Käsiteltävän laitoksen valitseminen, datatiedoston lukeminen ja lataus.
2. Omien tuloksiin, haluttuihin kuviin ja testaustapoihin liittyvien parametrien asettaminen.
3. Mittausdatan skaalaus ja käsittely.
4. Valittujen algoritmien opettaminen, saatujen tulosten tallentaminen ja graafinen esittäminen.
5. Tulosten vertailun suorittaminen.

Ohjelmakoodi on rakennettu niin, että kohdissa 1 ja 2 tulee määritellä algoritmien vaatimat parametrit sekä joitakin omia parametreja. Näiden omien parametrien mukaan valitaan käsiteltävä laitos ja ajanjakso sekä piirretään halutut graafiset esitykset. Loppu ohjelmakoodi suorittaa kaikkien neljän algoritmin opetuksen ja tulosten esittämisen parametrien määrittelemällä tavalla. Yhdellä suorituskerralla tapahtuu aina yhden pumppaamon ja ajanjakson kaikkien mallien koulutus samalla opetusdatajoukolla. Jokaisella ajokerralla algoritmien automaattisesti optimoitavat parametrit määritellään käytettävälle opetusdatajoukolle sopivaksi ja mallit koulutetaan niiden mukaan. Tulosten esittämistä varten opetuksessa käytetyt datapisteet tallennetaan jokaisen algoritmin kohdalle erikseen omiksi DataFrame-taulukoiksi, joihin lisätään uusiksi muuttujiksi tieto datapisteen poikkeavuudesta sekä mahdollinen poikkeavuusarvo. Tämän lisäksi kunkin mallin tunnistamat poikkeavat pisteet kerätään omiin taulukoihin, mallien löytämiä poikkeamia vertaillaan toisiinsa ja alun määritelmien perusteella saaduista tuloksista piirretään havainnoivia kuvaajia ja muita graafisia esityksiä.

Ohjaamattomien menetelmien käyttäminen tekee parametrien optimoinnista haastavampaa ohjattuihin menetelmiin verrattuna. Opetettuja malleja ei voida validoida tai niiden toimintaa vertailla kuten ohjatun oppimisen menetelmissä voidaan tehdä tunnuslukujen, kuten tarkkuuden ja täsmällisyyden, avulla. Parametrien ja niiden määrittelyn avulla pyritään löytämään optimaaliset asetukset jokaiselle erilliselle algoritmille,

opetusdatajoukolle ja käsiteltävälle tapaukselle. Joidenkin parametrien kohdalla voidaan käyttää yleisiä ohjearvoja, erilaisia apukeinoja, kuten kappaleessa 3.5.2 esiteltyä kyynärpäämetodia, tai muuta kohteeseen tai käytettävän datan luonteeseen liittyvää tietämystä sopivien arvojen valitsemiseksi.

Työssä K-means klusteroinnille, iForestille ja LOF:lle käytettävät funktiot vaativat poikkeamien osuuden määrittelemistä yhtenä algoritmeille syötettävänä parametrina, kun taas DBSCAN päättelee itse opetusdatan ja algoritmille annettujen parametrien perusteella poikkeamat ja niiden määrän. Koska todellisista häiriötilanteista ja datan poikkeamista ei ollut etukäteen tietoa, päätettiin työssä käyttää kustakin opetusdatajoukosta DBSCAN-algoritmin mallin päättelemää poikkeamien määrää. Jokaisella mallien opetuskerralla laskettiin DBSCAN:in mallin tunnistamien poikkeamien osuus koko koulutusdatasta ja asetettiin muille malleille parametriksi vastaava poikkeamien osuus. Näin kaikki samaa opetusdatajoukkoa käyttävät mallit löytävät saman määrän poikkeamia ja niiden saamia tuloksia voidaan vertailla suoraan toisiinsa.

Muille algoritmien vaatimille parametreille määriteltiin arvot erikseen. Osa parametrien valinnoista riippui suoraan dataan tai piirteisiin liittyvistä ominaisuuksista, kuten dimensioiden tai datapisteiden lukumäärästä. Tällaiset valinnat pyrittiin automatisoimaan pythonin omien tai sen kirjastojen tarjoamien funktioiden avulla. Kaikille parametreille ei kuitenkaan ole etukäteen määriteltävissä optimiarvoa. Tällaisille parametreille haettiin arvot manuaalisesti iteroimalla muutaman eri työssä käytetyn opetusdatajoukon avulla ja päätettiin pitää nämä arvot vakioina kaikissa työn tapauksissa. Lopullisessa työkalussa parametrit tulee parhaan lopputuloksen takaamiseksi optimoida tarkasti tapauskohtaisesti kyseisen tapauksen ja opetukseen käytettävän datan mukaan.

Ennen DBSCAN-mallin koulutusta tulee sille valita parametrit ε ja *minPts*. Epsilonin ε arvon valitsemiseen voidaan käyttää apuna kyynärpäämetodia K-etäisyyskäyrään, joka kuvaa pisteiden etäisyyttä niitä lähimpänä olevaan pisteeseen. Etäisyys kunkin pisteen ja sen lähimmän naapurin välillä lasketaan lähimmän naapurin menetelmän avulla pythonin NearestNeigbors-funktion avulla. Epsilonin arvo valitaan K-etäisyyskäyrästä kyynärpäämetodilla käyttäen pythonin kneed-kirjaston KneeLocator-funktiota. Funktio etsii sille annetun datan perusteella taitekohdan, jota vastaava arvo asetetaan epsiloniksi. Taitekohdan etsiminen funktion avulla automatisoi parametrin arvon valinnan ja säästää visuaalisen kuvaajan avulla tehtävältä arviointityöltä, jonka täsmällisyys ja toistotarkkuus voivat vaihdella arvioinnin suorittajan mukaan. Parametrille *minPts* suositellaan valittavaksi muun tiedon puutteessa arvoksi *minPts=2*dim*, jossa *dim* on datan sisältämien piirteiden lukumäärä [56, 57].

Syötetyn koulutusdatan ja asetettujen parametrien perusteella sklearnin cluster-kirjaston DBSCAN-funktio luo mallin, joka sijoittaa pisteet klustereihin luvussa 3.1 esitetyn toimintamallin mukaan. Algoritmi päättelee itse datan perusteella sopivan klustereiden määrän ja asettaa kohinalle sekä poikkeaville pisteille klusteriksi -1.

Asetettujen parametrien suuruus määrittelee klustereihin luokiteltaville pisteille vaatimukset, joten ne vaikuttavat suoraan poikkeamiksi laskettavien pisteiden määrään. Esimerkiksi pienempi *minPts* muodostaa datasta useampia klustereita, mutta klusterit saattavat sisältää myös kohinaa tai poikkeamia. Suuremmalla *minPts*:n arvolla malli jättää enemmän pisteitä klustereiden ulkopuolelle eli poikkeamiksi. Parametreja säätämällä voitaisiin siis vaikuttaa saatavien poikkeamien määrään. Nyt kuitenkin päätettiin suorittaa mallien opettaminen käyttäen yleisesti suositeltuja parametrien arvoja.

Ennen K-means klusteroinnin suorittamista datalle tehdään pääkomponenttianalyysi dimensioiden vähentämiseksi. PCA:n suorittamiseen käytettiin sklearnin decomposition-kirjaston valmista PCA-funktiota. Funktion muodostamista pääkomponenteista valitaan vain merkityksellisimmät komponentit käytettäväksi K-means klusteroinnin mallin opettamiseen. Käytettävien pääkomponenttien määrä valittiin Pythonin KneeLocator-funktion avulla kullekin koulutusdatajoukolle siitä muodostettujen komponenttien ja niitä vastaavan kumulatiivisen selitettävän varianssin kuvaajan kyynärpääkohdasta. Pääkomponenttien tapauksessa taitekohta sijoittuu sen pääkomponentin kohdalle, jonka jälkeen komponenttien määrän kasvattaminen ei enää lisää merkittävästi selittävää varianssia. Taitekohtaa vastaavan komponenttien määrän valitsemisella varmistetaan, että pois jätettävien pääkomponenttien mukana ei menetä oleellista informaatiota. Työssä käytettyjen mittausdatajoukkojen piirteiden määrän pääkomponenttianalyysi vähensi datajoukosta riippuen 4–9 piirteeseen.

Ideaalinen klustereiden määrä riippuu täysin tutkimuskysymyksestä ja se tulisi aina valita tapauskohtaisesti. Tässä tapauksessa ei optimaalisesta klustereiden määrästä ollut käyttökohteen asettamia vaatimuksia tai muuta tietoa, joten myös niiden määrä valittiin kyynärpäämetodin avulla. Pythonin KneeLocator-funktio laski annettujen klustereiden määrien ja niiden inertia-arvojen kuvaajan kyynärpääkohdan, jota vastaava klustereiden määrä valittiin mallissa käytettäväksi määräksi.

Klusterointi toteutettiin sklearnin cluster-kirjaston KMeans-funktion avulla. Tämän jälkeen laskettiin jokaisen pisteen etäisyys oman klusterinsa sentroidiin. Poikkeamiksi merkittiin DBSCAN:in tulosten mukaan määriteltyä poikkeamien osuutta vastaava osuus suurimmalla etäisyydellä omista sentroideistaan olevista pisteistä.

iForestin mallien luomiseen käytettiin sklearnin ensemble-kirjaston valmista IsolationForest-funktiota. iForestin toiminta perustuu poikkeavuusarvojen laskemiseen ja IsolationForest-funktio määrittelee pisteen poikkeavuuden sille parametrina annetun poikkeamien osuuksien ja suuruusjärjestykseen listattujen poikkeavuusarvojen mukaan. Pythonin valmiit funktiot eivät anna suoraan luvussa 3.3 esitetyn iForestin määritelmän mukaisia poikkeavuusarvoja, vaan decision_function-metodi antaa poikkeamille muunnellun version poikkeavuusarvosta [76]. Suurempi negatiivinen arvon kertoo suuremmasta poikkeavuudesta, kun taas normaalit pisteet saavat positiivisen poikkeavuusarvon. Funktiolle määriteltävät parametrit valittiin kokeilemalla. Pienemmillä n_estimators ja max_samples arvoilla mallin tuottamat tulokset muuttuivat jokaisella koulutuskerralla. Suurentamalla n_estimators-arvoa algoritmin koulutusaika pitenee. Asetetun arvon 1000 jälkeen eri koulutuskertojen väliset erot saaduissa tuloksissa eivät enää pienentyneet merkittävästi verrattuna koulutusajan pidentymiseen. Parametrin max_samples arvoksi asetettiin käytetyn koulutusdatan pisteiden määrä.

Local outlier factorin mallin suorittamiseen käytettiin sklearnin valmista neighbours-kirjaston LocalOutlierFactor-funktiota. Annetun poikkeamien osuuden perusteella poikkeamiksi merkitään vastaava osuus koulutusdatan pisteistä funktion laskemien LOF-arvojen suuruuden mukaan. Funktiolle on annettava parametreina laskennassa huomioon otettavien naapurien määrä sekä käytetty etäisyysfunktio. Käytettävä naapurien määrä valittiin kokeilemalla. Aluksi arvon kasvattaminen lisäsi LOF:in ja muiden algoritmien löytämien samojen poikkeamien määrää, mutta noin arvojen 300–400 kohdilla samankaltaisuuksien määrä alkoi heilahtelemaan algoritmeista riippuen. Lopulta päädyttiin asettamaan parametrin n_neighbors arvoksi 350.

4.7 Mallien toimivuuden määrittäminen

Ohjaamattoman oppimisen menetelmien suoriutumisen arvioiminen voi olla haasteellista, sillä merkitsemättömän opetusdatan käyttämisen takia malleille ei ole selkeitä tunnuslukuja suorituskyvyn ja tarkkuuden arvioimiseen, kuten ohjatun oppimisen menetelmillä on. Tämä tekee mallien tulosten analysoinnista ja arvioinnista monimutkaisempaa ja epävarmempaa. Tässä työssä mallien arvioiminen päätettiin suorittaa eri tavoin toteutettujen mallien tuloksia vertailemalla sekä visuaalisesti poikkeamia tarkastelemalla erilaisten graafisten esitysten avulla. Poikkeamien ja käytössä olevan datan ominaisuuksien sekä käytettävien algoritmien tuomien mahdollisuuksien mukaan suunniteltiin erilaisia tapoja algoritmien ja mallien arviointiin.

Yksi vertailun kohteista on samoja datajoukkoja käsittelevien mallien tunnistamien samojen poikkeamien lukumäärä. Mitä enemmän samoja poikkeamia eri mallit tunnistavat,

sitä varmemmin voidaan olettaa mallien tunnistaneen oleellisia poikkeamia ja tunnistettujen poikkeamien voidaan uskoa olevan todellisuudessa muista eroavia pisteitä. Vertailtavien tulosten yhtenäistämiseksi käytetään samoja esikäsiteltyjä datajoukkoja kaikilla käytetyillä malleilla sekä asetetaan samoja opetusdatajoukkoja käsittelevien mallien poikkeamien osuudet samoihin arvoihin.

Kuten aiemmin on todettu, vaikuttavat eri vuodenajat ja niiden tyypilliset sääolosuhteet jätevedenpumppaamoilta saatavaan mittausdataan. Kaikille pumppaamoille päätettiin luoda koko vuoden mittausdatan perusteella opetetun mallin lisäksi erikseen mallit käyttäen kausittaisen vaihtelun mukaan jaettavia jaksoja. Mittausdatan vaihteluiden perusteella koko vuoden mittausdata jaettiin neljään yhtä suureen, peräkkäiset aikaleimat sisältävään osaan, joista jokainen sisälsi noin kolmen kuukauden pituisen ajanjakson. Jokaiselle ajanjaksolle suoritettiin itsenäinen mallien koulutus ja vastaavien algoritmien malleista saadut poikkeamat kerättiin koko vuoden listaksi. Kausittaiset mallit pystyvät oppimaan esimerkiksi sateisemmille neljänneksille tyypillisemmät käyttäytymismallit ominaan ja näin se osaa tunnistaa juuri tämän ajanjakson lokaalin normaalin toiminnan ja erottamaan siitä poikkeamat. Tällöin myös vähemmän sateisille ajanjaksoille sijoittuvia poikkeamia malli osaa mahdollisesti analysoida paikallisesti tarkemmin, kuin koko vuoden datasta opetetulla mallilla. Toisaalta taas kunkin mallin muodostamiseen käytetyn opetusdatan määrä vähenee huomattavasti, mikä voi aiheuttaa esimerkiksi kohinan tai mittauksissa esiintyvien virheiden vaikutuksien kasvamista mallin koulutuksessa. Myös erilaisten tarkasteluajanjaksolle osuneiden epätyypillisten ajanjaksojen vaikutus malliin on näin suurempi.

Jokaiselle pumppaamolle luotiin siis viisi erillistä mallia kustakin neljästä valitusta algoritmista käyttäen vuoden neljänneksien sekä koko vuoden dataa. Jokainen tuotettu malli arvioi jokaisen sille annetun koulutusdatapisteen kuuluvan joko normaaleihin tai muusta datasta poikkeaviin pisteisiin. Mallien tunnistamat poikkeamat kerättiin omille listoilleen ja kunkin koulutusdatajoukon luoman mallin löytämiä poikkeamia vertailtiin toisten samaa ajanjaksoa käsittelevien mallien löytämiin poikkeamiin. Eri algoritmien löytämien poikkeamien vertailun lisäksi voidaan vertailla koko vuotta käsittelevän mallin tunnistamia poikkeamia saman algoritmin paloittain koulutettujen mallien löytämiin poikkeamiin. Näissä tapauksissa mallien löytämiä poikkeamia on eri määrä, mikä tulee huomioida vertailua tehtäessä.

Luvussa 4.4 todettiin pumppujen käyntiajoissa ja niiden suhteissa esiintyvien muutosten tai äkillisten vaihteluiden olevan mahdollisia merkkejä pumppaamoissa esiintyvistä häiriöistä. Tästä syystä päätettiin tarkastella erikseen pelkästään pumppujen käyntiaikojen osuuksia. Käyntiaikojen tuntikohtainen tarkastelu todettiin olevan liian lyhyt ajanjakso

todellisten häiriöiden tunnistamiseksi, joten mallien luomiseen päätettiin käyttää käyntiaikojen laskettujen osuuksien päiväkohtaisia keskiarvoja. Vuorottelukäytössä olevien pumppujen käyntiaikojen osuuksilla ei pitäisi olla kausittaista vaihtelua, joten osuuksista luodaan mallit pelkästään koko vuoden datan avulla. Toisaalta myös päiväkohtaisten keskiarvojen käyttö vähentää koulutusdatan määrää radikaalisti, eikä datamäärän jakaminen enää pienempiin osiin ole todennäköisesti kannattavaa.

Pelkkä eri mallien saamien tuloksien vertaileminen toisiinsa ei välttämättä riitä kertomaan mallien sopivuudesta juuri kyseiseen tehtävään. Mallien saamat samat tulokset saattaisivat johtua vain mallien samanlaisista perusperiaatteista tai toimintatavoista. Saatuja tuloksia tarkastellaan myös pistetasolla. Osa potentiaalisista häiriötilanteista näkyvät esimerkiksi piikkeinä mittauksissa, joten eri mallien tunnistamia poikkeamia voidaan tarkastella tutkimalla niiden muuttujien arvoja. Eri mallien tunnistamat poikkeamat voidaan merkitä samoihin kuvaajiin ja niiden mittausarvoja sekä tunnistettujen poikkeamien käyttäytymistä ja sijoittumista voidaan pyrkiä tutkimaan kuvaajien avulla.

Toinen käytettävä visuaalinen arviointimenetelmä on tulosten merkitseminen mittauksista tehtävään kaksiulotteiseen PCA-kuvaan. Alkuperäiselle koulutusdatajoukolle suoritetaan pääkomponenttianalyysi ja sen kahdesta ensimmäisestä pääkomponentista luodaan PCA-kuva, johon merkitään mallin tunnistamat poikkeavat ja normaalit pisteet eri väreillä. PCA:n avulla kuvataan datan sisältämää vaihtelua, joten sitä voidaan käyttää apuna poikkeamiksi luokiteltujen pisteiden todellisen poikkeavuuden arviointiin. Poikkeavat pisteet sisältävät muihin pisteisiin verrattuna eri määrän koko datajoukon sisältämästä vaihtelusta, joten PCA-kuvassa ne esiintyvät etäällä muista pisteistä.

Käyntiaikojen osuuksien mallien löytämiä poikkeamia arvioidaan myös pistekaaviomatriisien avulla. Vähädimensioinen data on helposti tarkasteltavissa muuttujaparien pistekuvaajien avulla. Kuvaajien perusteella voidaan arvioida poikkeamien sijoittumista muuttujien arvojen vaihteluvälille sekä toisaalta taas poikkeamien sisältämiä muuttujien arvoja.

5. TULOKSET JA JOHTOPÄÄTÖKSET

Tässä luvussa käydään läpi koneoppimismallien tuottamia tuloksia ja analysoidaan eri algoritmien suoriutumista annetusta tehtävästä. Valituista algoritmeista koulutettiin pumppaamoiden mittausdatan avulla malleja, joiden tehtävä oli luokitella jokainen annettu mittauspiste normaaliksi pisteeksi tai poikkeamaksi. Algoritmeista ja koulutusdatajoukoista luotiin luvussa 4.7 esiteltyjen tapojen mukaan erilaisia malleja, jotta kunkin algoritmin suorituskykyä ja sopivuutta esitellyn ongelman ratkaisemiseksi voitaisiin arvioida. Saadut tulokset esitellään ja niitä arvioidaan vertailemalla ja analysoimalla eri mallien ja tapojen tuottamia tuloksia sekä tuodaan esille muita työn aikana tehtyjä havaintoja. Lopuksi myös pohditaan työkalun toiminnan sekä yleisesti koneoppimisen käyttämisen asettamia vaatimuksia sekä mahdollisia haasteita.

5.1 Mittausdatojen mallit

Ensimmäisessä osuudessa mallien opettamiseen käytettiin luvussa 4.5 esiteltyjä pumppujen toimintaan, pumppaukseen tai mahdollisiin häiriöihin liittyviä piirteitä. Kaikista neljästä valitusta algoritmista opetettiin jokaista pumppaamoa edustavasta opetusdatajoukosta omat mallit ja kunkin mallin tunnistamat poikkeamat kerättiin omiin listoihin. Mallien opettamiseen käytettiin aluksi pumppaamoiden koko vuoden mittausdatoista esikäsiteltyjä opetusdatajoukkoja. Tämän jälkeen suoritettiin luvussa 4.7 esitetyn tavan mukainen opetusdatan jakaminen jaksoihin ja opetettiin mallit jokaiselle jaksolle erikseen. Kunkin algoritmin kausittaisten mallien poikkeamat kerättiin yhteen listaan kuvastamaan kaikkia vuoden aikana löydettyjä poikkeamia. Mallien tuottamia tuloksia arvioitiin pumppaamokohtaisesti tarkastelemalla ja vertailemalla yksittäisestä pumppaamosta luotujen mallien poikkeamalistoja.

Aluksi vertailtiin toisiinsa pumppaamokohtaisesti kunkin algoritmin koko vuoden mallin sekä kausittaisten mallien löytämien poikkeamien yhtäläisyyksien määrää. Tunnistettujen poikkeamien määrä määräytyy DBSCAN-mallin ja sille asetettujen parametrien mukaan kustakin käsiteltävästä opetusdatajoukosta tapauskohtaisesti. Muille algoritmeille aseteltiin poikkeamien osuuden määrittelevät parametrit niin, että kaikki mallit tunnistavat kustakin opetusdatajoukosta saman määrän poikkeamia. Parametrien arvoja ei kuitenkaan lähdetty säätämään kausittaisille toteutuksille niin, että niiden tunnistamien poikkeamien yhteismäärä olisi sama kuin koko vuoden mallin tunnistamien poikkeamien määrä. Saatuja tuloksia tarkastellessa on siis hyvä ottaa huomioon löydettyjen poikkeamien eri määrät. Koko vuoden mallien sekä kausittaisten mallien tunnistamien

poikkeamien lukumäärät on kerätty taulukon 3 vasemmanpuoleisiin sarakkeisiin. Kunkin algoritmin tulosten vertailussa löydettyjen yhtäläisyyksien määrät löytyvät taulukon oikeanpuoleisista sarakkeista.

Taulukko 3: *Koko vuoden mallin ja kausittain koulutettujen mallien tunnistamien poikkeamien lukumäärät sekä eri algoritmien mallien löytämien yhtäläisyyksien määrät.*

	Jakso		Käytetty algoritmi			
Pumppaamo	Koko vuosi	Kausittaiset yhteensä	iForest	LOF	K-means	DBSCAN
Roope	282	471	242	237	249	224
Pähkinäkallio	179	455	153	79	130	165
Kankkula	218	462	196	187	68	199
Lähtösaha	446	218	192	124	2	184
Taivallammi	187	290	106	94	8	115
Suntinmäki	144	391	114	108	70	101
Saarenmaa	234	601	189	182	91	189

Taulukon toisesta ja kolmannesta sarakkeesta löytyvät jokaisen pumppaamon koko vuoden datasta tuotetun mallin löytämien poikkeamien määrät sekä vuoden neljäsosien datoista koulutettujen mallien löytämien poikkeamien yhteenlasketut määrät. Taulukosta nähdään, että lähes kaikkien pumppaamoiden kohdalla kausittaisten mallien tunnistamien poikkeamien määrä on huomattavasti suurempi kuin koko vuoden mallien. Ainoa poikkeus edelliseen on Lähtösahan pumppaamo. Koko vuoden mallien poikkeamien osuudet koko koulutusdatasta vaihtelevat välillä 1,7–5,1 % ja kausittain tuotettujen mallien keskimääräiset poikkeamien osuudet ovat 2,5–6,9 %.

Käytettyjen algoritmien sarakkeissa olevat luvut kuvaavat kunkin algoritmin koko vuoden mallin ja kausittain opetettujen mallien löytämien samojen poikkeamien lukumäärää. Näistä lukumääristä voidaan nähdä eri algoritmien löytämien yhtäläisyyksien osuuden vaihtelevan algoritmien lisäksi myös selkeästi eri pumppaamoiden välillä. Esimerkiksi Roopen kohdalla K-means klusteroinnin mallit löysivät eniten samoja poikkeamia, mutta lähes kaikissa muissa pumppaamoissa K-means klusteroinnin mallien tuloksissa yhtäläisyyksiä oli selkeästi vähiten. Toisaalta taas Lähtösahan ja Taivallammen kohdalla kaikkien algoritmien vertailuissa yhtäläisyyksien osuudet olivat pienimpiä.

Poikkeamien lukumääristä on taulukkoon 4 laskettu kunkin algoritmin koko vuoden mallien sekä kausittaisten mallien löytämien yhtäläisyyksien lukumäärien keskiarvo. Taulukon toiselle riville on laskettu kaikkien yhtäläisyyksien prosentuaalinen osuus kaikista molemmilla tavoilla löydetyistä poikkeamista ja alin rivi kuvaa yhtäläisyyksien prosentuaalista osuutta vähemmän poikkeamia löytäneen tavan koko poikkeamamäärästä. Tässä vertailussa pienempää poikkeamien määrää edustivat siis Lähtösahaa lukuun ottamatta kaikilla pumppaamoilla koko vuoden mallien tunnistamat poikkeamat.

Taulukko 4: *Kaikkien pumppaamoiden yhtäläisyyksien osuuksien keskiarvo eri algoritmien tunnistamista poikkeamista.*

	iForest	LOF	K-means	DBSCAN
Keskiarvo (kpl)	170	144	88	168
Keskiarvoinen osuus kaikista poikkeamista (%)	51	43	26	51
Keskiarvoinen osuus pienemmästä poikkeamien määrästä (%)	84	70	43	83

Edellisten taulukoiden mukaan iForestia ja DBSCAN:ia käyttävät mallit löysivät eniten samoja poikkeamia koko vuoden mallin ja kausittaisten mallien välisessä vertailussa. Molemmat löysivät keskimäärin saman verran samoja poikkeamia. Noin puolet kaikista niiden tunnistamista poikkeamista löytyivät molemmilta listoilta ja 83–84 % määrällisesti vähemmän poikkeamia tunnistaneiden mallien tuloksista löytyi myös toisella tavoin muodostetusta listasta. LOF:ia käyttävät mallit löysivät taas 43 % ja 70 % samoja poikkeamia vastaavissa vertailuissa. K-means klusteroinnin mallit löysivät selvästi vähiten yhtäläisyyksiä saadessaan osuuksiksi 26 % ja 43 %.

Samoista algoritmeista tuotettujen mallien vertailun lisäksi voidaan eri algoritmeista opetettujen mallien saamia tuloksia vertailla toisiinsa. Löydettyjen poikkeamien pumppaamokohtaiset osuudet oli työssä asetettu kaikille samoja opetusdatajoukkoja käsitteleville algoritmeille samoiksi, joten eri algoritmiparien yhtäläisyyksien osuuksia voidaan vertailla suoraan keskenään. Taulukkoon 5 on kerätty kunkin algoritmiparin tunnistamien samojen poikkeamien prosentuaalinen osuus kaikista niiden tunnistamista poikkeamista. Osuudet on laskettu erikseen koko vuoden mallille sekä kausittaisten mallien kaikille poikkeamille.

Taulukko 5: *Kaikkien pumppaamoiden yhtäläisyyksien osuuksien keskiarvo eri algoritmien tunnistamista poikkeamista.*

Vertailtavat algoritmit	Koko vuosi (%)	Kausittaiset (%)
iForest ja LOF	55	64
iForest ja K-means	36	38
iForest ja DBSCAN	74	71
LOF ja K-means	25	34
LOF ja DBSCAN	63	77
K-means ja DBSCAN	27	33

Taulukon tulokset osoittavat kausittaisten mallien käytön tuottavan lähes kaikissa työn tapauksissa suuremman osuuden samoja poikkeamia. Ainoa poikkeus tässä oli iForest ja DBSCAN niidenkin välisen eron ollessa vain 3 %. Kaikki iForestin, LOF:in ja DBSCAN:in väliset vertailut löysivät keskimäärin yli 50 % samoja poikkeamia. K-means klusteroinnin mallit löysivät selkeästi vähiten yhtäläisyyksiä muiden algoritmien mallien tunnistamien poikkeamien kanssa.

Eri pumppaamoiden mallien vertailuiden tuloksissa on nähtävissä eroja. Taulukkoon 6 on kerätty pumppaamoilta kaikkien koko vuoden mallien niistä löytämien samojen poikkeamien määrät ja niiden osuudet koko datasta.

Taulukko 6: *Kaikkien mallien tunnistamat poikkeamat ja niiden osuus kaikista poikkeamista.*

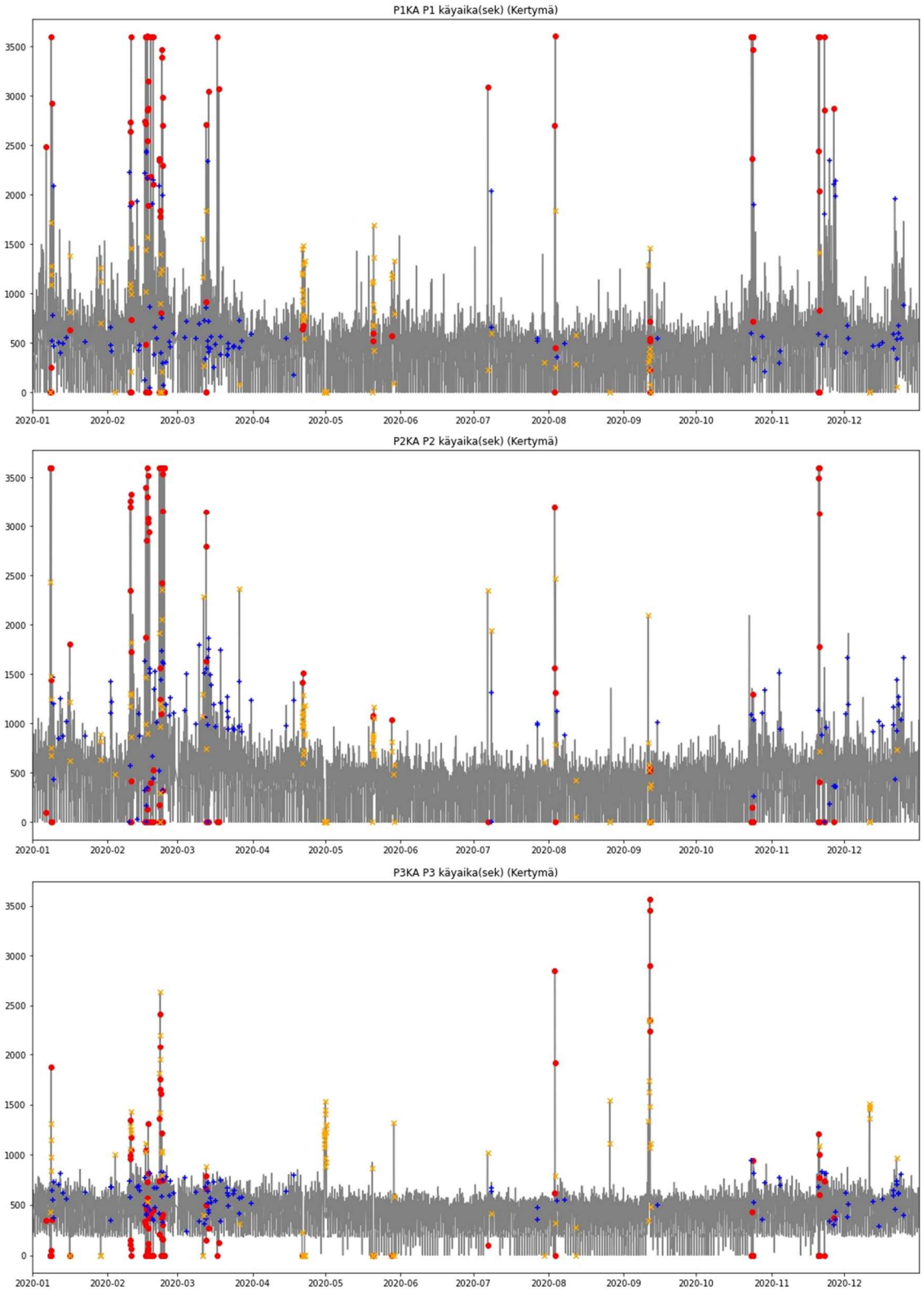
Pumppaamo	Kaikkien tunnistamat poikkeamat (kpl)	Osuus kaikista poikkeamista (%)
Roope	83	29
Pähkinäkallio	55	31
Kankkula	108	50
Lähtösaha	20	6
Taivallammi	10	5
Suntinmäki	59	52
Saarenmaa	56	24

Taulukosta nähdään yhtäläisyyksien osuuksien vaihtelevan 5–52 % välillä. Yksi selkeä syy kaikkien löytämien yhtäläisyyksien vaihtelevalle määrälle on K-meansin muiden kanssa löytämien samojen poikkeamien määrän suuri vaihtelu. Taivallammen ja Lähtösahan pumppaamoille K-means tunnisti selkeästi vähiten samoja poikkeamia niin muihin algoritmeihin verrattuna, kuin sen omien kausittaisten ja koko vuoden mallien kohdalla.

Edellisten taulukoiden ja niistä tehtyjen päätelmien perusteella voidaan todeta K-means klusteroinnin mallien toimivan työn tapauksissa selkeästi eri tavalla muihin algoritmeihin verrattuna. Lähes kaikissa tutkituissa tapauksissa se löysi vähiten samoja poikkeamia vertailukohteidensa kanssa ja eri pituisten tarkastelujaksojen käyttö vaikutti merkittävästi algoritmin tuottamiin tuloksiin. Suuri vaihtelu tuloksissa lisää metodin toimimisen epävarmuutta ja voi tehdä esimerkiksi koulutusdatajoukon valinnasta vaikeampaa.

Eri algoritmien mallien tuottamia tuloksia voidaan tutkia myös suoraan tarkastelemalla poikkeamiksi luokiteltujen pisteiden alkuperäisiä mittausarvoja. Esimerkiksi kuvaajia sekä mittausten arvoja tarkastelemalla voidaan todeta kaikkien mallien tunnistaneen lähes kaikki pumppaamoilla esiintyneiden ylivuotojen mittauspiikkien ajankohdat poikkeamiksi. Ylivuodot ovat selkeitä merkkejä pumppaamoissa mahdollisesti esiintyvistä häiriöistä, joten häiriöiden etsintään tarkoitettujen mallien tulisikin tunnistaa ylivuotoa sisältävät pisteet poikkeamiksi.

Tarkemmin poikkeamien alkuperäisiä mittausarvoja ja poikkeamien eroja päätettiin tutkia DBSCAN:in ja K-means klusteroinnin antamista tuloksista. Jokaiselta pumppaamolta piirrettiin kuvaajat kustakin mittauksesta ja kuvaajiin merkittiin erilaisella merkeillä poikkeamat, jotka vain DBSCAN tai K-means tunnisti tai jotka molemmat tunnistivat poikkeamiksi. Kuvaan 15 on Kankkulan pumppujen käyntiaikojen kuvaajiin merkitty sinisellä kaikki pelkän K-meansin tunnistamat poikkeamat, oranssilla pelkän DBSCAN:in tunnistamat poikkeamat ja punaisella kaikki molempien mallien tunnistamat poikkeamat.



Kuva 15: *Kankkulan pumppaamon käyntiaikojen kuvaajat, joihin on merkitty K-meansin ja DBSCAN:in tunnistamat poikkeamat.*

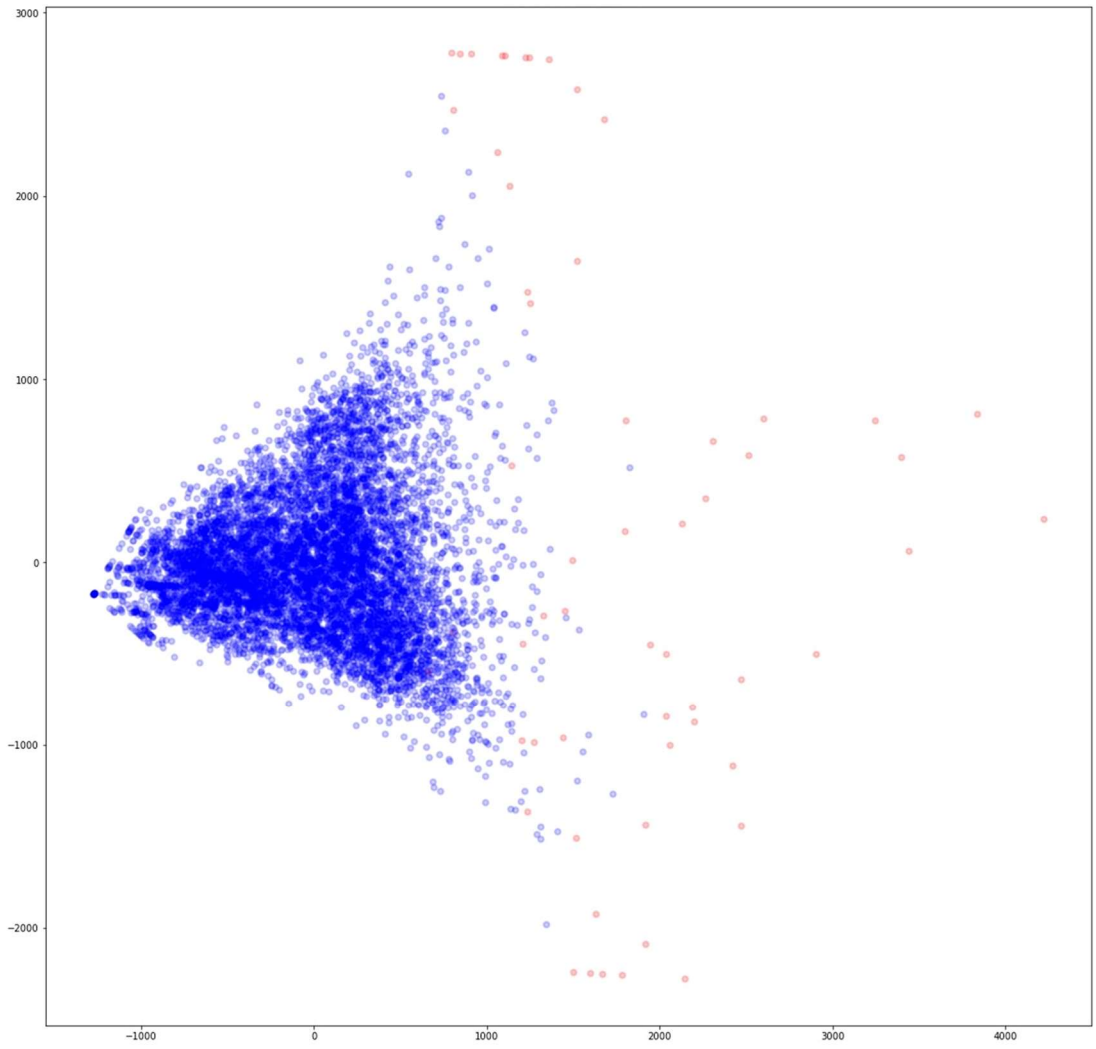
Kuvasta nähdään, että molemmat mallit ovat tunnistaneet poikkeamiksi lähes kaikki suurimmat käyntiaikojen mittauspiikit ja suuri osa niiden yhteisesti tunnistamista

poikkeamista sijaitsevatkin tällaisissa mittauspiikeissä. Kun näiden poikkeavien pisteiden kaikkia käyntiaikojen mittaustuloksia tarkastelee, on osassa tapauksissa vain selkeästi kyse pumppaustarpeen äkillisestä kasvamisesta, sillä kaikkien pumppujen käyntiajat ovat korkeat. Osassa tapauksista taas ainakin yhden pumpun käyntiaika on nolla yhden tai useamman tunnin ajan peräkkäin, jolloin muiden pumppujen käyntiajat ovat kasvaneet yhden tai useamman pumpun ollessa poissa vuorottelukäytöstä.

Kaikkien pumppaamoiden vastaavia kuvia tarkastellessa voidaan huomata eri algoritmien toimivan silmämääräisesti tarkasteltuna vaihtelevasti riippuen pumppaamon datasta sekä sen sisältämästä vaihtelusta ja mittauspiikeistä riippuen. Molempien mallit vaikuttavat tunnistaneen poikkeamiksi kohtalaisen suuren osan selkeistä yksittäisistä suurimmista mittauspiikeistä. Muita mallien löytämiä poikkeamia visuaalisesti tarkastellessa ei nähdä selvästi mitään kaikkia tapauksia koskevia selkeitä toimintatapoja. Keskikokoisten mittauspiikkien tunnistaminen on vaihdellut paljon eri pumppaamoiden välillä. Esimerkiksi K-means- ja DBSCAN-mallien tuloksia tarkastellessa Roopen kuvaajissa, K-meansin tunnistamat poikkeamat sijoittuivat käyntiaikojen piikkeihin paremmin, kun taas Taivallammen kohdalla DBSCAN tunnisti enemmän piikkejä, K-meansin poikkeamien sijoittuessa mittausten minimiarvoihin, joita ei kuvaajien perusteella voisi päätellä poikkeamiksi.

PCA:ta voidaan käyttää apuna saatujen tulosten visuaalisessa analysoinnissa. Kunkin pumppaamon koulutusdatalle tehtiin pääkomponenttianalyysi ja pisteet kerättiin kuvaan kahden ensimmäisen pääkomponentin sisältämästä informaatiosta. Tarkastelua varten pumppaamoiden datoista piirrettiin PCA-kuvat erikseen jokaisen mallin tunnistamille poikkeamille sekä kaikkien mallien löytämistä yhteisistä poikkeamista. Kuvassa 16 on esitetty Suntinmäen koko vuoden mittausdatasta muodostettu PCA-kuva. Kuvaan on merkitty punaisella kaikkien algoritmien mallien poikkeamaksi luokittelemat pisteet. Siniset pisteet ovat siis ainakin yhden mallin normaaliksi määrittelemiä pisteitä.

Kuvassa siniset pisteet sijoittuvat pääosin korkean tiheyden alueelle, kun taas poikkeamat ovat levittäytyneet kuvassa laajalle alueelle. PCA kuvastaa datan vaihtelua, joten poikkeamien oletetaankin sijaitsevan PCA-kuvassa kauempana pisteryhmittymistä. Kuvan perusteella voidaan olettaa kaikkien mallien tunnistaneen poikkeamiksi hyvin muista PCA:n mukaan eroavia pisteitä.



Kuva 16: *Kaikkien mallien löytämät poikkeamat Suntinmäen PCA-kuvassa.*

Kuvassa siniset pisteet sijoittuvat pääosin korkean tiheyden alueelle kun taas poikkeamat ovat levittäytyneet kuvassa laajalle alueelle. PCA kuvastaa datan vaihtelua, joten poikkeamien oletetaankin sijaitsevan PCA-kuvassa kauempana pisteryhmittymistä. Kuvan perusteella voidaan olettaa kaikkien mallien tunnistaneen poikkeamiksi hyvin muista PCA:n mukaan eroavia pisteitä.

Kaikkien pumppaamoiden vastaavat PCA-kuvat on kerätty liitteeseen B. Kuvien perusteella voidaan todeta, että kaikkien mallien tunnistamat poikkeamat eivät kaikilla pumppaamoilla olleet PCA:n perusteella yhtä selkeitä poikkeamia. Esimerkiksi Suntinmäen ja Pähkinäkallion kaikkien mallien tunnistamat poikkeamat sijaitsivat pääosin pisteryhmittymien ulkopuolella, kun taas Roopen PCA-kuvassa pumppaamodatasta löydetyt poikkeamat sijaitsevat selkeämmin muun datan seassa.

Myös yksittäisen mallien tuloksista tehtyjä PCA-kuvia tarkasteltiin lähemmin. Monen pumppaamon kohdalla yksittäisten mallien PCA-kuvia verratessa kaikkien mallien

löytämistä poikkeamista tehtyihin PCA-kuviin voidaan huomata niiden eroavan varsinkin satunnaisemmin dataryhmittymien seassa olevien poikkeamien kohdalla. Useat mallit ovat tunnistaneet poikkeamiksi jonkin verran pääkomponenttien selkeiden pisteryhmittymien alueella sijaitsevia pisteitä. Nämä tunnistetut pisteet kuitenkin selkeästi vaihtelevat eri algoritmeilla. Tämä viittaisi siihen, että suurin osa mallien tulosten eroista johtuu vähemmän poikkeavien pisteiden tunnistamiseen liittyvistä eroista.

5.2 Käyntiaikojen osuuksien mallit

Kappaleessa 4.7 esitetiin edellisen testauskokonaisuuden lisäksi pumppujen käyntiaikojen osuuksien käyttäminen algoritmien testaamisessa. Jokaiselle pumpulle laskettiin osuus kaikkien pumppujen kokonaiskäyntiajasta ja käyntiaikojen osuuksien päiväkeskiarvoja käytettiin mallien luomiseen. Aiemmin määriteltyjen DBSCAN-algoritmin parametrien kanssa pelkkien osuuksien datasta koulutetut mallit arvioivat melko suuren osuuden pisteistä poikkeamiksi, joten sen parametreja päätettiin tarkastella uudestaan. Lopulta *minPts*:n arvo päätettiin asettaa koulutusdatajoukon dimensioiden määrän arvoon, sillä pienemmillä arvoilla DBSCAN ei joiltakin pumppaamoilta löytänyt yhtäkään poikkeamaa. Taulukkoon 7 on koottu kunkin pumppaamon käyntiaikojen osuuksien datasta tunnistettujen poikkeamien lukumäärä ja määriteltyjen poikkeamien osuus koko datasta.

Taulukko 7: *Mallien löytämien poikkeamien lukumäärä kunkin pumppaamon käyntiaikojen osuuksista ja poikkeamien osuudet koko datasta.*

Pumppaamo	Poikkeamien lukumäärä (kpl)	Poikkeamien osuus (%)
Roope	40	11,0
Pähkinäkallio	27	7,4
Kankkula	39	10,7
Lähtösaha	9	2,5
Taivallammi	11	3,0
Suntinmäki	8	2,2
Saarenmaa	21	5,8

Valituilla DBSCAN:in parametrien arvoilla pumppaamoiden mittausdatasta tunnistettujen poikkeamien lukumäärä vaihteli välillä 8–40, joten pumppaamoiden poikkeamien osuudet koko datasta olivat 2,2–11,0 %. Osuuksien vaihteluväli oli siis huomattavasti suurempi kuin koko mittausdatajoukolle tehdyissä malleissa ja joidenkin pumppaamoiden kohdalla poikkeamien osuus aiempiin malleihin verrattuna oli moninkertainen.

Kustakin algoritmista opetetun mallin tunnistamia poikkeamia vertailtiin toisten algoritmien mallien tuloksiin. Pumppaamokohtaiset algoritmiparien löytämien yhtäläisyyksien osuudet kaikista poikkeamista löytyvät liitteestä C. Taulukkoon 8 on kerätty näiden algoritmiparien yhtäläisyyksien keskiarvot.

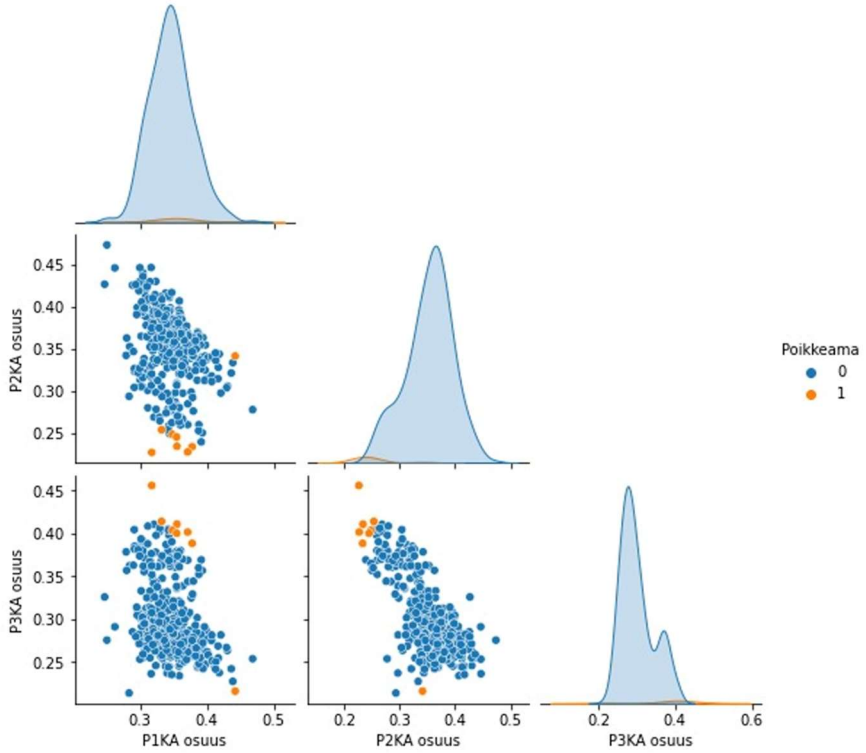
Taulukko 8: *Kaikkien pumppaamoiden yhtäläisyyksien osuuksien keskiarvo eri algoritmien tunnistamista poikkeamista.*

Vertailtavat algoritmit	Koko vuosi (%)
iForest ja LOF	82
iForest ja K-means	41
iForest ja DBSCAN	74
LOF ja K-means	33
LOF ja DBSCAN	65
K-means ja DBSCAN	27

Tässä tapauksessa iForest ja LOF löysivät siis selkeästi eniten samoja poikkeamia, yltäen yli 80 % yhtäläisyyteen. Molempien tunnistamista poikkeamista suurin osa oli myös DBSCAN:in tunnistamia. K-means klusterointi löysi taas vähiten samoja poikkeamia muiden algoritmien kanssa, päästessään vain 27–41 % keskimääräisiin yhtäläisyyksiin.

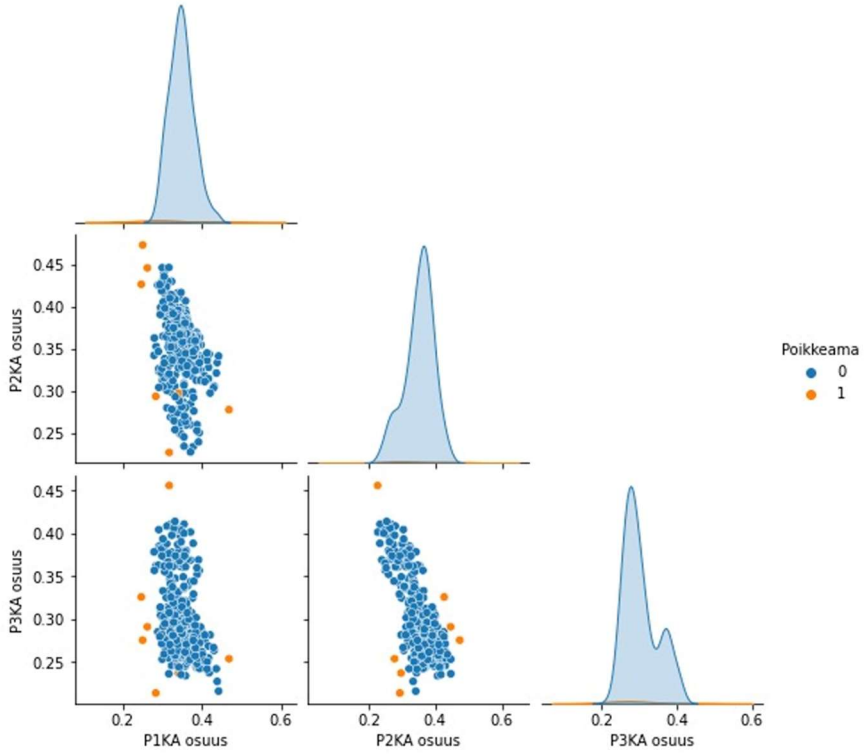
Algoritmiparien yhtäläisyydet vaihtelivat paljon myös eri pumppaamojen välillä. Erityisesti vähän poikkeamiksi tunnistettuja pisteitä sisältävissä pumppaamoissa osa algoritmipareista tunnisti muihin pumppaamoihin verrattuna huomattavasti pienemmän osuuden samoja poikkeamia. Toisaalta vähäinen poikkeamien kokonaismäärä pienentää myös todennäköisyyttä luokitella vahingossa samoja pisteitä poikkeamiksi.

Eri mallien tunnistamia poikkeamia voidaan analysoida pumppaamoiden osuuksien pistekaaviomatriisin avulla. Kaikkien mallien tuloksien pistekaaviomatriisit on kerätty liitteeseen D. Kuvassa 17 on Suntinmäen pumppaamon pistekaaviomatriisi, jossa on merkittynä K-means klusteroinnin tunnistamat poikkeamat oranssilla ja normaalit pisteet sinisellä.



Kuva 17: *Suntinmäki käyntiaikojen osuuksien pistekaaviomatriisi K-meansin poikkeamista.*

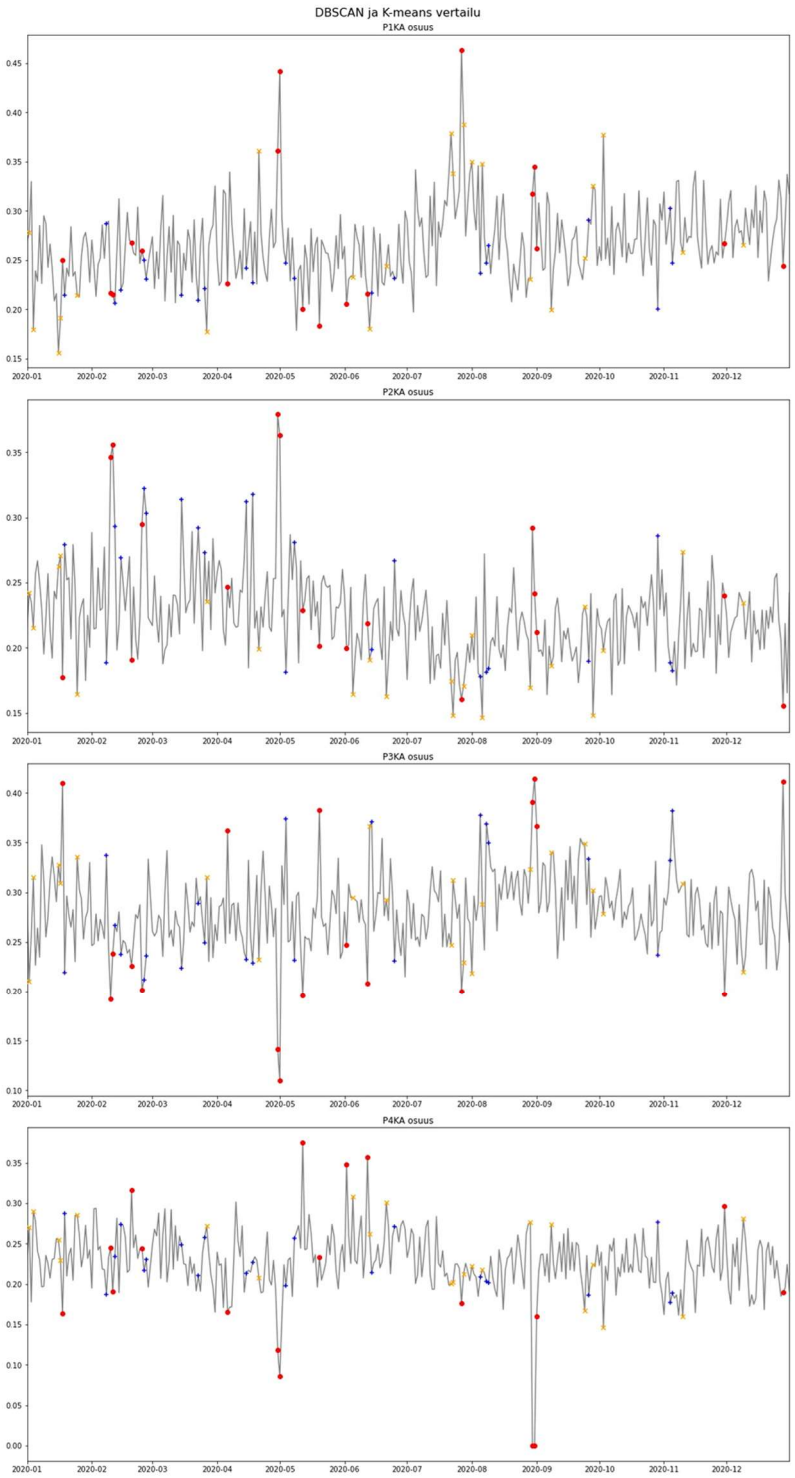
Pistekaaviomatriisista nähdään poikkeamien sijoittuminen kunkin muuttujaparin pistekuvaajaan. K-meansin tuloksista piirretystä kuvasta nähdään, että yksittäisissä pistekaavioissa suuri osa mallin tunnistamista poikkeamista sijaitsevat kussakin kuvaajassa yhdessä ryhmittymässä. Datajoukon poikkeamille ominaista tulisi olla sijoittuminen etäälle muista pisteistä. Kuvassa 18 on samasta datajoukosta opetetun DBSCAN:in mallin tulokset vastaavassa pistekaaviomatriisissa. DBSCAN:in poikkeamista suurin osa sijaitsee kuvassa kauempana datapisteiden ryhmittymistä eli ne ovat piirteidensä puolesta poikkeavia pisteitä.



Kuva 18: *Suntinmäki käyntiaikojen osuuksien pistekaaviomatriisi DBSCAN:in poikkeamista.*

Kaikkien pumppaamoiden pistekaaviomatriiseja tarkastellessa voidaan todeta iForestin ja LOF:in pistekaaviomatriisit olivat hyvin samanlaisia DBSCAN:in vastaaviin kuviin verrattuna. Pistekaaviomatriiseista nähdään, että algoritmeista DBSCAN, iForest ja LOF opetetut mallit tunnistivat pääosin kaikkien pisteparien suurimmista ryhmittymästä eroavat pisteet poikkeamiksi. K-meansin suoriutuminen pistekaaviomatriisien perusteella vaihteli suuresti pumppaamoiden välillä. Joidenkin pumppaamoiden kohdalla K-means tunnisti selkeästi kuvaajien mukaan muista poikkeavia pisteitä ja sen matriisiin merkityt tulokset vastasivat muiden algoritmien antamia tuloksia. Osalla pumppaamoista taas K-meansin tunnistamat poikkeamat sijoittuivat pääosin kuvan 17 mukaisiin ryhmittymiin.

Mallien tunnistamia poikkeamia tutkittiin myös tutustumalla pisteiden piirteiden arvoihin tarkastelemalla käyntiaikojen osuuksien kuvaajia. Kuvassa 19 on Roopen kuvaajiin merkitty sinisellä kaikki pelkän K-meansin tunnistamat poikkeamat, oranssilla pelkän DBSCAN:in tunnistamat poikkeamat ja punaisella kaikki molempien mallien tunnistamat poikkeamat.



Kuva 19: *Roopen käyntiaikojen osuuksien kuvaajat, joihin on merkitty K-meansin ja DBSCAN:in tunnistamat poikkeamat.*

Kuvasta nähdään monen molempien poikkeamaksi tunnistamien pisteiden olevan ainakin jonkin käyntiajan osuuden huomattava mittauspiikki. Eroavien poikkeamien sijoittuminen taas vaikuttaa vaihtelevan kuvaajakohtaisesti. Eri pumppaamoiden välillä eroavien poikkeamien erot olivat myös vaihtelevia. Kuvaajien perusteella Lähtösahan

pumppaamoa lukuun ottamatta DBSCAN löysi melko hyvin selkeästi suurimmat piikit käyntiaikojen osuuksissa. K-means vaikuttaa kuvaajien perusteella tässäkin tapauksessa toimineen hyvin vaihtelevasti. Esimerkiksi Kankkulan kuvaajien mukaan pelkästään K-meansin tunnistamat poikkeamat eivät ole selkeitä poikkeamia. Lähtösaha oli pumppaamoista ainoa, jossa se tunnisti selkeitä poikkeamia DBSCAN:ia paremmin.

5.3 Johtopäätökset

Tässä työssä tehtyjen huomioiden ja analyysien perusteella voidaan todeta käytettyjen algoritmien ja niistä tuotettujen mallien toimineen melko vaihtelevasti tutkituilta pumppaamoilta poikkeamia etsittäessä. Vaikka yhtäläisyyksien vertailujen perusteella tulokset vaihtelivat suuresti, voidaan tulosten perusteella piirrettyjä kuvaajia, PCA-kuvia ja pistekaaviomatriiseja tarkastelemalla todeta suuren osan työn malleista tunnistaneen poikkeamiksi selkeästi muusta datasta poikkeavia pisteitä ja selviä mittauspiikkejä. Epäselvempien ja muusta datasta vähemmän erottuvien poikkeamien tunnistaminen vaihteli hyvin paljon algoritmien välillä.

Vaikka testaukset osoittavat mallien toimimisen vaihtelevan eri pumppaamoiden välillä, voidaan yleisesti todeta K-means klusteroinnin toimivan testatuista algoritmeista huonoiten suurimmassa osassa työssä tutkituista tapauksista. Tähän johtopäätökseen päädyttiin sen perusteella, että sen mallit löysivät vähiten samoja poikkeamia muiden algoritmien mallien kanssa sekä samasta mittausdatasta eri tavoin toteutetut K-means klusteroinnin mallit tunnistivat vähiten samoja pisteitä poikkeamiksi. Myös tuloksista luotujen graafisten esitysten perusteella voidaan todeta K-means klusteroinnin mallien tunnistaneen poikkeamiksi työn tavoitteen kannalta epäoleellisia pisteitä. Vaikka joidenkin pumppaamoiden ja koulutusdatajoukkojen kanssa K-means saavuttikin työn kannalta hyviä tuloksia, ei algoritmia sen vaihtelevan suoriutumisen takia voida suositella käytettäväksi työn ongelman yleisenä ratkaisuvaihtoehtona.

Työn tapauksissa merkittävimmät erot DBSCAN:in ja iForestin käytön välillä olivat niille syötettävät parametrit malleilta saatujen tulosten muoto. DBSCAN arvioi itse sille annettujen parametrien ja datan perusteella poikkeamien määrän, joten mallin säätäminen tai toivottujen poikkeamien määrän asettelu halutulle tasolle voi olla monimutkaisempaa. Toisaalta taas DBCSAN-algoritmia käytettäessä ei tarvitse itse arvioida koulutusdatan sisältämien poikkeamien osuutta. iForest laskee kullekin pisteelle poikkeavuusarvon ja arvio pisteen poikkeamaksi kaikkien pisteiden poikkeavuusarvojen ja määritellyn poikkeamien osuuden mukaan. iForestia käytettäessä poikkeavuuden binäärisen luokittelun lisäksi voidaan hyödyntää myös poikkeavuuden suuruutta. Poikkeamien osuutta

muuttamalla pystytään myös helposti säätelemään mallin reagointiherkkyyttä, eli kuinka poikkeava pisteen täytyy olla, jotta se luokitellaan poikkeamaksi.

Vertailtaessa kausittaisten mallien eri algoritmien yhtäläisyyksiä koko vuoden datan mallien yhtäläisyyksiin, lähes kaikissa tapauksissa kausittaiset mallit löysivät enemmän samoja poikkeamia. Kausittaisen toteutuksen valintaa pohtiessa tulee kuitenkin huomioida sen toteuttamisen olevan monimutkaisempaa, vähäisemmän koulutusdatamäärän tuovan joitain mahdollisia ongelmia sekä erilaisten vuosien aiheuttavan vaihtelua mittausdatassa. Työssä käytetty kausien määrä ei myöskään ole mallien toiminnan kannalta kaikkein ideaalisimpia, vaan kausien jakamista tulisi tarkastella tarkemmin. Kausittaista toteutusta käytettäessä olisi myös hyvä tutkia, vaatiiko eri tavoin käyttäytyvät kaudet erilaisia parametrien arvoja toimiakseen ideaalisesti. Työssä käytetyillä parametreilla kausittaiset mallit luokittelivat melko suuren määrän pisteitä poikkeamiksi.

Parametreja asetettaessa arviot sopivista parametreista perustuivat yleisiin ohjeisiin tai joissain tapauksissa itse parhaiksi koettuihin arvoihin. Esimerkiksi poikkeamien osuuden määrittelevät parametrit valittiin melko mielivaltaisesti. Eri pumppaamoiden mittaukset saattavat käyttäytyä hyvin eri tavoin ja tulisi tutkia, vaatiiko optimaalisten parametrien löytäminen jokaisen pumppaamon parametrien erillistä optimointia, vai pystytäänkö kaikki parametrit optimoimaan onnistuneesti jollain tietyllä kaavalla.

Poikkeamien osuuden tai siihen vaikuttavien parametrien määritteleminen lopulliseen malliin tulee tehdä huolellisesti, sillä ne määrittävät työkalun herkkyyden. Työssä toteutetut mallit löysivät melko paljon poikkeamia ja vertailujen sekä graafisten esitysten tarkastelujen perusteella kaikki löydetyistä poikkeamista eivät todennäköisesti ole todellisia häiriötilanteisiin viittaavia poikkeamia. Oleellista olisi löytää FP- ja FN-arvojen sopiva suhde, jolloin malli tunnistaisi kaikki todelliset häiriöt, mutta ei luokittelisi normaaleja arvoja poikkeamiksi liian usein ja näin aiheuta häiritsevää määrää vääriä hälytyksiä.

Työn tekemistä varten käytössä olevan materiaalin ja työn aikana tehtyjen selvitysten perusteella ei voida todeta löydettyjen poikkeamien vastaavan varmasti kaikkia pumppaamoiden toiminnan kannalta oleellisia häiriöitä ja muita poikkeavia tilanteita. Tuloksia syvemmin tarkastellessa pystyttiin kuitenkin toteamaan mallien tunnistaneen poikkeamiksi joitakin hyvin todennäköisiä häiriötilanteita. Näin ollen voidaan todeta löydettyjen poikkeamien olevan jossain määrin juuri näitä pumppaamon toiminnan kannalta kiinnostavia pisteitä. Työkalun jatkokehitystä suunnitellessa on kuitenkin suositeltavaa tutustua mallien tunnistamiin poikkeamiin vielä yksityiskohtaisemmin ja analysoida tarkemmin eri mittausten roolia häiriöiden havainnoimisessa ja eri piirteiden vaikutuksista mallien toimintaan.

5.4 Soveltaminen käytäntöön

Työssä toteutettu työkalu on laadittu analysointi- ja arviointikäyttöön, joten se sisältää useita puutteita eikä ole käytettävissä suoraan todellisessa järjestelmässä. Työkalun toteuttamisen yhteydessä ja siitä saatujen tulosten perusteella pystyttiin kuitenkin pohtimaan erilaisia vaatimuksia ja kehityskohteita sekä ratkottavia ongelmia, jotka tulisi selvittää parempien tulosten saavuttamiseksi ennen lopullisen työkalun toteuttamista ja käyttöönottoa.

Koneoppivien mallien käyttö reaaliaikaisessa järjestelmässä vaatii tiedonvaihtoa mallin suorittavan ohjelman sekä mittausdatan raportointityökalun välille. Eri sovellusten välille on rakennettava rajapinta, jonka kautta reaaliaikainen mittausdata siirretään raportointijärjestelmästä mallin käsiteltäväksi. Mallin antamat tulokset voidaan lähettää joko takaisin raportointityökaluun tai ne voidaan esittää jossain muussa käytössä olevassa käyttöliittymässä.

Datan siirtäminen suoritettiin tässä työssä manuaalisesti, mutta reaaliaikaisissa järjestelmissä datan siirtymisen pitäisi tapahtua automaattisesti. Mallissa käytetyn datan taajuuden takia työkalun käyttö ei vaadi jatkuvaa tiedonvaihtoa, vaan tässä tapauksessa voitaisiin esimerkiksi tunnin välein suorittaa automaattinen kaikkien mittausdatojen siirto raportointityökalusta malleille käytettäväksi.

Työssä osa datan esikäsittelystä suoritettiin Microsoft Excelin avulla. Tämä tapa ei kuitenkaan sovellu kovin hyvin reaaliaikajärjestelmiin, vaan datan esikäsittely tulisi suorittaa automaattisesti esimerkiksi pythonin datankäsittelykirjastojen avulla. Käsittely tulee automatisoida, jotta mallin käyttö ei vaadi jatkuvaa manuaalista työtä.

Työn toisessa osuudessa käytettiin pumppujen käyntiaikojen osuuksien päiväkohtaisia keskiarvoja mallien luomiseen. Liian lyhyen aikavälin osuuksia tarkastellessa näkyy niiden arvoissa normaalin pumppujen vuorottelun aiheuttamat vaihtelut liian suurina. Käyntiaikojen osuuksille tulee laskea arvot tarpeeksi pitkän ajanjakson ajalta, jotta vuorottelukäytön aiheuttamat hetkittäiset vaihtelut tasaantuvat ja häiriöistä johtuvat pidemmät käyntiaikojen muutokset erottuvat selkeämmin. Tarkasteltavan ajanjakson pituus vaikuttaa myös suoraan mallin mahdolliseen reagointinopeuteen. Reaaliaikaisessa järjestelmässä mallin suoritus voidaan toteuttaa esimerkiksi joko niin, että se tarkastelee tietyin väliajoin viimeisen vuorokauden tai muun ajanjakson käyntiaikojen osuuksia tai suorittaa tarkastuksen aina tietyn ajanjakson välein edellisen tarkastuksen jälkeen kerätylle datalle.

Työkalun käyttöönottoa suunnitellessa on ehkä hyvä tarkastella, onko kaikissa tapauksissa järkevää käyttää raskaampia koneoppimismalleja, jos esimerkiksi raja-

arvotarkasteluiden avulla voitaisiin hoitaa sama asia. Esimerkiksi pumppujen käyntiajoille ja niiden osuuksille voisi asettaa hälytyksen aiheuttavat raja-arvot pumppujen määrän perusteella. Tällöin kuitenkin olisi hyvä tarkastella ja määrittää erikseen kaikki mahdolliset häiriöiden tuottamat merkit mittausdatassa ja muodostaa jokaisesta oma sääntönsä.

Käytössä olevan mallin ylläpitäminen ja päivitys vaativat sitoutumista. Koneoppimisen käyttämisessä vaarana on, että malli opetetaan vain kerran käyttöönoton yhteydessä ja sen säännöllinen päivitys ja uudelleenkoulutus jäävät käytännössä toteuttamatta. Yksi vaihtoehto voisi olla mallin automaattinen uudelleenkouluttaminen määrätyin väliajoin päivitetyllä opetusdatajoukolla. Pumppaamolta saadaan koko ajan lisää mittausdataa ja näin käytettävän opetusdatan määrä kasvaa ajan kuluessa, joten säännöllinen päivitys suuremmalla määrällä dataa saattaa myös parantaa mallin toimintaa. Erilaiset muutokset pumppaamossa ja sen toiminnassa voivat vaikuttaa merkittävästi mittausdatoihin ja sitä kautta mallin toimimiseen, jolloin malli on opetettava uudelleen muutoksen jälkeistä dataa käyttäen. Heti muutosten jälkeen tämä on käytännössä mahdotonta, mutta säännöllinen koulutus voisi osin auttaa tässäkin ongelmassa, kun saadaan kerättyä uutta dataa yhä enemmän ja malli vastaamaan muuttunutta tilannetta.

Suunnitellessa poikkeamien tunnistamisen käyttöä osana kunnonvalvontaa tulee muistaa, että kaikki poikkeamatilanteet eivät välttämättä ole oikeita häiriöitä pumppaamoiden toiminnassa. Erilaiset normaalista poikkeavat ympäristöolosuhteet tai muut ulkoiset tekijät saattavat näkyä pumppaamoiden mittausdatassa poikkeamina. Toisaalta taas työkalu ei osaa erotella eri häiriötilanteita toisistaan tai osoittaa häiriön perimmäistä aiheuttajaa. Näin ollen lopullinen tilanteen arviointi ja kunnossapitopäätöksen tekeminen jää aina operaattorille.

Työkalun kehittäminen tarjoaa laajempia ja edistyksellisempiä mahdollisuuksia häiriötilanteiden tunnistamiseen ja tarkempaan diagnosointiin. Työssä suunniteltu työkalu ei arvioi tunnistettujen poikkeamien tyyppiä tai aiheuttajaa. Ohjattujen poikkeamien tunnistamisen tapojen käyttäminen mahdollistaisi poikkeamien ja häiriöiden tarkemman arvioinnin ja luokittelemisen mallin avulla. Jätevedenpumppaamon toiminnan yksityiskohtaisen tunteminen ja esimerkiksi käytössä tunnistettujen poikkeamien arvioiminen ja häiriöiden laatujen tai syiden nimeäminen voisivat mahdollistaa ohjatun poikkeamien tunnistamisen mallin luomisen. Seurannan aikaista mittausdataa ja siihen merkittyjä poikkeamia ja niiden tarkempia luokituksia voitaisiin käyttää ohjatun poikkeamien tunnistamisen koulutusdatana. Tämä kuitenkin vaatisi operaattoreilta ja kunnossapitotyöntekijöiltä tarkkoja ja yhtenäisiä merkintäkäytäntöjä tapahtuvista poikkeamista ja häiriöistä. Seurannan täytyisi tapahtua pidemmän aikaa, jotta mahdollisimman paljon erilaisia häiriöitä pääsisi

esiintymään ja niiden vaikutukset mittausdataan saataisiin tallennettua. Vastaavanlainen tarkempi seuranta olisi toteutettava jokaiselle laitokselle erikseen, jotta jokaista mallia varten saataisiin kohdekohtaiset koulutusdatat. Tämä taas lisää huomattavasti toteutukseen vaadittavaa työmäärää.

6. YHTEENVETO

Tämän työn tarkoituksena oli selvittää, miten pumppaamoiden mittausdatasta voidaan koneoppimisen avulla tunnistaa häiriötilanteita pumppaamoiden toiminnassa. Häiriöiden tiedetään olevan poikkeavia ja harvinaisia tilanteita pumppaamon normaalin toiminnan seassa ja vaikuttavan pumppaamolta saatuun mittausdataan, joten työn lähtökohtana oli tutkia erilaisia poikkeamien tunnistamisen menetelmiä tunnistamaan mittausdatan poikkeavaa käyttäytymistä. Aiemmin vikatilanteita on tunnistettu tarkastelemalla mittausten arvoja ja niistä luotuja graafisia esityksiä. Mahdollisten häiriöiden havaitseminen haluttiin automatisoida ja näin siirtää osa operaattorien tekemästä manuaalisesta tarkastelutyöstä koneoppimista hyödyntävälle mallille sekä mahdollistaa aikaisempi vikojen havaitseminen. Tarkempi poikkeaman analysointi ja päätös sen vaatimista toimenpiteistä jää edelleen operaattoreille.

Työn toteuttamista varten käytettiin seitsemän eri jätevedenpumppaamon mittausten historiadataa. Pumppaamoilla aiemmin tapahtuneista häiriöistä ei ole saatavilla päteviä merkintöjä tai tietoja tarkoista tapahtumahetkistä, joten poikkeavien tilanteiden tunnistamiseen on käytettävä ohjaamattoman oppimisen menetelmiä. Tutkitun kirjallisuuden, tarkasteltujen menetelmien ominaisuuksien, käyttötarkoituksen asettamien ehtojen sekä käytössä olevan datan ja sen tuomien rajoitusten perusteella päädyttiin käyttämään neljää eri ohjaamattoman oppimisen koneoppimisalgoritmia poikkeamien tunnistamiseen. Valitut algoritmit olivat DBSCAN, K-means klusterointi, isolation forest ja local outlier factor.

Ohjaamattoman koneoppimisen poikkeamien tunnistamisen menetelmät ovat melko suoraviivaisia toteuttaa. Tärkeimpänä tuloksiin vaikuttavana vaiheena voidaan pitää käytettävän koulutusdatajoukon esikäsittelyä sekä koulutukseen käytettävien piirteiden valintaan. Mallien toteutus ja analysointi tapahtui Python-ohjelmointikielen ja sen valmiiden kirjastojen avulla. Työkalu toteutettiin siten, että se käsittelee kerralla yhdeltä pumppaamolta saadun opetusdatajoukon, laskee kyseiselle datajoukolle parametrit, kouluttaa jokaisesta algoritmista omat mallit, kerää kunkin mallin tunnistamat poikkeamat omiin listoihin, laskee listoille yhtäläisyydet sekä piirtää halutut graafiset esitykset.

Yksi ohjaamattomaan oppimiseen liittyvistä haasteista oli algoritmien suoriutumisen arvioiminen. Sen malleille ei voida laskea selkeitä suorituskykyä kuvaavia tunnuslukuja, vaan mallien validointi sekä toimivuuden analysointi on tehtävä muilla keinoin. Algoritmien ja niistä muodostettujen mallien arvioiminen päätettiin suorittaa samoista

datajoukoista eri tavoilla toteutettujen mallien tuloksia vertailemalla sekä tuloksista piirrettyjä kuvaajia, PCA-kuvia ja muuttujaparien pistekaaviomatriiseja tarkastelemalla.

Eri mallien tuloksien yhtäläisyyksien suuristakin vaihteluista huolimatta kuvaajien ja PCA-kuvien perusteella suurin osa malleista tunnisti datan joukosta kaikkein selkeimmät poikkeamat melko hyvin. Suurimmat erot tuloksissa johtuivat pääosin siis vähemmän selkeiden poikkeamien tunnistamisesta. Tulosten yhtäläisyyksien sekä kuvien tarkastelun perusteella voidaan päätellä DBSCAN-klusteroinnin ja iForestin toimineen työn tapauksissa parhaiten. Ne saivat useimmissa tapauksissa korkeimpia yhtäläisyysprosentteja muiden mallien kanssa sekä kuvien perusteella tunnistivat poikkeamiksi hyvin muihin verrattuna poikkeavasti käyttäytyvät pisteet.

DBSCAN:in ja iForestin mallien toteutuksen kannalta suurimpina eroina ovat niiden poikkeamien osuuden määrittelytavat ja tätä eroa voisikin käyttää yhtenä lopullisen algoritmin valintaperusteena. iForestille tulee määritellä parametrina koulutusdatajoukon poikkeamien osuus, kun taas DBSCAN päättelee poikkeamien osuuden sille annettujen muiden parametrien perusteella. Toisaalta taas iForestin jokaiselle pisteelle laskemat poikkeavuusarvot mahdollistavat pisteiden poikkeavuuden tarkemman tason määrittelemisen helposti.

Edellisten kahden algoritmin kanssa melkein yhtä hyvin toimi myös LOF. Eniten muiden algoritmien tuloksista erosivat K-means klusteroinnin mallien tuottamat tulokset. Lähes poikkeuksetta siihen kohdistuvat vertailut löysivät vähiten yhtäläisyyksiä tunnistetuista poikkeamista. Sen suoriutuminen myös vaihteli suuresti koulutusdatajoukon mukaan. K-meansin tuloksista piirrettäviä kuvia analysoitaessa voidaan todeta sen joidenkin mallien jättäneen tunnistamatta todella selkeitäkin poikkeamia.

Vaikka koneoppimista voidaan pitää helposti yleispätevänä ratkaisuna kaikkeen, saattaa toimivan koneoppimistyökalun luominen vaatia hyvin tarkkaa räätälöintiä juuri kyseiseen tapaukseen sopivaksi ja pienetkin muutokset pumppaamon toiminnassa tai mittausarvoissa voivat vaikuttaa mallin toimimiseen. Vaikka työn pumppaamoiden toimintaperiaatteet ovat hyvin samanlaisia, eroavat niistä saadut mittaukset sekä mittausten arvot. Jokainen erillinen koulutusdatajoukko vaatii yksilöllistä esikäsittelyä ja algoritmien käyttämät parametrit tulee optimoida aina tapauskohtaisesti. Osa parametreista on toki optimoitavissa yleisten ohjeiden avulla esimerkiksi datan ominaisuuksien perusteella. Pumppaamoiden mittausdatat sisältävät myös eri määrän häiriöitä tarkasteluajanjaksolla, joten mallien reagointiherkkyyden tarkempi säätö voi vaatia sen yksityiskohtaisempaa perehtymistä. Käytössä olevan työkalun tarkka säätäminen on sen toimimisen kannalta

tärkeää, jotta malli löytäisi kaikki oleelliset häiriöistä kertovat poikkeamat, mutta ei aiheuttaisi suurissa määrin vääriä hälytyksiä

Työssä löydettiin mahdollisia algoritmeja käytettäväksi mittausdatan poikkeamia reaaliaikaisesti tunnistavan järjestelmän toteuttamiseksi ja muodostettiin työkalu, jonka avulla malleja luotiin ja niiden antamia tuloksia arvioitiin. Työssä tehtiin joitakin mallien luomiseen sekä työkalun toteuttamiseen liittyviä huomioita sekä haasteita. Luotua työkalua voitaisiin pyrkiä edelleen kehittämään esimerkiksi arvioimalla tarkemmin eri häiriöiden heijastumista mittauksiin sekä tekemällä arviointien perusteella tarkemmat valinnat mallin koulutukseen käytettävistä piirteistä.

LÄHTEET

[1] Abuzaid AH. Detection of outliers in univariate circular data by means of the outlier local factor (LOF). Statistics in Transition New Series 2020; 21: 39–51.

[2] Aggarwal CC. Outlier analysis. 2nd ed. Cham: Springer Nature, 2016.

[3] Ahmed M, Mahmood AN, Islam MR. A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems 2016; 55: 278–288.

[4] Ahmed U, Suresh KM. Hands-On Exploratory Data Analysis with Python. Packt Publishing, 2020.

[5] Alla S, Adari SK. Beginning Anomaly Detection Using Python-Based Deep Learning: With Keras and Pytorch. Berkeley, CA: Apress L. P, 2019.

[6] Angelopoulos A, Michailidis ET, Nomikos N, et al. Tackling Faults in the Industry 4.0 Era-A Survey of Machine-Learning Solutions and Key Aspects. Sensors (Basel, Switzerland); 2019; 20: 109.

[7] Bakshi K, Bakshi K. Considerations for artificial intelligence and machine learning: Approaches and use cases. AERO 2018; 1–9.

[8] Berthold MR, Borgelt C, Höppner F, et al. Guide to Intelligent Data Analysis How to Intelligently Make Sense of Real Data. 1st ed. London: Springer London, 2010. Epub ahead of print 2010. DOI: 10.1007/978-1-84882-260-3.

[9] Bishop CM. Pattern recognition and machine learning. New York: Springer, 2006.

[10] Bloch HP. Pump Wisdom: Problem Solving for Operators and Specialists. 1st; 1st ed. Hoboken: Wiley, 2011. Epub ahead of print 2011. DOI: 10.1002/9781118056745.

[11] Breunig MM, Kriegel H-P, Ng RT, et al. LOF: Identifying Density-Based Local Outliers.

[12] Chandola V, Banerjee A, Kumar V. Anomaly detection. ACM computing surveys 2009; 41: 1–58.

[13] Cielen D, Meysman A, Ali M. Introducing data science : big data, machine learning, and more, using Python tools. 1st ed. Shelter Island, NY: Manning Publications, 2016.

[14] Deshpande B, Kotu V. Data Science, 2nd Edition. Morgan Kaufmann, 2018.

[15] Ding C, He X. K-Means Clustering via Principal Component Analysis. In: Proceedings of the Twenty-First International Conference on Machine Learning. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 29.

[16] Elasha F, Shanbr S, Li X, et al. Prognosis of a Wind Turbine Gearbox Bearing Using Supervised Machine Learning. Sensors (Basel, Switzerland); 2019; 19: 3092.

[17] Esposito D, Esposito F. Introducing Machine Learning. Pearson Education, https://learning.oreilly.com/library/view/introducing-machine-learning/9780135588338/ (2020).

[18] Ester, M, Kriegel, H P, Sander, J, and Xiaowei, Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. United States: N. p., 1996.

[19] Et-taleby A, Boussetta M, Benslimane M. Faults Detection for Photovoltaic Field Based on K-Means, Elbow, and Average Silhouette Techniques through the Segmentation of a

Thermal Image. International journal of photoenergy; 2020. Epub ahead of print 2020. DOI: 10.1155/2020/6617597.

[20] Flach P. Machine learning the art and science of algorithms that make sense of data. Cambridge: Cambridge University Press, 2012.

[21] Géron A. Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems. Sebastopol, California: O'Reilly Media, Inc, 2017.

[22] Giannoni F, Mancini M, Marinelli F. Anomaly detection models for IoT time series data Data Mining -Project Report. ArXiv, abs/1812.00890, 2018

[23] Han J, Pei J, Kamber M. Data Mining: Concepts and Techniques. Saint Louis: Elsevier Science & Technology, 2011. Epub ahead of print 2011. DOI: 10.1016/C2009-0-61819-5.

[24] Hanna P, Swartling E. Anomaly Detection in Time Series Data using Unsupervised Machine Learning Methods A Clustering-Based Approach. KTH Royal Institute of Technology, 2020.

[25] Härkönen J. Kaivosten kunnossapitojärjestelmät. Oulun ammattikorkeakoulu, 2014.

[26] Higgs PA, Parkin R, Jackson M, et al. A survey on condition monitoring systems in industry. In: Proceedings of the 7th Biennial Conference on Engineering Systems Design and Analysis, ESDA 2004. 2004. p. 163–78.

[27] Honda K, Notsu A, Ichihashi H. Fuzzy PCA-Guided Robust k-Means Clustering. IEEE Transactions on Fuzzy Systems 2010; 18: 67–79.

[28] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of educational psychology 1933; 24: 417–441.

[29] Hu W, Liao Y, Vemuri R. Robust Anomaly Detection Using Support Vector Machines. Proceedings of the International Conference on Machine Learning, 2003.

[30] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing 2006; 20: 1483–1510.

[31] Jinka P, Schwartz B. Anomaly Detection for Monitoring. 1st ed. O'Reilly Media, Inc, 2016.

[32] Jolliffe IT. Principal component analysis. 2nd ed. New York: Springer, 2002.

[33] Kang M. 6 Machine Learning: Anomaly Detection. In: Pecht MG, Kang M (eds) Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things. John Wiley & Sons, Incorporated, 2018, pp. 131–162.

[34] Kansanaho J. Data-Driven Methods for Diagnostics of Rolling Element Bearings. Jyväskylän yliopisto, http://urn.fi/URN:ISBN:978-951-39-7936-2 (2019).

[35] Karttunen E. Vesihuoltoverkkojen suunnittelu. 1, Perusteet ja toiminnallisuus. Helsinki: Suomen rakennusinsinöörien liitto RIL, 2010.

[36] Karttunen E. Vesihuoltoverkkojen suunnittelu. 2, Mitoitus ja suunnittelu. Helsinki: Suomen rakennusinsinöörien liitto RIL, 2010.

[37] Karttunen E, Tuhkanen T. Vesihuolto. 1. Helsinki: Suomen rakennusinsinöörien liitto RIL, 2003.

[38] Karttunen E, Tuhkanen T, Kiuru H. Vesihuolto. 2. Helsinki: Suomen rakennusinsinöörien liitto RIL, 2004.

[39] Korving H, Clemens FHLR, van Noortwijk M. J. Statistical Modeling of the Serviceability of Sewage Pumps. Journal of hydraulic engineering (New York, NY) 2006; 132: 1076–1085.

[40] Kumpulainen P. Anomaly detection for communication network monitoring applications. Tampere University of Technology, 2014.

[41] Liu FT, Kai MT, Zhou Z-H. Isolation Forest. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008, pp. 413–422.

[42] Liu FT, Ting KM, Zhou Z-H. Isolation-Based Anomaly Detection. ACM transactions on knowledge discovery from data 2012; 6: 1–39.

[43] Liu H. Feature Engineering for Machine Learning and Data Analytics. 1st ed. Milton: CRC Press, 2018. Epub ahead of print 2018. DOI: 10.1201/9781315181080.

[44] Lo D, Cheng H, Han J, et al. Classification of software behaviors for failure detection: a discriminative pattern mining approach. International Conference on Knowledge Discovery and Data Mining 2009; 557–566.

[45] Malik A, Tuckfield B. Applied unsupervised learning with R : uncover hidden relationships and patterns with K-Means clustering, hierarchical clustering, and PCA. 1st ed. Birmingham: Packt Publishing Ltd, 2019.

[46] Meleshko A v, Desnitsky VA, Kotenko I v. Machine learning based approach to detection of anomalous data from sensors in cyber-physical water supply systems. IOP conference seriesMaterials Science and Engineering; IOP ConfSer: MaterSciEng 2020; 709: 33034.

[47] Miszta-Kruk K. Reliability and failure rate analysis of pressure, vacuum and gravity sewer systems based on operating data. Engineering Failure Analysis 2016; 61: 37–45.

[48] Müller AC, Guido S. Introduction to machine learning with Python a guide for data scientists. 1st ed. Sebastopol, Calif: O'Reilly Media, 2017.

[49] Nelli F. Python Data Analytics: Data Analysis and Science Using Pandas, Matplotlib and the Python Programming Language. 1st ed. Berkeley, CA: Apress L. P, 2015.

[50] Nohynek P, Lumme VE. Kunnonvalvonnan värähtelymittaukset. 2nd ed. Rajamäki: KP-Media, 2004.

[51] Nykyri M. Data analytics for predictive maintenance in a pulp mill — case electric motors. Lappeenranta University of Technology, 2018.

[52] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research; 12.

[53] Pileggi V, Budziakowski J, Manoharan M, et al. Design Guidelines For Sewage Works, https://www.ontario.ca/document/design-guidelines-sewage-works-0, 2016.

[54] Preiss BR. Data structures and algorithms with object-oriented design patterns in Java. New York (NY): Wiley, 1999.

[55] Salama MA, Hassanien AE, Fahmy AA. Reducing the influence of normalization on data classification. CISIM 2010; 609–613.

[56] Sander J, Ester M, Kriegel H-P, et al. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data mining and knowledge discovery 1998; 2: 169–194.

[57] Schubert E, Sander J org, Ester M, et al. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM TransDatabase Syst; 42. Epub ahead of print 2017. DOI: 10.1145/3068335.

[58] Schutt R, O'Neil C. Doing Data Science: Straight Talk from the Frontline. Sebastopol: O'Reilly Media, Incorporated, 2013.

[59] Singh S, Shiv P, Ahmed A. Outlier Modeling in Gear Bearing Using Autoencoder for Remaining Useful Life Prediction. Epub ahead of print 2019. DOI: 10.20944/preprints201907.0112.v1.

[60] Sipponen J. Vesihuoltolaitoksen jätevedenpumppaamojen saneeraus. Hämeen ammattikorkeakoulu, 2014.

[61] Stamboliska Z, Rusiński E, Moczko P. Proactive Condition Monitoring of Low-Speed Machines. 1st ed. Cham: Springer International Publishing, 2015. Epub ahead of print 2015. DOI: 10.1007/978-3-319-10494-2.

[62] Stetco A, Dinmohammadi F, Zhao X, et al. Machine learning methods for wind turbine condition monitoring: A review. Renewable Energy 2019; 133: 620–635.

[63] Susto GA, Schirru A, Pampuri S, et al. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. - IEEE Transactions on Industrial Informatics 2015; 11: 812–820.

[64] Swarbrick B. Multivariate Data Analysis for Dummies. England: John Wiley & Sons, Ltd, 2012.

[65] Syakur MA, Khotimah BK, Rochman EMS, et al. Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster. IOP Conf.Ser.: Mater.Sci.Eng 2018; 336: 12017.

[66] Tashman Z, Gorder C, Parthasarathy S, et al. Anomaly Detection System for Water Networks in Northern Ethiopia Using Bayesian Inference. Sustainability (Basel, Switzerland) 2020; 12: 2897.

[67] Xu R, Wunsch DC. Clustering. Piscataway, New Jersey: IEEE Press, 2015. Epub ahead of print 2015. DOI: 10.1002/9780470382776.

[68] Zheng A. Evaluating Machine Learning Models. 1st ed. O'Reilly Media, Inc, 2015.

[69] Zhou H, Wang P, Li H. Research on adaptive parameters determination in DBSCAN algorithm. Journal of information and computational science 2012; 9: 1967–1973.

[70] Pumppaamoiden suunnittelu, käyttö ja huolto. Parhaat menettelytavat Vantaanjoen valuma-alueella. Helsingin seudun ympäristöpalvelut HSY, 2020

[71] Kunnossapito. Käsitteet ja määritelmät., PSK Standardointi, PSK 6201, 2011, 30 s.

[72] Kunnonvalvonnan värähtelymittaus. Käsitteet ja määritelmät. Käytettävät suureet ja mittayksiköt, PSK Standardointi, PSK 5701, 2017, 15 s.

[73] Heartbeat tehnologia. 2021, Saatavissa (viitattu 10.3.2021): https://www.fi.endress.com/fi/automaatioprosessin-hallinta-kentt%C3%A4laitteet/innovatiivinen-mittausteknologia/Heartbeat-teknologia-mittauksen-pulssilla.

[74] sklearn.neighbors.LocalOutlierFactor — scikit-learn 0.24.2 documentation. 2021, Saatavissa (viitattu 3.8.2021): https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html

[75] sklearn.preprocessing.StandardScaler — scikit-learn 0.24.2 documentation. 2021, Saatavissa (viitattu 20.7.2021): https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

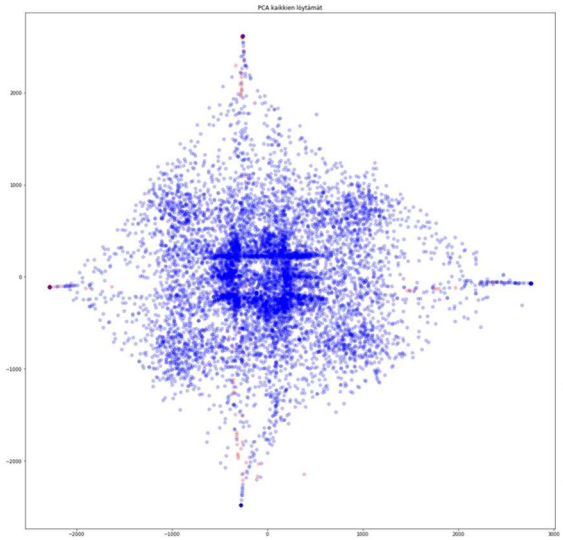
[76] sklearn.ensemble.IsolationForest — scikit-learn 0.24.2 documentation. 2021, Saatavissa (viitattu 2.8.2021): https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

LIITE A: KOKO VUODEN JA KAUSITTAISTEN MALLIEN LÖYTÄMIEN YHTÄLÄISYYKSIEN OSUUDET

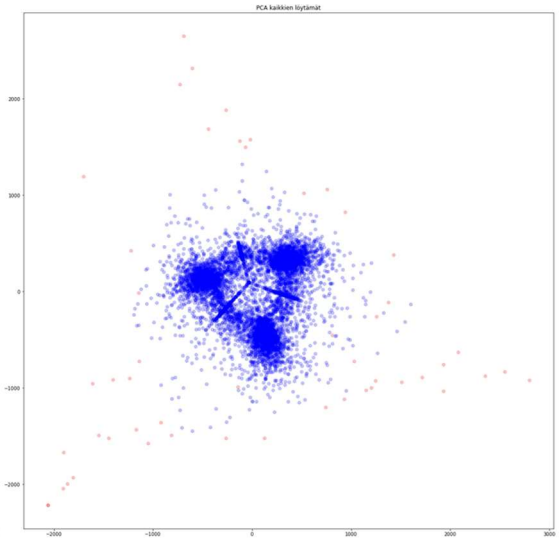
	Vertailu	Roope	Pähkinäkallio	Kankkula	Lähtösaha	Taivallammi	Suntinmäki	Saarenmaa
Koko vuosi	iForest - LOF	64	23	68	39	62	77	49
	iForest - Kmeans	51	50	22	6	11	69	41
	iForest - DBSCAN	59	85	91	76	72	77	57
	LOF - Kmeans	51	9	19	6	5	57	25
	LOF - DBSCAN	72	23	71	43	81	82	72
	Kmeans - DBSCAN	30	46	21	6	5	54	29
Kausittaiset keskimäärin	iForest - LOF	62	75	70	65	59	58	58
	iForest - Kmeans	48	28	34	39	36	39	40
	iForest - DBSCAN	66	80	78	76	70	66	63
	LOF - Kmeans	41	31	32	38	32	32	31
	LOF - DBSCAN	79	81	73	74	73	77	79
	Kmeans - DBSCAN	39	30	30	36	34	32	31
Vertailu koko vuosi vs. kaikki kausittaiset (laskettuna kaikista)	iForest	64	48	58	58	45	45	45
	LOF	63	25	55	37	39	43	44
	Kmeans	66	41	20	1	3	28	22
	DBSCAN	59	52	59	55	48	40	45
Vertailu koko vuosi vs. kaikki kausittaiset (pienemmästä määrästä laskettuna)	iForest	86	85	90	88	56	100	81
	LOF	84	44	86	57	50	95	78
	Kmeans	88	73	31	1	4	61	39
	DBSCAN	79	92	91	84	61	89	81
Poikkeamia	Kausittain yhteensä (kpl)	471	455	462	218	290	391	601
	Koko vuosi (kpl)	282	179	218	446	187	114	234

LIITE B: KAIKKIEN MALLIEN TUNNISTAMAT POIKKEAMAT PCA-KUVISSA

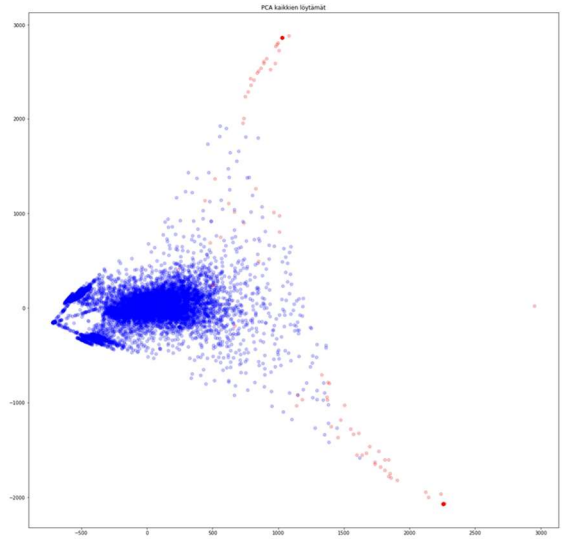
Roope



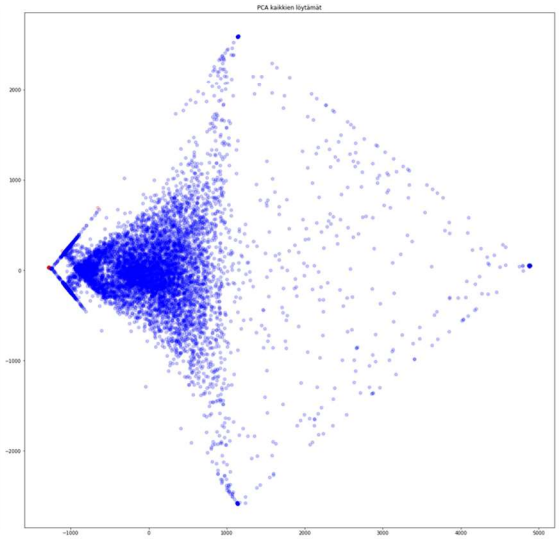
Pähkinäkallio



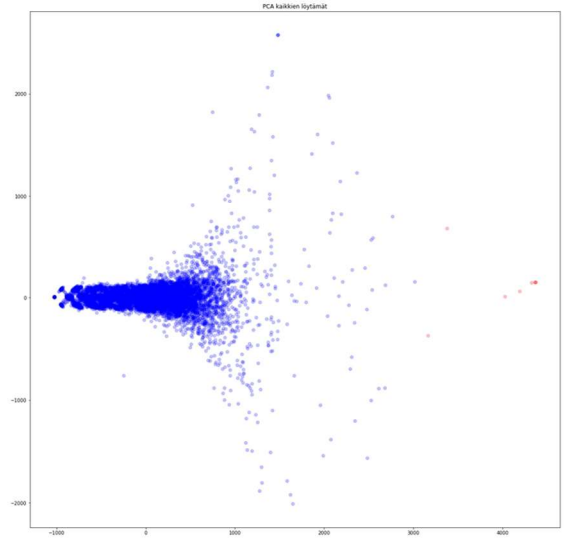
Kankkula



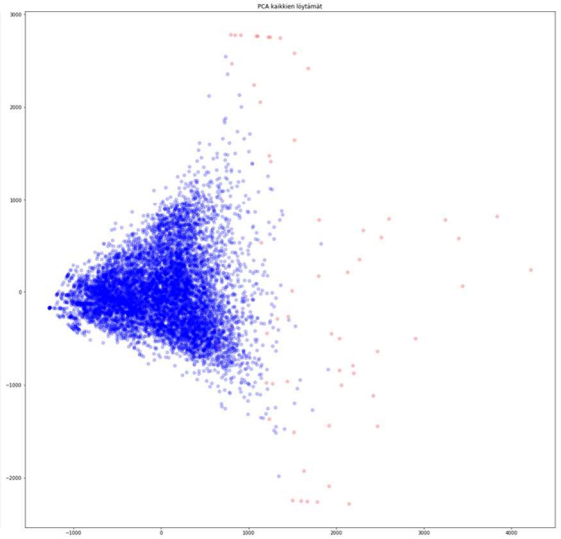
Lähtösaha

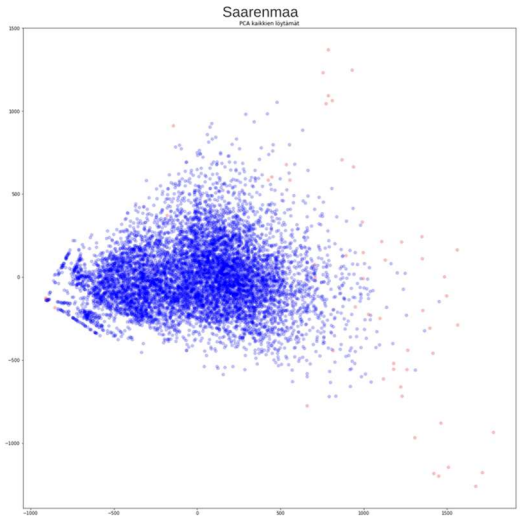


Taivallammi



Suntinmäki

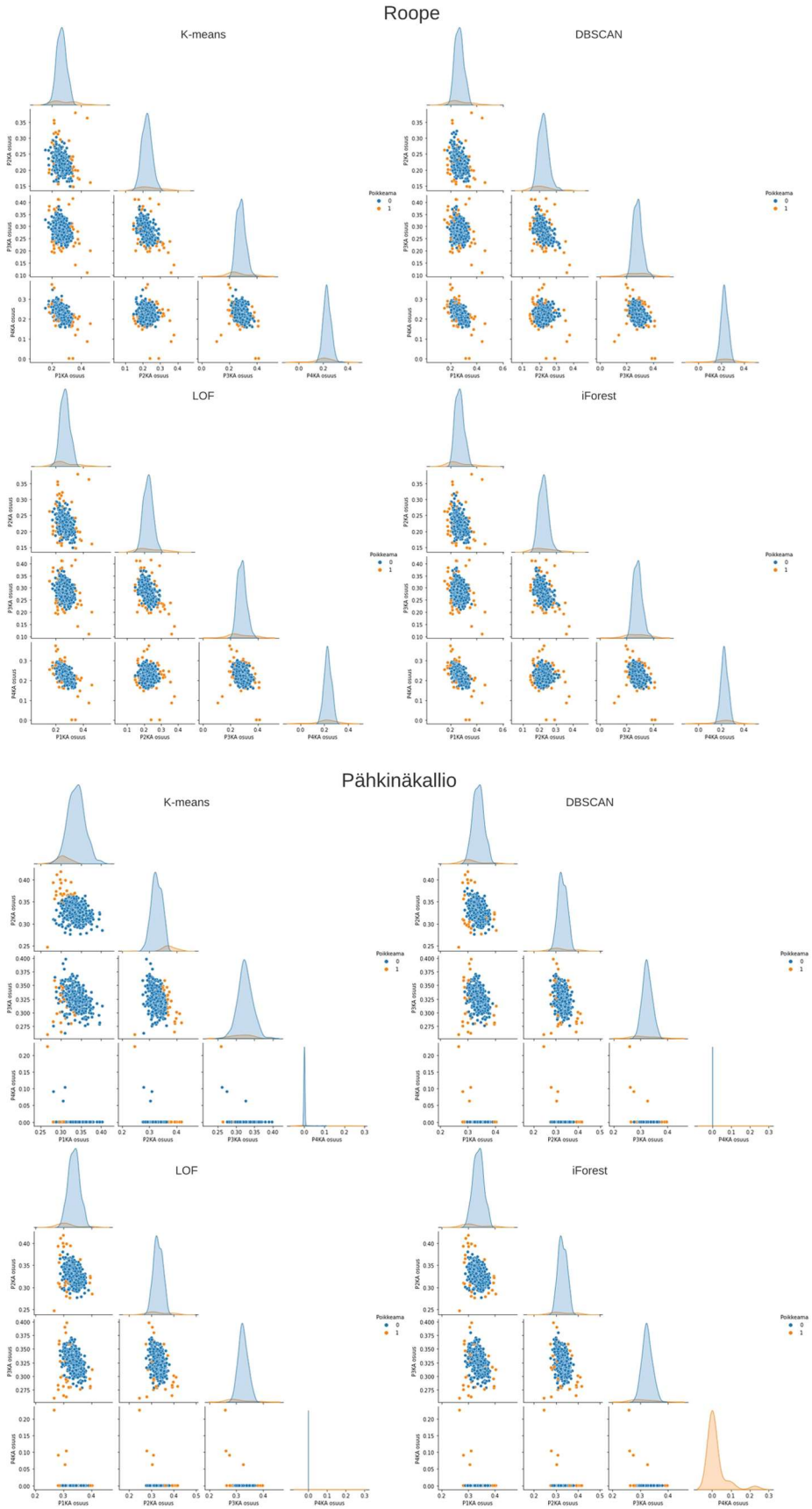


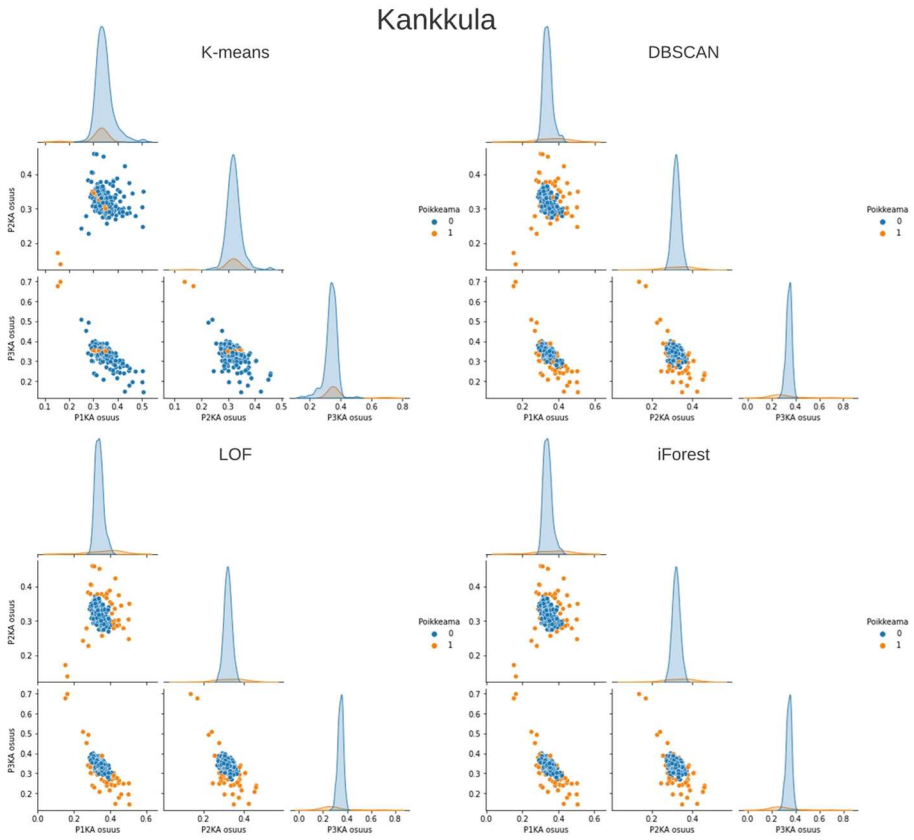

Saarenmaa
PCA kaikkien löytämät
1500
1000
500
0
−500
−1000
−1000
−500
0
500
1000
1500

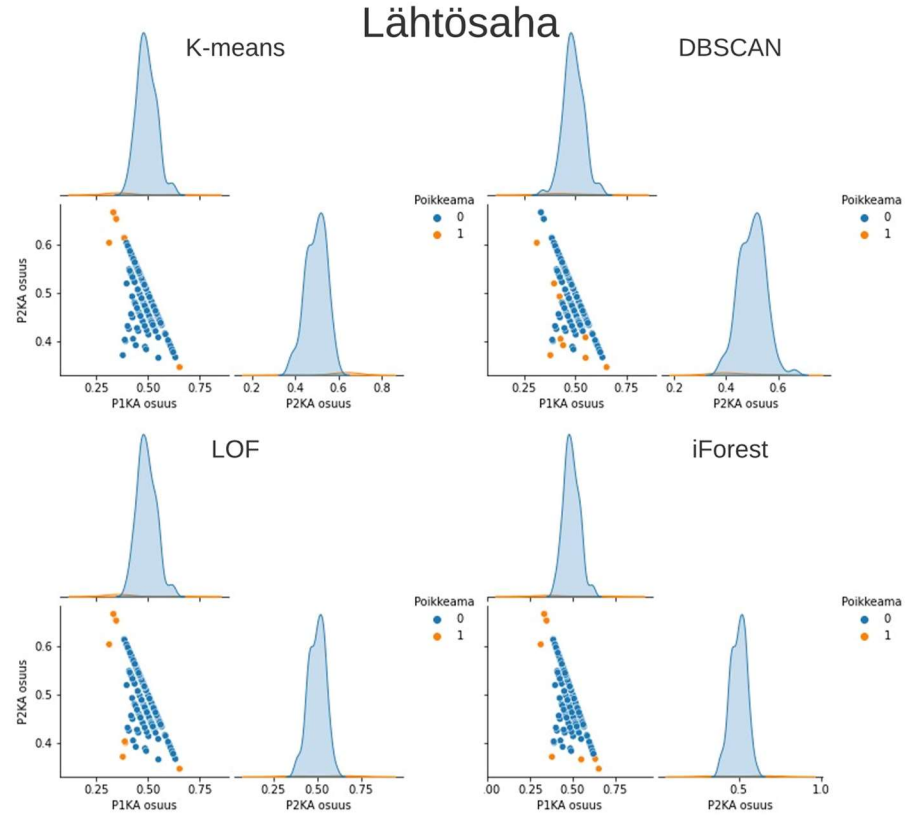
LIITE C: KÄYNTIAIKOJEN OSUUKSIEN MALLIEN PUMPPAAMOKOHTAISET ALGORITMIPARIEN YHTÄLÄISYYKSIEN PROSENTUAALISET OSUUDET KAIKISTA PUMPPAAMON POIKKEAMISTA

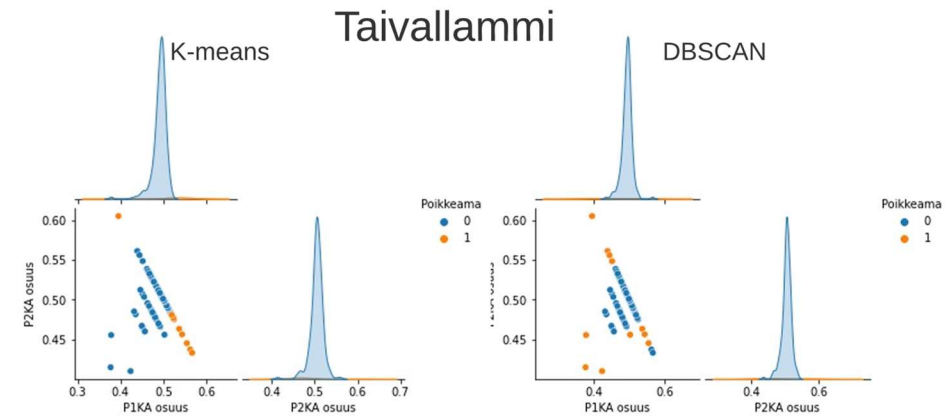
Vertailu	Roope	Pähkinäkallio	Kankkula	Lähtösaha	Taivallammi	Suntinmäki	Saarenmaa
iForest - LOF	88	81	95	78	64	75	90
iForest - Kmeans	57	44	21	67	55	38	48
iForest - DBSCAN	85	74	90	44	82	75	67
LOF - Kmeans	65	37	21	67	36	12	43
LOF - DBSCAN	78	78	87	33	45	75	62
Kmeans - DBSCAN	45	37	23	22	36	12	43
Keskiarvo	70	59	56	52	53	48	59
Poikkeamien lukumäärä	40	27	39	9	11	8	21

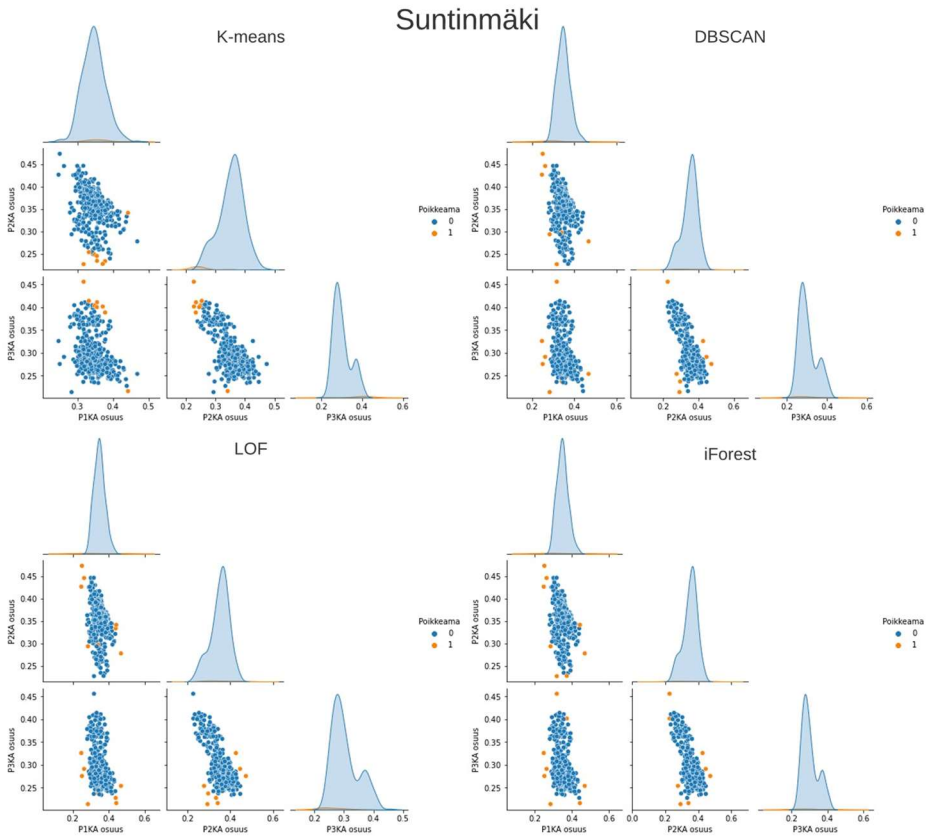
LIITE D: MALLIEN TUNNISTAMAT POIKKEAMAT KÄYNTIAIKOJEN OSUUKSIEN PISTE-KAAVIOMATRIISEISSA

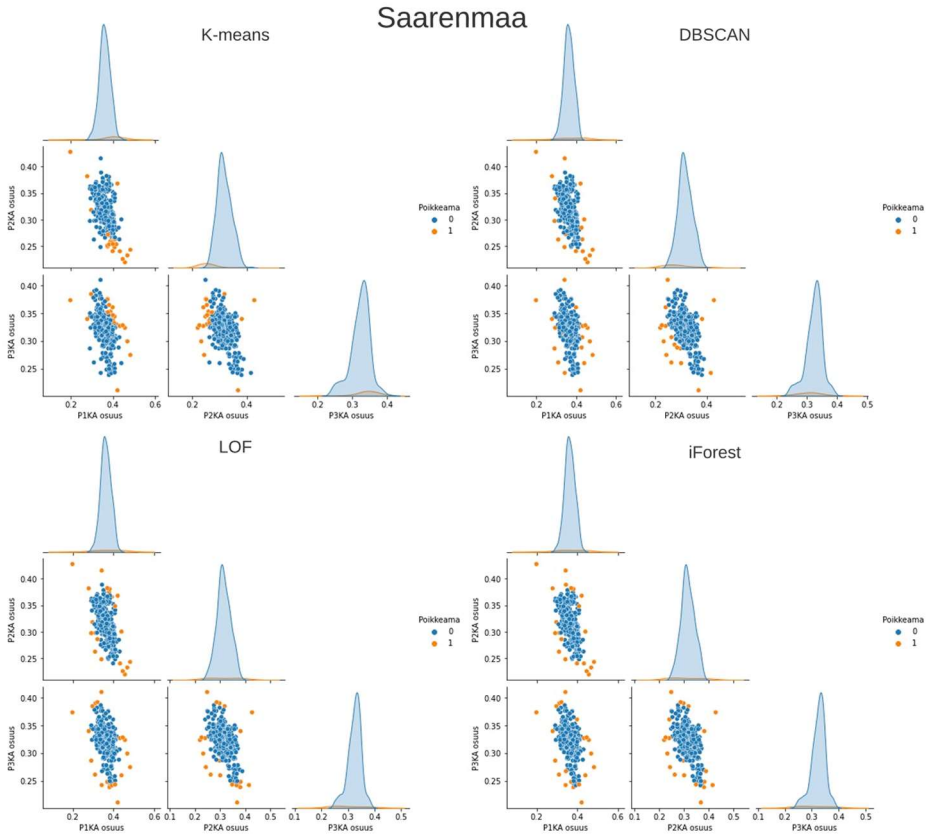



Kankkula
K-means
DBSCAN
LOF
iForest
P1KA osuus
P2KA osuus
P3KA osuus
Poikkeama
0
1


Lähtösaha
K-means
DBSCAN
LOF
iForest
P1KA osuus
P2KA osuus
Poikkeama
0
1


Taivallammi
K-means
DBSCAN
LOF
iForest
Poikkeama
0
1
P1KA osuus
P2KA osuus


Suntinmäki
K-means
DBSCAN
LOF
iForest
Poikkeama
0
1
P1KA osuus
P2KA osuus
P3KA osuus


Saarenmaa
K-means
DBSCAN
LOF
iForest
P1KA osuus
P2KA osuus
P3KA osuus
Poikkeama
0
1