

Joe Hyvärinen

**Poikkeavien energiankulutusten havaitseminen
rakennustyypeittäin**

Tietotekniikan pro gradu -tutkielma

18. elokuuta 2023

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Joe Hyvärinen

Yhteystiedot: jojohyva@student.jyu.fi

Ohjaaja: Raino Mäkinen

Työn nimi: Poikkeavien energiankulutusten havaitseminen rakennustyypeittäin

Title in English: Abnormal energy consumption report of buildings

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 53+0

Tiivistelmä: Rakennukset kuluttavat merkittävän määrän energiaa koko maailman energiankulutuksesta. Rakennusten energiankulutusten poikkeamien tunnistaminen on merkittävä haaste ja askel eteenpäin ilmastonmuutoksen ja kestävä kehityksen suhteen. Tässä tutkielmassa luotiin menetelmä havaita poikkeamia rakennusten energiankulutuksissa rakennustyypeittäin. Tutkielmassa kartoitettiin myös erilaisia teknologioita, jotka soveltuvat tällaisen kokonaisuuden kehittämiseen.

Avainsanat: Poikkeamien havainnointi, aikasarjat, koneoppiminen, tekoäly, azure, rakennusten energiankulutus

Abstract: Buildings consume a significant amount of energy from the entire world's energy consumption. Identifying anomalies in energy consumption in buildings is a significant challenge and a step forward in terms of climate change and sustainable development. In this thesis, a method was created to detect anomalies in the energy consumption of buildings by building type. The thesis also surveyed various technologies that are suitable for developing such solution.

Keywords: Anomaly detection, time series, machine learning, ai, azure, buildings energy consumption

Esipuhe

Tein tämän pro gradu -tutkielman Jyväskylän yliopiston informaatioteknologian tiedekunnassa vuonna 2023. Haluan kiittää yliopiston puolesta professori Raino Mäkistä työn ohjauksesta.

Haluan kiittää myös työnantajaani Granlund Oy:tä tutkielman tekemisen mahdollistamisesta, sekä ohjaajaani Tauno Hyväristä. Lisäksi haluan myös kiittää Kuopion Kaupunkia todellisten rakennusten energiankulutus datan mahdollistamisesta tutkielmaa varten.

Kuopiossa 18. elokuuta 2023

Joe Hyvärinen

Termiluettelo

API	Tulee sanoista Application Programming Interface. Se on joukko sääntöjä ja protokollia, joka sallii eri ohjelmistojen välisen kommunikoinnin. API määrittelee miten eri ohjelmisto komponenttien tulisi keskustella ja tarjoaa standardoidun tavan eri järjestelmille vaihtaa tietoja ja toiminnallisuuksia keskenään.
Azure	Azure on Microsoftin tarjoama pilvipalvelu alusta, joka tarjoaa laajan valikoiman työkaluja ja resursseja sovellusten ja palveluiden rakentamiseen, käyttöönottoon ja hallintaan.
Azure Functions	Azure functions on Microsoftin tarjoama palvelimeton ratkaisu, jonka avulla voi kirjoittaa vähemmän koodia, ylläpitää vähemmän infrastruktuuria ja säästää kustannuksissa.
Azure SQL	Azure SQL on Microsoftin tarjoama pilvipohjainen relaatiotietokanta, joka tarjoaa hallitun alustan strukturoidun tiedon tallentamiseen ja kyselyihin.
C#	C# on moderni, Microsoftin kehittämä olio-ohjelmointikieli.
DevOps	DevOps tulee sanoista Development ja Operations. Se on ohjelmistokehityksen metodologia, jossa yhdistyy ohjelmistokehitys (Dev) ja IT-toiminnot (Ops) tehostaakseen ohjelmistojen toimitusprosessia, mikä mahdollistaa nopeamman ja luotettavamman sovellusten kehittämisen, käyttöönoton ja ylläpidon.
JavaScript	Dynaaminen ohjelmointikieli, jota käytetään ensisijaisesti webkehitykseen. Se myös mahdollistaa interaktiivisen ja dynaamisen toiminnan verkkosivustoilla.
.NET	.NET on ilmainen avoimen lähdekoodin monialustainen ohjelmistokehitys modernien sovellusten ja tehokkaiden pilvipalvelujen rakentamiseen.
ML.NET	Microsoftin kehittämä avoimeen lähdekoodiin perustuva koneoppimisen ohjelmistokehitys, jonka avulla ohjelmistokehittäjät pystyvät rakentamaan koneoppimismalleja ja integroimaan

	niitä omiin .NET sovelluksiinsa.
Power BI	Microsoftin kehittämä liiketoiminta-analytiikka- ja datan visualisointialusta, jonka avulla käyttäjät voivat yhdistää, muuntaa ja visualisoida tietoja eri lähteistä saadakseen näkymiä ja tehdäkseen perusteltuja päätöksiä.
React	Metan kehittämä JavaScript kirjasto web käyttöliittymien kehittämiseen.
Redux	JavaScript kirjaston sovelluksen tilan hallitsemiseen.
REST	Tulee sanoista Representational state transfer. Se on ohjelmistoarkkitehtuuri tyyli tilattomille web palveluille. Nykyään yleinen tapa sovelluksille keskustella keskenään internetin yli.
TypeScript	Microsoftin kehittämä ohjelmointikieli, joka on JavaScriptin ylijoukko (superset) ja sisältää staattisen tyyppityksen. TypeScript koodi käännetään JavaScriptiksi ohjelman suoritusta varten.

Kuviot

Kuvio 1. Common energy app arkkitehtuurikuva	22
Kuvio 2. Raportti rakennusten normeeratuista ominaiskulutuskäyristä	33
Kuvio 3. Granlund Managerin uusi asetussivu	35
Kuvio 4. Kokonaisuuden ylätason arkkitehtuuri	36

Sisällys

1	JOHDANTO	1
2	TUTKIMUSONGELMA JA TUTKIMUSMENETELMÄ.....	3
2.1	Tutkimuksen motiivi	3
2.2	Tutkimuskysymykset.....	4
2.3	Tutkielman laajuus ja kohde	4
2.4	Tutkielmaan liittyvät riskit	4
2.5	Suunnittelutiede	5
2.6	Suunnittelutieteen soveltuvuus	6
3	AIKASARJAT JA -TIETOKANNAT	8
3.1	Aikasarjat	8
3.2	Aikasarjojen ominaisuudet.....	9
3.3	Aikasarjatietokannat	11
4	TEKOÄLYMALLIT	13
4.1	Tekoälymallien perusteet.....	13
4.2	Aikasarjamallit	15
4.3	Poikkeamien havainnointi mallit	16
5	RAKENNUSTEN ENERGIATEHOKKUUTEEN VAIKUTTAVAT TEKIJÄT	18
5.1	Rakennusneliöt ja -kuutiot sekä ominaiskulutus	18
5.2	Rakennusvuosi, -määräykset ja sijainti	18
5.3	Rakennustyyppit	19
6	INTEGROITAVAT SOVELLUKSET	21
6.1	Granlund Manager ja Designer	21
6.2	Common energy app	22
7	OHJELMISTON VAATIMUKSET	23
7.1	Mitä ovat ohjelmiston vaatimukset?.....	23
7.2	Alustavat vaatimukset	23
7.3	Toiminnalliset vaatimukset	24
7.4	Ei-toiminnalliset vaatimukset.....	25
7.5	Aluevaatimukset.....	25
8	TEKNOLOGIAVALINNAT	27
8.1	Aikasarjatietokanta	27
8.2	Ohjelmointikielet	27
8.3	Ohjelmistokehykset ja kirjastot	28
8.4	Raportointityökalu	29
8.5	Yhteenveto	29
9	TULOKSET.....	31

9.1	Yhteenveto	31
9.2	Tuotokset	31
9.3	Mikä onnistui?.....	37
9.4	Missä olisi parannettavaa?	38
10	POHDINTA	39
	LÄHTEET	41

1 Johdanto

Rakennukset kuluttavat merkittävän määrän energiaa, muodostaen noin 40 % maailman energiankulutuksesta ja tuottavat vastaavan määrän hiilidioksidipäästöjä (Buildings, Construction ja United Nations Environment Programme 2019). Rakennusten energiankulutuksen vähentäminen on hyvin tärkeä askel ilmastonmuutoksen hillitsemisessä ja kestävä kehityksen tavoitteiden saavuttamisessa. Rakennusten energiahäviön ja poikkeavuuksien tunnistaminen voi kuitenkin olla haastavaa rakennusjärjestelmien monimutkaisuuden ja heterogeenisyyden sekä niiden tuottaman suuren tietomäärän vuoksi. Täten tutkimuksen aiheena on kehittää tekoälymalli rakennusten energiakulutusten poikkeamien havainnointiin rakennustyypeittäin.

Rakennusten energiankulutusten poikkeamien havainnointi on lupaava tapa tunnistaa puutteita rakennusten energiatehokkuudessa. Rakennusten energiankulutusten poikkeamien havainnointi hyödyntää muun muassa koneoppimisalgoritmeja ja data-analytiikan tekniikoita havaitakseen poikkeamia rakennusten energiankulutuksista, tunnistakseen poikkeamien syyt ja ehdottaakseen tapoja korjata ne. Poikkeamien havainnoinnilla on potentiaalia auttaa rakennusten omistajia ja käyttäjiä optimoimaan rakennusten energiatehokkuutta, alentamaan energiakustannuksia sekä parantamaan rakennusten suorituskykyä ja asukkaiden mukavuutta.

Vaikka poikkeamien havainnoinnilla on monia etuja, on niiden mallien kehittämisessä ja toteutuksessa vielä monia haasteita. Yksi merkittävimmistä haasteista on tiedon laatu ja sen saatavuus, sillä rakennusten energiatiedot ovat usein puutteellisia ja epä johdonmukaisia. Toinen huomionarvoinen ongelma on, ettei rakennuksia ole tyypitetty järjestelmiin. Lopuksi vielä yksi haaste rakennusten sijainnin huomioiminen, varsinkin Suomessa, sillä kun vertaillaan ääripäiden Lapin ja Helsingin rakennusten energiankulutusta, tulee niissä olemaan merkittäviä eroavaisuuksia vuodenajasta riippuen. Näin ollen, tässä tutkimuksessa tutkitaan ja kehitetään poikkeamien havaitsemismalli, joka on käyttökelpoinen oikean maailman tilanteisiin.

Tutkimuksen toisessa kappaleessa käydään läpi tutkimusongelmaa ja -menetelmää. Kappa-

leissa 3-5 esitetään tutkimuksen teoriataustaa. Aiheita, joita käydään läpi ovat aikasarjatie-
tokannat, rakennusten energiatehokkuuteen vaikuttavat tekijät ja tekoälymallit. Kappaleessa
6 käydään läpi kehitettävän ohjelmiston vaatimukset. Kappaleessa 7 esitetään valitut tekno-
logiat kehitettävään ohjelmistoon. Kappaleessa 8 vedetään yhteen tutkimuksen tulokset ja
lopuksi kappaleessa 9 tehdään johtopäätöksiä saatujen tulosten perusteella.

2 Tutkimusongelma ja tutkimusmenetelmä

Tässä kappaleessa käydään läpi tutkimusongelmaa ja käytettyä tutkimusmenetelmää. Aluksi esitellään tutkimuksen motiivia. Seuraavaksi tarkastellaan tutkimuskysymyksiä, jota seuraa tutkimuksen laajuuden ja kohteen tarkastelu. Lisäksi tarkastellaan tutkimukseen liittyviä riskejä, ennen kuin käydään läpi tutkimusmenetelmää. Riskien tarkastelua seuraa tutkimusmenetelmän tarkastelu ja soveltuvuus tähän työhön.

2.1 Tutkimuksen motiivi

Poikkeamien havainnointi on monimutkainen ja nopeasti kehittyvä ala, täten tutkimuksen aiheeksi on valittu rakennusten energiankulutusten poikkeamien havainnointi rakennustyypeittäin. Jotta poikkeamien havainnoinnista saataisiin sen täysi potentiaali irti, tulee tälle alalle suorittaa enemmän tutkimusta ja projektityötä (Himeur ym. 2021). Lisäksi suorittamalla tutkimusta poikkeamien havainnoinnista, saadaan syvempää ymmärrystä sen teorioista, algoritmeista ja tekniikoista. Kaiken kaikkiaan sen tutkimuksesta saadaan uutta tietoa ja menetelmiä, sekä samalla ratkaistaan oikean maailman ongelmia ja jalostetaan sen kehitystä.

Havainnoimalla poikkeamia myös, pystytään tunnistamaan ei-energiatehokkaita rakennuksia. Poikkeavien havaintojen avulla kiinteistöpäälliköt pystyvät tekemään oikeita päätöksiä optimoidakseen rakennuksen energiatehokkuuden. Tämän lisäksi rakennusten omistajat pystyvät havaitsemaan syitä, mitkä johtavat korkeisiin kustannuksiin.

Tutkimukseen tuo motiivia myös yhteistyö nykyisen työnantajani Granlund Oy:n kanssa. Heidän kauttansa saadaan oikeiden rakennusten energiakulutus aikasarjatietoja, joita voidaan käyttää mallien koulutuksessa. Lisäksi Granlund on jo yli 60 vuotta vanha kiinteistö- ja rakennusalankonserni, jonka kautta saadaan sekä syvempää ymmärrystä alasta että mahdollista tukea esimerkiksi energia-asiantuntijoilta. Tutkimuksen tuloksilla saadaan energia-asiantuntijoille näyttöä asiakkaille, joilla on poikkeavan paljon energiaa kuluttavia rakennuksia. Lopuksi Granlundilla pääsen kehittämään omaa osaamistani uusien asioiden kanssa ja saan apua alan ammattilaisilta.

2.2 Tutkimuskysymykset

Yleisesti ottaen, tutkimuksen alussa on tärkeä tunnistaa ne teknologiat, joilla on parasta lähteä kehittämään rakennusten energiankulutuksen poikkeamien havainnoinnin kokonaisuutta. Teknologioita, jotka tulee ottaa huomioon, ovat aikasarjatietokannan valinta, ohjelmointikieliset, ohjelmistokehykset ja -kirjastot ja raportointityökalu.

Tämän tutkimuksen tutkimuskysymykset:

1. Miten voidaan havaita rakennusten poikkeavia energiankulutuksia rakennustyypeittäin?
2. Mitkä teknologiat ovat sopivia kokonaisuuden kehittämiseen?

2.3 Tutkielman laajuus ja kohde

Tutkielman pääkohteena on kehittää tekoälymalli, jolla pystytään havaitsemaan poikkeamia rakennusten energiankulutuksissa rakennustyypeittäin. Vaikka poikkeamien havainnoinnin tekoälymalli onkin tutkimuksen tärkein artefakti, tulee siinä samalla kehitettyä useampi artefakti. Näitä muita artefakteja ovat:

- käyttöliittymä, mistä käyttäjät voivat kouluttaa tekoälymallin heidän valitsemalleen rakennukselle,
- raportti, missä käyttäjät voi tarkastella rakennusten poikkeamia,
- tietämystä teknologioista tällaisen kokonaisuuden kehittämiseksi.

2.4 Tutkielmaan liittyvät riskit

Ennen teoriaosioita käydään vielä läpi tutkielmaan liittyviä riskejä. Ensimmäisenä riskinä on datan saatavuus ja laatu. Rajallinen tai vaillinainen data voi vaikuttaa poikkeamien havainnointien tarkkuuteen ja tehokkuuteen. Datan laadussa voi olla riskejä, sillä siellä voi olla puuttuvia, poikkeavia tai virheellisiä arvoja. Huono datan laatu voi vaikuttaa poikkeamien havainnointien luotettavuuteen.

Toisena riskinä on yksityisyys ja eettisyys. Oikeiden rakennusten energiakulutustietojen kä-

sittelyyn liittyy mahdollisesti arkaluonteisten tietojen käsittelyä. Tietosuojamääräysten ja tietojen käyttöä ja säilytystä koskevien eettisten näkökohtien noudattamisen varmistaminen on välttämätöntä yksityisyyden suojaamiseksi.

Mallin kyky sopeutua erilaisiin rakennustyyppeihin ja energia kulutustietoihin on tärkeää. Kolmantena riskinä onkin mallin ylisovittaminen. Jos malli on ylisovitettu yhdelle rakennustyyppille tai tietojoukolle, poikkeamien havainnointien tarkkuus saattaa kärsiä muissa tilanteissa, kuin mihin malli on koulutettu.

Poikkeamien havaitsemismallit toimivat usein mustina laatikoina, jolloin havaittujen poikkeavuuksien syiden tulkitseminen on haastavaa, mistä päästään neljänteen riskiin. Poikkeamien havaintojen tulkittavuuden puute voi hankaloittaa taustalla olevien syiden ymmärtämistä ja asianmukaisten toimenpiteiden toteuttamista. Tähän liittyy myös poikkeamien havainnoinnin tekninen haastavuus. Siinä pitää ottaa huomioon useita teknisiä haasteita, kuten ominaisuuksien ja algoritmin valinta, parametrien hienosäätäminen ja mallin validointi.

Edellä mainittujen riskien voittaminen voi olla hyvin aikaa vievää, mistä päästään viimeiseen riskiin, joka on aikarajoitukset. Vakaan mallin kehittäminen ja sen suoriutumisen arviointi maisterintutkielman aikajakson aikana voi olla haastavaa. Riskien lieventämiseksi on tärkeää suunnitella tutkimus- ja kehitystyö ja pyytää neuvoa ohjaajilta avun tarpeessa.

2.5 Suunnittelutiede

Suunnittelutiede on tutkimusmenetelmä, joka pyrkii kehittämään ja validoimaan teorioita luomalla uusia artefakteja tai malleja (Vaishnavi, Kuechler ja Petter 2004). Se on tieteenala, joka yhdistää tieteelliset menetelmät teknisen suunnittelun periaatteisiin luodakseen innovatiivisia ratkaisuja todellisiin ongelmiin (Peffer ym. 2007). Gregor, Jones ym. (2007) mukaan suunnittelutieteen tutkimusta on luonnehdittu myös ongelmanratkaisun näkökulmasta siten, että sillä pyritään ymmärtämään ongelman taustalla olevia syitä ja kehittämään siihen perustuvia ratkaisuja.

Vaishnavi, Kuechler ja Petter (2004) esittivät yhden viisivaiheisen mallin suunnittelutieteen tutkimusprosessista. Mallissa tunnistetaan aluksi ongelma, jonka jälkeen etsitään mahdolli-

sia ratkaisuja ongelmaan. Kolmannessa vaiheessa kehitetään tai suunnitellaan artefakti, jota sitten testataan ongelma ratkaisemiseen. Neljännessä vaiheessa arvioidaan tuotettua artefaktia ja ratkaisukykyä ongelmaan. Arvioinnin jälkeen prosessia toistetaan, kunnes ollaan tyytyväisiä lopputulokseen ja saadaan jokin tulos. Suunnittelutieteen metodologiaan siis kuuluu iteratiivisia suunnittelun, toteutuksen ja arvioinnin syklejä, joiden tavoitteena on luoda vankka ja tehokas ratkaisu tiettyyn ongelmaan. Tuloksena oleva artefakti arvioidaan sen tehokkuuden perusteella ongelman ratkaisemisessa ja tutkimusprosessia jalostetaan tulosten perusteella.

Suunnittelutiedettä on käytetty monella eri alalla, kuten ohjelmistokehityksessä, terveydenhuollossa, koulutuksessa ja liiketaloudessa. Informaatioteknologian alalla tätä tutkimusmenetelmää on käytetty erityisen paljon erinäisten artefaktien kehittämiseen, joita ovat esimerkiksi tieto- ja päätöksentekijärjestelmät (Peffers ym. 2007). Tämä tutkimusmenetelmä onkin erityisen käyttökelpoinen monimutkaisten ongelmien ratkaisemiseen, joita on hankalampi ratkoa perinteisillä tutkimusmenetelmillä. Täten suunnittelutiede valittiin tämän tutkielman tutkimusmenetelmäksi ratkaisemaan jälleen yksi kompleksi ongelma.

2.6 Suunnittelutieteen soveltuvuus

Edellisessä kappaleessa esitetty viisivaiheinen malli soveltuu hyvin rakennusten energiankulutusten poikkeamien havainnointiin liittyvien ongelmien ratkaisuun luomalla ja arvioimalla uusia artefakteja ja malleja. Suunnittelutieteen käyttäminen poikkeamien havainnoinnissa tarkoittaa sitä, että luodaan uusi algoritmi tai järjestelmä, jolla saadaan ratkottua käsillä oleva ongelma. Luomuksena tulleen artefaktin tehokkuutta ongelman ratkaisemiseen sitten arvioidaan, kuinka hyvin se pystyy havaitsemaan poikkeamia ja tulkitsemaa tuloksia.

Tarkemmin tarkastelemalla Vaishnavi, Kuechler ja Petter (2004) viisivaiheisen mallin hyödyntämistä tämän tutkimuksen kehitykseen, saadaan sen soveltuvuudesta paremmin kiinni. Ensimmäisessä vaiheessa tulee tunnistaa ongelma, joka poikkeamien havainnoinnin tapauksessa on huono tulosten tulkitseminen tai liian iso väärin positiivisten suhde (false positive rate). Tämän jälkeen, kun ongelma on tunnistettu, luodaan artefakti ratkaisemaan tämä ongelma. Tässä tapauksessa luodaan joko oma algoritmi tai käytetään tekoälymalleja. Ar-

tefaktin luonnin jälkeen arvioidaan sen tehokkuutta tunnistetun ongelman ratkaisemiseen. Tässä voidaan käyttää oikeiden rakennusten energiankulutuksen aikasarjatietoja. Yhden tai useamman iteraation perusteella, kun ollaan tyytyväisiä tulokseen, on tutkimusprosessi viety maaliin.

3 Aikasarjat ja -tietokannat

Tässä kappaleessa käydään läpi, mitä ovat aikasarjat, niiden ominaisuudet ja aikasarjatietokannat. Aluksi tarkastellaan aikasarjan määritelmää ja niiden käyttöä, jota seuraa niiden ominaisuuksien tarkastelu. Tämän jälkeen käsitellään aikasarjatietokannan määritelmää, ominaisuuksia ja hyötyjä sekä käyttöä.

3.1 Aikasarjat

Aikasarjat ovat jonkin muuttujan, kuten lämpötilan, arvojen järjestetty sarja tasaisin aikavälein (Naqvi, Yfantidou ja Zimányi 2017). Myös IBM (2021) määrittelyn mukaan, aikasarja on järjestetty kokoelma arvoista, jotka on mitattu säännöllisin väliajoin. Esimerkiksi aikaleimatut tiedot, kuten joidenkin antureiden mittauksia voidaan pitää aikasarjoina. Mittaukset, jotka muodostavat aikasarjan, ovat järjestetty aikajanelle, joka paljastaa tietoa taustalla olevista malleista. Joshi ym. (2016) mukaan aikasarjojen järjestämisellä on merkitystä, koska se sisältää riippuvuuden ajan ja mittauksen välillä ja järjestyksen muuttaminen voi muuttaa tiedon merkityksen. Esimerkki aikasarjoista voisi olla sääaseman lämpötilan mittaukset tunnin välein tai pörssisähkön spot-hintojen ennuste.

IBM (2021) mukaan jokainen aikasarja on yleensä luokiteltavissa yhteen luokkaan. Näitä luokkia on riippuvaisuus (dependent), ennuste (predictor), tapahtuma (event) ja interventio (intervention). Riippuvaisuus tarkoittaa aikasarjaa, jota halutaan ennustaa, kun taas ennusteella tarkoitetaan aikasarjaa, joka voi auttaa selvittämään tutkittavaa kohdetta. Tapahtumat ja interventiot ovat omalaatuisia aikasarjoja. Tapahtumalla tarkoitetaan ennusteaikasarjaa, jolla ennustetaan toistuvia tapahtumia. Toisaalta intervention avulla otetaan huomioon kertaluontaisesti tapahtuneita menneitä tapahtumia, kuten sähkökatkoja.

Aikasarjoille on useita eri käyttötapauksia, joista yleisimmät kuitenkin ovat aikasarja-analyysi ja -ennustaminen, sekä regressioanalyysi. Hayes (2022) mukaan aikasarja-analyysillä tutkitaan, kuinka jokin tietyn muuttujan arvo vaihtelee ajan myötä. Aikasarjaennuste taas käyttää historiallisia arvoja ja niihin liittyviä malleja tulevaisuuden ennustamiseksi. Regressioanalyysillä voidaan tarkastella, miten johonkin tiettyyn muuttuajaan liittyvät muutokset mahdol-

lisesti aiheuttavat muutoksia muissa muuttujissa samalla ajanjaksolla.

3.2 Aikasarjojen ominaisuudet

Tutkimalla aikasarjoja, kuten piirtämällä käyrän niiden arvoista, voidaan huomata niissä yksi tai useampi ominaisuus. Yleisiä ominaisuuksia ovat trendit, kausittaiset ja ei kausittaiset syklit, sykäykset ja askeleet ja poikkeavat arvot (IBM 2021). Myös Wang, Smith-Miles ja Hyndman (2009) mukaan, trendit ja kausittaisuus ovat aikasarjojen yleisiä ominaisuuksia. He myös listasivat muita paljon muualla käytettyjä ominaisuuksia, kuten jaksollisuus, auto-korrelaatio, vinous (skewness) ja huipukkuus (kurtosis). Aikasarjojen tutkimisella voidaan havaita kuvioita ja tehdä parempia ennustuksia.

Wang, Smith-Miles ja Hyndman (2009) mukaan trendi on olemassa silloin, kun aikasarjan pitkällä aikavälillä tapahtuu muutos sen arvojen keskiarvossa. Toisaalta IBM (2021) mukaan trendit ovat aikasarjan arvojen joko nousevaa tai laskevaa sarjaa ajan myötä. Trendit voivat olla myös joko paikallisia tai globaaleja, mutta yhdestä sarjasta voi löytyä molempia. Tämän lisäksi ne voivat olla myös lineaarisia ja epälineaarisia.

Aikasarjoissa esiintyvä kausittainen sykli on taas ennustettavissa oleva kuvio aikasarjan arvoista (IBM 2021). Wang, Smith-Miles ja Hyndman (2009) taas totesivat, että aikasarjan kausittaiset kuviot syntyvät, kun se toistaa itseään kiinteän aikavälin välein. Toisaalta, ei-kausittainen sykli on toistuva, mahdollisesti arvaamaton kuvio aikasarjan arvoissa. Kausittaiset syklit on sidottu aikasarjojen aikaväleihin. Esimerkiksi rakennuksen energiakulutuksen aikasarja on yleensä kesäisin syklin alhaalla ja talvisin syklin huipulla. Ei-kausittaisissa sykleissä toistuu usein jokin tapahtuma, kuten työttömyys. Sitä on kuitenkin vaikea ennustaa milloin se on alhaisimmillaan ja milloin korkeimmillaan, koska se ei toistu säännöllisesti tai kausittain.

IBM (2021) mukaan monissa aikasarjoissa esiintyy äkillisiä tasomuutoksia, joita on yleensä kahta tyyppiä, sykäyksiä ja askelia. Sykäykset ovat äkkinäisiä väliaikaisia muutoksia aikasarjassa, kun taas askeleet ovat äkkinäisiä pysyviä muutoksia aikasarjassa. Kun havaitaan pulssi tai askel aikasarjasta, sille on tärkeää myös löytää hyvä selitys, koska aikasarjamalleja ei ole suunniteltu ottamaan huomioon äkillisiä muutoksia. Mikäli pulssi tai askel pystytään

selittämään, se voidaan mallintaa interventiolla tai tapahtumalla (IBM 2021).

Aikasarjoista löytyvät muutokset, joille ei löydy selitystä, ovat poikkeamia. Poikkeamat ovat ristiriidassa lopun aikasarjan kanssa ja voivat vaikuttaa analyysiin ja aikasarjamallin ennusteeseen (IBM 2021). Yleisiä poikkeamia, joita löytyy aikasarjoista, ovat additiiviset (additive), innovatiiviset (innovational), tason siirto (Level shift), lyhytaikaiset muutos (transient change), kausittaiset additiiviset ja lokaalit trendipoikkeamat. Additiiviset poikkeamat ilmenevät yhtenä suurena tai pienenä havaintoina, joka ei vaikuta myöhempisiin havaintoihin. Innovatiivisella poikkeamalla esiintyy yleensä alustava vaikutus, jonka vaikutukset kestävät vielä havaitessaan myöhempiä havaintoja. Tason siirrossa kaikki arvot poikkeaman jälkeen siirtyvät uudelle tasolle. Tällä poikkeaman tyypillä on pysyvä vaikutus ja se vaikuttaa useaan havaintoon. Lyhytaikaiset muutokset ovat samankaltaisia kuin tason siirron poikkeamat, mutta niiden vaikutus pienenee tulevien havaintojen myötä, kunnes lopulta aikasarja palaa normaalille tasolleen. Kausittaiset additiiviset poikkeamat taas esiintyvät yllättävän suurena tai pienenä arvona säännöllisin väliajoin. Lokaaleja trendipoikkeamia syntyy, kun alkuperäisen poikkeaman jälkeen aikasarja lähtee ajelehtimaan (drift).

Aikasarjan jaksollisuudella tarkoitetaan sitä, että se on mahdollista jakaa samanpituisiin osiin, jotka ovat melkein samanlaisia (Elfeky, Aref ja Elmagarmid 2005). Wang, Smith-Miles ja Hyndman (2009) mukaan jaksollisuus on erittäin tärkeä ominaisuus, jota käytetään aikasarjan kausien määrittelyyn ja sen syklien tutkimiseen.

Autokorrelaatiolla kuvaillaan jonkin aikasarjan muuttujan nykyisen ja menneen arvon suhdetta tietyllä aikavälillä (IBM 2021). Jos muuttujan korrelaatio on nolla, korrelaatiota ei ole ja jokainen havainto on riippumaton toisistaan. Kun taas, jos muuttujan korrelaatio kääntyy toista kohti, menneet arvot vaikuttavat tuleviin havaintoihin.

Wang, Smith-Miles ja Hyndman (2009) mukaan vinoudella mitataan aikasarjan symmetriää. Aikasarja on symmetrinen, jos se näyttää samalta keskipisteen vasemmalla ja oikealla puolella. Vinouman mittaa käytetään aikasarjan keskiarvon ympärillä olevien arvojen epäsymmetrisyyden kuvaamiseen (Wang, Smith-Miles ja Hyndman 2009).

Huipukkuudella mitataan, ovatko arvot huippuja vai tasaisia normaalijakauman suhteen (Wang, Smith-Miles ja Hyndman 2009). Jos tietojoukolla on korkea huippu, se on yleensä selkeä

huippu keskiarvon lähellä ja laskee melko nopeasti. Toisaalta, jos tietojoukolla on alhainen huipukkuus, niillä on yleensä myös tasaisempi huippu keskiarvon lähellä.

3.3 Aikasarjatietokannat

Nykyään enemmistö yrityksistä tuottaa suuren määrän mittauslukemia ja -tapahtumia, mikä on tehnyt aikasarjatietokantojen tarpeesta välttämättömän. Naqvi, Yfantidou ja Zimányi (2017) mukaan aikasarjatietokannat ovat tietokantatyyppejä, jotka ovat optimoitu aikasarjoille. Ne on rakennettu erityisesti mittareiden tapahtumien tai aikaleimalla merkittyjen mittauksien käsittelyä varten. Aikasarjatietokannat on optimoitu mittaamaan muutosta ajan myötä. Ne antavat käyttäjille mahdollisuuden luoda, listata, päivittää, tuhota ja järjestää erilaisia aikasarjoja tehokkaammin.

Tärkeimmät ominaisuudet, jotka erottavat aikasarjatiedot tavallisesta tiedosta, ovat yhteen vetäminen, tiedon elinkaaren hallinta ja useiden tietueiden laajat tarkastelut (large range scan). Naqvi, Yfantidou ja Zimányi (2017) määrittelevät aikasarjatietokantojen vaadituiksi ominaisuuksiksi tiedon sijainnin, nopeat ja helpot aluekyselyt (range queries), tehokkaat kirjoitusoperaatiot, tiedonpakkauksen, skaalattavuuden ja käytettävyyden. Tiedon sijainti aikasarjatietokannoissa on tärkeä, koska jos tiedot eivät sijaitse missään fyysisessä tallennustilassa, tietokantakyselyt voivat olla erittäin hitaita ja johtaa aikakatkaisuihin. Aikasarjatietokannat säilyttävät toisiinsa liittyvät tiedot yhdessä, minkä avulla aluekyselyt ovat nopeita. Kirjoitusoperaatioidenkin tulee olla tehokkaita, koska aikasarjoja tallennetaan miltei joka sekunti. Koska aikasarjaa kerätään melkein joka sekunti, tulee aikasarjatietokannan käyttää normaalia parempaa tiedonpakkauksen menetelmää. Tiedon määrän jatkuvan kasvamisen takia, tulee niiden myös olla helposti skaalattavissa. Käytettävyyden kannalta aikasarjatietokantojen olisi hyvä tarjota operaatioita ja metodeja aikasarja-analyysiä varten.

Aikasarjatietokantojen hyötyjen myötä ei tarvitse enää aina tukeutua tavanomaiseen tietokantaan, esimerkiksi Sql Serveriin. Nykyään yleisiä aikasarjatietokannan sovelluksia ovat IoT (Internet of things) ja data-analytiikka. Yksi aikasarjatietokantojen hyödyistä on niiden valtava skaalautuvuus ja suorituskyky (Riak 2023). Riak (2023) mukaan niillä on myös lyhyemmät käyttökatkokset ja häiriöajat. Lisäksi aikasarjatietokannoilla on alhaisemmat ku-

lut, kuin tavallisella relaatiotietokannalla. Niiden ansiosta organisaatiot pystyvät tekemään nopeampia ja tarkempia päätöksiä, koska aikasarjatiетokannat mahdollistavat tietojen reaaliaikaisen seuraamisen ja analysoinnin.

Naqvi, Yfantidou ja Zimányi (2017) esittivät myös joitain yleisiä käyttötapauksia aikasarjatiетokannoille, kuten ohjelmistojen ja fyysisten järjestelmien monitoroinnin. Lisäksi voidaan seurata esimerkiksi ympäristöä, laitteita tai ihmiskehoa. Toinen käyttötapaus heidän mukaansa on kaupanteko järjestelmissä. Viimeisimpänä he mainitsivat, että niillä voidaan seurata, mitä käyttäjät tai asiakkaat tekevät. Niitä voidaan myös käyttää liiketoimintatiedon työkaluina seuraamalla avain mittareita ja organisaation kuntoa.

4 Tekoälymallit

Tässä kappaleessa käydään läpi tekoälymalleja. Ensimmäisessä alaluvussa käydään yleisesti mitä ovat tekoälymallit. Toisessa alaluvussa käydään läpi mitä ovat aikasarjamallit ja millaisia erilaisia aikasarjamalleja on. Kolmannessa alaluvussa taas käsitellään poikkeamien havainnoinnin malleja.

4.1 Tekoälymallien perusteet

Tekoälymallit ovat laskennallisia järjestelmiä, joiden tarkoituksena on jäljitellä tai simuloida ihmisen älykkyyttä. Nämä mallit hyödyntävät kehittyneitä algoritmeja ja tekniikoita suurten tietomäärien käsittelyyn ja tulkitsemiseen, mikä mahdollistaa sekä monimutkaisten tehtävien suorittamisen että älykkäiden päätösten tekemisen. Yleisesti tekoälymallit voidaan jakaa erinäisiin tyypeihin, joita ovat ohjattu, ohjaamaton, osittain ohjattu, vahvistus-, siirto- ja ensemble-oppiminen.

Ohjattu oppiminen sisältää syötemuuttujien ja tulostemuuttujan välisen kartoituksen oppimista ja tämän kartoituksen hyödyntämistä tulosten ennustamiseen (Cunningham, Cord ja Delany 2008). Myös Hastie ym. (2009) mukaan ohjatussa oppimisessä tavoitteena on käyttää syötemuuttujia tulostemuuttujien arvojen ennustamiseen. Bishop ja Nasrabadi (2006) kuvailivat, että ohjatussa oppimisessä tekoälymallit koulutetaan käyttämällä merkittyjä tietojoukkoja, missä syöte- ja tulostemuuttujat on annettu. Tekoälymalli oppii datassa olevia kuvioita ja suhteita tehdäkseen ennustuksia tai luokituksia, kun sille annetaan uutta, aikaisemmin näkemätöntä dataa. Cunningham, Cord ja Delany (2008) vielä totesivat, että ohjattu oppiminen on koneoppimisen tärkein menetelmä, jolla on myös merkittävä rooli multimediatatan käsittelyssä.

Bishop ja Nasrabadi (2006) mukaan ohjaamattomassa oppimisessä koulutetaan tekoälymalleja ei-merkatuilla tietojoukoilla, missä mallit pyrkivät etsimään kuvioita, rakenteita ja suhteita datasta ilman ohjausta. Hastie ym. (2009) kuvailivat ohjaamatonta oppimista kielikuvan kautta, missä tekoälymalli on oppilas ja sitä kouluttava henkilö on opettaja. Heidän mukaansa ohjaamattomassa oppimisessä oppilas oppii ilman opettajan apua. Siinä missä ohjatussa

oppimisessa opettaja kertoo, onko vastaus oikein tai väärin, ohjaamattomassa oppimisessa vastaukseen ei reagoida. Myös Ghahramani (2003) määritteli ohjaamattoman oppimisen siten, että tekoälymalli saa syötemuuttujia, mutta ei yhtään tulostemuuttujia. Sekä Bishop ja Nasrabadi (2006) että Ghahramani (2003) mukaan yleisiä esimerkkejä ohjaamattoman oppimisen käytöstä ovat klusterointi ja dimensiovähennys. Lisäksi Bishop ja Nasrabadi (2006) mukaan poikkeamien havainnointi on yksi esimerkki ohjaamattoman oppimisen tehtävistä.

Osittain ohjattu oppiminen taas sijoittuu käsitteellisesti ohjatun ja ohjaamattoman oppimisen väliin (Van Engelen ja Hoos 2020). Zhu (2005) mukaan osittain ohjattu oppiminen käyttää suuria määriä ei merkattua dataa merkatun datan kanssa rakentaakseen parempia luokittelijoita. Myös Van Engelen ja Hoos (2020) totesivat, että osittain ohjattu oppiminen hyödyntää saatavilla olevaa suurta määrää merkkamatonta dataa monissa käyttötapauksissa yhdessä pienemmän merkatun tietojoukon kanssa. Zhu (2005) vielä totesi, että ohjaamaton oppiminen on erittäin kiinnostava aihe sekä teoriassa että käytännössä, koska se vaatii vähemmän vaivaa ihmiseltä ja tuottaa parempia tarkkuuksia.

Vahvistusoppimisessa tekoälymallit oppivat olemalla vuorovaikutuksessa ympäristönsä kanssa (Sutton ja Barto 2018). Myös Ghahramani (2003) totesi, että vahvistusoppimisessa malli on vuorovaikutuksessa ympäristönsä kanssa tekemällä toimintoja, jotka vaikuttavat ympäristön tilaan. Täten tekoälymalli saa palautetta palkinnon tai rangaistuksen muodossa. Molemmat Sutton ja Barto (2018) ja Ghahramani (2003) totesivat, että tekoälymallin tavoitteena on sopeuttaa omia tekojaan ja käytöstään palautteen perusteella siten, että se pystyy maksimoimaan tulevat palkinnot. Sutton ja Barto (2018) mukaan tämä lähestymistapa sopii tehtäviin, joissa pitää tehdä päätöksiä ja peräkkäisiä toimenpiteitä, kuten pelien pelaaminen tai autonominen navigointi.

Torrey ja Shavlik (2010) mukaan siirto-oppimisessa pyritään nimensä mukaisesti siirtämään opittua tietoa yhdestä tai useammasta lähteestä ja käyttää sitä tämän hetkisen tehtävän koulutuksen kehittämiseen. Muun muassa Weiss, Khoshgoftaar ja Wang (2016) kuvailivat siirto-oppimisen käyttöä siten, että sitä käytetään jonkin alueen koulutuksen parantamiseen siirtämällä tietoa toiselta samaan asiaan liittyvältä alueelta. Siirto-oppimisen tavoitteena on parantaa oppimista tämänhetkisen tehtävän kanssa hyödyntämällä tietoa jostain lähteestä (Torrey ja Shavlik 2010). Tästä voidaan käyttää havainnollistavaan esimerkkiä, missä on kaksi hen-

kilöä, joista toinen on pelannut elämänsä jääkiekkoa ja toinen ei ole pelannut ikinä mitään mailapeliä. Molemmat haluavat oppia jonkin uuden mailapelin, kuten salibandyn. Henkilö, jolla on kokemusta jääkiekosta, oppii salibandyn pelaamisen nopeammin, sillä hän voi siirtää aiemmin opittuja mailankäsittelytaitojaan tähän uuteen tehtävään, eli salibandyn oppimiseen.

Ensemble oppiminen hyödyntää useita tekoälymalleja tehdäkseen ennusteita tai päätöksiä (Hastie ym. 2009). Yhdistelemällä yksittäisten mallien tulosteita ensemble mallien tarkkuus, roustaisuus ja yleistävyys paranee. Hastie ym. (2009) mukaan yleisiä ensemble oppimisessa käytettyjä tekniikoita ovat bagging, boosting ja stacking.

4.2 Aikasarjamallit

Aikasarjamalleja käytetään jonkin aikaan sidotun muuttujan arvojen ennustamiseen menneiden havaintojen perusteella (Adhikari ja Agrawal 2013). Tällainen muuttuja voisi olla esimerkiksi lämpötila tai pörssisähkön spot-hinta. Malleilla voidaan analysoida ja ennustaa trendejä, kausivaihtelua ja tunnistaa muita kuvioita tiedosta. Malleja käytetään yleensä taloustieteen, rahoituksen ja meteorologian aloilla.

Yksi yleisimmästä ja eniten käytetyistä stokastisista aikasarjamalleista on autoregressiivinen integroitu liukuva keskiarvomalli (ARIMA) (Box ja Jenkins 1970). Adhikari ja Agrawal (2013) mukaan ARIMA-mallilla on alaluokkia muista malleista, kuten autoregressiivinen (AR), liukuva keskiarvo (MA) ja autoregressiivinen liukuva keskiarvomalli (ARMA). Box ja Jenkins (1970) esittivät variaation ARIMA-mallista, jolla pystyy tekemän kausittaista aikasarja ennustamista (SARIMA).

ARIMA-mallien lisäksi keinotekoisien neuroverkon mallit (ANN) ovat kiinnittäneet huomiota aikasarjojen ennustamisessa (Adhikari ja Agrawal 2013). Muun muassa Zhang (2003) loivat hybridimallin, jossa he hyödynsivät ARIMA- ja ANN-mallin uniikkeja vahvuuksia lineaariseen ja epälineaariseen mallintamiseen. Yleisin variaatio ANN on monitasoiset perseptronit (MLP) (Zhang, Patuwo ja Hu 1998). Myös ANN-malleista on luotu kausittaisen aikasarjan ennustamisen malli (SANN) (Hamzaçebi 2008).

Cao ja Tay (2003) mukaan tukivektorikone (SVM) mallit ovat herättäneet huomiota usealla alalla, mukaan lukien aikasarjojen ennustamisessa. Myös SVM-malleista on useita variaatioita, joista suosituimmat ovat pienimmän neliösumman SVM (LS-SVM) ja dynaaminen pienimmän neliösumman SVM (DLS-SVM) (Adhikari ja Agrawal 2013). Esimerkiksi Suykens ja Vandewalle (2000) esittivät artikkelissaan LS-SVM-mallin luokittelu ja staattisten epälineaaristen funktioiden arviointiongelmien. Toisaalta Fan, Li ja Song (2006) käyttivät DLS-SVM:ää 4-karboksibenstraalidikynin pitoisuuden ennustamiseen puhdistetun tereftaalihapon hapetusprosessissa.

4.3 Poikkeamien havainnointi mallit

Poikkeamien havainnointi on yleinen ongelma usealla eri alalla. Siinä tarkoitus on havaita epätavallisia tapahtumia tai kuvioita tiedosta, jotka voisivat viitata väärään, virheelliseen tai muutoin epätavallisiin tapahtumiin (Chandola, Banerjee ja Kumar 2009). Hodge ja Austin (2004) mukaan poikkeamien havainnoinnin mallit on suunniteltu oppimaan järjestelmän tai prosessin normaalia käyttäytymistä ja sitten havaitsemaan harhauksia siitä käytöksestä, jotka voivat viitata poikkeamiin.

Chandola, Banerjee ja Kumar (2009) mukaan poikkeamien havaitsemismalleja on useita, mukaan lukien tilastolliset mallit, koneoppimismallit ja hybridimallit, joissa yhdistyvät sekä tilastolliset että koneoppimistekniikat. Tilastollisia malleja käytetään usein poikkeamien havainnointiin aikasarjasta, missä tarkoituksena on tunnistaa trendejä tai kuvioita, jotka ovat odotetun joukon ulkopuolella (Hawkins 1980). Toisaalta Chandola, Banerjee ja Kumar (2009) mukaan koneoppimismallit ovat taas suunniteltu tunnistamaan poikkeamia monimutkaisista tietojoukoista, joilla on moniulotteisia piirreavaruuksia (feature spaces).

Myös hybridimalleista on tulossa yhä suosituimpia poikkeamien havaitsemisessa, sillä ne pystyvät hyödyntämään sekä tilastollisten että koneoppimistekniikoiden vahvuuksia (Chandola, Banerjee ja Kumar 2009). Hodge ja Austin (2004) mukaan nämä mallit voivat käyttää tilastollisia tekniikoita, kuten aikasarja-analyysia, tunnistamaan normaalia käyttäytymistä ja sitten käyttää koneoppimisalgoritmeja, kuten SVM tai keinotekoisia neuroverkkoja, tunnistamaan poikkeamia.

Poikkeamien havaitsemismalleja käytetään useilla eri aloilla, mukaan lukien petosten tunnistamisessa rahansiirroissa, tietokoneiden tietoverkkojen tunkeutumisten tunnistuksessa ja vikojen havaitsemisessa teollisissa prosesseissa (Chandola, Banerjee ja Kumar 2009). Poikkeamien havainnoinnin merkitys johtuu siitä, että datassa olevat poikkeavuudet viittaavat merkittäviin ja usein kriittisiin tietoihin, joiden takia tulisi tehdä toimia (Chandola, Banerjee ja Kumar 2009).

5 Rakennusten energiatehokkuuteen vaikuttavat tekijät

Tässä kappaleessa käydään läpi rakennusten energiankulutukseen vaikuttavia tekijöitä. Aluksi katsotaan rakennusneliöiden ja -kuutioiden sekä ominaiskulutuksen merkitystä, jota seuraa rakennusvuoden, -määräysten ja sijainnin tarkastelu. Lopuksi katsotaan vielä rakennustyyppisiä ja niiden merkitystä.

5.1 Rakennusneliöt ja -kuutiot sekä ominaiskulutus

Rakennusten energiakulutuksessa on tärkeä huomioida joissain tapauksissa tilavuus ja pinta-ala. Esimerkiksi loft-asuntojen tilat voivat olla moninkertaisesti korkeampia, kuin *Ympäristöministeriön asuin-, majoitus- ja työtiloista* (2017) minimimääräykset, jolloin rakennusten tilavuus aiheuttaisi poikkeamia neliöperusteiseen vertailuun. Myös Kärkimaa (2015) totesi, että kiinteistön energiankulutukseen vaikuttaa sen tilavuus.

Rakennusneliöissä on myös tärkeää ottaa huomioon se, että käytetään aina samaa neliötä, sillä on olemassa kaksi eri tapaa esittää rakennuksen koko neliöinä. Ensimmäinen on kerrosala, joka sisältää myös kaikki kylmät tilat. Toinen on huoneistoala, joka taas sisältää vain sisäpuolella olevan lämpimän tilan.

Seppänen (2001) mukaan ominaiskulutuksella tarkoitetaan rakennuksen energiankäyttöä joltain suoriteyksikköä kohden. Yleensä kokonaiskulutus jaetaan rakennuksen tilavuudella tai pinta-alalla, mutta muitakin suoriteyksikköjä, kuten lämpötilaa, voidaan käyttää. Kärkimaa (2015) totesi, että ominaiskulutuksen tarkoitus on esittää energiatehokkuutta. Rakennusten energiakulutuksesta on hyvä vielä mainita kulutustietojen normeeraus. Se tarkoittaa kulutuksen suhteuttamista vertailuajanjakson ja tarkasteltavan vuoden ulkolämpötiloihin (Kärkimaa 2015).

5.2 Rakennusvuosi, -määräykset ja sijainti

Koivula (2016) mukaan rakennusmääräysten tavoitteena on rakennusten energiatehokkuuksien parantaminen sekä uusiutuvien energiamuotojen käytön edistäminen. Hänen mukaan ra-

kennusmääräykset kiristyvät jatkuvasti sen takia, että rakentamisessa pyritään kohti nollae-nergiaarakentamista. Suomen voimassa olevien *Rakentamismääräykset (2023)* myötä, uudet rakennukset ovat merkittävästi energiatehokkaampia, kuin vanhat rakennukset. Täten rakennusten valmistusvuodella on myös merkitystä, kun tarkastellaan sen energiakulutusta. On myös hyvä huomioida, että vanhempien rakennusten energiatehokkuutta on mahdollisesti parannettu niiden rakennus- tai toimenpideluvanvaraisen korjaus- ja muutostyön tai rakennuksen käyttötarkoituksen muutoksen yhteydessä, mikäli se on ollut teknisesti, toiminnallisesti ja taloudellisesti toteutettavissa.

Rakennusten energiankulutukseen vaikuttaa useampi tekijä. Suomessa varsinkin sijainnil- la on merkitystä, sillä Pohjois-Suomessa on talvella merkittävästi kylmempi kuin Etelä-Suomessa. Lumen määrässäkin on huomattavia eroavaisuuksia sijainnin mukaan ympäri Suomea. Koivula (2016) mukaan sijaintiin vaikuttaa myös tekijät, kuten onko rakennus au- ringossa vai varjossa, onko rakennusmaaperä kosteaa vai kuivaa ja millaiset tuuliolosuhteet kohdistuvat rakennukseen. Myös Kärkimaa (2015) mainitsi, että kiinteistöjen energiankulu- tukseen vaikuttaa sekä ikkunoiden suuntaus ja suuruus että rakennuksen suuntaus tontilla. Ikkunoiden suuntauksella ja koolla on merkitystä, koska suuntaamalla isot ikkunat on etelää kohti, saadaan talvella enemmän lämpöä auringosta.

5.3 Rakennustyyppit

Tilastokeskus (2018) toteaa, että rakennusluokituksen mukaan rakennukset voidaan jakaa viiteentoista eri pääluokkaan. Pääluokat on merkitty kahdella numerolla ja ne on jaettu kol- minumerotason luokkiin ja vielä nelinumerotason luokkiin. Käsiteltävät rakennukset luoki- tellaan aina alimman tason, eli nelinumerotason mukaisesti.

Poikkeamien havainnoinnissa on myös tärkeä ottaa huomioon käsiteltävän rakennuksen ra- kennustyyppi, sillä esimerkiksi uimahallit kuluttavat merkittävästi enemmän energiaa, kuin paritalot. Rakennustyyppien perusteella saadaan selville rakennusten käyttötarkoitus, joka antaa osviittaa niiden tyyppillisestä energiakulutuksesta. Vesala (2022) myös mainitsi, että ra- kennusten energiakulutukseen vaikuttaa rakennustyyppi, asukkaiden ja laitteiden lukumäärä sekä näiden laitteiden kulutus- ja käyttötottumukset. Tämän lisäksi, Kärkimaa (2015) mu-

kaan kiinteistön energiankulutukseen vaikuttaa kiinteistön käyttötarkoitus (rakennustyyppi), tilojen käyttöaste sekä kiinteistön käyttäjätottumukset.

6 Integroitavat sovellukset

Tässä kappaleessa esitetään kaksi sovellusta, joihin tutkimustyössä kehitettävän sovelluksen tulee integroitua. Ensimmäiseksi käydään läpi Granlund Manager -ohjelmistoa, jota seuraa mikropalvelun Common energy app:n tarkastelu.

6.1 Granlund Manager ja Designer

Granlund Manager on kiinteistöjen ylläpitoprosessien johtamiseen ja tiedonhallintaan tarkoitettu ohjelmisto (*Granlund* 2021). Sen avulla pystyy tehostamaan kiinteistöjohtamista, jolloin säästää aikaa, energiaa, rahaa ja pystyy samalla panostamaan enemmän ihmisten ja kiinteistöjen hyvinvointiin. Granlund Manageria käytetään useilla eri aloilla, kuten kaupan alalla, hotelleissa, ravintoloissa, suojelluissa rakennuksissa ja kulttuurikohteissa, oppilaitoksissa, sairaaloissa, toimitiloissa ja teollisuudessa (*Granlund* 2021). Tämän lisäksi sen käyttäjiin kuuluu kaupunkeja ja kuntia sekä erikoiskohteita kuten konesaleja ja urheilu- sekä liikuntatiloja.

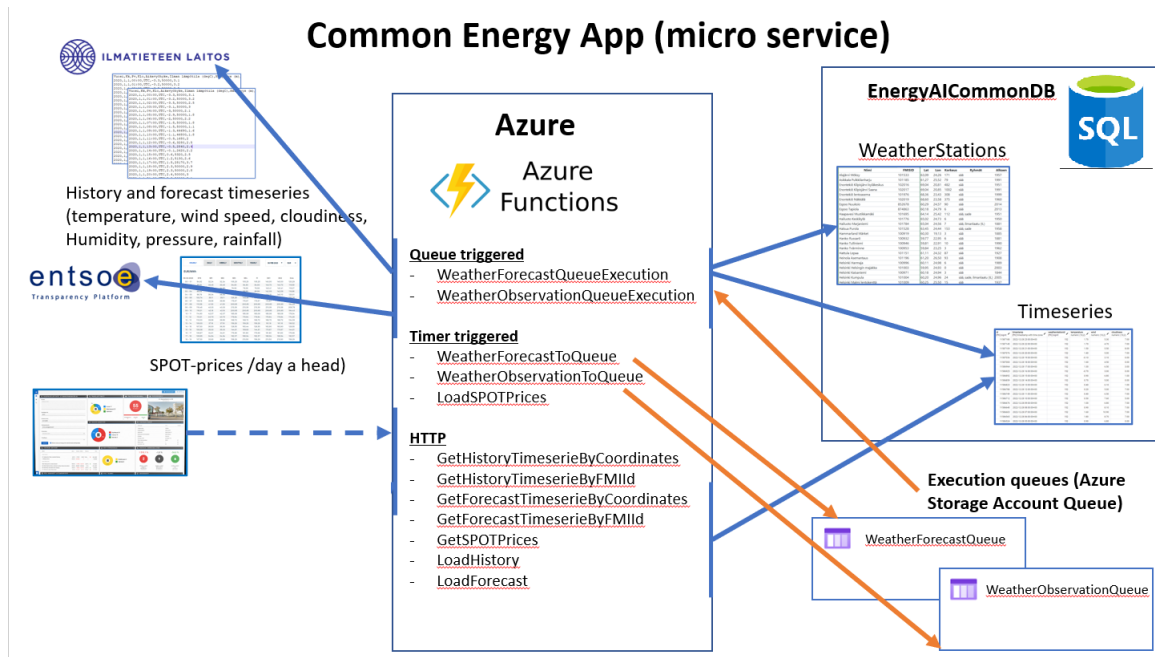
Granlund Manager on myös markkinoiden laaja-alaisin ohjelmisto (*Granlund* 2021). Se sisältää lukuisia ominaisuuksia, kuten huoltokirjan, palvelupyynnöt, huoltosuunnitelman, pitkän tähtäimen suunnitelmat, energianhallinta, dynaaminen raportointi ja integraatiot muihin järjestelmiin. Tässä tutkimuksessa ollaan kiinnostuneita Granlund Managerin energiaprosessista, jonka kautta lisätään jatkossa rakennus poikkeamien havainnointiin. Granlund Managerissa on myös erillinen REST Api, jolla on lukuisia eri toiminnallisuuksia. Sen kautta saadaan haettua rakennusten energiankulutusten tuntisarjoja.

Granlund (2021) on myös laitetietojen hallinta -ohjelmisto, nimeltä Granlund Designer. Se on selaimessa toimiva pilvipohjainen laiteluettelo-ohjelmisto talotekniikan laitetiedon hallintaan. Sitä kautta voidaan luoda ja päivittää sähköisiä laiteluetteloita milloin tahansa suunnittelun, rakentamisen ja ylläpidon aikana. Tarvittavien laitetietojen ollessa samassa paikassa helpottaa rakennuttajia, valvojia ja kiinteistönomistajien arkea huomattavasti.

6.2 Common energy app

Common energy app on mikropalveluarkkitehtuurilla toteutettu sovellus. Sen tarkoituksena on hakea sekä sääennusteita ja -historia tietoja että sähkön SPOT-hintoja. Sääennusteet sekä -historia tiedot haetaan *Ilmatieteen laitos* avoimen datan rajapinnan kautta. Sähkön SPOT-hinnat taas haetaan *entsoe* Transparency Platform RESTful API:n kautta.

Kuva 1 esittää Common energy app:n arkkitehtuurin, kuten se on tutkimuksen ajan hetkellä ollut. Ohjelmisto on toteutettu Azure Funktioilla ja pyörii Azuressa. Kuten kuvasta 1 huomaa, se sisältää jono- (Queue), ajastin- (Timer) ja HTTP-triggeröitäviä funktioita. Tämän lisäksi Common energy App:illa on oma Azure Sql -tietokanta. Tässä tutkimuksessa ollaan kiinnostuneita sää historia tietojen hakemiseen tältä palvelulta.



Kuvio 1. Common energy app arkkitehtuurikuva

7 Ohjelmiston vaatimukset

Tässä kappaleessa käydään läpi tutkimuksessa kehitettävän ohjelmiston vaatimuksia ja mitä ovat ohjelmistojen vaatimukset, joita käydään ensimmäisessä kappaleessa. Toisessa kappaleessa katsotaan alustavia vaatimuksia ja sitä seuraa toiminnallisten ja ei-toiminnallisten vaatimusten käsittely.

7.1 Mitä ovat ohjelmiston vaatimukset?

Vaatimussuunnittelu on tekniikan ala, joka ottaa huomioon reaali maailman järjestelmien tavoitteet, toiminnot ja rajoitukset (Laplante ja Kassab 2022). Siinä käsitellään myös näiden tekijöiden suhdetta järjestelmän määrittysten mukaiseen käyttäytymiseen ja niiden kehitystä sekä ajan myötä että eri järjestelmien välillä. Sommerville (2011) mukaan järjestelmälle asetetut vaatimukset ovat kuvauksia siitä, mitä järjestelmän tulisi tehdä, palveluista joita se tarjoaa ja sen toiminnan rajoitukset. Prosessi, jossa selvitetään, analysoidaan, dokumentoidaan ja tarkistetaan näitä palveluita ja rajoituksia, kutsutaan vaatimussuunnitteluksi (Sommerville 2011).

7.2 Alustavat vaatimukset

Alla oleva lista vetää yhteen ohjelmistolle asetetut alustavat vaatimukset, jotka on ennen työntekoa määritelty. Alustavat vaatimukset luovat pohjan, jonka perusteella projektia rakennetaan. Lisäksi ne toimivat ohjeistuksena suunnittelussa, kehittämisessä ja toteutuksessa.

Projektin alustavat vaatimukset:

1. Poikkeavasti energiaa kuluttavien rakennusten havainnointi koneoppimisen avulla
2. Rakennuksia, joille halutaan luoda tekoälymalleja, pitää pystyä lisäämään Granlund Manager -sovelluksesta. Sovelluksen tulee integroitua sekä Granlund Managerin että toisen mikropalvelun, Common Energy App:n kanssa.
3. Rakennuksilla tulee olla seuraavat vaaditut ominaisuudet. Kaukolämmön tuntisarjaa vähintään vuosi (ja oltava kaukolämmöllä lämmitettävä), pitää olla tiedossa rakennuk-

sen rakennustilavuus, huoneistoala, koordinaatit ja rakennustyyppi.

4. Pitää toteuttaa raportti, josta voidaan vertailla rakennuksia rakennustyypeittäin.
5. Ohjelma tulee kirjoittaa modulaarisesti.
6. Ohjelma tulee toteuttaa mikropalveluarkkitehtuurilla.
7. Ohjelman tulee olla pilvinatiivi

Yllä olevasta listasta huomataan, miten ohjelmistoa lähdetään kehittämään ja mitä ominaisuuksia alustavasti tämä tulee sisältämään.

7.3 Toiminnalliset vaatimukset

Toiminnalliset vaatimukset kuvaavat, mitä palveluita järjestelmän tulisi tarjota, kuinka se reagoi syötteisiin ja kuinka järjestelmä käyttäytyy tietyissä tilanteissa (Sommerville 2011). Sommerville (2011) mukaan nämä vaatimukset ovat riippuvaisia kehitettävästä ohjelmistosta, ohjelmiston odotetuista käyttäjistä ja organisaation yleisestä lähestymistavasta vaatimuksia kirjoittaessa. Laplante ja Kassab (2022) mukaan toiminnallisten vaatimusten tulisi myös määritellä, millainen käytös järjestelmältä ei ole sallittua. Yksinkertaisuudessaan järjestelmän toiminnallisten vaatimusten tulisi siis kuvailla, mitä järjestelmän pitäisi ja ei pitäisi tehdä. Kehitettävän järjestelmän toiminnallisten vaatimusten tulisi olla määritelty johdonmukaisella tavalla (Sommerville 2011). Sommerville (2011) totesi, että suurissa ja monimutkaisissa järjestelmissä tämän saavuttaminen on käytännössä mahdotonta. Hän totesi myös, että toiminnallisten ja ei-toiminnallisten vaatimusten ero ei ole aina selkeä.

Alustavissa vaatimuksissa on muutamia toiminnallisia vaatimuksia. Ensimmäinen on vaatimus #1, joka määrittää mitä sovelluksen tulisi tehdä, eli havaita poikkeamia hyödyntäen koneoppimista. Toiseksi vaatimus #2 määrittää sovelluksen käytön ja keskustelun toisten sovellusten rajapintojen kanssa. Kolmanneksi on vaatimus #3, mikä määrittää lisättäville rakennuksille pakollisia ominaisuuksia. Sovelluksen tulee siis kyetä käsittelemään näitä rakennusten ominaisuuksia. Neljänneksi, vaatimus #4 määrittelee selkeästi, että sovelluksen pitää olla kykeneväinen toteuttamaan raportti, missä voidaan vertailla rakennuksia keskenään.

7.4 Ei-toiminnalliset vaatimukset

Siinä missä toiminnalliset vaatimukset kuvaavat, mitä järjestelmän kuuluisi tehdä, ei-toiminnalliset vaatimukset kuvaavat, miten järjestelmän tulisi ne toteuttaa. Laplante ja Kassab (2022) selittivät, että ei-toiminnalliset vaatimukset käsittelevät järjestelmän havaittavia ominaisuuksia, kuten käytettävyyttä, luotettavuutta tai ylläpidettävyyttä. Sommerville (2011) taas selitti, että ei-toiminnalliset vaatimukset ovat ongelmia, jotka eivät ole suoranaisesti linkitetty järjestelmän tarjoamiin palveluihin. Hän totesi myös, että ei-toiminnalliset vaatimukset ovat usein tärkeämpiä, kuin yksittäiset toiminnalliset vaatimukset. Hänen mukaan, käyttäjät löytävät kiertotien jonkin puutteellisen toiminnallisuuden kiertämiseen, mutta ei-toiminnallisen vaatimuksen ollessa puutteellinen, koko järjestelmästä voi tulla käyttökelvoton.

Ei-toiminnallisia vaatimuksia on myös muutamia listassa. Ensimmäiseksi on vaatimus #5, mikä määrittää, että ohjelmisto tulee kirjoittaa modulaarisesti. Tämä vaatimus edistää järjestelmän muunneltavuutta ja ylläpidettävyyttä. Tämä vaatimus linkittyy hyvin toiseen ei-toiminnalliseen vaatimukseen #6, missä ohjelmiston arkkitehtuuriksi on valittu mikropalveluarkkitehtuuri. Mikropalveluarkkitehtuurin ansiosta sovelluksesta tulee skaalautuvampi ja virheiden paikannus helpottuu. Viimeisenä ei-toiminnallisena vaatimuksena on #7, joka määrittää, että ohjelmiston tulee olla pilvinatiivi. Pilvinatiivisuuden ansiosta saadaan vähennettyä sovelluksen kustannuksia.

7.5 Aluevaatimukset

Aluevaatimukset (Domain requirements) taas ilmenevät toimialueesta, missä ohjelmistoa käytetään. Sommerville (2011) mukaan aluevaatimukset johdetaan järjestelmän sovellusalueesta, eikä järjestelmän käyttäjien tarpeista. Ne voivat olla uusia toiminnallisia vaatimuksia omana itsenään, rajoittaa olemassa olevia toiminnallisia vaatimuksia tai määrittää, kuinka tietyt laskennat on suoritettava. Aluevaatimusten ongelmana on se, että ohjelmistokehittäjät eivät välttämättä ymmärrä kehitettävän sovelluksen toimialueen erityispiirteitä (Sommerville 2011). Sommerville (2011) vielä lisäsi, että ohjelmistokehittäjät eivät usein tunnista, puuttuuko jokin aluevaatimus tai onko ne ristiriidassa jonkin toisen vaatimuksen kanssa. Esimerkiksi toiminnallinen vaatimus ohjelmistolle voisi olla, että sitä kautta pystyy tilaamaan ja

maksamaan tuotteita ilman iän tarkistusta. Toisaalta aluevaatimus määrittelee sen, että joillekin tuotteille on lailliset määräykset suorittaa iän tarkistaminen ennen tilausta.

Alustavista vaatimuksista, yksi lukeutuu myös aluevaatimukseen. Vaatimus #3, josta keskusteltiin toiminnallisissa vaatimuksissa, on myös toimialakohtainen, missä rakennuksilla tulee olla tietyt ominaisuudet, jotta niitä voidaan käyttää ohjelmistossa. Siinä missä tämä vaatimus on toiminnallinen, on se myös samanaikaisesti aluevaatimus.

8 Teknologiavalinnat

Tässä kappaleessa käydään läpi valitut teknologiat, joita on käytetty artefaktien kehittämiseen. Ensimmäiset neljä kappaletta käyvät läpi aikasarjatietokannan, ohjelmointikielien, käyttöliittymän ohjelmistokehityksen ja raportointityökalun valintaa. Viidennessä aliluvussa vedetään yhteen valitut teknologiat.

8.1 Aikasarjatietokanta

Struckov ym. (2019) tutkivat neljää eri aikasarjatietokantaa, Clickhouse, TimescaleDB, InfluxDB ja OpenTSDB. He arvioivat tietokantojen sopivuutta eri käyttötarkoituksiin. Struckov ym. (2019) mukaan TimescaleDB olisi sopivin sellaiseen käyttötarkoitukseen, missä lukuoperaatioita on enemmän kuin kirjoitusoperaatioita.

Lisäksi Stenfors (2021) tutki prosessiautomaation aikasarjadatan tallentamista. Hän suoritti suorituskykytestejä InfluxDB:lle ja TimescaleDB:lle. Suorituskyky testeissä hän testasi tietokantojen kirjoitus- ja lukunopeutta. Hän päätyi lopulta samaan tulokseen, että TimescaleDB on suorituskykyisempi lukuoperaatioissa, kuin InfluxDB.

Myös Timescalen omassa blogissa (Freedman ja Sewrathan 2020) tutkittiin TimescaleDB:n ja InfluxDB:n suorituskykyä. Suorituskykytesteissä TimescaleDB suoriutui merkittävästi paremmin lukuoperaatioissa, kuin InfluxDB. Näiden tutkimusten perusteella aikasarjatietokannaksi valittiin TimescaleDB, koska kehitettävän artefaktiin tärkein käyttötapaus on tietokannasta arvojen lukeminen.

8.2 Ohjelmointikielet

Tämän tutkielman pääasiallisena tavoitteena on luoda rakennusten energiankulutusten poikkeamien havaitsemismalli. Mallia varten oli kaksi kandidaattia ohjelmointikieleksi, Python ja C#. Lopulta päädyin valitsemaan C# ohjelmointikieleksi sen lukuisten hyötyjen, sekä oman kokemukseni perusteella.

C# hyötyjä poikkeamien havaitsemisessa on monia, joista ensimmäinen on se, että se on olio-ohjelmointikieli. Käytännössä se tarkoittaa sitä, että se tukee luokkien, olioiden ja metodien käyttöä. Tämän ansiosta voidaan kirjoittaa modulaarisempaa ja uudelleenkäytettävämpää ohjelmakoodia. Nämä asiat ovat tärkeitä poikkeamien havainnoinnissa, koska työssä tulee mahdollisesti useita algoritmeja tai malleja poikkeamien havaitsemiseen. Yleisesti ottaen staattisesti tyypitetyt ja käännettävät ohjelmointikielet ovat suorituskyvyltään nopeampia kuin dynaamisesti tyypitetyt ja tulkittavat ohjelmointikielet. Tämä voi olla hyödyksi, kun joudutaan käsittelemään suuria määriä dataa. Lisäksi Salihu ja Tafa (2020) vertailivat C# ja Pythonin suorituskykyä konvoluutioneuroverkon toteutuksessa ja C# osoittautui laskennallisesti tehokkaammaksi.

C# sisältää myös edistyneempiä ominaisuuksia, joista on hyötyä poikkeamien havaitsemisessa. Yksi merkittävimmistä on LINQ (Language Integrated Query), jonka avulla voidaan prosessoida ja suorittaa tietokantakyselyitä tehokkaasti. Tästä voi olla paljon apua, kun käytetään isoja tietojoukkoja tai suoritetaan monimutkaisia laskuja poikkeamien havaitsemisessa. Tämän lisäksi voidaan ottaa käyttöön muita .NET teknologioita, kuten ML.NET ohjelmistokehys.

Asiakaspuolen käyttöliittymää varten mahdollisia ohjelmointikieliä ovat TypeScript ja JavaScript. Työssä tullaan laajentamaan olemassa olevaa web-sovellusta. Työtä varten päätin valita TypeScriptin käyttöliittymän ohjelmointikieleksi, koska koen sen olevan parempi kuin JavaScriptin. Henkilökohtaisesti miellän staattisesti tyypitettyjä ohjelmointikieliä kohtaan, joten luontaisesti päätin valita TypeScriptin.

8.3 Ohjelmistokehykset ja kirjastot

Molempiin sekä poikkeamien havainnointiin että asiakaspuolen käyttöliittymään tullaan hyödyntämään ohjelmistokehyksiä. Poikkeamien havainnointiin tullaan käyttämään Microsoft (2023) ML.NET ohjelmistokehystä. ML.NET on avoimen lähdekoodin koneoppimis ohjelmistokehys, joka on myös alustariippumaton. ML.NET:n avulla voidaan uudelleenkäyttää kaikkea tietämystä, taitoa, koodia ja kirjastoja, joita .NET ohjelmistokehittäjällä on ja integroida ne helposti olemassaoleviin web-sovelluksiin. Sen skaalautuvuuden ansiosta voi-

daan työskennellä niin isoilla kuin pienillä tietojoukoilla lukemattomissa laiteissa ja ympäristöissä ja mikä tärkeintä, se mahdollistaa täydellisen pipeline luomisen ja jakamisen tehokkaasti (Ahmed ym. 2019).

Käyttöliittymää varten valittiin Meta (2023) kehittämä avoimeen lähdekoodiin perustuva JavaScript kirjasto, React. Reactin avulla voidaan tehdä uudelleenkäytettäviä käyttöliittymän komponentteja, joka helpottaa ohjelmakoodin ylläpidettävyyttä ja skaalautuvuutta. Tämän lisäksi Reactin kanssa voidaan käyttää muitakin käyttöliittymän ohjelmistokehyksiä ja kirjastoja. React on myös yhteensopiva useiden nykyaikaisten selainten kanssa.

8.4 Raportointityökalu

Työn yhtenä vaatimuksena oli myös mahdollistaa raportti, josta pystytään vertailemaan useita rakennuksia. Raportointia varten päädyttiin käyttämään Power BI:tä. Power BI valittiin raportointityökaluksi sen ominaisuuksien ja kyvykkyyksien vuoksi. Sen kehittyneet datan visualisointityökalut tarjosivat siitä sopivan alustan tälle työlle. Power BI:n avulla myös voidaan helposti integroida useita tietolähteitä ja käsitellä reaaliaikaista dataa.

Tämän lisäksi Power BI:n käyttäjäystävällinen käyttöliittymä teki siitä helppokäyttöisen henkilölle, jolla ei ollut aiempaa kokemusta raporttien luonnista. Päätöksentekoprosessi sisälsi vaihtoehtoisten raportointityökalujen arvioinnin, josta Power BI nousi sopivimmaksi vaihtoehdoksi kattavien ominaisuuksien, skaalautuvuuden ja laajennettavuuden ansiosta.

8.5 Yhteenveto

Alla on yhteenveto kaikista teknologioista, joita on hyödynnetty ohjelmiston kehityksessä.

REST Api, jossa suoritetaan poikkeamien havainnoinnin laskenta:

- Kirjoitettu C#:lla.
- Käytetty ML.NET ohjelmistokehystä koneoppimiseen.
- Tehty HTTP triggeröitävillä Azure Funktioilla.
- TimescaleDB aikasarjatietokanta, jossa säilötään rakennusten energiakulutusta.

Käyttöliittymälaajennus Granlund Manageriin:

- Kirjoitettu TypeScriptillä, käyttäen Reactia.
- Käyttöliittymän tilanhallinta on tehty Redux:lla.
- Käyttöliittymään on käytetty hyödyksi myös KendoReact käyttöliittymä kirjastoa.

Rakennusten vertailun raportti:

- Visualisoitu Power BI:llä.
- Data säilötty raporttia varten Azure SQL tietokannassa.

9 Tulokset

Tämä kappale vetää yhteen tutkimuksen tulokset. Kappale alkaa yhteenvetämällä tutkielmassa tehtyä työtä. Seuraava kappale kuvailee työssä syntyneitä tuotoksia. Viimeiset kaksi kappaletta korostaa tutkimusprojektin onnistumisia ja epäonnistumisia.

9.1 Yhteenveto

Työssä merkittävimmät tehdyt työt olivat:

1. Vaatimusten kerääminen selvittääkseni, mitä minun pitäisi rakentaa. Vaatimuksia arvioitiin ja jalostettiin koko tutkimusprojektin ajan.
2. Teknologioiden valinta kokonaisuuden kehittämistä varten, mukaan lukien ohjelmointikielien, ohjelmistokehykset, kirjastot, aikasarjatietokanta ja raportointityökalu.
3. Itse työn kehittäminen.
4. Tämän tutkielman kirjoittaminen raportoidakseen tulokset.

Vaikka ylläoleva lista on esitetty numeroituna listana, työtä ei silti tehty tiukasti seuraten listan järjestystä ja suurin osa töistä jatkui läpi projektin.

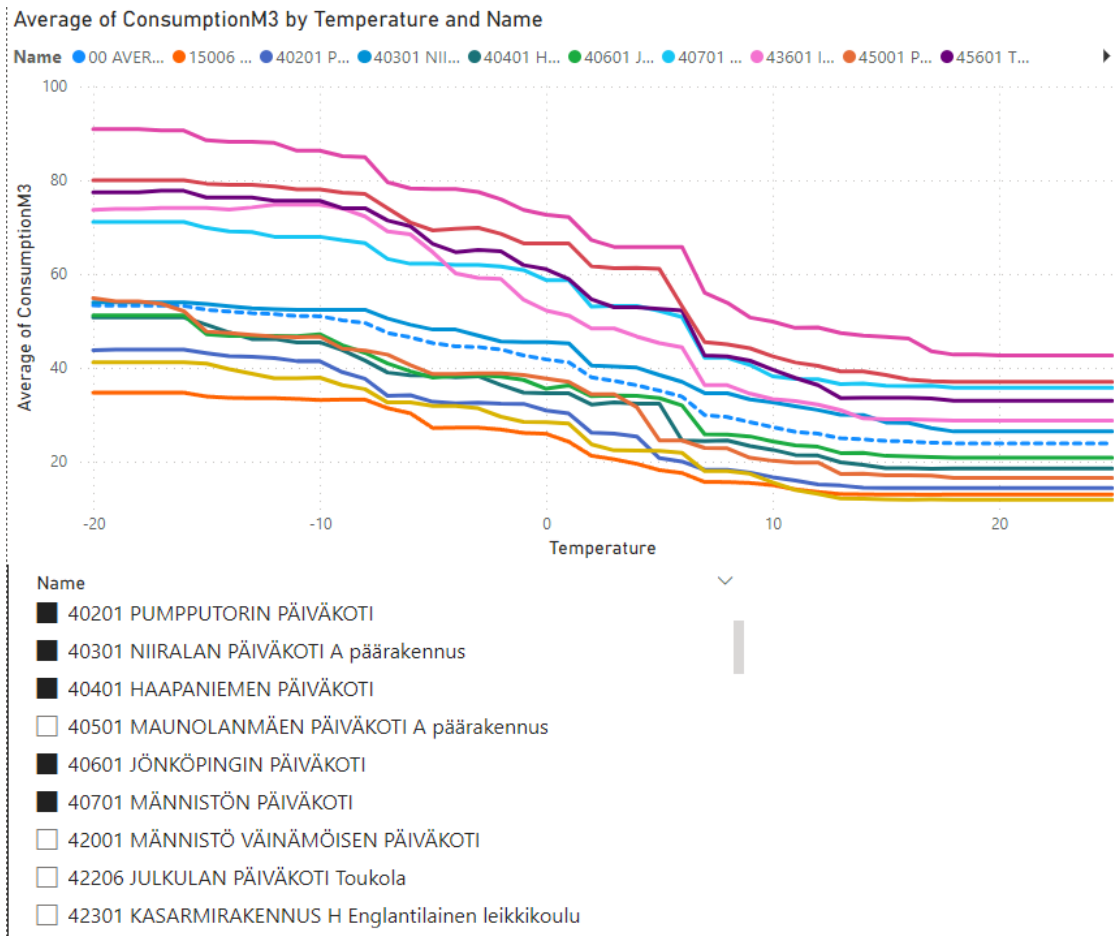
9.2 Tuotokset

Tutkimustyön aikana kehitettiin muutama itsenäinen tuotos, joista muodostui kattava kokonaisuus. Itsenäisiä tuotoksia oli REST Api, jonka kautta suoritetaan poikkeamien havainnointi, raportti, jossa voidaan vertailla rakennuksia ja käyttöliittymä rakennusten lisäämiseksi. Näiden lisäksi myös laajennettiin Granlund Managerin REST Apia, jolla pystytään hakemaan rakennusten energiakulutuksen tuntisarjat.

REST Apin pääasiallisena tarkoituksena oli havaita poikkeamia rakennusten energiakulutuksesta. Tätä ongelmaa lähdettiin aluksi taklaamaan ML.NET:n poikkeamien havainnointi metodeilla. Metodeja, joilla poikkeamia yritettiin havainnoida, oli DetectChangePointBySsa, DetectSpikeBySsa, DetectAnomalyBySrCnn, RandomizedPca ja DetectEntireAnomaly-

BySrCnn. Muutospiste (Change Point) menetit, kuten DetectChangePointBySsa, osoittavat pitkäaikaisia muutoksia aikasarjan arvoissa, joita on esimerkiksi arvojen tason muutokset ja trendit. DetectChangePointBySsa metodilla yritetään ennustaa muutospisteitä käyttäen Singular Spectrum Analysis -menetelmää (Ssa). Piikkien (Spike) havaitsemis metodeissa on tavoitteena tunnistaa äkillisiä, mutta väliaikaisia sarjoja, jotka eroavat merkittävästi suurimmista osista aikasarjan arvoista. DetectSpikeBySsa käyttää samaa Ssa menetelmää piikin havaitsemiseksi. Piikit ovat lyhytaikaisempia muutoksia, kuin muutospisteen havaitsemat muutokset aikasarjan arvoissa. Myös kokeiltiin käyttää SrCnn (Super-resolution convolutional neural network) -metodeja poikkeamien havainnointiin. Näitä oli kahta erilaista, DetectAnomalyBySrCnn ja DetectEntireAnomalyBySrCnn. DetectAnomalyBySrCnn metodilla havaitaan aikasarjan poikkeavuuksia käyttäen SrCnn-algoritmia, kun taas DetectEntireAnomalyBySrCnn metodi havaitsee aikasarjapoikkeamat koko syötteelle käyttäen samaa algoritmia. Viimeiseksi kokeiltiin vielä RandomizedPca metodia. Se luo pääkomponenttianalyysi mallin käyttäen satunnaistettua singulaariarvohajotelma-algoritmia.

Yleiseksi ongelmaksi ML.NET:n poikkeamien havainnointi metodeissa ilmeni se, että ne eivät olleet kovin kypsiä tapauksissa, joissa haluttiin käyttää useampaa muuttujaa (multivariate). Kun käytettiin monimutkaisia aikasarjoja, joissa oli useita muuttujia ominaisuuksina, poikkeamien havainnointi tuntui erittäin hankalalta. Metodeja käyttämällä tekoälymalleja oli vaikea kouluttaa havaitsemaan oikeita poikkeamia. Tekoälymalleille testattiin koulutuksessa eri ominaisuuksia (feature), kuten lämpötila, päivä, kuukausi, onko työpäivä ja eilisen päivän kulutus ja lämpötila. Eri ominaisuuksien yhdistelmiä testattiin, mutta jatkuvasti törmätettiin samoihin ongelmiin. Ongelmia, joita tuli jatkuvasti vastaan oli, että alhaiset kulutukset ja olemattomat muutokset energiankulutuksen aikasarjoissa ilmoitettiin poikkeamina. Se, että liian pienet arvot havaittiin poikkeamina ei ollut niin iso ongelma, sillä ne pystyttiin karsimaan helposti pois. Toisaalta trendien ja tasojen muutosten havaitseminen vaikutti miltei mahdottomalta tehtävältä monimutkaiselle aikasarjalle. Näiden lisäksi kokeiltiin myös luoda oma ohjelma, joka laski energiakulutuksen keskiarvon per aste käymällä läpi aikasarjojen arvot. Ohjelma antoi suurimmalle osalle rakennuksista epäluotettavia tuloksia, joten se ei ollut validi ratkaisu. Poikkeamien havainnointia yritettiin pitkään saada toimimaan edellä mainituilla keinoilla, ennen kuin päädyttiin valittuun ratkaisuun.



Kuvio 2. Raportti rakennusten normeeratuista ominaiskulutuskäyristä

Lopulta päädyttiin innovatiiviseen ratkaisuun, jossa lasketaan rakennuksille normeerattu ominaiskulutuskäyrä, jonka perusteella voidaan havaita poikkeavan paljon kuluttavat rakennukset. Tätä varten luotiin rakennustyyppikohtainen raportti 2, jossa voidaan vertailla rakennusten normeerattua ominaiskulutusta keskiarvoon ja toisiin rakennuksiin. Rakennusten normeeratulla ominaiskulutuskäyrällä tarkoitetaan keinoa tarkastella rakennuksen energiankulutusta lämpötilan suhteen. Ja, koska energiankulutus on normeerattua, voidaan rakennuksia vertailla rakennustyypeittäin. Rakennusten normeerattu ominaiskulutuskäyrä on uusi termi lanseerattu rakennusalan energia-asiantuntijoita varten. Raportilla voit itse valita rakennukset, joita haluat vertailla keskiarvoon. Tämä keskiarvo on laskettu kaikkien asiakkuuksien rakennusten normeeratuista ominaiskulutuksista, mutta käyttäjä voi vertailla vain oman asiakkuuden rakennuksiin siihen. Asiakkuudella tässä kontekstissa tarkoitetaan Granlundin asiakasta, joka käyttää Granlund Manager -ohjelmistoa. Raportin tarkoituksena on havain-

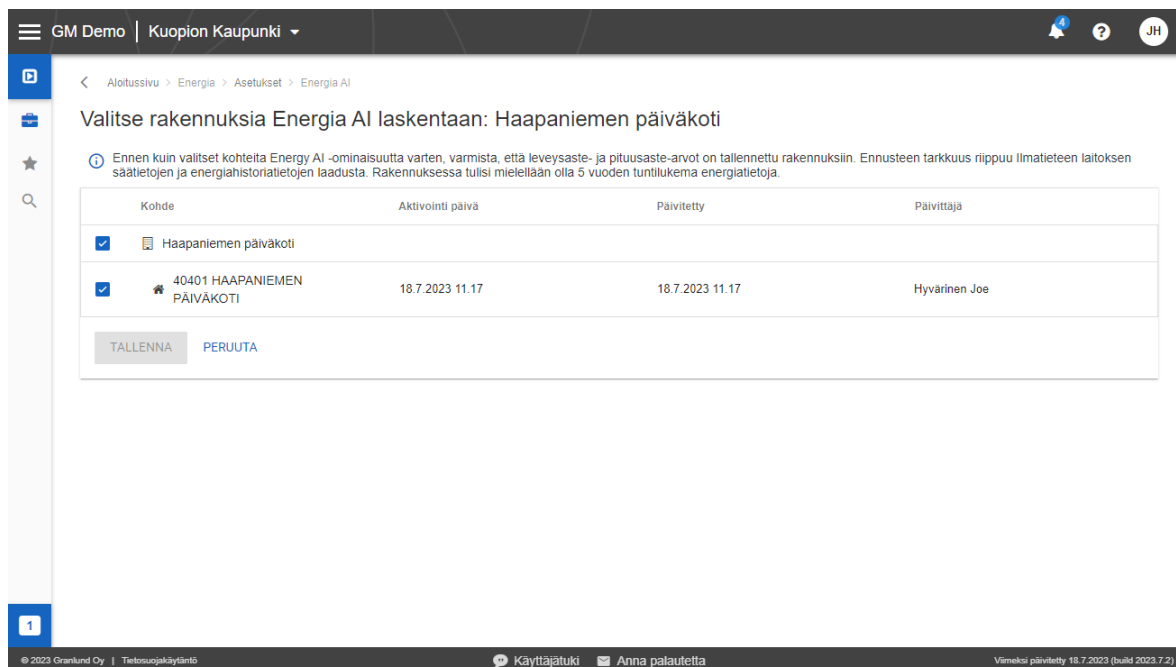
nollistaa, miten paljon tietty rakennus tai rakennukset kuluttavat energiaa verrattuna muihin saman rakennustyyppin rakennuksiin yli asiakkuuksien. Esimerkiksi, energia-asiantuntijat pystyvät näyttämään raporttia asiakkailleen ja havainnollistamaan sen avulla, että joku heidän omistamista rakennuksista kuluttaa enemmän energiaa kuin muut rakennukset. Täten energia-asiantuntijat näkevät suoraan poikkeavan paljon kuluttavat rakennukset ja pystyvät sitä kautta etsimään syitä ja ratkaisuja rakennuksen poikkeavalle energiakulutukselle.

Rakennusten normeerattu ominaiskulutus on laskettu lämpötilojen -20 ja +25 välisille arvoille, johon on käytetty ML.NET:n FastForest metodia laskemaan ominaiskulutukset. Tekoälymallille annettiin koulutuksessa ominaisuuksina lämpötila, kuukausi, onko työpäivä ja edellisen päivän kulutus. Eri ominaisuuksia ja niiden yhdistelmiä kokeiltiin mallin koulutuksessa. Aiemmin mainittujen ominaisuuksien lisäksi kokeiltiin myös antaa ominaisuuksina päivä ja edellisen päivän lämpötila. Dataa, jota sen koulutuksen käytettiin, haettiin Granlund Managerin REST Apista, jonne on säilötty rakennusten energiakulutuksen tuntisarjoja.

Alustavasti tuntisarjat oli tarkoitus säilöä valittuun aikasarjatietokantaan, TimescaleDB:een, mutta ajallisista syistä, sitä ei toteutettu tämän tutkimuksen yhteydessä. Energiakulutuksen aikasarjan lisäksi mallille annettiin samalta ajalta säätietoja koulutukseen. Säätiedot haettiin olemassa olevasta mikropalvelusta, Common Energy App:sta, joka hakee tiedot ilmatieteenlaitokselta. Mallin koulutuksessa testattiin eri aikasarjan aggregointeja, kuten tuntisarjaa, päiväsarjaa ja kuukausisarjaa. Parhaaseen mallin tarkkuuteen, edellä mainittujen ominaisuuksien kanssa, lopulta päästiin käyttäen päiväsarjaa mallin koulutuksessa. Mallin tarkkuus oli keskimäärin 90%, kun paras mahdollinen tarkkuus on 100%, ja se riippui paljon rakennuksen energiakulutuksen aikasarjasta, sekä sen alueen säätiedoista. Aikasarjat ja säätiedot haettiin viiden vuoden takaa, jotta saataisiin mahdollisimman paljon arvoja eri lämpötiloillemallin koulutukseen. On myös hyvä tuoda esille, että tällä samalla ratkaisulla pystytään myös hyvin ennustamaan rakennuksen tuleva kulutus sääennustuksen mukaan.

Toisena tuotoksena tutkimusprojektissa syntyi käyttöliittymä 3, jossa käyttäjä voi lisätä rakennuksen poikkeaman havainnointiin. Tällä sivulla suoritetaan kappaleen 7.2 vaatimuksen #3 määrittämiä validointeja. Kuten edellisestä kappaleesta huomataan, ilman rakennustilavuutta ei voida laskea rakennuksen ominaiskulutuskäyrää. Tämän lisäksi, riittämättömällä määrällä energiakulutuksen tuntisarjaa malleista tuli epätarkkoja. Tämän takia rakennukset,

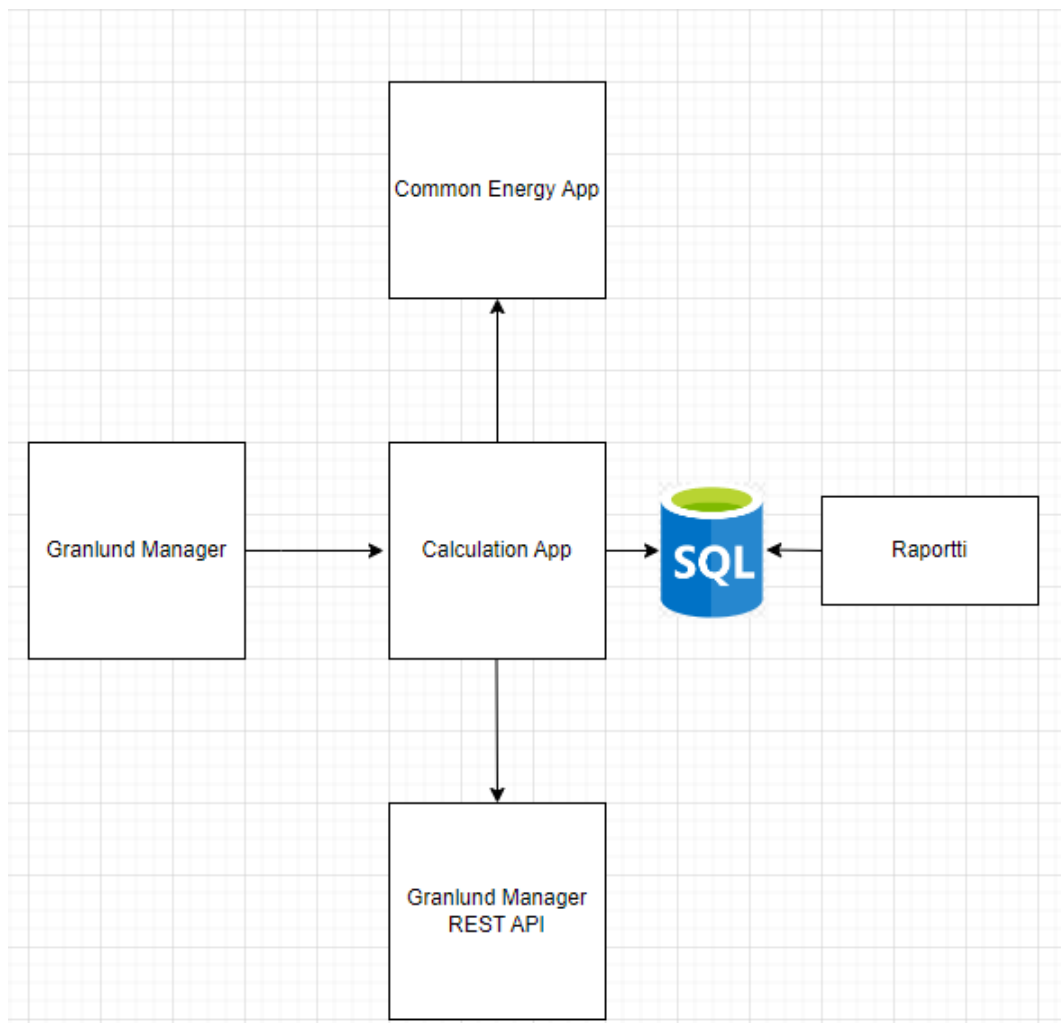
joilla ei ole riittävästi energiakulutuksen tuntisarjaa, ei otettu mukaan laskentaan. Koordinaatitkin ovat välttämättömän tärkeitä olla rakennuksella, jotta saadaan haettua rakennuksen lähimmän sääaseman säähistoria tiedot.



Kuvio 3. Granlund Managerin uusi asetussivu

Kuvasta 4 huomataan kokonaisuudesta muodostunut ylätason arkkitehtuuri, missä työssä kehitetty sovellus on nimetty Calculation App:ksi. Asetus sivun kautta, kun tallennetaan rakennus laskettavaksi, kutsutaan kehitettyä sovellusta. Tämän jälkeen haetaan tarvittavat tiedot mallin koulutusta varten. Säähistoriatiedot haetaan Common energy app:sta ja rakennuksen energiankulutukset haetaan Granlund Managerin REST Apista. Normeeratun ominaiskulutuskäyrän laskennan jälkeen arvot tallennetaan Azure Sql -tietokantaan, jonka kautta Power BI saa tiedot raportilla.

Tutkimassa tuotettu menetelmä havaita poikkeamia on yksi mahdollinen tapa suorittaa tehtävä. Tämän vuoksi esitän kaksi aiempaa kokonaisuutta konkreettisina tutkimuksen tuotoksina. Näiden lisäksi, tutkimuksessa tuotettiin muitakin tuotoksia. Nämä tuotokset syntyivät sekä tiedosta, joka mahdollisesti kahden aiemmin mainitun tuotoksen kehittämisen, että näitä tuotoksia kehitettäessä saaduista opeista.



Kuvio 4. Kokonaisuuden ylätasen arkkitehtuuri

Ennen kehitystyön alkua, määritettiin kehitettävälle kohteelle olennaisia vaatimuksia. Tätä varten määritettyjen vaatimusten tulokset on esitetty kappaleessa 7. Vaatimukset antoivat suuntaa ja pohjaa kehitystyön aloittamiselle. Näin ollen esitän määritetyt vaatimukset omana tuotoksenaan.

Vaatimusten lisäksi teknologioiden valinta työtä varten auttoi sen aloittamisessa huomattavasti. Varsinkin ohjelmointikieli, jolla itse koneoppimista lähdettiin toteuttamaan. Ilman sen kummempaa taustatyötä, ohjelmointikieleksi olisi voinut valita Pythonin sen valtavan huomion ansiosta, koneoppimisen keskuudessa. Lopulta, kun päädyin valitsemaan C# ohjelmointikieleksi koen, että valinta oli oikea. Esitän myös kappaleessa 8.5 listatut teknologiat

omana tuotoksenaan. Käytettyjä teknologioita voidaan jatkossakin käyttää uusien koneoppimiskokonaisuuksien kehittämiseen.

Kaiken kaikkiaan tutkimuksessa hypättiin syvään päätyyn ison poikkeamien havainnoinnin kokonaisuuden kehittämisessä. Poikkeamien havainnointi on ollut tutkimusalana kovassa kasvussa viime aikoina ja tällä tutkimuksella saadaan tuotua uudenlaista näkökulmaa sille, tarkemmin rakennusten energiakulutusten poikkeamien havainnointiin.

9.3 Mikä onnistui?

Ensimmäiseksi, olen tyytyväinen kehitystä varten valittuihin teknologioihin. Teknologiat, joita käytettiin, ovat moderneja ja tunnettuja, jonka takia tätä olisi helppo jatkokehittää muidenkin toimesta. Erityisesti tyytyväinen olen ohjelmointikielen C# valinnasta Pythonin yli. Uskon, että varsinkin ML.NET:n kehittyessä, C# mahdollisuudet koneoppimisen hyödyntämisessä tulevat olemaan suuret. Tarkempia tietoja teknologia valinnoista löydät kappaleesta 8.

Toiseksi, kokonaisuuden kehittämisen onnistumisesta olen tyytyväinen. Vaikka alkuperäinen suunnitelma meni uusiksi, missä oli tarkoituksena luoda poikkeamien havainnoinnin tekoälymalli, olen silti tyytyväinen lopputulokseen. Ottaen huomioon tutkimukseen liittyvät riskit 2.4, varsinkin aikarajoitteet, koen onnistuneeni hyvin lopullisen kokonaisuuden suhteen. Tämän lisäksi energia-asiantuntijoilta tuli hyvin positiivista palautetta, kun näytin heille raportin ensimmäistä versiota. Yksi kysymys, minkä heiltä heti kuuli, oli: "Saako tätä jo näyttää asiakkaalle?".

Lopuksi, mielestäni tutkimusmenetelmän valinta oli oikea tähän tutkimukseen. Tutkimusmenetelmän ansiosta pystyin luomaan uuden ratkaisun rakennusten energiankulutuksen poikkeamien havainnointiin rakennustyypeittäin. Tämän lisäksi olen tyytyväinen tutkimuksen teoriaosuuteen. Se antaa hyvän pohjustuksen keskeisistä asioista, jotka on olennaista tietää.

9.4 Missä olisi parannettavaa?

Ensimmäiseksi parannuksen kohteeksi nostaisin esille sen, että näin laajassa kokonaisuudessa ja tutkimusaiheessa, kannattaisi keskittyä vain johonkin tiettyyn asiaan. Esimerkiksi, tutkimuksessa olisi voitu keskittyä pelkästään rakennustyypeittäin rakennuksen energiankulutuksen poikkeamien havainnoinnin tekoälymallin kehittämiseen ja tutkia pelkästään sitä. Loppujen lopuksi antaa tällaisen kokonaisuudenkin kehittäminen ja tutkiminen uudenlaista kulmaa koko poikkeamien havainnoinnin tutkimusalalle.

Toiseksi, vaikka tutkimusmenetelmän valinta oli mielestäni oikea, olisi sen käyttöä pitänyt suunnitella tarkemmin. Sen soveltaminen tämän työn kannalta tapahtui melko sattumanvaraisesti. Tälle olisi voinut suunnitella jonkin tietyn tutkimusprosessin, jota noudattaa. Esimerkiksi olisi voinut olla hyvä idea kehystää kokonaisuuden kehittäminen ja tutkimuksen kirjoittaminen ennalta määritettyihin iteraatioihin. Näin jokaisella iteraatiolla olisi voinut olla jokin tavoite ja koko prosessia olisi voinut olla helpompi arvioida ja analysoida.

10 Pohdinta

Tutkielmassa tutkittiin rakennusten energiankulutuksen poikkeamien havainnointia rakennustyypeittäin. Useiden haasteiden kautta, saatiin kehitettyä toimiva kokonaisuus tälle tutkimusalalle. Tutkimuksessa kartoitettiin mahdollisia teknologioita ja rajoituksia tällaisen kokonaisuuden kehittämiseen. Kappaleessa 2.2 määritettiin kaksi tutkimuskysymystä, joita käydään seuraavaksi läpi

Ensimmäinen kysymys oli löytää keino rakennusten energiankulutusten poikkeamien havainnointiin rakennustyypeittäin. Alustavana suunnitelmana oli luoda tekoälymalli hyödyntäen ML.NET:n poikkeamien havainnoinnin metodeja. Matkan varrella tuli vastaan haasteita, joita käsiteltiin aliluvussa 9.2. Lopulta päädyttiin uuteen innovatiiviseen ratkaisuun, jota on tarkemmin kuvailtu myös aiemmin mainitussa luvussa. Olen tyytyväinen ratkaisuun, johon tutkimuksessa päädyttiin ja uskon sen olevan kelvollinen ratkaisu tähän ongelmaan.

Toinen tutkimuskysymys käsitteli mahdollisia teknologioita kokonaisuuden kehittämiseksi. Kuten aiemmassa kappaleessa mainitsin, olen tyytyväinen teknologioihin, jotka valitsin. Tällaisen kokonaisuuden kehittämiseen on olemassa lukemattomia määriä eri teknologioita, mutta uskon, että valitsemani teknologiat antavat vähintäänkin hyvän joukon teknologioita, joita kannattaa miettiä tällaista ratkaisua luodessa.

Tarkastellessani valmista tuotetta, uskon että se oli hyvä yritys toteuttaa tällainen kokonaisuus ja se antaa uutta näkökulmaa rakennusten energiankulutuksen poikkeamien havainnoinnin tutkimusalalle. Vaikka tämä on tutkimusalana ollut kasvussa, on sillä vielä paljon työtä edessä saavuttaakseen sen täyden potentiaalin. Alla on listattu muutamia jatkotutkimuksen aiheita, joita tästä tutkimuksesta pystyi johtamaan:

- Ratkaisun vertailu tavanomaiseen poikkeamien havainnoinnin ratkaisuun.
- Poikkeavan paljon kuluttavien rakennusten ominaisuuksien tutkiminen Granlund Designer ja Manager ohjelmistoista.
- Analysoi, miten rakennuksen energiankulutuksen poikkeavuudet liittyvät poikkeamiin muissa rakennusjärjestelmissä, kuten LVI, valaistus ja ilmanvaihto.
- Eri rakennustyyppien energiankulutusten kuvioiden ja mallien tutkiminen. Kehittää

kuviontunnistustekniikoita poikkeamien havaitsemiseksi keskiverto kulutuskäyttäytymisestä.

- Vertailla tavanomaista relaatiotietokantaa ja aikasarjatietokantaa (TimescaleDB) rakennuksen aikasarjoilla. Hyödyt ja haitat.

Kaiken kaikkiaan uskon, että tämä tutkimus oli kohtuullisen onnistunut toteuttamaan ratkaisun, jolla voidaan havaita poikkeamia rakennusten energiankulutuksista rakennustyypeittäin. Vaikka sitä ei saatu tehtyä perinteikkäästi tekoälymallin avulla, uskon että se silti tuottaa uudenlaista näkemystä tälle tutkimusalalle. Koen, että tästä tutkimuksesta on yleisesti hyötyä alan ammattilaisille.

Lähteet

Adhikari, Ratnadip, ja Ramesh K Agrawal. 2013. “An introductory study on time series modeling and forecasting”. *arXiv preprint arXiv:1302.6613*.

Ahmed, Zeeshan, Saeed Amizadeh, Mikhail Bilenko, Rogan Carr, Wei-Sheng Chin, Yael Dekel, Xavier Dupre, Vadim Eksarevskiy, Senja Filipi, Tom Finley ym. 2019. “Machine learning at Microsoft with ML. NET”. Teoksessa *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2448–2458.

Bishop, Christopher M, ja Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*. Nide 4. 4. Springer.

Box, G.E.P., ja G.M. Jenkins. 1970. *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day. ISBN: 9780816210947. <https://books.google.fi/books?id=5BVfnXaq03oC>.

Buildings, Global Alliance for, International Energy Agency Construction ja the United Nations Environment Programme. 2019. *2019 global status report for buildings and construction: towards a zero-emission, efficient and resilient buildings and construction sector*.

Cao, Li-Juan, ja Francis Eng Hock Tay. 2003. “Support vector machine with adaptive parameters in financial time series forecasting”. *IEEE Transactions on neural networks* 14 (6): 1506–1518.

Chandola, Varun, Arindam Banerjee ja Vipin Kumar. 2009. “Anomaly detection: A survey”. *ACM computing surveys (CSUR)* 41 (3): 1–58.

Cunningham, Pádraig, Matthieu Cord ja Sarah Jane Delany. 2008. “Supervised learning”. Teoksessa *Machine learning techniques for multimedia: case studies on organization and retrieval*, 21–49. Springer.

Elfeky, Mohamed G, Walid G Aref ja Ahmed K Elmagarmid. 2005. “Periodicity detection in time series databases”. *IEEE Transactions on Knowledge and Data Engineering* 17 (7): 875–887.

- entsoe. https://transparency.entsoe.eu/content/static_content/Static%20content/web%20api/Guide.html.
- Fan, Yugang, Ping Li ja Zhihuan Song. 2006. “Dynamic least squares support vector machine”. Teoksessa *2006 6Th World Congress on Intelligent Control and Automation*, 1:4886–4889. IEEE.
- Freedman, Mike, ja Avthar Sewrathan. 2020. “TimescaleDB vs. InfluxDB: Purpose Built Differently for Time-Series Data”. <https://www.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/>.
- Ghahramani, Zoubin. 2003. “Unsupervised learning”. Teoksessa *Summer school on machine learning*, 72–112. Springer.
- Granlund. 2021. <https://www.granlund.fi/palvelut/granlund-manager-ohjelmisto/>.
- Granlund. 2021. <https://www.granlund.fi/palvelut/laitetiedon-hallinta/>.
- Gregor, Shirley, David Jones ym. 2007. “The anatomy of a design theory”. Association for Information Systems.
- Hamzaçebi, Coşkun. 2008. “Improving artificial neural networks’ performance in seasonal time series forecasting”. *Information Sciences* 178 (23): 4550–4559.
- Hastie, Trevor, Robert Tibshirani, Jerome H Friedman ja Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Nide 2. Springer.
- Hawkins, Douglas M. 1980. *Identification of outliers*. Nide 11. Springer.
- Hayes, Adam. 2022. “What Is a Time Series and How Is It Used to Analyze Data?” <https://www.investopedia.com/terms/t/timeseries.asp>.
- Himeur, Yassine, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali ja Abbes Amira. 2021. “Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives”. *Applied Energy* 287:116601.
- Hodge, Victoria, ja Jim Austin. 2004. “A survey of outlier detection methodologies”. *Artificial intelligence review* 22:85–126.

IBM. 2021. “Characteristics of time series”. <https://www.ibm.com/docs/en/spss-modeler/saas>.

Ilmatieteen laitos. <https://www.ilmatieteenlaitos.fi/avoin-data>.

Joshi, Prateek, John Hearty, Bastiaan Sjardin, Luca Massaron ja Alberto Boschetti. 2016. *Python: Real world machine learning*. Packt Publishing Ltd.

Koivula, Kim. 2016. “Rakennuksen energiatehokkus”.

Kärkimaa, Milja. 2015. “Tampereen ammattikorkeakoulun pääkampuksen ominaiskulutukset”.

Laplante, Phillip A, ja Mohamad H Kassab. 2022. *Requirements engineering for software and systems*. CRC press.

Meta. 2023. “Introducing react.dev – React”. <https://react.dev/blog/2023/03/16/introducing-react-dev>.

Microsoft. 2023. “ML.NET | Machine learning made for .NET”. <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>.

Naqvi, Syeda Noor Zehra, Sofia Yfantidou ja Esteban Zimányi. 2017. “Time series databases and influxdb”. *Studienarbeit, Université Libre de Bruxelles* 12.

Peppers, Ken, Tuure Tuunanen, Marcus A Rothenberger ja Samir Chatterjee. 2007. “A design science research methodology for information systems research”. *Journal of management information systems* 24 (3): 45–77.

Riak. 2023. “Enterprise NoSQL Database | Scalable Database Solutions | Riak”. <https://riak.com/resources/time-series-databases/>.

Salihu, Bashkim, ja Zhibert Tafa. 2020. “On Computational Performances of the Actual Image Classification Methods in C# and Python”. Teoksessa *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, 1–5. IEEE.

Seppänen, Olli. 2001. *Rakennusten lämmitys*. 2., päivitetty painos. Helsinki: Suomen LVI-yhdistysten liitto.

Sommerville, Ian. 2011. “Software engineering (ed.)” *America: Pearson Education Inc.*

- Stenfors, Risto. 2021. “Prosessiautomaation aikasarjadataan tallentamisratkaisu”. Tutkielma.
- Struckov, Alexey, Semen Yufa, Alexander A Visheratin ja Denis Nasonov. 2019. “Evaluation of modern tools and techniques for storing time-series data”. *Procedia Computer Science* 156:19–28.
- Sutton, Richard S, ja Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Suykens, Johan AK, ja Joos Vandewalle. 2000. “Recurrent least squares support vector machines”. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47 (7): 1109–1114.
- Tilastokeskus. 2018. *Tilastokeskus*. <https://www2.stat.fi/fi/luokitukset/rakennus/>.
- Torrey, Lisa, ja Jude Shavlik. 2010. “Transfer learning”. Teoksessa *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, 242–264. IGI global.
- Vaishnavi, Vijay, William Kuechler ja Stacie Petter. 2004. “Design science research in information systems”. *January* 20:2004.
- Van Engelen, Jesper E, ja Holger H Hoos. 2020. “A survey on semi-supervised learning”. *Machine learning* 109 (2): 373–440.
- Wang, Xiaozhe, Kate Smith-Miles ja Rob Hyndman. 2009. “Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series”. *Neurocomputing* 72 (10-12): 2581–2594.
- Weiss, Karl, Taghi M Khoshgoftaar ja DingDing Wang. 2016. “A survey of transfer learning”. *Journal of Big data* 3 (1): 1–40.
- Vesala, Jarna. 2022. “Lämmitysenergiankulutuksen eroavaisuudet kerrostaloasunnoissa: asunnon sijainnin vaikutus kulutukseen”.
- Rakentamismääräykset*. 2023. <https://ym.fi/rakentamismaaraykset>.
- Ympäristöministeriön asetus asuin-, majoitus- ja työtiloista*. 2017. <https://www.finlex.fi/fi/laki/alkup/2017/20171008t>.

Zhang, G Peter. 2003. "Time series forecasting using a hybrid ARIMA and neural network model". *Neurocomputing* 50:159–175.

Zhang, Guoqiang, B Eddy Patuwo ja Michael Y Hu. 1998. "Forecasting with artificial neural networks:: The state of the art". *International journal of forecasting* 14 (1): 35–62.

Zhu, Xiaojin Jerry. 2005. "Semi-supervised learning literature survey".