



Lauri Leppälä

Koneoppimisen hyödyntäminen järjestelmäintegraatiossa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

16.4.2024

Tiivistelmä

Tekijä: Lauri Leppälä
Otsikko: Koneoppimisen hyödyntäminen järjestelmäintegraatiossa
Sivumäärä: 57 sivua
Aika: 16.4.2024

Tutkinto: Insinööri (AMK)
Tutkinto-ohjelma: Tieto- ja viestintätekniikka
Ammatillinen pääaine: Ohjelmistotuotanto
Ohjaaja: Osaamisaluejohtaja Janne Salonen

Insinööriyössä esitellään ohjelmisto ja sen prosessi, jonka tavoitteena oli nopeuttaa kiinteistön rakennusautomaatioverkon ja web-pohjaisen analysointiohjelman välistä liitostyötä. Ongelma järjestelmien liitostyössä on löytää tarvittavat tietolähteet rakennusautomaatioverkon skannaustiedostosta ja kohdentaa tarvittaville tietolähteille analysointiohjelmia varten tarvittavat tiedot. Lisäksi rakennusautomaatioverkon tietolähteille tulee laatia kansiorakenteet ja mittarilista, joka sisältää tietolähteiden osoitteet ja tarvittavat tiedot analysointiohjelmia varten sekä luoda tiedonsiirtotiedosto varten palvelimen ohjelmalle.

Ohjelma laadittiin käyttämällä Python-ohjelmointikieltä ja siinä käytettiin pääasiassa Pandas- ja Numpy-kirjastoja. Ohjelma koostuu kahdesta erikseen käytettävistä ohjelmista: MetersToRAU.py ja ADDStandardNODEIDS.py. Ohjelman MetersToRAU.py tarkoituksena on etsiä rakennusautomaatioverkon skannaustiedoston suuresta tietomäärästä tarvittavat tietolähteet ja niiden perusteella luoda tiedonsiirron tiedosto, kansiorakenteiden tiedosto sekä mittarilista. Uusilla mittarilistoilla laajennetaan ohjelman opetusaineistoa. Koneoppimisen algoritmi vertaa uusia tietolähteitä olemassa olevaan tietokantaan. Ohjelman ADDStandardNODEIDS.py tarkoituksena oli lisätä mittarilistoihin vaadittavat kansiorakenteiden tunnukset ja nimet käyttämällä ohjelman MetersToRAU.py luomaa mittarilistatiedostoa sekä kansiorakennetiedostoa.

Ohjelmalla saavutettiin hyvä tulos tekemällä haku 1948 rivin pituiselle rakennusautomaation skannaustiedostolle, jolla ei ollut aiemmin tietoja opetusaineistossa. Tarvittavia tietolähteitä oli 105 kpl. Ilman opetusaineistoa löydettiin 75% tarvittavia tietolähteitä opetusaineistosta (50839 kpl), joista virheellisten raporttiryhmiin osuus oli 41%. Opetusaineistoon lisättiin 45 kpl tietolähteitä, jonka jälkeen löydettiin tarvittavat tietolähteet ja oikeat raporttiryhmit. Ohjelma kykeni löytämään ja luomaan oikeat tiedot samoja tunnuksia käyttäville tietolähteille samoista ryhmistä, joka nopeuttaa integraatiotyötä huomattavasti.

Avainsanat: koneoppiminen, Python

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check-ohjelmalla.

Abstract

Author: Lauri Leppälä
Title: Utilizing machine-learning in system integration
Number of Pages: 57 pages
Date: 16 April 2024

Degree: Bachelor of Engineering
Degree Programme: Information and communication technologies
Professional Major: Software engineering
Supervisor: Janne Salonen, Head of expertise

This final thesis presents a program and a process which target was to speed up the process of integrating meters from buildings automation network to web-based analyzing tool. Main problem in integration work is to find needed meters from building automation network scan file and to point needed results for needed meters. In integration work is also needed to create an archive-structures file and a meterlist that includes meters addresses and needed information for web- based analyzing tool and to create a data transfer file for server's program.

Program was created using a Python-programming language and using mainly Pandas- and Numpy- libraries. The program has two separately used programs: MetersToRAU.py and ADDStandardNODEIDS.py. Program MetersToRAU.py is meant to find needed meters from big data mass of building automation network scan file and based on founded data to create a data transfer file, an archive-structure file and a meter list. Training data is expanded with new meter lists. Machine-learning model algorithm compares scan file data to existing training data. Program ADDStandardNODEIDS.py adds credentials to archive-structures and names by using MetersToRAU.py program's created meter list and archive-structure file.

Good result was achieved by making a search for 1948 pieces of meters in building automation network scan file which information didn't exist in training data before. Needed amount of information sources was 105 pieces. Without training data 75% of needed meters was found from training data (50839 meters) which included 41% faulty reportgroups. To training data was added information of 45 pieces of needed meters and after new scan all needed meters with right reportgroups was found. Program was able to find other meters that used same credentials that were added to training data which speeds up integration process significantly.

Keywords: Python, machine-learning

Sisällys

Lyhenteet

1	Johdanto	1
2	Koneoppiminen on tekoälyn osa-alue	2
2.1	Koneoppimismallien päätyypit	2
2.2	Koneoppimismallien päätyypit	4
2.3	Ohjelmassa hyödynnettävät koneoppimismallit	5
2.3.1	Lähimmän naapurin luokitin	5
2.3.2	Looginen päättely	5
3	Ohjelman käyttötarkoitus	7
3.1	Tarvittavien tietojen määrittäminen ohjelmaa varten	9
3.2	Ohjelman periaatteellinen toiminta	12
4	Ohjelman MetersToRAU.py toiminta	14
4.1	Tietokantojen luonti	15
4.1.1	Tuntemattomien syötetietojen tietokannan luonti	15
4.1.2	Opetusaineiston tietokannan luonti	16
4.1.3	Tyhjä excel-pohja analysointiohjelmaa varten	17
4.2	Tietolähteiden nimien käsittely	18
4.3	Tarpeettomien tuntemattomien syötetietojen suodatus	23
4.4	Meters_no_groups.xlsx-tiedoston luonti	24
4.5	Meters_to_RAU.xlsx-tiedoston luonti	24
4.6	RAU-sourcefile.csv-tiedoston luonti	28
4.7	Nodetree.xlsx-tiedoston luonti	29
5	Ohjelman ADDStandardNODEIDS.py toiminta	38
6	Ohjelmien tulosten arviointi	47
6.1	Opetusaineiston lisäys NEW_meters.xls-tiedostoon	49
6.2	Ohjelman uudelleen suoritus lisätyn opetusaineiston kanssa	51
7	Johtopäätökset	56
	Lähteet	57

Lyhenteet

.csv	comma separated values = pilkulla erotetut arvot (tiedoston tallennusmuoto, jossa käytettävät merkkijonot on eroteltu pilkulla)
.xlsx	Microsoft Excel Open Xml Spreadsheet (Microsoft Excel-tiedostojen tallennusmuoto)
.py	Python-ohjelmointikielen ohjelman suoritustiedosto

1 Johdanto

Työssä esitellään ohjelmisto, jolla on automatisoitu rakennusautomaatioverkosta tehtävää tiedon hakua erillistä analysointiohjelmaa varten tehtävää integraatiotyötä varten. Ohjelmisto on tehty toimimaan rakennusautomaatioverkon analysointiohjelman ja rakennusautomaatioverkon skannausohjelman välillä. Rakennusautomaatioverkon skannausohjelma on ohjelma, joka lukee ja lähettää tietoja rakennusautomaatioverkosta analysointiohjelmaan. Analysointiohjelma on web-pohjainen ohjelmisto, jolla analysoidaan rakennusautomaatioverkon mittareiden tietoja. Analysointiohjelmaa käytetään kiinteistöjen taloteknisten järjestelmien analysointiin, joita ovat mm. palo-,turva-,lämpö-, vesi- ja ilmanvaihtojärjestelmät.

Rakennusautomaatioverkon ja analysointiohjelman järjestelmien integraatiotyössä tuodaan vain tarvittavat mittarit rakennusautomaatioverkosta analysointiohjelman tietokantaan, jonka avulla tehdään energijärjestelmien analysointia sekä tarkkaillaan järjestelmien toimivuutta. Tarvittavia mittareita käytetään analysointiohjelman säännöissä, raporteissa ja muissa toiminnallisuuksissa.

Opinnäytetyössä tutkittiin mitä koneoppimisen malleja ohjelmassa on sovellettu sekä käydään läpi ohjelman toiminta ja tulokset. Tuloksissa on vertailtu ohjelman kykyä luoda analysointiohjelmaa varten tarvittavat tiedot rakennusautomaatioverkon skannaustiedostosta ilman kohteen mittareille luotua opetusaineistoa sekä opetusaineistoon lisätyillä kohteen mittaritiedoilla.

2 Koneoppiminen on tekoälyn osa-alue

Tekoäly on laaja käsite ja se ei rajoitu pelkästään tietokoneiden ohjelmointiin vaan se käsittää joukon erilaisia menetelmiä, teknologioita, sovelluksia ja tutkimussuuntia. Tekoälyn avulla pyritään tehostamaan, helpottamaan ja automatisoimaan prosesseja, jotka vaatisivat muuten paljon ihmistyövoimaa työn suoritukseen. Tarkoituksena on saada erilaiset prosessit toimimaan oikein tehtävien ja tilanteiden vaatimustason mukaan toivotulla tavalla [1].

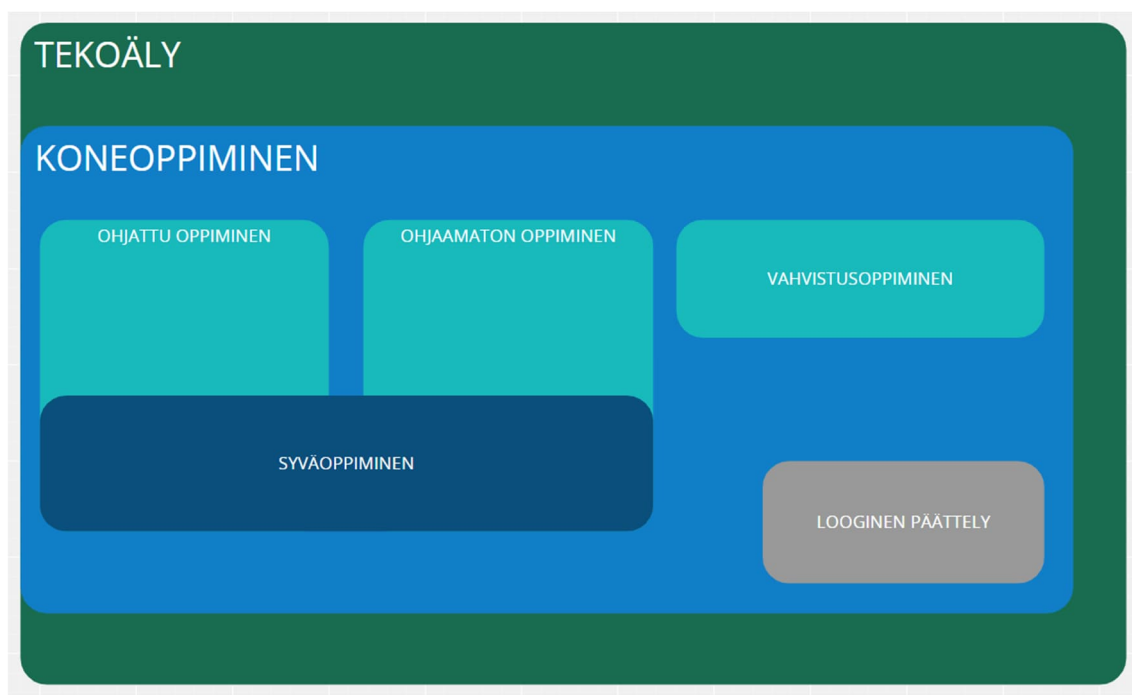


Kuva 1. Tekoälyn 10 keskeistä osaamisaluetta

Koneoppiminen on osa tekoälyä ja koneoppimista voisi parhaiten kuvailla siten, että sen avulla pyritään etsimään tiedosta säännönmukaisuuksia ja matkimaan ihmisen toimintaa. Ihmisen toimintaa matkimalla vähennetään käsin tehtävää työtä ja nopeutetaan prosessien kulkua ja pyritään automatisoimaan tehtäviä. Säännönmukaisuuksilla tarkoitetaan sitä, että tunnistetaan tiettyjen tapahtumien ja asioiden toistuvuuksia, esimerkiksi liikennesuuhkien esiintyminen tietyillä alueilla tiettyyn aikaan. Koneoppimismallia ohjelmoitaessa tehdään algoritmeja, jotka ovat sääntöjä miten tietoa käsitellään eri tapauksissa. Algoritmit ohjaavat ohjelman toimintaa ja käsittelevät tietoa.

2.1 Koneoppimismallien päätyypit

Koneoppimismallit jaetaan pääasiassa kolmeen päätyyppiin, joita ovat ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen [2, s 152].



Kuva 2. Koneoppiminen osana tekoälyä

Syväoppiminen on ohjatun ja ohjaamattoman koneoppimisen osa-alue, joka perustuu neuroverkkoihin, jotka jäljittelevät ihmisaivojen toimintaa.

Syväoppiminen ei ole mahdollista ilman suurta määrää koulutusdataa sekä laskentakapasiteettia [3, s. 122]. Looginen päättely edustaa tekoälyn klassisen koulukunnan koneoppimismallia ja siksi sitä ei usein mainita koneoppimisen materiaaleissa. Loogisen päättelyn keskeisessä roolissa ovat asiantuntijajärjestelmät, jotka ovat rakennettu loogisen päättelyn pohjalta [2, s.174]. Loogisen päättelyn perustoiminta on tehdä päätelmiä onko asia tosi vai epätosi. Asiantuntijajärjestelmät hyödyntävät tunnettuja esitietoja ja tosiasioita tieto- ja tietämuskannassa ja tekee niiden perusteella päätöksen onko kysytty asia totta vai ei. Huomioitavaa on, että jos asiantuntijajärjestelmän tietokannan ja tietämuskannan voi rakentaa automaattisesti opetusaineistosta aikaisempien ohjatun oppimisen menetelmien tapaan niin se on koneoppimista [2, s.181].

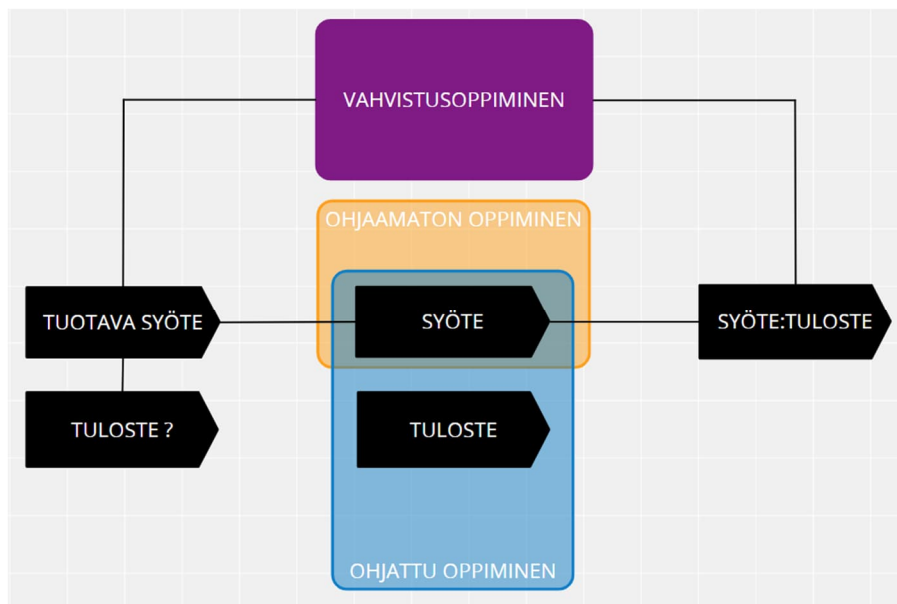
2.2 Koneoppimismallien päätyypit

Koneoppimismallien toiminta erotellaan niiden kyvystä käyttää syötteitä ja tulosteita.

Ohjatussa oppimismallissa tunnetaan syöte ja tuloste, joiden joukkoa kutsutaan opetusaineistoksi. Opetusaineistossa on kaikki tieto mitä ohjelma tietää. Tavoitteena on löytää syötteiden ja tulosteiden perusteella tuotavalle syötteelle tulos [2, s.152].

Ohjaamattomassa oppimisessä on käytössä vain syötetiedot eikä apuna käytetä tulostetietoa vaan tieto ryhmitellään ominaisuuksien perusteella ja löydetään niiden välille merkitys [2, s. 94]. Ohjaamattoman oppimisen malli on esimerkiksi ryvästys, jossa annetusta näytejoukosta etsitään ryppäitä, tiivistymiä ja sen perusteella tehdään päätöksiä [2, s.152].

Vahvistusoppimisessä ei ole käytössä syötteitä tai tulosteita vaan se oppii vuorovaikutuksen kautta [2, s. 152 - 153]. Toimintamallissa pyritään ympäristöstä saatavan palautteen avulla parhaaseen tulokseen.



Kuva 3. Ohjattu, ohjaamaton ja vahvistusoppiminen

Ohjattu oppiminen jaetaan kahteen alalajiin, joita ovat luokittelu ja regressio. Luokittelumallissa tutkittava tieto voidaan jakaa eri ryhmiin ja regressiossa mitataan jatkuvasti muuttuvaa tietoa. Luokittelumallia käytetään esimerkiksi sähköpostin jaotteluun roskapostiin ja regressiomallia esimerkiksi lämpötilan määrittämiseen [2, s. 94]. Ohjatun oppimisen menetelmiä ovat mm. lähimmän naapurin menetelmä, päätöspuu ja satunnaismetsä [2, s.152].

2.3 Ohjelmassa hyödynnettävät koneoppimismallit

Ohjelmien tietokannoilla oli selkeät syöte- ja tulosteparit, joten oli loogista käyttää ohjatun oppimismallin luokittelua luokittelualgoritmien luomisessa. Loogisella päättelyllä rakennettiin mm. kansiorakennetiedosto, jossa tietolähteet luokiteltiin niiden ominaisuuksien perusteella ryhmiin.

2.3.1 Lähimmän naapurin luokitin

Lähimmän naapurin luokitin on ohjatun oppimismallin lineaarisen luokittelun malli. Lineaarisen luokittelun perusajatuksena on selvittää kuuluuko uusi tuntematon syöte olemassa olevaan opetusaineistoon ja tavoitteena on löytää sille tuloste [2, s.39].

Lähimmän naapurin luokitin mittaa luokiteltavan syötteen etäisyyden kaikkiin olemassa oleviin syötteisiin ja valitsee etäisyyksistä lyhimmän ja palauttaa sitä vastaavan tulosteen [2, s.41-42].

2.3.2 Looginen päättely

Loogisen päättelyn osalta ohjelmistossa on hyödynnetty lauselogiikkaa. Lauselogiikan avulla selvitetään kuuluuko lause joukkoon tosi (true) tai epätosi (false). Esimerkiksi tehdään ohjelmaan toiminta, jossa kaikki sarakkeen arvot, jotka ovat alle 100 ovat tosia (1) ja muut epätosia (0) ja tehdään tuloksen perusteella jatkotoiminto ohjelmaan. Lauselogiikassa

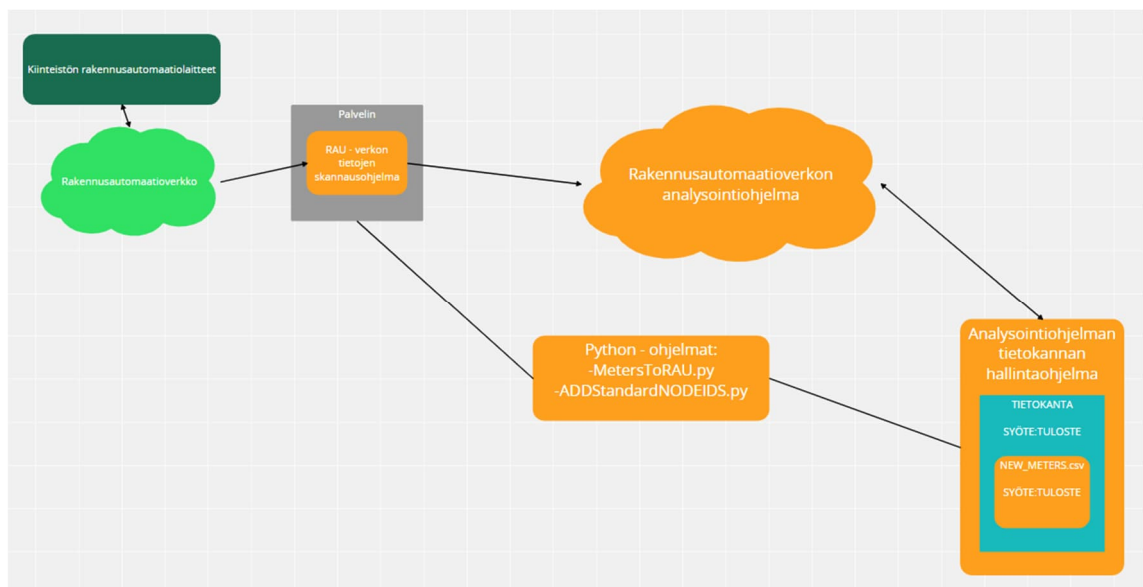
merkityksen, merkinnän ja lukutavan yhteys on kuvattu alla olevassa taulukossa [2, s.177].

Taulukko 1. Lauselogiikan loogisia symboleita ja niiden merkityksiä

Merkitys	Merkintä	Lukutapa
negaatio	$\neg P$	ei P
konjunktio	$P \wedge Q$	P ja Q
disjunktio	$P \vee Q$	P tai Q
implikaatio	$P \rightarrow Q$	jos P niin Q
ekvivalenssi	$P \leftrightarrow Q$	P jos ja vain jos Q

3 Ohjelman käyttötarkoitus

Python-ohjelmat MetersToRAU.py ja ADDStandardNODEIDS.py on rakennettu toimimaan rakennusautomaatioverkon skannausohjelman ja analysointiohjelman tietokannan hallintajärjestelmän välillä ja niitä käytetään manuaalisesti. Rakennusautomaatioverkon skannausohjelma on ohjelmisto, jolla luetaan rakennusautomaatioverkossa toimivien laitteiden tietoja ja lähetetään rakennusautomaatiopisteiden mittaustietoja analysointiohjelman tietokantaan. Skannausohjelma välittää rakennusautomaatiopisteiden mittaustietoja analysointiohjelman tietolähteen tyyppistä riippuen 15 – 60 minuutin välein.



Kuva 4. Järjestelmien ja ohjelman yhteyksien prosessikuvaus

Analysointiohjelmaa käytetään mm. kiinteistöjen taloteknisten järjestelmien energiankulutuksen seurantaan, järjestelmien analysointiin ja vikojen tunnistamiseen, jotka toimivat algoritmeihin rakennettujen sääntöjen perusteella. Sääntö voi tarkkailla esimerkiksi tuloilmakoneen lämmöntalteenoton hyötysuhdetta ja ilmoittaa mikäli hyötysuhde tippuu alle tarkkailuarvon tai seurata lämmitysverkoston painetta ja hälyttää mikäli verkoston paine tippuu alle tarkkailuarvon.

Analysointiohjelman tietokannan hallintajärjestelmä on ohjelma, jolla hallitaan analysointiohjelmassa olevaa tietokantaa (tietolähteitä sekä kansiorakenteita). Analysointiohjelman tietokanta koostuu kansiorakenteista, jotka sisältävät rakennusautomaatiopisteiden tiedot, tietolähteet. Jokaiselle tietolähteelle on määritelty tiedot, joita käytetään analysointiohjelman säännöissä sekä muissa toiminnoissa. Tietolähteillä tulee olla analysointijärjestelmässä seuraavat tiedot:

- mittarin nimi (esim. tuloilman lämpötila TK01TE10 BVAK1 TK01 T10)
- tietolähde (esim. BVAK1'A'TK01'T10'T)
- solmun (kansion) tunnus ja nimi (esim. NodeID: 123 Nodelocation:TK01)
- väliaine (esim. lämpötila)
- väliaineen tunnus (esim. 621)
- raporttiryhmä (esim. tuloilman lämpötila)
- raporttiryhmän tunnus (esim. 100021)
- mittayksikkö (esim. °C)
- intervalli (esim. 15 min) kertoo miten usein mittaustietoa lähetetään järjestelmään
- näyttötila (esim. hetkellinen), kertoo mittaustiedon pakkaustavasta. Kumulatiiviset mittarit kuten vesimittarit lisäävät seuraavan mittauksen tuloksen edelliseen
- mittarityyppi (esim. muu mittari), mittarit jaotellaan tyyppien mukaan sen perusteella mitä ne mittaavat

Ohjelmien tavoitteena oli ratkaista liitosongelmat rakennusautomaation ja analysointiohjelman välillä, joita ovat:

- määrittää tarvittavat rakennusautomaatiopisteet (syötteet) rakennusautomaatioverkon skannaustiedostosta
- määrittää tarvittaville syötteille tarvittavat tiedot (tulosteet): Medium (väliaine), Medium (väliaineen tunnus), ReportGroup (raporttiryhmä), ReportGroupID (raporttiryhmän tunnus), UOM (yksikkö), Interval, Displaymode ja Metertype
- luoda kansiorakenne-tiedosto ja kohdentaa tietolähteet oikeisiin kansioihin
- luoda tiedonsiirtotiedosto rakennusautomaatiopisteiden tietojen lähettämistä varten tarvittavista syötteistä

3.1 Tarvittavien tietojen määrittäminen ohjelmaa varten

Rakennusautomaatioverkosto on tietoverkko, johon on liitetty kiinteistön laitteet, jotka keräävät tietoa sekä mittaavat ja säätelevät kiinteistön taloteknisten järjestelmien toimintaa. Ongelmana rakennusautomaatioverkoston ja analysointiohjelman välistä liitosta tehtäessä on määrittää tarpeelliset pisteet rakennusautomaatioverkostosta, joita tarvitaan analysointijärjestelmän toimintoja varten. Rakennusautomaatioverkosto voi sisältää järjestelmän koosta riippuen tuhansista kymmeniin tuhansiin rakennusautomaatiomittareiden tietoa ja näistä analysointijärjestelmä tarvitsee toimiakseen arviolta noin 5-10% kaikista rakennusautomaation mittareiden tiedoista. Tämä tarkoittaa siis, että kohde, joka sisältää noin 5000 mittarin tietoa tulee näistä analysointijärjestelmää varten liittää noin 250-500 mittarin tietoa. Loput rakennusautomaation mittareiden tiedot ovat turhia analysointijärjestelmän kannalta ja ne sisältävät esimerkiksi lämpötilakäyrien x- ja y-kordinaattien tietoja tai lämpötilojen hälytystietoja. Turhien tietojen lukeminen ja lähetys analysointijärjestelmään skannausohjelmasta kuormittaa myös turhaan rakennusautomaatioverkon toimintaa sekä skannausohjelmaa.

Analysointiohjelman säännöt tarvitsevat toimiakseen jokaisesta tarvittavasta rakennusautomaation mittarista seuraavat tiedot: Medium (väliaine), MediumID (väliaineen tunnus), ReportGroup (raporttiryhmä), ReportGroupID (raporttiryhmän tunnus), UOM (yksikkö), interval (intervalli), displaymode (näyttötila), metertype (mittarityyppi). Lisäksi tietolähde tulee liittää oikeaan kansioon. Rakennusautomaation mittareita kutsutaan analysointiohjelman järjestelmässä tietolähteiksi (Datasource). Tietolähde on esimerkiksi mallia B'A'TK'FanEh'PCtr'Fe'm3s.

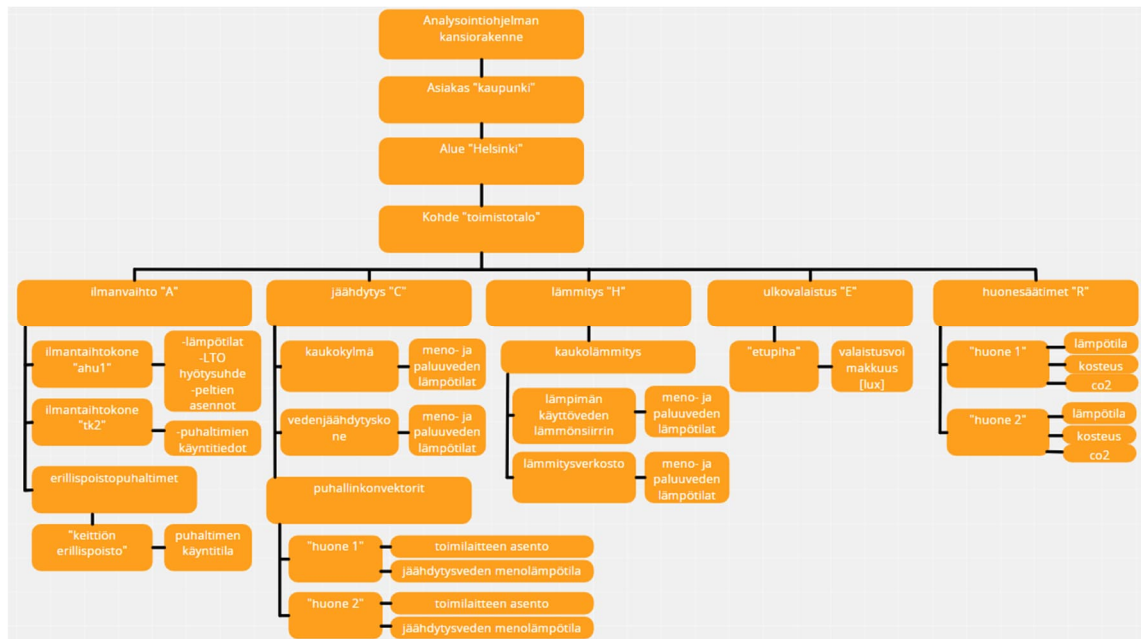
D	E	F	G	H	I	J	K	L	M	N	O
Datasource	N	No	Ol	At	De	De	Medium	MediumId	ReportGroup	ReportGroupId	UOM
B'A'TK'FanEh'PCtr'Fe'm3s							FDD Ilmavirta	576	Poistoilmavirta	10000020	m ³ /s
B'A'TK'FanEh'PCtr'Scc							FDD Käyntinopeus (%)	587	Poistoilmapuhaltimen taajuusmuuttajan säätöviesti	100000609	%
B'A'TK'FanSu'PCtr'Fe'P							FDD Paine	611	Tuloilmanpaine IV-koneella	100001180	Pa
B'A'TK'FanSu'PCtr'Scc							FDD Käyntinopeus (%)	587	Tuloilmapuhaltimen taajuusmuuttajan säätöviesti	100000611	%
B'A'TK'FanEh'PCtr'Fe'P							FDD Paine	611	Poistoilmanpaine IV-koneella	100001176	Pa
B'A'TK'FanSu'PCtr'Fe'm3s							FDD Ilmavirta	576	Tuloilmavirta	100000021	m ³ /s
B'H'DshFncT'QQ							Lämpö (57)	57	Lämmitys (2)		2 MWh
BVAK'A'TK'ErcP'LTOc							FDD Asento (%)	608	LTO pyörimisnopeuden säätö (%)	100001163	
BVAK'A'TK'TT							FDD Lämpötila	621	Tuloilman lämpötila	100000520	
BVAK'H'IV'P							FDD Paine	611	Lämmitysverkoston paine	100001501	bar
BVAK'U'Eg'VIm'VImOut							Kylmävesi (7)	7	Kylmävesi (16)		16 m ³
BVAK1'Uas01'TE00'T							FDD Lämpötila	621	Ulkolämpötila		1 °C
BVAK1'E'AL02'val							FDD Lämpötila	621	Ulkolämpötila		1 °C

Kuva 5. Esimerkki tietolähteistä ja tiedoista

Tietolähteiden muodot noudattavat rakennusautomaatioverkostoon luotujen kansioiden hierarkiaa ja nimet muodostuvat sen perusteella mikä on niiden toiminta kussakin järjestelmässä. Tietolähteiden nimet on eroteltu käyttäen erikoismerkkejä kuten ' tai , tai / . Tietolähteiden nimien luomiseen ei ole olemassa standardia, joten esimerkiksi ilmanvaihtokoneessa voidaan käyttää lyhenteitä: ahu, tk tai tik.

Esimerkiksi tietolähde: B'A'TK'FanEh'PCtr'Fe'm3s tarkoittaa rakennusautomaatioverkoston kansiorakenteessa: B (kohde)'A (ilmanvaihto)'TK (tuloilmanvaihtokone)'FanEh (poistoilmapuhallin)'PCtr (poistoilmapuhaltimen ohjaus)'Fe (ilmavirran mittausanturi)'m3s (yksikkö). Kyseinen tietolähde lähettää siis mittaustietoa ilmanvaihtokoneen poistoilmapuhaltimen ilmavirrasta [m³/s].

Tietolähteet (Datasource) tulee järjestää analysointijärjestelmää varten oikeisiin kansiorakenteisiin, jotta järjestelmien toimivuutta tarkkailevat säännöt toimivat oikein. Tietolähteet tulee ryhmitellä analysointijärjestelmän kansioihin siten, kuten ne on luotu rakennusautomaatioverkkoon.



Kuva 6. Analysointiohjelman kansiorakenne

Järjestelmätason kansioita (kuten ilmanvaihto ja jäähdytys) on kymmeniä erilaisia, mutta järjestelmätasoon liittyviä alikansioita kymmenistä satoihin. Esimerkiksi jos kohteessa on huonesäätimiä 150 huoneessa niin silloin järjestelmätason kansioon: Olosuhteiden hallinta, sisään tulee luoda 150 kansiota. Ohjelman käyttö nopeuttaa mittareiden liittämistä oikeisiin kansioihin kun käsitellään suurta määrää tietoa.

Vaatuksena rakennusautomaatioverkostosta analysointijärjestelmään liitettävistä tietolähteistä on määritettävä:

- tarvittavat syötetiedot (tietolähteet) rakennusautomaatioverkostosta
- tarvittaville tietolähteille vaaditut tiedot analysointiohjelman sääntöjen hallintaa varten: Medium (väliaine), MediumID (väliaineen tunnus), ReportGroup (raporttiryhmä), ReportGroupID (raporttiryhmän tunnus), UOM (yksikkö), interval (intervalli), displaymode (näyttötila) ja metertype (mittarityyppi)
- kansiot analysointijärjestelmään ja kohdentaa tuotavat syötetiedot oikeisiin kansioihin
- tarvittavien tietolähteiden tiedonsiirtotiedosto skannausohjelmaa varten, joka sisältää tarvittavien pisteiden tiedot (mm. tietolähteen

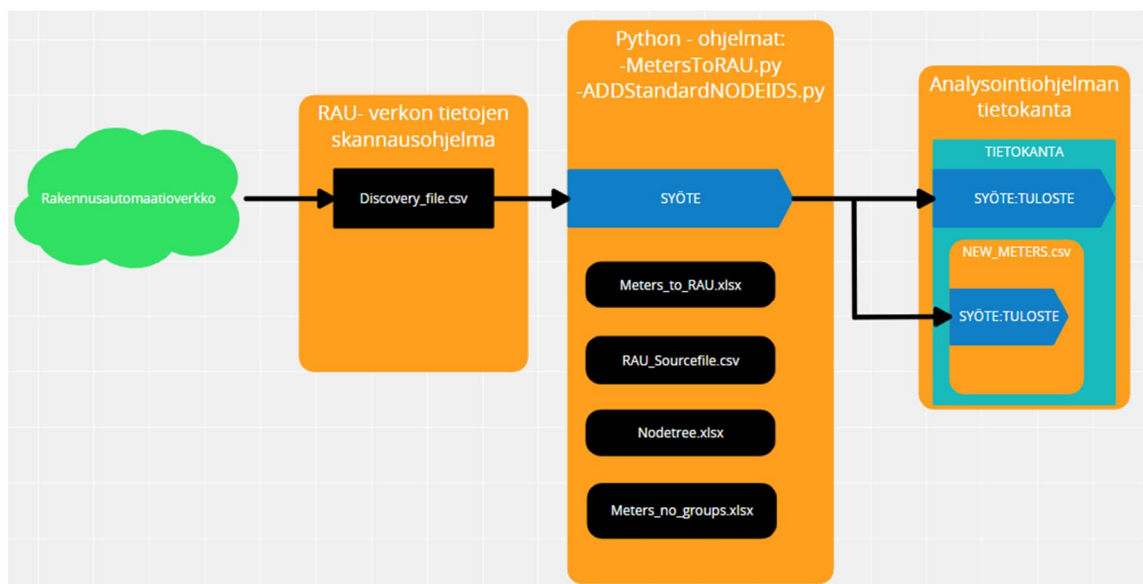
nimi (B'A'TK'FanEh'PCtr'Fe'm3s), yksikkö (esim.degrees_celcius),
verkon protokolla (esim.BACNET), verkon ip-osoite, jne)

3.2 Ohjelman periaatteellinen toiminta

Ohjelma sisältää kaksi erillistä Python-ohjelmaa: MetersToRAU.py (1263 riviä koodia) ja ADDStandardNODEIDS.py (113 riviä koodia). Ohjelma

MetersToRAU.py tekee 4 tiedostoa:

- Meters_to_RAU.xlsx, joka sisältää tarvittavien tietolähteiden tiedot analysointijärjestelmään
- RAU_Sourcefile.csv, joka sisältää tiedonsiirtotiedoston skannausohjelmaa varten
- Meters_no_groups.xlsx, joka sisältää tietolähteet joille ei löytynyt tuloksia opetusaineistosta
- Nodetree.xlsx, joka luo listan analysointijärjestelmään tehtävistä kansiorakenteista



Kuva 7. Ohjelman toimintakaavio

Ohjelma MetersToRAU.py käsittelee skannausohjelmalla tehdyn discovery_file.csv-tiedoston, joka sisältää rakennusautomaatioverkon skannattujen laitteiden tiedot eli syötteet. Tuntemattomille syötteille etsitään

parhaat vastaavat syötetietolähteet ja liitetään niiden tulosteet (tiedot) tuntemattomille syötteille opetusaineistosta. Opetusaineistona toimii analysointiohjelman tietokannan hallintajärjestelmän mittarilistat, joihin on tallennettu olemassa olevien mittareiden tiedot. Ohjelmaa varten tietokannasta ladattiin 50839 tietolähteen tiedot. Tuntemattomista syötteistä suodatetaan pois syötteet, joita ei ole tallennettu tietokantaan (opetusaineisto). Mikäli halutaan määrittää tuntemattomille syötteille tulostiedot (raporttiryhmä, väliaine, yksikkö, jne) niin ne tehdään käsin NEW_METERS.csv-tiedostoon järjestelmätasolla. Ohjelman algoritmi on rakennettu niin, että saman järjestelmätason laitteet, jotka käyttävät samoja tunnuksia löytyvät kun opetusaineistoon määritetään yhden laitteen yksilöidyt tunnukset.

Ohjelma ADDStandardNODEIDS.py kohdentaa tiedoston Meters_to_RAU.xlsx tuotavat syötetiedot oikeisiin kansioihin tiedoston Nodetree.xlsx tietojen perusteella. Ohjelmat tuli tehdä erillisinä, koska ohjelman MetersToRAU.py suorittamisen jälkeen Nodetree.xlsx-tiedoston pääryhmät: asiakas, kohde sekä järjestelmät kuten ilmanvaihto, lämmitys ja jäähdytys tulee luoda käsin analysointijärjestelmään web-käyttöliittymän kautta, jonka jälkeen kansiot saavat yksilöidyt numerotunnukset. Luodut numerotunnukset liitetään tiedostoon Nodetree.xlsx käsin. Tämän jälkeen tiedosto Nodetree.xlsx voidaan ladata analysointiohjelman tietokannan hallintajärjestelmään, jossa ne saavat yksilöidyt kansionumerot. Tämän jälkeen ohjelma osaa kohdentaa oikeat numero- ja nimitunnukset jokaiselle tietolähteelle.

Yleensä rakennusautomaatiojärjestelmiin on määritelty mm. lämpötilojen nimet yksilöidysti kuten TE1 (tuloilman lämpötila), TE2 (poistoilman lämpötila), TE3 (tuloilman lämpötila lämmöntalteenoton jälkeen), TE4 (poistoilman lämpötila lämmöntalteenoton jälkeen) ja TE5 (raitisilman lämpötila ennen lämmön talteenottoa). Tästä johtuen ennestään tuntemattoman kohteen skannaus ohjelmalla vaatii yksilöityjen tietolähteiden merkitsemisen NEW_METERS.csv-tiedostoon, joka toimii uusien tuntemattomien tietojen opetusaineistona. Ohjelman algoritmi toteutettiin niin, että riittää kun opetusaineistoon määritetään yksilöidyt tunnukset yhdelle samantyyppisten päätteiden ryhmän käyttävälle

laitteelle jos ne on ohjelmoitu rakennusautomaatiojärjestelmässä käyttämään samanpäätteisiä tunnuksia. Alla olevassa kuvassa on esimerkki miten ilmastointikoneen yksilöidyt tunnuksat on ohjelmoitu rakennusautomaatioverkkoon ja miten ne esitetään analysointiohjelman mittarilistojen tiedostoissa.

D	E	F	G	H	I	J	K	L	M	N	O
Datasource	Ni	No	Ol	At	De	De	Medium	MediumId	ReportGroup	ReportGroupId	UOM
kaupunki'Helsinki'toimistotalo'A'ahu1'TE2'TE							FDD Lämpötila	621	Poistoilman lämpötila	100000006	°C
kaupunki'Helsinki'toimistotalo'A'ahu1'TE4'TE							FDD Lämpötila	621	Poistoilman lämpötila LTO:n jälkeen	100001052	°C
kaupunki'Helsinki'toimistotalo'A'ahu1'TE5'TE							FDD Lämpötila	621	Raitisilman lämpötila ennen LTO:a	100001319	°C
kaupunki'Helsinki'toimistotalo'A'ahu1'TE1'TE							FDD Lämpötila	621	Tuloilman lämpötila	100000520	°C
kaupunki'Helsinki'toimistotalo'A'ahu1'TE3'TE							FDD Lämpötila	621	Tuloilman lämpötila LTO:n jälkeen	100001291	°C

Kuva 8. Esimerkki ilmastointikoneen tietolähteistä ja tiedoista

4 Ohjelman MetersToRAU.py toiminta

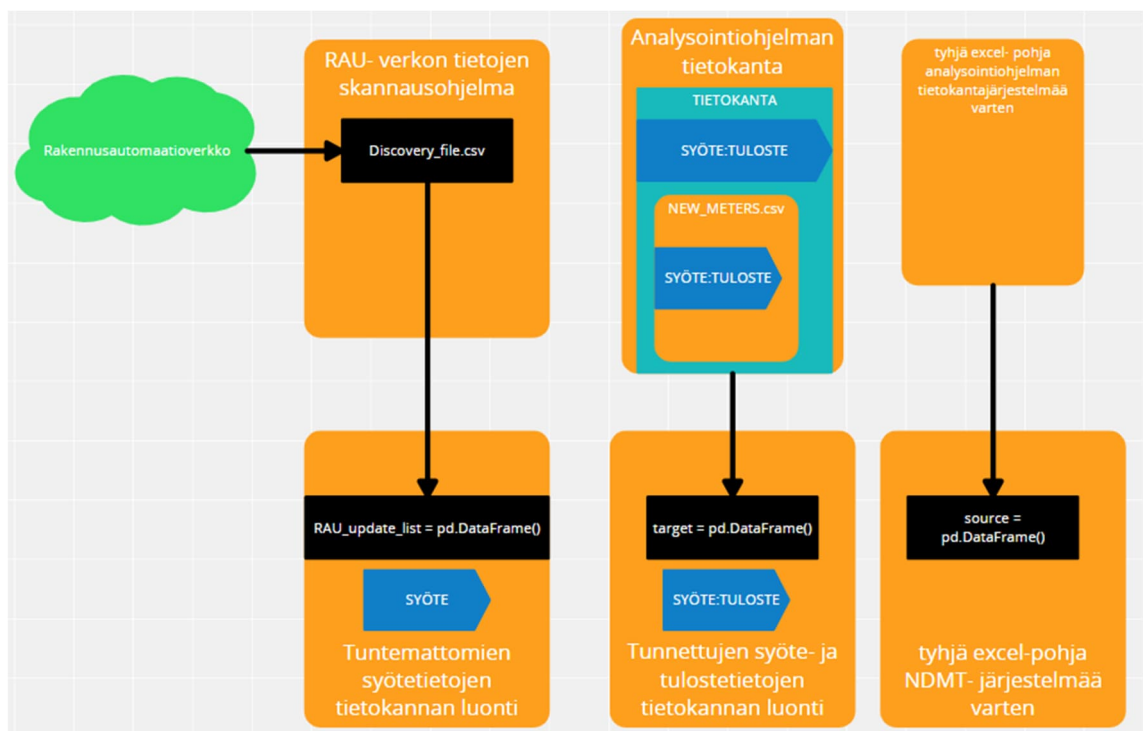
Ohjelman toimintahierarkia on tehty toimimaan seuraavasti:

- tietokannan luonti opetusaineistosta (analysointiohjelman tietokanta)
- tietokannan luonti tuotavista syötteistä (skannausohjelman skannaustiedosto)
- tuotavien syötteiden tietolähteiden suodatus poistamalla tuntemattomat tietolähteet ja luodaan niistä tiedosto Meters_no_groups.xlsx
- tunnettujen syötteiden tietolähteiden nimiä verrataan opetusaineiston syötteiden nimiin
- algoritmi laskee tuntemattomien syötteiden ja opetusaineiston syötteiden välisen samankaltaisuuden → ohjattu oppimismalli: lähimmän naapurin menetelmä
- liitetään parhaiden samankaltaisuustuloksien saaneiden syötteiden tulostetiedot opetusaineistosta tuntemattomille syötteille ja luodaan niistä tiedosto Meters_to_RAU.xlsx
- suodatetaan skannausohjelman skannaustiedosto tiedoston Meters_to_RAU.xlsx tietolähteiden perusteella ja tehdään tästä tiedosto RAU_Sourcefile.csv

- luodaan kansiorakenteet loogista päättelyä käyttäen, esim A = ilmanvaihtojärjestelmä, Ahu = ilmanvaihtokone ja tehdään tiedosto Nodetree.xlsx

4.1 Tietokantojen luonti

Ohjelma luo kolme erillistä tietokantaa (RAU_update_list, target ja source) alla olevan toimintakaavion mukaisesti tehden näistä tietokehyksiä, jotka on merkitty alla olevaan kuvaan tunnuksella pd.DataFrame().



Kuva 9. Tietokehyksien luonti tietokannoista

4.1.1 Tuntemattomien syötetietojen tietokannan luonti

Ohjelma saa tuntemattomat syötetiedot skannausohjelman skannaustiedostosta (Discovery_file.csv), jossa voi olla tietolähteiden tietoja

rakennusautomaatioverkon koosta riippuen tuhansista kymmeneen tuhansiin riveihin.

Tietokantojen luonnin koodi:

```
#luetaan tiedot source - kansioista: huom tallennus csv - utf 8 muotoon
Excelissä
print("\nCreating source-database from RAU-discovery file from source-
folder")
path2 = "C:/metersToRAU/source/"
files2 = [file for file in os.listdir(path2) if not
file.startswith('.')]
RAU_update_list = pd.DataFrame()
time_length2 = len(files2)
with tqdm(total=time_length2) as pbar:
    for file in files2:
        current_data2 = pd.read_csv(path2+"/"+file ,sep= ";" ,
encoding = "UTF-8", keep_default_na=False, dtype='unicode')
        RAU_update_list = pd.concat([RAU_update_list,current_data2])
        pbar.update(1)
```

Ohjelma luo tietokehyksen tuntemattomista syötetiedoista, jonka pituus tämän kohteen osalta on 1948 riviä.

Index	EquipID	System	EquipType	Point Name	Descriptor	Function	Unit Type	Engineering Units	Slope	Inter
0	Kotir^A501		BVAK1		VAK1 Kotiranta		LAO		1	0
1	Kotir^A501		BVAK1^A		Ilmastointi		LAO		1	0
2	Kotir^A501		BVAK1^A^TK01		TK01 Pakettikone huoneet		LAO		1	0
3	Kotir^A501		BVAK1^A^TK01^Cmd		TK01 Ohjaus		LENUM		1	0
4	Kotir^A501		BVAK1^A^TK01^Ind		TK01 Tilatieto		LAO		1	0
5	Kotir^A501		BVAK1^A^TK01^Ind^Aim		TK01 Ohjausristiriite		LDO		1	0
6	Kotir^A501		BVAK1^A^TK01^Ind^AimDlyLo		Häilytyksen päättämisekytk.viive		LAO		1	0
7	Kotir^A501		BVAK1^A^TK01^Ind^AimDlyLo.In		Tulo		LDO		1	0
8	Kotir^A501		BVAK1^A^TK01^Ind^AimDlyLo.Out		Lähtö		LDO		1	0
9	Kotir^A501		BVAK1^A^TK01^Ind^AimDlyLo.Reset		Palautus		LDO		1	0
10	Kotir^A501		BVAK1^A^TK01^Ind^AimDlyLo.Timod		Ajastinmodus		LENUM		1	0
11	Kotir^A501		BVAK1^A^TK01^Ind^Ind		TK01 Tilatieto		LENUM		1	0
12	Kotir^A501		BVAK1^A^TK01^MS		TK201TF01 Käskytkin		LENUM		1	0
13	Kotir^A501		BVAK1^A^TK01^Sched		TK01 Aikaohjelma		LAO		1	0

Kuva 10. Tuntemattomien syötteiden tietokannan tietokehys

4.1.2 Opetusaineiston tietokannan luonti

Ohjelma luo opetusaineiston syöte- ja tulostetiedot analysointiohjelman tietokannan olemassa olevista mittarilistoista sekä uusien tuntemattomien tietolähteiden tiedostosta NEW_METERS.csv.

Opetusaineiston tietokannan luonnin koodi:

```
#luodaan target kansion listojen tiedot dataframeen
path = "C:/metersToRAU/target/"
files = [file for file in os.listdir(path) if not
file.startswith('.')]
target = pd.DataFrame()

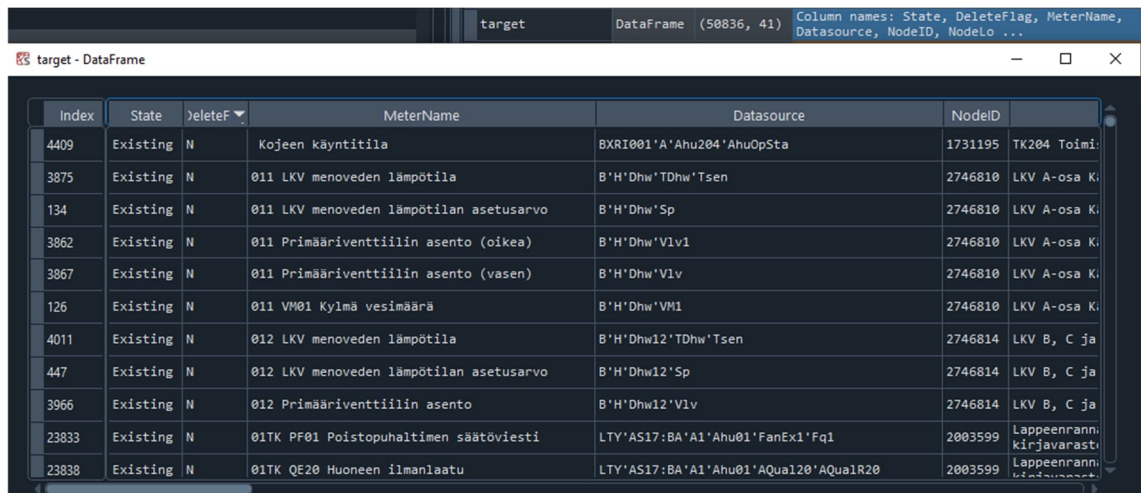
time_length1 = len(files)
print("Creating target-database from target- folder files")

with tqdm(total=time_length1) as pbar:

    for file in files:
        current_data = pd.read_csv(path+"/"+file ,sep= ";" , encoding
= "UTF-8", keep_default_na=False, dtype='unicode')
        target = pd.concat([target,current_data])

        pbar.update(1)
```

Ohjelma luo useiden eri kohteiden mittareiden tiedoista yhden tietokehyksen, jossa on 50839 kpl tietolähteen tiedot.



Index	State	DeleteF	MeterName	Datasource	NodeID	
4409	Existing	N	Kojeen käyttila	BXRI001'A'Ahu204'AhuOpSta	1731195	TK204 Toimi
3875	Existing	N	011 LKV menoveden lämpötila	B'H'Dhw'TDhw'Tsen	2746810	LKV A-osa K.
134	Existing	N	011 LKV menoveden lämpötilan asetusarvo	B'H'Dhw'Sp	2746810	LKV A-osa K.
3862	Existing	N	011 Primääriiventtiin asento (oikea)	B'H'Dhw'V1v1	2746810	LKV A-osa K.
3867	Existing	N	011 Primääriiventtiin asento (vasen)	B'H'Dhw'V1v	2746810	LKV A-osa K.
126	Existing	N	011 VM01 Kylmä vesimäärä	B'H'Dhw'VM1	2746810	LKV A-osa K.
4011	Existing	N	012 LKV menoveden lämpötila	B'H'Dhw12'TDhw'Tsen	2746814	LKV B, C ja
447	Existing	N	012 LKV menoveden lämpötilan asetusarvo	B'H'Dhw12'Sp	2746814	LKV B, C ja
3966	Existing	N	012 Primääriiventtiin asento	B'H'Dhw12'V1v	2746814	LKV B, C ja
23833	Existing	N	01TK PF01 Poistopuhaltimen säätöviesti	LTY'AS17:BA'A1'Ahu01'FanEx1'Fq1	2003599	Lappeenrann kirjavarasti
23838	Existing	N	01TK QE20 Huoneen ilmanlaatu	LTY'AS17:BA'A1'Ahu01'AQual20'AQualR20	2003599	Lappeenrann kirjavarasti

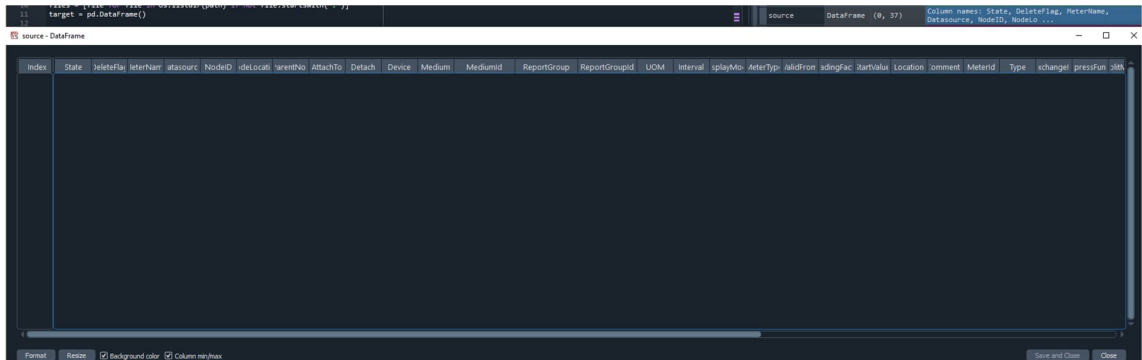
Kuva 11. Opetusaineiston tietokannan tietokehys

4.1.3 Tyhjä excel-pohja analysointiohjelmaa varten

Ohjelma avaa valmiin tyhjän source.csv-tiedoston ja luo siitä tyhjän tietokehyksen, johon lisätään tiedot tunnetuille tietolähteille. Tietokehyksessä on valmiina oikean nimiset sarakkeet analysointiohjelmaa varten.

Tyhjän tietokehysten luonnin koodi:

```
#luodaan tyhjä source dataframe  
source = pd.read_csv("C:/metersToRAU/files/source.csv", sep= ";" ,  
encoding = "UTF-8", dtype='unicode')
```



Kuva 12. Tyhjä tietokehys tietojen lisäystä analysointiohjelmalla varten

4.2 Tietolähteiden nimien käsittely

Ohjelma käsittelee tuntemattomien ja tunnettujen syötteiden nimet ja poistaa niistä tarpeettomat tiedot. Tiedoista poistetaan mm. kohdetiedot sekä numerot, jotka ovat epäolennaisia tuntemattomien syötetietojen suodatuksessa, esimerkiksi 'kohde1'A'Ahu1'Fansu2'Cmd muokataan muotoon A'Ahu'Fansu'Cmd.

Tarkoituksena on suodattaa tuntemattomista syötetiedoista pois tietolähteet, joita ei tunneta. Tässä kohtaa voidaan poistaa tietolähteiden lopusta myös numerotunnukset mm. lämpötila-antureista, koska halutaan suodattaa pois tietyt päätteet pois, joita ei tunneta. Tässä kohtaa ohjelma ei käsittele esimerkiksi mihin raporttiryhmiin tuntemattomat syötteet liitetään.

Point Name	Descriptor	Point Name
BVAK1'A'TK01'Cmd	TK01 Ohjaus	A'TK'Cmd
BVAK1'A'TK01'Ind	TK01 Tilatieto	A'TK'Ind
BVAK1'A'TK01'Ind'Alm	TK01 Ohjausristiriita	A'TK'Ind'Alm
BVAK1'A'TK01'Ind'AlmDlyLo	HÄÄlytyksen päättämällekytk.vii	A'TK'Ind'AlmDlyLo
BVAK1'A'TK01'Ind'AlmDlyLo.In	Reset	A'TK'Ind'AlmDlyLo.In
BVAK1'A'TK01'Ind'AlmDlyLo.Out	LÄÄhtö	A'TK'Ind'AlmDlyLo.Out
BVAK1'A'TK01'Ind'AlmDlyLo.Reset	Palautus	A'TK'Ind'AlmDlyLo.Reset
BVAK1'A'TK01'Ind'AlmDlyLo.TiMod	Ajastinmodus	A'TK'Ind'AlmDlyLo.TiMod
BVAK1'A'TK01'Ind'Ind	TK01 Tilatieto	A'TK'Ind'Ind
BVAK1'A'TK01'MS	TK201TF01 Käänsikytkin	A'TK'MS
BVAK1'A'TK01'Sched	TK01 Aikaohjelma	A'TK'Sched
BVAK1'A'TK01'T10	TK01TE10 Tuloilman lt.	A'TK'T
BVAK1'A'TK01'T10'AlmFnct	TK01TE10 HäÄlytysasetukset	A'TK'T'AlmFnct

Kuva 13. Tietolähteiden nimien käsittely

Rakennusautomaatiopisteiden tietolähteiden nimet erotellaan käyttämällä erikoismerkkejä kuten ' tai , tai /. Tästä johtuen tuntemattomien syötetietojen ja opetusaineiston erikoismerkit tuli ottaa huomioon, jotta tietoja voitiin suodattaa tietokehyksistä niiden erikoismerkkien perusteella. Yleensä tietokehyksien komennoissa käytetään vakioituja erikoismerkkejä kuten alla olevassa koodissa käytetään tyhjää välilyöntiä koodin kohdassa `str.split("käytettävä merkki")`.

```
meternames_descriptor_expand = meternames.Descriptor.str.split("
",expand=True)
```

Tässä luodaan uusi tietokehys siten, että jaetaan tekstikentät `meternames-` tietokehyksen `Descriptor`-sarakeesta siten, että erotellaan merkkijonot välien (" ") kohdalta uuden datakehysen sarakkeisiin 0,1,2,3, jne.

Ongelmaksi muodostui se, että rakennusautomaatiopisteiden nimeämisessä on saatettu käyttää ' merkkejä ensimmäiset 100 riviä ja seuraavat 400 riviä käyttäen _ merkkiä. Tästä johtuen tuli rakentaa algoritmit, jotka osaavat ottaa myös huomioon jokaisella tuntemattomalla syötteen rivillä esiintyvän erikoismerkin.

Eri variaatiot järjestelmistä sisältäen erilaiset erikoismerkit tehtiin koodilla:

```
systems = ["A", "C", "E", "F", "H", "M", "LL", "Pool", "R", "RFnct", "U", "Uas"]
separators_list = []
separators_list_target = []
separators_list_source = []

for i in target['Datasource']:
    if i:
        separators_list.append(''.join(filter(lambda x: not
x.isalnum(), i)))
        separators_list_target.append(''.join(filter(lambda x: not
x.isalnum(), i)))

for i in RAU_update_list['Point Name']:
    if i:
        separators_list.append(''.join(filter(lambda x: not
x.isalnum(), i)))
        separators_list_source.append(''.join(filter(lambda x: not
x.isalnum(), i)))

separators = pd.DataFrame({'separator': separators_list})
separators['separator'] = separators['separator'].apply(lambda x: '
'.join(set(x)))

separators_target = pd.DataFrame({'separator':
separators_list_target})

separators_source = pd.DataFrame({'separator':
separators_list_source})

separators = separators.drop_duplicates()
separators_split = separators['separator'].str.split(expand=True)

separators_list = []

for index, row in separators_split.iterrows():
    for column in separators_split.columns:
        separators_list.append(f"{row[column]}")

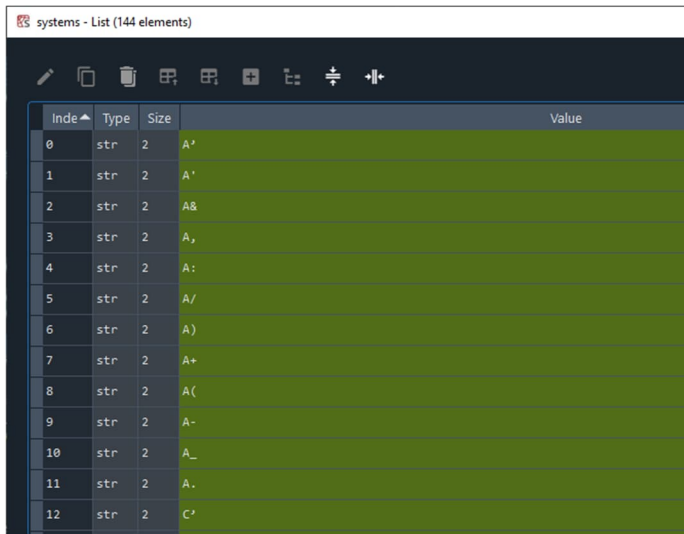
separators_list = [item for item in separators_list if not
any(c.isalpha() for c in item)]
separators_list = list(set(separators_list))

combined_values = []

for system in systems:
    combined_values.append(system)
    for separator in separators_list:
        combined_values.append(system + separator)

combined_values = [value for value in combined_values if value not in
systems]
systems = combined_values
```

Tässä luotiin siis järjestelmille systems erilaiset erikoismerkkitaupukset, joita tässä kohtaa tuli 144 kappaletta.



Index	Type	Size	Value
0	str	2	A+
1	str	2	A'
2	str	2	A&
3	str	2	A,
4	str	2	A:
5	str	2	A/
6	str	2	A)
7	str	2	A+
8	str	2	A(
9	str	2	A-
10	str	2	A_
11	str	2	A.
12	str	2	C'

Kuva 14. Systems (järjestelmien) erikoismerkkitaupukset

Opetusaineiston tietolähteiden nimet muokataan kuten tuntemattomat tietolähteet ja sen tietokehyksestä tehdään tuple – lista suodatusalgoritmia varten koodilla:

```

filteredEndings = pd.DataFrame()
filteredEndings["Datasource"] = savetargetdatasource["Datasource"]
filteredEndings['Datasource'] =
filteredEndings['Datasource'].str.split(':').str[-1]

filteredEndings['Datasource'] =
filteredEndings['Datasource'].apply(lambda x: ''.join(c for c in x if
c.isalpha() or c in separators_list))

escaped_systems = [re.escape(s) for s in systems]
pattern = '|'.join(escaped_systems)

filteredEndings =
filteredEndings[filteredEndings['Datasource'].str.contains(pattern,
na=False)]

filteredEndings =
filteredEndings.drop_duplicates(subset=["Datasource"])

pattern = '|'.join(map(re.escape, systems))

filteredEndings =
filteredEndings[filteredEndings['Datasource'].str.contains(pattern,
case=False, na=False)]
filteredEndings['Datasource'] =
filteredEndings['Datasource'].str.replace(f'.*({pattern})', r'\1',
regex=True)
filteredEndings =
filteredEndings.drop_duplicates(subset=["Datasource"])

filteredEndings["Datasource"] =
filteredEndings["Datasource"].str.lower()

filteredEndings_toLIST = filteredEndings["Datasource"].tolist()
endings = tuple(filteredEndings_toLIST)

```

The image displays three sequential screenshots from a Jupyter Notebook, illustrating the transformation of a DataFrame into a tuple of strings.

First Screenshot: A DataFrame with 28 rows and 2 columns. The 'Index' column contains integers from 0 to 27, and the 'Datasource' column contains alphanumeric strings representing file paths or identifiers.

Index	Datasource
0	Paino\801AS01\001\fw\SPK1\CueEgM
1	Paino\801AS01\001\fw\VM04
3	Paino\801AS01\001\fw\SHK1\CueEgM
4	Paino\801AS06\001\A1\TK09\Msen202
5	Paino\801AS01\001\fw\SHK2\CueEgM
6	Paino\801AS06\001\A1\TK09\Msen203
7	Paino\801AS01\001\fw\SHK3\CueEgM
8	Paino\801AS06\001\A1\TK09\Msen200
9	Paino\801AS06\001\A1\TK10\Msen202
10	Paino\801AS06\001\A1\TK10\Msen200
11	Paino\801AS06\001\A1\TK10\Msen203
13	Paino\801AS07\001\P101\P102\SH1\CueEgM
14	Paino\801AS07\001\P101\P102\SH2\Vka
15	Paino\801AS07\001\P101\P101\SH1\CueEgM
16	Paino\801AS07\001\P101\P102\SH1\CueEgM
17	Paino\801AS07\001\P101\P102\SH1\Vka
18	Paino\801AS07\001\P101\P1001\SH1\CueEgM
19	Paino\801AS07\001\P101\P101\SH2\Vka
20	Paino\801AS07\001\P101\P1001\SH1\Vka
21	Paino\801AS01\001\Euv\UV1013\ManSul
22	Paino\801AS01\001\Euv\UV1013\ManSul
23	Paino\801AS01\001\HI\K101\Q001
24	Paino\801AS01\001\Euv\UV1014\ManSul
25	Paino\801AS01\001\Euv\UV1016\ManSul
26	Paino\801AS01\001\Euv\UV1012\ManSul
27	Paino\801AS01\001\Euv\UV1015\ManSul
28	Paino\801AS01\001\Euv\UV1011\ManSul

Second Screenshot: A DataFrame with 114 rows and 2 columns. The 'Index' column contains integers from 1 to 114, and the 'Datasource' column contains the alphanumeric strings from the first screenshot, now filtered and cleaned.

Index	Datasource
1	h\vm\vm
4	a\k\msen
8	a\k\msen
23	h\k1\oe
30	h\k1\fg
34	a\k\mansul
44	a\k\ff\bo
48	a\k\ff\fg
49	a\k\pf\fg
61	a\k\bo
67	a\k\pf\cl
79	h\pv\sp\sp
80	h\l\pv\sp
81	h\l\sp\sp
83	a\k\sp
86	h\pv\tv
87	a\k\sp
89	h\l\tv
91	h\l\tv
96	h\pv\te\al
98	h\pv\te\al
100	h\l\te\al
101	h\k1\te
104	a\k\te
105	h\l\te\al
108	h\l\pv\al
114	a\k\pv\pt

Third Screenshot: A tuple containing 114 elements, each being a string of alphanumeric characters, representing the final output of the data processing pipeline.

Index	Type	Size	Value
0	str	12	a\sh\ppc
1	str	9	a\sh\tr\tsen
1236	str	12	a\seop\alfurenit\al
1227	str	14	a\seop\alfurenit\vaista\prval
3282	str	10	a\seop\k\k\nc\cct1
3287	str	11	a\seop\k\k\nc\k\ansod
3285	str	10	a\seop\k\k\nc\k\prsp
3286	str	9	a\seop\k\k\nc\tr
3229	str	10	a\seop\k\k\run\cct1
3226	str	14	a\seop\k\k\run\ffsp
3228	str	7	a\seop\k\k\run\k\ansod
3227	str	14	a\seop\k\k\run\k
3245	str	7	a\seop\k\k\run\cct1
3249	str	7	a\seop\k\k\run\k\ansod
3252	str	10	a\seop\k\k\run\prsp
3251	str	10	a\seop\k\k\run\tr
1229	str	10	a\seop\tr\prval
2017	str	14	a\se\fa\sq\al
2022	str	8	a\se\dep\dephofec\cmd\bo
2023	str	14	a\se\fa\sh\cmd\bo
3237	str	7	a\se\fa\se\efb
1835	str	7	a\se\fa\se\cmd\bo
2019	str	11	a\se\fa\se\cmd\prval
2018	str	10	a\se\fa\se\sttupool\al
735	str	10	a\se\fa\se\cfval
1824	str	12	a\se\reban\mansul
2021	str	14	a\se\reopod

Kuva 15. Opetusaineiston tietolähteiden nimien muokkaus endings(tuple) varten

4.3 Tarpeettomien tuntemattomien syötetietojen suodatus

Tutkittavasta tietokannasta suodatetaan turhat rakennusautomaation mittareiden tietolähteet pois tietokannassa olemassa olevien päätteiden perusteella käyttämällä loogista päättelyä esimerkiksi jos A'fansu'cmd = A'fansu'cmd niin silloin lisätään tulos on true (1) ja muuten tulos on false (0). Esimerkiksi A'fansu'cmd on olemassa sekä tuntemattomassa, että tunnetuissa syötetiedoissa niin silloin syötetieto tallennetaan listan filterLIST-tietokantaan arvolla 1. Ohjelma luo binäärisen [1,0] taulukon ja lisää tuotavien pisteiden viimeiseen sarakkeeseen arvon 1 jos tuntematon syöte löytyy tunnetusta syötetietokannasta ja arvon 0 jos sitä ei ole olemassa.

```
# filtteröidään mittarilista olemassa olevien mittareiden päätteiden
perusteella

filterLIST = []

print("Comparing source-database point-name similarities to target-
database point-names")
time_length4 = len(filterLIST)

with tqdm(total=time_length4) as pbar:
    for i in RAU_update_list['Point Name']:
        # print(i.lower())

        if i.lower().endswith(endings):
            filterLIST.append(1)
        else:
            filterLIST.append(0)
        pbar.update(1)

found_knowns = []
found_unkowns = []

for i in filterLIST:
    if i == 1:
        found_knowns.append(i)
    else:
        found_unkowns.append(i)

print("\nSource- database lookup report: ",'\nFound known point
names:',len(found_knowns) , ' / ', len(filterLIST),' ('
,round((len(found_knowns) / len(filterLIST)*100),2) , ' % ' ,')'
```

```
, '\nNot found:', len(found_unkowns), ' / ', len(filterLIST), ' ('  
, round((len(found_unkowns) / len(filterLIST)*100),2) , ' % ' , ')')  
RAU_update_list['found'] = filterLIST
```

Ohjelma tulostaa tiedon paljonko tuntemattomasta tietokannasta tiedetään olemassa olevia tietolähteitä. Tässä kohtaa tuntemattomista syötteistä 1883 kpl tietolähteitä tunnettiin 106 kpl eli 5.63 % kokonaismäärästä.

```
Comparing source-database point-name similarities to target-database point-names  
1883it [00:00, 40073.44it/s]  
Source- database lookup report:  
Found known point names: 106 / 1883 ( 5.63 % )  
Not found: 1777 / 1883 ( 94.37 % )  
  
Creating Meters_no_groups - excel list  
  
Meters_no_groups - excel list ready
```

Kuva 16. Ohjelman MetersToRAU.py visuaalinen toiminta

4.4 Meters_no_groups.xlsx-tiedoston luonti

Ohjelma luo Meters_no_groups.xlsx-tiedoston luomalla notfound-tietokehyksen ja lisäämälle siihen tietolähteet, jotka saivat listalle filterLIST arvon 0.

```
print("\nCreating Meters_no_groups - excel list")  
#tässä kohtaa tulostaa erillisen listan mittareista joita ei löytynyt  
eli --> found = 0  
  
notfound = pd.DataFrame()  
notfound = RAU_update_list.copy(deep=True)  
notfound['Point Name'] = savesourcedatasource['Datasource']  
notfound.drop(notfound[notfound.found != 0].index, inplace = True)  
notfound = notfound[notfound.columns.drop('found')]
```

4.5 Meters_to_RAU.xlsx-tiedoston luonti

Ohjelma jatkaa tunnettujen syötteiden tulostetietojen määrittelyä samankaltaisuustarkastuksessa. Koneoppimisalgoritmi perustuu lähimmän naapurin luokittimeen, jossa mitataan tuntemattoman syötteen etäisyyttä opetusaineiston kaikkiin syötteisiin. Näiden välinen paras samankaltaisuus,

lyhin etäisyys valitaan ja liitetään tuntemattomalle syötteelle samat tulostetiedot (raporttiryhmä, uom, jne) kuin opetusaineiston syötteellä.

Samankaltaisuustarkastuksesta tehdään similarity-tietokehys, johon lisätään tunnettujen syötteiden tietolähteiden nimet, opetusaineiston tietolähteiden nimet ja näiden väliset samankaltaisuuslaskentojen arvot.

```
similarity = pd.DataFrame()  
similarity['Source'] = sourcelist  
similarity['Datasource'] = targetlist  
similarity['ratio'] = ratio
```

Alla olevassa kuvassa on similarity-tietokehys, jossa on tunnettujen syötteiden (Source) ja opetusaineiston (Datasource) tietolähteiden välisten samankaltaisuuksien laskentojen arvot (ratio).

similarity - DataFrame

Index	Source	Datasource	ratio
159954	BVAK1'A'TK01'T10'T	Bs07'A'Ahu11'TSu'TSen	0.461538
160060	BVAK1'A'TK01'T10'T	Bs05'A'Ahu05'TSu'TSen	0.461538
160166	BVAK1'A'TK01'T10'T	Bs10'A'Ahu43'TSu'TSen	0.25641
160272	BVAK1'A'TK01'T10'T	Bs09'A'Ahu10'TSu'TSen	0.410256
160378	BVAK1'A'TK01'T10'T	Bs09'A'Ahu52'TSu'TSen	0.410256
160484	BVAK1'A'TK01'T10'T	Bs09'A'Ahu51'TSu'TSen	0.461538
160590	BVAK1'A'TK01'T10'T	Bs15'A'Ahu04'TSu'TSen	0.512821
160696	BVAK1'A'TK01'T10'T	Bs16'A'Ahu401'TSu'TSen	0.55
160802	BVAK1'A'TK01'T10'T	Bs20'A'Ahu21'TSu'TSen	0.461538
160908	BVAK1'A'TK01'T10'T	Bs20'A'Ahu22'TSu'TSen	0.410256
161014	BVAK1'A'TK01'T10'T	Bs24'A'Ahu34'TSu'TSen	0.410256
161120	BVAK1'A'TK01'T10'T	Bs21'A'Ahu39'TSu'TSen	0.461538
161226	BVAK1'A'TK01'T10'T	Bs17'A'Ahu404'TSu'TSen	0.5
161332	BVAK1'A'TK01'T10'T	Bs25'A'Ahu17'TSu'TSen	0.461538
161438	BVAK1'A'TK01'T10'T	Bs24'A'Ahu35'TSu'TSen	0.410256
161544	BVAK1'A'TK01'T10'T	Bs17'A'Ahu403'TSu'TSen	0.5
161650	BVAK1'A'TK01'T10'T	Bs25'A'Ahu18'TSu'TSen	0.461538

Kuva 17. Similarity-tietokehyksen samankaltaisuuksien laskenta-arvot

Ohjelma jättää parhaimmat vastaavuusarvot tietolähteiden sarakkeiden Source ja Datasource välille ja similarity-tietokehykseen liitetään opetusaineiston tulostetiedot target- tietokehyksestä koodilla:

```
#filteröidään similarity dataframesta löydetyt nimien vastaavuudet
siten että jätetään vain maksimi-arvot listaan
similarity =
similarity.iloc[similarity.groupby('Source')['ratio'].agg(pd.Series.idxmax)]
similarity = similarity.merge(target, on='Datasource', how='left')
similarity = similarity.drop_duplicates(subset=["Source"],
keep="first")
```

Alla kuvassa similarity-tietokehys parhaimpien arvojen suodatuksen jälkeen Source ja Datasource välillä:

Index	Source	Datasource	ratio
588406	BVAK1'A'TK01'T10'T	B1601'A'TK01'T10'T	0.833333
587665	BVAK1'A'TK01'T30'T	B1601'A'TK01'T30'T	0.833333
604096	BVAK1'A'TK02'T10'T	B1303'A'TK302'T10'T	0.810811
604415	BVAK1'A'TK02'T30'T	B1303'A'TK302'T30'T	0.810811
587350	BVAK1'A'TK201'CasCtr'SpMaxSu	B1601'A'TK01'CasCtr'SpTR	0.730769
587351	BVAK1'A'TK201'CasCtr'SpMaxSu1	B1601'A'TK01'CasCtr'SpTR	0.716981
4411090	BVAK1'A'TK201'CasCtr'TCasCtr.SpLoR	BXOL126'A'TK310'CasCtr'TCasCtr.SpLoR	0.828571
603359	BVAK1'A'TK201'CasCtr'TCasCtr.XctrSu	B1601'A'TK01'CasCtr'TCasCtr.XctrSu	0.898551
586400	BVAK1'A'TK201'DmpEh'Dmp	B1601'A'TK01'DmpEh'Dmp	0.844444
586825	BVAK1'A'TK201'DmpSu'Dmp	B1601'A'TK01'DmpSu'Dmp	0.844444
601878	BVAK1'A'TK201'Erc1P'IcPrt'P	B1302'A'TK301'Erc1P'IcPrt'P	0.851852
600501	BVAK1'A'TK201'Erc1P'LTOc	B1302'A'TK301'Erc1P'LTOc	0.833333
601456	BVAK1'A'TK201'Erc1P'MtrEg'Pwr	B1302'A'TK301'Erc1P'MtrEg'Pwr	0.862069
587041	BVAK1'A'TK201'FanEh'Ind'Ind	B1601'A'TK01'FanEh'Ind'Ind	0.867925
873348	BVAK1'A'TK201'FanSu'Cmd11	B2'VAK01'A'1TK1'FanSu'Cmd1t	0.769231
873349	BVAK1'A'TK201'FanSu'Cmd12	B2'VAK01'A'1TK1'FanSu'Cmd1t	0.769231
586514	BVAK1'A'TK201'FanSu'Ind'Ind	B1601'A'TK01'FanSu'Ind'Ind	0.867925
589271	BVAK1'A'TK201'FilEh'DP	B1601'A'TK01'FilEh'DP	0.837209
589060	BVAK1'A'TK201'FilSu'DP	B1601'A'TK01'FilSu'DP	0.837209
588637	BVAK1'A'TK201'Hcl'TE5'T	B1601'A'TK01'Hcl'TE45'T	0.826087
586624	BVAK1'A'TK201'Hcl'V1v	B1601'A'TK01'Hcl'V1v	0.829268
876429	BVAK1'A'TK201'NgthH'Sp	B2'VAK01'A'1TK1'TSu'Sp	0.681818
587792	BVAK1'A'TK201'T1'T	B1601'A'TK01'T31'T	0.777778
587899	BVAK1'A'TK201'T2'T	B1601'A'TK01'T02'T	0.777778
587688	BVAK1'A'TK201'T3'T	B1601'A'TK01'T30'T	0.777778
587689	BVAK1'A'TK201'T6'T	B1601'A'TK01'T30'T	0.722222

Kuva 18. Similarity-tietokehyksen samankaltaisuuksien parhaat laskenta-arvot

Ratio-arvot ovat välillä [0 -1] eli kuvan mukaiset tietolähteet ovat hyvin samankaltaisia opetusaineiston syötetietoihin verrattuna.

Tunnetuista tietolähteistä tehdään Meters_to_RAU.xlsx-tiedosto.

BVAK1'A'TK203'T1'T				FDD Lämpötila	621	Jäteilman lämpötila LTO:n jälkeen
BVAK1'A'TK203'T2'T				FDD Lämpötila	621	Huonelämpötila
BVAK1'A'TK203'T3'T				FDD Lämpötila	621	Jäteilman lämpötila LTO:n jälkeen
BVAK1'A'TK203'T6'T				FDD Lämpötila	621	Jäteilman lämpötila LTO:n jälkeen

Kuva 19. Kuvakaappaus Meters_to_RAU.xlsx-tiedostosta

Listan tiedoista havaittiin, että ainakin ilmanvaihtokoneiden lämpötilojen tunnukset ovat väärin, joten ne tuli muokata opetusaineiston tiedostoon NEW_meters.csv. Similarity-tietokehyksen kuvasta voidaan havaita, että ohjelma löytää lämpötilatunnuksille (4 alinta riviä) hyvin lähellä olevat samankaltaisuudet 0,72-0,77, mutta näiden osalta samankaltaisuuden tulisi olla lähes 1, jotta tiedot ovat täysin oikein.

4.6 RAU-sourcefile.csv-tiedoston luonti

Tiedosto luodaan muokkaamalla skannausohjelman skannatustiedostoa suodattamalla siitä jäljelle vain tietolähteet, jotka löytyvät Meters_to_RAU.xlsx-tiedostosta. Alla koodi:

```
print("Creating RAU_Sourcefile.csv -list")
#luodaan source kansion RAU discovery dataframe
path3 = "C:/metersToRAU/source"
files3 = [file for file in os.listdir(path3) if not
file.startswith('.')]
RAU_source_file = pd.DataFrame()

time_length5 = len(files3)

with tqdm(total=time_length5) as pbar:
    for file in files3:
        current_data = pd.read_csv(path3+"/"+file , sep= ", " ,
encoding = "UTF-8", keep_default_na=False, dtype='unicode')
        RAU_source_file = pd.concat([RAU_source_file,current_data])
        pbar.update(1)

RAU_source_file = RAU_source_file[RAU_source_file['Point
Name'].isin(source['Datasource'])]

RAU_source_file.to_csv('RAU_Sourcefile.csv', encoding="utf-8", index =
False)
print("RAU_Sourcefile.csv -list ready")
```

Tuloksena saadaan tiedosto RAU_Sourcefile.csv, joka ladataan skannausohjelmaan, joka lähettää valittujen tietolähteiden tietoja rakennusautomaatioverkosta pyydetyllä aikasyklillä analysointiohjelmaan.

4.7 Nodetree.xlsx-tiedoston luonti

Analysointiohjelmaa varten tulee tehdä kansiorakenteet ja tietolähteet asettaa oikeisiin kansioihin. Kansioden luontia varten tietolähteiden edestä poistetaan turhat tiedot ja tehdään kansiorakenteet siten, että tietolähteiden teksikentät alkavat systems – tunnuksilla (systems = ["A","C","E","F","H","M","LL","Pool","R","RFnct","U","Uas"]) käyttäen koodia:

```
print("Creating node structures for Nodetree.xls - list")
nodetree = pd.DataFrame()
nodetree = source.copy(deep=True)

#poistetaan "kohde" ennen systems
nodetree['Datasource'] = nodetree['Datasource'].str.split(':').str[-1]
escaped_systems = [re.escape(s) for s in systems]
pattern = '|'.join(escaped_systems)
nodetree = nodetree[nodetree['Datasource'].str.contains(pattern,
na=False)]
pattern = '|'.join(map(re.escape, systems))
nodetree = nodetree[nodetree['Datasource'].str.contains(pattern,
case=False, na=False)]
nodetree['Datasource'] =
nodetree['Datasource'].str.replace(f'.*({pattern})', r'\1',
regex=True)
```

Nodetree-tietokehyksen Datasource-sarake levitetään auki huomioiden mitä erikoismerkkiä tietolähteiden nimissä on käytetty ja luodaan uusi tietokehys: nodetree_datasource_expand.

Index	Datasource	Nod
25	BVAK1 TK01 T10	A'TK01'T10'T
37	B BVAK1 TK01	A'TK01'T30'T
61	BVAK1 TK02 T10	A'TK02'T10'T
73	B BVAK1 TK02	A'TK02'T30'T
78	BVAK1 TK201	A'TK201'CasCtr'SpMaxSu
79	vo, yläraja	A'TK201'CasCtr'SpMaxSu1
98	4K1 TK201	A'TK201'CasCtr'TCasCtr.SpLoR
103	vo BVAK1 TK201	A'TK201'CasCtr'TCasCtr.XctrSu
124	C201F22 BVAK1	A'TK201'DmpEh'Dmp
126	C201F21 BVAK1	A'TK201'DmpSu'Dmp
146	li TK201PDE3	A'TK201'Erc1P'IcPrt'P
149	li TK201FZ3/FZ4	A'TK201'Erc1P'LTOc
158	201 Erc1P	A'TK201'Erc1P'MtrEg'Pwr
171	la TK201PF01	A'TK201'FanEh'Ind'Ind

Index	Kohde	odeType	0	1	2	3	4	5	jund_valk
25	Add building name	A	TK01	T10	T		nan	nan	0
37	Add building name	A	TK01	T30	T		nan	nan	0
61	Add building name	A	TK02	T10	T		nan	nan	0
73	Add building name	A	TK02	T30	T		nan	nan	0
78	Add building name	A	TK201	CasCtr	SpMaxSu		nan	nan	0
79	Add building name	A	TK201	CasCtr	SpMaxSu1		nan	nan	0
98	Add building name	A	TK201	CasCtr	TCasCtr.SpLoR		nan	nan	0
103	Add building name	A	TK201	CasCtr	TCasCtr.XctrSu		nan	nan	0
124	Add building name	A	TK201	DmpEh	Dmp		nan	nan	0
126	Add building name	A	TK201	DmpSu	Dmp		nan	nan	0
146	Add building name	A	TK201	Erc1P	IcPrt	P		nan	0
149	Add building name	A	TK201	Erc1P	LTOc			nan	0
158	Add building name	A	TK201	Erc1P	MtrEg	Pwr		nan	0
171	Add building name	A	TK201	FanEh	Ind			nan	0

Kuva 20. nodetree-tietokehyksen Datasource-kohtien merkkijonot laajennettuna huomioiden alimerkkijonoja erottavan erikoismerkin käytön

Kansiorakenteiden hierarkia analysointiohjelmassa on kuvattu sivulla 9 ja koodin def map_strings-algoritmi tekee tietolähteiden nimien perusteella kansiorakenteiden tunnukset koodilla:

```
def map_strings(s):
    if isinstance(s, str):
        if s.startswith("A"):
            return "Ilmanvaihtojärjestelmä"
        elif s.startswith("C"):
            return "Jäähdytysjärjestelmä"
        elif s.startswith("E"):
            return "Sähköjärjestelmä"
        elif s.startswith("F"):
            return "Palovalvonta"
        elif s.startswith("H"):
            return "Lämmitysjärjestelmä"
        elif s.startswith("M"):
            return "Mittarointi"
        elif s.startswith("LL"):
            return "Lämmitysjärjestelmä"
        elif s.startswith("Pool"):
            return "Lämmitysjärjestelmä"
        elif s.startswith("R"):
            return "Olosuhteiden hallinta"
        elif s.startswith("RFnct"):
            return "Huonesäädöt"
        elif s.startswith("S"):
            return "Vesi ja viemäri"
        elif s.startswith("U"):
            return "Erillispisteet"
        elif s.startswith("Uas"):
            return "Erillispisteet"
        else:
            return ""
    return s

for i in nodetree_datasource_expand.index:
    if pd.notna(nodetree_datasource_expand.at[i, 'found_value']) and
    isinstance(nodetree_datasource_expand.at[i, 'found_value'], int):
        found_value_col_index = int(nodetree_datasource_expand.at[i,
        'found_value'])
        cell_value = nodetree_datasource_expand.at[i,
        found_value_col_index]
        nodetree_datasource_expand.at[i, 'NodeType_1'] =
        map_strings(cell_value)
```

Tietokehyksen nodetree_datasource_expand sarake NodeType_1 saa järjestelmätunnuksen.

Index	Kohde	NodeType_1 ▲	0	1	2	3	4	
1765	Add building name	Erillispisteet	U	Alm	AlmFire	Alm	nan	nan
1766	Add building name	Erillispisteet	U	Alm	AlmFire	Alm1	nan	nan
1767	Add building name	Erillispisteet	U	Alm	AlmFire	Alm2	nan	nan
1768	Add building name	Erillispisteet	U	Alm	AlmFire	Alm4	nan	nan
1895	Add building name	Erillispisteet	U	Eg1	Vlm02	alm	Sp1	nan
1900	Add building name	Erillispisteet	U	Eg1	Vlm02	CI	nan	nan
1903	Add building name	Erillispisteet	U	Eg1	Vlm02	VlmOut	nan	nan
25	Add building name	Ilmanvaihtojärjestelmä	A	TK01	T10	T	nan	nan
37	Add building name	Ilmanvaihtojärjestelmä	A	TK01	T30	T	nan	nan
61	Add building name	Ilmanvaihtojärjestelmä	A	TK02	T10	T	nan	nan
73	Add building name	Ilmanvaihtojärjestelmä	A	TK02	T30	T	nan	nan
78	Add building name	Ilmanvaihtojärjestelmä	A	TK201	CasCtr	SpMaxSu	nan	nan
79	Add building name	Ilmanvaihtojärjestelmä	A	TK201	CasCtr	SpMaxSu1	nan	nan
98	Add building name	Ilmanvaihtojärjestelmä	A	TK201	CasCtr	TCasCtr.SpLoR	nan	nan

Kuva 21. Järjestelmätunnuksien luonti

Tämän jälkeen muutetaan laitteet kuten ahu,tk ja pk vastaamaan kansiorakenteen nimeä esim. Ahu = ilmanvaihtokone. Alla esimerkki loogisen päättelyn ehdoista koodissa:

```

# lämmitysryhmään liittyvät

mask_Lämmitysjärjestelmä =
((nodetree_datasource_expand[0].str.contains('H')) &

nodetree_datasource_expand['ReportGroup'].str.contains('Kaukolämpövede
n tulolämpötila'))
nodetree_datasource_expand.loc[mask_Lämmitysjärjestelmä, 'NodeType_2']
= 'Lämmöntuotto'

mask_lämmitysverkosto = ((nodetree_datasource_expand['NodeType_1'] ==
'Lämmitysjärjestelmä') &

(nodetree_datasource_expand['ReportGroup'].str.contains('Lämmityspiiri
n menoveden lämpötila')) |
(nodetree_datasource_expand[0] == 'Pool'))
nodetree_datasource_expand.loc[mask_lämmitysverkosto, 'NodeType_2'] =
'Lämmitysverkosto'

mask_lämmönsiirrin = ((nodetree_datasource_expand['NodeType_1'] ==
'Lämmitysjärjestelmä') &

nodetree_datasource_expand['ReportGroup'].str.contains('Primääriventti
ilin asento'))

#~nodetree_datasource_expand['ReportGroup'].isin(['Lämmitysverkoston
paine'])) # Primääriventtiilin asento &
nodetree_datasource_expand[22].str.contains('hgrp')
nodetree_datasource_expand.loc[mask_lämmönsiirrin, 'NodeType_2'] =
'Lämmönsiirrin'

mask_LKVLS = ((nodetree_datasource_expand['NodeType_1'] ==
'Lämmitysjärjestelmä') &
(nodetree_datasource_expand[22].str.contains('dhw|dw') |

nodetree_datasource_expand['ReportGroup'].str.contains('LKV'))
nodetree_datasource_expand.loc[mask_LKVLS, 'NodeType_2'] =
'Käyttöveden lämmönsiirrin'

```

Loogisen päättelyn perusteella luodaan nodetree_datasource_expand-tietokehyksen NodeType_2- sarakkeeseen kansioden nimet.

nodetree_datasource_expand - DataFrame

Index	Kohde	NodeType_1	0	1	2	3	4	5	suund_val	ReportGroup	22	NodeType_2
1391	Add building name	Lämmitysjärjestelmä	H	IV101	StupCt1	SPeff	nan	nan	0	Lämmityspiirin menoveden lämpötilan asetusarvo	iv	Lämmitysverkosto
1413	Add building name	Lämmitysjärjestelmä	H	IV101	T1	T	nan	nan	0	Lämmityspiirin menoveden lämpötila	iv	Lämmitysverkosto
1691	Add building name	Lämmitysjärjestelmä	H	LV103	Sp	nan	nan	nan	0	Lämmityspiirin menoveden lämpötilan asetusarvo	iv	Lämmitysverkosto
1703	Add building name	Lämmitysjärjestelmä	H	LV103	T1	T	nan	nan	0	Lämmityspiirin menoveden lämpötila	iv	Lämmitysverkosto
291	Add building name	Lämmitysjärjestelmä	H	Sp	nan	nan	nan	nan	0	Tuloilman lämpötilan asetusarvo	sp	
1347	Add building name	Lämmitysjärjestelmä	H	IV101	P1	P	nan	nan	0	Primääripumpun käyntinopeus	iv	
1355	Add building name	Lämmitysjärjestelmä	H	IV101	Pu	Ind	nan	nan	0	Primääripumpun käyntitila	iv	

Kuva 22. Kansioden nimien luonti

Kansioden nimissä tulee olla merkin : jälkeen rakennusautomaatiopisteen tietolähteen alkutunnus, jotta tietolähteet pystytään laittamaan oikeisiin kansioihin ohjelmalla ADDStandardNODEIDS.py laitekohtaisesti. Tämä muokkaus on tehty kuvan viimeiseen sarakkeeseen final_name.

d_valu	ReportGroup	22	NodeType_2	Name_Original	separator	me_stripp	final_name
	Tuloilman lämpötila	tk	Ilmanvaihtokone	A'TK01'T10'T	'	T	Ilmanvaihtokone TK01 :A'TK01
	Tuloilman lämpötila	tk	Ilmanvaihtokone	A'TK02'T10'T	'	T	Ilmanvaihtokone TK02 :A'TK02
	Tuloilmapuhaltimen käyntitila	tk	Ilmanvaihtokone	A'TK201'CasCtr'SpMaxSu	'	SpMaxSu	Ilmanvaihtokone TK201 :A'TK201
	Tuloilman lämpötilan asetusarvo	sp	Lämmitysjärjestelmä	H'Sp	'	Sp	
	Tuloilmapuhaltimen käyntitila	tk	Ilmanvaihtokone	A'TK202'CasCtr'SpMaxSu	'	SpMaxSu	Ilmanvaihtokone TK202 :A'TK202
	Tuloilmapuhaltimen käyntitila	tk	Ilmanvaihtokone	A'TK203'CasCtr'SpMaxSu	'	SpMaxSu	Ilmanvaihtokone TK203 :A'TK203
	Tuloilmapuhaltimen käyntitila	tk	Ilmanvaihtokone	A'TK207'CasCtr'SpMaxSu	'	SpMaxSu	Ilmanvaihtokone TK207 :A'TK207
	Lämmityspiirin menoveden lämpötilan asetusarvo	iv	Lämmitysverkosto	H'IV101'StupCtl'SPeff	'	SPeff	Lämmitysverkosto IV101 :H'IV101
	Lämmityspiirin menoveden lämpötilan asetusarvo	lv	Lämmitysverkosto	H'LV103'Sp	'	Sp	Lämmitysverkosto LV103 :H'LV103
	Kylmävesi (16)	lv	Osa	H'LV103'V1v	'	V1v	Vesimittaukset LV103 :H'LV103

Kuva 23. Sarakkeen final_name muokkaus

Sarakkeen final_name kentissä on tieto järjestelmästä, tunnus ja tietolähteen alkuosa kuten Ilmanvaihtokone TK01 : A'TK01. Tämän jälkeen kaikki tietolähteet joissa esiintyy A'TK01 kohdistetaan tähän kansioon ohjelmalla ADDStandardNODEIDS.py.

Kansioden tyyppien NodeType_2 täytyy noudattaa analysointiohjelman kansioden hierarkiaa kuten: lämmitysjärjestelmä, lämmöntuotto, lämmönsiirrin, käyttöveden lämmönsiirrin, lämmitysverkosto. Tämä johtuu siitä että kansioden ensimmäinen nimi on ryhmän hierarkian isäntä ja seuraavat niiden sisällä olevia alikansioita. Alla esimerkkikuva kansiorakenteiden hierarkiasta.

	A	B	C	D	E	F
1	State	Name	NodeType	NodeID	ParentID	
2	Existing	Kaupunki	Customer	3732738	9774	
3	Existing	Esimerkki kaupunki	Kohde	3734797	3732738	
4	Existing	Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	3735249	3734797	
5	New	Ilmanvaihtokone TK01 :B'A'TK01	Ilmanvaihtokone		3735249	
6	New	Ilmanvaihtokone TK02 :B'A'TK02	Ilmanvaihtokone		3735249	
7	Existing	Jäähdytysjärjestelmä	Jäähdytysjärjestelmä	3735250	3734797	
8	New	Vedenjäähdytin VJK01 :B'C'VJK01	Vedenjäähdytin		3735250	
9	New	Vedenjäähdytin CGrp2 :B'C'CGrp2	Vedenjäähdytin		3735250	
10	New	Vedenjäähdytin VJK03 :B'C'VJK03	Vedenjäähdytin		3735250	
11	Existing	Lämmitysjärjestelmä	Lämmitysjärjestelmä	3735251	3734797	
12	Existing	Lämmöntuotto DshFnct :B'H'DshFnct	Lämmöntuotto	3735252	3735251	
13	New	Lämmönsiirrin LS1 :B'H'LS1	Lämmönsiirrin		3735252	
14	New	Lämmönsiirrin LS2 :B'H'LS2	Lämmönsiirrin		3735252	
15	New	Lämmönsiirrin LS3 :B'H'LS3	Lämmönsiirrin		3735252	
16						
17						

Kuva 24. Esimerkki kansiorakenteiden hierarkiasta

Tästä johtuen kansiot järjestetään oikeaan järjestykseen koodilla:

```
def custom_sort(row):
    order = {
        'Ilmanvaihtojärjestelmä':
        ['Ilmanvaihtokone', 'Lämpöpumppu', 'Palvelualue', 'Tila', 'Ilmanmääräsäädin',
        'Kiertoilmakoje', 'Ohjauskeskus (laboratorio)', 'Vakioilmavirtasäädin', 'Muuttuvan ilmavirran säädin',
        'Poistopuhallinryhmä', 'Tuloilmapuhallin', 'Poistopuhallin'],
        'Jäähdytysjärjestelmä': ['Cooling Distribution',
        'Vedenjäähdytin', 'Lämpöpumppu', 'Lauhduttimet', 'Lauhdutin',
        'Lauhdevesipumput', 'Lämmönsiirrin', 'Sekundääripiiri',
        'Jäähdytysverkosto', 'Lämmitysverkosto', 'Jäähdytysverkoston
```

```

lämmönsiirrin', 'Sekoituspiiri', 'Tila', 'Jäähdytyslaite',
'Jäähdytyslaitteet'],
    'Lämmitysjärjestelmä': ['Lämmöntuotto', 'Lämmönsiirrin',
'Käyttöveden lämmönsiirrin', 'Lämmitysverkosto', 'Sekoituspiiri',
'Tila'],
    'Sähköjärjestelmä': ['Katkaisin', 'Tila'],
    'Rakennus/ yksikkö': ['Alamittaustaso-1', 'Osa', 'Tila'],
    'Olosuhteiden hallinta': ['Tila']
}

NodeType = row['NodeType']
NodeType_2 = row['NodeType_2']

if pd.isna(NodeType_2):
    return -1 # np.nan ensimmäiseksi

if NodeType in order and NodeType_2 in order[NodeType]:
    return order[NodeType].index(NodeType_2)

return float('inf') # jos tietoja ei löydy order listalta niin
nodetype_2 teksti laitetaan viimeiseksi

nodetree_new['sorting_key'] = nodetree_new.apply(custom_sort, axis=1)

```

Tässä luodaan sarake `sorting_key`, jonka avulla järjestetään sarakkeen `NodeType_2` nimet haluttuun järjestykseen ryhmäkohtaisesti `NodeType` – nimen perusteella.

nodetree_new - DataFrame

Index	State	Name	NodeType	NodeID	ParentID	NodeType_2	final_name	sorting_key
0	Existing	Add building name	Ilmanvaihtojärjestelmä	nan	nan	nan	nan	-1
1	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK01 :A'TK01	0
2	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK02 :A'TK02	0
3	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK201 :A'TK201	0
4	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK202 :A'TK202	0
5	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK203 :A'TK203	0
6	New	Add building name	Ilmanvaihtojärjestelmä	nan	nan	Ilmanvaihtokone	Ilmanvaihtokone TK207 :A'TK207	0
7	Existing	Add building name	Kohde	nan	nan	nan	nan	-1
8	Existing	Add building name	Lämmitysjärjestelmä	nan	nan	nan	nan	-1
9	New	Add building name	Lämmitysjärjestelmä	nan	nan	Lämmitysverkosto	Lämmitysverkosto IV101 :H'IV101	3
10	New	Add building name	Lämmitysjärjestelmä	nan	nan	Lämmitysverkosto	Lämmitysverkosto LV103 :H'LV103	3
11	New	Add building name	Lämmitysjärjestelmä	nan	nan	Lämmitysjärjestelmä		inf
12	Existing	Add building name	Rakennus/ yksikkö	nan	nan	nan	nan	-1
13	New	Add building name	Rakennus/ yksikkö	nan	nan	Osa	Vesimittaukset LV103 :H'LV103	1

Kuva 25. Sarakkeen `sorting_key` luonti `nodetree_new` tietokehykseen

Tarkistus- ja muokkausfunktioiden jälkeen tietokehyks `nodetree_new` järjestetään analysointiohjelman vaatimaan muotoon ja lopulta tietokehyks saa muodon:

nodetree_new - DataFrame

Index	State	Name	NodeType	NodeID	ParentID
0	Existing	Remember to add from Navigator	Customer	Remember to add from Navigator	Remember to add from Navigator
1	Existing	Add building name	Kohde	Remember to add from Navigator	Remember to add from Navigator
2	Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	Remember to add from Navigator	Remember to add from Navigator
3	New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone	nan	nan
4	New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone	nan	nan
5	New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone	nan	nan
6	New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone	nan	nan
7	New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone	nan	nan
8	New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone	nan	nan
9	Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	Remember to add from Navigator	Remember to add from Navigator
10	Existing	Lämmöntuotto	Lämmöntuotto	Remember to add from Navigator	Remember to add from Navigator
11	New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto	nan	nan
12	Existing	Add building name	Rakennus/ yksikkö	Remember to add from Navigator	Remember to add from Navigator
13	New	Vesimittaukset LV103 :H'LV103	Osa	nan	nan

Kuva 26. Kansiorakenteet valmiiksi muotoiltuna

Tämän jälkeen se tulostetaan tiedostoksi Nodetree.xlsx.

	A	B	C	D	E
1	State	Name	NodeType	NodeID	ParentID
2	Existing	Remember to add from Navigator	Customer	Remember to add from Navigator	Remember to add from Navigator
3	Existing	Add building name	Kohde	Remember to add from Navigator	Remember to add from Navigator
4	Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	Remember to add from Navigator	Remember to add from Navigator
5	New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone		
6	New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone		
7	New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone		
8	New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone		
9	New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone		
10	New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone		
11	Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	Remember to add from Navigator	Remember to add from Navigator
12	Existing	Lämmöntuotto	Lämmöntuotto	Remember to add from Navigator	Remember to add from Navigator
13	New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto		
14	Existing	Add building name	Rakennus/ yksikkö	Remember to add from Navigator	Remember to add from Navigator
15	New	Vesimittaukset LV103 :H'LV103	Osa		
16					
17					

Kuva 27. Nodetree_new-tietokehys tulostettuna muotoon Nodetree.xlsx

Käyttäjää muistutetaan lisäämään tunnukset sarakkeisiin D ja E sekä rakennuksen nimi sarakkeeseen B.

Tämän jälkeen analysointiohjelman web-käyttöliittymällä luodaan tiedostossa Nodetree.xlsx olevat sarakkeen State-arvot Existing käsin ja kopioidaan tunnukset tiedostoon Nodetree.xlsx. Tämän jälkeen tiedosto Nodetree.xlsx

ladataan analysointiohjelmaan tietokannan hallintajärjestelmän kautta xlsx-tiedostona.

State	Name	NodeType	NodeID	ParentID
Existing	Remember to add from Navigator	Customer	1	2
Existing	Add building name	Kohde	3	4
Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	5	3
New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone		5
New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone		5
New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone		5
New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone		5
New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone		5
New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone		5
Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	6	3
Existing	Lämmöntuotto	Lämmöntuotto	7	3
New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto		7
Existing	Add building name	Rakennus/ yksikkö	8	3
New	Vesimittaukset LV103 :H'LV103	Osa		8

Kuva 28. Kansiorakenteiden hierarkia kansiotunnuksineen ennen analysointiohjelmaan latausta

Analysointiohjelmaan lataamisen jälkeen kansio ladataan uudestaan ja alikansioille on tehty omat yksilöidyt kansiotunnukset.

State	Name	NodeType	NodeID	ParentID
Existing	Remember to add from Navigator	Customer	1	2
Existing	Add building name	Kohde	3	4
Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	5	3
New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone	51	5
New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone	52	5
New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone	53	5
New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone	54	5
New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone	55	5
New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone	56	5
Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	6	3
Existing	Lämmöntuotto	Lämmöntuotto	7	3
New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto	71	7
Existing	Add building name	Rakennus/ yksikkö	8	3
New	Vesimittaukset LV103 :H'LV103	Osa	81	8

Kuva 29. Kansiorakenteiden hierarkia tunnuksineen analysointijärjestelmästä lataamisen jälkeen

Tämän jälkeen tarvittavat tiedot ohjelmaa ADDStandardNODEIDS.py varten on olemassa.

5 Ohjelman ADDStandardNODEIDS.py toiminta

Ohjelman tarkoituksena on liittää kaikki tuotavat uudet tietolähteet oikeisiin kansioihin analysointiohjelmaa varten. Ohjelma käyttää kahta tiedostoa tietolähteinä, jotka ovat aiemmin luodut tiedostot Nodetree.csv ja Meters_to_RAU.csv.

Tiedosto Nodetree.csv sisältää analysointiohjelmaan vaaditut kansioiden nimet, tyypit ja tunnukset.

State	Name	NodeType	NodeID	ParentID
Existing	Remember to add from Navigator	Customer	1	2
Existing	Add building name	Kohde	3	4
Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	5	3
New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone	51	5
New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone	52	5
New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone	53	5
New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone	54	5
New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone	55	5
New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone	56	5
Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	6	3
Existing	Lämmöntuotto	Lämmöntuotto	7	3
New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto	71	7
Existing	Add building name	Rakennus/ yksikkö	8	3
New	Vesimittaukset LV103 :H'LV103	Osa	81	8

Kuva 30. Kuvakaappaus tiedostosta Nodetree.csv

Tiedosto Meters_to_RAU.csv sisältää uusien tietolähteiden tiedot.

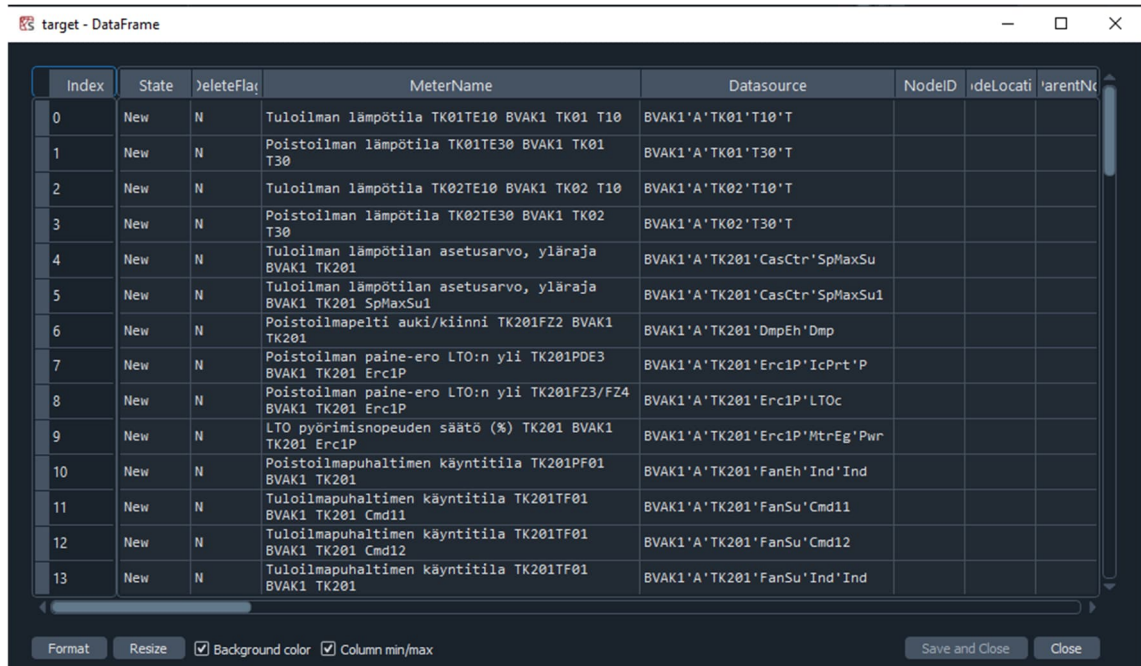
#	A	B	C	D	E	F	G	H	I	J	K	L	M	
	State	DeleteFla	MeterName	Datasource	NodeID	NodeLocation	OldParent	AttachTo	Detach	Device	Medium	Mediumrc	ReportGrc	Rep
1	New	N	Tuloilman lämpötila TK01TE10 BVAK1 TK01 T10	BVAK1'A'TK01'T10T							FDD Lämp	621	Tuloilman	
2	New	N	Poistoilman lämpötila TK01TE30 BVAK1 TK01 T30	BVAK1'A'TK01'T30T							FDD Lämp	621	Poistoilmi	
3	New	N	Tuloilman lämpötila TK02TE10 BVAK1 TK02 T10	BVAK1'A'TK02'T10T							FDD Lämp	621	Tuloilman	
4	New	N	Poistoilman lämpötila TK02TE30 BVAK1 TK02 T30	BVAK1'A'TK02'T30T							FDD Lämp	621	Poistoilmi	
5	New	N	Tuloilman lämpötilan asetusarvo, yläraja BVAK1 TK201	BVAK1'A'TK201'CasCtr'SpMaxSu							FDD Lämp	621	Tuloilman	
6	New	N	Tuloilman lämpötilan asetusarvo, yläraja BVAK1 TK201 SpMaxSu1	BVAK1'A'TK201'CasCtr'SpMaxSu1							FDD Lämp	621	Tuloilman	
7	New	N	Poistoilmapeltti auki/kiinni TK201FZ2 BVAK1 TK201	BVAK1'A'TK201'DmpEh'Dmp							FDD Statu	613	Poistoilmi	
8	New	N	Poistoilman paine-ero LTO:n yli TK201P0E3 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'LeP'P							FDD Paine	611	Poistoilmi	
9	New	N	Poistoilman paine-ero LTO:n yli TK201FZ3/FZ4 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'LTOe							FDD Paine	611	Poistoilmi	
10	New	N	LTO pyörimisnopeuden säätö (%) TK201 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'MtrEg'Pwr							FDD Aseni	608	LTO pyörii	
11	New	N	Poistoilmapuhaltimen käyntitila TK201PF01 BVAK1 TK201	BVAK1'A'TK201'FanEh'Ind'Ind							FDD Statu	613	Poistoilmi	
12	New	N	Tuloilmapuhaltimen käyntitila TK201F01 BVAK1 TK201 Cmd11	BVAK1'A'TK201'FanSu'Cmd11							FDD Statu	613	Tuloilmap	
13	New	N	Tuloilmapuhaltimen käyntitila TK201F01 BVAK1 TK201 Cmd12	BVAK1'A'TK201'FanSu'Cmd12							FDD Statu	613	Tuloilmap	
14	New	N	Tuloilmapuhaltimen käyntitila TK201F01 BVAK1 TK201	BVAK1'A'TK201'FanSu'Ind'Ind							FDD Statu	613	Tuloilmap	
15	New	N	Poistoilmasuodattimen paine-ero TK201PDE2 BVAK1 TK201	BVAK1'A'TK201'FIEH'DP							FDD Paine	611	Poistoilmi	
16	New	N	Tuloilmasuodattimen paine-ero TK201PDE1 BVAK1 TK201	BVAK1'A'TK201'FIEH'Su'DP							FDD Paine	611	Tuloilmas	
17	New	N	Jälkilämmityspatterin paluuveden lämpötila TK201TES BVAK1 TK201 TES	BVAK1'A'TK201'Hcl'TES'T							FDD Lämp	621	Jälkilämm	
18	New	N	IV-lämmityspatterin venttiilin asento (%) TK201TV1 BVAK1 TK201	BVAK1'A'TK201'Hcl'Viv							FDD Aseni	608	IV-lämmitt	

Kuva 31. Kuvakaappaus tiedostosta Meters_to_RAU.xlsx

Ohjelman tarkoituksena on liittää oikeat kansiotunnukset tiedoston Meters_to_RAU.csv sarakkeeseen E (NodeID) sekä kansion nimi sarakkeeseen

F (NodeLocation). Nämä tiedot saadaan tiedoston Nodetree.csv sarakkeiden kohdista D (NodeID) ja B (Name).

Ohjelma luo ensin Meters_to_RAU.csv-tiedostosta target-nimisen tietokehyksen.



Index	State	DeleteFlag	MeterName	Datasource	NodeID	NodeLocation	ParentNodeID
0	New	N	Tuloilman lämpötila TK01TE10 BVAK1 TK01 T10	BVAK1'A'TK01'T10'T			
1	New	N	Poistoilman lämpötila TK01TE30 BVAK1 TK01 T30	BVAK1'A'TK01'T30'T			
2	New	N	Tuloilman lämpötila TK02TE10 BVAK1 TK02 T10	BVAK1'A'TK02'T10'T			
3	New	N	Poistoilman lämpötila TK02TE30 BVAK1 TK02 T30	BVAK1'A'TK02'T30'T			
4	New	N	Tuloilman lämpötilan asetusarvo, yläraja BVAK1 TK201	BVAK1'A'TK201'CasCtr'SpMaxSu			
5	New	N	Tuloilman lämpötilan asetusarvo, yläraja BVAK1 TK201 SpMaxSu1	BVAK1'A'TK201'CasCtr'SpMaxSu1			
6	New	N	Poistoilmapelti auki/kiinni TK201FZ2 BVAK1 TK201	BVAK1'A'TK201'DmpEh'Dmp			
7	New	N	Poistoilman paine-ero LTO:n yli TK201PDE3 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'IcPrt'P			
8	New	N	Poistoilman paine-ero LTO:n yli TK201FZ3/FZ4 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'LTOc			
9	New	N	LTO pyörimisnopeuden säätö (%) TK201 BVAK1 TK201 Erc1P	BVAK1'A'TK201'Erc1P'MtrEg'Pwr			
10	New	N	Poistoilmahuuhtimen käyntitila TK201PF01 BVAK1 TK201	BVAK1'A'TK201'FanEh'Ind'Ind			
11	New	N	Tuloilmahuuhtimen käyntitila TK201TF01 BVAK1 TK201 Cmd11	BVAK1'A'TK201'FanSu'Cmd11			
12	New	N	Tuloilmahuuhtimen käyntitila TK201TF01 BVAK1 TK201 Cmd12	BVAK1'A'TK201'FanSu'Cmd12			
13	New	N	Tuloilmahuuhtimen käyntitila TK201TF01 BVAK1 TK201	BVAK1'A'TK201'FanSu'Ind'Ind			

Kuva 32. target-tietokehys

Ohjelma luo tiedostosta Nodetree.csv source-nimisen tietokehyksen.

1	Existing	Add building name	Kohde	3	1
2	Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	5	3
3	New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone	51	5
4	New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone	52	5
5	New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone	53	5
6	New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone	54	5
7	New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone	55	5
8	New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone	56	5
9	Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	6	3
10	Existing	Lämmöntuotto	Lämmöntuotto	7	6
11	New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto	71	7
12	Existing	Add building name	Rakennus/ yksikkö	8	3
13	New	Vesimittaukset LV103 :H'LV103	Osa	81	8

Kuva 33. source-tietokehys

Ohjelma muokkaa source-tietokehysen Name-saraketta siten, että poistetaan kaikki rivit, joissa ei esiinny merkki :

source - DataFrame

Index	State	Name	NodeType	NodeID	ParentID
3	New	A'TK01	Ilmanvaihtokone	51	5
4	New	A'TK02	Ilmanvaihtokone	52	5
5	New	A'TK201	Ilmanvaihtokone	53	5
6	New	A'TK202	Ilmanvaihtokone	54	5
7	New	A'TK203	Ilmanvaihtokone	55	5
8	New	A'TK207	Ilmanvaihtokone	56	5
11	New	H'IV101	Lämmitysverkosto	71	7
13	New	H'LV103	Osa	81	8

Kuva 34. source-tietokehysen Name-sarakkeen muokkaus

Ulkolämpötila ja ulkovaloisuus ovat ainoat tietolähteet, jotka liitetään pääkansion alle, joten source-tietokehyksestä tehdään kopio tätä tietolähdettä varten koodilla:

```
source_copy = pd.DataFrame()  
source_copy = source.copy(deep=True)
```

Ohjelman algoritmi laskee tietolähteiden ja kansioden nimien samankaltaisuuden ja niiden tuloksista tehdään similarity-tietokehys jokaisen tietolähteen Datasource ja source-tietokehyksen Name välisistä tuloksista.

similarity - DataFrame

Index	Name	Datasource	ratio
0	BVAK1'A'TK01	BVAK1'A'TK01'T10'T	1
1	BVAK1'A'TK02	BVAK1'A'TK01'T10'T	0.733333
2	BVAK1'A'TK201	BVAK1'A'TK01'T10'T	0.774194
3	BVAK1'A'TK202	BVAK1'A'TK01'T10'T	0.709677
4	BVAK1'A'TK203	BVAK1'A'TK01'T10'T	0.709677
5	BVAK1'A'TK207	BVAK1'A'TK01'T10'T	0.709677
6	BVAK1'H'KL100	BVAK1'A'TK01'T10'T	0.645161
7	BVAK1'H'LV103	BVAK1'A'TK01'T10'T	0.580645
8	BVAK1'H'IV101	BVAK1'A'TK01'T10'T	0.580645
9	BVAK1'A'TK01	BVAK1'A'TK01'T30'T	1
10	BVAK1'A'TK02	BVAK1'A'TK01'T30'T	0.733333
11	BVAK1'A'TK201	BVAK1'A'TK01'T30'T	0.774194
12	BVAK1'A'TK202	BVAK1'A'TK01'T30'T	0.709677
13	BVAK1'A'TK203	BVAK1'A'TK01'T30'T	0.774194
14	BVAK1'A'TK207	BVAK1'A'TK01'T30'T	0.709677
15	BVAK1'H'KL100	BVAK1'A'TK01'T30'T	0.645161
16	BVAK1'H'LV103	BVAK1'A'TK01'T30'T	0.580645
17	BVAK1'H'IV101	BVAK1'A'TK01'T30'T	0.580645
18	BVAK1'A'TK01	BVAK1'A'TK02'T10'T	0.8
19	BVAK1'A'TK02	BVAK1'A'TK02'T10'T	1

Kuva 35. similarity-tietokehyksen samankaltaisuuksien laskentojen arvot

Näistä jätetään jokaiselle Datasource-tietolähteen nimelle paras tulos koodilla:

```
#filteröidään similarity dataframesta löydetyt nimien vastaavuudet
siten että jätetään vain maksimi-arvot listaan
similarity =
similarity.iloc[similarity.groupby('Datasource')['ratio'].agg(pd.Series.idxmax)]
```

similarity - DataFrame

Index	Name	Datasource	ratio
676	BVAK1'A'TK02	BVAK1'Uas01'TE00'T	0.6
0	BVAK1'A'TK01	BVAK1'A'TK01'T10'T	1
9	BVAK1'A'TK01	BVAK1'A'TK01'T30'T	1
19	BVAK1'A'TK02	BVAK1'A'TK02'T10'T	1
28	BVAK1'A'TK02	BVAK1'A'TK02'T30'T	1
38	BVAK1'A'TK201	BVAK1'A'TK201'CasCtr'SpMaxSu	1
47	BVAK1'A'TK201	BVAK1'A'TK201'CasCtr'SpMaxSu1	1
56	BVAK1'A'TK201	BVAK1'A'TK201'DmpEh'Dmp	1
65	BVAK1'A'TK201	BVAK1'A'TK201'Erc1P'IcPrt'P	1
74	BVAK1'A'TK201	BVAK1'A'TK201'Erc1P'LTOc	1
83	BVAK1'A'TK201	BVAK1'A'TK201'Erc1P'MtrEg'Pwr	1
92	BVAK1'A'TK201	BVAK1'A'TK201'FanEh'Ind'Ind	1
101	BVAK1'A'TK201	BVAK1'A'TK201'FanSu'Cmd11	1
110	BVAK1'A'TK201	BVAK1'A'TK201'FanSu'Cmd12	1
119	BVAK1'A'TK201	BVAK1'A'TK201'FanSu'Ind'Ind	1
128	BVAK1'A'TK201	BVAK1'A'TK201'FilEh'DP	1
137	BVAK1'A'TK201	BVAK1'A'TK201'FilSu'DP	1
146	BVAK1'A'TK201	BVAK1'A'TK201'Hc1'TE5'T	1
155	BVAK1'A'TK201	BVAK1'A'TK201'Hc1'V1v	1
164	BVAK1'A'TK201	BVAK1'A'TK201'T1'T	1
173	BVAK1'A'TK201	BVAK1'A'TK201'T2'T	1
182	BVAK1'A'TK201	BVAK1'A'TK201'T3'T	1
191	BVAK1'A'TK201	BVAK1'A'TK201'T6'T	1

Kuva 36. similarity-tietokehyksen samankaltaisuuksien parhaat laskenta-arvot

Similarity-tietokehykseen liitetään source-tietokehyksen tiedot Name-sarakkeen perusteella koodilla:

```
similarity = similarity.merge(source, on='Name', how='left')
```

The image shows two side-by-side Jupyter Notebook windows, both titled 'similarity - DataFrame'. The left window displays a DataFrame with the following data:

Index	Name	Datasource	ratio
0	A'TK01	A'TK01'FG01TuloPeltiO	1
8	A'TK01	A'TK01'FG02LTOS	1
16	A'TK01	A'TK01'FG30PoistoPeltiO	1
24	A'TK01	A'TK01'LTLaskHyotys	1
32	A'TK01	A'TK01'ME20SaunaKosteusM	1
40	A'TK01	A'TK01'PDIE01M	1
48	A'TK01	A'TK01'PDIE10TuloM	1
56	A'TK01	A'TK01'PDIE30PoistoM	1
64	A'TK01	A'TK01'PDIE31M	1
72	A'TK01	A'TK01'PDIE35LTOM	1
80	A'TK01	A'TK01'PF025C01SaunaS	1
88	A'TK01	A'TK01'PF045C01Porrass	1
96	A'TK01	A'TK01'PF055C01Porrass	1
104	A'TK01	A'TK01'PU40I	1

The right window displays the result of a merge operation, showing the same data as the left window but with additional columns: State, NodeType, NodeID, and ParentID. The data is as follows:

Index	Name	Datasource	ratio	State	NodeType	NodeID	ParentID
0	A'TK01	A'TK01'FG01TuloPeltiO	1	New	Ilmanvaihtokone	51	5
1	A'TK01	A'TK01'FG02LTOS	1	New	Ilmanvaihtokone	51	5
2	A'TK01	A'TK01'FG30PoistoPeltiO	1	New	Ilmanvaihtokone	51	5
3	A'TK01	A'TK01'LTLaskHyotys	1	New	Ilmanvaihtokone	51	5
4	A'TK01	A'TK01'ME20SaunaKosteusM	1	New	Ilmanvaihtokone	51	5
5	A'TK01	A'TK01'PDIE01M	1	New	Ilmanvaihtokone	51	5
6	A'TK01	A'TK01'PDIE10TuloM	1	New	Ilmanvaihtokone	51	5
7	A'TK01	A'TK01'PDIE30PoistoM	1	New	Ilmanvaihtokone	51	5
8	A'TK01	A'TK01'PDIE31M	1	New	Ilmanvaihtokone	51	5
9	A'TK01	A'TK01'PDIE35LTOM	1	New	Ilmanvaihtokone	51	5
10	A'TK01	A'TK01'PF025C01SaunaS	1	New	Ilmanvaihtokone	51	5
11	A'TK01	A'TK01'PF045C01Porrass	1	New	Ilmanvaihtokone	51	5
12	A'TK01	A'TK01'PF055C01Porrass	1	New	Ilmanvaihtokone	51	5
13	A'TK01	A'TK01'PU40I	1	New	Ilmanvaihtokone	51	5

Kuva 37. source-tietokehyksen tietojen liittäminen similarity-tietokehykseen

Similarity-tietokehyksen kansiotunnukset NodeID liitetään target-tietokehyksen NodeID-sarakkeeseen Datasource-tunnuksien perusteella. Tietokehyksen savesourcedatasource sarake Name liitetään target-tietokehyksen sarakkeeseen NodeLocation käyttäen NodeID saraketta koodilla:


```

target['NodeID'] =
target['Datasource'].map(similarity.set_index('Datasource')['NodeID'])
target['NodeLocation'] =
target['NodeID'].map(savesourcedatasource.set_index('NodeID')['Name'])

```

The image displays three screenshots of a Jupyter Notebook interface, showing the results of data processing operations on DataFrames.

Screenshot 1: similarity - DataFrame

Index	Name	Datasource	ratio	State	NodeType	NodeID	ParentID
0	A'TK01	A'TK01'FG01TuloPeltio	1	New	Ilmanvaihtokone	51	5
1	A'TK01	A'TK01'FG02LTOS	1	New	Ilmanvaihtokone	51	5
2	A'TK01	A'TK01'FG30PoistoPeltio	1	New	Ilmanvaihtokone	51	5
3	A'TK01	A'TK01'LTOlaskHyotys	1	New	Ilmanvaihtokone	51	5
4	A'TK01	A'TK01'ME20SaunaKosteusM	1	New	Ilmanvaihtokone	51	5

target - DataFrame

Index	State	DeleteFlag	MeterName	Datasource	NodeID	NodeLocation	ParentNo	AttachTo
0	New	N	Raitisilmapelti auki/kiinni TK01 FG01 TK01 FG01TuloPeltio	A'TK01'FG01TuloPeltio				
1	New	N	LTO-pellin asento (%) TK01 FG02 TK01 FG02LTOS	A'TK01'FG02LTOS				
2	New	N	Jäteilmapelti auki/kiinni FG30 TK01 FG30PoistoPeltio	A'TK01'FG30PoistoPeltio				
3	New	N	TK01 LTO laskettu hyötysuhde	A'TK01'LTOlaskHyotys				
4	New	N	TK01 PF02 Sauna kosteusmittaus	A'TK01'ME20SaunaKosteusM				

target - DataFrame

Index	MeterName	Datasource	NodeID	NodeLocation	ParentNo	AttachTo
0	itisilmapelti auki/kiinni TK01 FG01 TK01 01TuloPeltio	A'TK01'FG01TuloPeltio	51	Ilmanvaihtokone TK01 :A'TK01		
1	0-pellin asento (%) TK01 FG02 TK01 02LTOS	A'TK01'FG02LTOS	51	Ilmanvaihtokone TK01 :A'TK01		
2	teilmapelti auki/kiinni FG30 TK01 30PoistoPeltio	A'TK01'FG30PoistoPeltio	51	Ilmanvaihtokone TK01 :A'TK01		
3	01 LTO laskettu hyötysuhde	A'TK01'LTOlaskHyotys	51	Ilmanvaihtokone TK01 :A'TK01		
4	01 PF02 Sauna kosteusmittaus	A'TK01'ME20SaunaKosteusM	51	Ilmanvaihtokone TK01 :A'TK01		

Kuva 38. kansiodien tietojen lisäys target-tietokehykseen

Tarkistusalgoritmi käy läpi sisältääkö target-tietokehyksen sarake Reportgroup sanaa Ulkolämpötila tai Ulkovaloisuus (lauselogiikka) ja muuttaa kansion tunnuksen ja nimen näiden kohdalle oikeaksi koodilla:

```

for i in target.index:
    for a in source_copy.index:
        # print(target.at[i,'ReportGroup'], ' &&& '
        ,source_copy['Name'][a])
        if target.at[i,'ReportGroup'] == 'Ulkolämpötila' or
target.at[i,'ReportGroup'] == 'Ulkovaloisuus':# and
(source_copy.at[a,'Name']) in target.at[i,'Datasource']:
            target.at[i,'NodeID'] = source_copy.at[a,'NodeID']
            target.at[i,'NodeLocation'] = source_copy.at[a,'Name']

```

The image shows two screenshots of a data table with columns: Index, State, Yleisefilä, MeterName, Datasource, NodeID, NodeLocation, IrentNo, AttachTo, Detach, Device, Medium, Vedumic, and ReportGroup. The top screenshot shows the original data. The bottom screenshot shows the modified data, where rows 35 and 26 have been updated with NodeID and NodeLocation values from the source data.

Index	State	Yleisefilä	MeterName	Datasource	NodeID	NodeLocation	IrentNo	AttachTo	Detach	Device	Medium	Vedumic	ReportGroup
34	New	N	Sekundääripumpun ohjaus PM41 L52 PM411	H\LS2\PM411	81	Vesimittaukset LV103 :H\LV103					FOO Status	613	Sekundääripumpun ohjaus
20	New	N	TK01 TE18 tulolima	A\TK01\TE18TuloH	51	Ilanvaihtokone TK01 :A\TK01					FOO Lämpötila	621	Tuloliman lämpötila
18	New	N	Tuloliman lämpötila LTO:n jälkeen TK01 TE02 TK01 TE02TuloTOGH	A\TK01\TE02TuloTOGH	51	Ilanvaihtokone TK01 :A\TK01					FOO Lämpötila	621	Tuloliman lämpötila LTO:n jälkeen
19	New	N	Tuloliman lämpötilan asetusarvo TK01 TE10 TK01 TE10Lassas	A\TK01\TE10Lassas	51	Ilanvaihtokone TK01 :A\TK01					FOO Lämpötila	621	Tuloliman lämpötilan asetusarvo
6	New	N	Tulolimapaive IV-koneella TK01 POIE10 TK01 POIE10TuloH	A\TK01\POIE10TuloH	51	Ilanvaihtokone TK01 :A\TK01					FOO Paine	611	Tulolimapaive IV-koneella
14	New	N	TK01 TP01 SC10 tulopuhallin käynnillä	A\TK01\SC10Tuloi	51	Ilanvaihtokone TK01 :A\TK01					FOO Status	613	Tulolimapuhallin käynnillä
15	New	N	Tulolimapuhallimen taausmuuttajan säätöviesti TK01 SC10 TK01 SC10TuloS	A\TK01\SC10TuloS	51	Ilanvaihtokone TK01 :A\TK01					FOO Käyntinopeus (%)	587	Tulolimapuhallimen taausmuuttajan säätöviesti
5	New	N	Tulolimasuodattimen paine-ero TK01 POIE01 TK01 POIE01H	A\TK01\POIE01H	51	Ilanvaihtokone TK01 :A\TK01					FOO Paine	611	Tulolimasuodattimen paine-ero
35	New	N	Ulkolämpötila TE00 L52 TE00UkolTH	H\LS2\TE00UkolTH	81	Vesimittaukset LV103 :H\LV103					FOO Lämpötila	621	Ulkolämpötila
26	New	N	Ulkovaloisuus VENN\UlkovaloisuusM	E\VENN\UlkovaloisuusM	71	Lämmitysverkosto IV101 :H\IV101					FOO Valoisuus	685	Ulkovaloisuus
23	New	N	Valaistuksen tilatieto (päällä/pois) UWR2	E\UWR2\PylvaaTO	52	Ilanvaihtokone TK02 :A\TK02					FOO Status	613	Valaistuksen tilatieto (päällä/pois)

Index	State	Yleisefilä	MeterName	Datasource	NodeID	NodeLocation	IrentNo	AttachTo	Detach	Device	Medium	Vedumic	ReportGroup
18	New	N	Tuloliman lämpötila LTO:n jälkeen TK01 TE02 TK01 TE02TuloTOGH	A\TK01\TE02TuloTOGH	51	Ilanvaihtokone TK01 :A\TK01					FOO Lämpötila	621	Tuloliman lämpötila LTO:n jälkeen
19	New	N	Tuloliman lämpötilan asetusarvo TK01 TE10 TK01 TE10Lassas	A\TK01\TE10Lassas	51	Ilanvaihtokone TK01 :A\TK01					FOO Lämpötila	621	Tuloliman lämpötilan asetusarvo
6	New	N	Tulolimapaive IV-koneella TK01 POIE10 TK01 POIE10TuloH	A\TK01\POIE10TuloH	51	Ilanvaihtokone TK01 :A\TK01					FOO Paine	611	Tulolimapaive IV-koneella
14	New	N	TK01 TP01 SC10 tulopuhallin käynnillä	A\TK01\SC10Tuloi	51	Ilanvaihtokone TK01 :A\TK01					FOO Status	613	Tulolimapuhallin käynnillä
15	New	N	Tulolimapuhallimen taausmuuttajan säätöviesti TK01 SC10 TK01 SC10TuloS	A\TK01\SC10TuloS	51	Ilanvaihtokone TK01 :A\TK01					FOO Käyntinopeus (%)	587	Tulolimapuhallimen taausmuuttajan säätöviesti
5	New	N	Tulolimasuodattimen paine-ero TK01 POIE01 TK01 POIE01H	A\TK01\POIE01H	51	Ilanvaihtokone TK01 :A\TK01					FOO Paine	611	Tulolimasuodattimen paine-ero
35	New	N	Ulkolämpötila TE00 L52 TE00UkolTH	H\LS2\TE00UkolTH	3	Add building name					FOO Lämpötila	621	Ulkolämpötila
26	New	N	Ulkovaloisuus VENN\UlkovaloisuusM	E\VENN\UlkovaloisuusM	3	Add building name					FOO Valoisuus	685	Ulkovaloisuus
23	New	N	Valaistuksen tilatieto (päällä/pois) UWR2	E\UWR2\PylvaaTO	52	Ilanvaihtokone TK02 :A\TK02					FOO Status	613	Valaistuksen tilatieto (päällä/pois)

Kuva 39. target-tietokehyksen tietolähteiden Ulkolämpötila ja Ulkovaloisuus kansioden tunnuksien muokkaus

Ylemmässä kuvassa rivi nro 35 (Ulkolämpötila) ja rivi nro 26 (Ulkovaloisuus) NodeID ja NodeLocation on muutettu tarkistusalgoritmissa liittämään tietolähteen kansioon Add building name (Kohde).

Tämän jälkeen target-tietokehyksen tulostetaan excel-tiedostoksi koodilla:

```
with pd.ExcelWriter('RAU_update_list.xlsx') as writer:
    target.to_excel(writer, index = False)
```

A	B	C	D	E	F	G	H
State	DeleteFlag	MeterName	Datasource	NodeID	NodeLocation	OldParentNodeID	Attachto
New	N	Raitisilmapelletti auki/kiinni TK01 FG01 TK01 FG01TuloPeltiO	A'TK01'FG01TuloPeltiO	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	LTO-pellin asento (%) TK01 FG02 TK01 FG02LTO	A'TK01'FG02LTO	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Jäteilmapelletti auki/kiinni FG30 TK01 FG30PoistoPeltiO	A'TK01'FG30PoistoPeltiO	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 LTO laskettu hyötysuhde	A'TK01'LTOlaskHyotys	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PF02 Sauna kosteusmittaus	A'TK01'ME20SaunaKosteusM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Tuloilmasuodattimen paine-ero TK01 PDIE01 TK01 PDIE01M	A'TK01'PDIE01M	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Tuloilmapaine IV-koneella TK01 PDIE10 TK01 PDIE10TuloM	A'TK01'PDIE10TuloM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PDIE30 Poistoilmakanava paine	A'TK01'PDIE30PoistoM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Poistoilmasuodattimen paine-ero TK01 PDIE31 TK01 PDIE31M	A'TK01'PDIE31M	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PDIE35 LTO paine-ero	A'TK01'PDIE35LTO	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PF02 Poistoilmahuollin saunatilat säätö	A'TK01'PF02SC01SaunaS	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PF04 Poistoilmahuollin porrashuone säätö	A'TK01'PF04SC01PorrasS	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 PF05 Poistoilmahuollin porrashuone säätö	A'TK01'PF05SC01PorrasS	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Heating Coil Pump Start/Stop Command TK01 PU40 TK01 PU40I	A'TK01'PU40I	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 TF01 SC10 tulopuhallin kayntitila	A'TK01'SC10'Tuloi	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Tuloilmahuollin taajuusmuuttajan säätöviesti TK01 SC10 TK01 SC10TuloS	A'TK01'SC10TuloS	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 SC30 poistoilmahuollin käyntitila	A'TK01'SC30PoistoI	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Poistoilmahuollin taajuusmuuttajan säätöviesti TK01 SC30 TK01 SC30PoistoS	A'TK01'SC30PoistoS	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Tuloilman lämpötila LTO:n jälkeen TK01 TE02 TK01 TE02TuloLTOJM	A'TK01'TE02TuloLTOJM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Tuloilman lämpötilan asetusarvo TK01 TE10 TK01 TE10LaskAs	A'TK01'TE10LaskAs	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 TE10 tuloilma	A'TK01'TE10TuloM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Poistoilman lämpötila TK01 TE30 TK01 TE30PoistoM	A'TK01'TE30PoistoM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Poistoilman lämpötila LTO:n jälkeen TK01 TE31 TK01 TE31PoistoLTOJM	A'TK01'TE31PoistoLTOJM	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	IV-lämmityspatterin venttiilin asento (%) TK01 TV45 TK01 TV45S	A'TK01'TV45S	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	TK01 TZA45 lämmityspatterin paluuvesi ja lämmitysvara	A'TK01'TZA45H	51	Ilmanvaihtokone TK01 :A'TK01		
New	N	Valaistuksen tilatieto (päällä/pois) UV02	E'UV02'Pylvaato	52	Ilmanvaihtokone TK02 :A'TK02		
New	N	Ulkovaloisuus VE00UlkovaloisuusM	E'VE00UlkovaloisuusM	3	Add building name		
New	N	Kaukolämpöveden tulolämpötila TE44 TE44MenoM	H'KLTE44MenoM	81	Vesimittaukset LV103 :H'LV103		
New	N	Kaukolämpöveden paluulämpötila TE49 TE49PaluuM	H'KLTE49PaluuM	81	Vesimittaukset LV103 :H'LV103		
New	N	LKV menoveden lämpötila TE40 LS1 TE40MenoM	H'LS1TE40MenoM	81	Vesimittaukset LV103 :H'LV103		
New	N	LKV paluueden lämpötila TE45 LS1 TE45PaluuM	H'LS1TE45PaluuM	81	Vesimittaukset LV103 :H'LV103		
New	N	Primääriventtiilin asento TV40.1 LS1 TV401S	H'LS1TV401S	71	Lämmitysverkosto IV101 :H'IV101		
New	N	Primääriventtiilin asento TV40.2 LS1 TV402S	H'LS1TV402S	81	Vesimittaukset LV103 :H'LV103		
New	N	Lämmitysverkoston paine PE01 LS2 PE01VerkostoPaineM	H'LS2'PE01VerkostoPaineM	53	Ilmanvaihtokone TK201 :A'TK201		
New	N	Sekundääripumpun ohjus PU41 LS2 PU41I	H'LS2'PU41I	81	Vesimittaukset LV103 :H'LV103		

Kuva 40. target-tietokehys tulostettuna muotoon RAU_update_list.xlsx

Tiedosto RAU_update_list.xlsx ladataan analysointiohjelmaan tietokannan hallintajärjestelmän kautta ja mittarit on liitetty analysointiohjelmaan.

6 Ohjelmien tulosten arviointi

Esimerkissä tarkasteltiin 1883 kappaleen rakennusautomaatiopisteen tuntemattomien syötteiden tietojen etsimistä opetusaineistosta. Kohdetta ei oltu aiemmin lisätty opetusaineistoon eli tarkoituksena oli tarkastella kuinka monta tietolähdettä oli tunnistettavissa olemassa olevalla opetusaineistolla 50839 kpl tietolähteitä.

Ohjelma tulostaa raportin, josta nähdään kuinka monen tuntemattoman syötteen tiedot löytyvät opetusaineistosta. Tässä kohtaa tuntemattomista syötteistä 1883 kpl tietolähteitä tunnettiin 106 kpl eli 5.63 % kokonaismäärästä.

```
Comparing source-database point-name similarities to target-database point-names
1883it [00:00, 40073.44it/s]
Source- database lookup report:
Found known point names: 106 / 1883 ( 5.63 % )
Not found: 1777 / 1883 ( 94.37 % )

Creating Meters_no_groups - excel list
Meters_no_groups - excel list ready
```

Kuva 41. Ohjelman MetersToRAU.py visuaalinen toiminta

Koska nostettavassa järjestelmässä oli rakennusautomaation grafiikkakuvien perusteella laskettu olevan 105 kappaletta niin havaittiin, että tietolähteitä oli löytynyt oikea määrä, mutta syöte- ja tulosteparien tiedot tuli tarkistaa Meters_to_RAU.xlsx-tiedostosta ja verrata niitä skannausohjelman discovery_file.csv-tiedoston tietoihin.

Nodetree.xlsx-tiedostosta havaittiin myös, että siinä oli virheellisiä sekä puutteellisia tietoja verrattuna Meters_no_groups.xlsx-tiedoston pohjalta laadittuun kansiorakenteeseen.

	A	B	C	D	E
1	State	Name	NodeType	NodeID	ParentID
2	Existing	Remember to add from Navigator	Customer	Remember to add from Navigator	Remember to add from Navigator
3	Existing	Add building name	Kohde	Remember to add from Navigator	Remember to add from Navigator
4	Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	Remember to add from Navigator	Remember to add from Navigator
5	New	Ilmanvaihtokone TK01 :A'TK01	Ilmanvaihtokone		
6	New	Ilmanvaihtokone TK02 :A'TK02	Ilmanvaihtokone		
7	New	Ilmanvaihtokone TK201 :A'TK201	Ilmanvaihtokone		
8	New	Ilmanvaihtokone TK202 :A'TK202	Ilmanvaihtokone		
9	New	Ilmanvaihtokone TK203 :A'TK203	Ilmanvaihtokone		
10	New	Ilmanvaihtokone TK207 :A'TK207	Ilmanvaihtokone		
11	Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	Remember to add from Navigator	Remember to add from Navigator
12	Existing	Lämmöntuotto	Lämmöntuotto	Remember to add from Navigator	Remember to add from Navigator
13	New	Lämmitysverkosto IV101 :H'IV101	Lämmitysverkosto		
14	Existing	Add building name	Rakennus/ yksikkö	Remember to add from Navigator	Remember to add from Navigator
15	New	Vesimittaukset LV103 :H'LV103	Osa		
16					
17					

Kuva 42. Tiedoston Nodetree.xlsx sisältö ennen opetusaineiston laadintaa

	A	B	C	D
1	New	N	Ilmastointi	BVAK1'A
2	New	N	TK01 Pakettikone huoneet	BVAK1'A'TK01
3	New	N	TK02 Pakettikone aulat	BVAK1'A'TK02
4	New	N	TK201 Yleisilmanvaihto	BVAK1'A'TK201
5	New	N	TK202 Keittiö	BVAK1'A'TK202
6	New	N	TK203 Osastot	BVAK1'A'TK203
7	New	N	TK207	BVAK1'A'TK207
8	New	N	Lämmitys	BVAK1'H
9	New	N	IV101 IV-verkosto	BVAK1'H'IV101
10	New	N	KL100 Kaukolämpö	BVAK1'H'KL100
11	New	N	LL101 Lattialämmitys vanha IV-koneh	BVAK1'H'LL101
12	New	N	LL102 Lattialämmitys	BVAK1'H'LL102
13	New	N	LL103 Lattialämmitys	BVAK1'H'LL103
14	New	N	LV103 Lämminkäyttövesi	BVAK1'H'LV103
15	New	N	Yleinen data AS01	BVAK1'Uas01
16	New	N	RE00 Valoisuus	BVAK1'Uas01'RE00
17	New	N	TE00 ulkolämpötila	BVAK1'Uas01'TE00'T
18	New	N	KV300VM Kylmävesi	BVAK1'U'Eg1'VIm02
19				

Kuva 43. Meters_no_groups.xlsx-listalta kerätyt järjestelmien tunnuksat

Ilmanvaihtokoneita löytyi oikea määrä 6 kpl, mutta lämmitysverkostosta jäi puuttumaan kansiot lattialämmitysverkostoille H'LL101, H'LL102 ja H'LL103. Lisäksi havaittiin, että tiedoston Nodetree.xlsx kansio: Vesimittaukset LV103:H'LV103 on virheellisesti tehty, koska sen pitäisi olla Lämmöntuotto-kansion alla nimettynä kansioksi: Käyttöveden lämmönsiirrin.

Tiedoston Meters_to_RAU.xlsx tietolähteiden tiedot tarkistettiin ja verrattiin niitä skannausohjelman discovery_file.csv-tiedoston tietoihin. Tarkastuksessa kirjattiin seuraavat tiedot:

Taulukko 3. tulokset ennen opetusaineiston laadintaa

Tieto	kpl	osuus [%]
Tarvittavat syötetiedot	105	
Tuntemattomista syötetiedoista tunnettiin	106	6
Tuntemattomista syötetiedoista tarpeettomia tietolähteitä	26	25
Tuntemattomista syötetiedoista tarpeellisia tunnettiin	80	75
Tarpeellisista syötetiedoista virheellisiä raporttiryhmiä	33	41
Tarpeellisista syötetiedoista puuttuvia syötetietoja	25	31

Yhteensä 25 kpl tarpeellisia tietolähteitä jäi puuttumaan tarpeellisista löydetyistä tietolähteistä. Tarpeellisista löydetyistä tietolähteistä 80 kpl virheellisten raporttiryhmiä osuus oli 41%. Ilman opetusaineiston laatimista tuntemattomille syötteille löydettiin kokonaisuudessaan 106 kpl määrästä 80 kpl tarpeellisia pisteitä ja 25 kpl jäi vielä täydennettäväksi opetusaineiston NEW_METERS.csv – listaan.

6.1 Opetusaineiston lisäys NEW_meters.xls–tiedostoon

Opetusaineisto lisättiin NEW_METERS.csv – tiedostoon. Yhteensä lisättäviä tietoja lisättiin 45 kpl. Opetusaineistoon lisättiin samantyyppisiä tietolähteiden tunnuksia, joita oli käytetty myös muissa järjestelmissä. Alla kuva ilmanvaihtokoneiden osalta tehdystä opetusaineistosta.

BVAK1'A'TK201'Hcl'TE5'T				FDD Lämpötila	621	IV-lämmityspatterin paluuveden lämpötila	100001322	°C
BVAK1'A'TK201'T1'T				FDD Lämpötila	621	Tuloilman lämpötila	100000520	°C
BVAK1'A'TK201'T2'T				FDD Lämpötila	621	Poistoilman lämpötila	100000006	°C
BVAK1'A'TK201'T3'T				FDD Lämpötila	621	Poistoilman lämpötila LTO:n jälkeen	100001052	°C
BVAK1'A'TK201'T6'T				FDD Lämpötila	621	Tuloilman lämpötila LTO:n jälkeen	100001291	°C
BVAK1'A'TK202'Erc1P'IcPrt'T8'T				FDD Lämpötila	621	LTO-patterin menolämpötila	100001288	°C
BVAK1'A'TK202'Erc1P'IcCmd				FDD Status	613	Primäärinummiin käyntitila	100000989	-

Kuva 44. Esimerkki ilmanvaihtokoneille tehdystä opetusaineistosta

Tässä on lisätty lämpötilojen tunnukset yhdelle ilmanvaihtokoneelle. Opetusaineistoon ei lisätty samoja tunnuspäätteitä käyttävien laitteiden tietolähteiden tietoja. Opetusaineistoon lisättiin tietoja lämmitysryhmille siten, että yksilöidyt tunnukset lisättiin omiaan ja samoja tunnuksia kuten Lämmitysverkoston paine tai Lämmityksen asetusarvo lisättiin 1 kpl samaan järjestelmään kuuluvalla ryhmällä. Tässä pätee myös sama logiikka, eli samantyyppisten järjestelmien samoja tunnuksia käyttävien laitteiden osalta riittää yhden esimerkin lisääminen opetusaineistoon.

BVAK1'A'TK202'ReCcl'T10'T				FDD Lämpötila	621	Huonelämpötila	119	°C
BVAK1'A'TK202'ReCcl'T9'T				FDD Lämpötila	621	Tuloilman lämpötila	100000520	°C
BVAK1'A'TK202'ReHcl'Hcl'TE7'T				FDD Lämpötila	621	IV-lämmityspatterin paluuveden lämpötila	100001322	°C
BVAK1'A'TK202'ReHcl'Hcl'Vlv				FDD Asento (%)	608	Jälkilämmityspatterin venttiilin asento (%)	100000602	%
BVAK1'A'TK202'ReHcl'T3'T				FDD Lämpötila	621	Tuloilman lämpötila	100000520	°C
BVAK1'A'TK202'ReHcl'T4'T				FDD Lämpötila	621	Huonelämpötila	119	°C
BVAK1'A'TK203'Hcl'TE5'T				FDD Lämpötila	621	IV-lämmityspatterin paluuveden lämpötila	100001322	°C
BVAK1'A'TK207'Hcl'TE4'T				FDD Lämpötila	621	IV-lämmityspatterin paluuveden lämpötila	100001322	°C
BVAK1'H'IV101'P1'P				FDD Paine	611	Lämmitysverkoston paine	100001501	bar
BVAK1'H'IV101'Vlv				FDD Asento (%)	608	Primääriventtiilin asento	100001166	%
BVAK1'H'KL100'T2'T				FDD Lämpötila	621	Kaukolämpöveden tulolämpötila	100000980	°C
BVAK1'H'KL100'T3'T				FDD Lämpötila	621	Kaukolämpöveden paluulämpötila	100000977	°C
BVAK1'H'LL101'Pu				FDD Status	613	Primääripumpun käyntitila	100000989	-
BVAK1'H'IV101'Pu				FDD Status	613	Primääripumpun käyntitila	100000989	-
BVAK1'H'LL101'StupCtl'SPeff				FDD Lämpötila	621	Lämmityksen asetusarvo	100001300	°C
BVAK1'H'LL101'T2'T				FDD Lämpötila	621	Lämmityspiirin menoveden lämpötila	100001293	°C
BVAK1'H'IV101'T1'T				FDD Lämpötila	621	Lämmityspiirin menoveden lämpötila	100001293	°C
BVAK1'H'LL101'T3'T				FDD Lämpötila	621	Huonelämpötila	119	°C
BVAK1'H'LL101'Vlv				FDD Asento (%)	608	Primääriventtiilin asento	100001166	%
BVAK1'H'LL102'Pu40'Ind'Ind				FDD Status	613	Primääripumpun käyntitila	100000989	-
BVAK1'H'LL102'T40'T				FDD Lämpötila	621	Lämmityspiirin menoveden lämpötila	100001293	°C
BVAK1'H'LV103'T1'T				FDD Lämpötila	621	LKV paluuveden lämpötila	100001305	°C
BVAK1'H'LV103'Vlv				FDD Asento (%)	608	Primääriventtiilin asento	100001166	%
BVAK1'Uas01'RE00'T				FDD Valoisuus	685	Ulkovaloisuus	100001446	Lux
BVAK1'Uas01'TE00'T				Ulkolämpötila (20)	20	Ulkolämpötila	1	°C

Kuva 45. tiedoston NEW_METERS.csv opetusaineistoa

Tietojen lisäämisen jälkeen ohjelma MetersToRAU.py-suoritettiin uudelleen.

6.2 Ohjelman uudelleen suoritus lisätyn opetusaineiston kanssa

Ohjelma käynnistettiin uudelleen ja nyt tunnettuja syötetietoja havaittiin 138 kpl (7,33 %). Tuntemattomien syötteiden samankaltaisuuslaskenta opetusaineiston syötteisiin kesti noin 9 minuuttia (kuvassa kohta Creating Meters_to_RAU.xls - list). Kokonaisuudessaan ohjelman suorittaminen kesti noin 20 minuuttia.

```
Creating target-database from target- folder files
100%|██████████| 12/12 [00:00<00:00, 13.92it/s]

Creating source-database from RAU-discovery file from source-folder
100%|██████████| 1/1 [00:00<00:00, 21.32it/s]
Comparing source-database point-name similarities to target-database point-names
1883it [00:00, 20073.23it/s]
Source- database lookup report:
Found known point names: 138 / 1883 ( 7.33 % )
Not found: 1745 / 1883 ( 92.67 % )

Creating Meters_no_groups - excel list

Meters_no_groups - excel list ready
Creating Meters_to_RAU.xls - list
95%|██████████| 6581597/6915870 [08:31<00:32, 10159.92it/s]
```

Kuva 46. Ohjelman visuaalinen toiminta

Ohjelman suorittamisen jälkeen tarkistettiin uudelleen Meters_to_RAU.xlsx-tiedoston tiedot. Tietojen tarkistamisen jälkeen kirjattiin seuraavat tiedot.

Taulukko 4. tulokset opetusaineiston laatimisen jälkeen

Tieto	kpl	osuus [%]
Tarvittavat syötetiedot	105	
Tuntemattomista syötetiedoista tunnettiin	138	7
Tuntemattomista syötetiedoista tarpeettomia tietolähteitä	33	24
Tuntemattomista syötetiedoista tarpeellisia tunnettiin	105	76
Tarpeellisista syötetiedoista virheellisiä raporttiryhmiä	0	0
Tarpeellisista syötetiedoista puuttuvia syötetietoja	0	0

Kaikki tarvittava tieto löydettiin opetusaineistoon tehtyjen lisäyksien jälkeen. Havaittiin, että tarpeettomat tietolähteet olivat lämpötilasäätimien asetusarvoja, hälytyspisteitä sekä muutamia kansiotunnuksia kuten primääripumpun käyntitila BVAK1'H'LL103'Pu40, josta tarvittiin tietolähde BVAK1'H'LL103'Pu40'Cmd. Kansiotunnus voidaan poistaa järjestelmästä lataamalla raportti, josta selviää onko tietoa liikkunut rakennusautomaatioverkosta analysointiohjelmaan, joten näiden eteen ei ohjelmaa laatiessa ajateltu tehtävän mitään. Ajatuksena oli, että suuremman massan luonnin jälkeen ladataan raportti ja poistetaan tietolähteet, joihin ei ole tullut tietoa 1-2 päivän jälkeen.

Meters_to_RAU.xlsx-tiedostoa tarkasteltaessa havaittiin, että algoritmi osasi löytää samoja tietolähdetunnuksia käyttäviä järjestelmiä, kuten ilmanvaihtokoneiden osalta. Opetusaineistoon lisättiin yhdelle ilmanvaihtokoneelle alla olevat tiedot lämpömittareista:

BVAK1'A'TK201'Hc'TE5'T	FDD Lämpötila	621	IV-lämmityspatterin paluuvien lämpötila	100001322 °C
BVAK1'A'TK201'T1'T	FDD Lämpötila	621	Tuloilman lämpötila	100000520 °C
BVAK1'A'TK201'T2'T	FDD Lämpötila	621	Poistoilman lämpötila	100000006 °C
BVAK1'A'TK201'T3'T	FDD Lämpötila	621	Poistoilman lämpötila LTO:n jälkeen	100001052 °C
BVAK1'A'TK201'T6'T	FDD Lämpötila	621	Tuloilman lämpötila LTO:n jälkeen	100001291 °C
BVAK1'A'TK202'Erc1P'IcPrt'T8'T	FDD Lämpötila	621	LTO-patterin menolämpötila	100001288 °C
BVAK1'A'TK202'Erc1P'Pu'Cmd	FDD Status	613	Primäärinumpun käyntitila	100000989 -

Kuva 47. ilmanvaihtokoneeseen lisätyt yksilöidyt tiedot opetusaineistoon

Meters_to_RAU.xlsx-tiedostosta havaittiin, että samoja lämpömittareiden tunnuksia kohdistui myös muihin ilmanvaihtokoneisiin, jotka oli ohjelmoitu käyttämään samoja päätetunnuksia rakennusautomaatiojärjestelmään.

Poistoilman lämpötila TK01TE30 BVAK1 TK01 T30	BVAK1'A'TK01T30T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila TK02TE30 BVAK1 TK02 T30	BVAK1'A'TK02T30T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila TK201TE2 BVAK1 TK201 T2	BVAK1'A'TK201T2T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila TK202TE2 BVAK1 TK202 T2	BVAK1'A'TK202T2T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila TK203TE2 BVAK1 TK203 T2	BVAK1'A'TK203T2T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila TK207TE2 BVAK1 TK207 T2	BVAK1'A'TK207T2T	FDD Lämp 621	Poistoilman lämpötila	100000000 °C
Poistoilman lämpötila LTO:n jälkeen TK201TE3 BVAK1 TK201 T3	BVAK1'A'TK201T3T	FDD Lämp 621	Poistoilman lämpötila LTO:n jälkeen	100001052 °C
Poistoilman lämpötila LTO:n jälkeen TK203TE3 BVAK1 TK203 T3	BVAK1'A'TK203T3T	FDD Lämp 621	Poistoilman lämpötila LTO:n jälkeen	100001052 °C

Kuva 48. Kuva tiedostosta Meters_to_RAU.xlsx

Opetusaineistoon määriteltiin tiedot lattialämmitysverkostolle LL101 sekä tarvittavat muut yksilöidyt tietolähteiden tiedot.

BVAK1'A'TK207Hcl'TE4'T		FDD Lämp	621 IV-lämmityspatterin paluuveden lämpötila	1E+08 °C
BVAK1'H'IV101'P1'P		FDD Paine	611 Lämmitysverkoston paine	1E+08 bar
BVAK1'H'IV101'Vlv		FDD Asenl	608 Primääriventtiilin asento	1E+08 %
BVAK1'H'KL100'T2'T		FDD Lämp	621 Kaukolämpöveden tulolämpötila	1E+08 °C
BVAK1'H'KL100'T3'T		FDD Lämp	621 Kaukolämpöveden paluulämpötila	1E+08 °C
BVAK1'H'LL101'Pu		FDD Statu	613 Primääripumpun käyntitila	1E+08 -
BVAK1'H'IV101'Pu		FDD Statu	613 Primääripumpun käyntitila	1E+08 -
BVAK1'H'LL101'StupCt'l'SPeff		FDD Lämp	621 Lämmityksen asetusarvo	1E+08 °C
BVAK1'H'LL101'T2'T		FDD Lämp	621 Lämmityspiirin menoveden lämpötila	1E+08 °C
BVAK1'H'IV101'T1'T		FDD Lämp	621 Lämmityspiirin menoveden lämpötila	1E+08 °C
BVAK1'H'LL101'T3'T		FDD Lämp	621 Huonelämpötila	119 °C
BVAK1'H'LL101'Vlv		FDD Asenl	608 Primääriventtiilin asento	1E+08 %
BVAK1'H'LL102'Pu40'Ind'Ind		FDD Statu	613 Primääripumpun käyntitila	1E+08 -
BVAK1'H'LL102'T40'T		FDD Lämp	621 Lämmityspiirin menoveden lämpötila	1E+08 °C
BVAK1'H'LV103'T1'T		FDD Lämp	621 LKV paluuveden lämpötila	1E+08 °C
BVAK1'H'LV103'Vlv		FDD Asenl	608 Primääriventtiilin asento	1E+08 %

Kuva 49. Tiedoston NEW_METERS.csv opetusainestoa

Lämmitysryhmien kohdalla havaittiin, että ilman opetusaineistoa puuttuneet 3 kpl lattialämmitysryhmät löytyivät nyt ja samoja päätetunnuksia käyttävien tietolähteiden tiedot löytyivät myös. Kuvaan on merkitty keltaisella värillä turhia tietolähteitä.

Kuvasta voidaan havaita, että esimerkiksi lattialämmitysryhmien LL102 ja LL103 raporttiryhmä: Lämmityspiirin menoveden lämpötila kohdentui molemmille samoille päätetunnuksille T40'T, joten tietolähde löytyi myös LL103 ryhmästä vaikka se määriteltiin vain ryhmälle LL102.

Kaukolämpöveden tulolämpötila KL100TE2 BVAK1 KL100 T2	BVAK1\KL100T2T	FDD Lämpö621	Kaukolämpöveden tulolämpötila	100000984 °C
Kaukolämpöveden paluulämpötila KL100TE3 BVAK1 KL100 T3	BVAK1\KL100T3T	FDD Lämpö621	Kaukolämpöveden paluulämpötila	100000977 °C
Primääripumpun käyntitila LL101PU2 BVAK1 LL101	BVAK1\LL101Pu	FDD Statu 613	Primääripumpun käyntitila	100000985-
Lämmityksen asetusarvo BVAK1 LL101	BVAK1\LL101StupCtI'Speff	FDD Lämpö621	Lämmityksen asetusarvo	100001360 °C
Lämmityspiirin menoveden lämpötila LL101TE2 BVAK1 LL101 T2	BVAK1\LL101T2T	FDD Lämpö621	Lämmityspiirin menoveden lämpötila	100001291 °C
Huonelämpötila LL101TE3 BVAK1 LL101 T3	BVAK1\LL101T3T	FDD Lämpö621	Huonelämpötila	19 °C
Primääriventtiilin asento LL101TV2 BVAK1 LL101	BVAK1\LL101TV	FDD Asem608	Primääriventtiilin asento	100001166 %
Primääripumpun käyntitila LL102PU40 BVAK1 LL102 Pu40	BVAK1\LL102Pu40	FDD Statu 613	Primääripumpun käyntitila	100000985-
Primääripumpun käyntitila LL102PU40 BVAK1 LL102 Pu40	BVAK1\LL102Pu40Cnd	FDD Statu 613	Primääripumpun käyntitila	100000985-
Primääripumpun käyntitila LL102PU40 BVAK1 LL102 Pu40	BVAK1\LL102Pu40Ind	FDD Statu 613	Primääripumpun käyntitila	100000985-
Lämmityksen asetusarvo BVAK1 LL102	BVAK1\LL102StupCtI'Speff	FDD Lämpö621	Lämmityksen asetusarvo	100001360 °C
Lämmityspiirin menoveden lämpötila LL102TE40 BVAK1 LL102 T40	BVAK1\LL102T40T	FDD Lämpö621	Lämmityspiirin menoveden lämpötila	100001291 °C
Primääriventtiilin asento LL102TV40 BVAK1 LL102	BVAK1\LL102TV	FDD Asem608	Primääriventtiilin asento	100001166 %
Primääripumpun käyntitila LL103PU40 BVAK1 LL103 Pu40	BVAK1\LL103Pu40	FDD Statu 613	Primääripumpun käyntitila	100000985-
Primääripumpun käyntitila LL103PU40 BVAK1 LL103 Pu40	BVAK1\LL103Pu40Cnd	FDD Statu 613	Primääripumpun käyntitila	100000985-
Primääripumpun käyntitila LL103PU40 BVAK1 LL103 Pu40	BVAK1\LL103Pu40Ind	FDD Statu 613	Primääripumpun käyntitila	100000985-
Lämmityksen asetusarvo BVAK1 LL103	BVAK1\LL103StupCtI'Speff	FDD Lämpö621	Lämmityksen asetusarvo	100001360 °C
Lämmityspiirin menoveden lämpötila LL103TE40 BVAK1 LL103 T40	BVAK1\LL103T40T	FDD Lämpö621	Lämmityspiirin menoveden lämpötila	100001291 °C
Primääriventtiilin asento LL103TV40 BVAK1 LL103	BVAK1\LL103TV	FDD Asem608	Primääriventtiilin asento	100001166 %
Primääriventtiilin asento LV103TE1 BVAK1 LV103	BVAK1\LV103Sp	FDD Asem608	Primääriventtiilin asento	100001166 %
LKV paluuveden lämpötila LV103TE1 BVAK1 LV103 T1	BVAK1\LV103T1T	FDD Lämpö621	LKV paluuveden lämpötila	100001360 °C
Primääriventtiilin asento LV103TV1 BVAK1 LV103	BVAK1\LV103TV	FDD Asem608	Primääriventtiilin asento	100001166 %
Kojeen käyntitila PH01 BVAK1	BVAK1\U'Alm'AlmFire'Alm	FDD Statu 613	Kojeen käyntitila	100000576-
Kylmävesi (16) PH03 BVAK1 Alm1	BVAK1\U'Alm'AlmFire'Alm1	Kylmävesi7	Kylmävesi (16)	16 m³
Kojeen käyntitila PH02 BVAK1 Alm2	BVAK1\U'Alm'AlmFire'Alm2	FDD Statu 613	Kojeen käyntitila	100000576-
Kojeen käyntitila PH04 BVAK1 Alm4	BVAK1\U'Alm'AlmFire'Alm4	FDD Statu 613	Kojeen käyntitila	100000576-
Ulkovaloisuus RE00 BVAK1 Uas01 RE00	BVAK1\Uas01 RE00T	FDD Valoi 685	Ulkovaloisuus	100001440 Lux
Ulkolämpötila T000 BVAK1 Uas01 TE000	BVAK1\Uas01 TE000T	Ulkolämpö50	Ulkolämpötila	5 °C
Kylmävesi (16) BVAK1 Eg1 Vlm02 Sp1	BVAK1\U'Eg1 Vlm02'alm'Sp1	Kylmävesi7	Kylmävesi (16)	16 m³
Kylmävesi (16) KV300VM BVAK1 Eg1 Vlm02	BVAK1\U'Eg1 Vlm02'Cl	Kylmävesi7	Kylmävesi (16)	16 m³
Tilavuus KV300VM BVAK1 Eg1 Vlm02	BVAK1\U'Eg1 Vlm02'VlmOut	Virtausm59	Tilavuus	58 m³

Kuva 50. ohjelman algoritmi kohdentaa samoja tunnuksia samoissa järjestelmätyypeissä käytäville mittareille tiedot opetusaineiston perusteella

Tätä tietoa voidaan pitää merkittävänä, koska jos ajatellaan, että integroitava rakennusautomaatiojärjestelmä olisi kooltaan esimerkiksi 20 000 tietolähdettä ja se sisältäisi useita samoja päätetunnuksia käyttäviä ilmanvaihtokoneita, huonesäätimiä, lämmitysryhmiä, yms, niin silloin tietojen kohdistaminen syötteisiin onnistuu varsin nopeasti johtuen algoritmin kyvystä kohdistaa samoja tietolähdetyyppejä useisiin samantyyppisiin tietolähteisiin.

Alla kuva tiedostosta Nodetree.xlsx.

State	Name	NodeType	NodeID	ParentID
Existing	Remember to add from Navigator	Customer	Remember to add from Navigator	Remember to add from Navigator
Existing	Add building name	Kohde	Remember to add from Navigator	Remember to add from Navigator
Existing	Add building name Ilmanvaihtojärjestelmä	Ilmanvaihtojärjestelmä	Remember to add from Navigator	Remember to add from Navigator
New	Ilmanvaihtokone TK207 :A^TK207	Ilmanvaihtokone		
New	Ilmanvaihtokone TK203 :A^TK203	Ilmanvaihtokone		
New	Ilmanvaihtokone TK202 :A^TK202	Ilmanvaihtokone		
New	Ilmanvaihtokone TK201 :A^TK201	Ilmanvaihtokone		
New	Ilmanvaihtokone TK02 :A^TK02	Ilmanvaihtokone		
New	Ilmanvaihtokone TK01 :A^TK01	Ilmanvaihtokone		
Existing	Add building name Lämmitysjärjestelmä	Lämmitysjärjestelmä	Remember to add from Navigator	Remember to add from Navigator
Existing	Lämmöntuotto KL100 :H^KL100	Lämmöntuotto	Remember to add from Navigator	Remember to add from Navigator
New	Käyttöveden lämmönsiirrin LV103 :H^LV103	Käyttöveden lämmönsiirrin		
New	Lämmitysverkosto IV101 :H^IV101	Lämmitysverkosto		
New	Lämmitysverkosto LL103 :H^LL103	Lämmitysverkosto		
New	Lämmitysverkosto LL102 :H^LL102	Lämmitysverkosto		
New	Lämmitysverkosto LL101 :H^LL101	Lämmitysverkosto		
Existing	Add building name	Rakennus/ yksikkö	Remember to add from Navigator	Remember to add from Navigator
New	Vesimittaukset Alm :U^Alm	Osa		
New	Vesimittaukset Eg1 :U^Eg1	Osa		

Kuva 51. Tiedoston Nodetree.xlsx sisältö opetusaineiston laatimisen jälkeen

Alla kuva rakennusautomaation järjestelmästä kerätystä kansiorakenteesta.

	A	B	C	D
1	New	N	Ilmastointi	BVAK1'A
2	New	N	TK01 Pakettikone huoneet	BVAK1'A'TK01
3	New	N	TK02 Pakettikone aulat	BVAK1'A'TK02
4	New	N	TK201 Yleisilmanvaihto	BVAK1'A'TK201
5	New	N	TK202 Keittiö	BVAK1'A'TK202
6	New	N	TK203 Osastot	BVAK1'A'TK203
7	New	N	TK207	BVAK1'A'TK207
8	New	N	Lämmitys	BVAK1'H
9	New	N	IV101 IV-verkosto	BVAK1'H'IV101
10	New	N	KL100 Kaukolämpö	BVAK1'H'KL100
11	New	N	LL101 Lattialämmi vanha IV-koneh	BVAK1'H'LL101
12	New	N	LL102 Lattialämmi	BVAK1'H'LL102
13	New	N	LL103 Lattialämmi	BVAK1'H'LL103
14	New	N	LV103 Lämminkäyttövesi	BVAK1'H'LV103
15	New	N	Yleinen data AS01	BVAK1'Uas01
16	New	N	RE00 Valoisuus	BVAK1'Uas01'RE00
17	New	N	TE00 ulkolämpötila	BVAK1'Uas01'TE00'T
18	New	N	KV300VM Kylmävesi	BVAK1'U'Eg1'VIm02
19				

Kuva 52. Kohteen järjestelmien tiedot

Voidaan havaita, että ilmanvaihtokoneita löytyi oikea määrä 6kpl, kaukolämpöryhmä H'KL100 on oikein ylemmässä kuvassa ja sen kansiotyyppinä on Lämmöntuotto, joka on oikein. Lisäksi muut lämmöntuottoryhmään kuuluvat kansiot ovat rakentuneet oikein. Kansio Vesimittaukset Eg1 :U'Eg1 - kansio on oikein, koska tietolähteistä löydettiin 1 kpl kiinteistön vedenkulutuksen mittaus. Kansio Vesimittaukset Alm :U'Alm – kansio on turha, koska se sisältää kylmän veden hälytyspisteen. Hälytyspisteitä on lisätty analysointiohjelman mittareiden tietokantaan joissakin kohteissa, joissa on tarvittu kyseistä tietoa, mutta tässä integraatiotyössä ei nähty tarpeelliseksi saada tietoa veden kulutuksen mittaustiedon hälytyksestä.

7 Johtopäätökset

Ohjelmalla saavutettiin hyvä tulos ilman opetusaineistoa. Ongelmana on opetusainestossa olevat tietolähteet, joita ei mahdollisesti uudessa integraatiotyössä tarvita kuten hälytyspisteet ja lämpötilojen säätöarvot. Olemassa olevan opetusaineiston oikeellisuuden tarkastukseen tulisi kehittää jokin algoritmi, jotta voidaan olla varmoja, että tiedot ovat oikein. Koneoppimisen selkäranka on laadukas ja hyvin laadittu opetusaineistoa, jota ilman tuntemattomien syötetietojen kohdistaminen oikeisiin tulostetietoihin on mahdotonta.

Integraatiotyö vaatii tietojen tarkastusta ja sen täysin automaattiseksi luominen vaatisi vielä algoritmin kehittämistä skannaustiedostoa varten, joka voisi luoda opetusaineiston ilman ihmisen käsityövoimaa. Silti vaikka kyseinen algoritmi saataisiin tuotettua on vaikeaa tehdä integraatiotyö ilman ihmisen tekemää tarkastustyötä.

Työ oli haastava ja mielenkiintoinen ja se tarjoaa vielä paljon kehityskohteita kuten esimerkiksi opetusaineiston automaattisen algoritmin luomisen sekä olemassa olevan opetusaineiston tietokannan tietojen oikeellisuuden tarkistamisen. Lisäksi olisi mielenkiintoista selvittää miten ohjelmisto voitaisiin toteuttaa toimimaan esimerkiksi Azure-järjestelmän yhteyteen, jolloin opetusaineiston tietokanta päivittyisi automaattisesti.

Kaiken kaikkiaan ohjelman laatimisessa päästiin hyvin pitkälle ja hyvään lopputulokseen. Ohjelma pyrittiin laatimaan siten, että se muokkautuu automaattisesti erityyppisiin tiedostoihin kuten erilaisten tekstikenttien erikoismerkkien käyttöön. Erityyppisiä erikoismerkkejä saattoi esiintyä samassa rakennusautomaatioverkon skannaustiedostossa eri riveillä ja se teki ohjelman algoritmien rakentamisesta haastavaa.

Lähteet

- 1 Ailisto, Heikki. 2018. Tutkimusprofessori, Teknologiantutkimuskeskus VTT Oy. Tekoälyn käsitekartta. Verkkoaineisto. 12/6/2018.
<<https://tietokayttoon.fi/documents/1927382/2158283/Teko%C3%A4lyn+k%C3%A4sitekartta/a5c4b469-d8ae-4ce1-a5fc-f12981bae796>>. Luettu 3.3.2024.
- 2 Kämäräinen, Joni. 2023. Koneoppimisen perusteet. Otatieto.
- 3 Kallio, Aleks ja Kolari, Jukka. 2023 . Tekoäly 123 Matkaopas tulevaisuuteen. Docendo.