

Opinnäytetyö (YAMK)

Teknologiaosaamisen johtaminen

2020

Pouya Jafarzadeh

IOT-POHJAISEN ÄLYKODIN ENERGIANKULUTUKSEN ENNUSTAMINEN KONEOPPIMISEN AVULLA

Pouya Jafarzadeh

IOT-POHJAISEN ÄLYKODIN ENERGIANKULUTUKSEN ENNUSTAMINEN KONEOPPIMISEN AVULLA

Opinnäytetyön tavoitteena oli kehittää järjestelmä, jonka avulla voidaan ennustaa pientalon energiankulutusta koneoppimisen menetelmillä. Tutkimuksessa käsitellään myös asukkaiden läsnä- ja poissaolon, ympäristön muutokseen liittyvien tekijöiden, kuten lämpötilan ja kosteuden sekä ympäristön valaistusvoimakkuuden vaikutusta energiankulutukseen.

Tutkimusta varten rakennettiin Arduino-mikrokontrolleriin perustuva prototyyppi, jonka avulla dataa kerättiin taloon asennettujen sensoreiden avulla. Dataa analysoitiin mahdollisten energiankulutukseen vaikuttavien tekijöiden välisten korrelaatioiden selvittämiseksi. Talon hetkellistä energiankulutusta ennustettiin kahden tunnetun koneoppimisen mallin, k-lähimmän naapurin (KNN) ja monikerroksisen perceptron (MLP) -mallin avulla. Talon seuraavan tunnin energiankulutuksen ennustamisessa hyödynnettiin pitkän lyhytaikaisen muistin verkot (LSTM)-mallia, joka on tunnettu toistuvien neuroverkkojen koneoppimisen malli.

Tutkimuksesta saadut tulokset osoittivat, että MLP-mallin avulla saadaan tarkempia ennustustuloksia talon energiankulutuksesta KNN-malliin verrattuna. Tulosten perusteella MLP-mallin keskineliövirheen neliöjuuren arvo (1,62) on pienempi kuin KNN-mallilla saatu arvo (1,80). Vastaavasti LSTM-mallin avulla ennustettiin talon energiankulutusta seuraavalta tunnilta minimaalisella keskineliövirheen neliöjuuren arvolla 0,07. Tutkimuksessa saatuja tuloksia voidaan hyödyntää asuintalojen energiankulutuksen optimoinnissa, energian tuotannon suunnittelussa sekä uusiutuvien energialähteiden käytön edistämisessä.

ASIASANAT:

energiankulutus, ennustaminen, koneoppiminen, esineiden internet, älykoti, IoT, AI, neuroverkot

MASTER'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Master's Degree Programme in Technological Competence Management

2020 | 57 pages

Pouya Jafarzadeh

IOT- BASED SMART HOUSE ENERGY USAGE PREDICTION BY MEANS OF MACHINE LEARNING

The purpose of the present thesis is to implement a system in order to predict the energy consumption of a dwelling using machine learning methods. The study will also address the changes in factors such as the presence and absence of the residents, as well as environmental impacts such as the temperature and humidity, and the effect of ambient luminous intensity on energy consumption.

A prototype based on the Arduino microcontroller was built for the purpose of the study and was used to collect data. The data were collected using sensors installed in the house. The data obtained from the sensors were analyzed to determine the correlations between the possible factors influencing energy consumption. Two well-known prediction models were utilized i.e. Multi-Layer Perceptron (MLP) and K-Nearest Neighbor (KNN), in order to predict the energy consumption at the current time. Furthermore, it employs Long Short-Term Memory (LSTM) as one of the common recurrent neural networks for forecasting the next hour energy consumption.

Experimental results show that the MLP based prediction model provides substantial improvement over the KNN based model in terms of prediction accuracy. The prediction accuracy in terms of Root Mean Square Error (RMSE) metric for KNN and MLP are (1.80) and (1.62), respectively. In addition, LSTM-based prediction model achieves a very minimum RMSE value (0.07) for forecasting the energy consumption in the next one hour. The results of the study can be used to optimize the energy consumption of residential buildings, to plan energy production, and to promote the use of renewable energy sources.

KEYWORDS:

energy consumption forecasting, machine learning, internet of things, smart house, IoT, AI, neural networks

SISÄLTÖ

KÄYTETYT LYHENTEET TAI SANASTO	7
1 JOHDANTO	9
1.1 Tutkimustavoitteet ja tutkimuskysymykset	10
1.2 Aikaisemmat tutkimukset	11
1.3 Opinnäytetyön rakenne	13
2 TIETOPERUSTA	14
2.1 Esineiden Internet	14
2.2 Koneoppiminen	18
2.2.1 Keinotekoiset neuroverkot	20
2.2.2 Pitkät lyhytaikaisen muistin verkot	21
2.2.3 K-lähimmän naapuri	23
3 PROTOTYYPIN SUUNNITTELU JA RAKENTAMINEN	26
3.1 Prototyypin alustan valinta	26
3.2 Arduino	27
3.3 Sensori	28
3.4 Prototyypin rakentaminen	28
3.4.1 Sensoreiden kytkentä	29
3.4.2 Arduino Ethernet Shieldin asentaminen	29
3.5 Testiympäristö	30
3.6 Sensoreiden sijainti talossa	33
3.7 Ohjelmointi	33
3.8 Ubidots-alusta	34
3.9 Energiankulutus	35
4 ÄLYKKÄÄN MALLIN TOTEUTUS	36
4.1 Energiankulutuksen ennustamisen prosessi	36
4.2 Datan keräämisen kerros	38
4.3 Datan esivalmistelukerros	39
4.4 Ennustamisen kerros	43
4.5 Suorituskyvyn arviointi	44
5 TULOKSET	46

5.1 Monikerroksinen Perceptron (MLP)	46
5.2 K-lähimmän naapuri (KNN)	48
5.3 Pitkät lyhytaikaisen muistin verkot (LSTM)	49
6 YHTEENVETO	51
LÄHTEET	55

KAAVAT

Kaava 1. Kahden arvon, ennustetun ja todellisen havainnon erotuksen neliö.	20
Kaava 2. Kaltevuuden laskeutuminen (Guo & Gelfand, 1991).	21
Kaava 3. Sisääntuloportin, unohda-portin, ulostuloportin, solun tilan laskentakaavat (Hochreiter & Schmidhuber, 1997).	22
Kaava 4. Piilokerroksen ulostulon laskentakaava (Hochreiter & Schmidhuber, 1997).	22
Kaava 5. Datan normalisointi (Ojha, 2019).	40
Kaava 6. Keskineliövirheen laskentakaava (Binieli, 2018).	44
Kaava 7. Keskineliövirheen neliöjuuren laskentakaava (Binieli, 2018).	45

KUVAT

Kuva 1. IoT world forum reference model (Cisco, 2014).	16
Kuva 2. Ohjattu- ja ohjaamaton oppiminen (Lawtomated, 2019).	19
Kuva 3. Monikerroksinen Perceptron-malli.	21
Kuva 4. Pitkät lyhytaikaisen muistin verkkojen -rakenne (Xuan-Hien& Hung, 2019).	23
Kuva 5. K-lähimmän naapurin luokittelumalli (Raghunath, 2019).	24
Kuva 6. Arduino Mega.	27
Kuva 7. Arduino Ethernet Shield.	30
Kuva 8. Energiankulutuksen ennustamisen prosessikaavio.	37

KUVIOT

Kuvio 2. Kahden vuorokauden sensoriarvot.	35
---	----

Kuvio 3. Kerätyn sensoridatan pistediagrammi 26.11.2018 – 3.12.2018.	42
Kuvio 4. MLP-mallin MSE ja RMSE arvot eri piilokerroksilla	46
Kuvio 5. Energiankulutuksen ennustaminen MLP-mallin avulla yhdellä piilokerroksella.	47
Kuvio 6. Energiankulutuksen ennustaminen MLP-mallin kahdella piilokerroksella.	47
Kuvio 7. Energiankulutuksen ennustaminen MLP-mallin avulla kolmella piilokerroksella.	48
Kuvio 8. Energiankulutuksen ennustaminen KNN- menetelmän avulla.	49
Kuvio 9. Seuraavan tunnin energiankulutuksen ennustaminen LSTM-mallin avulla.	50

TAULUKOT

Taulukko 1. Projektissa käytetyt sensorit, niiden ominaisuudet ja sijoittelu.	28
Taulukko 2. E-luvun laskennan lähtötiedot.	31
Taulukko 3. Rakennuksen energiatehokkuus.	32
Taulukko 4. Kerätyn sensoridatan keskiarvot, hajonnat ja minimi- sekä maksimiarvot.	39
Taulukko 5. Neuronien määrä jokaisessa piilokerroksessa.	46

KÄYTETYT LYHENTEET TAI SANASTO

AMQP	Advanced Message Queuing Protocol, edistynyt viestijönojärjestelmä
ANFIS	Adaptive Network-based Fuzzy Inference System, neurosumea päättelyjärjestelmä
ANN	Artificial Neural Networks, keinotekoiset neuroverkot
CBR	Case-based reasoning, tapausperusteinen päättelymalli
DCPS	Data-Centric Publish-Subscribe, datakeskeinen julkaisutilaus
DDS	Data Distribution Service (Standard), tiedonjakelupalvelustandardi
DELML	Deep Extreme Learning Machine, äärimmäisen syväoppisen kone
DLRL	Data Local Reconstruction Layer, tietojen paikallinen jälleenrakennuskerros
GRNN	Generalized Regression Neural Network, yleisen regression neuroverkkomalli
HTTP	Hypertext Transfer Protocol, hypertekstin siirtoprotokolla
IoT	Internet of Things, esineiden internet
KNN	K-Nearest Neighbors, K-lähimmän naapurin koneoppimisen menetelmä
LOOCV	Leave-One-Out Cross-Validation, yksi pois -ristiinvalidointi mallin validointitekniikka
LSTM	Long Short-Term Memory, pitkät lyhytaikaisen muistin verkot, arkkitehtuuri
M2M	Machine to Machine, laitteiden välinen viestintä
MAE	Mean Absolute Error, keskimääräinen absoluuttinen virhe
MLP	Multi Layer Perceptron, monikerroksinen perseptroniverkko
MLR	Multiple Linear Regressions, monikerroksinen lineaarinen regressio -malli
MQTT	Message Queue Telemetry Transport, jonottava telemetrinen kuljetusverkkoprotokolla
MSE	Mean Square Error, keskineliövirhe
RBFNN	Radial Basis Function Network, radiaalisen perustoiminnon neuroverkkomalli

RASE	Square Root of Average Squared Error, keskimääräisen neliövirheen neliöjuuri
RMSE	Root Mean Square Error, keskineliövirheen neliöjuuri
RNN	Recurrent Neural Network, takaisinkytketty neuroverkkoarkkitehtuuri
SLR	Simple Linear Regression, yksinkertaisen lineaarinen regressio malli
SVM	Support Vector Machine, tukivektorikone, lineaarinen luokintamalli
TCP	Transmission Control Protocol, tietoliikenneprotokolla tietokoneiden väliseen luotettavaan tiedonsiirtoon
UDP	User Datagram Protocol, offline tietoliikenneprotokolla

1 JOHDANTO

Elämme maailmassa, jossa teknologia ja digitalisaatio vaikuttavat paljon jokaisen ihmisen arkielämään. Teknologian kehityksen, yleistymisen ja saatavuuden kasvu ovat edesauttaneet näiden integrointia päivittäisessä elämässämme. Tämän mullistuksen keskipisteessä on internet, joka yhdistää meitä kaikkia maailmanlaajuiseen verkkoon sekä avaa käyttäjilleen rajoittamattomia mahdollisuuksia ajasta ja paikasta riippumatta. Edellä mainitut seikat huomioon ottaen ei ole aiheutonta kutsua nykymaailmaa maailmankyläksi. Teknologian kehitys tuo mukanaan rajoittamattoman määrän mahdollisuuksia, jotka auttavat meitä kaikkia rakentamaan entistä älykkäämpää, turvallisempaa ja energiaystävällisempää yhteiskuntaa.

Viime vuosien tärkeimpiä teknologiaan liittyviä trendejä ovat muun muassa esineiden internet ja koneoppiminen. Esineiden internetin idea perustuu laitteiden yhdistämisestä internetiin, tiedonsiirtämiseen, automatisointiin, ohjaukseen sekä etähallinnan tarkoituksiin. Toinen edellä mainituista teknologiatrendeistä, joka on viime vuosina saanut huomiota, on koneoppiminen. Se on tekoälyn alaryhmä, joka pyrkii rakentamaan mallin, jossa kone oppii toimimaan ilman erillistä ohjeistusta tai käskyä hyödyntämällä käytössä olevaa dataa.

Teknologian ja siihen liittyvien infrastruktuurien eksponentiaalinen kasvu ja kehitys on mahdollistanut niiden integrointia jopa yksinkertaisimmissakin rutiinitoimenpiteissä. Esineiden internetin ja koneoppimisen avulla voidaan suunnitella ja rakentaa nykyaikaisia, automatisoituja, energiaystävällisiä ja etähallittavissa olevia älykoteja. Näiden kahden teknologian kombinaatio mahdollistaa automatisoituja toimenpiteitä, jotka lisäävät asumisen mukavuutta ja turvallisuutta sekä edesauttavat talon energiankulutuksen optimointia energiankulutuksen ennustamisen avulla.

Työ- ja elinkeinoministeriön teettämän raportin mukaan Suomen rakennussektorin energiankulutuksen osuus kokonaiskulutuksesta on 40% ja kasvihuonekaasupäästöistä 30% (Työ- ja elinkeinoministeriö, 2011). Tilanne vaatii konkreettisia ratkaisuja kohti tehokkaampaa optimoitua energiankulutusta.

Energiaystävällisiä älykoteja on tutkittu, kehitetty ja toteutettu vuosien ajan sekä tutkimuslaitoksissa että teollisuudessa. Koneoppimisen ja esineiden internetin kombinaatio mahdollistaa kuitenkin ainutlaatuisen tilaisuuden hyödyntää näiden kahden teknologian vahvuudet talon energiankulutuksen ennustamisessa. Energiankulutuksen ennustamisen ensisijaisia edellytyksiä ovat kulutukseen vaikuttavien tekijöiden tunnistaminen, seuranta ja analyysi. Koneoppimisen menetelmiä hyödyntämällä voidaan näiden kerättyjen tietojen perusteella rakentaa älykkäitä järjestelmiä, jotka mahdollistavat energiankulutuksen ennustamista ja optimointia.

1.1 Tutkimustavoitteet ja tutkimuskysymykset

Talon asukkaiden käyttäytyminen, elämäntavat, tottumukset sekä ulkopuoliset tekijät, kuten esimerkiksi talotekniikka ja sen kunto sekä myös ympäristön lämpötila, kosteus ja valaistusvoimakkuus vaikuttavat energiankulutukseen. Tämän projektin tavoitteena oli kerätä energiankulutukseen vaikuttavista tekijöistä tietoa sekä hyödyntää kerättyä mittausdataa energiankulutuksen ennustamisessa. Tutkimukseen tarvittavaa dataa kerättiin projektia varten rakennetulla prototyypillä. Tutkimuksessa hyödynnetään sensoreista saatua dataa energiankulutuksen ennustamiseen koneoppimisen menetelmien avulla. Energiankulutuksen ennustamisen avulla pyritään optimoimaan energiankulutusta.

Tutkimuksessa käsitellään monia tekniikan käsitteitä ja aloja, kuten elektroniikka, esineiden internet ja koneoppiminen, joten tutkimuksen tarkka rajaaminen on välttämätöntä projektin onnistumisen kannalta. Opinnäytteessä keskitytään yksittäisen pientalon energiankulutukseen. Testiympäristönä käytetyssä talossa sähköenergia on ainoa energianlähde. Tutkimusta rajataan energiankulutuksen ennustamiseen, joka edesauttaa energian optimointia. Varsinainen energiankäytön optimointi jätetään jatkotutkimusaiheeksi.

Tutkimuksen perusteella vastataan tutkimuskysymyksiin:

1. Miten asukkaiden läsnä- ja poissaolo ja aktiivisuus sisällä, sisä-/ulkolämpötila, sisäilman kosteuden määrä sekä valaistusvoimakkuuden määrä vaikuttavat talon energiankulutukseen?
 - 1.1 Miten ja millä kriteereillä tarvittavaa dataa kerätään älykkään mallin opettamiseen?
 - 1.2 Millaisia relaatioita kerätyn datan arvojen välillä esiintyy?
2. Miten koneoppimisen avulla voidaan ennustaa talon energiankulutusta sensoreista saatujen tietojen perusteella?
 - 2.1 Mitkä koneoppimisen menetelmät ja algoritmit sopisivat parhaiten energiankulutuksen ennustamiseen?

1.2 Aikaisemmat tutkimukset

Tässä tarkastellaan muutamia aikaisempia tutkimusaiheeseen liittyviä tutkimuksia. Tiedot on koottu eri tutkimusryhmien julkaistuista tieteellisistä artikkeleista, jotka on esitetty alan konferensseissa. Platonin tutkimusryhmä on hyödyntänyt kahta keinotekoisien neuroverkkojen (ANN) ja Tapausperusteinen päättely -mallia (CBR) rakennuksen energiankulutuksen ennustamisessa seuraavilta tunneilta. Mallin opetusdata perustuu ulkolämpötilan ja kosteuden sekä rakennuksen energian seurantajärjestelmään, joka seuraa rakennuksen veden lämmityksen sekä ilmanvaihdon energiankulutusta. Tutkimustuloksissa todettiin, että ANN-malli saavuttaa tarkempia energianennustustuloksia seuraavilta tunneilta CBR-malliin verrattuna. Tulosten perusteella ANN-mallin keskineliövirheen neliöjuuren arvo 0,07 on pienempi kuin CBR-malliin 0,13. (Platon ym. 2015.)

Tso ja Yau (2007) ovat tutkimuksessaan hyödyntäneet lineaariseen regressioon, päätöspuuhun sekä neuroverkkoihin perustuvaa koneoppimisen menetelmää virastojen, asuinrakennusten sekä mökkien energiankulutuksen ennustamisessa. Tutkimusta varten on kerätty dataa asukkaiden lukumäärästä, perheen koosta, rakennuksen iästä ja tyypistä sekä käytössä olevista

elektronisista laitteista. Tutkimustuloksissa todettiin, että päätöspuumenetelmä saavuttaa tarkempia energianennustustuloksia muihin menetelmiin verrattuna. Tulosten perusteella päätöspuumenetelmän keskimääräisen neliövirheen neliöjuuriarvo (RASE) 39,36 on pienempi kuin lineaarisen regression 39,42 ja neuroverkon 39,52. (Tso & Yau, 2007.)

Wallnerströmin tutkimusryhmä on työskennellyt energiankulutuksen ennustamisen parissa hyödyntämällä yksinkertaisen lineaarisen regression (SLR), sekä monikerroksisen lineaarisen regression mallia (MLR). Heidän tutkimuksensa perustuu kolmeen erilaiseen tapaukseen. Ensimmäisessä tapauksessa energiankulutukseen liittyvät tiedot perustuivat tunnin, toisessa tapauksessa vuoden ja viimeisessä tapauksessa päivittäisiin ruuhka-aikoihin. Tulosten pohjalta SLR-algoritmin avulla saatiin tarkempia ennustuksia kuin MLR-algoritmin ensimmäisen ja toisen tapauksen kohdalla. Ensimmäisen tapauksen kohdalla alin keskimääräinen absoluuttinen virhearvo oli 13,70%, toisen tapauksen kohdalla 0,25% ja viimeisen tapauksen kohdalla 3,20%. (Wallnerström ym. 2010.)

Troungin johtama tutkimusryhmä on lähestynyt energiankulutuksen ennustamista graafisen mallin avulla. Heidän tavoitteenaan on ollut löytää ihmisen arkipäivän rutiinien ja energiankulutuksen välistä riippuvuutta. Tarvittavaa dataa on kerätty kodinkoneiden energiankulutuksesta sekä esim. päivittäistä käyttöaikaa. Tulosten mukaan sovellettu malli ennustaa energiankulutusta 47% tarkkuudella. (Truong ym. 2013.)

Olofsson on tutkimuksessaan esittänyt neuroverkkoihin perustuvaa mallia, jossa tavoitteena on ennustaa pitkän ajanjakson energiankulutusta. Mallin opetusdata perustuu lyhytaikaiseen energiankulutukseen, sisä- ja ulkolämpötilaan sekä lämmityksessä käytettyyn energiamäärään. Tulosten perusteella vuosittaisen ennustetun ja mitatun päivittäisen energiantarpeen välinen poikkeama on ollut noin 4%. (Olofsson & Andersson, 2001.)

Fyazin tutkimusryhmän tavoitteena on ollut ennustaa asuinrakennusten energiankulutusta äärimmäisen koneoppimisen (DELM), adaptiivisen neuro-

sumea päättelyjärjestelmän (ANFIS), sekä keinotekoisien neuroverkkojen (ANN) avulla. Projektin tavoitteena on ollut ennustaa asuinrakennuksen energiankulutusta seuraavalle viikolle ja kuukaudelle. Tutkimusryhmä on kokeilut eri vaihtoehtoja, joilla saadaan mahdollisimman tarkkoja tuloksia. He ovat kokeilleet erilaisia ja eri suuruuksia jäsenyysfunktioita (ANFIS) algoritmile, sekä hyödyntäneet neuroverkoihin liittyviä arkkitehtuureita erilaisilla piilokerrosmäärillä. Tulosten perusteella DELM-algoritmin keskineliövirheen neliöjuuren arvo oli 2,24 , ANFIS-algoritmin 2,46 ja ANN-algoritmin kohdalla 2,6. (Fayaz & Dohyeun, 2018.)

Linin johtama tutkimusryhmä on hyödyntänyt takaisin leviämisen neuroverkkojen (ANN), radiaalisen perustoiminnon neuroverkkoja (RBFNN), yleisen regression neuroverkkojen (GRNN) sekä tukivektorikone -mallia (SVM) vuotuisen asuinrakennusten energiakulutuksen ennustamiseen. Malleissa hyödynnettyä dataa kerättiin 59 rakennuksesta ja testattiin 9 rakennuksessa. Tulosten mukaan GRNN- ja SVM -mallien avulla saatiin parempia ja tarkempia tuloksia. Tulosten perusteella GRNN-mallin keskineliövirheen neliöjuuren arvo oli 4,91 ja SVM-mallin kohdalla 1,89.(Qiong, 2010.)

1.3 Opinnäytetyön rakenne

Opinnäytteen teoriaosuudessa käsitellään esineiden internetin ja koneoppimisen taustat ja menetelmät, jotka ovat edellytyksenä seuraavien kokonaisuuksien hahmottamiseen.

Teoriaosuuden jälkeen siirrytään varsinaiseen prototyypin suunnitteluun ja toteutukseen. Tässä osiossa keskitytään erityisesti sensoridatan mittaamiseen, keräämiseen ja analysointiin liittyviin vaiheisiin.

Tämän jälkeen siirrytään vaiheeseen, joka on koneoppimisen algoritmien ja menetelmien hyödyntäminen tutkimuksen tavoitteiden saavuttamiseksi.

2 TIETOPERUSTA

2.1 Esineiden Internet

Esineiden internetillä (engl. Internet of Things = IoT) tarkoitetaan yksinkertaisesti elinympäristössämme päivittäin käytössämme olevien fyysisten sekä virtuaalisten laitteiden yhdistämistä internetiin (Ashton, 2009). Esineiden internetin tavoitteena on yhdistää yksittäiset itsenäisesti toimivat laitteet laajempaan kokonaisuuteen. Esineiden internetin avulla voidaan etähallita laitteita, kerätä dataa sekä automatisoida laitteen tai järjestelmän toimintaa. Esineiden internetiä voidaan hyödyntää yksinkertaisissa laitteissa, kuten jääkaapissa ja pesukoneessa tai vastaavasti suuressa kokonaisuudessa, kuten kaupunkiliikenteen infrastruktuurissa. (Meola, 2016.) Arvioiden mukaan vuoteen 2030 mennessä noin 125 miljardia laitetta on kytketty verkkoon (Abdulzاهر, 2019, 101).

Historia

Esineiden internetin konsepti ei ole kovin vanha, mutta siihen liittyviä keksintöjä on tehty jo 1800-luvulta lähtien, kun ensimmäiset lankapuhelimet tulivat markkinoille. Seuraavat keksinnöt, kuten langattomat äänilaitteet 1900-luvulla sekä tietokoneiden yleistyminen 1950-luvulla rakensivat hyvän pohjan IoT:n syntymiseen. Internet, joka on yksi tärkeimmistä esineiden internetin elementeistä, kehitettiin vuonna 1969 Yhdysvallan Puolustusministeriön alaisuudessa tutkimusprojektissa (Giovanni, 2016). Ensimmäiset verkkoon yhdistyneet kodinkoneet esiteltiin Hollannissa vasta vuonna 1989. Ensimmäisten joukossa oli John Romeklyn sekä Simon Hacketin suunnittelema etäohjattava leivänpaahdin, joka tuli markkinoille vuonna 1990. Vuonna 2000 LG julkaisi ensimmäisen älykkään jääkaapin, josta tuli yksi esineiden internetin symboli. Ensimmäinen akateeminen konferenssi IoT:sta järjestettiin Zürichissä vuonna 2008. (van Rijmenam, 2014.)

Internet

Internet on maailmanlaajuisen tietoverkkojen verkosto. Se on tietoverkko, jossa eri toimijoiden, kuten yksityisten, julkisten, yritysten sekä valtiohallintojen tietoverkot kytkeytyvät toisiinsa. Korkeakoulut, yritykset ja media-alan toimijat olivat ensimmäisiä, jotka käyttivät internetiä 1980-luvulla. Internetin kaupallistuminen alkoi 1990-luvulla. (Leiner, 1997.)

Esineet (Things)

Esineellä tarkoitetaan yksilöitävissä olevaa laitetta tai sensoria, joka pystyy aistimaan ympäristössään tapahtuvia muutoksia, keräämään ja prosessoimaan dataa sekä kommunikoimaan muiden laitteiden kanssa. Esineissä integroidut sensorit mahdollistavat edellä mainittujen muutosten tunnistamista, seurantaan sekä mittaamista. Sensorit toimivat joko yksitellen tai sulautetuissa järjestelmissä. Tyypillisesti käytössä olevat sensorit ovat lämpötila-anturit, liiketunnistimet, gyroskoopit, valotunnistimet ja äänitason sensorit, mutta on olemassa paljon muitakin sensoreita, jotka sopivat eri tarkoituksiin. (Inventrom, 2014.)

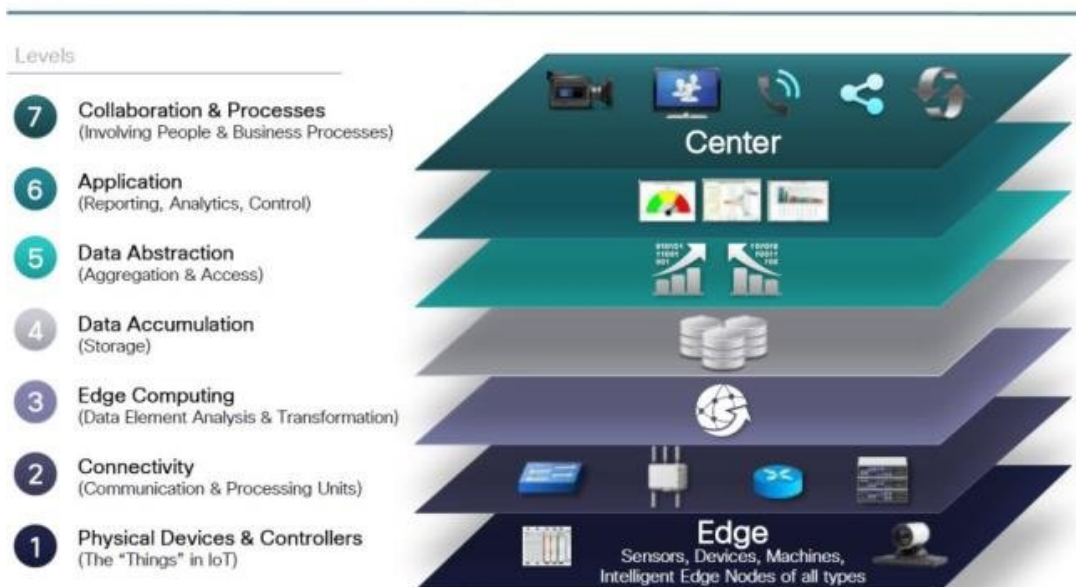
Arkkitehtuuri

Tietoliikennearkkitehtuureja ja -protokollia mallinnetaan kerrosrakenteisilla malleilla (esim. OSI-malli), joissa tiedon abstraktiotaso kasvaa kerrokselta (engl. layer) ylemmälle siirryttäessä. Kuva 1 esittää IoT- arkkitehtuuria, joka kutsutaan nimellä IoT World Forum Reference Model. IoT-mallin ensimmäinen taso käsittelee fyysisiä laitteita. Sensorit ja kontrollerit eli esineet ovat mallin ensimmäisellä tasolla. Niiden tehtävä on kerätä dataa ympäristöstä, lähettää ja vastaanottaa tietoja sekä suorittaa mahdollisia käskyjä ja operaatioita. Seuraavalla tasolla käsitellään esineiden liitettävyyttä ja kommunikaatiota. Tällä kerroksella määritellään, millä menetelmillä ja standardeilla liitettävyyttä toteutetaan. Kolmas taso käsittelee edge-laskentaa, jonka tarkoituksena on

prosessoida kerättyä dataa mahdollisimman lähellä datanlähdettä toiminnan nopeuden sekä sujuvuuden tehostamiseksi. Tässä vaiheessa priorisoidaan, suodatetaan sekä muutetaan kerättyä dataa sopivaksi muodoksi seuraavia tasoja varten. Seuraavalla tasolla kerättyä ja prosessoitua dataa tallennetaan jatkotoimenpiteitä varten. Tässä vaiheessa määritellään tallennusmuoto. Vaihtoehtoina ovat väliaikaiset muistit reaaliaikaista käsittelyä varten tai vaihtoehtoisesti tallennus massamuisteille. Seuraava vaihe eli ”Data Abstraction”-vaihe valmistaa kerättyä dataa erilaisiin applikaatioihin, sovelluksiin sekä tilastollisiin tarkoituksiin. Tässä vaiheesta monesta eri lähteestä ja eri formaateilla kerättyä dataa yhdenmukaistetaan ja viimeistellään tietoturvallisuutta parantavien parametrien osalta.(Sukanya, 2015.)

Kun kaikki edelliset vaiheet on suoritettu, voidaan tietoja siirtää applikaatioon, jonka avulla kaikkia toimenpiteitä ohjataan. Viimeisellä kerroksella tarkastellaan IoT:tä sidosryhmien eli teollisuuden, liiketalouden, sekä kuluttajien näkökulmasta. Opinnäytetyön kehittämissuunnitelmaa rajataan tässä vaiheessa referenssimallin kuudenteen kerrokseen asti. Mahdolliset seitsemänten kerrokseen liittyvät yhteistyöt sekä liiketoimintamallit jätetään jatkotoimenpiteeksi.

IoT World Forum Reference Model



Kuva 1. IoT world forum reference model (CiSCo, 2014).

Standardit

Esineiden internetin liittyviä standardeja ja protokolleja on lukuisia. Tässä osiossa käsitellään joitakin IoT:n merkittävimpiä ja käytetyimpiä protokollia ja standardeja. MQTT (Message Queue Telemetry Transport) on yksinkertaisten datavirtojen lähettämiseen tarkoitettu protokolla. Se on myös yhteensopiva laitteiden kanssa, jolla on pienempi muisti ja alhaisempi energiankulutus. DDS (Data Distribution Service) on reaaliaikaiseen M2M (Machine to Machine) kommunikaatioon liittyvä standardi. Se koostuu kahdesta osasta: DCPS:stä, joka vie informaation vastaanottajalle sekä DLRL:sta, joka tarjoaa rajapinnan välittäjälle. Zigbee on suosittu langaton verkko, joka mahdollistaa esineiden välisen kommunikoinnin universaalien kielen avulla. (Sakovich, 2018.)

Esineiden Internetin käyttökohteet

Älykoodit ovat yksi esineiden internetin tärkeimmistä ja eniten käytössä olevista käyttökohteista (Woodford, 2019). Viime vuosien teknologiakehitys on lisännyt IoT-laitteiden ja palveluiden tarjontaa sekä alentanut hintoja, minkä seurauksena laitteiden hankkimisen ja käyttämisen kynnys on merkittävästi alentunut. Älykodissa tyypillisesti käytössä olevat IoT-laitteet ovat termostaatit, kodinkoneet, lämpötilan seurantalaitteet, hälytysjärjestelmät, lukot ja valojärjestelmät. Myös puettavat IoT-laitteet ovat yleistyneet kovalla vauhdilla. Suosittuja puettavia laitteita ovat älykellot, aktiivisuusrannekkeet sekä erilaiset vaatteet, jotka sisältävät jonkin sensorin kehon analyysia varten.

Toinen hyvin nopeasti yleistynyt IoT:n käyttökohde on erilaiset kulkuneuvot. Tällä hetkellä suurin osa uusista autoista on yhdistetty internetiin. IoT:n avulla voidaan hoitaa auton vikadiagnostiikkaa ja huoltoa, lisätä käyttömukavuutta, parantaa turvallisuutta ja etähallita ajoneuvoa. Viimeisenä ja ehkä suurimpana hankkeena ovat älykaupungit. IoT:n avulla voidaan tasapainottaa suurkaupunkien liikenneongelmia, parantaa turvallisuutta, elämäntasoa ja ilmanlaatua. (Meola, 2016.)

Älykoti

Yleisellä tasolla älykodilla tarkoitetaan taloa, joka sisältää laitteita ja järjestelmiä, jotka helpottavat asukkaidensa elämää, parantavat turvallisuutta, optimoivat energiankulutusta sekä automatisoivat osan kodin toiminnoista. Älykoti on kokonaisuus, joka sisältää monia sensoreita, ohjaimia, releitä ja mittareita sekä järjestelmäkokonaisuuksia niiden ohjaamiseen. Älykodin tarkoitus on kerätä dataa asukkaiden käyttäytymisestä ja tottumuksista ja tehdä niiden pohjalta päätöksiä. Teknologian kehitys, tietoliikenneverkkojen edistys ja siirto-nopeuden kasvu sekä vastaavasti laitekustannusten lasku ovat edesauttaneet älykotien suosiota. (Karilahti, 2017.)

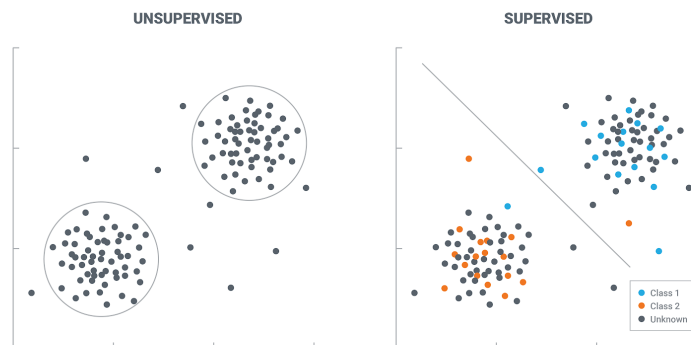
Ensimmäinen nykymallinen älykoti nimeltään 'House of Future' esiteltiin vuonna 1989, Chriet Titulaerin projektissa. Projektin tavoitteena oli esittää malli tulevaisuuden talosta sekä vuorovaikutuksesta ihmisen ja koneen välillä (van Rijmenam, 2014). Tyypillinen älykoti sisältää valaisimia, pistokkeita, termostaatteja ja mekanismeja, jotka ovat helposti etäohjattavissa eri paikoista älylaitteiden avulla. Asukkaiden ei enää tarvitse olla huolissaan valojen sammuttamisesta tai lämpötilasta. Edellä mainitut asiat sekä muitakin toimintoja voidaan suorittaa paikasta ja ajasta riippumatta älykotijärjestelmien avulla. Nykyisissä älykodeissa monet toiminnot perustuvat tekoälyyn. Tekoälyn avulla järjestelmä oppii asukkaiden elämäntapoja ja rutiineja ja automatisoi sitä kautta toimintoja. Esimerkkinä voidaan mainita valojen ja lämpötilan säädöt asukkaiden läsnä- ja poissaolojen aikana. Älykotijärjestelmät ovat integroitavissa jo olemassa oleviin perinteisiin järjestelmiin ja sähkökeskuksiin, jonka avulla voidaan optimoida talon energiankulutusta. (Stevens, 2018.)

2.2 Koneoppiminen

Koneoppimisen termi perustuu kahteen asiaan: koneseen, joka tässä tapauksessa tarkoittaa tietokonetta tai tietokoneohjelmaa sekä oppimiseen, jolla tarkoitetaan kykyä oppia esimerkkien avulla. Koneoppimisen näkökulmasta ohjelmoijan ei tarvitse luoda sääntöjä tai algoritmeja jokaista tilannetta varten,

vaan kone oppii toimimaan itsenäisesti pohjatietojen, tapahtumien sekä esimerkkien avulla. Koneoppimisella on rajattomasti mahdollisuuksia ja käyttökohteita. Yleisimmät ja tunnetuimmat käyttökohteet, jotka todennäköisesti tunnemme, ovat hakukoneet, autonomiset autot sekä tietokonepelit. Koneoppimisen mahdollistava ratkaiseva ydinasia on datalähde. Dataa on tällä hetkellä rajattomasti saatavilla eri lähteistä, esimerkiksi organisaatioiden tietokannoista, kuten terveydenhuollosta, pankeista, teollisuudesta ja sosiaalisesta mediasta.(Roos, 2018)

Koneoppiminen jaetaan kahteen tärkeään pääkategoriaan, ohjattuun ja ohjaamattomaan oppimiseen sekä näiden lisäksi vahvistusoppimiseen ja puoli-ohjattuun koneoppimiseen. Ohjattu oppiminen perustuu algoritmeihin, joiden alkutilanteen syöttödata sekä lopputulos ovat opetusvaiheessa tiedossa eli syöte on luokiteltua. Kyseisen opetusdatan ja esimerkkien avulla kone oppii luokittelemaan uusia syötteitä. Metodien onnistuminen ja oikeiden tulosten saavuttaminen on täysin riippuvainen opetusaineiston tarkkuudesta ja määrästä. Ohjaamattomassa oppimisessä tilanne on erilainen, kun oppimisdataa sekä syöttöpareja ei ole. Ohjaamaton oppiminen pyrkii löytämään piilossa olevia ja huomaamattomia yhteyksiä datasyötteiden välillä ja pyrkii rakentamaan toimivan mallin niistä. (Joutsijoki, 2017.) Kuvassa 2 havainnollistetaan kahden ohjatun ja ohjaamattoman oppimisen menetelmää, missä ohjatussa oppimisen menetelmässä syöte on luokiteltu erilaisilla väreillä.



Kuva 2. Ohjattu- ja ohjaamaton oppiminen (Lawtomated, 2019).

2.2.1 Keinotekoiset neuroverkot

Keinotekoisissa neuroverkoissa (ANN) pyritään ottamaan mallia ihmisen aivoista ja luonnollisista neuroverkoista. Neuroverkot koostuvat joukosta keinotekoisia neuroneita, jotka ovat yhteyksissä toisiinsa liitosten avulla. Neuronit koostuvat kolmesta osiosta: Synapsi on kahden neuronin välinen kytkentä, jolle on määritelty painoarvo. Summaaja laskee kaikkia syöttösynapsista tulleita tulosignaaleja yhteen ja viimeisenä on aktivaatiofunktio, joka määrittelee, miten neuronin tulos lasketaan. Tieto käsitellään ja lähetetään solmusta toiseen. Malli koostuu yhdestä syöttökerroksesta, yhdestä tai monesta piilokerroksesta sekä lopuksi tuloskerroksesta. Piilokerrokset sijoittuvat syöttökerroksen ja tuloskerroksen välille, jossa neuronit vastaanottavat painotettuja arvoja ja tuottavat ulostuloarvon aktivaatiofunktion avulla. Mallin opettaminen perustuu yhteyksien löytämiseen opetusdatan alkioden välillä. Tässä vaiheessa opetusdatan merkitys korostuu, koska se on edellytyksenä realististen tulosten saamiseksi. Mallin tavoitteena on tarkastella alku- ja lopputilanteita annettujen esimerkkien eli opetusdatan avulla ja määrittellä oikeanlaiset painoarvot eri reiteille, kunnes arvioitu tulos on mahdollisimman lähellä realistista tulosta. (Roos, 2018.)

Yksinkertainen esimerkki keinotekoisesta neuroverkosta on Multi Layer Perceptron (MLP). Sen avulla voidaan mallintaa epälineaarista suhdetta tulo- ja lähtömuuttujien välillä useiden toisiinsa kytkettyjen prosessointielementtien läpi. MLP-malli koostuu kolmesta pääkerroksesta, jotka ovat syöttökerros, piilokerros sekä ulostulokerros. Jokaiseen neuronien väliseen yhteyteen liittyy tietty painoarvo. MLP-malli koulutetaan säätämällä painoja virheen minimoimiseksi. Viimeisen kerroksen virhe voidaan laskea seuraavasti:

$$E = (y_i - \hat{y}_i)^2$$

Kaava 1. Kahden arvon, ennustetun ja todellisen havainnon erotuksen neliö.

Missä y edustaa todellista tai mitattua ulostuloa ja vastaavasti \hat{y} ennustettua ulostuloa.

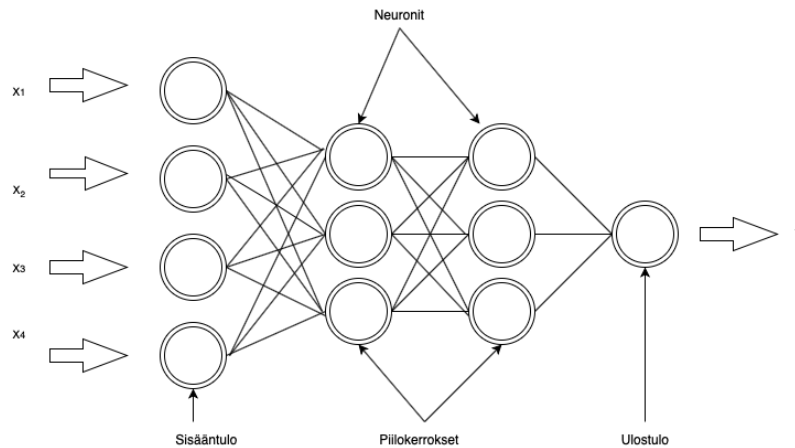
Mallin opetusvaiheen aikana jokaisen solmun j paino i kerroksella päivitetään alla olevan funktion avulla.

$$w_{ji+1} = w_{ji} - \alpha \left(\frac{\partial e}{\partial w_{ji}} \right)$$

Kaava 2. Kaltevuuden laskeutuminen (Guo & Gelfand, 1991).

Missä w_{ji} edustaa j solmun painoarvoa i kerroksella ja α oppimisen nopeutta. Osittaisderivaatta $\left(\frac{\partial e}{\partial w_{ji}} \right)$ määritellään takaisinkytkentämenetelmällä.

Kuvassa 3 esitetty monikerroksinen Perceptron -malli, koostuu yhdestä syöttökerroksesta, kahdesta piilokerroksesta sekä tuloskerroksesta



Kuva 3. Monikerroksinen Perceptron-malli.

2.2.2 Pitkät lyhytaikaisen muistin verkot

Pitkät lyhytaikaisen muistin verkot (engl. Long short-term memory, LSTM) on eräänlainen takaisinkytkettyjen neuroverkkojen (RNNs) tyyppi. RNNs:ssä on takaisinkytkentäyhteyksiä, minkä vuoksi sitä hyödynnetään sekvenssien ja ajallisten riippuvuuksien mallintamiseen. LSTM:n aktivointitoiminto laskee ajallisen riippuvuuden ottamalla huomioon aikaisemman ajan ($t - 1$). LSTM sisältää sisääntuloportin i , unohda-portin f , ulostuloportin o sekä solun tilan c . Yleensä kolme edellä mainittua porttia ohjaa tiedonkulkua. (Hochreiter & Schmidhuber, 1997)

Nykyisellä ajanhetkellä t sisääntulo on x_t , piilokerroksen ulostulo on h_t ja sen edellinen ulostulo on h_{t-1} , solun ulostulon tila on \hat{c}_t ja edellisen solun ulostulon tila on c_{t-1} .

c_t ja h_t lasketaan alla olevien kaavojen avulla järjestyksessä:

$$i_t = \sigma(w_1^i \cdot x_t + w_h^i \cdot h_{t-1} + b_i)$$

$$f_t = \sigma(w_1^f \cdot x_t + w_h^f \cdot h_{t-1} + b_f)$$

$$o_t = \sigma(w_1^o \cdot x_t + w_h^o \cdot h_{t-1} + b_o)$$

$$\hat{c}_t = \tanh(w_1^c \cdot x_t + w_h^c \cdot h_{t-1} + b_c)$$

Kaava 3. Sisääntuloportin, unohda-portin, ulostuloportin, solun tilan laskentakaavat (Hochreiter & Schmidhuber, 1997).

Missä $w_1^i, w_1^f, w_1^o, w_1^c$ ovat painojen matriisit, jotka yhdistävät x_t kolmeen porttiin, sekä solun sisääntuloon. Vastaavasti $w_h^i, w_h^f, w_h^o, w_h^c$ ovat painojen matriisit, jotka yhdistävät h_{t-1} kolmeen porttiin, sekä solun sisääntuloon. b_i, b_f, b_o, b_c edustavat bias-arvoa. Vastaavasti σ ja \tanh edustavat sigmoidin ja tangentin funktioita.

Toiseksi solun ulostulotilaa mitataan seuraavasti:

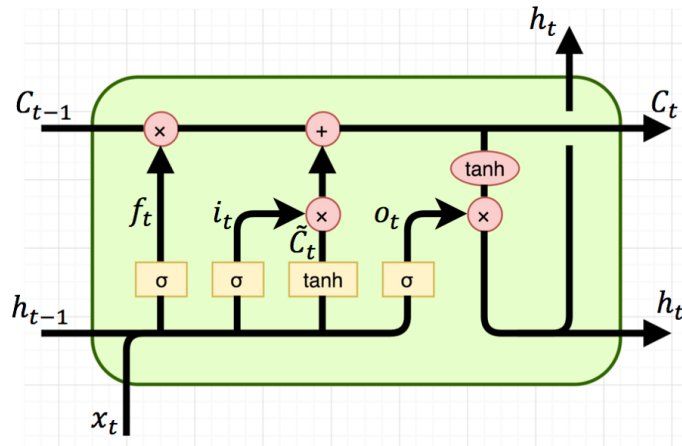
$$c_t = i_t \times \hat{c}_t + f_t \times c_{t-1}$$

Lopuksi piilokerroksen ulostulo lasketaan seuraavasti:

$$h_t = o_t \times \tanh(c_t)$$

Kaava 4. Piilokerroksen ulostulon laskentakaava (Hochreiter & Schmidhuber, 1997).

Kuvassa 4 on esitetty edellä tarkastellun pitkät lyhytaikaisen muistin verkot mallin rakenne.



Kuva 4. Pitkät lyhytaikaisen muistin verkkojen rakenne (Xuan-Hien& Hung, 2019).

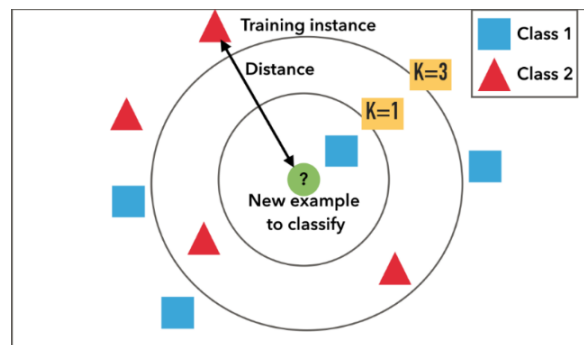
2.2.3 K-lähimmän naapuri

K-lähimmän naapurin algoritmi (engl. k-nearest neighbor, KNN) on algoritmi datan luokittelua varten. Algoritmia voidaan käyttää sekä luokitteluun että regressioennustamiseen liittyvissä ongelmissa. Kun algoritmia käytetään luokitteluun, jossa algoritmi ottaa tietyn määrään (riippuen k :n arvosta) lähellä olevia naapureita käsittelyyn, tuloksesta voidaan päätellä, mihin luokkaan tietty syöte voi kuulua. Algoritmin avulla voidaan tarkastella erityisesti reaali maailmassa olevaa dataa, joka ei seuraa mitään lineaarista erotettavuutta tai tasaista jakautumista. K-lähimmän naapurin algoritmi kuuluu ns. laiskan oppimisen algoritmeihin, jotka eivät tarvitse etukäteistä opetusta, toisin kuin esimerkiksi lineaarisen regression algoritmit. Kyseinen ominaisuus mahdollistaa uuden datan käsittelyn viiveettä. Algoritmi voidaan toteuttaa vain kahdella parametrilla, jotka ovat k :n arvo ja etäisyyden funktio. Haittapuolena on algoritmin toimintakyvyn heikkeneminen suurimmissa datakokonaisuuksissa etäisyyden ja luokitteluun liittyvien haasteiden takia. (Hochreiter & Schmidhuber, 1997.)

K-lähimmän naapurin algoritmi perustuu yksinkertaisesti uuden datapisteen ja esimerkkijoukkojen välisen etäisyyden tarkasteluun. Etäisyys voi olla joko euklidinen tai Manhattan-etäisyys. Luokitusta määritellään tarkastelemalla, millä

dataluokitteluilla on enemmän jäseniä K :n määritellyllä alueella. Oletetaan datasetti, jolla on m kappaletta näytteitä. Jokaiselle näytteelle x_i on määritelty sisääntuloarvoja n sekä ulostulo y_i eli $x_i = x_{i1}, x_{i2}, \dots, x_{in}, y_i$. Tavoitteena on oppia funktio $f: x \rightarrow y$, joka tunnetaan regressiofunktiona, joka mallintaa sisääntulomuuttujan ja ulostulomuuttujan välistä suhdetta. KNN-regressio estimoii funktion ottamalla opetusdatan paikallisen keskiarvon. Paikallisuus määritellään arviointinäytteen läheisyydessä olevien näytteiden k perusteella. Koska KNN algoritmin suorituskyky on hyvin riippuvainen k parametrissa, on olennaista löytää optimaalinen k . Suurempi k arvo vähentää kohinaa ja minimoi ennustushäviöitä. Toisaalta pienempi k arvo mahdollistaa vain yksinkertaisemman toteutuksen. (Mitchell, 1997, s. 30-35.)

Kuvassa 5 esitetyssä K -lähimmän naapurin algoritmin mallissa uusi testinäyte luokitellaan joko sinisten neliöiden ensimmäiseen luokkaan tai punaisten kolmioiden toiseen luokkaan riippuen k :n arvosta.



Kuva 5. K -lähimmän naapurin luokittelumalli (Raghunath, 2019).

Ristiin validoinnin avulla voidaan arvioida luokittelun tarkkuutta ja oikeellisuutta eri k arvoilla. Kaksi yleisintä ristiin validoinnin menetelmää ovat k -kertainen ristiin validointi sekä yksi pois -ristiin validointi (LOOCV). K -kertaisessa ristiin validointimenetelmässä datasettiä jaetaan k saman suuruiseen osaan satunnaisesti. Jaossa on otettava huomioon aineiston koko ja laajuus sekä luokkien suuruus. Tämän jälkeen suoritetaan k kertaa kierroksia, jossa aineisto jaetaan jokaisessa kierroksessa vuorotellen kahteen testi- ja opetusjoukkoon, niin, että jokainen havainto olisi $k - 1$ kertaa jokaisella kierroksella. Jokaisen kierroksen jälkeen lasketaan testijoukkueiden poikkeavuuksia eli ennustusvirheitä ja lasketaan

kaikkien kierrosten ennustevirheistä keskiarvo. Kierrosten lukumäärä on ratkaiseva tekijä tarkkojen arvojen aikaansaamiseksi. Ei ole olemassa tarkkaa ohjeistusta, vaan on käytävä kaikki mahdollisuudet läpi ja valittava sen jälkeen oikea määrä kierroksia. Tämän prosessin jälkeen valitaan paras parametriarvo, jolla määritellään muodostettava malli.(Joutsijoki, 2017.)

Ristiin validointi (LOOCV) on tietynlainen K -kertainen ristiin validointi, jossa k edustaa datasetissä olevien näytteiden lukumäärää. Projektissa olevien näytteiden määrästä johtuen tässä työssä on keskitytty (LOOCV) menetelmään.

3 PROTOTYYPIN SUUNNITTELU JA RAKENTAMINEN

3.1 Prototyypin alustan valinta

Prototyypin rakentaminen aloitettiin vertailemalla kahta tunnettua prototyyppialustaa Arduino -mikrokontrolleri sekä Raspberry Pi. Vertailussa tarkasteltiin alustojen ominaisuuksia, arkkitehtuuria, käyttötarkoitusta, lisäosien saatavuutta sekä yhteensopivuutta projektin tavoitteiden kanssa.

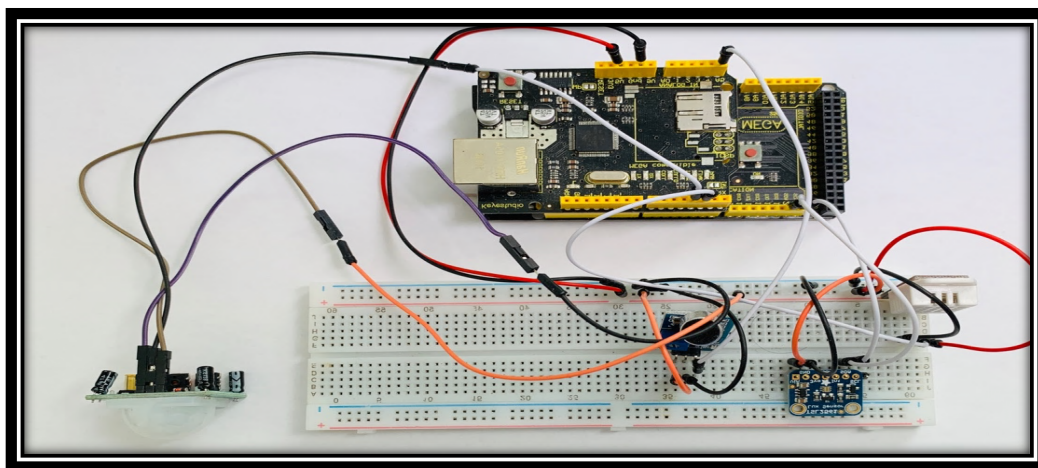
Raspberry Pi on yhden piirilevyn tietokone, joka julkaistiin ensimmäistä kertaa vuonna 2012 opetus- ja koulutustarkoituksiin. Raspberry Pin konseptin ideana on hyödyntää tietokonetta, joka sisältää kaikki tarvittavat portit, pinnit ja osat yhdellä ainoalla piirilevyllä. Raspberry Pi on kehittynyt vuosien ajan ja kehitys on hyvin havaittavissa: viimeisessä versiossa (Pi4) on 1.5 GHz quad-core 64-bit ARM prosessori, 4 Gt keskusmuistia, USB-C ja micro-HDMI-liitin. Toinen vaihtoehto oli ensimmäistä kertaa vuonna 2008 julkaistu Arduino -mikrokontrolleri. Arduino oli vuonna 2005 aloitetun opetuksen kehittämisprojektin tulos. Projektin tavoitteena oli rakentaa edullinen, yksinkertainen sekä helposti laajennettava alusta, jota voisivat hyödyntää sekä alan ammattilaiset että harrastajat. Arduinon suosituin alusta Arduino Uno sisältää 16Mhz kellon, 32 KB keskusmuistia, sekä 14 GPIO pinnejä. (Cicuito, 2018.)

Raspberry Pi on suorituskykyisempi kuin kilpailijansa Arduino. Asia herättää kysymyksen Arduinon menestyksestä markkinoilla näin pienillä resursseilla. Vastaus riippuu kuitenkin hyvin projektin tyypistä ja tavoitteista. Yleisellä tasolla Arduino on tarkoitettu laitteistoperusteisiin projekteihin, joissa esimerkiksi kerätään dataa eri sensoreiden avulla, kun taas Raspberry Pi on sopivampi ohjelmistoperusteisiin projekteihin, joissa vaaditaan datan prosessointitehoa. Projektin luonteesta riippuen on mahdollistaa integroida kumpaakin alustaa täydentämään toisiaan. Edellä mainittujen asioiden perusteella valittiin Arduino prototyypin toteuttamiseen.

3.2 Arduino

Arduino on mikrokontrolleri, joka perustuu avoimeen laitteistoon. Arduino sisältää digitaalisia kytkentäpinnejä, jotka käytetään joko sisään- tai ulostulopinneinä. Digitaalisilla kytkentäpinneillä on joko 0-voltin jännite eli "Low"-tila tai +5 voltin jännite eli "High"-tila. Arduino sisältää myös analogisia kytkentäpinnejä, jotka lukevat jännitteitä 0-5 voltin välillä sekä jännitelitaintää varten 5V sekä maapinnejä. Mikrokontrolleri ohjelmoidaan USB-liittimen kautta. Arduino toimii 6 - 15 voltin ja 2,1 mm virtaliittimen virransyötöllä. Arduino-mikrokontrollereita on eri malleja ja kokoja riippuen niiden pinnien määrästä, keskusmuistista ja EEPROM-tyypistä. Kuvassa 1 on projektissa hyödynnetty Arduino Mega, jolla on 54 digitaalista ja 16 analogista pinniä ja joka perustuu ATmega2560 piiriin.

Mikrokontrolleri ohjelmoidaan Arduinon ohjelmointiympäristön avulla, joka on nimeltään Arduino (IDE). Arduino IDE tukee C/C++-ohjelmointikieliä, mutta koska kyseessä on mikrokontrollerin ohjelmointi, vaadittavat kirjastot sekä ohjelmoinnin rakenne poikkeavat jonkin verran tyyppillisestä ohjelmoinnista. Pinnien määrittely, analogisten arvojen lukeminen, datan siirto ja vastaanotto sekä datan lähettäminen pilveen ovat tärkeitä asioita Arduinon ohjelmoinnissa.




Kuva 6. Arduino Mega.

3.3 Sensori

Sensori (anturi) on laite, joka mittaa, tunnistaa ja reagoi tiettytyypisiin ympäristössä oleviin fysikaalisiin suureisiin ja niiden muutoksiin. Mitattava suure voi olla valo, lämpötila, kosteus, värinä jne. Sensoreista saatuja signaaleja välitetään erillisille järjestelmille, kuten automaatio- tai mittausjärjestelmiin jatkotoimenpiteitä varten. Sensoreista saadut signaalit ovat yleensä analogisia, mutta niitä voidaan muuttaa digitaliseen muotoon joko itse anturissa tai erillisen muuntimen avulla. Sensoreita on eri tarkoituksiin erilaisilla ominaisuuksilla ja tarkkuuksilla. Teknologian kehityksen seurauksena, sensoreiden koko ja hinta ovat laskeneet merkittävästi sekä samalla niiden tarkkuus ja toimivuus parantunut. (Rouse, 2013.)

Taulukossa 1 esitellään projektissa hyödynnettyjä sensoreita, joiden avulla tarvittavaa dataa kerätään.

Taulukko 1. Projektissa käytetyt sensorit, niiden ominaisuudet ja sijoittelu.

	DHT22	LM386	TSL2591	HC-SR501
				
Mittattava suure	Lämpötila/kosteus	Äänen amplitudi	Valaistusvoimakkuus	Infrapunasäteilyn taso
Mittausalue	-40 °C – 125 °C / 0-100% RH +0,5 °C /+2% RH	-56- 72dBA	0.1 - 88,000 lx	700nm-1mm 7-10 m
Jännite	3 – 5 V	5 V	3 – 5 V	4,5 – 12 V
Näytteenottoväli	5s	5s	5s	5s
Sijoituspaikka	Olohuone	Oleskelutila	Oleskelutila	Salin katto
Vaikutus pilottijärjestelmässä	Lämpötilamuutosten vaikutus lämmitysjärjestelmässä	Läsnäolon seuranta, sekä aktiviteetti	Sisävalaistus/ rutiinit/läsnäolo	Läsnä/ poissa olo/rutiinit

3.4 Prototyypin rakentaminen

Prototyypin rakentamista varten tilattiin tarvittavat komponentit, työkalut ja tarvikkeet sekä rakennettiin pienenmuotoinen kotilaboratorio.

Tutkimussuunnitelman mukaisesti testitalon sisätiloista piti kerätä erilaisia mittaustietoja. Ensimmäisessä vaiheessa valittiin oikeanlaiset sensorit tarvittavan datan keräämiseen. Tähän tarkoitukseen käytettiin seuraavia sensoreita: 1. kosteus/lämpötilasensori, jonka avulla saadaan reaaliaikaista dataa talon sisälämpötilasta sekä kosteusprosentista, 2. kirkkauden/valon määrän mittaamiseen valosensori, jonka avulla voidaan tarkastella valoisuuden tasoa. 3. äänisensori, jonka käyttötarkoitus projektissa oli seurata asukkaiden läsnä- tai poissaoloa äänentason perusteella. Edelliseen tarkoitukseen rinnalla käytettiin myös PIR-sensoria mahdollisen talossa tapahtuvan liikkeen havainnointia varten.

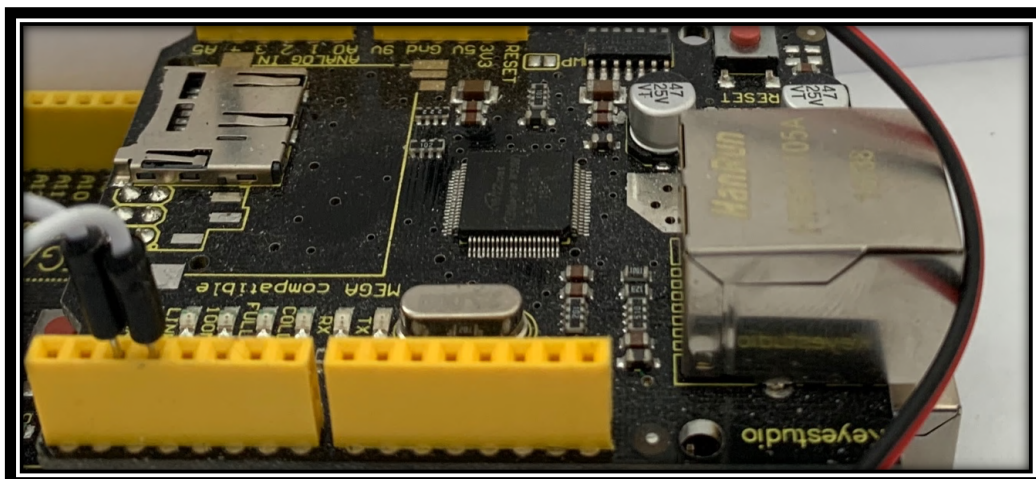
3.4.1 Sensoreiden kytkentä

Sensoreiden suunnittelun, yhteensopivuuden varmistamisen sekä valinnan jälkeen oli vuorossa kytkentävaihe. Sensoreiden kytkentä mikrokontrolleriin toteutettiin koekytkentälevyä hyödyntämällä. Koekytkentälevyn avulla voidaan kytkeä sensoreita ja komponentteja mikrokontrolleriin ilman juottamista. Koekytkentälevyn pystyriivit ovat virransyöttöä ja maata varten ja vaakariivit johdinta, johon komponentteja sijoitetaan. Kytkeminen tapahtuu tähän tarkoitetuilla johdoilla.

Ensimmäisessä vaiheessa koekytkentälevyn pystyriiviin kytkettiin Arduinon +5V ja GND- nastat. Seuraavaksi valittiin oikeanlaiset nastat jokaiselle sensorille riippuen siitä, onko signaali digitaalinen vai analoginen. Jokainen sensori kytkettiin koekytkentälevyllä olevaan pystyriivin virransyöttöä varten ja mikrokontrolleriin datan välittämistä varten.

3.4.2 Arduino Ethernet Shieldin asentaminen

Arduino Ethernet Shield mahdollistaa Arduinon kytkennän internetiin RJ45-kaapelilla. Arduino Ethernet Shield tarjoaa IP-osoitetta TCP- ja UDP:n -protokollilla. Se on tehokas ja kätevä avoimeen lähdekoodiin perustuva komponentti, joka käyttää ainoastaan yhtä pinniä mikrokontrollerista. Se toimii 5V jännitteellä ja tarjoaa 10/100 Mb/s yhteysnopeutta. (Arduino, 2019).



Kuva 7. Arduino Ethernet Shield.

3.5 Testiympäristö

Projekti toteutettiin vuonna 2017 rakennetussa omakotitalossa. Talon Kerrosala on 156m² ja se sijaitsee Varsinais-Suomessa. Talon lämmitys toteutetaan sähkökäyttöisen NibeF470 poistoilmalämpöpumpun ja vesikiertoisen lämmitysjärjestelmän avulla, joka säädetään huonekohtaisilla termostaateilla sekä graafisella käyttöympäristöllä etänä. Poistoilmalämpöpumppu saa tarvittavan lämmitysenergiansa talosta poistettavasta lämmenteestä ilmasta. Vastaanotettua lämpöä hyödynnetään tuloilman, vesikiertoisen lämmitysjärjestelmän sekä veden lämmittämiseen. Vesikiertoisen lämmitysjärjestelmän kiertovettä lämmitetään myös NibeF470-poistoilmalämpöpumpun avulla.

Suomessa rakennuksen energiankulutus luokitellaan energiatehokkuuden vertailuluvun eli E-luvun avulla ($kwh_E/m^2/a$). Jokaiselle energiamuodolle määritellään kerroin. Rakennuksen energiankulutuksen tarkastelussa huomioidaan energiankulutukseen vaikuttavia seikkoja sekä määritellään rakennuksen tarvittavaa ostoenergiaa. Rakennuksessa tarvittava energiaa käytetään rakennuksen lämmitykseen, ilmanvaihtoon, käyttöveden lämmitykseen, valaistukseen ym. Rakennuksen kunto, materiaali ja tekniset ominaisuudet vaikuttavat energiankulutuksen tasoon. E-luvun laskennan lähtötietojen kartoituksessa tarkastellaan rakennuksen pinta-alaa, rakennuksen

vaippaa, ikkunoiden suuntausta, ilmanvaihto- ja lämmitysjärjestelmää, käyttöveden lämmitystapaa, jäähdytysjärjestelmää sekä rakennuksen lämpökuormaa. (Motiva,2018). Taulukossa 2 tarkastellaan projektissa käytetyn rakennuksen E-luvun laskennan lähtötietoja.

Taulukko 2. E-luvun laskennan lähtötiedot.

E-LUVUN LASKENNAN LÄHTÖTIEDOT				
Rakennuskohde				
Rakennuksen käyttötarkoitusluokka	Yhden asunnon talot			
Rakennuksen valmistumisvuosi	2017	Lämmitetty nettoala	140	m ²
Rakennusvaippa				
Ilmanvuotoluku q_{50}	0,4	m ³ /(h m ²)		
	A	U	U×A	Osuus lämpöhäviöistä
	m ²	W/(m ² K)	W/K	%
Ulkoseinät	90,9	0,17	15,5	17%
Yläpohja	139,7	0,08	11,2	12%
Alapohja	139,7	0,17	23,7	26%
Ikkunat	24,9	1,00	24,9	27%
Ulko-ovet	8,5	1,00	8,5	9%
Kylmäsiilit	-	-	8,6	9%
Ikkunat ilmansuunnittain				
	A	U	g_{kohtisuora}-arvo	
	m ²	W/(m ² K)	-	
Pohjoinen	7,2	1,00	0,56	
Koillinen				
Itä	1,1	1,00	0,56	
Kaakko				
Etelä	15,8	1,00	0,56	
Lounas				
Länsi	0,9	1,00	0,56	
Luode				
Ilmanvaihtojärjestelmä				
Ilmanvaihtojärjestelmän kuvaus:	Nilan EC9 (tulo+poisto)			
	Ilmavirta tulo/poisto	Järjestelmän SFP-luku	LTO:n lämpötilasuhde	Jäätymisenesto
	(m ³ /s) / (m ³ /s)	kW / (m ³ /s)	-	°C
Pääilmanvaihtokoneet	0,06 / 0,06	1,58	80%	3,0
Erillispoistot	0,00 / 0,00	0,00	-	-
Ilmanvaihtojärjestelmä	0,06 / 0,06	1,58	-	-
Rakennuksen ilmanvaihtojärjestelmän LTO:n vuosihyötysuhde:	73%			
Lämmitysjärjestelmä				
Lämmitysjärjestelmän kuvaus:	Poistoilmalämpöpumppu, vesikiertoinen lattialämmitys + varaava tulisija /			
	Tuoton hyötysuhde	Jaon ja luovutuksen hyötysuhde	Lämpökerroin¹	Apulaitteiden sähkönkäyttö²
	-	-	-	kWh/(m ² vuosi)
Tilojen ja iv:n lämmitys		80%	3,2	2,5
Lämpimän käyttöveden valmistus		92%	3,2	0,0
¹ vuoden keskimääräinen lämpökerroin lämpöpumpulle				
² lämpöpumppujärjestelmissä voi sisältyä lämpöpumpun vuoden keskimääräiseen lämpökertoimeen				
	Määrä	Tuotto		
	kpl	kWh		
Varaava tulisija	1	2 000		
Ilmalämpöpumppu				
Jäähdytysjärjestelmä				
	Jäähdytyskauden painotettu kylmäkerroin			
Jäähdytysjärjestelmä	-			
Lämmin käyttövesi				
	Ominaiskulutus	Lämmitysenergian nettotarve		
	dm ³ /(m ² vuosi)	kWh/(m ² vuosi)		
Lämmin käyttövesi	518	30		
Sisäiset lämpökuormat eri käyttöasteilla				
	Käyttöaste	Henkilöt	Kuluttajalaitteet	Valaistus
	-	W/m ²	W/m ²	W/m ²
Henkilöt ja kuluttajalaitteet	60%	2,0	3,0	
Valaistus	10%			8,0

Talossa suoritetun tiiveysmittausraportin mukaan talon rakennusvaipan ilmanvuotoluku q_{50} on $0,4 \text{ m}^3 / \text{m}^2 \text{ h}$, joka kertoo talon korkeatasoisesta tiiveydestä. Talon energiatodistuksen mukaan talon energiatehokkuusluokka on C ja laskennallinen kokonaisenergian kulutus (E-luku) $146 \text{ kWh}_E / \text{m}^2 / \text{a}$. Energiatodistuksessa mainittu varaava tulisija on ollut projektin datankeräysvaiheessa pois käytöstä. Taulukossa 3 esitellään E-luvun laskennan tuloksia.

Taulukko 3. Rakennuksen energiatehokkuus.

E-LUVUN LASKENNAN TULOKSET				
Rakennuskohde				
Rakennuksen käyttötarkoitusluokka	Yhden asunnon talot			
Rakennuksen valmistumisvuosi	2017			
Lämmitetty nettoala, m^2	139,7			
E-luku, $\text{kWh}_E / (\text{m}^2 \text{vuosi})$	146			
E-luvun erittely				
Käytettävät energiamuodot	Laskettu ostoenergia kWh/vuosi	Energiamuodon kerroin	Energiamuodon kertoimella painotettu energiankulutus $\text{kWh}_E / \text{vuosi}$ $\text{kWh}_E / (\text{m}^2 \text{vuosi})$	
sähkö	11 052	1,7	18 788	135
Uusiutuva polttoaine (Puu)	3 125	0,5	1 563	12
YHTEENSÄ	14 177		20 351	146
Uusiutuva omavaraisenergia, hyödyksikäytetty osuus				
		kWh/vuosi	$\text{kWh}/(\text{m}^2 \text{vuosi})$	
Rakennuksen teknisten järjestelmien energiankulutus				
		Sähkö $\text{kWh}/(\text{m}^2 \text{vuosi})$	Lämpö $\text{kWh}/(\text{m}^2 \text{vuosi})$	Kaukojäähdytys $\text{kWh}/(\text{m}^2 \text{vuosi})$
Lämmitysjärjestelmä				
Tilojen lämmitys ¹		2,5	42,0	-
Tuloilman lämmitys		8,5	12,4	-
Lämpimän käyttöveden valmistus		0,0	34,0	-
Ilmanvaihtojärjestelmän sähköenergiankulutus		0,0	-	-
Jäähdytysjärjestelmä		0,0	0,0	0,0
Kuluttajalaitteet ja valaistus		22,8	-	-
YHTEENSÄ		34,0	89,0	0,0
¹ Ilmanvaihdon tuloilman lämpeneminen tilassa ja korvausilman lämmitys kuuluu tilojen lämmitykseen				
Energian nettotarve				
		kWh/vuosi	$\text{kWh}/(\text{m}^2 \text{vuosi})$	
Tilojen lämmitys ²		6 695	48	
Ilmanvaihdon lämmitys ³		6 732	49	
Lämpimän käyttöveden valmistus		4 200	31	
Jäähdytys		0	0	
² sisältää vuotoilman, korvausilman ja tuloilman lämpenemisen tilassa				
³ laskettu lämmöntalteenoton kanssa				
Lämpökuormat				
		kWh/vuosi	$\text{kWh}/(\text{m}^2 \text{vuosi})$	
Aurinko		5 654	41	
Henkilöt		1 469	11	
Kuluttajalaitteet		2 203	16	
Valaistus		979	8	
Lämpimän käyttöveden kierrosta ja varastoinnin häviöistä		92	1	
Laskentatyökalun nimi ja versio numero				
Laskentatyökalun nimi ja versio numero		www.laskentapalvelut.fi, versio 1.4 (7.12.2017)		

3.6 Sensoreiden sijainti talossa

Sensoreiden sijainti talossa on suunniteltu ja toteutettu niin, että sensoreista saadaan paras hyöty. Sensoreiden sijoittelussa huomioitiin mm. kerättävän datan luonne, toistuvuus, herkkyys sekä häiriöt. Sensorit sijoitettiin niiden kategorioiden mukaisesti, esimerkiksi asukkaiden läsnä/poissaoloa seuranneet sensorit, kuten liiketunnistin ja äänisensori sijoitettiin olohuoneeseen, jossa asukkaat viettävät eniten aikaa ja heidän aktiviteettiansa seurauksena aiheutuvaa ääni rekisteröityy parhaiten sensorille. Lämpötilan sekä kosteuden sensori asetettiin suojaiseen paikkaan, jossa saadaan mahdollisimmat realistiset ja vakaat arvot ilman häiriötä. Valaistusvoimakkuuden sensori sijoitettiin sitten, että se edustaisi parhaiten talon sisävalaistusta.

3.7 Ohjelmointi

Arduinon ohjelmointi tai yleensäkin mikrokontrollerien ohjelmointi on jonkin verran monimutkaisempaa kuin perinteisen sovelluksen ohjelmointi. Sensoreiden määrä, niiden yhteensopivuus sekä liittäminen yhteen kokonaisuuteen aiheuttivat haasteita projektissa. Toisaalta kaikista sensoreista ja komponenteista ei ollut datalehteä, joten ainoana mahdollisuutena oli kokeilla kaikki vaihtoehdot läpi ja löytää paras kytkentä-tai ohjelmointitapa.

Ohjelmoinnin alkuvaiheessa määriteltiin tarvittavia kirjastoja sensoreiden ominaisuuksista riippuen. Tämän jälkeen konfigurointiin ohjelmaa Ubidots-alustaa varten. Jokainen kommunikaatio prototyypin ja Ubidots-alustan välillä vaati tokenin, jonka tarkoituksena on varmentaa ja turvata tiedonsiirto laitteen ja alustan välillä. Tämän jälkeen määriteltiin seuraavia sensorin perustuvia muuttajia:

- Volume, äänen havaitsemiseen talossa
- Humidity, kosteudenprosentti talossa
- Temperature, talon sisäinen lämpötila
- Pir, liikettä talon sisällä

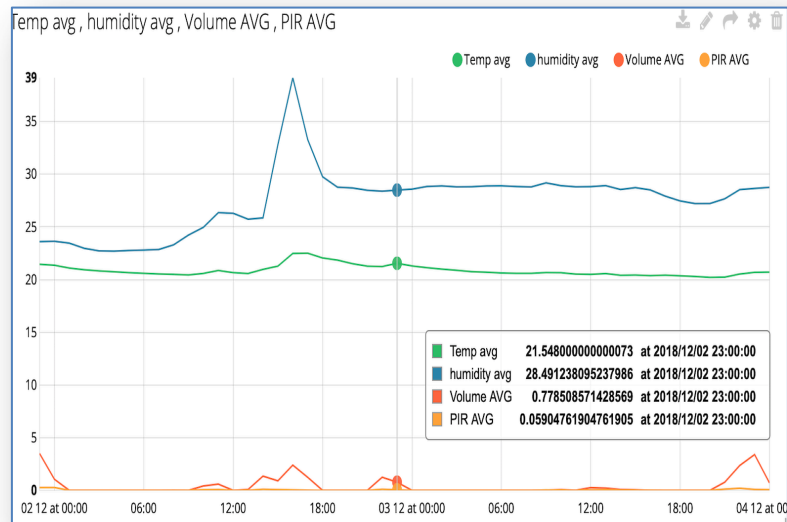
- Lux, valon määrä talon sisällä

Seuraavassa vaiheessa määriteltiin Ethernet Shieldin Mac-osoite, joka on verkkosovittimen ethernet-verkossa olevan uniikki tunnus. Tämän vaiheen jälkeen määritellään jokaiselle sensorille digitaalinen/analoginen sisääntulo riippuen sensorin tyypistä. Tämän jälkeen ohjelmoidaan jokaista sensoria varten silmukka ja tietty odotusaika (ms) ennen uutta kierrosta. Tarkemmat dataan liittyvät suodatukset ja normalisoinnit tapahtuvat Ubidots-alustalla.

3.8 Ubidots-alusta

Arduinosta lähtevän datan käsittelyä varten pilvessä on käytetty Ubidots-alustaa. Ubidots-alusta mahdollistaa sensoriarvojen siirtoa pilvipalveluun nopeasti ja sujuvasti. Sensoriarvot siirtyvät Arduinon ohjelmointialustan alkukäsittelyjen jälkeen suoraan Ubidots-ympäristöön analyysiä varten.

Ubidots-alusta luo jokaisen vastaanotetun sensoriarvon jälkeen datapisteen sekä siihen liittyvään aikaleiman. Jokainen datapiste voi sisältää jopa 16 numeroisen arvon, aikaleiman millisekunnissa sekä tarvittaessa ryhmän avainarvopareja esimerkiksi GPS- laitteiden leveys- ja pituuskoordinaatistossa. Näiden perusteella järjestelmä luo aikajanan tietojen analyysiä varten. Mikrokontrollerin ja Ubidots-järjestelmän kommunikaatio tapahtuu HTTP / MQTT / TCP / UDP-protokollalla. Kuviossa 2 on esimerkki Ubidots- järjestelmässä luodusta graafista, jossa on kuvaajat ajan funktiona sensoreista saatujen tietojen keskiarvojen perusteella.



Kuvio 1. Kahden vuorokauden sensoriarvot.

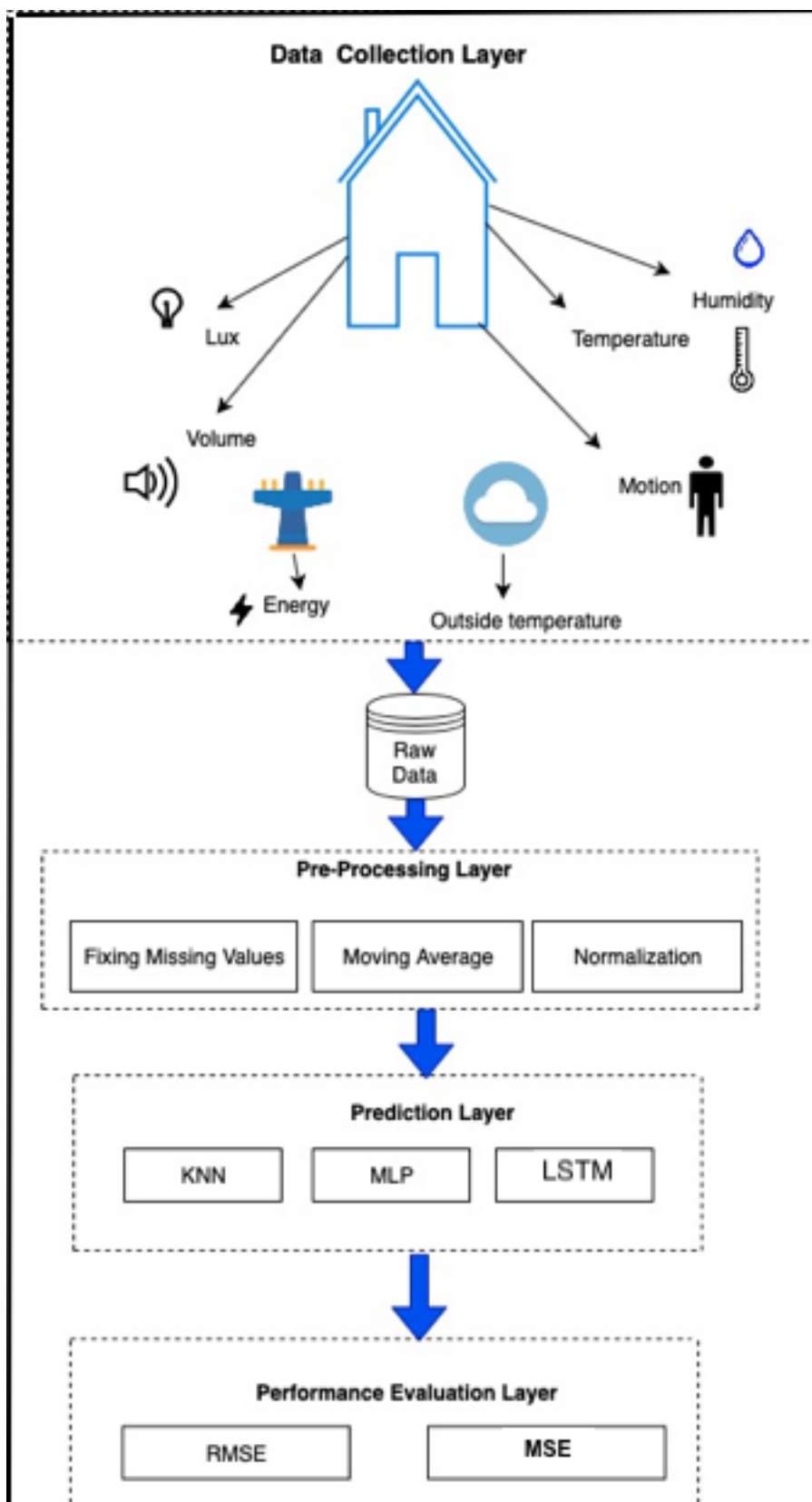
3.9 Energiankulutus

Projektin kannalta yksi merkittävimmistä datalähteistä oli talon energiankulutus. Talon pääasiallinen energianlähde on sähkö, joka on myös talon lämmitysjärjestelmän energianlähde. Talon energiankulutukseen liittyviä tietoja ja arvoja on saatu Carunan sähköjakelujärjestelmästä. Arvoja käytetään koneoppimisen liittyvään älykkään mallin opetusvaiheessa.

4 ÄLYKKÄÄN MALLIN TOTEUTUS

4.1 Energiankulutuksen ennustamisen prosessi

Energiankulutuksen ennustamisen prosessi koostuu eri vaiheista ja kerroksista. Ensimmäisessä kerroksessa sijaitsevat sensorit sekä ulkopuoliset datalähteet, jotka keräävät älykkään mallin toteuttamiseen vaadittavaa dataa. Dataa kerätään valikoiduista energiankulutukseen vaikuttavista tekijöistä. Kerättyä raakadataa lähetetään pilvipalvelun kautta seuraavalle tasolle eli datan esivalmistelukerrokselle. Tällä kerroksella dataa tarkastellaan ja viimeistellään sopivaan muotoon varsinaisen älykkään malliin käytettäväksi. Datan keräämisen ja viimeistelyvaiheen jälkeen aloitetaan koneoppimisen menetelmien toteuttaminen. KNN- ja MLP-menetelmiä on hyödynnetty energiankulutuksen ennustamisessa kuluvalta tunnilta ja vastaavasti LSTM-menetelmää seuraavan tunnin arvojen ennustamiseen. Viimeisellä kerroksella eli suorituskyvyn arviointikerroksella arvioidaan mallin suorituskykyä sekä vertaillaan toteutuneita malleja ja menetelmiä keskineliövirheen neliöjuuren ja keskiarvon avulla. Kuvassa 8 olevan prosessikaavion avulla voidaan tarkastella prosessin vaiheita ja käytettyjä menetelmiä energiankulutuksen ennustamisessa.



Kuva 8. Energiankulutuksen ennustamisen prosessikaavio.

4.2 Datan keräämisen kerros

Älykkään mallin rakentamisessa hyödynnettiin aikasarjadataa energiankulutukseen vaikuttavista tekijöistä. Kerätyn datan avulla ennustetaan energiankulutusta. Dataa kerättiin 26.11.2018 – 3.12.2018 viidestä sensorista. Ulkolämpötilaa, jota käytetään myös yhtenä sisääntuloarvona, kerättiin Ilmatieteenlaitokselta. Talon ainoa energianlähde on sähköenergia, jonka kulutusta seurattiin sähkölaitoksen tietojärjestelmästä. Se esitetään kilowattitunteina (kWh), joka edustaa talon todellista reaaliaikaista energiankulutusta tunnissa. Tulosten laadun parantamiseksi sekä laskenta-ajan säästämiseksi kaikista lähteistä tulevaa dataa käsitellään keskiarvona tunnin ajalta. Datasetti sisältää 179 arvoa seitsemästä eri lähteestä, joista yksi edustaa outputia eli talon energiankulutusta kilowattitunteina. Taulukossa 4 esitetään kerätyn datasetin datamäärä, keskiarvo, keskihajonta, minimi ja maksimi jokaisen energiankulutukseen vaikuttavan tekijän kohdalla.

Taulukossa 4 on seuraavia tekijöitä:

- Illuminance = valaistusvoimakkuuden keskiarvo/tunti, joka kertoo tietyn suuruiseen pintaan kohdistuvan valovirran suuruuden. Data kerätään valaistusvoimakkuuden sensorin avulla, joka mittaa 0.1 - 40,000 lux-arvoa. Anturi sisältää sekä infrapuna- että spektridiodeja.
- Indoor temperature = Talon sisälämpötilan keskiarvo tunnin aikana. Dataa kerätään (DHT22) lämpötila- ja kosteussensorin avulla. Se pystyy mittaamaan välillä -40 °C – 125 °C, 0,5 °C:n tarkkuudella.
- Outside temperature = Talon ulkolämpötilan keskiarvo tunnin aikana. Dataa kerätään Ilmatieteenlaitoksen tietojärjestelmästä mittauspisteinä Kaarinan kaupunki.
- Humidity = Talon sisäisen suhteellisen kosteuden keskiarvo tunnin aikana. Se kertoo ilmassa olevan vesihöyryn määrän. Sensori mittaa 0-100% suhteellisen kosteuden 5% tarkkuudella
- PIR = Liiketunnistimesta saadut 0/1-arvot, jotka ilmaisevat liikkeen talon sisällä tunnin jaksoissa. PIR-sensorin avulla voidaan tunnistaa erityisesti

ihmisen liikettä anturin näkökentällä. Anturin toiminta perustuu infrapunasäteilyn tason mittaukseen. Anturi tunnistaa infrapunasäteilyjen nopeat muutokset, jotka johtuvat esimerkiksi ihmisen havaitsemisesta.

- Volume = Äänisensorista saadut 0/1-arvot, jotka kertovat mahdolliset talon sisällä olevat desibelimuutokset tunnin jaksoissa. Sensori pystyy mittaamaan äänen amplitudin mittaamalla analogisen jännitteen noin 50 dB herkkyydellä.
- Energy = Talon sähkönkulutuksen määrää kilowattitunteina, joka kertoo käytetyn wattimäärän tunnin ajalta. Dataa kerättiin sähkölaitoksen tietojärjestelmästä.

Taulukko 4. Kerätyn sensoridatan keskiarvot, hajonnat ja minimi- sekä maksimiarvot.

	Illuminance lux	Indoor temperature C	Outdoor temperature C	Humidity %	PIR	Volume	Energy kWh
Mean	51.87	20.69	-1.72	25.87	-	-	2.83
Std	118.30	0.47	3.11	3.07	-	-	1.71
Min	1.00	19.88	-8.00	21.06	0.00	0.00	0.99
Max	821.47	22.51	3.50	39.15	1.00	1.00	13.29

4.3 Datan esivalmisteluserros

Reaalimaailmassa saatavilla oleva data voi olla puutteellista, epä johdonmukaista tai sisältää häiriötekijöitä. Jotta saataisiin paras mahdollinen tulos, data on käsiteltävä ennen varsinaisen konenoppimismenetelmien toteuttamista. Datan valmisteluvaiheen tarkkuudella on vaikutus tulosten tarkkuuteen. Datan esivalmistelu koostuu eri vaiheista ja toimenpiteistä. Datan esivalmistuskerroksella jakauma normalisoidaan hyödyntämällä log- tai neliöjuurimuunnosta sekä poistetaan mahdolliset poikkeavat havaintoarvot (outlier), jotka johtuvat mittausvirheistä. Lisäksi tarkastellaan datan muotoa, luokittelua, yhteensopivuutta sekä viimeistellään dataa tarkastamalla desimaalipilkkuja, päivämääriä ja päällekkäisyyksiä.

Jossain tapauksissa koneoppimisen menetelmien toteuttaminen edellyttää, että dataa normalisoidaan, jotta kaikilla data-arvoilla olisi samanlainen arvoalue 0 – 1 välillä. Arvojen normalisointi toteutettiin alla olevan kaavan avulla.

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}, i = 1, 2, \dots, n$$

Kaava 5. Datan normalisointi (Ojha, 2019).

Missä \hat{x}_i edustaa normalisoitua arvoa, x_i on i sisääntulo dataa, x_{min} ja x_{max} edustavat x_i datan minimi- ja maksimiarvoa ja n näytteiden määrää.

Datan visualisointi auttaa saamaan kokonaiskuvan ja ymmärryksen käsiteltävästä datasta. Kuviossa 3 olevassa pistediagrammissa esitellään kerätty raakadata. Pistediagrammin avulla voidaan selvittää, millaisia relaatioita, käyttäytymisiä ja suhteita kerätyn datan välillä on. Sen avulla voidaan havaita mahdollisia lineaarisia riippuvuuksia, sekä niiden laatua ja vahvuuksia. Tarkasteltava data jaetaan kahteen muuttujakategoriaan, jotka ovat riippumattomat ja riippuvat muuttujat, jonka jälkeen dataa analysoidaan. Seuraavaksi tarkastelemme pistediagrammia tarkemmin.

Talon energiankulutuksen taso on monen tekijän summa. Tutkimuksen kannalta olennaisin asia on tarkastella riippumattomien muuttujien vaikutusta riippuvaan muuttujaan (kWh-arvoon eli energiankulutukseen).

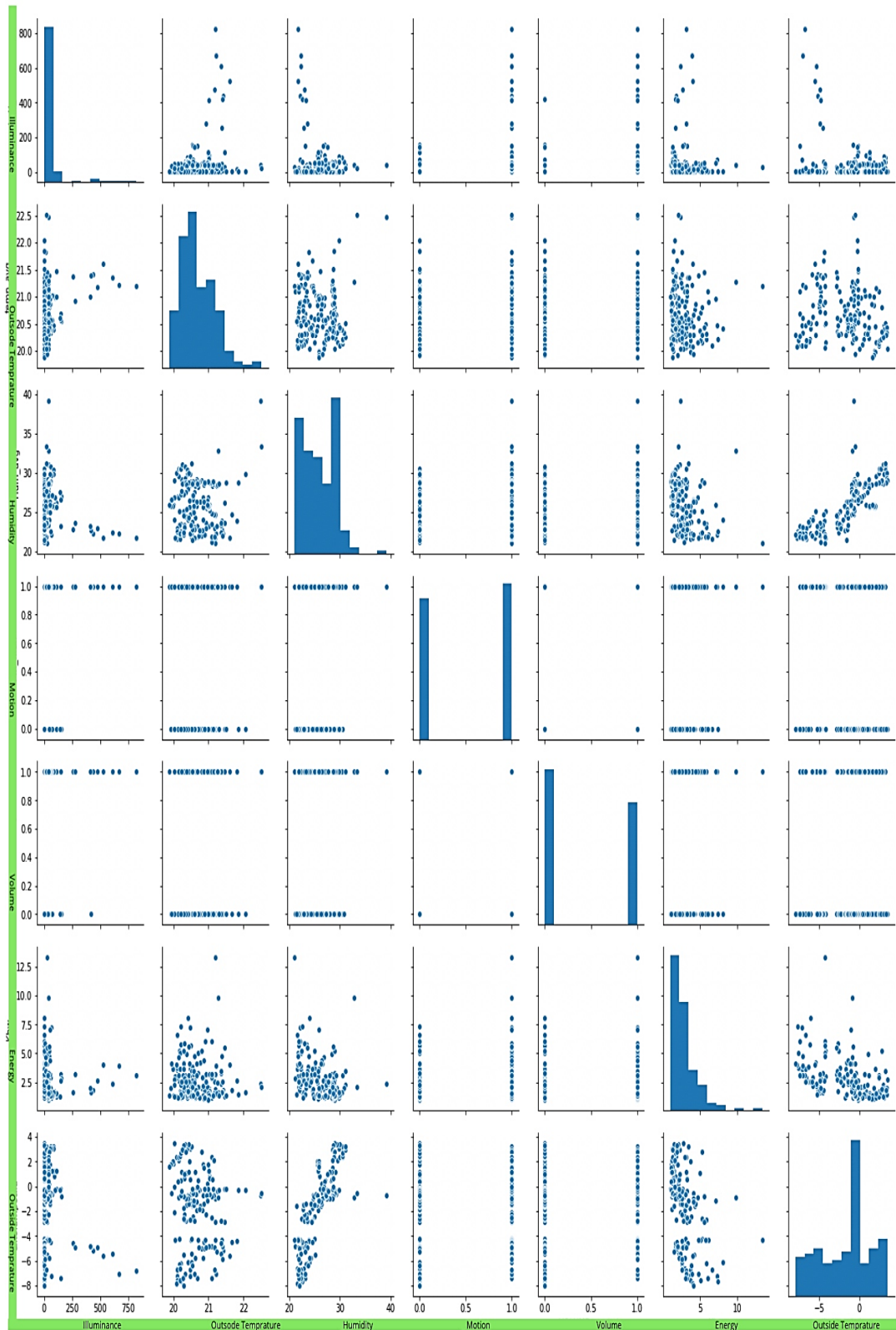
Ulkolämpötila on yksi tärkeimmistä energiankulutukseen vaikuttavista tekijöistä. Talvella merkittävää määrä energiankulutuksesta kuluu lämmitykseen. Pistediagrammin perusteella voidaan todeta, että energiankulutuksen ja ulkolämpötilan välillä on negatiivinen lineaarinen riippuvuus.

Asukkaan ääneen sekä liikkeeseen liittyvät sensoriarvot kertovat asukkaiden läsnäolosta talossa sekä näiden käyttäytymisestä, mikä tässä asiassa tarkoittaa äänen tasoa. Molempien kohdalla havaitaan suurempaa energiakulutusta asukkaan läsnä ollessa sekä silloin kun talossa on ihmisen tai toisen äänilähteen, kuten television ääni.

Valaistusvoimakkuuden arvon kohdalla, joka edustaa talossa olevan valaistuksen voimakkuutta, ei huomata näkyvää lineaarisuutta, mutta suurempiin energiankulutuksen lukemiin kuitenkin päästään pimeinä aikoina, jolloin luksiarvo on nollan tuntumassa. On kuitenkin huomattava, että Suomessa talven aikana valoisa aika on lyhyt. Toisaalta nykyisellä teknologialla saadaan energiatehokkuutta LED-valaistuksella. Toinen huomioitavaa asiaa valaistukseen käytettyyn energiaan liittyen on valaistuksen sammutus yöaikaan, kun asukkaat ovat nukkumassa.

Talon sisälämpötilan sekä kosteusprosentin ja energiankulutuksen välillä ei ole näkyvää vahvaa lineaarista suhdetta. Asia voi johtua esimerkiksi uudisrakennusten minimaalisesta lämpöhävikistä, joka vaikuttaa energiankulutukseen. Toinen vaikuttava asia on talon lämmitysjärjestelmä, joka termostaattien avulla ylläpitää talon lämpötilaa vakiona. Kosteusprosentilla on vaikutusta lämmön johtuvuuteen ja lämpötilan psykofyysiseen kokemukseen, mutta nykyisissä taloissa, jossa lämpötila on lähes vakio, kyseinen kosteuden vaikutus on hyvin minimaalinen.

Edellä olevat tarkastelut asettivat energiankulutuksen (kWh) riippuvaksi muuttujaksi pistediagrammin y-akseliin. Tässä vaiheessa on tärkeää huomioida, että muiden energiankulutukseen vaikuttavien tekijöiden väliset mahdolliset relaatiot sekä niiden tarkastelu energiankulutuksen näkökulmasta antaa laajemman näkemyksen kokonaisuudesta.



Kuvio 2. Kerätyn sensoridatan pistediagrammi 26.11.2018 – 3.12.2018.

4.4 Ennustamisen kerros

Tässä kerroksessa toteutetaan ennustamista muuttujien välisten tilastollisten yhteyksien perusteella. Ennustaminen toteutetaan kahdella tavalla:

- Kuluvan tunnin muutosten ennustaminen
- Seuraavan tunnin muutosten ennustaminen

Kuluvan tunnin energiankulutuksen ennustamisessa hyödynnetään sensoridatan sekä muista datalähteistä kerättyjen data-arvojen yhteyksiä energiankulutukseen. Mallin toteutumisessa hyödynnetään KNN ja MLP-menetelmiä.

Seuraavan tunnin muutosten ennustamista toteutetaan aikasarja-analyysin avulla. Aikasarjalla tarkoitetaan tietyllä aikavälillä kerättyjä data-arvoja, joka tässä projektissa on tunnin välein. Aikasarjaperusteisessa ennustamisessa käytetään LSTM-menetelmää.

Ohjelmointi

Mallin varsinainen ohjelmointi on toteutettu Python-ohjelmointikielellä. Python on ohjelmointikieli, joka tarjoaa rajattoman määrän mahdollisuuksia erityisesti data-analyysin ja koneoppimisen projekteihin. Pythonin avulla voidaan suorittaa data-analytiikkaan ja visualisointiin liittyviä projekteja. Koneoppimisen menetelmien toteuttaminen edellyttää kuitenkin vaadittavien ohjelmointikirjastojen asentamista. Pandas-nimistä ohjelmointikirjastoa käytettiin kerätyn datan käsittelyä ja analysointia varten. K-lähimmän naapurin menetelmän toteuttamisessa hyödynnettiin Pythonin Scikit-Learn-nimistä koneoppimiseen liittyvää ohjelmointikirjastoa. Se sisältää monia luokitteluun, ryhmittämiseen sekä regressioanalyysin liittyviä työkaluja. Se on alun perin kehitetty Google Summer Of Code nimisessä projektissa ja tullut yhdeksi suosituimmaksi kirjastoksi koneoppimisen projekteissa. Se on yhteensopiva Pythonin muiden matemaattisiin ja data-analyysin liittyvien kirjastojen, kuten Numpy:in ja SciPy:in kanssa. (Scikit-learn, 2019)

Projektin keinotekoisii neuroverkkoihin perustuvat ohjelmoinnit on toteutettu TensorFlow-ohjelmointikirjaston ja Keras-ohjelmointirajapinnan avulla. TensorFlow on avoimeen lähdekoodiin perustuva koneoppimisohjelmisto, jonka avulla suoritetaan numeerista laskentaa datavuokaavioiden avulla. TensorFlow'n avulla voidaan jakaa koneoppimiseen liittyvää laskentaa yhdelle tai useammalle prosessorille tai näytönohjaimille laskennan tehokkuuden ja nopeuden parantamiseksi. TensorFlow on alustariippumaton ja yhteensopiva grafiikkasuorittimen sekä Androidin- ja IOS:n käyttöjärjestelmien kanssa.(Unruh, 2017).

Keras on ohjelmointirajapinta, joka keskittyy neuroverkkojen ja erityisesti monikerroksisten syväoppimismallien toteuttamiseen. Keras helpottaa koneoppimisen mallien toteuttamista interaktiivisella tavalla, sen tavoitteena on yksinkertaistaa ohjelmointia sekä tunnistaa mahdollisia virheitä tehokkaammin. Se sisältää kaikkia neuroverkon toteuttamiseen tarvittavia työkaluja, funktioita ja matemaattista tukea. Se on täysin yhteensopiva Pythonin kanssa, joka helpottaa vaativien ohjelmakokonaisuuksien toteuttamista.

4.5 Suorituskyvyn arviointi

Viimeisessä vaiheessa eli suorituskyvyn arviointivaiheessa tarkastellaan mallin suorituskykyä testausvaiheessa saatujen tulosten perustella. Tähän käytetään kahta yleistä mittaria keskineliövirhettä (MSE) sekä keskineliövirheen neliöjuurta (RMSE) alla olevien yhtälöiden avulla:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Kaava 6. Keskineliövirheen laskentakaava (Binieli, 2018).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Kaava 7. Keskineliövirheen neliöjuuren laskentakaava (Binieli, 2018).

Kaavoissa y on todellisen energiankulutuksen arvo, \hat{y} on ennustettu energiankulutuksen arvo ja n näytteiden määrä.

5 TULOKSET

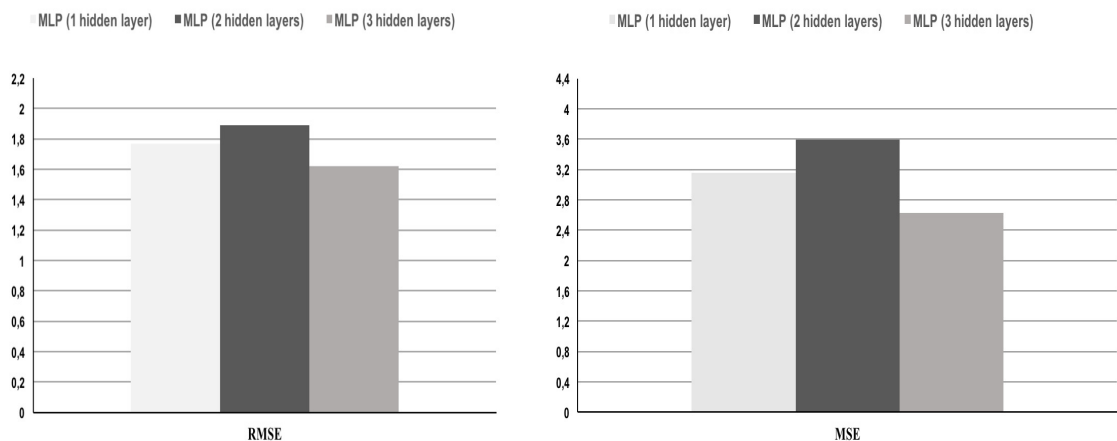
5.1 Monikerroksinen Perceptron (MLP)

Koska neuronien sekä piilokerrosten määrällä on suoranainen vaikutus mallin suorituskykyyn, testattiin kolme MLP mallia yhdellä, kahdella sekä kolmella piilokerroksella. Taulukossa 4 nähdään, kuinka monta neuronia jokaisen mallin piilokerroksessa on.

Taulukko 5. Neuronien määrä jokaisessa piilokerroksessa.

Model	Layer1	Layer2	Layer3
MLP (1 hidden layer)	6	-	-
MLP (2 hidden layers)	6	6	-
MLP (3 hidden layers)	6	6	6

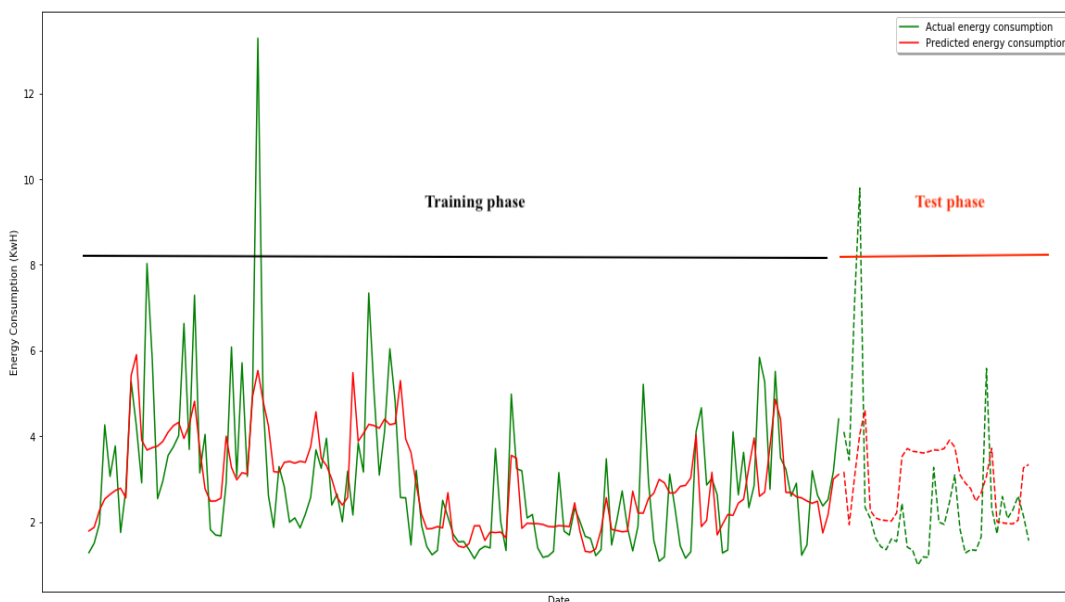
Mallin opetuksessa käytettiin Adam-optimointia 0,00001, 0,1 vähennyskerroksella. Kuten kuviossa 4 nähdään MLP-malli kolmella piilokerroksella saavutti parhaan tuloksen minimaalisella keskineliövirheen arvolla (2,63) ja keskineliövirheen neliöjuuren arvolla (1,62).



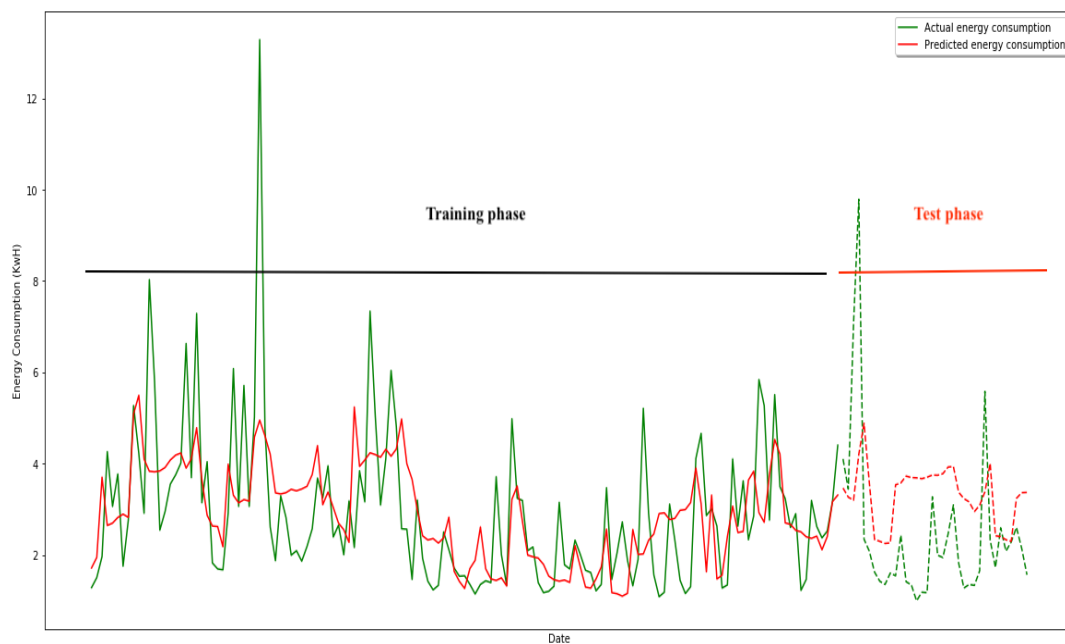
Kuvio 3. MLP-mallin MSE ja RMSE arvot eri piilokerroksilla

Kuviot 5-7 esittävät kolmen ehdotetun MLP-mallin ennustustuloksia kahdessa oppimis- ja testivaiheessa. Vihreä käyrä edustaa todellista energiankulutusta ja

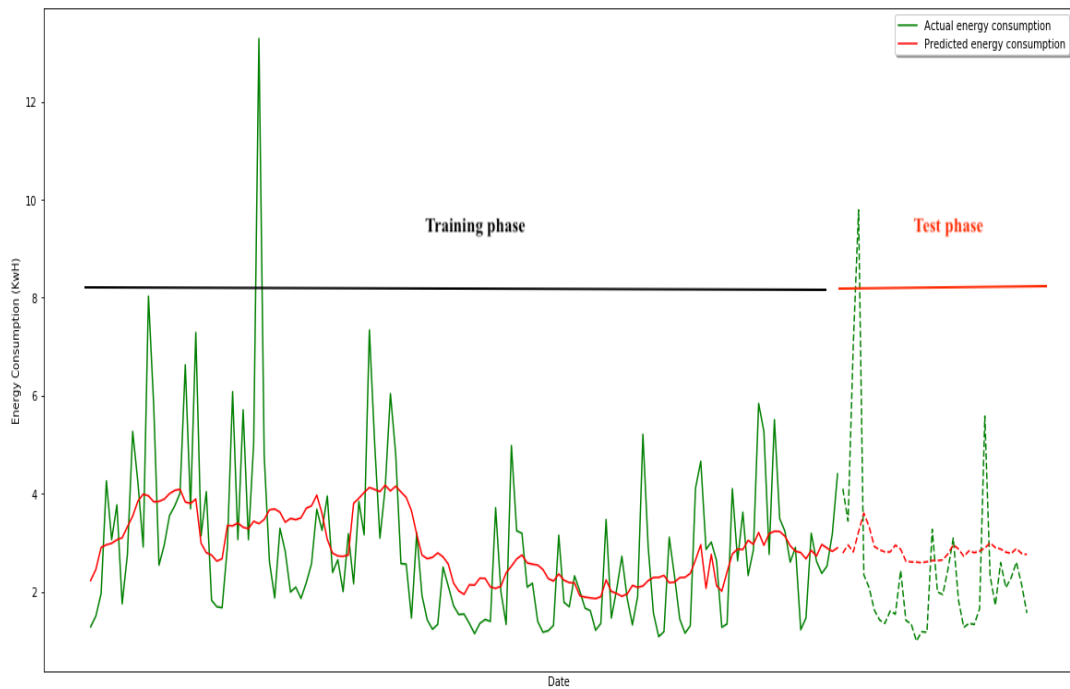
punainen ennustettua energiankulutusta. Kuten kuviossa 7 nähdään MLP-mallin kohdalla, jossa on kolme piilokerrosta, ennustetun ja todellisen energiankulutuksen välinen ero on pienempi, kuin MLP-malli, jossa on vähemmän piilokerroksia.



Kuvio 4. Energiankulutuksen ennustaminen MLP-mallin avulla yhdellä piilokerroksella.



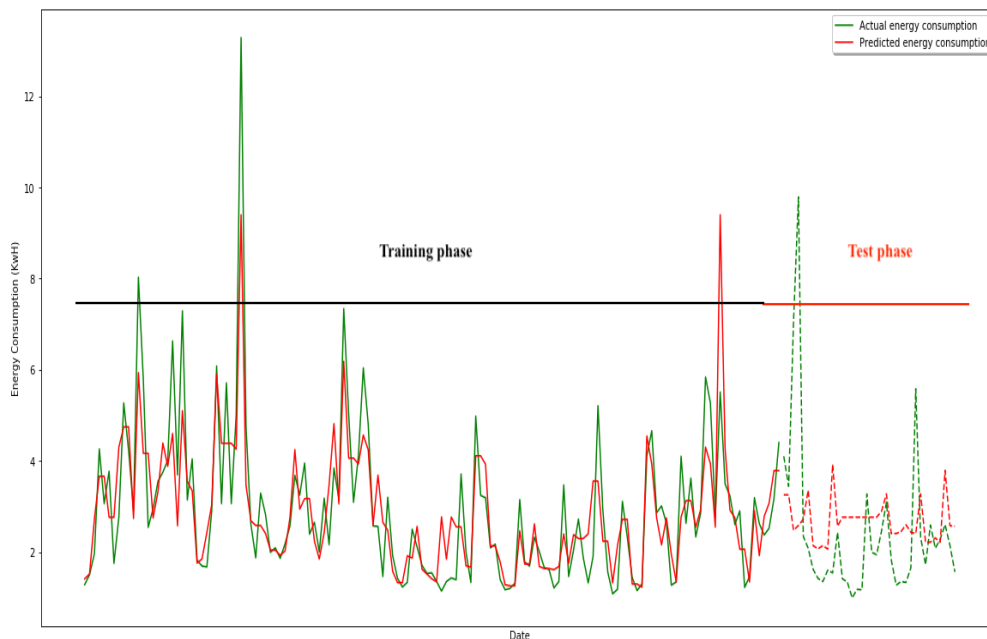
Kuvio 5. Energiankulutuksen ennustaminen MLP-mallin kahdella piilokerroksella.



Kuvio 6. Energiankulutuksen ennustaminen MLP-mallin avulla kolmella piilokerroksella.

5.2 K-lähimmän naapuri (KNN)

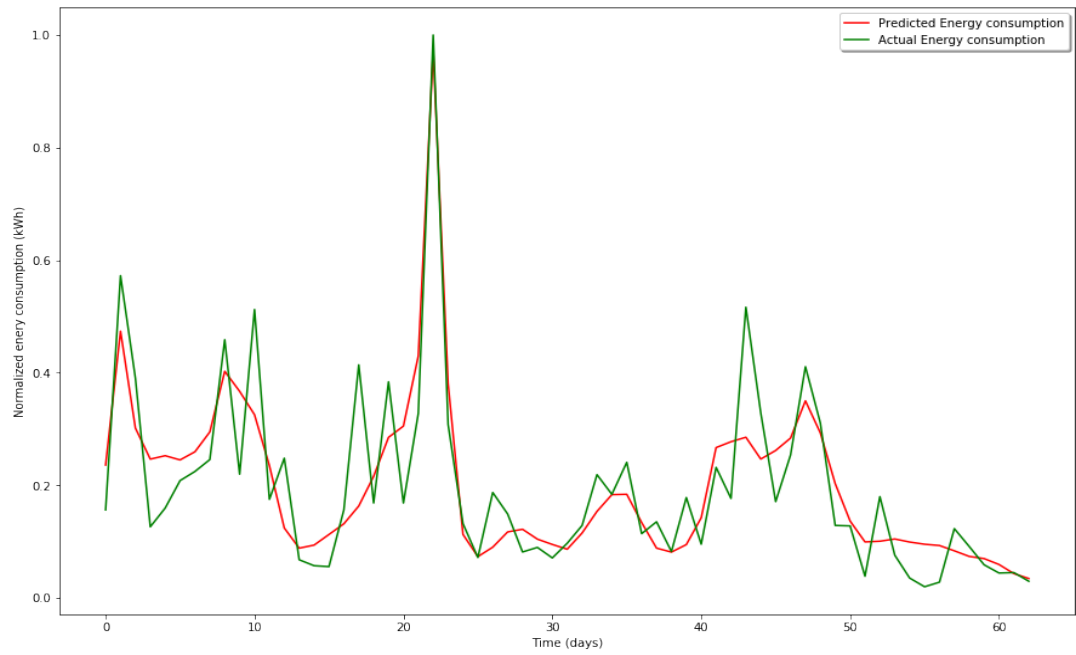
Tulokset osoittavat, että KNN -pohjainen malli saavuttaa hyvin tarkkoja ennustustuloksia. Tulosten perusteella mallin keskineliövirheen arvo on (3,25) ja keskineliövirheen neliöjuuren arvo (1,80). Kyseisen menetelmän kohdalla hyödynnettiin yksi pois -ristiinvaldointi LOOCV-metodia, jotta saataisiin paras k :n arvo. k :n arvolle määriteltiin arvoalueeksi 1-10 ja parhaan tuloksen saavutus tapahtui arvolla $k=2$. Kuviossa 8 nähdään KNN-mallin todellisten ja ennustettujen energiankulutuksen arvojen eroa, jota kuvataan vihreällä ja punaisella värillä kahdessa oppimisenvaiheessa sekä testivaiheissa.



Kuvio 7. Energiankulutuksen ennustaminen KNN- menetelmän avulla.

5.3 Pitkät lyhytaikaisen muistin verkot (LSTM)

LSTM-mallin avulla ennustettiin energiankulutusta seuraavalta tunnilta. Malli toteutettiin kolmella LSTM-kerroksella, jossa jokaisessa on 30 neuronia. Vastaavasti erän koolle ja ajan jakson arvoille asetettiin arvot 32 ja 200. Tulosten perusteella Adam-optimointia hyödyntämällä saadaan paras neuroverkkoihin perustuvan LSTM- mallin toteutus. Tulosten perusteella malli pystyy ennustamaan talon energiankulutusta seuraavalta tunnilta (0,0062) keskineliövirheen arvolla ja (0,0789) keskineliövirheen neliöjuuren arvolla. Kuvio 9 esittää talon todellisen energiankulutuksen ja ennustetun arvon välisiä eroavaisuuksia, jotka liittyvät talon seuraavan tunnin energiankulutuksen lukemiin.



Kuvio 8. Seuraavan tunnin energiankulutuksen ennustaminen LSTM-mallin avulla.

6 YHTEENVETO

Tutkimuksen tavoitteena oli ennustaa talon energiankulutusta koneoppimisen menetelmillä. Ideana oli rakentaa kokonaisuus, joka sisältäisi myös prototyypin, jonka avulla kerättäisiin dataa asuintalosta. Projekti koostui taustatietojen hankinnasta, fyysisen prototyypin rakentamisesta, prototyypin ohjelmoinnista, kerätyn datan analyysistä sekä älykkään mallin rakentamisesta.

Älykkään mallin rakentamisen edellytyksenä oli kerätä riittävä määrä dataa energiankulutuksen vaikuttavista tekijöistä, kuten asukkaiden läsnä- ja poissaolosta ja aktiviteeteista talossa, sisä- ja ulkolämpötilasta, sisäilman kosteuden tasosta sekä talon valaistusvoimakkuudesta. Tavoitteena oli selvittää edellä mainittujen tekijöiden vaikutusta energiankulutukseen sekä havaita korrelaatiota tekijöiden välillä. Tutkimusta varten rakennettiin Arduino-mikrokontrolleriin perustuva prototyyppi, jonka avulla tarvittavaa dataa kerättiin taloon asennettujen sensoreiden avulla. Sensoreista saatua dataa analysoitiin mahdollisten energiankulutukseen vaikuttavien tekijöiden välisten korrelaatioiden selvittämiseksi. Talon energiankulutusta ennustettiin kahden tunnetun koneoppimisen mallin, k-lähimmän naapuri (KNN) sekä monikerroksisen perceptron (MLP) –mallin, avulla. Talon seuraavan tunnin energiankulutusta ennustettiin pitkät lyhytaikaisen muistin verkot (LSTM)-mallin avulla.

Tulosten mukaan MLP-malli kolmella piilokerroksella ennusti parhaiten talon energiakulutusta keskineliövirheen arvolla 2,63 ja keskineliövirheen neliöjuuren arvolla (1,62. K-lähimmän naapurin (KNN)-mallin osalta keskineliövirheen arvo oli 3,25 ja keskineliövirheen neliöjuuren arvo 1,80.

Vastaavasti LSTM-mallin avulla ennustettiin talon energiankulutusta seuraavalta tunnilta minimaalisella keskineliövirheen arvolla 0,07.

Projektissa käytetyt menetelmät, kuten esimerkiksi neuroverkot, olivat monimutkaisia kokonaisuuksia, jotka edellyttivät osaamista ja teorian ymmärrystä ja aiheuttivat haasteita tekijälleen. Projektin tuloksena syntyi toimiva järjestelmä, joka pystyy käytännössä ennustamaan valikoitujen tekijöiden pohjalta talon

sähköenergian kulutusta. Järjestelmä on toiminut testiympäristössä eli asuintalossa hyvin ja pystynyt tekemään tarkkoja ennustuksia kuluvaista sekä tulevasta energiankulutuksen määrästä.

Koneoppimisen menetelmät perustuvat hyvin pitkälti saatavilla oleviin eri datalähteistä kerättyihin esimerkkietoihin, joiden perusteella varsinaista älykästä mallia toteutetaan. Laadukkaan mallin rakentaminen vaatii tietämyksen järjestelmästä, johon älykästä mallia integroidaan. Talon energiankulutus on monen tekijän summa. Olosuhteet sekä ympäristöön liittyvät tekijät ovat jatkuvassa muutoksessa, jotka voisivat aiheuttaa suuriakin haasteita mallin toimivuudessa eri tilanteissa ja ympäristöissä. Edellä mainittujen seikkojen takia mallin toteutuksessa huomioitiin ylisovittamisen haittavaikutuksia, jotta järjestelmä pystyisi toimimaan myös muissakin ympäristöissä onnistuneesti. Ylisovittamisella tarkoitetaan mallin liiallista sovittamista käytössä olevaan historiadataan, joka heikentäisi omalta osaltaan mallin toimivuutta uusissa tilanteissa.

Tuloksen luotettavuuden kannalta on hyvin tärkeää selvittää talon energiankulutukseen vaikuttavien tekijöiden todellista vaikutusta energiankulutukseen. Talon lämmitysjärjestelmän tuntemus on edellytyksenä toimivan älykkään ratkaisun toteuttamiseen. Tutkimuksessa on yritetty ottaa kyseisiä asioita mahdollisuuksien mukaan huomioon. On kuitenkin mainittavaa, että tutkimuksen tuloksena syntynyt älykäs malli esittää kokonaisuuden, miten energiankulutusta voitaisiin asuintaloissa ennustaa, vaikka tulosten luotettavuutta voitaisiinkin kyseenalaista tutkimuksessa käytettyjen valikoitujen tekijöiden todellisesta vaikutuksesta energiankulutuksen ennustamiseen.

Merkittävin tutkimuksen ja edellä mainittujen aikaisempien tutkimuksien välinen ero on datalähteiden laajuudella, joiden perusteella älykästä mallia toteutetaan. Aikaisemmat tutkimukset hyödyntävät ainoastaan yhtä tai kahta energiankulutukseen vaikuttavaa tekijää ennustamisessaan, kun tässä tutkimuksessa huomioitiin kaikkia vaikuttavia tekijöitä mukaan. Aikaisemmat tutkimukset ovat käyttäneet käytössä olevaa dataa, esimerkiksi sähkönkulutuksen tilastollista dataa, koneoppimisen mallin opetusdatana, kun

tässä tutkimuksessa rakennettiin tähän tarkoitukseen erikseen prototyyppi. Tämän tutkimuksen ja aikaisempien tutkimusten välisten vertailukelpoisten tulosten löytäminen ja vertailu on haastavaa. Tässä tutkimuksessa hyödynnetty opetusdatan keräysmenetelmät, ajanjakso sekä koneoppimisen menetelmien toteuttamiseen liittyvät poikkeavuudet edellisten tutkimusten kanssa vaikeuttavat tulosten suoranaista vertailua. Opinnäytteen aikaisemmassa tutkimusosiossa tarkasteltu Platonin tutkimusryhmä (Platon ym. 2014) on ennustanut rakennuksen energiankulutusta seuraavalta tunnilta ANN-mallia hyödyntämällä 0,083 keskineliövirheen neliöjuuren arvolla, mikä on lähes samaa tasoa kuin tässä tutkimuksessa saatu tulos RMSE 0,0789.

Tutkimuksessa saatuja tuloksia voidaan hyödyntää asuintalojen energiankulutuksen optimoinnissa, energian tuotannon suunnittelussa sekä uusiutuvien energialähteiden käytön edistämisessä.

Asuintalojen energiankulutuksen ennustamisen avulla voidaan optimoida energiankulutusta, alentaa kustannuksia, havaita energiankulutuksen poikkeavuuksia sekä edistää esineiden internetiä älykodissa. Talon energiankulutuksen ennustamisen avulla voidaan ennakoivien toimenpiteiden avulla vaikuttaa energiankulutuksen tasoon. Energiankulutuksen ennustamisen menetelmiä voidaan hyödyntää älykotien automatisoinnissa, jossa järjestelmä tekee energiankulutuksen vaikuttavia päätöksiä, jotka toteutetaan esineiden internetin toimivien laitteiden avulla.

Optimointia voidaan suorittaa älytermostaattien, älyvalaistuksen sekä muiden talossa olevien älylaitteiden avustuksella. Energiankulutuksen ennustamista voidaan hyödyntää myös asukkaiden ohjeistuksessa ja opetuksessa tilanteissa, joissa älykäs järjestelmä havaitsee kulutukseen vaikuttavien toimenpiteiden tarpeellisuutta. Energiankulutuksen ennustamisen sekä IoT:n avulla voidaan saada maksimaalinen hyöty aikaperusteisessa energiahinnoittelussa eli tässä tapauksessa sähkön hinnoittelussa suorittamalla osan talon rutiinitoimenpiteitä sähkönkulutuksen ruuhka-aikojen ulkopuolella. Aikaperusteisessa hinnoittelussa ruuhka-ajoille määritellään korotettuja taksoja.

Energiankulutuksen ennustamisen avulla voidaan suunnitella ja tasapainottaa energiantuotantoa käyttötarpeen mukaisesti. Sähköenergiantuotantoon liittyvät energiantarpeen ali- ja yliarviointi tuottavat haasteita tuotantotahoille. Sähköenergian tarpeen aliarviointi johtaa sähkökatkoksiin sekä toimituskustannusten kasvuun, kun toisaalta yliarviointi aiheuttaisi mahdollisia tyhjäkäyntihäviöitä. Uusiutuvien energialähteiden tuotanto on hyvin riippuvainen ympäristön ja muuttuvien olosuhteiden, kuten sään muutoksista. Energiankulutuksen ennustamisen avulla voidaan varautua näistä muutoksista johtuviin tuotannon määrän alenemisiin tasoittamalla ennustukseen perustuvaa energian tarpeen vastaavaa lisäystä fossiilisten energialähteiden avulla.

Tutkimuksessa keskityttiin pientalon energiankulutuksen ennustamiseen liittyvien menetelmien kehittämiseen. Tutkimuksen toteutuminen oli ensimmäinen askel energiankulutuksen optimointiin. Jatkotutkimuksissa voidaan siirtyä seuraaviin vaiheisiin, joissa tavoitteena olisi toteuttaa varsinaista optimointia energiankulutuksen ennustukseen perustuvien tuloksien pohjalta. Optimointia voidaan toteuttaa koneoppimisenmenetelmien sekä esineiden internetin avulla, joka integroidaan tämän projektin tuloksena syntyneeseen älyjärjestelmään.

LÄHTEET

- Abdulzاهر, M. 2019. Artificial Intelligence Journalism: the 4IR and media restructuring. Giza: Badael.
- Arduino. 2019. ArduinoEthernetShieldV1. Viitattu 20.03.2019 <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>.
- Ashton, K. 2009. That "Internet of Things" Thing. Viitattu 24.09. 2018 <https://www.rfidjournal.com/articles/view?4986>.
- Binieli, M. 2018. Machine Learning: An Introduction to Mean Squared Error and Regression Lines. Viitattu 20.11.2019 <https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/>.
- Cicuito. 2019. Arduino vs. Raspberry Pi: Which Board Is Best. Viitattu 13.08.2018 <https://www.cicuito.io/blog/arduino-vs-raspberry-pi/>.
- Cisco.2014. The Internet of Things Reference Model, Viitattu 13.01.2020 http://cdn.iotwf.com/resources/71/loT_Reference_Model_White_Paper_June_4_2014.pdf.
- Dencelin, L. 2016. Analysis of multilayer perceptron machine learning approach in classifying protein secondary structures. Viitattu 11.01.2020 <https://www.alliedacademies.org/articles/analysis-of-multilayer-perceptron-machine-learning-approach-in-classifying-protein-secondary-structures.html>.
- Fayaz, M. & DoHyeun, K. 2018. A Prediction Methodology of Energy Consumption Based on Deep Extreme Learning Machine and Comparative Analysis in Residential Buildings. Viitattu 01.10.2019 https://www.researchgate.net/publication/327948990_A_Prediction_Methodology_of_Energy_Consumption_Based_on_Deep_Extreme_Learning_Machine_and_Comparative_Analysis_in_Residential_Buildings.
- Giovanni, N. 2016. How the Internet was born: from the ARPANET to the Internet. Viitattu 09,10,2019 <https://theconversation.com/how-the-internet-was-born-from-the-arpamet-to-the-internet-68072>.
- Guo, H. & Gelfand, B. 1991. Analysis of Gradient Descent Learning Algorithms for Multilayer Feedforward Neural Network. Viitattu 16.02.2020 https://www.researchgate.net/publication/3184219_Analysis_of_gradient_descent_learning_algorithms_for_multilayer_feedforward_neural_networks.
- Hochreiter, S. & Schmidhuber, J. 1997. Long Short-Term Memory. Viitattu 09,10,2019 <https://www.bioinf.jku.at/publications/older/2604.pdf>.
- Inventrom. 2014. The "Thing" in "Internet of Things. Viitattu 19.07.2019 <https://inventrom.wordpress.com/2014/11/27/the-thing-in-internet-of-things/>.
- Joutsijoki, H. 2017. Koneoppiminen. Viitattu 19.03.2019 <https://coss.fi/wp-content/uploads/2017/12/4-Koneoppiminen.pdf>.
- Karilahti, O. 2017. Pelottava ajatus, että älykoti tietää, milloin koti on tyhjiään – onko pelkoon syytä?. Viitattu 7.03.2019 <https://www.mtvuutiset.fi/artikkeli/pelottava-ajatus-etta-alykoti-tietaa-milloin-koti-on-tyhjillaan-onko-pelkoon-syyta/6357814#gs.mnyb8h>.
- Khoshafian, S. 2016. The adaptive digital factory: IoT reference architectures. Viitattu 15.03.2019 <https://www.pega.com/insights/articles/adaptive-digital-factory-iot-reference-architectures>.

- Lawtomated. 2019. Supervised Learning vs Unsupervised Learning. Which is better?. Viitattu 13.01.2020 <https://lawtomated.com/supervised-vs-unsupervised-learning-which-is-better/>.
- Leiner, B. 1997. Brief History of the Internet. Viitattu 25.03.2019 <https://www.internet-society.org/internet/history-internet/brief-history-internet/>.
- Meola, A. 2016. Internet of Things devices, applications & examples. Viitattu 28.09.2018 <https://www.businessinsider.com/internet-of-things-devices-applications-examples-2016-8>.
- Mitchell, T. 1997. Machine Learning. Norwell: McGraw-Hill, Inc.
- Motiva. 2018. Näin luet energiatodistusta. Viitattu 13.01.2020 <http://energiatodistus.motiva.fi/mika-on-energiatodistus/nainluetenergiatodistusta/>.
- Nilan. 2019. Poistoilmalämpöpumppu (PILP) lämmitteää edullisesti. Viitattu 12. 09. 2019 <https://www.nilan.fi/poistoilmalampopumppu/>.
- Ojha, Y. 2019. A Beginners guide to Machine Learning — Data Preprocessing. Viitattu 10.01.2020 <https://medium.com/@yogeshojha/data-preprocessing-75485c7188c4>.
- Olofsson, T. & Andersson, S. 2001. Long-term energy demand predictions based on short-term measured data. Viitattu 02.09.2019 <https://www.sciencedirect.com/science/article/abs/pii/S0378778800000682>.
- Pientalon lämmitysjärjestelmät. 2018. Lämmityksen valinta. Viitattu 10.09.2019 <https://www.suomirakentaa.fi/omakotirakentaja/laemmitys/laemmityksen-valinta>.
- Platon, R. & Dehkordi, V. & Martel, J. 2015. Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. Viitattu 16.01.2020 <https://www.journals.elsevier.com/energy-and-buildings>.
- Puuinfo. 2019. Puurakentaminen. Viitattu 10.09.2019 <https://www.puuinfo.fi/puutieto/puurakentaminen>.
- Qiong, L. 2010. Prediction model of annual energy consumption of residential buildings. Viitattu 11.09.2019. <https://ieeexplore.ieee.org/document/5557576>.
- Rakentaja. 2019. Vesikiertoinen lattialämmitys - lämmitteää talvella, viilentää kesällä. Viitattu 12.09.2019 https://www.rakentaja.fi/artikkelit/10078/vesikiertoinen_lattialammitys_on_turvallinen.htm.
- Raghunath, D. 2019. k-nearest neighbors. Viitattu 01.01.2020 <https://medium.com/@rundayala/k-nearest-neighbors-a76d0831bab0>.
- Robinson, S. 2018. K-Nearest Neighbors Algorithm in Python and Scikit-Learn. Viitattu 15.02.2018 <https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/>.
- Roos, T. 2018. Johdatus tekoälyyn. Viitattu 01.08.2019 https://www.cs.helsinki.fi/webfm_send/932.
- Rouse, M. 2012. What Is Sensor. Viitattu 18.03. 2019 <https://whatis.techtarget.com/definition/sensor>.
- Rouse, M. 2017. What Is Sensor Fusion. Viitattu 25.03.2019 <https://whatis.techtarget.com/definition/sensor-fusion>.
- Sakovich, N. 2018. 8 Most Popular IoT Protocols and Standards You Need to Know. Viitattu 25.03.2019 <https://www.sam-solutions.com/blog/internet-of-things-iot-protocols-and-connectivity-options-an-overview>.

- Scikit-learn. 2019. machine learning in Python. Viitattu 12.05.2019 <https://scikit-learn.org/stable/>.
- Stevens, L. 2018. What Is a Smart Home. Viitattu 7.03.2019 <http://home.bt.com/tech-gadgets/internet/connected-home/what-is-a-smart-home-11364214165664>.
- Sukanya, M. 2015. Internet of Things (IoT) - Part 2 (Building Blocks & Architecture). Viitattu 19.05.2019 <https://www.c-sharpcorner.com/uploadfile/f88748/internet-of-things-part->
- Truong, N. 2013. Forecasting Multi-Appliance Usage for Smart Home Energy Management. Viitattu. 07.09. 2019 https://www.researchgate.net/publication/262318081_Forecasting_multi-appliance_usage_for_smart_home_energy_management.
- Tso, G. & Yau, K. 2007. Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Viitattu 15.09.2019 <https://www.sciencedirect.com/science/article/abs/pii/S0360544206003288>.
- Työ- ja elinkeinoministeriö. 2011. Energiatohokkuus julkisissa hankinnoissa. Viitattu 13.02.2020 https://www.motiva.fi/files/4644/Energiatohokkuus_julkisissa_hankinnoissa.pdf.
- Unruh, A. 2017. What Is the TensorFlow Machine Intelligence Platform? Viitattu 15.05.2019 <https://opensource.com/article/17/11/intro-tensorflow>.
- Van Rijmenam, M. 2014. Where Does The Internet of Things Come From? Viitattu 25.03.2019 <https://datafloq.com/read/where-does-the-internet-of-things-come-from/524>.
- Wallnerström, C. & Seteus, J. 2010. Model of capacity demand under uncertain weather. Viitattu 01.10.2019 https://www.researchgate.net/publication/224161401_Model_of_capacity_demand_under_uncertain_weather.
- Woodford, C. 2019. Smart homes and the Internet of Things. Viitattu 01.03.2019 <https://www.explainthatstuff.com/smart-home-automation.html>.
- Xuan-Hien, Le. & Hung Viet, Ho. 2019. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. Viitattu 10.01.2020 <https://www.mdpi.com/2073-4441/11/7/1387>.

