

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**TERMOREGULĀCIJAS SISTĒMAS IZSTRĀDE VIEDAJAI
MĀJAI**

BAKALAURA DARBS

Darba autors: **Rihards Mozalevskis**

Studenta apliecības nr.: rm11055

Darba vadītājs: Dr. dat. Matīss Rikters

Rīga 2019

Anotācija

Lietderīga un saprātīga mājas automatizācijas un mākslīgā intelekta izmantošana var pamatīgi uzlabot cilvēku dzīves līmeni un atbrīvot cilvēkus no dažādiem rutīnas darbiem. Šī bakalaura darba mērķis ir izveidot funkcionējošu termoregulācijas sistēmu viedajai mājai, kas reģistrē visas lietotāju darbības un apstākļus kādās tās veiktas, un izveidot tai piemērotu mašīnmācīšanās modeli, ko laika gaitā, kad tiks savākts pietiekams datu daudzums, apmācītu un ļautu sniegt ieteikumus par termoregulēšanu mājās. Darba rezultātā tika izveidota pilnīgi funkcionējoša termoregulācijas sistēma, ar kuras palīdzību var attālināti un ar runas vadību regulēt logus un radiatorus, kā arī pārraudzīt temperatūru divās mājas istabās.

Atslēgas vārdi – Raspberry Pi, viedā māja, mašīnmācīšanās, mājas automatizācija

Abstract

“Development of thermoregulation system for smart home”

A useful and reasonable use of home automation and artificial intelligence can greatly improve people's standard of living and free people from various routines. The aim of this bachelor's thesis is to create a functioning thermoregulation system for a smart home that records all users' activities and conditions in which they have been performed, and to create a suitable machine learning model that, over time, when sufficient data is collected, would train and allow recommendations on home thermoregulation. The work resulted in a fully functioning thermoregulation system that allows remote and speech control of windows and radiators, as well as monitoring temperatures in two rooms of the house.

Keywords – Raspberry Pi, smart home, machine learning, home automation.

SATURS

IEVADS	6
1. VIEDĀ MĀJA.....	7
1.1 Kas ir viedā māja	7
1.2 Vēsture.....	7
1.3 Arhitektūra.....	8
2. ESOŠIE RISINĀJUMI.....	9
2.1. Tet apkures vadība.....	9
2.2. Elektrum viedā māja.....	9
2.3. Ecobee 4	10
2.4. Paštaisīti tehniskie risinājumi	10
2.4.1. Radiatoru regulatori	11
2.4.2. Logu atvērēji	12
3. TERMOREGULĀCIJAS SISTĒMAS IZSTRĀDE	14
3.1. Raspberry Pi	15
3.2. Logu atvēršana.....	17
3.3. Radiatora regulēšana.....	19
3.4. Gaismas un temperatūras mērīšana	20
3.5. Kontroles un pārraudzības lietotne	21
3.6. Virtuālais asistents.....	26
3.7. Mašīnmācīšanās modeļa apmācība.....	27
3.7.1. Weka	29
3.7.2. Datu sagatavošana programmai “Weka”	29
3.7.3. Datu imports programmā “Weka”	32
3.7.4. Klasifikatora algoritma pielietošana	32
3.7.5. Izveidotā modeļa testēšana	35
REZULTĀTI.....	36
SECINĀJUMI.....	37

IEVADS

Izplatoties viedierīcēm un palielinoties interneta nozīmei cilvēku dzīvē, šo divu elementu kombinācijas tiek izmantotas aizvien vairāk cilvēku ikdienas uzlabošanai un dažādu sadzīves procesu automatizēšanai. Viedās mājās jeb mājas procesu automatizācija jau vairs nav nekas jauns. Pēdējos gados ar lielu vērienu attīstās arī mākslīgais intelekts, kas spēj mācīties no cilvēku veidotiem piemēriem un pēc kāda laika jau pieņemt lēmumus patstāvīgi.

Apvienojot viedās mājās principu ar mākslīgo intelektu, iespējams iegūt māju, kas tiek automatizēta nevis pēc strikti definētiem likumiem, bet gan pēc iedzīvotāju paradumiem, kurus ir apkopojis un apguvis mākslīgais intelekts.

Šī darba mērķis ir izveidot reālu termoregulācijas sistēmas moduli viedajai mājai, kas sevī ietver:

- tīmekļa lietotni, ar kuru regulēt šo sistēmu no jebkuras ierīces, kurai pieejams interneta pieslēgums un tīmekļa pārlūkprogramma;
- datu krātuvi, kurā tiek uzskaitītas visas iedzīvotāju darbības un apstākļi kādos tās veiktas;
- ar uzkrātajiem datiem apmācītu mašīnmācīšanās modeli, kas pats varēs pieņemt lēmumus iedzīvotāju vietā;
- sistēmas integrāciju ar gudro asistentu "Google Home".

Plānots, ka termoregulācijas sistēma spēs ieslēgt un izslēgt apkuri, kā arī atvērt un aizvērt logus. Šie ir divi galvenie procesi, kas regulē mājas iekšējo temperatūru. Lai nodrošinātu mākslīgā intelekta apmācības procesu, tiks uzkrāti dati par to, kādos apstākļos tiek izmantota termoregulācijas sistēma, tas ir, tiks saglabāta katras darbības laikā esošā temperatūra, gaismas daudzums, kā arī precīzs laiks, kad notikusi šī darbība.

Darba struktūra – darba pirmajā nodaļā ir aprakstīti galvenie viedās mājas principi un tās attīstības vēsture. Otrajā nodaļā ir izpētīti un salīdzināti dažādi gan komerciāli, gan bezpeļņas hobija līmeņa esošie risinājumi, kas publicēti interneta resursos. Trešajā nodaļā ir aprakstīts šajā darbā izstrādājamās termoregulācijas sistēmas tapšanas process un visas tās sastāvdaļas, kā arī izvēlētie risinājumi, kā arī aprakstīti iespējami izmantojamie mašīnmācīšanās modeļi. Noslēgumā ir aprakstīti iegūtie rezultāti un secinājumi.

1. VIEDĀ MĀJA

1.1 Kas ir viedā māja

Viedā māja ir sistēma, kas ļauj attālināti regulēt dažādas ierīces un procesus mājās, kā piemēram, apgaismojumu, apkuri, drošības ierīces un citas ierīces no jebkuras vietas pasaulē, kur pieejams interneta pieslēgums vai arī bez interneta pieslēguma no mājas iekšējā tīkla. Viedajā mājā dažādas ierīces ir savienotas savā starpā izmantojot interneta pieslēgumu, kas veido tā saucamo lietu-internetu. Lietu internets ir vienota sistēma, kas nodrošina dažādu ierīču saziņu bez tiešas cilvēka iesaistes [1].

1.2 Vēsture

Mājas automatizācijas procesus cilvēki sāka attīstīt jau 20. gadsimta sākumā, kad mājsaimniecībās sāka lietot veļas mašīnas, ledusskapjus, ūdens sildītājus un citas ar elektrību darbināmas ierīces, kas kaut kādā mērā atviegloja un automatizēja mājsaimniecības uzturēšanu.

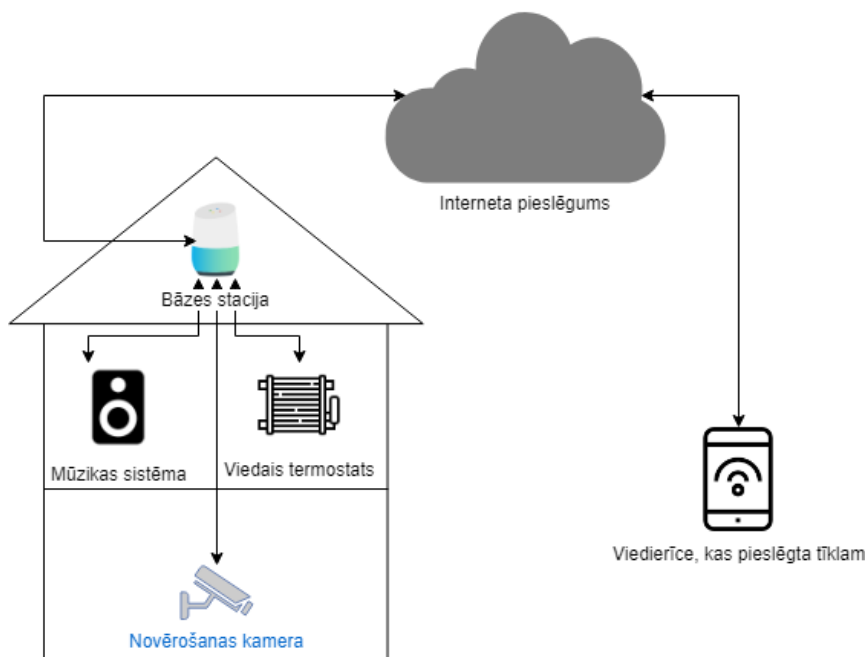
20. gadsimta 60. gados tika izveidota pirmā mājās viedierīce ECHO IV, kas spēja sagatavot iepirkumu sarakstus, kontrolēt mājas temperatūru, kā arī kontrolēt citas elektroniskas ierīces [2]. Šī un citas tā laika ierīces neguva patērētāju atzinību un netika pārdotas. 90. gadu sākumā tika attīstītas dažādas tehnoloģijas, kas padarītu drošāku vientuļo senioru dzīvi, piemēram, automātiskās trauksmes sistēmas, ja cilvēks ir viens mājās un ir radušās kādas veselības problēmas.

Viedās mājas koncepts popularitāti plašākā sabiedrībā ieguva 21. gadsimta sākumā, kad straujāk sāka izplatīties internets, un sāka attīstīties viedierīces, kā arī visa IT tehnika kļuva pieejamāka arī cenu ziņā.

1.3 Arhitektūra

Ar arhitektūru šajā nodaļā tiek aprakstīta nevis pašas mājas uzbūve, bet gan viedās mājas sistēmas uzbūve.

Viedās mājas arhitektūru iespējams izveidot dažādi, tomēr ir daži pamata elementi, kas nemainās.



1.3.1. att. Viedās mājas primitīva arhitektūra

Lai kontrolētu ierīces, ir nepieciešams interneta pieslēgums, bāzes stacija, kas ir pieslēgta internetam un ir ar iestrādātu drošības mehānismu, lai māju varētu kontrolēt tikai cilvēki, kuriem tas ir atļauts. Šī bāzes stacija ir kā vārti uz māju un tā tālāk padod komandas un sazinās ar pārējām viedās mājas ierīcēm, kas jau var būt dažādas un neskaitāmas, kā piemēram, radiatora termostats, gaismas slēdzis, video novērošana (1.3.1. att.).

2. ESOŠIE RISINĀJUMI

Pašlaik tirgū jau ir pieejami dažādi rīki un sistēmas, kas ļauj attālināti regulēt mājas temperatūru. Šādus pakalpojumus piedāvā arī uzņēmumi Latvijā.

2.1. Tet apkures vadība

Uzņēmums “tet”(agrāk “Lattelecom”) piedāvā apkures vadības sistēmu, kas, uzstādot dažādas ierīces, ļauj regulēt sildelementus vai apkures katlu no attāluma, kā arī uzstādīt automātisku siltuma regulēšanu, atkarībā no temperatūras sensora noteiktās temperatūras.

Lai uzstādītu šo risinājumu, jāiegādājas bāzes stacija, kas nodrošina komunikāciju starp sistēmas ierīcēm un pieslēdz tās internetam, temperatūras sensors vai gaisa kvalitātes sensors, kas spēj noteikt arī citus fizikālus lielumus, kā piemēram CO₂ līmeni telpā, kā arī slēdži un/vai regulatori, atkarībā no tā, vai tiks regulēts tikai radiators, apkures katls vai abi kopā. Ja tiktu iegādātas ierīces, kas atbilst šajā darbā uzstādāmajām ierīcēm, neskaitot logu atvēršanas mehānismu, kas šeit netiek piedāvāts, tad aptuvenās izmaksas šīs sistēmas uzstādīšanai būtu 163 EUR. Tet piedāvātajai bāzes stacijai interneta pieslēgums tiek nodrošināts ar Ethernet kabeli, kas nozīmē, ka tie ierobežota tā novietošanas vieta.

2.2. Elektrum viedā māja

Elektrum piedāvā uzstādīt dažāda veida apkures kontroles sistēmas. Tuvākais risinājums šajā darbā izstrādājamajam ir risinājums “Centrālapkure” [4]. Šis risinājums ir piemērots dzīvokļiem, kuru mājās ir apkures stāvvads un apkuri nepieciešams regulēt nevis katrā istabā atsevišķi, bet gan vienoti visā dzīvoklī. Šis risinājums ietver bāzes staciju, regulatoru un temperatūras sensoru un tā izmaksas ir 228 eur. Šis risinājums ir piemērots arī dzīvokļiem, kuriem ir apkures sistēmas bez termostatiem un apkuri iespējams regulēt tikai pilnībā to noslēdzot.

Temperatūras sensors tiek novietots telpā, pēc kuras temperatūras tiks regulēta apkure visā dzīvoklī, bet viedais regulators tiek uzstādīts uz pirmā vārsta aiz individuālā siltuma skaitītāja. Kad temperatūra telpā, kur novietots sensors, sasniedz norādīto robežu, regulators noslēdz vai pieslēdz apkuri visam dzīvoklim.

Elektrum piedāvā uzstādīt arī apkures regulācijas sistēmu pa istabām, tas ir, regulatorus un temperatūras sensorus uzstādīt katrā istabā, kur tas nepieciešams. Ir jāiegādājas tās pašas ierīces, kas uzstādot vienoto regulēšanas sistēmu, tikai katrai istabai atsevišķi.

2.3. Ecobee 4

Tehnoloģiju apskatnieki un ierīču lietotāji viedo mājas termostatu Ecobee 4 ir atzinuši par pagaidām labāko viedo termostatu, kāds pieejams tirgū [5].

Šis risinājums atšķiras no iepriekš aprakstītajiem ar to, ka pamatā tā ir tikai bāzes stacija, ar kuru iespējams kontrolēt dažādas citas viedierīces mājās. Iegādājoties šo bāzes staciju, komplektā ir iekļauts arī viedais sensors “Room sensor” [6], kas ir spējīgs nolasīt temperatūru un noteikt, vai telpā kāds atrodas. Šis ir bezvadu sensors un, pateicoties zemas jaudas radio frekvences saziņai ar “Ecobee 4”, tā akumulators darbojas līdz pat 2 gadu garumam [7].

Šajā termostatā ir iebūvēts “Amazon Alexa” viedais asistents, kas nodrošina šo termostatu ar visu to funkcionalitāti, ko nodrošina Amazon bāzes stacijas un ļauj to kontrolēt izmantojot balss komandas.

Šis risinājums neietver nekādus rīkus, kas varētu tieši kontrolēt apkuri vai mājas dzesēšanu – tā galvenais uzdevums ir plānot un kontrolēt citu saistīto viedierīču darbu mājās.

“Ecobee 4” pamata komplekta cena (bāzes stacija un viens bezvadu sensors) ir 249 ASV dolāri un tas pagaidām ir pieejams tikai ASV un Kanādas tirgū.

2.4. Paštaisīti tehniskie risinājumi

Internetā atrodami dažādi paštaisīti risinājumi, kurus varētu izmantot termoregulācijas sistēmas izstrādei, tas ir, automātiskai logu atvēršanai un radiatoru regulēšanai. Logu atvēršanas risinājumi ir veidoti galvenokārt diviem logu veidiem – atveramajiem logiem un bīdāmajiem logiem, tomēr mājā, kurai tiek veidota šī sistēma, ir tikai atveramie logi, tāpēc tiks apskatīti tikai šāda veida risinājumi. Radiatoru regulatori pārsvarā ir tikai viena veida – pagriežami ar rotācijas kustību.

2.4.1. Radiatoru regulatori

Populārākais un piemērotākais risinājums automātiskai radiatoru regulācijai ir lineārā spēka pievada soļu motors, kas griež radiatora vārstu. Galvenās sastāvdaļas šādiem risinājumiem ir:

- 1) Soļu motors, kas veic rotācijas kustību (2.4.1.1. att.);
- 2) Dators, kurā noprogammēt komandas;
- 3) Motora kontroliera modulis.



2.4.1.1. att. Soļu motora piemērs¹

Ir pieejami dažādu veidu soļu motori, kas atšķiras ar cenu, izmēru, spēku un citiem parametriem. Lai pēc visu detaļu saslēgšanas, risinājums izskatītos pievilcīgi un iederētos mājas interjerā, iespējams to dažādi iepakot, ielikt kādā kastītē. Internetā ir publicēts projekts, kurā ir izveidots šāds viedais radiatora regulators un speciāli tam, izmantojot 3D printeri, izdrukāta kastīte, kurā ielikt visas sastāvdaļas [13]. Visi resursos atrodamie risinājumi ir ar vadiem pieslēdzami pie elektrības. Dators, kas kontrolē motoru, ir pieslēgts tīklam, tādējādi tiek nodrošināta iespēja tam pieslēgties un attālināti vadīt radiatora regulatoru.

¹ <https://www.aliexpress.com/item/2PCS-NMB-Linear-Actuator-2-phase-4-wire-Stepper-Motor-5V-9V-6V-put-pull-motor/32509356978.html>

2.4.2. Logu atvērēji

Atveramos logus iespējams kontrolēt ar dažāda veida motoriem. Interneta vietnē www.instructables.com ir publicēts risinājums automātiskai loga atvēršanai kaķa vajadzībām [14], kur loga atvēršanai tiek izmantots servomotors, kas ir piestiprināts pie sienas un rotējošā daļa ir savienota ar logu(2.4.2.1. att.).



2.4.2.1. att. Loga atvēršanas mehānisms, izmantojot servomotoru².

² <https://www.instructables.com/id/Arduino-controlled-window-for-cat/>

Tiešsaistē atrodami arī risinājumi, kur logu atvēršana tiek nodrošināta ar lineārā spēka pievadu motoriem(2.4.2.2. att.), tas ir, ar motoriem, kuri izbīdās un iebīdās kādā noteiktā garumā.



2.4.2.2. att. **Lineārā spēka pievada motors³.**

Arī šāds risinājums ir ļoti piemērots atveramo logu kontrolēšanai. Šāda veida motori ir pielietojami jebkuru logu veidu atvēršanai. Lineārie spēka pievada motori pārsvarā darbojas, motora iekšējo rotācijas kustību pārvēršot lineārā kustībā, konkrēta darba veikšanai. Ir trīs pamata mehānismi, kā darbojas šie motori [15]:

- 1) Skrūves princips – rotējot skrūvei, skrūves galva kustas lineārā virzienā, kas arī veic nepieciešamo darbu;
- 2) Rats un ass – pie rotējoša rata ir piestiprināta ass, kas var būt ķēde, sikсна vai kas cits, ko var aptīt ap ratu, un, rotējot ratu, tiek pagarināta vai saīsināta ass;
- 3) Ekscentrs – pie rotora ir piestiprināts ekscentrisks rats, kurš rotējot kustina kādu detaļu, kura attiecīgi kustas lineāri.

Apskatot lineārā spēka pievada motoru darbības principus, ir skaidrs, ka arī izmantojot servomotoru, kas tika aprakstīts iepriekš, var izveidot šāda paša tipa improvizētu motoru, kas veic nepieciešamo darbu. Kā arī daļai lineāro spēka pievadu motoru iekšējo rotāciju nodrošina tieši servomotori.

³ <https://www.progressiveautomations.com/products/mini-linear-actuator>

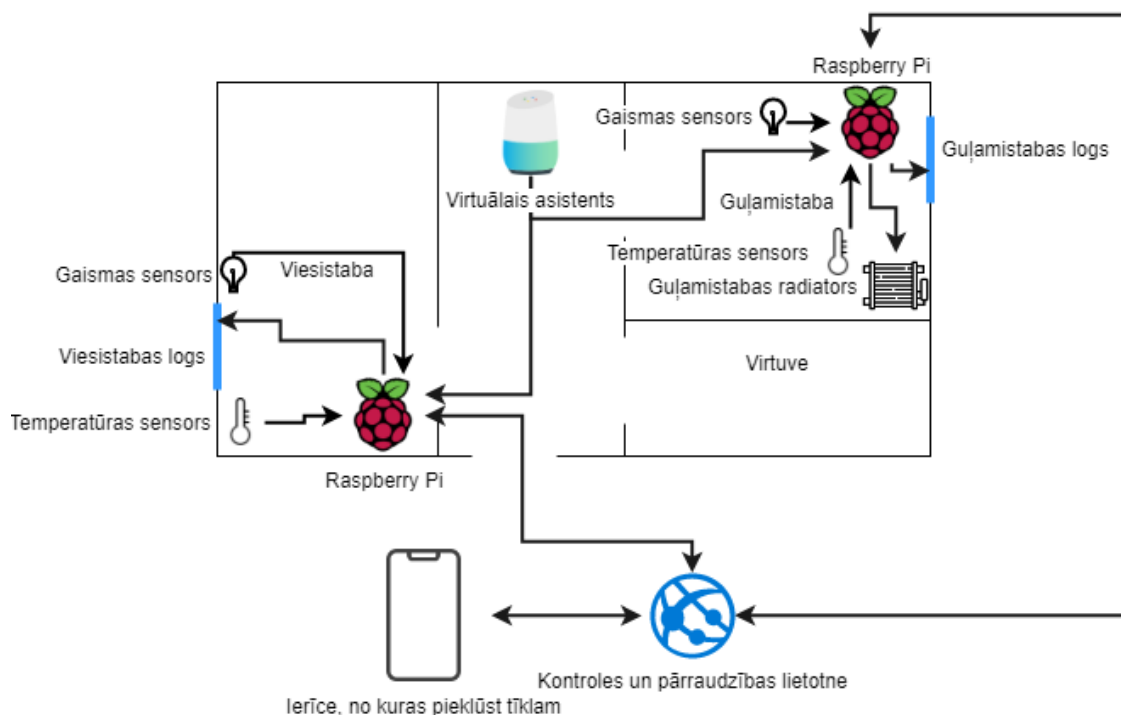
3. TERMOREGULĀCIJAS SISTĒMAS IZSTRĀDE

Šo risinājumu ir plānots izstrādāt, kā minimālo dzīvotspējīgo produktu, tas ir, tādu, kas veic primārās nepieciešamās darbības - sildelementu ieslēgšanu un izslēgšanu, kas šajā gadījumā ir radiatora vārsta atvēršana un aizvēršana, kā arī dzesēšanas elementu ieslēgšanu un izslēgšanu, kas šajā gadījumā ir loga atvēršana un aizvēršana. Kad būs pabeigta pamata funkcionalitātes izstrāde, tad tiks izveidota iespēja norādīt vēlamo istabas temperatūru, kuru sistēmai nepārtraukti ir jāuztur. Kā arī jāizveido iespēja norādīt dažādas temperatūras dažādos laika intervālos, piemēram, naktī nepieciešama zemāka gaisa temperatūra telpā nekā dienā.

Kā bāzes stacija šajā sistēmā tiek izmantota tīmekļa lietotne, kas izmitināta mākoņa platformā kombinācijā ar Google Home viedo asistentu, kas ļauj sistēmu regulēt ar balsi. Bāzes tīmekļa lietotne ļauj sistēmu darbināt no jebkuras vietas un ierīces, kur pieejams interneta pieslēgums.

Kā dators, kas kontrolēs temperatūru regulējošās ierīces, tiks izmantots “Raspberry Pi Zero W” minidators, kurš būs pieslēgts internetam.

Turpmāk šajā nodaļā tiks aprakstīti izvēlētie risinājumi, kas nodrošinās termoregulācijas sistēmas darbību dzīvoklī (3.1. att.).



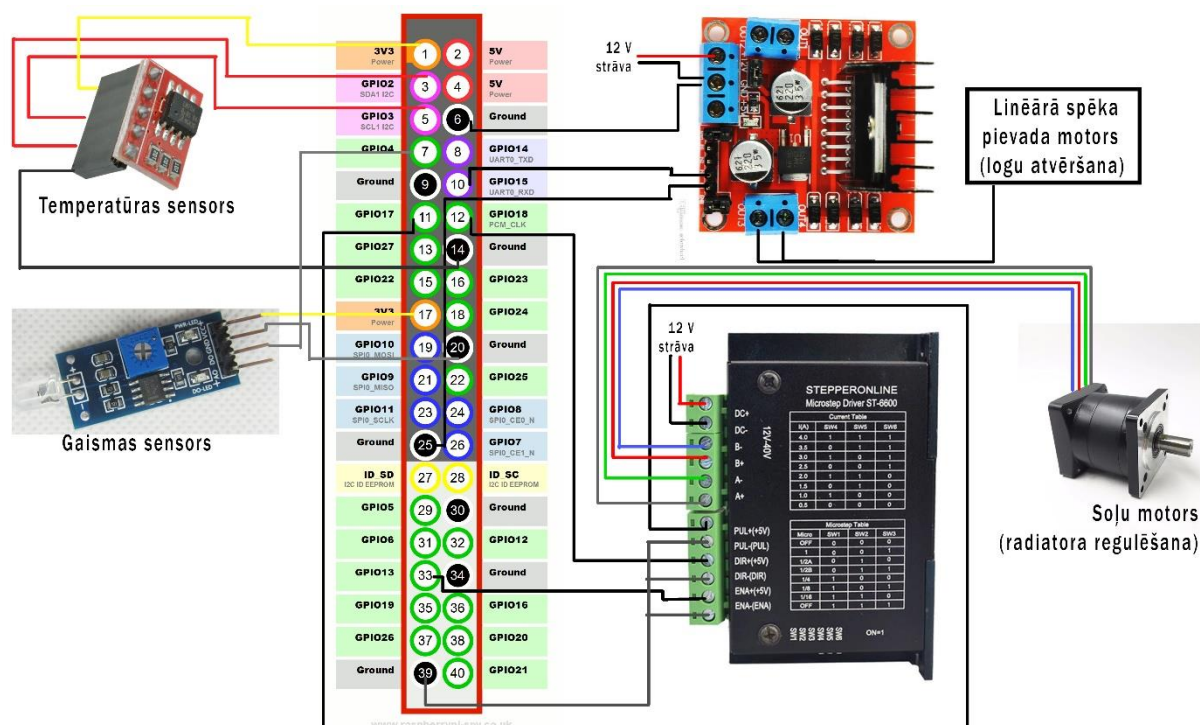
3.1. att. Termoregulācijas sistēmas izvietojuma plāns dzīvoklī

3.1. Raspberry Pi

“Raspberry Pi” ir kredītkartes izmēra minidators, kurš tika izstrādāts ar pamata mērķi, lai to izmantotu datorzinātņu un programmēšanas pamatu apguvei, tomēr lielu popularitāti tas iemantojis tieši automatizācijas sistēmu un robotu izstrādē [8]. Kopumā ir pieejami četrus veidu “Raspberry Pi” modeļi – 1, 2, “zero” un 3, kuriem, savukārt, ir pieejami arī dažādu veidu apakšmodeļi. Šie modeļi savā starpā pārsvarā atšķiras ar jaudu, pieslēguma veidiem un iespējamo pieslēgumu skaitu, kā arī ar izmēru [9].

2017. gadā tirgū parādījās "Raspberry Pi Zero W" modelis, kas bija aprīkots ar iebūvētu WiFi (802.11 b/g/n wireless LAN) un Bluetooth (4.1 un Bluetooth Low Energy) uztvērēju. Tas ir aprīkots ar viena kodola 1GHz ARM procesoru, 512 MB operatīvo atmiņu. Tam ir iestrādāts mini HDMI ports un viens USB “On-The-Go” ports, kā arī micro USB ports, kas tiek izmantots kā barošanas ports. “Raspberry Pi Zero W” ir aprīkots ar 40 spraudņu pieslēgvietām savietojamu moduļu pieslēgšanai [10].

Otrs projektā izmantotais minidators ir “Raspberry Pi Zero WH”, kas īpašību ziņā ir tāds pats kā ”Zero W” modelis, tikai tam papildus ir jau iestrādātas pieslēguma galviņas, pie kurām var pieslēgt citas vadāmas komponentes (3.1.1. att.).



3.1.1. att. Termoregulācijas sistēmas moduļa shēma

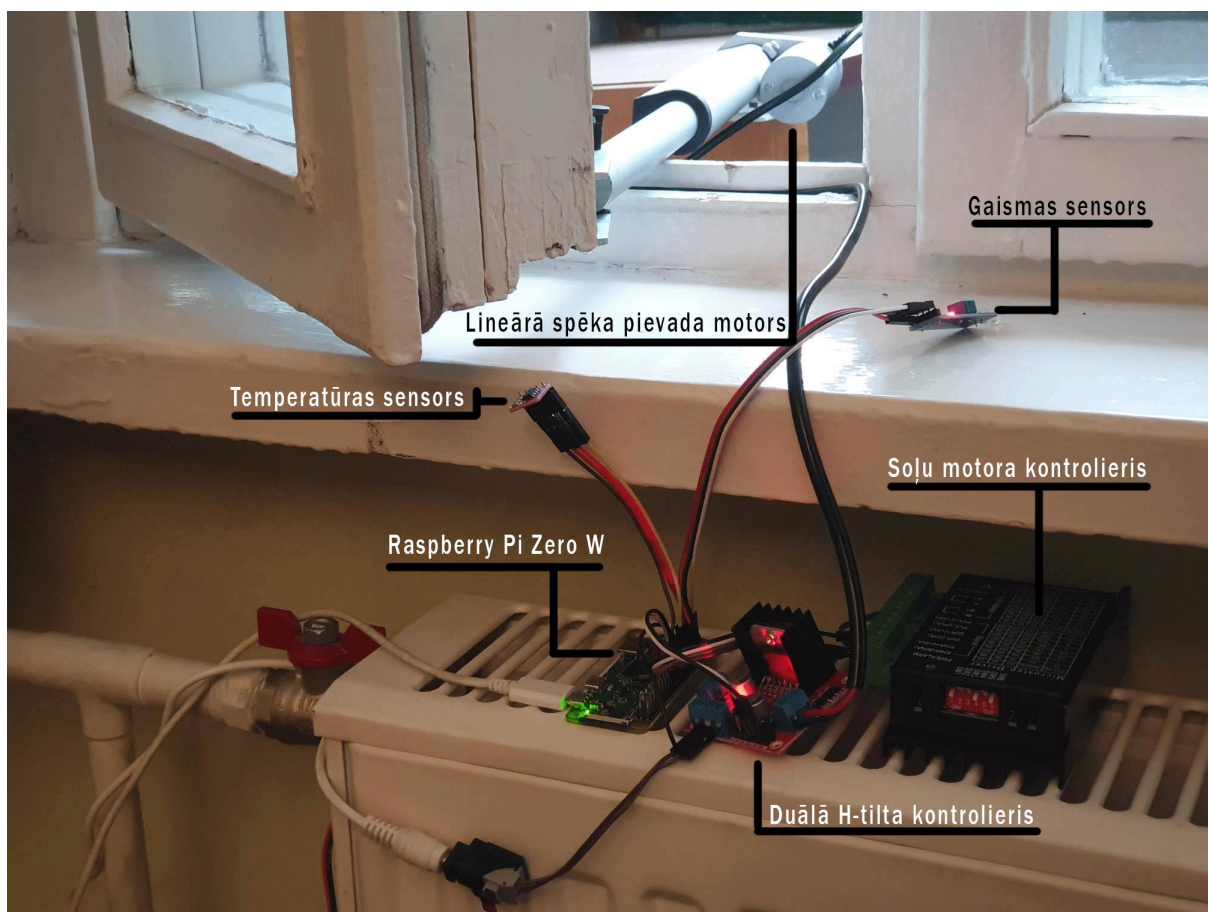
Tā kā projekta ietvaros tiek veidots risinājums divu istabu termoregulācijai, tad katrā istabā tiek izmantots viens “Raspberry Pi” minidators, ar kuru tiek kontrolēta gan logu

atvēršana, gan radiatoru regulēšana. Šāda risinājuma izvēle ļauj ietaupīt resursus, tomēr neļauj izveidot neatkarīgas komponentes, kuras var viegli pārvietot. Projektam attīstoties, noteikti tiks iegādāti papildu "Raspberry Pi" vai cita veida datori un izveidotas neatkarīgas komponentes mājas procesu regulēšanai.

Šim minidatoram ir uzinstalēta "Raspbian" operētājsistēma. "Raspbian" ir bezmaksas atvērta pirmkoda operētājsistēma, kuras pamatā ir "Debian" operētājsistēma un tā ir īpaši pielāgota "Raspberry Pi" minidatoriem. "Raspbian" satur visas nepieciešamās programmas un rīkus, lai darbinātu un programmētu "Raspberry Pi" minidatorus [11].

3.2. Logu atvēršana

Samazināt temperatūru mājās var, izmantojot dažādus paņēmienus, piemēram, izmantojot ventilatoru vai gaisa kondicionieri, taču vienkāršākais veids, kā to izdarīt, pie nosacījuma, ka āra gaisa temperatūra ir zemāka nekā gaisa temperatūra telpās, ir atverot mājas logu. Loga atvēršana ne tikai samazina gaisa temperatūru telpās, bet arī uzlabo gaisa kvalitāti, izvadot no telpas pieelpoto gaisu un ļaujot telpā ieplūst svaigam gaisam.



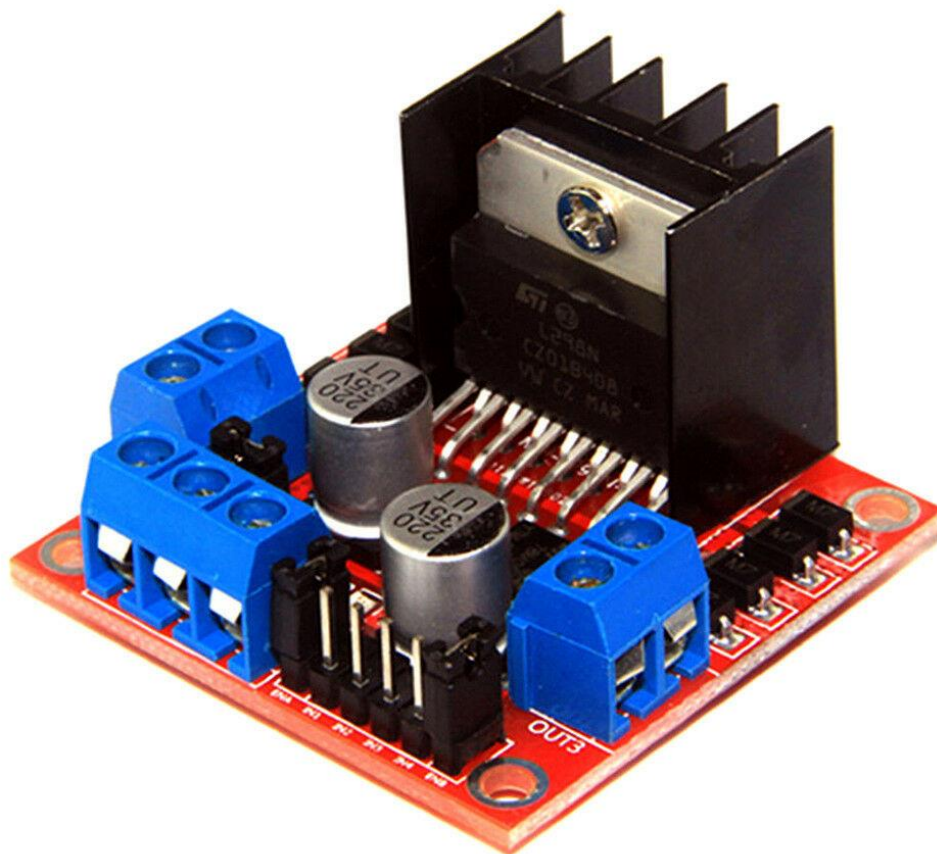
3.2.1. att. Guļamistabas kontrolieru komplekts

Eksistē dažādas ierīces, kas palīdz atvērt dažāda veida logus. Šajā projektā tiek izmantots lineārs mehānisms, kas ar elektromotora rotācijas kustības palīdzību pēc skrūves principa veido lineāru kustību, bīdot attiecīgā mehānisma detaļu (3.2.1. att.). Lai kontrolētu mehānisma darbības virzienu, ir jākontrolē strāvas plūšanas virziens uz rotora motoru, kas darbina mehānismu. Strāvas plūšanas virzienu var mainīt pārslēdzot pozitīvo un negatīvo vadu vietām. Šādu lineāro mehānismu iespējams iegādāties par aptuveni 40 EUR⁴. Cena ir atkarīga no

⁴ <https://www.ebay.com/itm/Heavy-Duty-1500N-Linear-Actuator-330lbs-Max-Lift-DC-12V-Electric-Motor-for-Auto/153111230469?var=452955873134>

mehānisma izmēra, ir pieejami dažādi garumi, kas jāizvēlas atkarībā no nepieciešamā loga atvēruma.

Logu atvēršanas mehānisms strādā ar 12V spriegumu un kustās ar spēku 750N. “Raspberry Pi” minidators saņem un spēj tālāk padot tikai 5V spriegumu, kā arī motora kustības virziena mainīšanai nepieciešams fiziski mainīt vadus, kas nozīmē, ka mājas procesi tiek nevis tiek automatizēti, bet gan nevajadzīgi sarežģīti, ir nepieciešams starposms starp minidatoru un loga atvēršanas mehānismu, kas pārveido spriegumu kā arī maina sprieguma plūšanas virzienu. Kā starposms tiek izmantots duālā “H-tilta” kontrolieris (3.2.2. att.), kas spējīgs padot 5-35V spriegumu, kā arī vārds “duālais”, nozīmē, ka pie tā vienlaicīgi var pieslēgt divus motorus. Šādu kontrolieri iespējams iegādāties par aptuveni 4 EUR⁵.



3.2.2. att. Duālā H-tilta kontrolieris⁵.

⁵ <https://www.ebay.com/p/Dual-H-Bridge-DC-Stepper-Motor-Drive-Controller-Board-Module-L298n-Compatible/2223515306?iid=231454411738>

3.3. Radiatora regulēšana

Šodien tirgū ir pieejami dažādi viedie termostati, kas uzstādāmi gan ar, gan bez vadiem un ļauj attālināti regulēt radiatorus. Šajā projektā mehānisms, kas regulē radiatora vārstu, tiek veidots pašrocīgi, izmantojot universālus mehānismus un detaļas.



3.3.1. att. Radiatora vārsta regulējošais motors.

Tā kā radiatora vārsts ir pagriežams, tad ir vajadzīgs rotora motors, kas spēj pagriezties par 90 grādiem. Šajā projektā radiatora regulators tiek veidots priekš vecā tipa radiatora vārstiem, kas ir vienkārši ir pagriežami par 90 grādiem no stāvokļa, kad radiatoram pilnīgi pieplūst sistēmā esošais ūdens, līdz stāvoklim, kad ūdens nepieplūst nemaz. Lai arī šis mehānisms ir ļoti vienkāršs, tajā pašā laikā, vārsts ir diez gan stingrs un, lai to pagrieztu, ir nepieciešamas pietiekami spēcīgs motors. Šajā projektā tiek izmantots 200 soļu rotora motors,

kura cena ir aptuveni 25 EUR⁶. Motora korpuss ir iestiprināts nekustīgā koka statīvā (3.3.1. att.), kā arī tika pielāgots esošā radiatora vārsta slēdzis, izurbjot tam caurumus abās pusēs, pie kuriem piestiprināt savienojumu ar motora rotējošo daļu.

Lai kontrolētu soļu motoru, ir nepieciešams motora kontrolieris. Šajā projektā tiek izmantots **TB6600** soļu motora kontrolieris, kur cena ir aptuveni 6 EUR⁷. Šis kontroliera maksimālā izejas strāva ir 4A, kas ir pietiekami lielākajai daļai soļu motoru. Izvēlētais kontrolieris spēj regulēt motora ātrumu un kustības virzienu [16].

Lai kontrolētu motoru, tiek izmantoti divi valodā “Python” uzrakstīti skripti – viens atver radiatoru, tas ir, griež motoru vienā virzienā, bet otrs aizver radiatoru tā tad griež motoru otrā virzienā.

3.4. Gaismas un temperatūras mērīšana

Lai iegūtu informāciju par to, kādos apstākļos tiek veiktas termoregulācijas darbības un vienkārši pārraudzītu situāciju mājā, katrā istabā tiek mērīta temperatūra un gaisma.

Lai mērītu temperatūru, tiek izmantots **LM75A** digitālais temperatūras sensors un tā mērapjoms ir no -55°C līdz +125 °C ar precizitāti +3 °C [18]. Šī sensora cena ir aptuveni 1 EUR⁸. Sensors temperatūras vērtību atgriež nokodētu binārā formā 9 bitu garumā, kuru var atkodēt un iegūt decimālu vērtību Celsija grādos pēc noteikta algoritma, kura pirmkods ir brīvi pieejams dažādos atvērtā pirmkoda projektos.

Sākotnēji bija paredzēts, ka gaismas mērīšana notiks luksos, kas ir SI sistēmas mērvienība apgaismojuma mērīšanai, tomēr iegādātais gaismas sensors spēj veikt tikai binārus mērījumus un pateikt to, vai gaisma ir (1) vai nav (0). Šis fakts varētu izmainīt mašīnmācīšanās modeļa apmācības rezultātus, jo apgaismojuma apzīmēšana luksos daudz precīzāk iezīmē apstākļus, kādos ir veikta konkrētā darbība. Konkrētais sensors vasaras sezonā, kad tiek veidots šis projekts, rāda 1 lielāko diennakts laiku, atskaitot naktīs. Šajā projektā tiek izmantots gaismas sensors **LM393**, kura cena ir aptuveni 1 EUR⁹.

⁶ <https://www.aliexpress.com/item/Planetary-gear-Box-Nema-17-Stepper-Motor-All-Ratio-Geared-For-3D-Printer-Extruder-mechanical-arm/32839229127.html>

⁷ <https://www.aliexpress.com/item/New-arrival-42-57-86-TB6600-stepper-motor-driver-32-segments-upgraded-version-4-0A-42VDC/32737927184.html>

⁸ <https://www.ebay.com/itm/I2C-Interface-Development-Board-Module-LM75A-Temperature-Sensor-For-Raspberry-Pi/162869073339?hash=item25ebc0d5bb:g:LZ0AAOSwh1paH8xy>

⁹ https://www.alibaba.com/product-detail/LM393-Photosensitive-Resistance-Sensor-Module-Light_60789821723.html?spm=a2700.7724838.2017115.99.72036307C7FBam&s=p

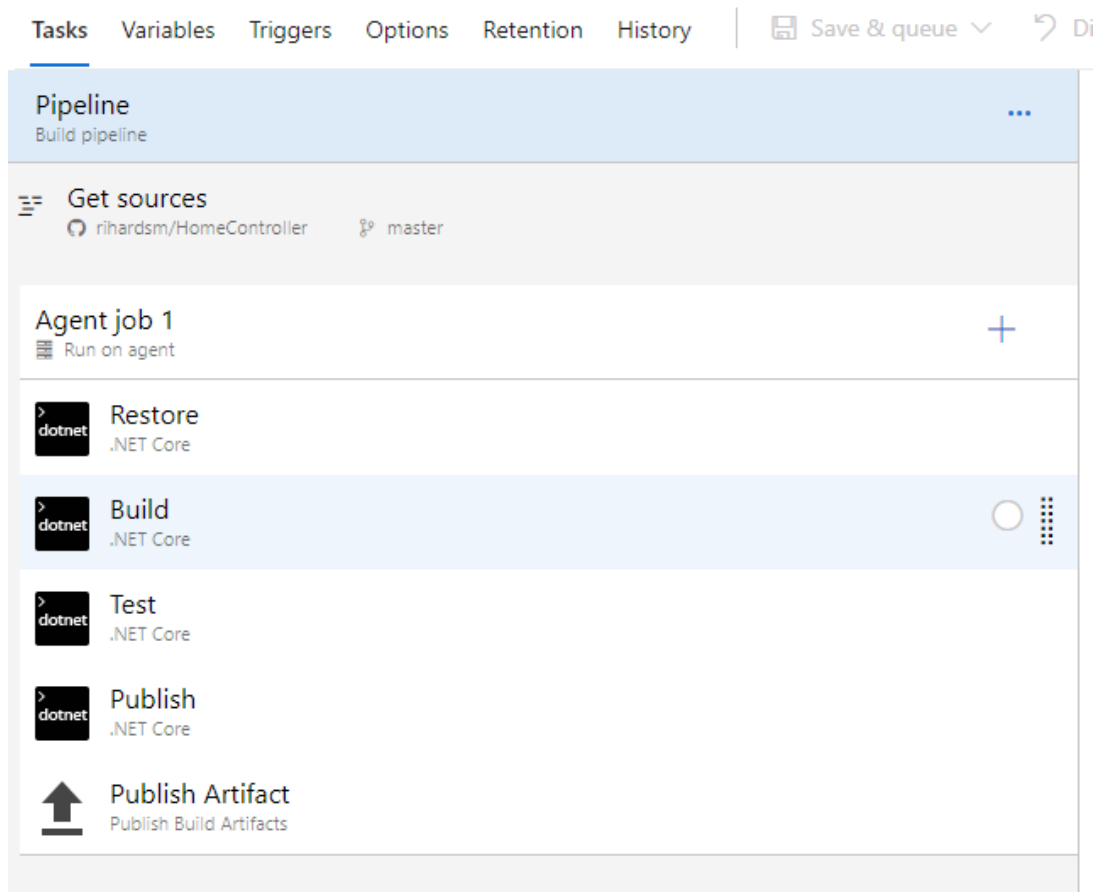
3.5. Kontroles un pārraudzības lietotne

Lai kontrolētu termoregulācijas sistēmu, tika izstrādāta tīmekļa lietotne, kurā iespējams ērti pārskatīt aktuālos sensoru rādījumus, redzēt stāvokli regulējošajām ierīcēm, kā arī izsaukt logu un radiatoru atvēršanu un aizvēršanu. Papildus iespējams apskatīt uzkrātos datus par sistēmas regulēšanas vēsturi, kas vēlāk tiks izmantoti mašīnmācīšanās modeļa apmācībai.

Lietotne tika veidota kā “Microsoft .NET core” vienas lapas lietotne (Single page application), kuras klientu puses kods ir veidots, izmantojot “Angular” ietvaru. Lietotnes pamats tika izveidots, izmantojot komandu “dotnet new angular”, kas ir iespējota, ja uz darbmašīnas ir uzinstalēta “.NET Core” 2.1 versija. Šī komanda izveido gatavu projektu uz kura bāzes iespējams tālāk veidot sev nepieciešamo lietotnes funkcionalitāti.

Lai uzkrātu datus tālākai mašīnmācīšanās modeļa apmācībai un darbību atsekošanai, ir izveidota “Microsoft SQL” datu bāze, kura arī tiek mitināta uz “Azure” mākoņplatformas. Ir izvēlēts pamata abonementa plāns, kas paredz datu bāzes izmēru līdz 2GB. Aptuvenās datubāzes uzturēšanas izmaksas mēnesī ir 4.50 EUR.

Tīmekļa lietotne tiek mitināta uz “Microsoft Azure” mākoņplatformas, izmantojot “Azure for students” abonēšanas plānu, līdz ar to, šīs lietotnes uzturēšana mākonī ir bez maksas. Lai optimizētu lietotnes izstrādes un uzturēšanas procesu, tika ieviests nepārtrauktās integrācijas / nepārtrauktās piegādes (Continuous integration / Continuous deployment) kanāls, izmantojot “Azrue DevOps” platformu. Šis rīks ļauj koncentrēties tikai uz pirmkoda radīšanu, ļaujot nedomāt par to, kā notiks lietotnes publicēšana. Lai sāktu lietot šo programmatūras piegādes mehānismu, ir jāizveido jauns “Azure DevOps” projekts un šis projekts jāsaista ar pirmkoda versiju kontroles sistēmu, kas šajā gadījumā ir “GitHub”. “Azure DevOps” portālā iespējams pilnībā pārvaldīt programmatūras projektus, tajā ir integrēta versiju kontroles sistēma, kas vizuāli atspoguļo visas darbības kādā konkrētā koda krātuvē. Portālā ir iestrādāta arī projektu plānošanas un pārvaldības funkcionalitāte “Boards”, tas ir, “Azure DevOps” portālā ir iespējams veidot sprintus jeb iterācijas, veidot un piešķirt komandas locekļiem dažādus darbu vienumus, kā lietotāju stāstus, uzdevumus un citus.



3.5.1. att. Programmas publicēšanas uzdevumu saraksts

Lai automatizētu programmas koda nokļūšanu produkcijā, tiek izveidota automātiska uzdevumu plūsma (3.5.1. att.), kas veic visas nepieciešamās darbības, kas vajadzīgas, lai publicētu programmu uz mākoņplatformas. Tā kā versiju kontroles sistēma “Git” šajā projektā ir savienota ar “Azure DevOps”, pēc katru kodu izmaiņu nonākšanas versiju kontroles galvenajā zarā, tiek izpildīti augstākminētie darbi un lietotne tiek automātiski publicēta uz mākoņplatformas. Pēc katra publicēšanas mēģinājuma projekta īpašniekam tiek nosūtīta e-pasta vēstule ar mēģinājuma informāciju.

```

<dx-data-grid [dataSource]="actionHistories"
  keyExpr="id">
  <dxo-export [enabled]="true" fileName="ActionHistory" [allowExportSelectedData]="true"></dxo-export>
  <dx-column dataField="roomString" caption="Istaba"></dx-column>
  <dx-column dataField="actionTypeString" caption="Darbība"></dx-column>
  <dx-column dataField="time" dataType="date" [format]="{ type: 'dd.MM.yyyy HH:mm' }" caption="Laiks"></dx-column>

```

3.5.2. att. Datu eksporta iespējas pievienošana Devextreme DataGrid tabulai

Kontroles un pārvaldības lietotnei ir izstrādāti divi galvenie skati – pārskata un kontroles, skats, kur iespējams redzēt tagadējo temperatūru katrā istabā, kā arī darbību vēstures skats, kur iespējams redzēt datus par visām iedzīvotāju veiktajām darbībām un apstākļiem kādos tās veiktas. Darbību vēstures skata tabula ir veidota, izmantojot “Devextreme” kontroli “GridList”, kas ietver jau gatavu ļoti plašu funkcionalitāti, kā piemēram, ar vienu koda rindiņu

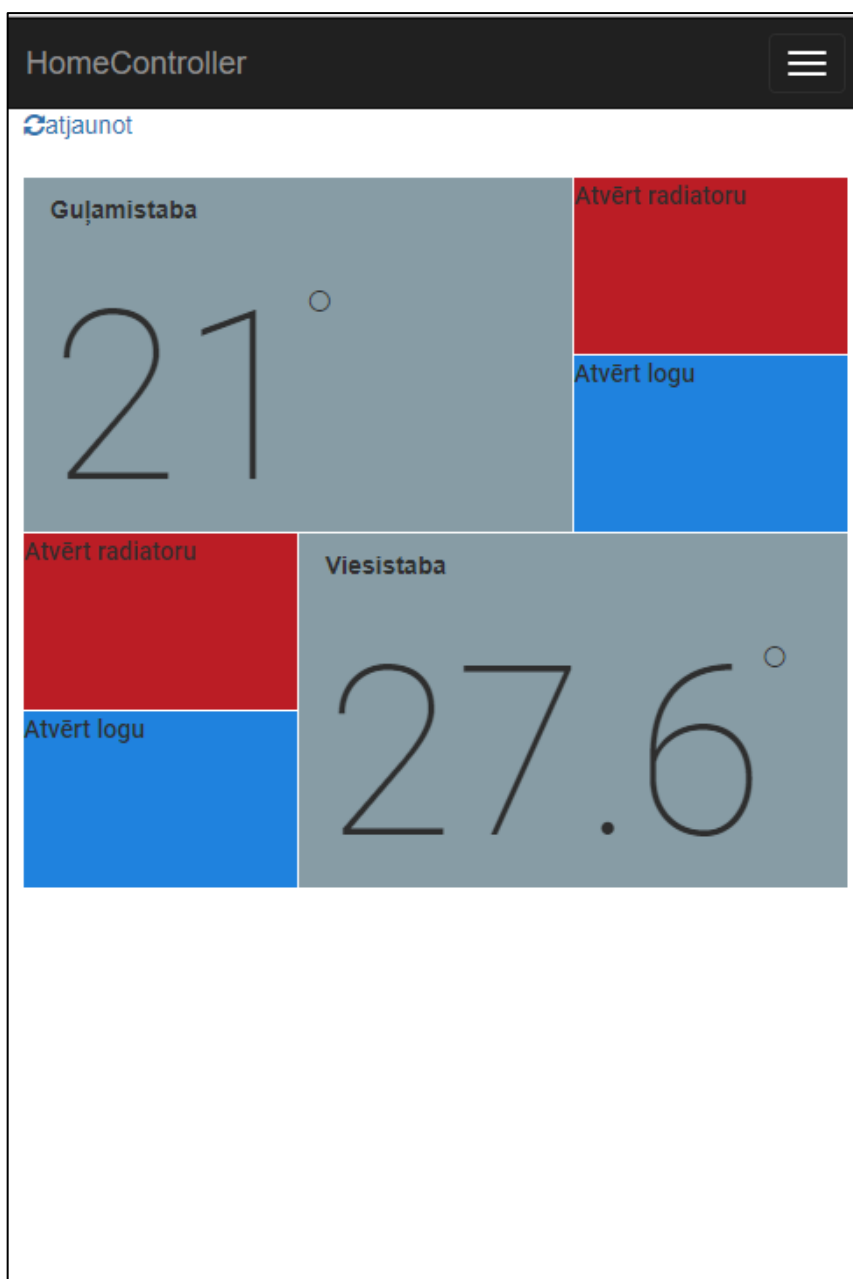
(3.5.2. att.) var pievienot iespēju eksportēt datus uz “Excel” vai “Csv” formāta failiem, lai tālāk tos izmantotu mašīnmācīšanās modeļa apmācībai. “Devexteme” rīkus bezpeļņas projektiem var izmantot bez maksas.

Istaba	Darbība	Laiks	Temp.	Gaisma	Logs darbības l...	Sildītājs darbīb...
Gujamistaba	Atvērt logu	23.05.2019 07:09	22.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:51	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 19:32	24.00	Tumšs	Atvērts	Izslēgts
Dzīvojamā istaba	Atvērt logu	22.05.2019 19:30	27.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:23	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 19:20	24.00	Tumšs	Atvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:19	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:14	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 19:14	24.00	Tumšs	Atvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:14	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Atvērt logu	22.05.2019 19:13	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 19:12	24.00	Tumšs	Atvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 18:59	24.00	Tumšs	Aizvērts	Izslēgts
Gujamistaba	Aizvērt logu	22.05.2019 18:59	24.00	Tumšs	Aizvērts	Izslēgts

3.5.3. att. Darbību vēstures skats

Pārskata un kontroles jeb sākuma skats tika veidots tā, lai tas būtu ērti pārskatāms uz viedtālruna vai ierīces ar mazu ekrānu (3.5.4. att.), paredzot, ka to galvenokārt lieto, piekļūstot lietotnei caur viedtālruni, kā arī ir plānots dzīvoklī izveidot vēl vienu termoregulācijas sistēmas sastāvdaļu – statisku sienas ekrānu, kurā visu laiku rādās istabu temperatūras un ir iespēja veikt logu un radiatoru regulēšanas darbības.

Kontroles skata pogu teksti rādās atkarībā no tā, kādā stāvoklī pašlaik ir radiatori un logi, tas ir, ja logs pašlaik ir atvērts, tad pogas teksts ir “Aizvērt logu”, bet, ja logs ir aizvērts, tad pogas teksts ir “Atvērt logu”. Pogu krāsas ir izvēlētas pēc klasiskās krāsu asociācijas par to, ka sarkans asociējas ar uguni, tātad siltumu, līdz ar to, ar sarkano pogu regulē radiatorus. Ar gaiši zilo krāsu parasti asociējas ūdens vai ledus, kas ir arī vēsuma simbols, līdz ar to, ar zilo pogu regulē logu atvēršanu vai, ja tāds būtu pieejams, tad ar šo pogu varētu regulēt gaisa kondicionieri.



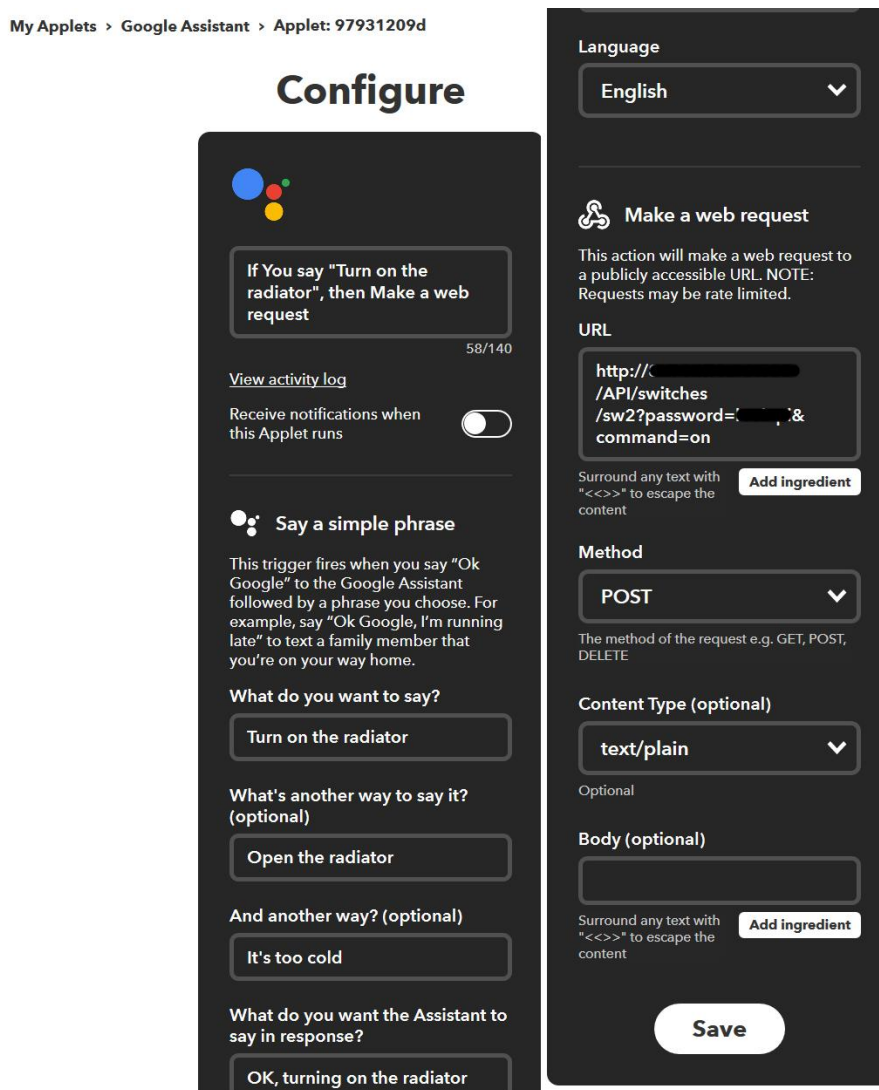
3.5.4. att. Pārskata un kontroles jeb sākuma skats

Kontroles lietotnei ir izstrādātas sekojošas metodes:

- 1) **GetActionHistories**, kas atgriež sarakstu ar visām veiktajām darbībām, sakārtotu pēc darbības datuma dilstošā secībā;
- 2) **SetAction**, kas kā parametru saņem objektu, kas satur datus par tikko veiktu darbību un saglabā to datu bāzē;
- 3) **GetStatus**, kas atgriež datus par to, kādas temperatūras ir istabās un kādos stāvokļos ir logi un radiatori;
- 4) **ToggleWindow**, kas kā parametru saņem istabas tipu, tas ir, guļamistaba vai viesistaba un veic darbību atkarībā no loga stāvokļa šajā istabā – ja logs tajā brīdī ir aizvērts, tad to atver un pretēji;
- 5) **ToggleHeater**, kas darbojas tā pat, kā **ToggleWindow**, tikai regulē radiatorus.

3.6. Virtuālais asistents

Attīstoties mākslīgā intelekta apakšnozarei – runas atpazīšanai, aizvien pieejamāki un izplatītāki kļūst dažādi virtuālie asistenti. Mūsdienās katrai konkurētspējīgai komerciālai platformai ir izstrādāts savs virtuālais asistents, piemēram, kompānija Apple savās ierīcēs ir iestrādājusi Siri, Microsoft piedāvā lietotājiem izmantot Cortana. Virtuālie asistenti tiek iestrādāti dažādās pakalpojumu sniedzēju lapās, kur šie rīki spēj aizvietot cilvēku darbu, palīdzot ar meklējamā satura vai cita veida informācijas atrašanu.



3.6.1. att. “IFTTT” sīkietotnes konfigurācija

Mājas automatizēšanai, ļoti labi noder virtuālie asistenti, kas ir iestrādāti skaļruņos un ir labs mājas dizaina un funkcionāls papildinājums. Šajā projektā tiek izmantots “Google Home Mini”. “Google Home” ir kompānijas “Google” veidota viedo skaļruņu sērija, kuros ir iestrādāts “Google Assistant” virtuālais asistents, līdz ar to, ir iespējams ar runu tam dot komandas, kā arī saņemt verbālas atbildes. Galvenā virtuālā asistenta loma šajā projektā ir tā,

ka tas sniedz iespēju ar runu atvērt logus un radiatorus mājās. “Google Assistant” var vienkārši savienot ar ļoti daudzām mājas viedierīcēm, kurās ir iestrādāta šāda funkcija, kā piemēram, viedā veļas mašīna vai viedā spuldze. Tomēr šī projekta ietvaros veidotās viedierīces, nav “Google Assistant” atbalstīto un atpazīto ierīču sarakstā. Šādos gadījumos ir jāizmanto “IFTTT”. “IFTTT” ir tīmeklī balstīts bezmaksas pakalpojums, kas ļauj izveidot dažādas sīklīetotnes, kas ir kā ķēdes reakcijas dažādiem saistītiem notikumiem (3.6.1. att.). Katrs sistēmā iesaistītais “Raspberry Pi” dators ir pieslēgts tīklam un tajā ir uzstādīts tīmekļa serveris, tātad tam var piekļūt ar noteiktu IP adresi. Katra datora serverim ir izveidoti API, kas veic noteiktas darbības, piemēram, atver logu. Tātad ir jāizveido loģiskā ķēde:

- 1) Mājas iedzīvotājs “Google Home” asistentam dod balss komandu “atvērt logu”;
- 2) “Google Assistant” šo komandu saprot un izsauc attiecīgo “IFTTT” sīklīetotni;
- 3) Attiecīgā “IFTTT” sīklīetotne izsauc tai norādīto API.
- 4) Uz “Raspberry Pi” uzstādītais serveris saņem pieprasījumu un izpilda komandas.

3.7. Mašīnmācīšanās modeļa apmācība

Lai padarītu šo termoregulācijas sistēmu viedu, ir plānots apmācīt mašīnmācīšanās modeli, izmantojot uzkrātos datus par lietotāja veiktajām darbībām. Mašīnmācīšanās nodrošina to, ka dators veic noteiktas darbības, nevis tikai sekojot precīzi ieprogrammētām instrukcijām, bet arī analizējot apmācības procesā saņemtos datus, kuros precīzi norādīts kādiem ievaddatiem atbilst kādi izvaddati. Piemēram, šajā gadījumā, ja termoregulācijas sistēmas lietotājs atver logu, kad istabā ir 24 grādi pēc celsija, pulkstenis ir 8:40 no rīta un apgaismojuma daudzums telpā ir 800 lx, kā arī sildīšanas elements jeb radiators ir izslēgts, tad sistēma darbību vēstures tabulā ieraksta rindu ar datiem, kurus vēlāk saņem mašīnmācīšanās modelis. Apmācības datu struktūra ir attēlota tabulā (3.7.1 tabula). Lai izveidotu veiksmīgu modeli, ir nepieciešami gan apmācības dati, gan arī testa dati. Testa datiem arī ir jābūt no mājas iedzīvotāju reāli veikto darbību saraksta, tikai tie nedrīkst būt tieši tie paši ieraksti, kas tika izmantoti kā apmācības dati, jo pastāv iespēja, ka modelis var iegaumēt atbildes uz kādiem jautājumiem, līdz ar to, testēšanas process dos maldinošus rezultātus.

3.7.1 tabula
Mašīnmācīšanās modeļa apmācības datu piemērs

Ievaddati			Izvaddati
Temperatūra	Laiks	Gaisma	Veicamā darbība
24	8:40	1	Atvērt logu

Ir dažādi mašīnmācīšanās algoritmi, kas risina dažāda veida problēmas jeb atbild uz dažādiem jautājumiem. Pamatā ir pieci jautājumi [12], uz kuriem atbild mašīnmācīšanās algoritmi:

- 1) **A vai B?** Lai atrisinātu šo jautājumu, tiek izmantoti klasifikācijas algoritmi (*classification algorithms*). Ja ir iespējami vairāki atbilžu varianti, tad tiek izmantoti multiklašu klasifikatoru algoritmi (*multiclass classification*). Piemēram, kas veikalam pienesīs vairāk jaunu klientu – 5 EUR dāvanu kartes vai 25% atlaides;
- 2) **Vai tas ir dīvaini?** Lai atbildētu uz šo jautājumu, tiek lietoti anomāliju noteikšanas algoritmi (*anomaly detection*). Piemēram, vai tikko veiktais darījums internetbankā atbilst tiem darījumiem, kādi tiek veikti parasti, tas ir, vai tas nav aizdomīgs?;
- 3) **Cik daudz?** Uz šo jautājumu atbild regresijas algoritmi (*regression*). Piemēram, kāda būs uzņēmuma šī pusgada peļņa vai arī, kāds būs laiks šajās brīvdienās?;
- 4) **Kā ir organizēti dati?** Datu organizētības noteikšanai izmanto pudurošanas (*clustering*) algoritmus. Piemēram, kuriem kino apmeklētājiem patīk viena veida filmas?;
- 5) **Ko darīt tālāk?** Šo problēmu risina stimulētās apmācības algoritmi (*reinforcement learning*). Piemēram, pašbraucošā mašīna pamana, ka luksoforā deg dzeltenā gaisma – ko tai darīt tālāk – paātrināties, lai pārbrauktu pāri ne pie sarkanās gaismas vai arī bremsēt.

Šajā gadījumā varētu izvēlēties divus no algoritmu veidiem – **A vai B?**, jo, iestājoties noteiktiem apstākļiem, ir iespējamās vairākas darbības, kā piemēram, atvērt logu (A), aizvērt logu (B), atvērt logu un izslēgt radiatoru (C) un citas darbību kombinācijas. Tomēr arī algoritms no problēmu klases **Ko darīt tālāk?** risinātu šo problēmu, jo iestājoties noteiktiem apstākļiem ir termoregulācijas sistēmai ir jāzina, ko darīt tālāk. Izplatītākie mašīnmācīšanās modeļu apmācības paņēmieni ir:

- 1) **Pārraudzītā apmācība (supervised learning)** – modelim tiek ievadīti apmācības dati, kas satur ieejas parametrus un tiem atbilstošo izejas rezultātu;
- 2) **Nepārraudzītā apmācība (unsupervised learning)** – algoritmi meklē sakritības datus un mēģina analizēt datu struktūras bez konkrētām atbildēm apmācības datus;
- 3) **Stimulētā apmācība (reinforcement learning)** – algoritms izvēlas darbības soļus pēc apmācības procesā gūtās pieredzes tā, lai konkrētajā situācijā iegūtu labāko rezultātu.

Termoregulācijas sistēmas mašīnmācīšanās modeļa apmācībai vispiemērotākais apmācības veids būtu pārraudzītā apmācība, kas tiks veikta par apmācības datiem izmantojot reģistrētās lietotāju darbības, tomēr jāpaiet ilgam laikam aktīvi lietojot sistēmu, lai iegūtu pietiekami apjomīgu datu kopu. Vajadzētu uzkrāt datus ar darbībām visos gadalaikos, lai modelis tiktu apmācīts arī ar tādiem datiem, kad lietotājs regulē apkuri mājās.

Tiešsaistē ir pieejami vairāki atvērtā pirmkoda vai bezmaksas mašīnmācīšanās rīki un bibliotēkas, piemēram, “Scikit-learn”, “TensorFlow” un “Weka”, kurus var izmantot, veidojot dažādas sistēmas, kā arī apgūstot mašīnmācīšanās pamatus un meklējot atbilstošāko risinājumu kādai problēmai. Šie rīki sniedz iespēju apmācīt mašīnmācīšanās modeļus, pielietojot izplatītākās apmācības metodes un algoritmus.

Apgūstot mašīnmācīšanās pamatus, ir svarīgi izvēlēties vienkāršus rīkus, kas varbūt negarantē iespēju atrisināt vissarežģītākās problēmas, taču iesākumā nodrošina vienkāršas saskarnes, saprotamas pamācības kā arī datu apmācības procesa un datu vizualizācijas iespējas. Apskatot internetā pieejamus rīku salīdzinājumus [19], kurus varētu izmantot apgūstot mašīnmācīšanās teoriju un praksi, kā arī uzklusot pieredzējušu cilvēku ieteikumus, tika izvēlēta “Weka” mašīnmācīšanās programmatūra.

3.7.1. Weka

Nosaukums “Weka” ir cēlies no tikai Jaunzēlandes salās sastopama nelidojoša putna ar zinātkāru dabu ar nosaukumu veka.

“Weka” ir mašīnmācīšanās algoritmu kopums datizraces un datu analīzes uzdevumiem. Tajā ir iekļauti datu sagatavošanas, klasifikācijas, regresijas, pudurošanas, asociatīvo likumu ieguves un vizualizācijas rīki. “Weka” atbalsta dziļo mašīnmācīšanos (*deep learning*)

Šīs programmatūras apguvei ir pieejami bezmaksas tiešsaistes kursi [20]. Tiešsaistes kursi ir pieejami gan teksta gan video formātā. Šajās apmācībās var uzzināt, kā sagatavot datus, pielietot dažādus mašīnmācīšanās algoritmus sagatavotajiem datiem, kā arī testēt izveidotos modeļus un vizualizēt tos.

3.7.2. Datu sagatavošana programmai “Weka”

Lai sagatavotu datus mašīnmācīšanās modeļa apmācībai, tika veikts eksports no darbību vēstures tabulas uz “csv” formāta failu. Mašīnmācīšanās algoritmi primāri ir veidoti darbam ar skaitliskiem un strukturētiem datiem. Svarīgi datu jēdzieni ir **instance**, kas ir viena datu rinda

jeb viens neatkarīgs notikums problēmas kontekstā un **atribūts**, kas ir datu kolona jeb kāda īpaša pazīme, kas nodrošina datu jēgu [21].

Atribūtiem var būt dažādi datu tipi:

- 1) Real – reāli skaitļi;
- 2) Integer – veseli skaitļi;
- 3) Nominal – kategorizētas vērtības, piemēram, istabas tips (guļamistaba vai dzīvojamā istaba);
- 4) String – tekstuālām vērtībām.

“Weka” atbalsta datu ielādi “ARFF” (*Attribute-Relation File Format*) formātā [22]. “ARFF” failiem ir divas atdalītas sadaļas. Pirmā daļa ir galvene “Header”, kurai seko datu sadaļa “Data”. Galvene satur sakarības (*Relation*) nosaukumu, kā arī atribūtu un to tipu sarakstu (3.7.2.1. att.).

Datu sadaļa sākas ar pazīmi “@data”. Katra datu instance tiek rakstīta jaunā rindā un atribūtu vērtības ir jāatdala ar komatu vai “tab” taustiņu. Rindās atribūtu vērtībām ir jāparādās tādā pašā secībā, kā tika deklarēti atribūti galvenē. Ja datu instance satur kādu nezināmu vērtību, tad tajā vietā raksta jautājuma zīmi “?”. “ARRF” faila formāts atbalsta arī instanču svaru norādīšanu, gadījumos, ja datos ir identiskas instances. Instances svaru var norādīt figūriekavās rakstot veselu skaitli, piemēram, “{4}”.

```
≡ ActionHistory.arff ×
1 @RELATION "thermoregulation action"
2
3 @ATTRIBUTE room {0, 1}
4 @ATTRIBUTE action {1,2,3,4}
5 @ATTRIBUTE time DATE "dd.MM.yyyy HH:mm"
6 @ATTRIBUTE temperature REAL
7 @ATTRIBUTE light {0,1}
8 @ATTRIBUTE windowState {0,1}
9 @ATTRIBUTE radiatorState {0,1}
10
11 @DATA
12 0,1,"23.05.2019 07:09",22,1,0,0
13 0,2,"22.05.2019 19:32",24,1,1,0
14 1,1,"22.05.2019 19:30",27,1,0,0
15 0,1,"22.05.2019 19:51",24,1,0,0
16 0,1,"22.05.2019 19:23",24,1,0,0
17 0,2,"22.05.2019 19:20",24,1,1,0
18 0,1,"22.05.2019 19:19",24,1,0,0
19 0,1,"22.05.2019 19:14",24,1,0,0
20 0,2,"22.05.2019 19:14",24,1,1,0
21 0,1,"22.05.2019 19:14",24,1,0,0
22 0,1,"22.05.2019 19:13",24,1,0,0
23 0,2,"22.05.2019 19:12",24,1,1,0
24 0,2,"22.05.2019 18:59",24,1,0,0
25 0,2,"22.05.2019 18:59",24,1,0,0
```

3.7.2.1. att. Datu sagatavošana programmai “Weka”

Pēc tam, kad dati ir sagatavoti “ARFF” formātam, attiecīgo failu var saglabāt ar “.arff” paplašinājumu, lai importētu programmā “Weka”. Aptuveni divu nedēļu ilgā termoregulācijas sistēmas darbības laikā, tika uzkrātas 42 datu instances.

3.7.3. Datu imports programmā “Weka”

Kad eksportētie dati ir pielāgoti “ARFF” formātam un ir iegūts “.arff” fails, datus var ieimportēt programmā “Weka”. Pēc datu importa programma piedāvā dažāda veida ievadīto datu analīzi un vizualizāciju, piemēram, var apskatīties, kāda ir zemākā, augstākā un vidējā temperatūra vai kādas biežumu attiecības darbību veikšanai pa istabām (3.7.3.1. att.).

The screenshot shows the Weka Explorer application window. The 'Preprocess' tab is active. The 'Filter' section shows 'None' selected. The 'Current relation' section displays 'Relation: thermoregulation action', 'Instances: 43', 'Attributes: 7', and 'Sum of weights: 43'. The 'Attributes' list includes: 1 room, 2 action, 3 time, 4 temperature (checked), 5 light, 6 windowState, and 7 radiatorState. The 'Selected attribute' section shows 'Name: temperature', 'Missing: 0 (0%)', 'Distinct: 11', and 'Type: Numeric Unique: 6 (14%)'. A table of statistics is displayed:

Statistic	Value
Minimum	19
Maximum	28
Mean	22.528
StdDev	2.68

Below the table, the class is set to 'radiatorState (Nom)'. A histogram shows the distribution of the 'temperature' attribute, with bars at 19 (count 21), 23.5 (count 17), and 28 (count 5). The status bar at the bottom shows 'OK' and a 'Log' button.

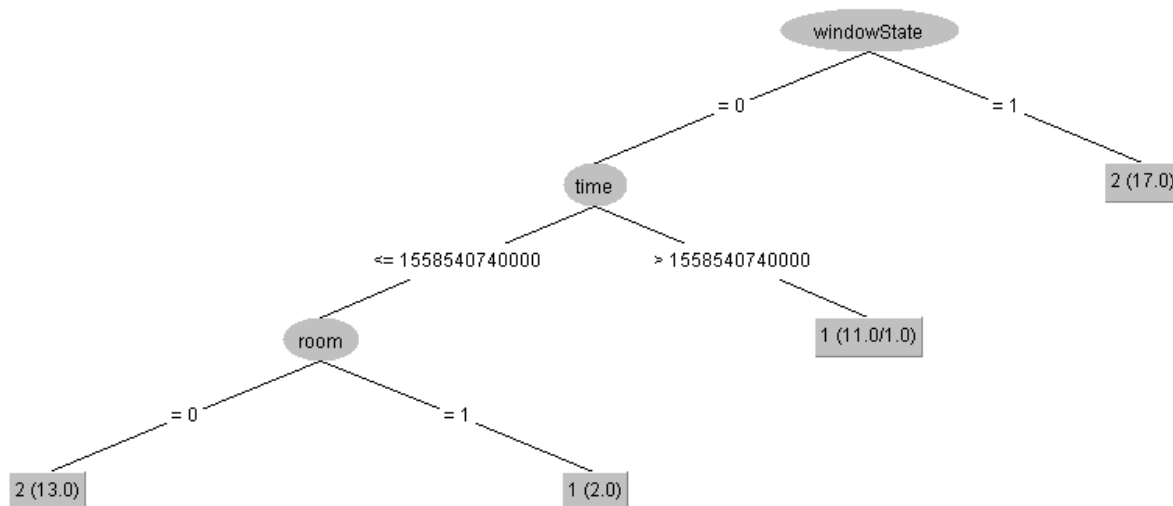
3.7.3.1. att. Programma “Weka” pēc datu importa

Pēc veiksmīga datu importa un apskates, var sākt datiem pielietot dažādus mašīnmācīšanās algoritmus. Iepriekš, aprakstot dažādas mašīnmācīšanās metodes, tika secināts, ka šai problēmai vislabāk varētu derēt algoritmi no klasifikācijas problēmu klases.

3.7.4. Klasifikatora algoritma pielietošana

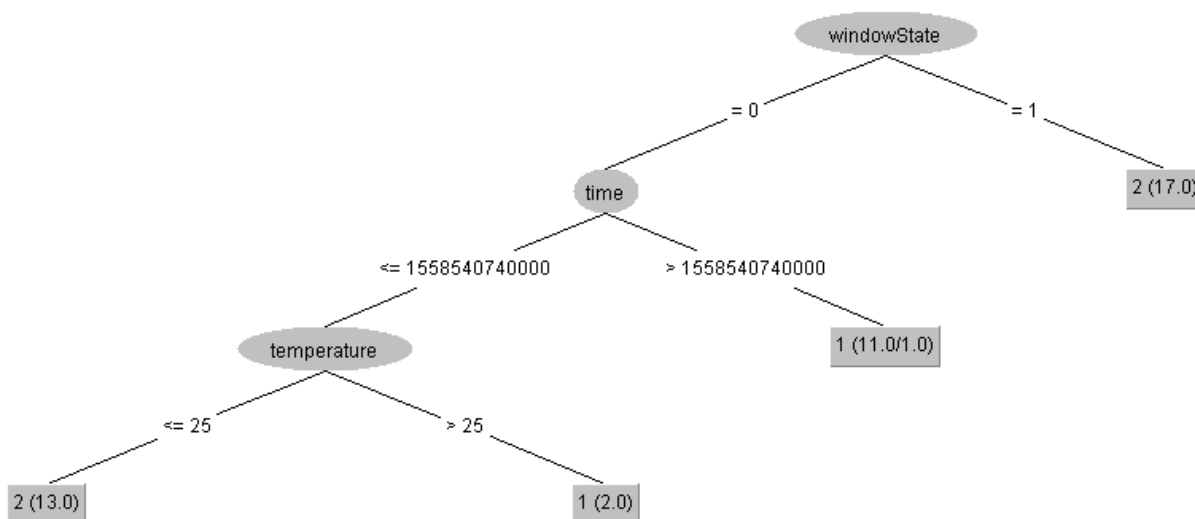
Pielietojot klasifikācijas algoritmu “J48” jeb “C4.5” algoritma jaunākā versija (4.8) pārrakstīta valodā “Java”, kas ir algoritms, kas veido lēmumu kokus, tiek iegūts lēmumu koks importētajiem datiem (3.7.4.1. att.). Uzģenerētais koks parāda, ka lēmumi tiek pieņemti nepilnīgi un tajā netiek ņemti vērā svarīgākie atribūti, tas ir, temperatūra un gaisma. Kā arī dziļākais bērns, kas nav lapa, ir atribūts “istaba”, pēc kura tiek pieņemts neloģisks lēmums – ja

konkrētā instance ir dzīvojamā istabā veikta darbība, tad logs ir jāatver, bet, ja tā ir guļamistabā veikta darbība, tad logs ir jāaizver. Šie novērojumi vedina uz domu, ka ir jāatdala dati par istabām, katrai istabai jāveido savs modelis, kā arī istabas atribūts jāizņem no datiem. Koka lapās ir norādīta veicamā darbība (skaitlis ārpus iekavām) un pareizo testēšanas rezultātu skaits šajā lapā (pirmais skaitlis iekavās), kā arī tas, cik reizes tika kļūdaini minēts, ka šī lapa ir pareizais risinājums (otrais skaitlis lapā, ja bija kāds kļūdainis minējums).



3.7.4.1. att. J48 algoritma izveidotais klasifikatora koks ar sākotnējiem datiem.

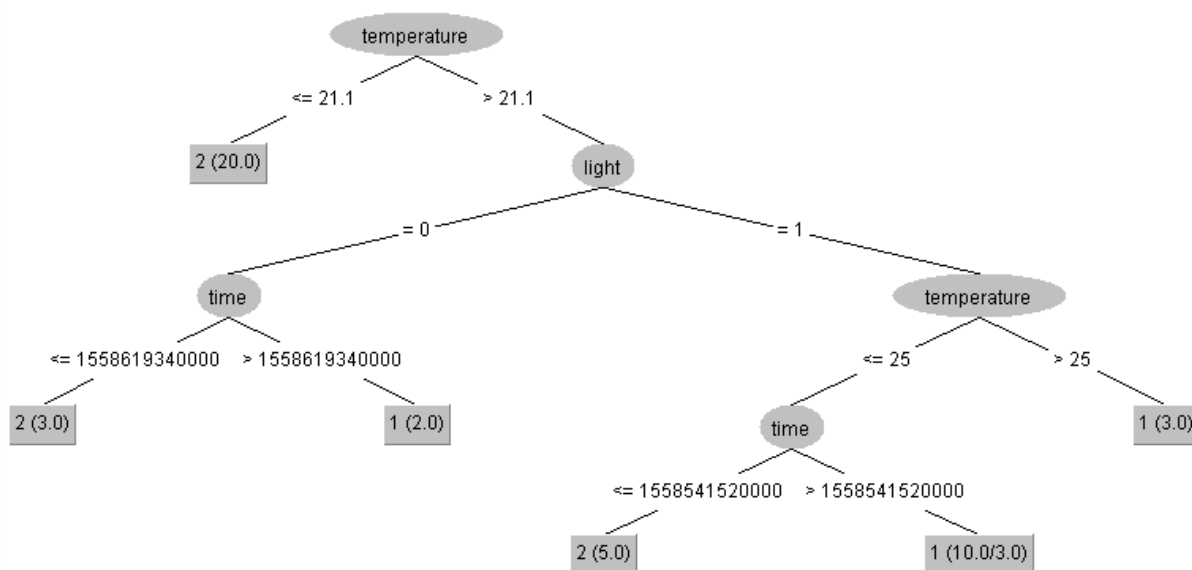
Pielietojot to pašu algoritmu sadalītiem datiem, tiek iegūts jau loģiskāks lēmumu koks, kurš jau ņem vērā arī galveno parametru – temperatūru.



3.7.4.2. att. Guļamistabas datu lēmumu koks (pēc tam, kad dati ir sadalīti pa istabām)

Arī pēc datu atdalīšanas ir pamanāmas problēmas (3.7.4.2. att.), piemēram, lielu logu ieņem tas, kāds ir bijis loga stāvoklis darbības laikā, tas ir, ja logs ir atvērts, tad tas noteikti izjāaizver, jo citas iespējas nav, bet, ja logs ir atvērts, tad tas jāatver. Šīs datu instances neparedz citas darbības, kā tikai “atvērt” un “aizvērt”, līdz ar to ir nepieciešama trešā iespējamā darbība – nedarīt neko.

Nepapildinot darbību sarakstu, trešo iespējamo darbību varētu panākt, izņemot no atribūtu saraksta loga stāvokli. Tādējādi tiek iegūts tas, ja modelis pieņem lēmumu, ka jāatver logs, bet logs jau ir atvērts, tad netiek darīts nekas.

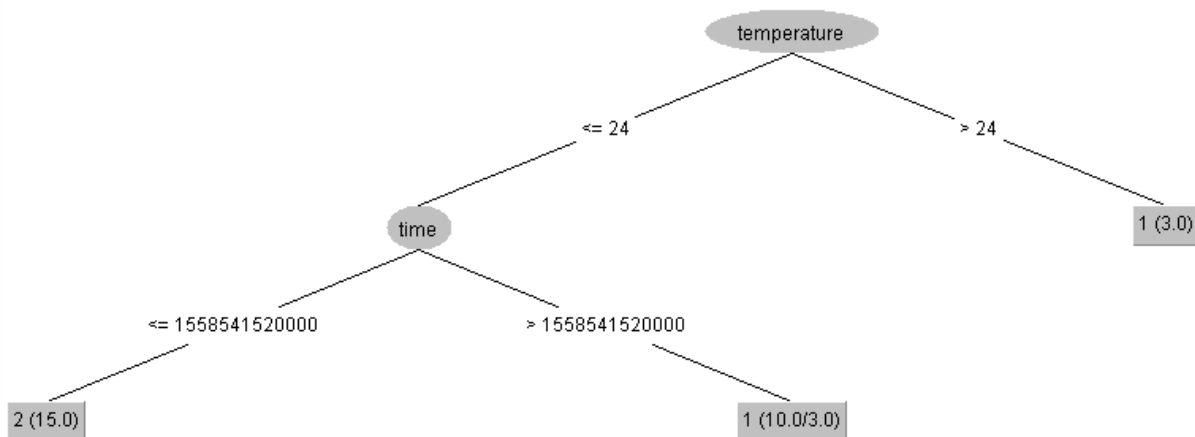


3.7.4.3. att. Lēmumu koks, kad nav zināms loga stāvoklis darbības laikā

Kad algoritms nezina, kāds ir precīzs loga stāvoklis darbības laikā, tiek iegūts daudz detalizētāks un interesantāks lēmumu koks, kas arī nodrošina nekā nedarīšanas iespēju, ja pieņemtais lēmums jau ir spēkā. Ar šādu atribūtu sarakstu tikai iegūts koks ar visvairāk virsotnēm un lielāko lapu skaitu. Šī klasifikatora koka (3.7.4.3. att.) lēmumu precizitāte ir 83.7%. Lēmumu koks būtu vēl sarežģītāks un ar vairākām virsotnēm, ja būtu pieejami arī dati par laiku, kad ir apkures sezona. Dati pašlaik ir ļoti nepilnīgi un šāds modelis nevarētu normāli funkcionēt, jo tas nekad nav saskāries ar datiem, kas iegūti ziemas periodā, kad ir ieslēgta apkure, tomēr, pieņemot, ka pašlaik šie ir pilnīgākie pieejamie dati un šis ir modelis, kuru varētu izmantot termoregulācijas darbību paredzēšanai, tad nākamais solis apmācības procesā ir modeļa testēšana.

3.7.5. Izveidotā modeļa testēšana

Lai iegūtu reālistiskus testēšanas rezultātus, ir jāizmanto reāli praksē iegūti dati. Taču tie dati, kurus izmantot testēšanai nevar būt tie paši dati, kurus izmanto modeļa apmācībai. Ja testēšanu veic ar to pašu datu kopu, ar kuru tika veikta apmācība, tad modeļa precizitāte ir **93%**. Tomēr šis rādītājs nav ticams, jo tika testēts ar tiem pašiem datiem, pēc kuriem tika veidots konkrētais lēmumu koks. Lai iegūtu ticamākus rezultātus, tiks sadalīti pieejamie dati – 65% ierakstu (27 datu instances) tiks izmantoti apmācībai, bet atlikušie 35% ierakstu (15 datu instances) tiks izmantoti kā testa datu kopa. Pēc treniņa ierakstu skaita samazināšanas, tika pārgenerēts modelis, un tas tika uzģenerēts atšķirīgs no tā, kāds tika izveidots no pilnas datu kopas (3.7.5.1. att.). Šīs testēšanas rezultātā modelis uzrādīja 66% precizitāti. Lai izveidotu precīzāku lēmumu koku, kā arī precīzāk to notestētu, jāuzkrāj ievērojami lielāks datu apjoms.



3.7.5.1. att. Lēmumu koks pēc apmācības datu kopas samazināšanas

REZULTĀTI

Šī darba izstrādes rezultātā tika iegūta pilnīgi funkcionējoša termoregulācijas sistēma ar pamata funkcionalitāti, kas ļauj attālināti pārraudzīt istabas temperatūru divās dzīvokļa istabās – guļamistabā un viesistabā, kā arī regulēt temperatūru, atverot vai aizverot logus un apkures sezonas laikā arī atvērt un aizvērt radiatora vārstu. Lai pēc noteikta laika varētu apmācīt mašīnmācīšanās modeli un likt sistēmai iemācīties iedzīvotāju paradumus, tiek reģistrētas visas sistēmā veiktās darbības ar pazīmēm par darbības laikā esošajiem apstākļiem konkrētajā istabā, tas ir, kāda bija gaisa temperatūra un vai istabā bija gaišs. Lai nodrošinātu šo funkcionalitāti, tika izveidoti attālināti vadāmi automātiski logu atvērēji, kā arī attālināti vadāma radiatora vārsta galva. Sistēmas kontroli nodrošina divi tīklam pieslēgti “Raspberry Pi” minidatori, uz kuros uzstādīti Node.js serveri, “Google Home” viedais skaļrunis ar “Google Assistant”, kā arī tīmekļa lietotne, kas nodrošina saziņu ar visām iepriekš minētajām sastāvdaļām.

Lai veiksmīgāk izveidotu termoregulācijas sistēmu un noskaidrotu kāda pamata funkcionalitāte ir nepieciešama, tika apskatīti un aprakstīti dažādi gan komerciāli, gan nekomerciāli risinājumi gan kompleksām un gatavām sistēmām, gan arī atsevišķām sistēmas sastāvdaļām, kā piemēram, logu atvērējam. Interneta resursos nav atrodama neviena kompleksa termoregulācijas sistēma, kas vienlaicīgi regulētu vairākus temperatūru noteicošos faktorus – pārsvarā dominē tikai tādas sistēmas, kas regulē mājas apsildi.

Pielietojot mašīnmācīšanās algoritmu uzkrātajiem datiem par lietotāju darbībām, tika pamanītas dažādas nepilnības datos, kā, piemēram, katrai istabai, kurā tiek izmantota šī sistēma ir jāapmāca savs modelis, kā arī, apmācāmajam modelim nedrīkst teikt priekšā, kāds ir konkrētā kontroles elementa (logs vai radiators) pašreizējais stāvoklis, jo tad modelis izdara secinājumu – ja stāvoklis ir atvērts, tad tas ir jāaizver. Stāvokļa slēpšana no modeļa nodrošina arī iespēju neko nedarīt.

SECINĀJUMI

Apskatot dažādus Latvijas tirgū piedāvātos mājas termoregulācijas sistēmu risinājumus, var skaidri pamanīt, ka šie risinājumi savu funkciju pilda tikai apkures sezonas laikā, jo tiem ir iespēja kontrolēt tikai apkures sistēmu – apkures katlu vai radiatora vārstus.

Lai veiksmīgi apmācītu mašīnmācīšanās modeli, kas spētu dot noderīgus padomus vai pārņemt termoregulācijas procesus, sistēma cilvēkiem ir aktīvi jālieto vismaz vienu pilnu gadu, lai tiktu ievākti dati gan apkures sezonā, gan siltajā sezonā, gan starpsezonās. Pašlaik – 2019. gada maija beigās - termoregulācijas sistēma tiek aktīvi izmantota tikai aptuveni divas nedēļas, kuru laikā ir ievākti aptuveni 40 darbību ieraksti. Tomēr realitātē cilvēki negrib gaidīt veselu gadu un paši regulēt temperatūru mājās, bet gan vēlas norādīt vēlamo temperatūru dažādos laika intervālos un ļaut sistēmai strādāt. Modeļa apmācības procesā, darba autors secināja, ka, visticamāk, termoregulācijas sistēmu darbībām vispiemērotākās ir precīzi ieprogrammētas un lietotāja pielāgotas komandas. Mākslīgais intelekts šajās sistēmās varētu tikt izmantots jau nākamajā fāzē, tas ir, brīdī, kad notiek telpas atdzesēšana vai uzsildīšana. Piemēram, ja ir pieejams gaisa kondicionieris, tad var ievākt datus, cik ilgā laikā un ar kādu jaudas režīmu darbojoties, tas spēj atdzesēt telpu, ņemot vērā, piemēram, temperatūru starpību telpā un ārpusē.

Turpmākie darbi termoregulācijas sistēmas izstrādē ir papildus funkcionalitātes izstrāde, kas ļauj lietotājam norādīt vēlamo temperatūru katrā istabā dažādos laika posmos un automātiski mēģināt to pielāgot, kā arī turpināt reģistrēt lietotāja veiktās darbības, lai atkārtoti mēģinātu izveidot veiksmīgu mašīnmācīšanās modeli. Ir plānots uzstādīt papildu sensorus – gaisa kvalitātes sensoru, kas automātiski kontrolētu to, kurā brīdī vajag atvērt logu, kā arī mitruma sensoru mājas ārpusē, lai pārliecinātos par to, vai logu atvērt ir droši. Nepieciešams arī izveidot dizainiski neitrālus un funkcionālus ietvarus visām ierīcēm, lai tās varētu vieglāk uzturēt un labāk iederētos kopējā mājas iekārtojumā.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. internet of things (IoT) [tiešsaiste] [skatīts 20.04.2019.]
<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
2. The History of Smart Homes [tiešsaiste] [skatīts 20.04.2019.]
<https://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm>
3. Tet apkures kontrole [tiešsaiste] [skatīts 20.04.2019.]
<https://gudraenergija.tet.lv/lv/tet-apkures-vadiba>
4. Centrālapkure (vienota siltuma regulēšana). [tiešsaiste] [skatīts 22.04.2019.]
<https://pakalpojumi.elektrum.lv/lv/majai/vieda-maja/risinajumi/risinajums-dzivoklim-ar-individualo-siltuma-uzskaiti-siltuma-regulesana-visa-dzivokli-vienoti>
5. Best Smart Home Thermostats For 2019. [tiešsaiste] [skatīts 25.04.2019.]
<https://www.cnet.com/topics/smart-thermostats/best-smart-thermostats/>
6. Ecobee Room Sensors [tiešsaiste] [skatīts 28.04.2019.]
<https://www.ecobee.com/room-sensors/>
7. Ecobee 4. [tiešsaiste] [skatīts 28.04.2019.]
<https://www.ecobee.com/ecobee4/>
8. About us. [tiesšaiste] [skatīts 28.04.2019.]
<https://www.raspberrypi.org/about/>
9. Featured products. [tiešsaiste] [skatīts 28.04.2019.]
<https://www.raspberrypi.org/products/>
10. Raspberry Pi Zero W [tiešsaiste] [skatīts 28.04.2019.]
<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
11. Raspbian. [tiešsaiste] [skatīts 28.04.2019.]
<https://www.raspbian.org/>
12. The 5 questions data science answers [tiešsaiste] [skatīts 12.05.2019.]
<https://docs.microsoft.com/lv-lv/azure/machine-learning/studio/data-science-for-beginners-the-5-questions-data-science-answers>
13. Smart Radiator Valve With Home Assistant [tiešsaiste] [skatīts 19.05.2019.]
<https://www.instructables.com/id/Smart-Radiator-Valve-With-Home-Assistant/>
14. Arduino Controlled Window for Cat [tiešsaiste] [skatīts 19.05.2019.]
<https://www.instructables.com/id/Arduino-controlled-window-for-cat/>
15. Linear actuator guide. [tiešsaiste] [skatīts 21.05.2019.]
<https://www.anaheimautomation.com/manuals/forms/linear-actuator-guide.php>

16. TB6600 Stepper Motor Driver. [tiešsaiste] [skatīts 23.05.2019.]
<https://thepihut.com/products/tb6600-stepper-motor-driver>
17. Frequently Asked Questions [tiešsaiste] [skatīts 23.05.2019.]
<https://js.devexpress.com/Licensing/#NonCommercial>
18. LM75A Digital Temperature Sensor and Thermal Watchdog With Two-Wire Interface.
[tiešsaiste] [skatīts 23.05.2019.]
<http://www.ti.com/lit/ds/symlink/lm75a.pdf>
19. 11 Most Popular Machine Learning Software Tools in 2019. [tiešsaiste] [skatīts 24.05.2019.]
<https://www.softwaretestinghelp.com/machine-learning-tools/>
20. Weka 3: Machine Learning Software in Java. [tiešsaiste] [skatīts 24.05.2019.]
<https://www.cs.waikato.ac.nz/ml/weka/>
21. How to prepare dataset in arff and csv format. [tiešsaiste] [skatīts 24.05.2019.]
<http://www.e2matrix.com/blog/2017/10/28/how-to-prepare-dataset-in-arff-and-csv-format/>
22. Overview. [tiešsaiste] [skatīts 24.05.2019.]
https://waikato.github.io/weka-wiki/arff_stabl

Bakalaura darbs „Termoregulācijas sistēmas izstrāde viedajai mājai” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Rihards Mozaļevskis

Rekomendēju/nerekomendēju darbu aizstāvēšanai (*nederīgo svītro vadītājs*)

Vadītājs: Dr. sc. comp. Matīss Rikters _____2019.

Recenzents: Dr. sc. comp. Artis Mednis

Darbs iesniegts Datorikas fakultātē 27.05.2019.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

____.06.2019. prot. Nr. _____

Komisijas sekretārs(-e): _____