



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Išmanaus pastato jutiklių stebėjimo ir anomalinio vandens naudojimo aptikimo sistemos kūrimas ir tyrimas**

Baigiamasis magistro studijų projektas

---

**Lukas Kulikovas**

Projekto autorius

**Doc. Šarūnas Packevičius**

Vadovas

---

**Kaunas, 2023**



**Kauno technologijos universitetas**

Informatikos fakultetas

# **Išmanaus pastato jutiklių stebėjimo ir anomalinio vandens naudojimo aptikimo sistemos kūrimas ir tyrimas**

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

---

**Lukas Kulikovas**

Projekto autorius

**Doc. Šarūnas Packevičius**

Vadovas

**Lekt. Dominykas Barisas**

Recenzentas

---

**Kaunas, 2023**

Kulikovas, Lukas. Išmanaus pastato jutiklių stebėjimo ir anomalinio vandens naudojimo aptikimo sistemos kūrimas ir tyrimas. Magistro studijų baigiamasis projektas / vadovas doc. Šarūnas Packevičius; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Programų sistemų inžinerija.

Reikšminiai žodžiai: anomalinis vandens naudojimas, neprižiūrimas mašininis mokymas, dalinai prižiūrimas mašininis mokymas.

Kaunas, 2023. 126 p.

### **Santrauka**

Šiame darbe yra pristatoma jutiklių stebėjimo ir anomalinio vandens naudojimo aptikimo sistema, skirta palengvinti jutiklių duomenų monitoringą bei galimo vandens nuotėkio aptikimą. Ši sistema yra sudaryta iš mobiliosios aplikacijos, kuri yra įdiegta naudotojo telefone, serverio bei mikrovaldiklio, kuris yra įrengtas naudotojo vandens tiekimo sistemoje. Serveris naudoja mašininio mokymo algoritmą, kad būtų galima efektyviai aptikti galimą vandens nuotėkį ir greičiau išvengti didelių nuostolių. Serveris taip pat siūnia aliarminius pranešimus naudotojui į mobilųjį įrenginį pranešti apie anomalinį vandens naudojimą su tikslu vartotojui leisti išvengti nepageidaujamų pasekmių efektyviai ir greitai.

Darbo tiriamojoje dalyje yra palyginami įvairūs mašininio mokymo algoritmai naudojant įvairius scenarijus bei metrikas, siekiant surasti efektyviausią modelį nustatant anomalinį vandens naudojimą. Tokio tipo sistema gali būti naudojama tiek privačiuose namuose, tiek ir viešosiose įstaigose. Projektas yra naudingas ne tik siekiant išvengti finansinių nuostolių, tačiau ir tausojant vieną iš svarbiausių aplinkos resursų – vandenį.

Kulikovas, Lukas. Research and development of a smart buildings sensors monitoring and anomalous water usage detection system. Master's Final Degree Project / supervisor abbreviation of the position, doc. Šarūnas Packedvičius; Informatics Faculty, Kaunas University of Technology.

Study field and area (study field group): Computer Sciences, Software Engineering.

Keywords: anomalous water usage, unsupervised machine learning, semi-supervised machine learning.

Kaunas, 2023. 126 p.

### **Summary**

In this paper we present a sensors monitoring and anomalous water use detection system to facilitate sensors data monitoring and detection of potential water leaks. The system consists of a mobile application that is installed on the user's phone, a server, and a microcontroller, that is installed in the user's water supply system. The server uses a machine learning algorithm to efficiently detect potential water leaks and prevent major losses. The server also sends alarm messages to the user's mobile device to report anomalous water use with the aim of allowing the user to avoid unwanted consequences quickly.

In the exploratory part of the work different machine learning algorithms are compared using different scenarios and metrics to find the most efficient model for detecting anomalous water usage. This type of system can be used in private homes as well as in public institutions. The project is not only beneficial for avoiding financial losses, but also for conserving one of the most important environmental resource – water.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>6</b>
<b>Paveikslų sąrašas .....</b>	<b>8</b>
<b>Santrumpų ir terminų sąrašas .....</b>	<b>10</b>
<b>Įvadas.....</b>	<b>12</b>
<b>1. Analitinė dalis .....</b>	<b>13</b>
1.1. Poreikis .....	13
1.2. Rinkos tyrimas.....	13
1.3. Vandens nuotėkio aptikimo algoritmai .....	14
1.4. Naudojamos architektūros, jų veikimo principai ir apribojimai.....	23
1.5. Egzistuojančių algoritmų palyginimas .....	27
1.6. Analizės išvados .....	28
<b>2. Projektinė dalis .....</b>	<b>29</b>
2.1. Reikalavimų specifikacija .....	29
2.2. Architektūros specifikacija .....	63
2.3. Testavimo medžiaga .....	81
<b>3. Tiriamoji dalis.....</b>	<b>94</b>
3.1. Kiekybinis tyrimas.....	94
3.2. Kokybinis tyrimas .....	99
<b>4. Eksperimentinė dalis .....</b>	<b>102</b>
4.1. Duomenys.....	102
4.2. Duomenų ir scenarijų paruošimas .....	102
4.3. Įverčiai.....	103
4.4. Modelių palyginimas skirtingų slenkančių langų režiuose .....	104
4.5. Modelių palyginimas naudojant skirtingus kontekstus .....	106
4.6. Modelių palyginimas naudojant skirtingas duomenų dimensijas.....	107
<b>Išvados .....</b>	<b>109</b>
<b>Literatūros sąrašas .....</b>	<b>110</b>
<b>Priedai.....</b>	<b>112</b>

## Lentelių sąrašas

<b>1 lentelė.</b> Panašių produktų palyginimas.....	13
<b>2 lentelė.</b> Tradicinių algoritmų palyginimas .....	27
<b>3 lentelė</b> Neprižiūrėjo mašininio mokymo algoritmų palyginimas .....	27
<b>4 lentelė.</b> Prižiūrėjo mašininio mokymo algoritmų palyginimas.....	28
<b>5 lentelė</b> Esamos situacijos veiklos .....	36
<b>6 lentelė</b> PA 1 Prisijungti .....	40
<b>7 lentelė</b> PA 2 Peržiūrėti jutiklių sąrašą .....	41
<b>8 lentelė</b> PA 3 Filtruoti jutiklius.....	41
<b>9 lentelė</b> PA 4 Peržiūrėti jutiklių.....	42
<b>10 lentelė</b> PA 5 Keisti jutiklio nustatymus.....	42
<b>11 lentelė</b> PA 6 Peržiūrėti nustatymus .....	43
<b>12 lentelė</b> PA 7 Keisti nustatymus .....	43
<b>13 lentelė</b> PA 8 Peržiūrėti įspėjimų sąrašą.....	44
<b>14 lentelė</b> PA 9 Trinti įspėjimus .....	45
<b>15 lentelė</b> PA 13 Gauti aliarminius pranešimus.....	45
<b>16 lentelė</b> PA 14 Peržiūrėti naudotojus .....	46
<b>17 lentelė</b> PA 15 Sukurti naudotoją .....	46
<b>18 lentelė</b> PA 16 Ištrinti naudotoją.....	47
<b>19 lentelė</b> PA 26 Siųsti duomenis .....	47
<b>20 lentelė</b> PA 27 Priimti duomenis .....	48
<b>21 lentelė</b> PA 28 Generuoti ataskaitą.....	48
<b>22 lentelė</b> PA 29 Įvertinti vandens naudojimo anomaliją .....	49
<b>23 lentelė</b> PA 30 Siųsti aliarminį pranešimą.....	49
<b>24 lentelė</b> Duomenų žodynas .....	51
<b>25 lentelė</b> Išvaizdos reikalavimas 1.....	53
<b>26 lentelė</b> Stiliaus reikalavimas 1.....	54
<b>27 lentelė</b> Naudojimosi paprastumo reikalavimas 1 .....	54
<b>28 lentelė</b> Naudojimosi paprastumo reikalavimas 2 .....	54
<b>29 lentelė</b> Naudojimosi paprastumo reikalavimas 3 .....	54
<b>30 lentelė</b> Personalizavimo reikalavimas 1 .....	55
<b>31 lentelė</b> Personalizavimo reikalavimas 2.....	55
<b>32 lentelė</b> Prieinamumo reikalavimas 1 .....	55
<b>33 lentelė</b> Užduočių vykdymo reikalavimas 1 .....	56
<b>34 lentelė</b> Tikslumo reikalavimas 1 .....	56
<b>35 lentelė</b> Veikimo sąlygos reikalavimas 1 .....	56
<b>36 lentelė</b> Prieigos reikalavimas 1.....	57
<b>37 lentelė</b> Prieigos reikalavimas 2.....	57
<b>38 lentelė</b> Vientisumo reikalavimas 1 .....	57
<b>39 lentelė</b> Privatumo reikalavimas 2 .....	57
<b>40 lentelė</b> Savisaugos reikalavimas 1.....	58
<b>41 lentelė</b> Kultūrinis rinkos reikalavimas 1 .....	58
<b>42 lentelė.</b> Projekto biudžetas .....	62
<b>43 lentelė</b> Perspektyvinis reikalavimas 1 .....	63
<b>44 lentelė</b> Perspektyvinis reikalavimas 2 .....	63

<b>45 lentelė</b>	Aplinkų reikalavimai bei konfigūracijos nustatymai .....	80
<b>46 lentelė</b>	Prisijungimo testavimo scenarijus .....	86
<b>47 lentelė</b>	Profilio informacijos keitimo testavimo scenarijus .....	86
<b>48 lentelė</b>	Jutiklių peržiūros testavimo scenarijus .....	86
<b>49 lentelė</b>	Jutiklių filtravimo testavimo scenarijus .....	86
<b>50 lentelė</b>	Jutiklio peržiūrėjimo testavimo scenarijus .....	87
<b>51 lentelė</b>	Jutiklio nustatymų keitimo testavimo scenarijus .....	87
<b>52 lentelė</b>	Įspėjimų peržiūros testavimo scenarijus .....	88
<b>53 lentelė</b>	Nustatymų peržiūros testavimo scenarijus .....	88
<b>54 lentelė</b>	Matavimo sistemos keitimo testavimo scenarijus .....	88
<b>55 lentelė</b>	Duomenų siuntimo testavimo scenarijus .....	89
<b>56 lentelė</b>	Duomenų priėmimo testavimo scenarijus .....	89
<b>57 lentelė</b>	Galimo vandens nuotėkio įvertinimo testavimo scenarijus .....	89
<b>58 lentelė</b>	Įspėjamojo signalo siuntimo testavimo scenarijus .....	89
<b>59 lentelė</b>	Testavimo įrankiai ir aplinka .....	90
<b>60 lentelė</b>	Testavimo aplinkos .....	90
<b>61 lentelė</b>	Elemento patenkinimo kriterijai .....	91
<b>62 lentelė</b>	Testavimo rezultatai .....	92
<b>63 lentelė.</b>	Vandens nuotėkio scenarijai .....	102

## Paveikslų sąrašas

<b>1 pav.</b>	Boudhaouia ir Wira autorių pasiūlytas vandens nuotėkio aptikimo algoritmas [9] .....	15
<b>3 pav.</b>	Keturių scenarijų vandens nuotėkio aptikimo algoritmas [6] .....	17
<b>4 pav.</b>	Fuentes ir Mauricio pasiūlyto algoritmo testavimo rezultatai .....	18
<b>6 pav.</b>	João Alves Coelho mašininio mokymo modelių analizės palyginimo rezultatai [5] .....	19
<b>7 pav.</b>	RNN-LSTM modelio struktūra [14] .....	20
<b>8 pav.</b>	Dalinai prižiūrimo mašininio mokymo metodas [15] .....	20
<b>9 pav.</b>	Active-MTSAD algoritmo procesas [16] .....	21
<b>2 pav.</b>	Dvikrypčio ilgalaikės trumpalaikės atminties modelio architektūra [10] .....	22
<b>5 pav.</b>	Ismail et al. mašininio mokymo modelių analizės palyginimo rezultatai [13] .....	23
<b>10 pav.</b>	Aukšto lygio vandens sistemos suvartojimo architektūros diagrama [6] .....	24
<b>11 pav.</b>	Fuentes ir Mauricio pasiūlyta fizinė vandens suvartojimo sistemos architektūra [6] .....	24
<b>12 pav.</b>	Fuentes ir Mauricio pasiūlyta technologinė vandens suvartojimo sistemos architektūra [6] .....	25
<b>13 pav.</b>	Daiktų interneto architektūros diagrama [22] .....	26
<b>14 pav.</b>	„SmartWater“ sprendimo dizainas [23] .....	27
<b>15 pav.</b>	„SmartWater“ procesas [23] .....	27
<b>16 pav.</b>	Sistemos išdėstymo diagrama .....	32
<b>17 pav.</b>	Sistemos konteksto diagrama .....	36
<b>18 pav.</b>	Naudotojo panaudojimo atvejų modelis .....	37
<b>19 pav.</b>	Administratoriaus panaudojimo atvejų modelis .....	38
<b>20 pav.</b>	Serverio panaudojimo atvejų modelis .....	39
<b>21 pav.</b>	Sistemos esybių ryšių diagrama .....	51
<b>22 pav.</b>	Sistemos kūrimo procesas .....	61
<b>23 pav.</b>	Sistemos paketų diagrama .....	64
<b>24 pav.</b>	Mobiliosios aplikacijos vartotojo sąsajos paketų diagrama .....	65
<b>25 pav.</b>	Mobiliosios aplikacijos duomenų paketų diagrama .....	66
<b>26 pav.</b>	Mobiliosios aplikacijos pagalbinių klasių paketų diagrama .....	67
<b>27 pav.</b>	Serverio autorizacijos bei duomenų gavimo paketų diagrama .....	68
<b>28 pav.</b>	Serverio servisų bei pagalbinių klasių paketų diagrama .....	69
<b>29 pav.</b>	Serverio modelių paketų diagrama .....	70
<b>30 pav.</b>	Jutiklio paketų diagrama .....	71
<b>31 pav.</b>	Detalios jutiklio informacijos peržiūros veiklos diagrama .....	71
<b>32 pav.</b>	Mašininio modelio duomenų analizės veiklos diagrama .....	72
<b>33 pav.</b>	Įspėjamojo pranešimo siuntimo veiklos diagrama .....	73
<b>34 pav.</b>	Prisijungimo sekų diagrama .....	74
<b>35 pav.</b>	Profilio informacijos koregavimo sekų diagrama .....	75
<b>36 pav.</b>	Jutiklių peržiūros sekų diagrama .....	75
<b>37 pav.</b>	Detalios jutiklio informacijos peržiūros sekų diagrama .....	76
<b>38 pav.</b>	Įspėjamųjų pranešimų nustatymų keitimo sekų diagrama .....	76
<b>39 pav.</b>	Jutiklio duomenų siuntimo sekų diagrama .....	77
<b>40 pav.</b>	Įspėjamojo pranešimo siuntimo sekų diagrama .....	78
<b>41 pav.</b>	Mašininio mokymo duomenų analizės sekų diagrama .....	79
<b>42 pav.</b>	Įspėjamojo pranešimo siuntimo būsenų diagrama .....	80
<b>43 pav.</b>	Mobiliosios aplikacijos vienetų testų pavyzdys .....	84



<b>44 pav.</b> Mobiliosios aplikacijos vienetų testų rezultatų pavyzdys .....	84
<b>45 pav.</b> Mobiliosios aplikacijos integracinių testų pavyzdys.....	85
<b>46 pav.</b> Sistemos lengvumo naudojimo vertinimas .....	95
<b>47 pav.</b> Sistemos padėjimo įvertinti anomalijos priežastis vertinimas .....	95
<b>48 pav.</b> Sistemos padėjimo sumažinti vandens suvartojimą pastate vertinimas .....	96
<b>49 pav.</b> Sistemos palengvinimo jutiklių valdymo pastate vertinimas .....	97
<b>50 pav.</b> Sistemos navigavimo lengvumo vertinimas.....	97
<b>51 pav.</b> Sistemos išbaigtumo vertinimas.....	98
<b>52 pav.</b> Sistemos veikimo vertinimas .....	98
<b>53 pav.</b> Sistemos kainos vertinimas .....	99
<b>54 pav.</b> Modelių palyginimas skirtingų valandų režiuose.....	105
<b>55 pav.</b> Modelių palyginimas skirtingų minučių režiuose .....	106
<b>56 pav.</b> Modelių palyginimas skirtinguose kontekstuose .....	107
<b>57 pav.</b> Modelių palyginimas skirtingose dimensijose .....	108

## Santrumpų ir terminų sąrašas

### Santrumpos:

**ANN** – dirbtinis neuroninis tinklas (angl. *Artificial Neural Network*).

**API** – taikomosios programos sąsaja (angl. *Application Programming Interface*).

**AVG** – vidutinis tĕkmĕs srautas (angl. *Average Per Hour*).

**CBLOF** – klasterizavimu pagrĭstas vietinių nuokrypių faktorius (angl. *Clustering Based Local Outlier Factor*).

**CF** – nuolatinis srautas (angl. *Continuous Flow*).

**CNN** – konvoliuciniai neuroniniai tinklai (angl. *Convolutional Neural Networks*).

**CNZ** – nepertraukiamas nenulinis vandens sunaudojimas (angl. *Continuous Non-Zero Water Consumption*).

**COPOD** – kopula pagrĭstas nukrypimų detektorius (angl. *Copula Based Outlier Detector*).

**C3S** – trys panašūs suvartojimai iš eilĕs (angl. *three similar consumptions in a row*).

**DT** – sprendimų medžiai (angl. *Decision Trees*).

**ECOD** – empiriniu kaupiamuoju pasiskirstymu pagrĭstas nuokrypių detektorius (angl. *Empirical Cumulative Distribution Based Outlier Detector*).

**GPS** – globali padĕties sistema (angl. *Global Position System*).

**HBOS** – histogramomis pagrĭstas nuokrypių detektorius (angl. *Histogram Based Outlier Detector*).

**IF** – izoliacijos medžiais paremtas nuokrypių detektorius (angl. *Isolation Forest Outlier Detector*).

**KNN** – k-artimiausių kaimynų nuokrypių detektorius (angl. *K-Nearest Neighbours Outlier Detector*).

**LOF** – nuokrypių detektorius paremtas vietiniu nuokrypių koeficientu (angl. *Local Outlier Factor*).

**MAE** – vidutinĕ absoliuti paklaida (angl. *Mean Absolute Error*).

**ML** – mašininis mokymas (angl. *Machine Learning*).

**MNF** – minimalus naktinis srautas (angl. *Minimum Night Flow*).

**NN** – neuroniniai tinklai (angl. *Neural Networks*).

**OCSVM** – vienos klasĕs atraminių vektorių mašina (angl. *One-Class Support Vector Machine*).

**PA** – aprašas, nurodantis kaip tam tikras naudotojas atliks tam tikrą funkciją sistemoje.

**PCA** – pagrindinių komponentų analizĕs nuokrypių detektorius (angl. *Principal Component Analysis Outlier Detector*).

**PWNC** – laikotarpis be nulinio suvartojimo (angl. *Period Without Null Consumption*).

**RF** – atsitiktinis miškas (angl. *Random Forest*).

**RMS** – vidutinė kvadratinė šaknis (angl. *Root Mean Squared*).

**RNN-LSTM** – pasikartojančių neuroninių tinklų ilgalaikė trumpalaikė atmintis (angl. *Recurrent Neural Networks-Long Short-Term Memory*).

**SSDO** – dalinai prižiūrimo mašininio mokymo nuokrypių aptikimo detektorius (angl. *Semi-Supervised Detection of Outliers*).

**SSLD** – savarankiškai prižiūrimas mokymo detektorius (angl. *Self-Supervised Learning Detector*).

**SVM** – atraminių vektorių klasifikatoriaus (angl. *Support Vector Machine*).

**UML** – unifikuota modeliavimo kalba (angl. *Unified Modelling Language*).

**VT** – veiklos taisyklės.

**XGBoost** – ekstremalaus gradientų padidinimas (angl. *Extreme Gradient Boost*).

#### **Terminai:**

**Aukšto sunaudojimo anomalijų scenarijus** – aukšto vandens sunaudojimo, lyginant su istoriniais, scenarijus.

**Firestore** – realaus laiko debesų kompiuterijos paslaugų sistema duomenims talpinti ir sinchronizuoti.

**LiveData** – duomenų klasė, kuri yra stebima kito gyvavimo ciklą turinčio klasės tipo objekto.

**Neigiamos tendencijos scenarijus** – duomenų per paskutines 24 valandas sumažėjimo scenarijus.

**Panašių sunaudojimų scenarijus** – duomenų, lyginant su prieš tai neseniai gautais, scenarijus.

**PT** – viena septyniasdešimt antroji ( $1/72$ ) colio dydžio matavimo vienetas, turint omeny fizinio įrenginio (telefono) ekrano dydžio aspektą.

**ViewModel** – klasės tipas, atsakingas už duomenų apdorojimą bei paruošimą atvaizdavimui.

**24 valandų sunaudojimo scenarijus** – per paskutines 24 valandas išgautų duomenų scenarijus.

## Ivadas

Vandens nuotėkis yra viena iš didžiausių namų ūkio nuostolių pasekmių, kuris, JAV statistikos duomenimis, sudaro daugiau nei 50 % nekilnojamojo turto žalos [1]. Taip pat JAV statistikos duomenimis per metus vandens nuotėkis gali siekti iki 3,5 trilijonų litrų nutekinto vandens per metus [2]. Visa tai gali sukelti nepakeičiamus turto nuostolius, pelėsių atsiradimą ir ilgalaikes išlaidas namo gyventojams [3]. Reikia nepamiršti, kad vandens kampanijos naudotojams suteiktas vanduo turi būti surinktas, padarytas geriamu bei būti tiekiamas per tūkstančius metrų besitęsiančius vamzdynus [3]. Atsižvelgiant į visą tai bei remiantis padidėjusiu susirūpinimu dėl vandens trūkumo ir netinkamo jo naudojimo, atsakingas vandens valdymas tapo svarbiausiu pasauliniu iššūkiu, o vienas iš būdų tai išspręsti – sukurti anomalinio vandens naudojimo aptikimo sistemą, siekiant aptikti ir sustabdyti vandens nuotėkį iš anksto.

### Darbo naujumas ir aktualumas

Anomalinio vandens nuotėkio aptikimo sistemos šiuo metu nėra plačiai paplitusios ar aktyviai naudojamos. Nors ir šiuo metu egzistuoja tokio tipo sistemos, dažnai jos yra lėtai besimokančios ir be visa to pateikiančios ne tik klaidingus aliarminius pranešimus, bet ir nesugebančios teisingai nustatyti vandens nuotėkį, dėl to nukenčia jas naudojantys vartotojai. Dėl to yra svarbu sukurti sistemą, kuri sugebėtų ne tik pranešti apie galimą vandens nuotėkį, bet ir kuri sugebėtų atpažinti kuo daugiau tokių scenarijų. Tokio tipo sistema yra svarbi tiek privačių pastatų savininkams, tiek ir viešo sektoriaus pastatams, siekiant apsaugoti savo infrastruktūrą bei tausoti gyvybiškai svarbų pasaulio resursą – vandenį.

**Darbo tikslas** – naudojant mašininio mokymo modelį nustatyti anomalinį vandens naudojimą naudotojo pastate. Šiam tikslui pasiekti buvo išskelti tokie uždaviniai:

1. išanalizuoti šiuo metu egzistuojančius mašininio mokymo algoritmus anomaliam vandens naudojimui nustatyti;
2. atlikti įvairių mašininio mokymo algoritmų analizę naudojant įvairius scenarijus bei metrikas, siekiant surasti efektyviausią vandens nuotėkiui nustatyti;
3. sukurti sistemą, kurioje naudotojas galėtų stebėti jutiklių informaciją bei gauti aliarminius pranešimus apie galimą vandens nuotėkį.

### Dokumento struktūra

Šiame dokumente yra pateikiama srities analizė bei rinkoje egzistuojantys sprendimai, kurie leidžia stebėti jutiklių informaciją bei aptikti vandens nuotėkį, sukurtos išmaniosios sistemos reikalavimai ir architektūra, tyrimas, kuriame yra atliekamas esamos sistemos kokybinis ir kiekybinis įvertinimas, ir eksperimentinė dalis, kurioje yra palyginami įvairūs mašininio mokymo algoritmai bei pateikiami gauti rezultatai ir išvados.

## 1. Analitinė dalis

### 1.1. Poreikis

#### 1.1.1. Projekto naudotojai ir klientai

Pagrindiniai programėlės naudotojai yra pastatų savininkai arba kiti asmenys, prižiūrintys tam tikrą vandens nuotėkio sistemą, kurią vienija vienas ar keli namai, bei norintys stebėti savo namo vandens suvartojimą ir gauti išpėjamuosius pranešimus apie galimą anomalinį vandens naudojimą.

#### 1.1.2. Naudotojų rolės ir tikslai

Atsižvelgiant į įvade pristatytas statistikas apie šiuo metu patiriamus vandens nuotėkio nuostolius akivaizdu teigti, kad vandens nuotėkis padaro labai didelę žalą. Netgi 14 tūkst. JAV gyventojų kiekvieną dieną patiria vandens nuotėkio nelaimės [4]. Net mažiausi vandens nuotėkiai prisideda prie 1 trilijono vandens išnaudojimo kiekvienais metais, o tai yra lygu 11 milijonų žmonių išnaudoto vandens savo reikmėms [2].

Nors ir šiuo metu egzistuoja įvairių visiems naudotojams prieinamų vandens nuotėkį aptinkančių produktų, tačiau naudotojo sąsaja yra ribota, dauguma jutiklių reikalauja būti montuojami kiekviename kambaryje arba prie buitinių įrenginių, o išmanesni produktai net ir praėjus ilgam laikui nesugeba prisitaikyti prie naudotojo sunaudoto vandens kiekio vandens nuotėkiui aptikti.

Yra atlikta įvairių tyrimų bei eksperimentų siekiant surasti efektyvų algoritmą nustatyti vandens nuotėkiui. Vienuose tyrimuose buvo remtasi tradiciniais metodais<sup>1</sup>, kituose – mašininio mokymusi, įtraukiant neprižiūrimo, dalinai prižiūrimo ir pilnai prižiūrimo mašininio mokymo metodus. Vieni mašininio mokymo paremti modeliai sugebėjo aptikti iki 75 % vandens nuotėkių [5], kiti, įtraukiant naudotojo būvimo koordinatas, sugebėjo nustatyti 100 % vandens nuotėkių pagal pateiktus naudotojų duomenis [6].

Būtent naudojant vandens anomalinio naudojimo aptikimo sistemą yra vienas iš efektyviausių būdų šiuo metu nustatyti vandens nuotėkį bei išvengti didelių nuostolių.

### 1.2. Rinkos tyrimas

Rinkoje galima rasti įvairių panašios paskirties produktų, tačiau vienos sistemos yra orientuotos į jutiklių išdėstymą į vietas, kuriose labiausiai tikėtina, kad bus nuotėkis (po kriaukle ar kranais), kitos – jutiklio įrengimą į vieną vietą po pagrindiniu kranu, kuris naudos tam tikrą algoritmą naudojant vartotojo sunaudojamą vandens kiekį vandens nuotėkiui aptikti.

Panašių produktų palyginimas su būsimąja „Sensors Tracker“ yra atvaizduotas **1 lentelėje**.

**1 lentelė.** Panašių produktų palyginimas

Lyginimo kriterijai	„Phyn Plus“ Smart Water Assistant	„Moen 900-001“ Smart Water Shutoff	„Govee“ Water Sensors	„Sensors Tracker“
Yra galimybė gauti perspėjančius pranešimus.	Taip	Taip	Taip	Taip

<sup>1</sup> Metodai, kurie nenaudoja mašininio mokymo ir yra paremti ekspertų taisyklėmis.

Palaikomos operacinės sistemos stebėjimui.	„Android“ ir „iOS“	„Android“	„Android“	„Android“
Automatiškai užsidarys pagrindinis vandens čiaupas įvykus avarijai.	Taip	Taip	Ne	Ne
Dedasi ant pagrindinio skaitliuko ar ant vamzdžio, o ne po kiekvienu įtaisu.	Taip	Taip	Ne	Taip
Galima išjungti vandens tekėjimą nuotoliniu būdu.	Taip	Taip	Ne	Ne
Programėlė turi papildomų apmokestintų funkcijų.	Ne	Taip	Ne	Ne
Įvertinimų kiekis (potencialų naudotojų)	110	1061	7434	5
Įvertinimas	3.3/5	4/5	4.3/5	4.9/5
Naudoja mašininį mokymą	Ne	Dalinis	Ne	Taip

Atlikus konkurentų analizę galima pastebėti, kad panašiausios vandens stebėjimo sistemos į kurią yra „Phyn Plus“ ir „Moen 900-001“ – jos tiek praneša įspėjimus, tiek montuojasi prie pagrindinio vandens skaitliuko, o ne po kiekvienu įrenginiu. Nors ir esama sistema nesuteikia galimybės automatiškai uždaryti vandens čiaupo ar išjungti vandens tekėjimą nuotoliniu būdu, tačiau pagrindinis privalumas – greitas vandens nuotėkio aptikimas.

### 1.3. Vandens nuotėkio aptikimo algoritmai

Šiuo metu egzistuoja įvairių vandens nuotėkio aptikimo algoritmų, kurie bus aptariami tolimesniuose skyriuose. Kiekvienas iš algoritmų turi savų teigiamų ir neigiamų savybių, kur vieni iš jų turi didesnę tikslumo rodiklį, kiti – yra sunkiau įgyvendinami bei reikalauja daugiau techninių bei programinių resursų sistemos palaikymui.

#### 1.3.1. Tradiciniai metodai

Vandens anomalinio aptikimo srityje tradicinių metodų nėra daug. Paprastai šie metodai remiasi ekspertų nustatytomis taisyklėmis. Tolimesniuose poskyriuose bus apžvelgiami vandens anomalijos algoritmais, kurie remiasi tradiciniais metodais.

##### 1.3.1.1. Masės tvermės dėsnio paremtas metodas

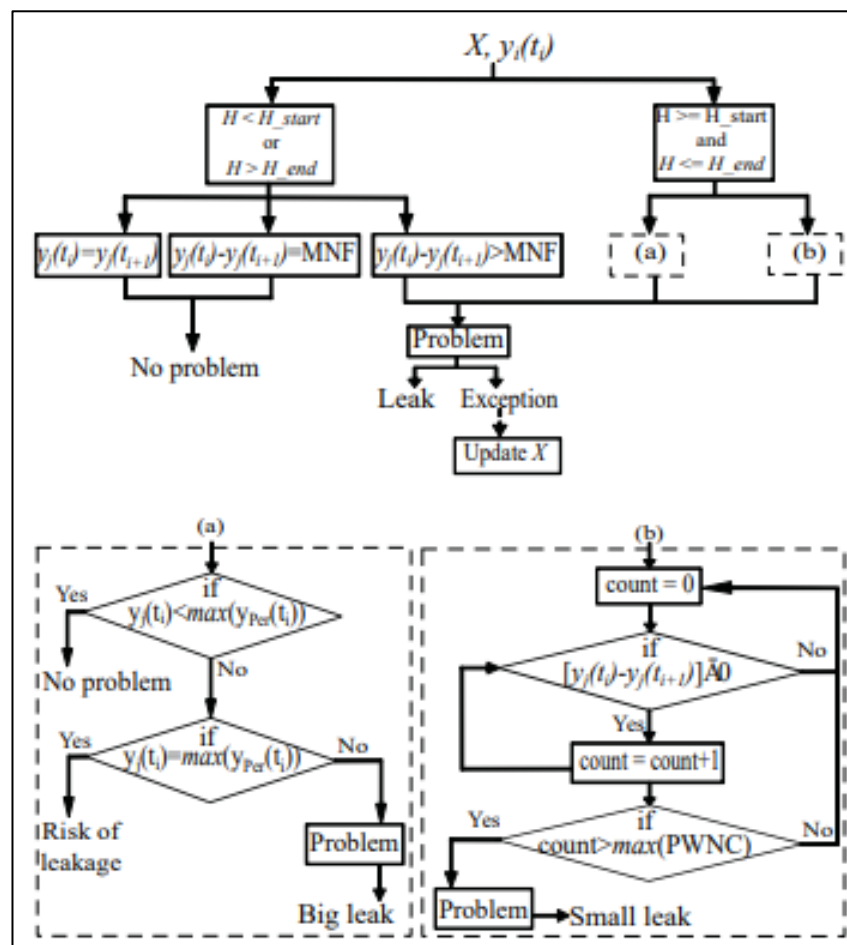
Sarangi [7] sukūrė vandens nuotėkio ir vagystės aptikimo sistemą. Pagrindinė pasiūlyto metodo idėja yra pagrįsta masės tvermės dėsniu, kuris teigia, kad materija negali būti sukurta ar sunaikinta. Taigi, autoriui tuo remiantis yra siūloma įrengti du jutiklius, kur vienas iš jų stebėtų vandenį, patenkantį į vamzdyną, o kitas – išeinantį iš jo. Šiuo atveju jeigu jutiklių matavimo skirtumas viršija tam tikrą nurodytą ribą, jutikliai išsiunčia įspėjimą pranešimą tolimesniam tos vietovės tyrimui dėl galimo nuotėkio. Atitinkamai, jeigu skirtumas tarp jutiklių yra labai didelis, tai gali reikšti neigiamai paveiktą vamzdžio infrastruktūrą arba vandens vagystę.

### 1.3.1.2. Akselerometru paremtas algoritmas

Moni et al. [8] parodė kaip jų būdas, naudojantis vandens vamzdynų vibracijas, gali padėti ūkininkams aptikti galimą vandens nuotėkį. Tam buvo panaudotas trigubos ašies akselerometras, kuris matuoja pagreitį trijose tiesinėse ašyse  $x$ ,  $y$  ir  $z$ . Šių trijų ašių duomenys buvo surenkami scenarijams kai buvo ir kai nebuvo vandens nuotėkis. Rezultatai parodė, kad RMS paklaida visada buvo mažesnė duomenyse, kuriuose buvo vandens nuotėkis. Galutiniai rezultatai taip pat parodė, kad modelio tikslumas siekė 87.9 %, kai buvo nuotėkis, ir 96.3 %, kai jo nebuvo.

### 1.3.1.3. Statistiniai parametrai remtas metodas

Boudhaouia ir Wira [9] pasiūlė sprendimą (žr. **1 pav.**), kuris yra paremtas trimis parametrais – maksimalios dienos apkrovos kreive, MNF ir PWNC. Esant nakties metui yra naudojamas minimalios nakties tėkmės parametras. Autorių teigimu, nakties metu neturėtų būti pastebimas joks vandens sunaudojimas – priešingu atveju tai turėtų būti užfiksuota kaip anomalinis vandens naudojimas. Dienos metu yra naudojami likę parametrai: maksimalios dienos apkrovos kreivė ir laikotarpis be nulio suvartojimo. Maksimalios dienos apkrovos kreivė žymi didžiausią vandens sunaudojimo limitą. Jeigu esamas vandens sunaudojimas viršija arba yra ant nurodytos maksimalaus vandens sunaudojimo ribos, atitinkamai yra sukuriamas vandens nuotėkio rizikos įspėjimas. Taip pat yra tikrinamas ir laikotarpis be nulinio suvartojimo – jeigu vandens suvartojimas viršija didesnę laikotarpį nei numatyta, algoritmas praneša apie nedidelį vandens nuotėkį. Autorių teigimu pasiūlytas algoritmas aptiko visus jų eksperimentų metu buvusius vandens nuotėkių scenarijus.



**1 pav.** Boudhaouia ir Wira autorių pasiūlytas vandens nuotėkio aptikimo algoritmas [9]

## 1.3.2. Neprižiūrimo mašininio mokymo algoritmai

Anomalijos dažnai atsitinka retai ir netikėtai, o žymių (angl. *labels*) duomenyse trūkumas autorius pastūmėjo neprižiūrimo mašininio mokymo link. Šiuo metu yra atlikta daug tyrimų, kuriuose autoriai demonstruoja savo pasiūlytus neprižiūrimo mašininio mokymo algoritmus vandens nuotėkiui nustatyti. Taip pat yra atlikta ir tyrimų, kuriuose yra palyginami įvairūs neprižiūrimo mašininio mokymo algoritmai. Tolimesniuose skyriuose bus aprašomi tiek sukurti įvairūs algoritmai, tiek jų palyginimai.

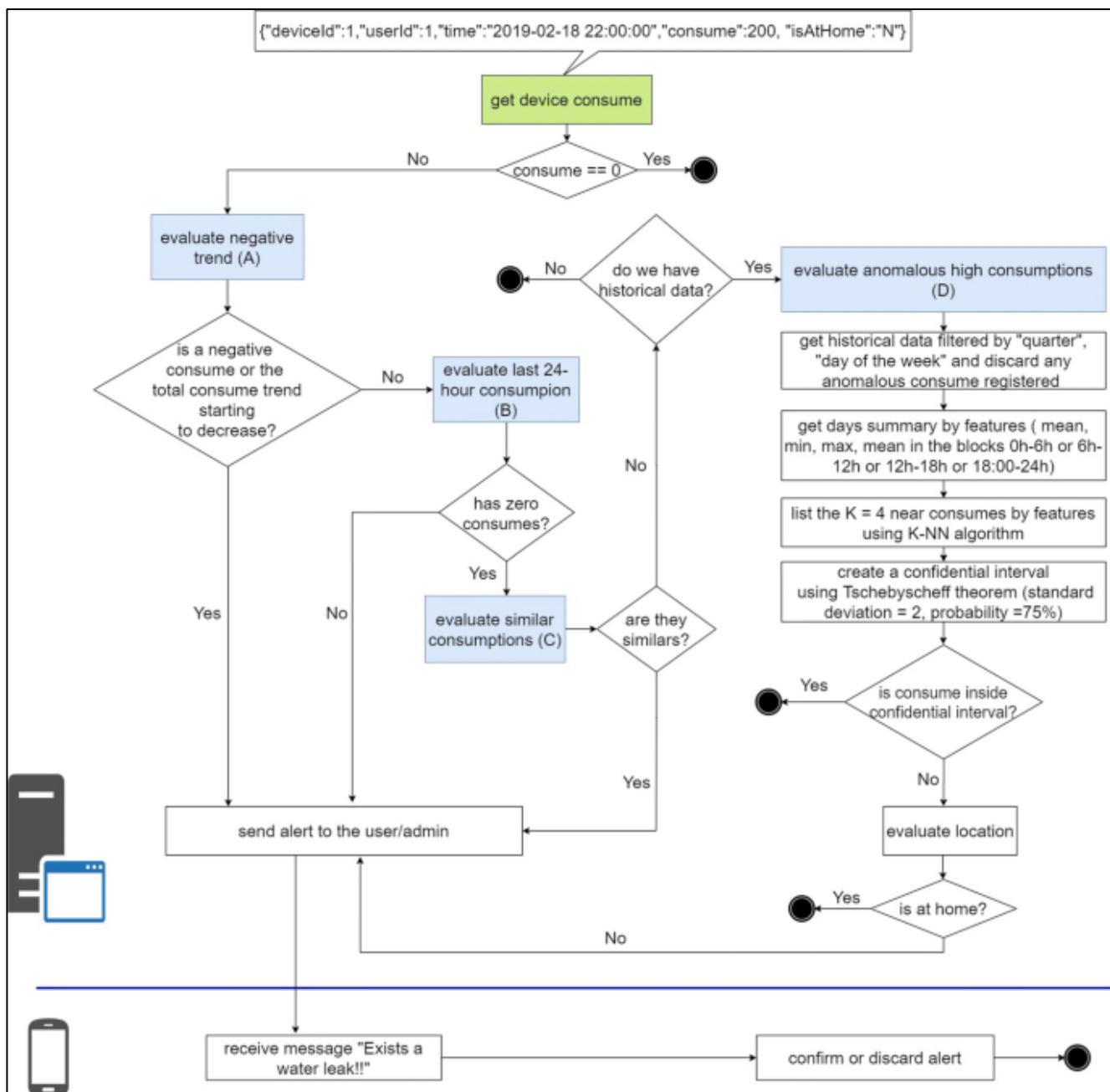
### 1.3.2.1. Keturių scenarijų algoritmas

Fuentes ir Mauricio pasiūlytas algoritmas (žr. **2 pav.**) aptinka galimus vandens nuotėkius naudojant keturis scenarijus. Šiam tikslui pasiekti į esamą algoritmą yra įrašomi keturi parametrai: įrenginio identifikacijos numeris „deviceId“, naudotojo identifikacijos numeris „userId“, laikas „time“, suvartotas vandens kiekis „consume“ (nurodytu laiko metu) bei vietovė (ar naudotojas yra namuose) „isAtHome“ .

Kiekvieno algoritmo iteracija yra įvykdoma nurodyto laiko pabaigoje ir algoritmas patikrina, ar vandens sunaudojimas tuo metu atitinka bent vieną iš keturių vandens nuotėkių scenarijų: neigiamos tendencijos (A), 24 valandų sunaudojimo (B), panašių sunaudojimų (C) bei aukšto sunaudojimo anomalijų (D). Šie tikrinimo žingsniai yra vykdomi iš eilės – pirmiausia A, paskui B, C ir galiausiai D. Jei bent vieno iš jų tikrinimo metu buvo nustatytas nuotėkis, įspėjamasis pranešimas yra išsiunčiamas naudotojui apie galimai aptiktą vandens nuotėkį [6]:

- 1. Scenarijus A:** patikrina, ar sunaudojamas vandens kiekis turi neigiamą reikšmę arba bendras sunaudojimas, sukauptas per paskutines 24 valandas, turi neigiamą tendenciją.
- 2. Scenarijus B:** patikrina, ar yra nepertraukiamas vandens tekėjimo sunaudojimas per paskutines 24 valandas nuo to laiko, kai nebuvo jokio vandens sunaudojimo (ši taisyklė dar yra žinoma kaip CNZ).
- 3. Scenarijus C:** patikrina, ar gautas vandens suvartotas kiekis atitinka paskutinius du sukurtus įrašus, kadangi mažai tikėtina, jog iš eilės būtų sunaudojama panašus vandens kiekis (ši taisyklė dar yra žinoma kaip C3S).
- 4. Scenarijus D:** patikrina, ar yra aukštas vandens sunaudojimas lyginant su istoriniais duomenimis.





2 pav. Keturių scenarijų vandens nuotėkio aptikimo algoritmas [6]

Fuentes ir Mauricio teigimu šis modelis turi 100 % tikslumą nustatant vandens nuotėkį [6]. Jie tai įrodė įvykdydami savo pasiūlytą bei kitus algoritmus tarp 69,194 tūkst. įrašų, kur rezultatuose atsispindėjo, kad autorių algoritmas nepadarė nė vienos klaidos, lyginant su kitais (žr. 3 pav.).

Scenario	UserId	Date	# Tests	Leakage/day	MNF	CF	AVG	Proposed algorithm
ACN	7	2019-02-28	22	1	0	0	5	1
ACN	8	2019-02-28	23	1	1	0	1	1
ACN	9	2019-02-28	23	2	1	0	2	2
ACNZ	3	2019-01-02	24	1	5	1	4	1
ACNZ	3	2019-01-18	24	1	5	1	5	1
ACNZ	7	2019-01-17	24	1	5	1	6	1
ACNZ	9	2019-01-15	24	1	5	1	6	1
ACS	1	2019-01-04	4	1	3	0	0	1
ACS	1	2019-02-02	20	1	0	0	5	1
ACS	8	2019-01-03	8	1	0	0	3	1
AHCW	1	2019-02-21	8	1	0	0	1	1
AHCW	5	2019-02-27	16	1	3	0	7	1
AHCW	5	2019-02-28	22	2	3	0	12	2
AHCW	6	2019-01-15	20	1	0	0	3	1
AHCW	6	2019-02-18	19	1	0	0	6	1
AHCWD	8	2019-02-23	24	1	0	0	1	1
AHCWD	8	2019-02-24	24	1	0	0	1	1
Total			329	19	31	4	68	19

**3 pav.** Fuentes ir Mauricio pasiūlyto algoritmo testavimo rezultatai

Rezultatuose akivaizdu matyti, kad pasiūlytas algoritmas sugebėjo aptikti kiekvieną vandens nuotėkio scenarijų lyginant su MNF, CF ir AVG modeliais.

### 1.3.2.2. ARIMA modelis

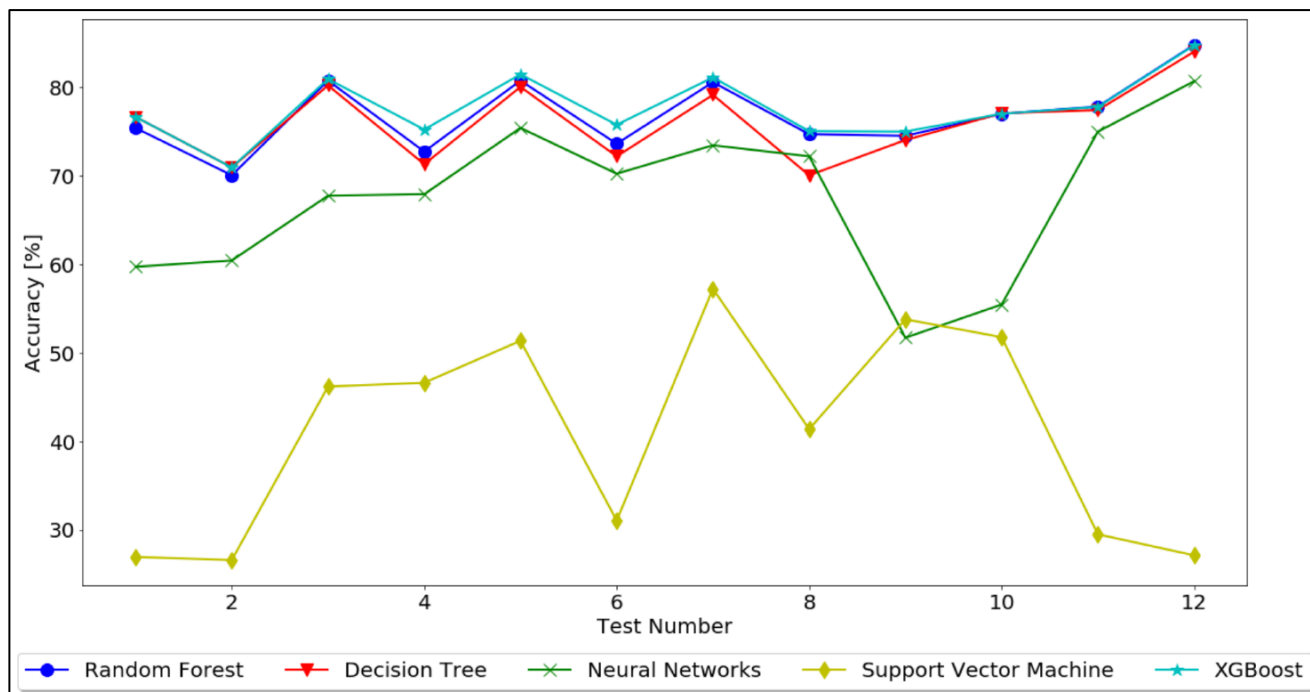
Ji et al. [11] palygino tiesinės regresijos, ARIMA bei adityvinės regresijos modelius. Algoritmams palyginti autoriai panaudojo Osan miesto naudotojų vandens suvartojimo duomenis. Apskaičiavus Durbin'o-Watson'o statistinę reikšmę paaikškėjo, kad tiesinės regresijos modelis negali būti pritaikytas duomenims, kadangi reikšmė buvo arti 0, todėl tolimesniems skaičiavimams buvo pasiūlytas ARIMA bei adityvinės regresijos Prophet modeliai. Galiausiai rezultatai parodė, kad ARIMA modelis turėjo 64 % tikslumą ir 39,230.88 MAE, Fbprophet modelis įskaitant metinę tendenciją grąžino šiek tiek mažesnę paklaidą – 17,635.15 ir mažesnę tikslumą – 46 %. Fbprophet modelio tikslumas neskaitant metinės tendencijos buvo 65 % su šiek tiek didesne paklaida – 23,566.54.

### 1.3.2.3. Artimųjų kaimynų metodu grįstas modelis

Patabendige et al. [12] sukūrė vandens anomalinio aptikimo algoritmą, kuris atsižvelgia į kontekstą, – jeigu lyginama diena yra darbo diena, tai tik dienos, kurios yra darbo dienos, bus lyginamos su esama, ir atvirkščiai – jeigu lyginama diena yra savaitgalio diena, tai bus lyginamos tik savaitgalio dienos. Pasiūlytas modelis įtraukia KNN algoritmą kartu su Gauso metodu paversti gautas KNN reikšmes į skaitinę tikimybę. Jų pasiūlyta sistema taip pat kas dieną sukuria ataskaitą, kurioje yra nurodoma anomalijos reikšmė tai dienai su papildomu tekstu, kuris nurodo, kas galėjo sukelti tokį neįprastą vandens naudojimą. Galiausiai ataskaita yra nusiunčiama naudotojui. Autorių teigimu siunčiant ataskaitas sistema įgauna vartotojų pasitikėjimą ir toliau naudotis jų produktu.

### 1.3.2.4. Modelių lyginimų tyrimai

Taip pat buvo atliktas tyrimas lyginant penkis mašininio mokymo modelius. João Alves Coelho pasirinko palyginti RF, DT, NN, SVM ir XGBoost modelius [5] (žr. **4 pav.**). 70 % duomenų buvo naudota mokymams, o likę 30 % - testavimui [5].



**4 pav.** João Alves Coelho mašininio mokymo modelių analizės palyginimo rezultatai [5]

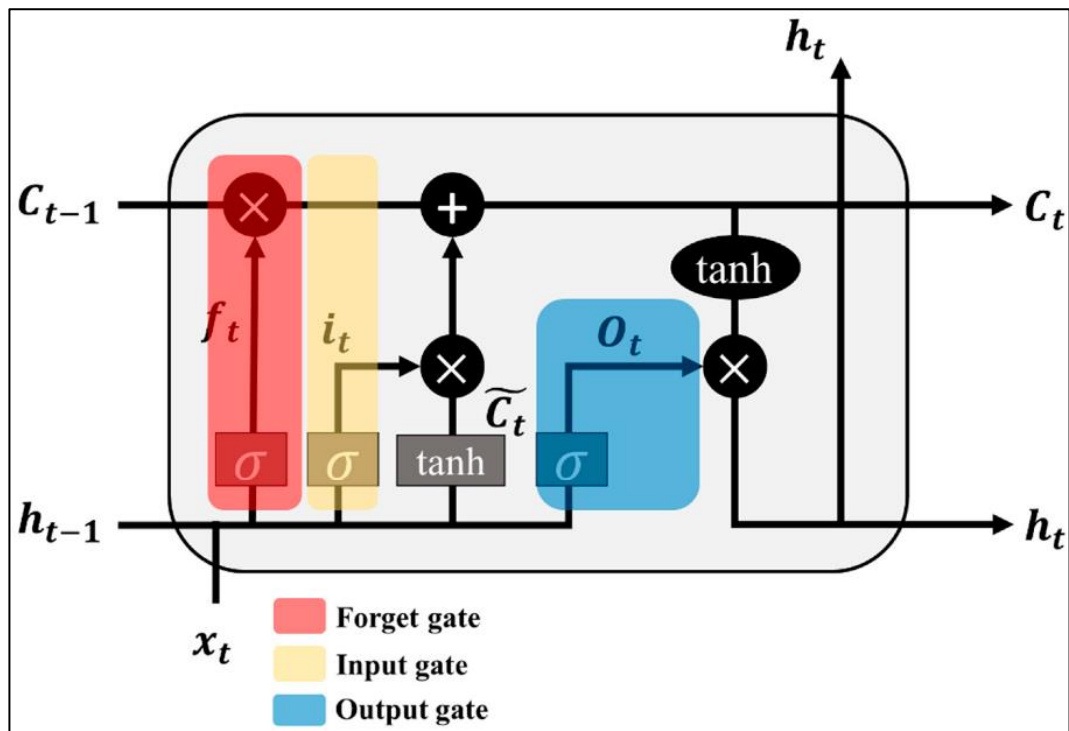
Iš rezultatų akivaizdu teigti, kad SVM modelis pasirodė blogiausiai, o geriausiai – RF modelis, turintis apie 75 % tikslumą. Autorius mašininio mokymo modelio apmokymą atliko naudojant 3703 duomenų įrašų. Alves Coelho taip pat teigia, kad tokie tam tikrų modelių prastesni rezultatai galėtų būti dėl per mažo duomenų rinkinio [5].

### 1.3.3. Dalinai prižiūrimi mašininio mokymo modeliai

Nors ir tokio tipo srityje gauti duomenų žymių yra sunku, tačiau autoriai pasiūlė įvairių algoritmų, kuriuose modelis pats bando priskirti tam tikrą žymę duomenims su tikslu pagerinti mašininio mokymo algoritmo tikslumą. Tolimesniuose skyriuose bus apžvelgiami tokio tipo dalinai prižiūrimi mašininio mokymo modeliai.

#### 1.3.3.1. RNN-LSTM modelis

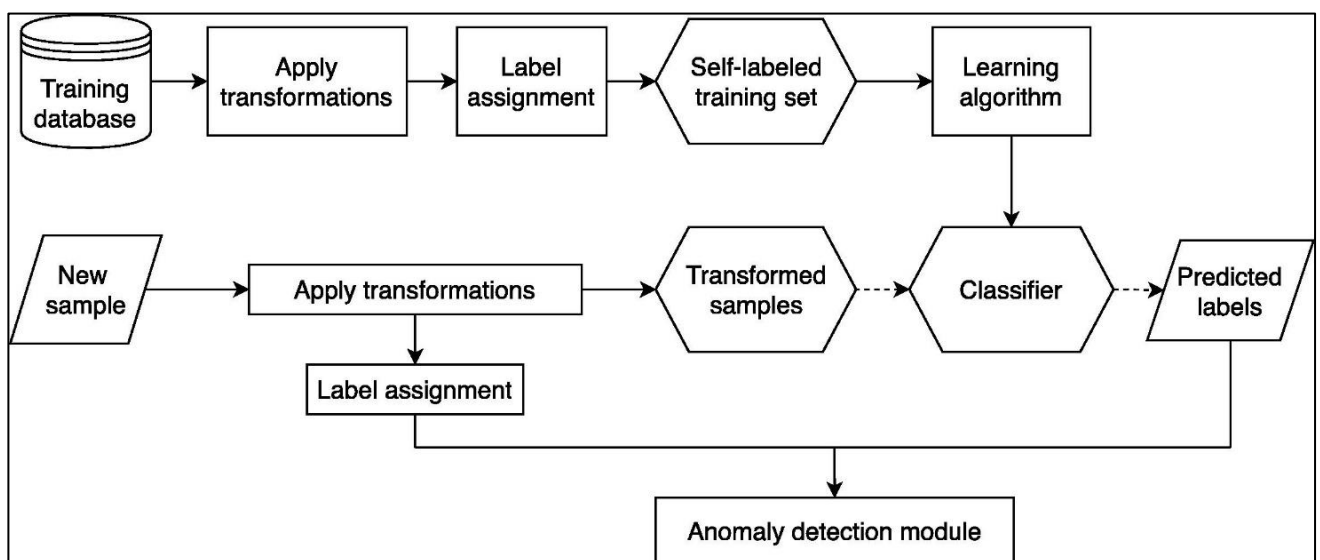
Lee et al. [14] sukūrė RNN-LSTM mašininio mokymo modelį (žr. **5 pav.**) Autoriams ištestavus modelį naudojant realius vandens nuotėkio scenarijų duomenis modelis grąžino didesnę nei 90 % tikslumą.



5 pav. RNN-LSTM modelio struktūra [14]

### 1.3.3.2. SSLD modelis

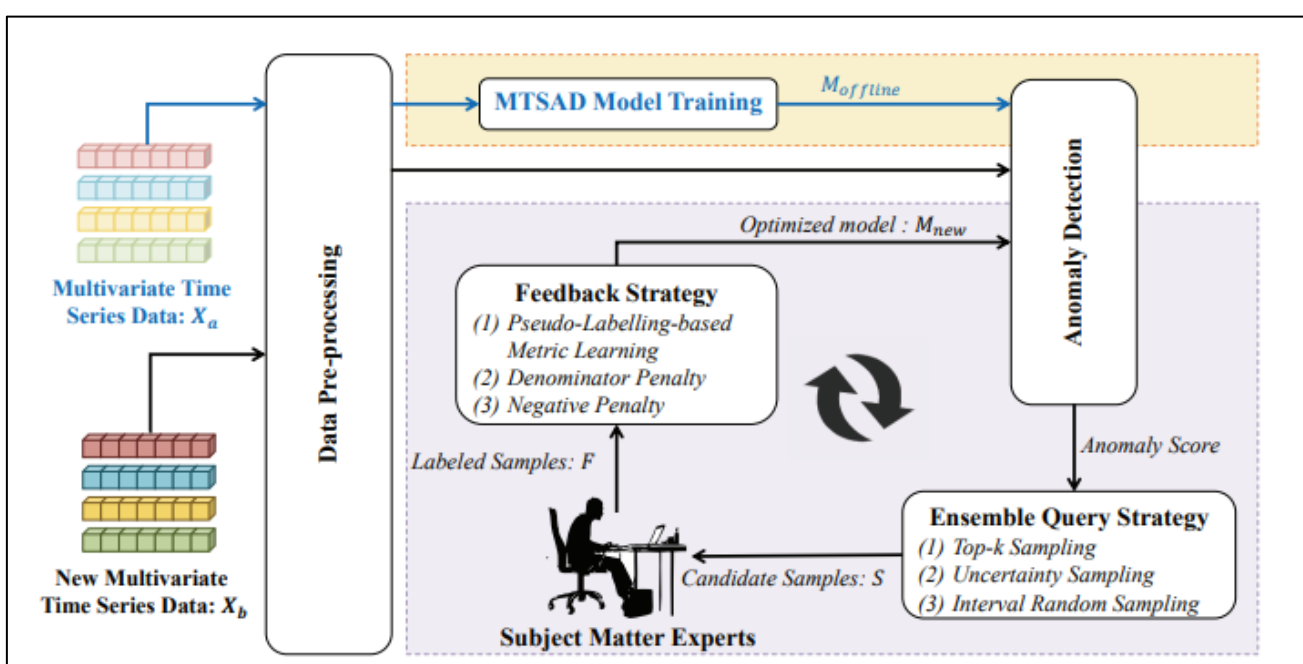
Blázquez-García et al. [15] pasiūlė SSLD modelį. Šis algoritmas nereikalauja jokių duomenų žymių, o vietoj to naudoja tas žymias, kurios buvo dirbtinai priskirtos sugeneruotiems duomenims. Pirmajame etape yra sukuriama save pažymintis (angl. *self-labeled*) mokymo duomenų rinkinys. Vėliau klasifikatorius yra mokinamas susieti tam tikrą duomenų įvesti su atitinkama žyme. Autorių išvadomis SSLD algoritmas suteikia geriausią balansą tarp tikslumo ir klaidingų teigiamų rodiklių (angl. *False Positive Rate (FPR)*). Kadangi modelis yra grindžiamas duomenimis, todėl jam nereikia jokių žinių apie duomenų struktūrą.



6 pav. Dalinai prižiūrimo mašininio mokymo metodas [15]

### 1.3.4. Aktyvaus apmokymo mašininio mokymo algoritmai

Taip pat daugelis autorių pasiūlė savo aktyviu apmokymu grįstus mašininio mokymo algoritmus, kurie gali įtraukti ekspertą ar naudotoją į duomenų žymėjimą taip siekiant prisitaikyti prie tam naudotojo vandens sunaudojimo. Wang et al. [16] pasiūlė aktyviai apmokomą mašininio mokymo algoritmą Active-MTSAD (žr. 7 pav.). Das et al. [17] pasiūlė aktyvų anomalijos atradimo (angl. *Active Anomaly Discovery (AAD)*) algoritmą. Zhu ir Yang [18] pasiūlė triskiltį (angl. *tripartite*) aktyvaus apmokymo metodą. Verduyssen et al. [19] pasiūlė apriboto grupavimo pagrindu remtą (angl. *constrained-clustering-based*) metodą. Aktyvaus apmokymo algoritmų pagrindinė strategija yra atrinkti neįdomius duomenis, kurie turi didžiausią tikimybę būti anomaliniai, ir pateikti juos ekspertui arba naudotojui. Paskui modelis yra atnaujinamas su atnaujintais duomenimis su tikslu prisitaikyti prie naudotojo vandens sunaudojimo. Šis ciklas vyksta tol, kol yra pasiekiamas norimas modelio tikslumas.



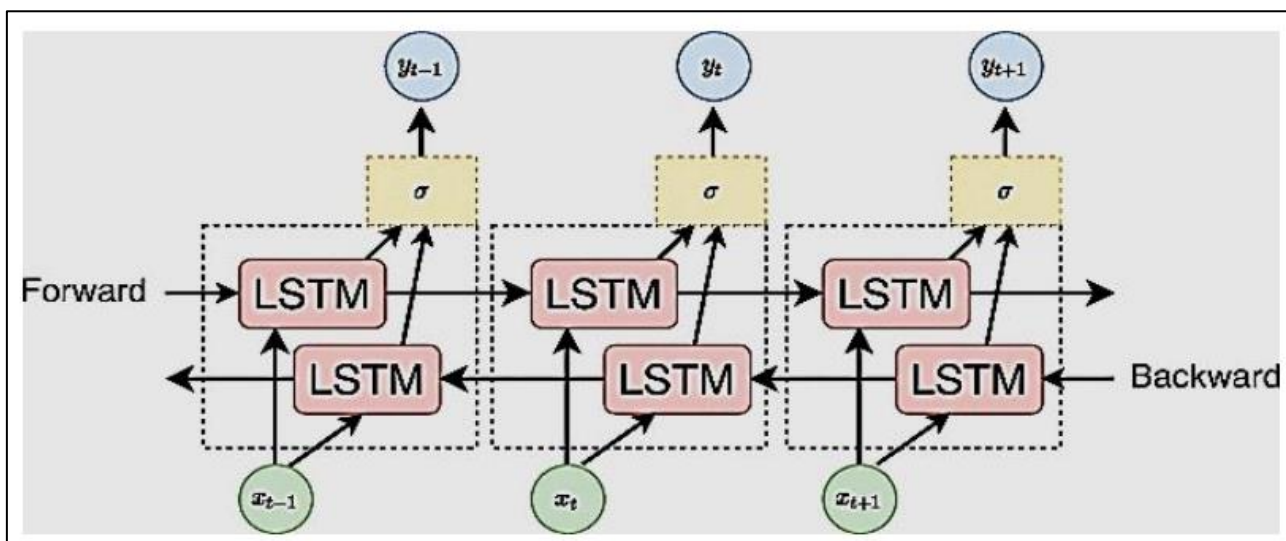
7 pav. Active-MTSAD algoritmo procesas [16]

### 1.3.5. Prižiūravimo mašininio mokymo algoritmai

Prižiūrimas mašininis mokymas vandens anomalijos aptikimo srityje yra sudėtingas, dėl to kad dažnai trūksta sužymėtų duomenų – kas yra vandens nuotėkis, o kas ne. Zese et al. [20] palygino keletą prižiūravimo mašininio mokymo modelių. Autorių rezultatais KNN parodė geriausius rezultatus aptinkant vandens nuotėkį. Galutiniai rezultatai parodė, kad neuroninių tinklų tikslumas, jautrumas ir statistiniai rodikliai „F-measure“ bei „AUC ROC“ siekė tarp 92 % ir 99 %. Fan et al. [21] parodė kaip ANN modelis gali tiksliai klasifikuoti scenarijus kai yra ar nėra vandens nuotėkio, tačiau toks tinklas reikalauja išbalansuotos duomenų struktūros, kurioje būtų tiek vandens nuotėkio scenarijų, tiek normalaus vandens suvartojimo. Nepaisant to autoriai teigia, kad jų neuroninio tinklo tikslumas buvo 100 % [21]. Toliau poskyriuose 1.3.5.1 ir 1.3.5.2 atitinkamai yra pateiktas BiLSTM neuroninis ir kiti įvairių modelių lyginimų tyrimai.

### 1.3.5.1. Dvikryptis ilgalaikės trumpalaikės atminties neuroninis tinklas

Amora et al. [10] suprojektavo BiLSTM neuroninį tinklą (žr. **8 pav.**) ir jį palygino su GRU ir autoregresiniais modeliais. Autorių gauti rezultatai parodė, kad jų pasiūlytas modelis pranoksta GRU ir autoregresinius modelius anomalinio vandens naudojimo aptikimo srityje.



**8 pav.** Dvikrypčio ilgalaikės trumpalaikės atminties modelio architektūra [10]

### 1.3.5.2. Modelių lyginimų tyrimai

Ismail et al. [13] palygino keturis neprižiūravimo mašininio mokymo algoritmus (žr. **9 pav.**). Pirmiausia duomenims buvo priskiriama žymė pagal vandens sunaudojimą – normalus, didelis ir nuotėkis, kur atitinkamai **9 pav.** atvaizduota stulpeliuose „LOW“, „HIGH“ ir „LEAK“. Tada duomenys buvo sunormalizuoti naudojant statistinį rodiklį „z-score“, kuris matuoja kiek standartinių nuokrypių yra tarp duomenų taško ir jo duomenų rinkinio vidurkio. Tai padeda nustatyti kaip koks taškas yra nutolęs nuo vidurkio. Pagal gautus rezultatus akivaizdu teigti, kad RF modelis gražino geriausius rezultatus, kur tikslumas (angl. *accuracy*) yra 87 %, atskirtumo jautrumas (angl. *recall*) - 83 %, o tikslumas (angl. *precision*) – 75 % lyginant su kitais trimis modeliais – DT, KNN ir Naïve Bayes.

Models	Class-wise Comparison			
		LEAK	LOW	HIGH
Decision Tree	PR	66%	84%	84%
	Rec	66%	90%	74%
	Acc	83%		
K-NN	PR	75%	87%	84%
	Rec	65%	90%	80%
	Acc	86%		
Naïve Bayes	PR	72%	91%	55%
	Rec	46%	80%	88%
	Acc	79%		
Random Forest	PR	<u>75%</u>	<u>89%</u>	<u>83%</u>
	Rec	<u>83%</u>	<u>90%</u>	<u>79%</u>
	Acc	<u>87%</u>		

9 pav. Ismail et al. mašininio mokymo modelių analizės palyginimo rezultatai [13]

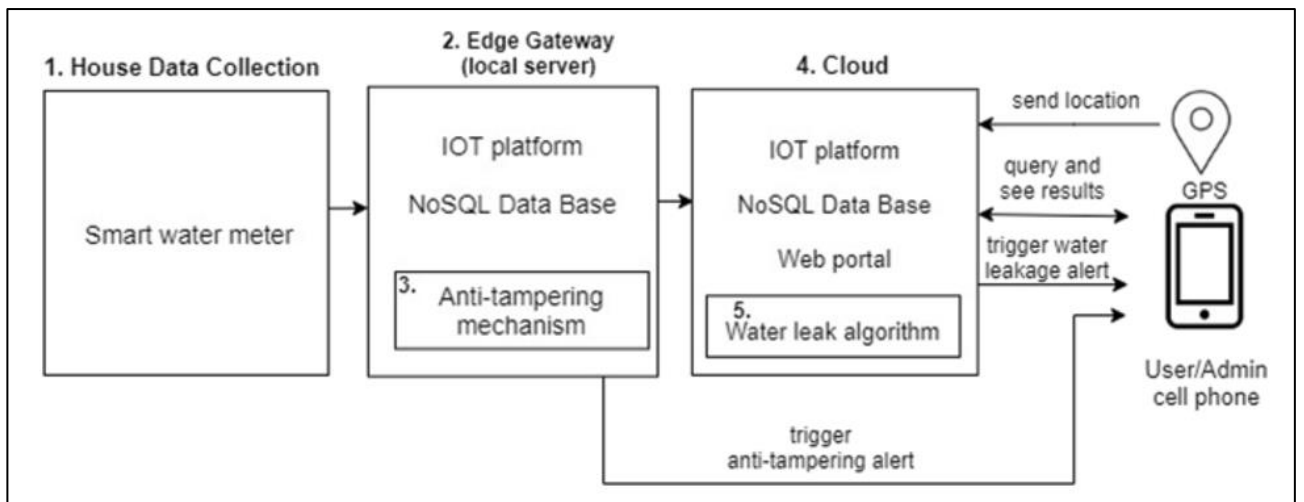
#### 1.4. Naudojamos architektūros, jų veikimo principai ir apribojimai

Vandens nuotėkio aptikimo sistemos gali būti realizuotos taikant įvairias architektūras, kurios bus aprašytos tolimesniuose skyriuose.

##### 1.4.1. Pirma išmanios vandens sistemos suvartojimo architektūra

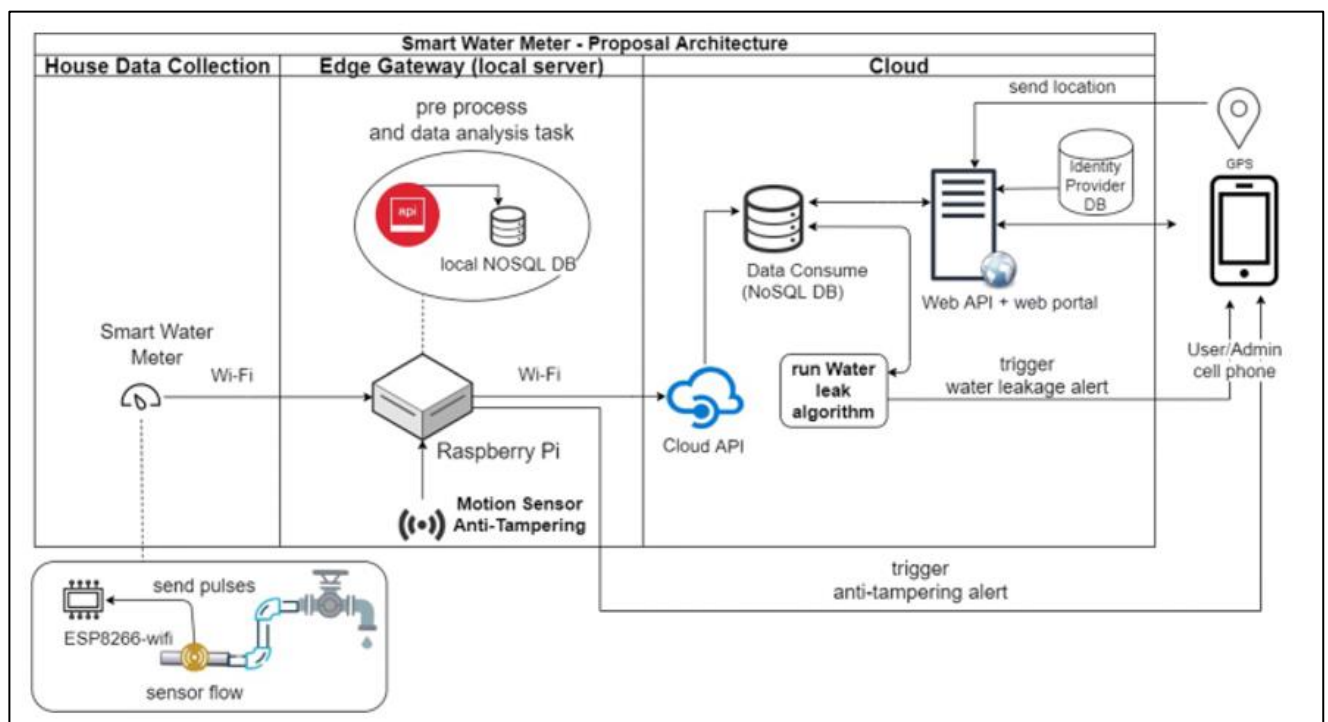
10 pav. yra atvaizduojama penki pagrindiniai sistemos komponentai, kurie leidžia išsaugoti, kaupti, analizuoti bei vizualizuoti vandens sunaudojimą. „House Data Collection“ komponente (laiko tarpas „t“ gali būti 1 minutė) yra išgaunama informacija iš išmanaus jutiklio, kuri yra siunčiama į „Edge Gateway“ komponentą duomenims talpinti [6]. Paskui, per kiekvieną laiko tarpą (gali būti 1 valanda), sukaupti duomenys yra siunčiami į debesų serverį „Cloud“ tam, kad duomenys būtų patalpinti kartu su naudotojo vietoje, kuri yra išgaunama iš naudotojo išmaniajame telefone esančios GPS [6].





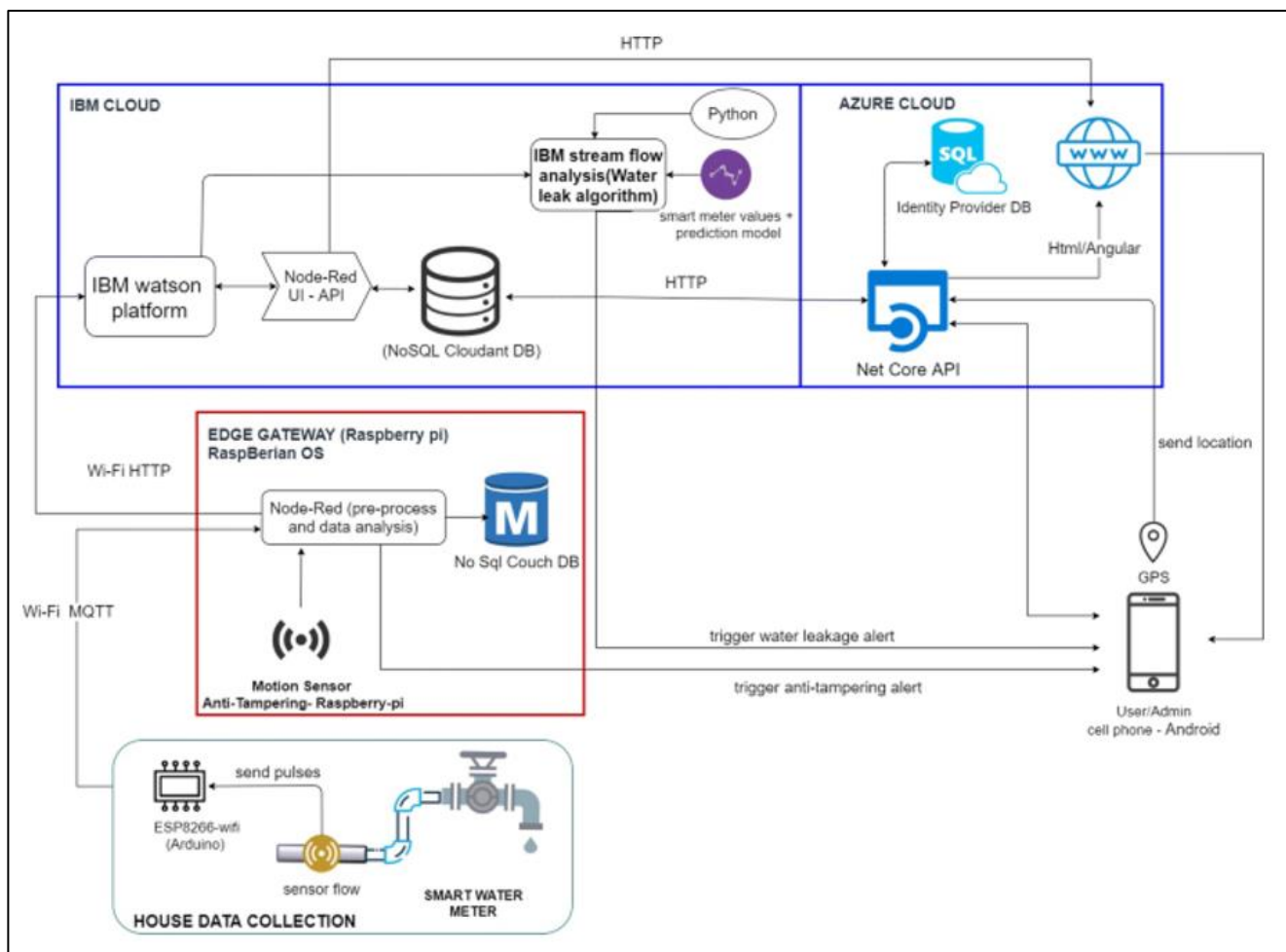
10 pav. Aukšto lygio vandens sistemos suvartojimo architektūros diagrama [6]

11 pav. ir 12 pav. atitinkamai yra atvaizduoti fizinis bei technologinis vaizdas iš pasiūlytos architektūros, kuris sujungia penkis anksčiau aptartus komponentus. Fiziniam vaizde fiziniai įrenginiai yra panaudoti kiekviename komponente kartu su pagrindiniais servisais. Technologiniame vaizde yra vaizduojama, kokie programiniai įrenginių paketai, programavimo kalbos, duomenų bazės bei platformos yra naudojamos [6].



11 pav. Fuentes ir Mauricio pasiūlyta fizinė vandens suvartojimo sistemos architektūra [6]

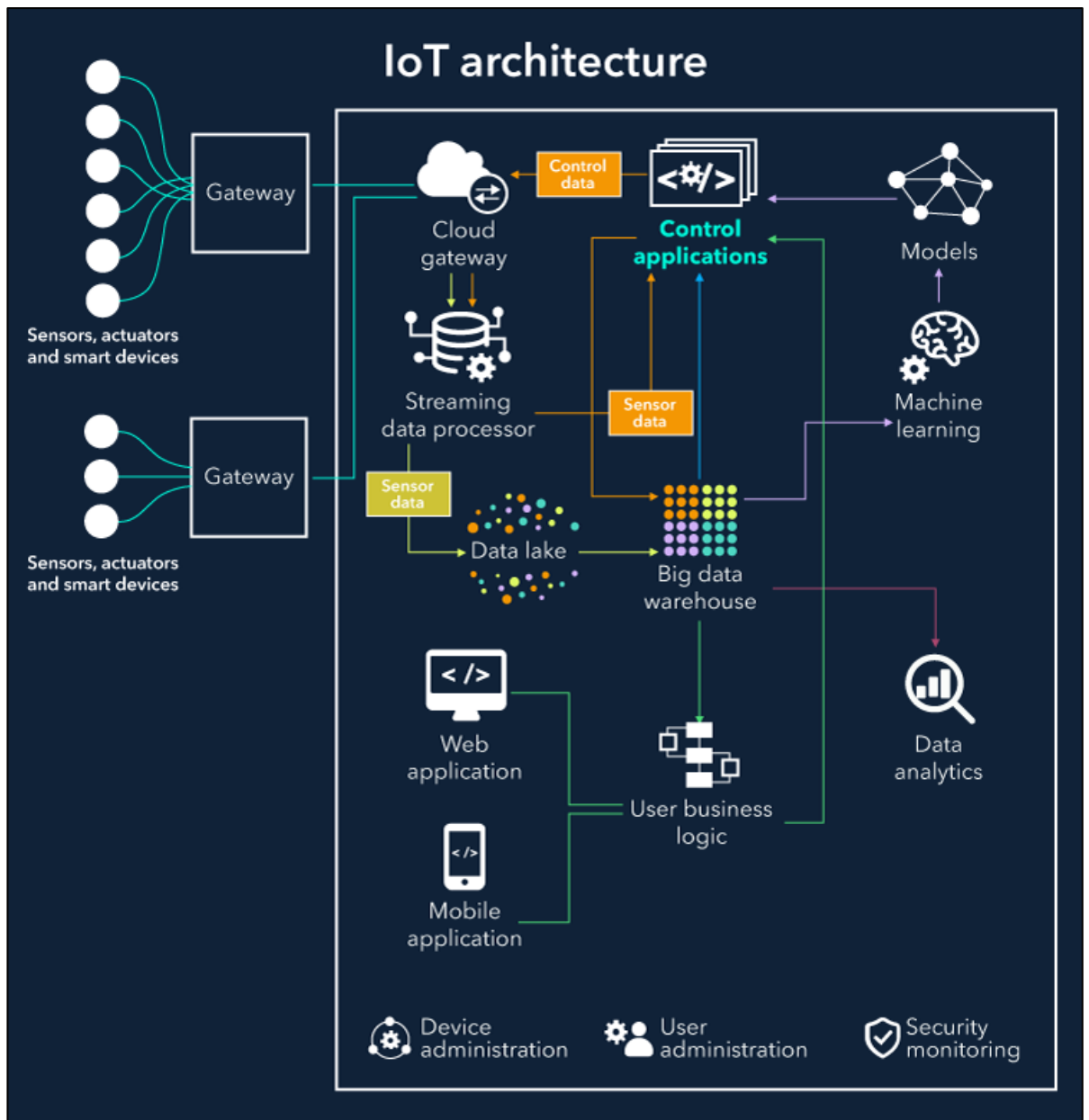




12 pav. Fuentes ir Mauricio pasiūlyta technologinė vandens suvartojimo sistemos architektūra [6]

#### 1.4.2. Antra išmanios vandens sistemos suvartojimo architektūra

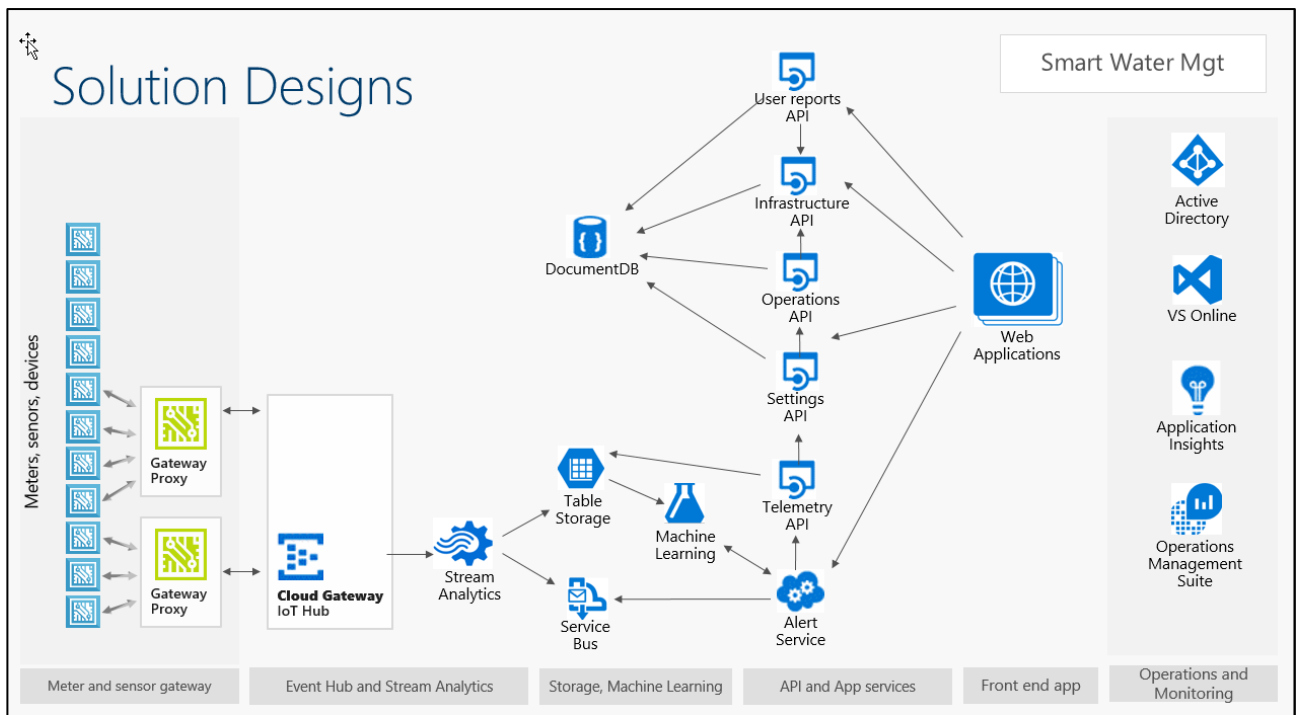
13 pav. yra pavaizduota Daiktų internetu pagrįsta išmanios vandens stebėjimo sistemos architektūra. Ši sistema yra sujungta keliais blokais užtikrinant, kad jutiklio sukurti duomenys yra priimami, talpinami ir apdorojami didžiųjų duomenų saugykloje (angl. *Big Data Warehouse*). Akivaizdu matyti, kad šioje sistemoje vienas iš svarbiausiai išskirtų aspektų yra saugumas, o tai yra didžiausias daiktų interneto temos rūpestis. Duomenys privalo būti saugiai išgauti bei patalpinti, užtikrinant jų vientisumą ir autentiškumą. Siekiant išvengti nemalonių duomenų saugos pažeidimų situacijų, Grizhnevich siūlo registruoti bei analizuoti sistemoje visus atliktus įvykius [6].



13 pav. Daiktų interneto architektūros diagrama [22]

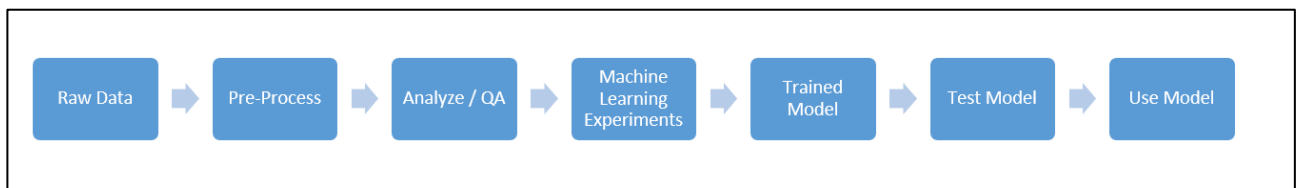
### 1.4.3. Trečia išmanios vandens suvartojimo sistemos architektūra

Trečioji siūloma vandens suvartojimo sistemos architektūra, pagrįsta Daiktų internetu, pavaizduota **14 pav.**, kuri remiasi „Microsoft“ siūlomais duomenų saugojimo ir apdorojimo servisais.



14 pav. „SmartWater“ sprendimo dizainas [23]

Visos eigos procesas yra atvaizduotas 15 pav..



15 pav. „SmartWater“ procesas [23]

Šio procesu metu iš jutiklio yra gaunami duomenys, kurie yra paruošiami apdorojimui ir analizei. Paruošti duomenys yra perduodami į mašininio mokymo eksperimento aplinką, kurioje mašininio mokymo modelis yra apmokomas bei ištestuojamas. Su gautu modeliu bus galima atlikti spėjimus tolimesniems spėjimams apie suvartojamą vandenį vandens nuotėkiui numatyti.

### 1.5. Egzistuojančių algoritmų palyginimas

2 lentelėje yra palyginami tradiciniai algoritmai, 3 lentelėje – neprižiūrimo mašininio mokymo algoritmai, 4 lentelėje – prižiūrimo mašininio mokymo algoritmai.

2 lentelė. Tradicinių algoritmų palyginimas

Algoritmas	Tikslumas
Akselerometru grįstas	92.1

3 lentelė Neprižiūrimo mašininio mokymo algoritmų palyginimas

Algoritmas	Tikslumas	Precizija	Jautrumas	F1 reikšmė	MAE
MFN	89 %	5 %	2 %	3 %	-

CF	98 %	21 %	100 %	98 %	-
AVG	84 %	53 %	10 %	17 %	-
Keturių scenarijų	100 %	100 %	100 %	100 %	-
ARIMA	64 %	-	-	-	39230.88
Fbprophet	55.5 %	-	-	-	20600.85
DT	83 %	78 %	76.7 %	-	-
K-NN	86 %	82 %	78.3 %	-	-
Naïve Bayes	79 %	72.7 %	71.3 %	-	-
RF	87 %	82.3 %	84 %	-	-
DT	76 %	-	-	-	-
NN	67.5 %	-	-	-	-
SVIM	40.8 %	-	-	-	-
XGBoost	77.6 %	-	-	-	-

**4 lentelė.** Prižiūravimo mašininio mokymo algoritmų palyginimas

Algoritmas	Tikslumas	Jautrumas	F-reikšmė	AUC ROC
Konvoliuciniai neuroniniai tinklai	92-99 %	92-99 %	92-99 %	92-99%
ANN	100 %	-	-	-

Iš palyginimo rezultatų galima teigti, kad geriausiai pasirodė prižiūravimo mašininio mokymo algoritmai įtraukiant konvoliucinius neuroninius tinklus ir dirbtinį intelektą. Tarp neprižiūravimo mašininio mokymo algoritmų geriausiai pasirodė keturių scenarijų ir atsitiktinių miškų algoritmai.

### 1.6. Analizės išvados

Išanalizavus įvairias mokslines publikacijas buvo rasta daug darbų, susijusių su anomalinio vandens naudojimo aptikimu, įtraukiant tiek tradicinius, neprižiūravimo, dalinai prižiūravimo ir prižiūravimo mašininio mokymo algoritmus. Gauti rezultatai kai kuriais atvejais yra stulbinantys, kurie parodė net 100 % tikslumą. Tačiau reikia įvertinti ir tai, kad modelių rezultatai labai priklauso nuo duomenų kokybės ir kiekio, taigi aklaui teigti, kad vienas ar kitas metodas turėtų būti pasirinktas kaip pagrindinis, nereikėtų. Taip pat daugelis tyrėjų neįvertino precizijos, jautrumo, F1 bei ankstyvių aptikimo įvertinimo reikšmių, kurios yra labai svarbios įvertinant modelio efektyvumą aptikti vandens nuotėkiui. Taigi, tuo remiantis šiame darbe ir bus siekiama palyginti įvairius modelius įvertinant preciziją, jautrumą, F1 reikšmę, AUC-ROC ir AUC-PR įverčius bei ankstyvaus aptikimo reikšmę siekiant išsiaiškinti kiekvieno modelio efektyvumą nustatant vandens nuotėkį.

## 2. Projektinė dalis

### 2.1. Reikalavimų specifikacija

#### 2.1.1. Sistemos paskirtis

##### 2.1.1.1. Projekto kūrimo pagrindas (pagrindimas)

Aš inicijuojau išmanaus pastato jutiklių stebėjimo ir anomalinio vandens naudojimo sistemos kūrimo projektą, kurio idėja yra palengvinti vandens nuotėkio aptikimą bei suvartojamo kiekio stebėjimą realiu laiku. Šiuo metu esamos sistemos suteikia ribotą funkcionalumą bei nesuteikia aukšto lygio vandens nuotėkio aptikimo. Taip pat šis projektas prijungia ir temperatūros jutiklių stebėjimą, ko kitose programose ar sistemose yra stoka. Šio projekto kūrimo priežastys yra šios:

- dauguma rinkoje esančių vandens nuotėkio aptikimo įrenginių-jutiklių nėra „protingi“ ir reikalauja įrengimų tose vietose, kur dažniausiai būtų galima aptikti vandenį;
- rinkoje esantys išmanūs jutikliai nėra greitai besimokantys ir reikalauja daug mėnesių tam, kad išmoktų ir prisitaikytų prie naudotojo suvartojamos vandens istorijos;
- egzistuojantys sprendimai yra orientuoti tik į vieno tipo jutiklių stebėjimą.

##### 2.1.1.2. Sistemos tikslai (paskirtis)

Apibendrinti sistemos kūrimo tikslai:

#### 1. Leisti naudotojams parsisiųsti programėlę.

- **Pagrindimas:** programėle turėtų naudotis tam tikra dalis žmonių, kad projekto įmonė galėtų toliau tęsti savo veiklą.
- **Tenkinimo kriterijus:** per mėnesį „Google Play“ platformoje turėtų būti bent 500 atsisiuntimų.

#### 2. Apsaugoti naudotojus nuo galimo vandens nuotėkio.

- **Pagrindimas:** naudotojams yra labai svarbu gauti aliarminius pranešimus apie galimą vandens nuotėkį bei apie per aukštą ar žemą temperatūrą, taip leidžiant išvengti nemalonių situacijų.
- **Tenkinimo kriterijus:** po 2 mėnesių bus atlikta programėlės naudotojų apklausa, kurios metu 95 % apklaustųjų pasakys, kad programėlė leido išvengti avarinių atvejų.

#### 3. Leisti naudotojams patogiai stebėti temperatūrą bei suvartojamą vandenį.

- **Pagrindimas:** naudotojams yra svarbu matyti ir planuoti savo suvartojamą vandenį taip taupant pinigus.
- **Tenkinimo kriterijus:** praėjus 3 mėnesiams nuo sistemos įdiegimo bus atlikta anoniminė sistemos naudotojų apklausa, kurios metu ne mažiau nei 80 % apklaustųjų įvertins, kad programėlė leido sutaupyti šildymo resursų bei pinigų, negu nenaudojant programėlės.

#### 4. Palengvinti jutiklių valdymą.

- **Pagrindimas:** naudotojai jaučiasi gerai, kai jie gali kontroliuoti jų esamą įrangą.

- **Tenkinimo kriterijus:** praėjus savaitei nuo sistemos įdiegimo bus atlikta anoniminė sistemos naudotojų apklausa, kurios metu ne mažiau nei 85 % apklaustųjų įvertins, kad programėlė palengvino jutiklių valdymą.

5. Leisti naudotojams lengviau nustatyti anomalinio vandens naudojimo priežastis.

- **Pagrindimas:** naudotojams yra naudinga žinoti galimas vandens naudojimo priežastis.
- **Tenkinimo kriterijus:** praėjus savaitei nuo sistemos įdiegimo bus atlikta anoniminė sistemos naudotojų apklausa, kurios metu ne mažiau nei 85 % apklaustųjų įvertins, kad programėlė leido lengviau nustatyti anomalinio vandens naudojimo priežastis.

## **2.1.2. Užsakovai, pirkėjai ir kiti sistema suinteresuoti asmenys**

### **2.1.2.1. Užsakovas (-ai)**

**Projekto užsakovas** – Lukas Kulikovas ir Šarūnas Packevičius. Aš, kaip užsakovas ir sistemos kūrėjas, būsiu atsakingas už:

- projekto finansavimą nuo pradžios iki pabaigos, įskaitant tiek programinės tiek techninės įrangos palaikymą;
- dalyvausiu reikalavimų surinkime, teikime bei vertinime;
- dalyvausiu testavime, programinės įrangos kūrime bei techninės įrangos paskirstyme;
- dalyvausiu programos marketingo sprendimų priėmime.

Šarūnas Packevičius bus atsakingas už:

- projektavimo metodologijų ir technologijų analizės konsultacijas;
- projekto paraiškos konsultacijas;
- projekto reikalavimų specifikacijos konsultacijas;
- sistemos įdiegimo vertinimą;
- ekspertinį vertinimą;
- projekto kontrolę.

### **Užsakovų kontaktai:**

- el. paštas: [lukas.kulikovas@ktu.edu](mailto:lukas.kulikovas@ktu.edu);
- el. paštas: [sarunas.packevicius@ktu.lt](mailto:sarunas.packevicius@ktu.lt).

### **2.1.2.2. Pirkėjas (-ai)**

Projekto užsakovas ir pirkėjas yra tas pats asmuo.

### **2.1.2.3. Kiti suinteresuoti asmenys**

Į šio produkto suinteresuotus asmenis yra įtraukiamas projekto vadovas – jis pateiks naudingos informacijos bei išvalgą apie sistemos dizainą bei patogų programėlės valdymą, dalyvaus projekto eigoje ir stebėjime.

## **2.1.3. Vartotojai**

Būsiami produkto naudotojai:

## 1. Pastatų savininkai

- **Tikslai:** išvengti nelaimingų vandens nuotėkio atsitikimų, per aukštos ar žemos temperatūros avaringų atvejų, lengviau prižiūrėti turimus jutiklius bei sutaupyti ir stebėti vandens sunaudojimą.
- **Bendros charakteristikos:** pastatų savininkai arba priežiūros atstovai, mokantys anglų kalbą.
- **Patirtis dalykinėje srityje:** vidutinė.
- **Patirtis technologinėje srityje:** vidutinė.
- **Prioritetas:** aukštas.
- **Dalyvavimas projekte:** naudosis programėle.

## 2. Svečias

- **Tikslai:** susipažinti su programėlės siūlomu funkcionalumu.
- **Bendros charakteristikos:** pastatų savininkai arba priežiūros atstovai, mokantys anglų kalbą.
- **Patirtis dalykinėje srityje:** vidutinė.
- **Patirtis technologinėje srityje:** vidutinė.
- **Prioritetas:** vidutinis.

## 3. Testuotojas

- **Tikslai:** turėti galimybę naudotis dar neišleista programėlės versija, siekiant tobulinti tolimesnės programėlės versijas.
- **Bendros charakteristikos:** baigę aukštąjį išsilavinimą, jaunesnio amžiaus žmonės, vyrai.
- **Patirtis dalykinėje srityje:** aukšta.
- **Patirtis technologinėje srityje:** aukšta.
- **Prioritetas:** žemas.
- **Dalyvavimas projekte:** testuos tarpinius prototipus.

### 2.1.4. Įpareigojantys apribojimai

#### 2.1.4.1. Apribojimai sprendimui

Mobiliosios vandens nuotėkio stebėjimo programėlės keliami apribojimai:

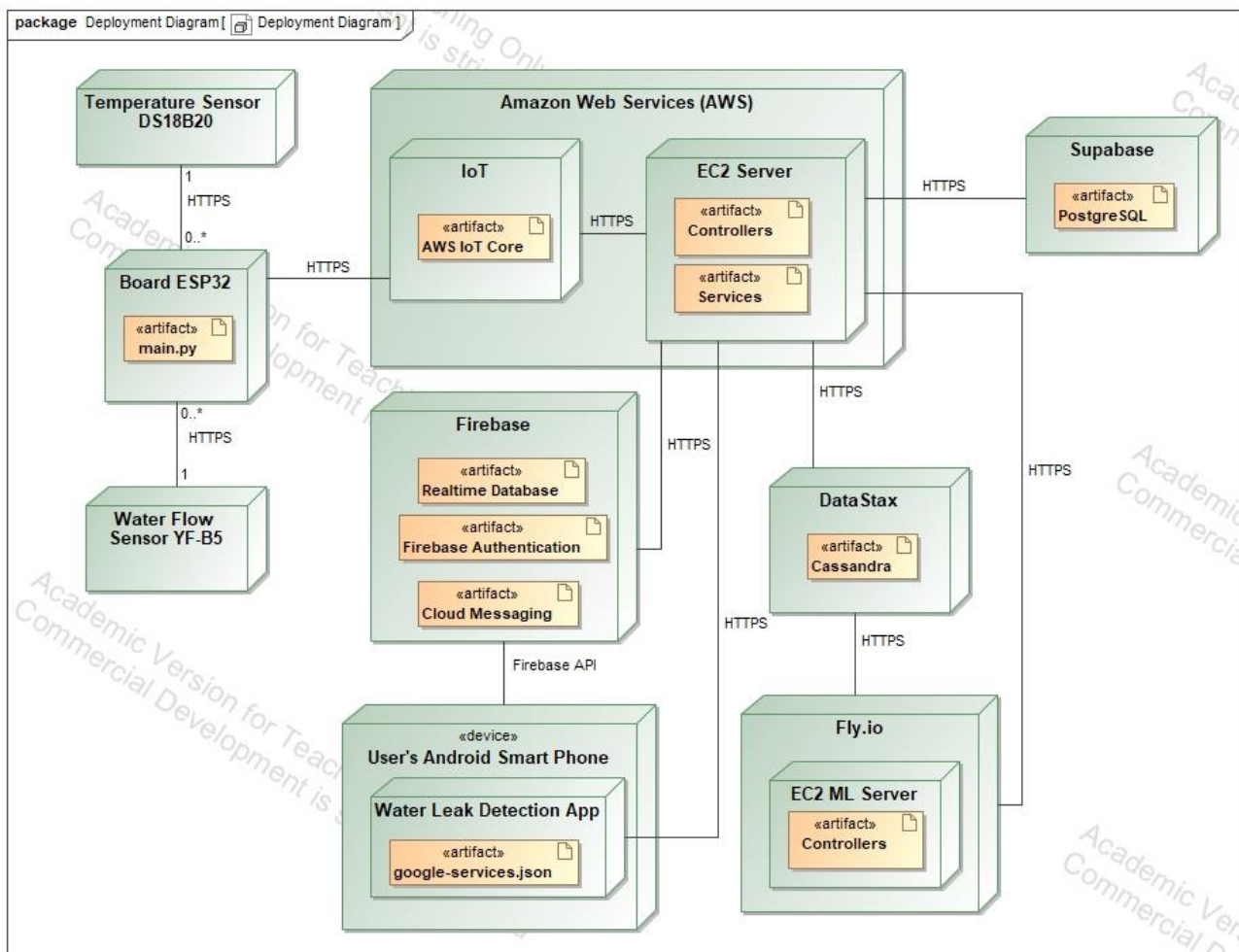
#### 1. Kuriama programėlė turi būti sukurta neviršijant numatyto biudžeto.

- **Pagrindimas:** tai yra užsakovo maksimaliai turimas ir galimas finansų kiekis, skirtas programėlei realizuoti.
- **Tenkinimo kriterijus:** programėlė neturi viršyti 11499 eurų ribos.

#### 2. Kuriama programėlė bus orientuota į įrenginius, turinčius „Android“ operacinę sistemą.

- **Pagrindimas:** užsakovo darbo vietoje programinė įranga yra specialiai skirta kurti „Android“ operacinę sistemą turintiems įrenginiams.
- **Tenkinimo kriterijus:** sistema turi būti skirta „Android 5.0“ ar naujesnę operacinę sistemą turintiems įrenginiams.

## 2.1.4.2. Diegimo aplinka



16 pav. Sistemos išdėstymo diagrama

Sistemos išdėstymą sudaro šios aplinkos:

- mikrovaldiklis „ESP-WROOM-32“;
- vandens „YF-B5“ ir temperatūros „DS18B20“ jutikliai;
- „AWS“ debesų kompiuterijos paslauga, kurioje yra „AWS IoT Core“ bei „EC2“ debesų kompiuterijos platformos;
- „Firebase“ debesų kompiuterijos paslauga, kurioje yra aktyvuotos „Realtime Database“, „Firebase Authentication“ ir „Cloud Messaging“ paslaugos;
- naudotojo išmanusis įrenginys, kurioje yra įdiegta programėlė;
- „DataStax“ realaus laiko duomenų talpinimo platforma, kurioje yra „Cassandra“ duomenų bazė;
- „Fly.io“ debesų kompiuterijos platforma, kurioje yra patalpintas anomalinio vandens naudojimo aptikimo algoritmas;
- „Supabase“ debesų kompiuterijos platforma, kurioje yra patalpinta „PostgreSQL“ duomenų bazė.



### 2.1.4.3. Bendradarbiaujančios sistemos

1. Kiekvienam jutikliui priklausantis „ESP-WROOM-32“ mikrovaldiklis yra atsakingas už matavimų (atitinkamai vandens suvartojimo ir temperatūros) išgavimą iš jutiklio ir perdavimą į „IoT Core“ debesų servisą (duomenų tarpininką);
2. „EC2“ serveris yra atsakingas už duomenų priėmimą iš „IoT Core“ debesų serviso, jų išsaugojimą „DataStax“ platformoje esančioje „Cassandra“ duomenų bazėje bei visapusišką duomenų mainų bendradarbiavimą tarp „Firebase“ paslaugos ir naudotojo;
3. „Firebase“ yra atsakingas už duomenų apie jutiklius, naudotojus bei prisijungimo žetonų saugojimą ir realaus laiko duomenų sinchronizavimą („Realtime Database“), vartotojų autentifikavimą („Firebase Authentication“) bei aliarminių žinučių siuntimą („Cloud Messaging“);
4. naudotojo išmanusis įrenginys, turintis *google-service.json* konfigūracijos failą, leidžiantį komunikuoti su „Firebase“ per „Firebase API“, yra atsakingas už duomenų sinchronizavimą tarp mobiliosios programėlės ir serverio, taip pat už įspėjamųjų pranešimų priėmimą iš „Firebase“ paslaugos;
5. „EC2 ML“ serveris yra atsakingas už anomalinio vandens naudojimo algoritmo vykdymą, galimo vandens nuotėkio įspėjimo signalo siuntimą bei kasdieninio vandens naudojimo ataskaitų generavimą ir jų perdavimą į „EC2“ serverį;
6. „DataStax“ talpinantis „Cassandra“ duomenų bazę yra atsakingas už duomenų, gautų iš jutiklių, talpinimą;
7. „Fly.io“ paslauga yra atsakinga už „EC2 ML“ serverio talpinimą;
8. „Supabase“ talpinantis „PostgreSQL“ duomenų bazę yra atsakingas už duomenų apie jutiklius, naudotojus ir naudotojų jutiklius talpinimą.

### 2.1.4.4. Komerciniai specializuoti programų paketai

Programinė įranga, kurią privaloma naudoti šio projekto rėmuose:

1. „Android Studio“ integruota kūrimo platforma.
  - **Pagrindimas:** tai yra oficiali platforma „Android“ operacinę sistemą turinčių išmaniųjų įrenginių programų kūrimui bei taip nurodyta darbo reikalavimuose.
2. „Kotlin“ programavimo kalba.
  - **Pagrindimas:** tai yra oficiali programavimo kalba kuriant „Android“ operacinę sistemą turinčių išmaniųjų įrenginių programėlių kūrimui su „Android Studio“ platforma.
3. „IntelliJ IDEA“ kūrimo platforma.
  - **Pagrindimas:** ši platforma yra gausiai dokumentuota, leidžia išvengti daug pasikartojančio kodo, suteikia sklandžią projekto kūrimo patirtį nuo dizaino, įgyvendinimo, kūrimo bei testavimo iki pat atnaujinimo bei pertvarkymo.
4. „Java“ programavimo kalba.
  - **Pagrindimas:** ši programavimo kalba yra šiuo metu naudojama kuriant esamą sistemą.
5. „Visual Studio Code“ kodo redaktorius.

- **Pagrindimas:** šis kodo redaktorius turi begalę bibliotekų bei palaikomų programavimo kalbų, suteikia galimybę testuoti bei paleisti kodą, be to jis yra naudojamas šiuo metu kuriamoje sistemoje.
6. „Spring Framework“ karkasas.
- **Pagrindimas:** šis karkasas yra labai plačiai dokumentuotas, ilgai esantis bei turintis didelę bendruomenę.
7. „GitHub“ kodo versijavimo sistema.
- **Pagrindimas:** ši sistema yra šiuo metu naudojama kodui talpinti bei versijuoti.
8. Debesų kompiuterijos paslauga „Firebase“.
- **Pagrindimas:** leidžia tiek saugoti, tiek sinchronizuoti realiu laiku duomenis tarp programėlės naudotojų, suteikia palengvinto testavimo galimybes, leidžia integruoti reklamas.
9. Debesų kompiuterijos paslauga „Amazon AWS“.
- **Pagrindimas:** ši paslauga suteikia serverio dalies kodo talpinimą, suteikia galimybę peržiūrėti ir stebėti suvartojamus resursus bei ateityje leis lengviau plėstis bei tobulėti.
10. „C++“ programavimo kalba.
- **Pagrindimas:** tai yra oficiali programavimo kalba programuojant „ESP-32“ jutiklius.
11. „Python“ programavimo kalba.
- **Pagrindimas:** tai yra labai populiari programavimo kalba, turinti begalę bibliotekų, kurios be galo padeda kurti mašininio ar dirbtinio intelekto modelius bei algoritmus.
12. „DataStax“ realaus laiko duomenų platforma.
- **Pagrindimas:** ši platforma leidžia sutaupyti infrastruktūros kaštus, taip pat suteikia saugią terpę talpinti duomenis.
13. „Supabase“ debesų paslauga.
- **Pagrindimas:** ši paslauga leidžia sutaupyti infrastruktūros kaštų.
14. „Fly.io“ debesų paslauga.
- **Pagrindimas:** ši paslauga leidžia sutaupyti infrastruktūros kaštų.

#### 2.1.4.5. Numatoma darbo vietos aplinka

Programėlė gali būti naudojama bet kur, kur yra interneto ryšys.

#### 2.1.4.6. Sistemos kūrimo terminai

Pageidautina, kad programėlė būtų sukurta ne vėliau nei per vienerius metus, o sistemos prototipas jau būtų matomas ne vėliau nei po dviejų mėnesių.

#### **2.1.4.7. Sistemos kūrimo biudžetas**

Sistema privalo būti sukurta už ne daugiau nei 11499 eurų.

#### **2.1.5. Svarbūs faktai ir prielaidos**

##### **2.1.5.1. Faktai**

1. Naudotojai turės vieną vandens tėkmės greičio matavimo jutiklį.
2. Naudotojai turės daugiau nei vieną temperatūros matavimo jutiklį.

##### **2.1.5.2. Veiklos taisyklės**

1. Vandens matavimo jutiklio duomenys yra atnaujinami kas 1 minutę.
2. Jutiklio vidutinis vandens sunaudojimas yra apskaičiuojamas pagal formulę:  
(visas vandens sunaudotas kiekis / atnaujinimų skaičius).
3. Maksimalus vandens tėkmės greitis, kurį jutiklius gali skaičiuoti, yra 30 litrų per minutę.
4. Temperatūros matavimo jutiklio duomenys yra atnaujinami kas 5 minutes.
5. Jutiklių vidutinė temperatūra apskaičiuojama pagal formulę:  
(visų temperatūrų suma / visų temperatūrų kiekio).

##### **2.1.6. Prielaidos**

Yra manoma, kad naudotojai žino anglų kalbą.

##### **2.1.7. Veiklos sudėtis**

Šiai veiklai daryti šiuo metu yra naudotini tokie produktai, kurie yra įvairiai pasiskirstę tarp technologiško lygio išsivystymo. Toliau bus pateikti šiuo metu naudojamų sprendimų paaiškinimai.

Šiuo metu vienas iš paprasčiausių ir populiariausių egzistuojančių būdų nustatyti vandens nuotėkį yra padėti drėgmės aptikimo jutiklius į vietas, kurios dažniausiai tikėtina, jog galėtų sukelti vandens nuotėkį. Šiuo atveju naudotojui reiktų įsirengti didelį kiekį tokių jutiklių, pastoviai stebėti ir prižiūrėti jų techninę būseną, o galiausiai – ne visais atvejais tokio tipo jutikliai gali aptikti vandens nuotėkį. Jeigu, pavyzdžiui, tualetu bakelis nepilnai užsidarytų, vanduo tekėtų pastoviai, o tokio tipo jutikliai, aprašyti aukščiau, negalėtų aptikti nuotėkio ir vartotojas patirtų didelę piniginę žalą.

Išmanesni sprendimai, kurie remiasi jutiklio montavimu ant pagrindinio vandens skaitliuko arba į vamzdį, turi didesnę tikimybę aptikti vandens nuotėkį. Tokių jutiklių reiktų įrengti tik vieną, tačiau tokio tipo jutikliai yra brangūs, o tie, kurie naudoja mašininį mokymą, reikalauja labai daug laiko, kad prisitaikytų prie naudotojo suvartojamo vandens duomenų.

Žvelgiant iš šiuo metu egzistuojančių temperatūros matavimo būdų, vienas iš paprasčiausiai egzistuojančių būdų nustatyti vamzdžio temperatūrai yra užmontuoti temperatūros matuoklį ant vamzdžio, kuris atvaizduos esamą vandens temperatūrą vamzdyje. Šiuo atveju naudotojas negaus jokių įspėjamųjų signalų - jis pats turės nueiti prie temperatūros matuoklio ir patikrinti esamą temperatūros situaciją.

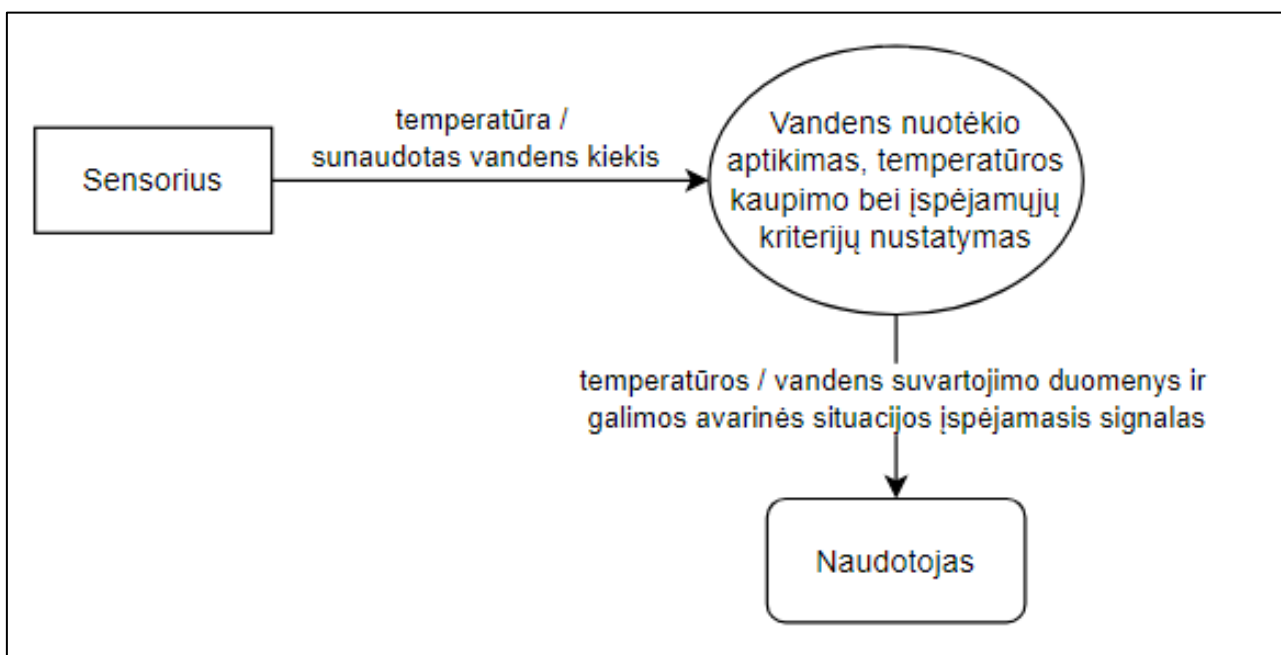
Šiek tiek išmanesni sprendimai, kurie taip pat remiasi jutiklio montavimu ant vandens vamzdžio, tačiau šiuo atveju pasiekus tam tikrą kritinę ribą, kuri negali būti nustatoma jokių rankiniu būdu, jutiklius pradeda skleisti garsų signalą, atkreipiantį naudotojo dėmesį apie galimai per aukštą

temperatūrą. Be to, esamos sistemos nepraneša apie per žemą temperatūrą. Tokie sprendimai yra pakankamai pavojingi, kadangi patys jutikliai nėra pilnai saugūs montuoti ant vamzdžių, dėl to, kad pasiekus aukštesnei nei 100 laipsnių pagal Celsijų temperatūrai, jutiklio apsaugos gali išsilydyti ar užsidegti, kadangi ne visas jutiklius yra sudarytas iš metalo.

Nepaisant neigiamų savybių, šiuo metu įsigijus vieną iš aukščiau paminėtų produktų jį galima lengvai įsirengti ir pačiam, lengvai užmontuojant jutiklių ant vandens vamzdžio.

### 2.1.7.1. Veiklos kontekstas

17 pav. yra pateiktas kuriamos sistemos kontekstas.



17 pav. Sistemos konteksto diagrama

### 2.1.7.2. Veiklos padalinimas

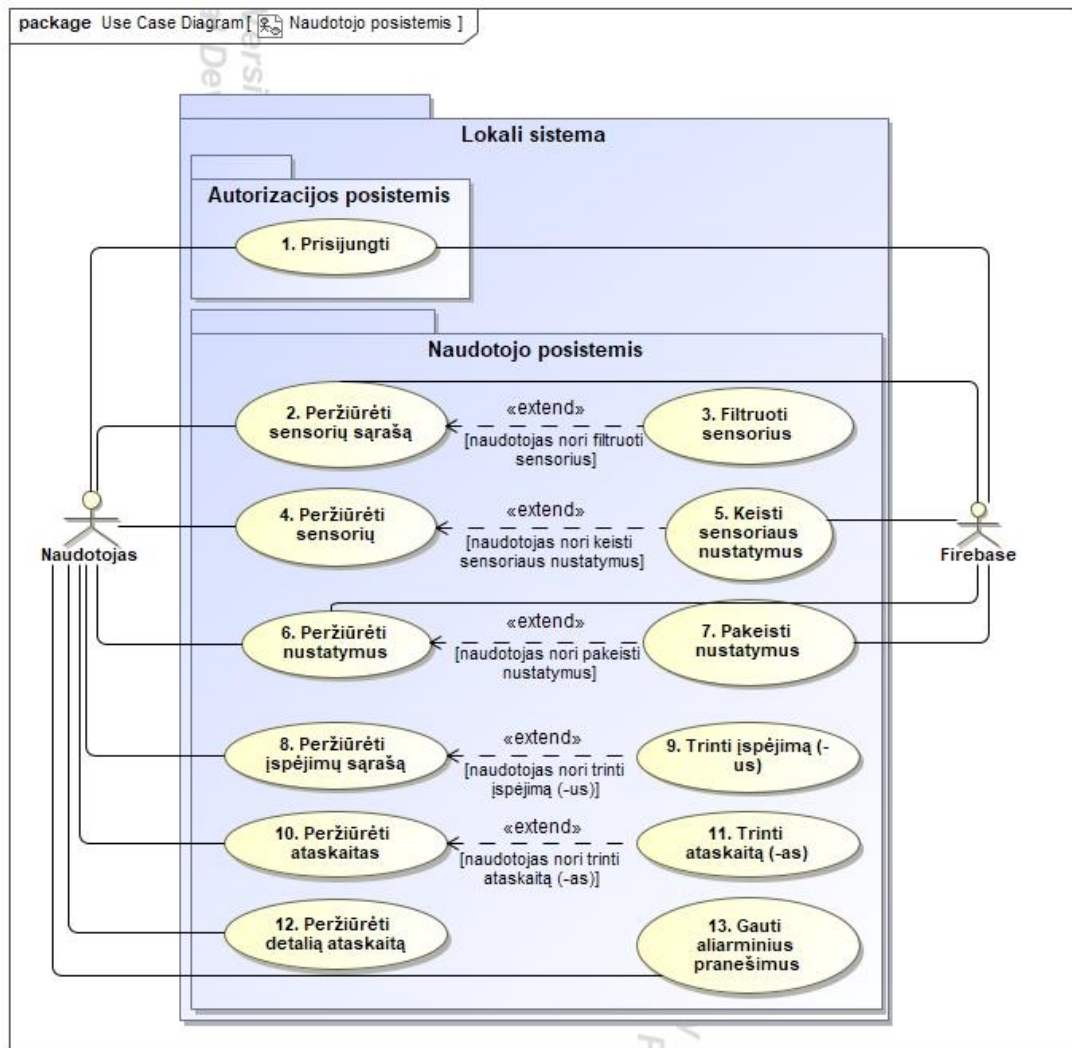
5 lentelė Esamos situacijos veiklos

Srauto pavadinimas	Įeinantis/Išeinantis	Aprašas
Temperatūros / vandens suvartojimo kiekis	Įeinantis	Temperatūros rodiklių bei vandens suvartojimo kiekis yra nuolatos matuojamas daviklio, kur pagrindine varijuoja daviklio rezultatas (temperatūra arba vandens suvartotas kiekis), dviem skaičiais po kabelio bei duomenų nuskaitymo laikas, išreikštas laiko žyma (angl. <i>timestamp</i> ).
Temperatūros / vandens suvartojimo duomenys ir galimos avarinės situacijos signalas	Išeinantis	Temperatūros ir vandens suvartojimas yra atvaizduojamas naudotojui, o pasiekus kritinę reikšmę naudotojui yra išsiunčiamas įspėjamasis pranešimas.

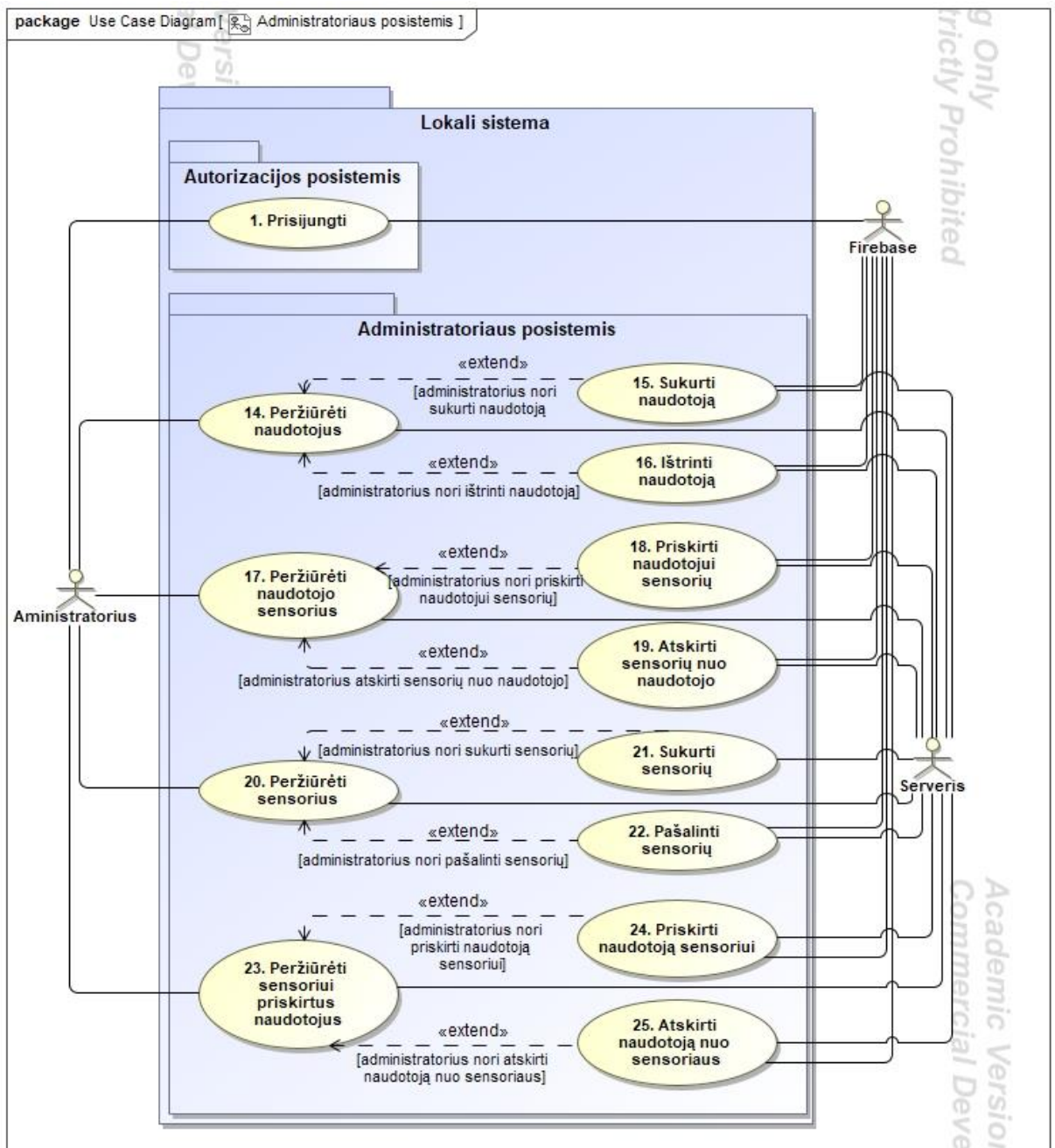
## 2.1.8. Sistemos sudėtis

### 2.1.8.1. Sistemos ribos

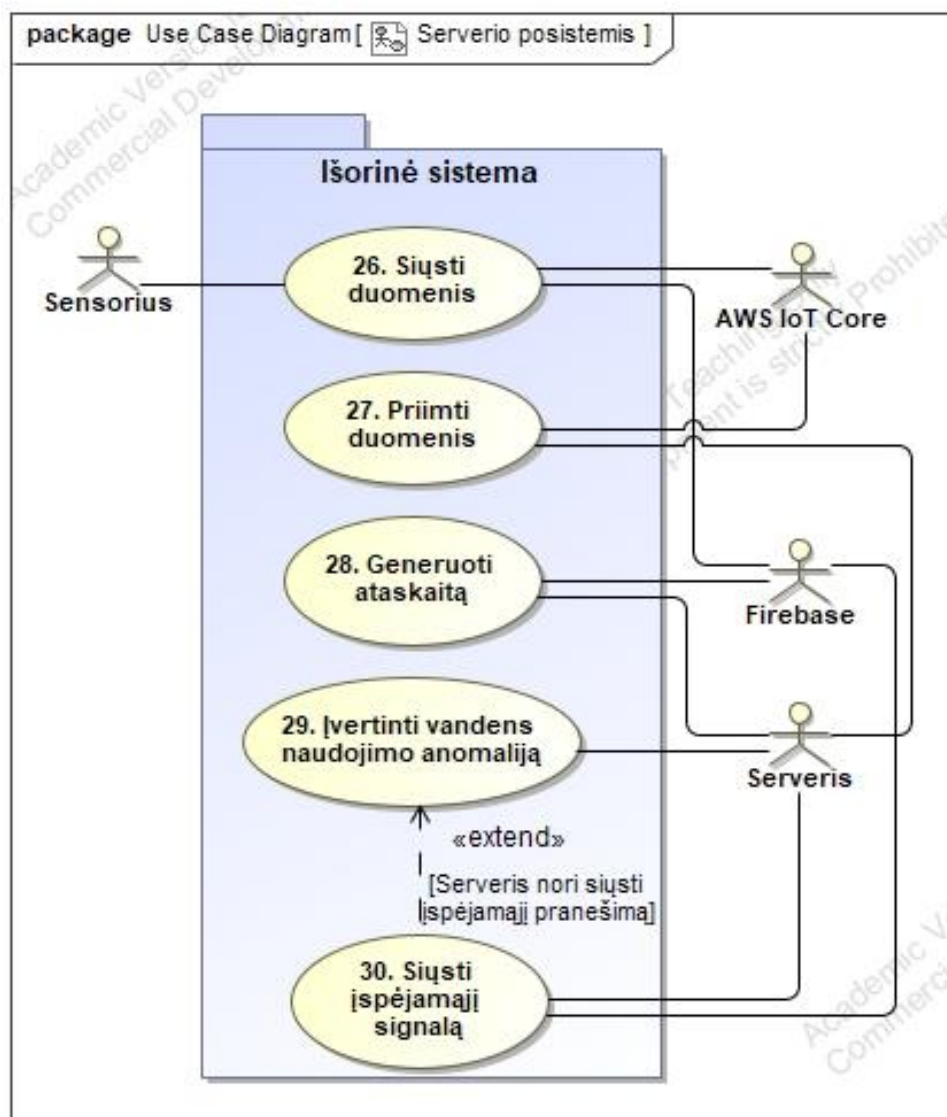
Toliau yra pateikta naudotojo, administratoriaus ir serverio panaudojimo atvejų modeliai atitinkamai **18 pav.**, **19 pav.** ir **20 pav.**



**18 pav.** Naudotojo panaudojimo atvejų modelis



19 pav. Administratoriaus panaudojimo atvejų modelis



20 pav. Serverio panaudojimo atvejų modelis

Naudotojo sistemos pakete yra atvaizduotos naudotojo galimybės programėlėje: autorizuotis į sistemą, peržiūrėti ir filtruoti jutiklius, peržiūrėti ir trinti įspėjimus, peržiūrėti ir trinti ataskaitas ar peržiūrėti detalią vandens sunaudojimo ataskaitą. Taip pat galima peržiūrėti programėlės nustatymus, kuriuose galima keisti spalvų foną, matavimo sistemą (į metrinę ar imperinę), peržiūrėti privatumo politiką, eksportuoti duomenų bazę ir atsijungti.

Administratoriaus sistemos pakete yra atvaizduotos administratoriaus galimybės programėlėje: autorizuotis į sistemą, peržiūrėti, sukurti ar ištrinti naudotoją, peržiūrėti, priskirti ar atskirti jutiklių nuo naudotojo, peržiūrėti, sukurti ar pašalinti jutiklių bei peržiūrėti, priskirti ar atskirti naudotoją nuo jutiklio.

Serverio sistemos pakete yra atvaizduotos funkcijos, kurios bus vykdomos išorinėje sistemoje: jutiklius kas tam tikrą intervalą siųs duomenis, o serveris galės priimti duomenis, saugoti juos duomenų bazėje, įvertinti galimą anomalinį vandens naudojimą, sugeneruoti vandens naudojimo ataskaitą bei siųsti įspėjamąjį pranešimą naudotojui.

Programėle bus galima naudotis tiek su internetu, tiek be interneto ryšio, tačiau norint peržiūrėti pačią naujausią informaciją, gautą iš serverio, reikės būti įjungus interneto ryšį. Taip pat, norint gauti įspėjamuosius pranešimus interneto ryšys privalo būti įjungtas.

### 2.1.8.2. Panaudojimo atvejų sąrašas

Toliau bus pateikti sistemos panaudojimo atvejai.

6 lentelė PA 1 Prisijungti

<b>Tikslas/uždavinys.</b> Prisijungti prie programėlės.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas autorizuoti naudotoją prie programėlės.	
<b>Prieš-sąlyga</b>	Naudotojas nėra prisijungęs prie sistemos ir yra pagrindiniame lange.
<b>Aktorius</b>	Naudotojas ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Naudotojas nori prisijungti.
<b>Veiklos taisyklės</b>	Privaloma pasirinkti „Google“ paskyrą.
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas inicijuoja prisijungimą, sąveikaujant su mygtuku „Prisijungti“.	1.1. Programėlė perkelia naudotoją į prisijungimo langą.
2. Naudotojas įveda el. paštą bei slaptažodį.	
3. Naudotojas inicijuoja prisijungimą, sąveikaujant su mygtuku „Prisijungti“.	3.1. Programėlė kreipiasi į „Firebase“ su prisijungimo duomenimis ir grąžina teigiamą rezultatą. 3.2. Naudotojas yra prijungiamas prie sistemos. 3.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojas autorizuotas prie sistemos.
<b>Alternatyvūs scenarijai</b>	
1. Naudotojas inicijuoja prisijungimą, sąveikaudamas su „Google“ prisijungimo mygtuku.	1.1. Programėlė pateikia galimų „Google“ paskyrų pasirinkimą.
2. Naudotojas pasirenka norimą paskyrą.	2.1. Programėlė kreipiasi į „Firebase“ su prisijungimo duomenimis ir grąžina teigiamą rezultatą. 2.2. Naudotojas yra autorizuojamas prie sistemos ir perkeliama jutiklių sąrašą. 2.3. Baigiamas alternatyvus scenarijus.
3. Naudotojas inicijuoja prisijungimą, sąveikaudamas su „Google“ prisijungimo mygtuku.	3.1. Programėlė pateikia galimų „Google“ paskyrų pasirinkimą.
4. Naudotojas pasirenka norimą paskyrą.	4.1. Programėlė kreipiasi į „Firebase“ su prisijungimo duomenimis ir grąžina neigiamą rezultatą. 4.2. Programėlė naudotojui pateikia informacinį langą apie klaidingus duomenis. 4.3. Baigiamas alternatyvus scenarijus.
5. Naudotojas suveda reikalavimus neatitinkančius arba ne visus duomenis.	5.1. Programėlė naudotojui pateikia informacinį langą apie neteisingai suvestus arba ne visus suvestus duomenis.



	5.2. Baigiamas alternatyvus scenarijus.
-	5.3. Programėlė kreipiasi į „Firebase“ su prisijungimo duomenimis ir grąžina neigiamą rezultatą. 5.4. Naudotojui yra pateikiamas informacinis langas apie neigiamą prisijungimo rezultatą. 5.5. Baigiamas alternatyvus scenarijus.

7 lentelė PA 2 Peržiūrėti jutiklių sąrašą

<b>Tikslas/uždavinys.</b> Peržiūrėti visus naudotojui priklausančius jutiklius.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas leisti naudotojui peržiūrėti visus jam priklausančius jutiklius.	
<b>Prieš-sąlyga</b>	Naudotojas yra prisijungęs prie sistemos.
<b>Aktorius</b>	Naudotojas ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Naudotojas nori peržiūrėti visus jam priklausančius jutiklius.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas tik prisijungė prie sistemos.	1.1. Programėlė kreipiasi į „Firebase“ su naudotojo identifikacijos numeriu ir grąžina jutiklių sąrašą. 1.2. Naudotojui yra atvaizduojami jutikliai. 1.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui atvaizduoti jam priklausančius jutikliai.
<b>Alternatyvūs scenarijai</b>	
1. Naudotojas, būdamas profilio peržiūros lange, inicijuoja grįžimą į pagrindinį langą, sąveikaudamas su atgaliniu mygtuku.	1.1. Programėlė kreipiasi į „Firebase“ su naudotojo identifikacijos numeriu ir grąžina jutiklių sąrašą. 1.2. Naudotojui yra atvaizduojami jutikliai, kur parodoma jutiklio pavadinimas ir esamas vandens sunaudojimas. 1.3. Baigiamas alternatyvus scenarijus.
-	1.1. Programėlė kreipiasi į „Firebase“ su naudotojo identifikacijos numeriu ir grąžina neigiamą rezultatą. 1.2. Programėlė naudotojui atvaizduoja klaidos pranešimą. 1.3. Baigiamas alternatyvus scenarijus.
2. Naudotojas nori filtruoti jutiklius.	2.1. Vykdomas PA 3 Filtruoti jutiklius.

8 lentelė PA 3 Filtruoti jutiklius

<b>Tikslas/uždavinys.</b> Atrinkti tik norimus matyti jutiklius.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas leisti naudotojui atrinkti tik norimus matyti jutiklius pagrindiniame jutiklių lange.	
<b>Prieš-sąlyga</b>	Naudotojas yra prisijungęs prie sistemos.
<b>Aktorius</b>	Naudotojas.

<b>Sužadinimo sąlyga</b>	Naudotojas nori filtruoti jutiklius.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas paspaudžia filtravimo piktogramą.	1.1. Programėlė atvaizduoja naudotojui filtravimo langą.
2. Naudotojas pasirenka norimus filtrus ir paspaudžia filtruoti.	2.1. Programėlė parodo tik filtrą tenkinančius jutiklius. 2.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui atvaizduoti atrinkti jutikliai.
<b>Alternatyvūs scenarijai</b>	
-	-

9 lentelė PA 4 Peržiūrėti jutiklių

<b>Tikslas/uždavinys.</b> Parodyti naudotojui detalią jutiklio informaciją.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas parodyti naudotojui tam tikrą jo pasirinktą jutiklio informaciją.	
<b>Prieš-sąlyga</b>	Naudotojas yra prisijungęs ir yra jutiklių sąrašo lange.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Naudotojas nori peržiūrėti jutiklio informaciją.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas inicijuoja jutiklio peržiūrą sąveikaudamas su tam tikro sąraše esančiu jutikliu.	1.1. Programėlė kreipiasi į Serverį su pasirinkto jutiklio identifikacijos numeriu ir grąžina jutiklio informaciją. 1.2. Naudotojas yra perkeliamas į jutiklio informacijos langą ir jame yra atvaizduojama jutiklio informacija. 1.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui parodyta jutiklio detali informacija.
<b>Alternatyvūs scenarijai</b>	
1. Naudotojas nori keisti jutiklio nustatymus.	1.1. Vykdomas PA 6 Peržiūrėti nustatymus.

10 lentelė PA 5 Keisti jutiklio nustatymus

<b>Tikslas/uždavinys.</b> Pakeisti tam tikro jutiklio įspėjimų gavimo parametrus.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas pakeisti tam tikro naudotojo jutiklio įspėjimų gavimo parametrus.	
<b>Prieš-sąlyga</b>	Naudotojas yra jutiklio informaciniame lange.
<b>Aktorius</b>	Naudotojas ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Naudotojas nori keisti įspėjimų gavimo parametrus.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas inicijuoja įspėjimų koregavimą, sąveikaudamas su viršuje dešinėje esančia redagavimo piktograma.	1.1. Programėlė kreipiasi į „Firebase“ su naudotojo bei jutiklio identifikacijos numeriu bei grąžina pasirinkto jutiklio įspėjimo parametrus.

	1.2. Naudotojas yra perkeliamas į jutiklio įspėjimų redagavimo langą ir jame yra atvaizduojami gauti parametrai.
2. Naudotojas naudodamas tempiamą žymą pasirenka norimą maksimalią ir minimalią reikšmes (temperatūros jutikliui) arba pasirenkama maksimalų leidžiamą vandens tekėjimo laikotarpį (vandens jutikliui).	
3. Naudotojas inicijuoja atnaujinimą, sąveikaudamas su mygtuku „Atnaujinti“.	3.1. Programėlė kreipiasi į „Firebase“ su atnaujintais jutiklio įspėjimo parametrais, juos atnaujina ir grąžina teigiama rezultata. 3.2. Serveris sureagoja ir pakeitimus bei atnaujina lokalius kintamuosius. 3.3. Naudotojui yra atvaizduojamas sėkmės pranešimas. 3.4. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojo jutiklio parametrai atnaujinti.
<b>Alternatyvūs scenarijai</b>	
-	1.1. Abejais atvejais, kai programėlė kreipiasi į „Firebase“ ir ji arba Serveris grąžina neigiamą rezultata, naudotojui yra atvaizduojamas klaidos pranešimas. 1.2. Baigiamas alternatyvus scenarijus.

**11 lentelė** PA 6 Peržiūrėti nustatymus

<b>Tikslas/uždavinys.</b> Peržiūrėti mobiliosios programėlės nustatymus.	
<b>Aprašymas.</b> Leisti naudotojui pakeisti mobiliosios programėlės nustatymus.	
<b>Prieš-sąlyga</b>	Naudotojas yra jutiklio informaciniame lange.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Naudotojas nori peržiūrėti nustatymus.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas paspaudžia nustatymų piktogramą.	1.1. Naudotojui atvaizduojamas nustatymų langas. 1.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui atvaizduotas nustatymų langas.
<b>Alternatyvūs scenarijai</b>	
1. Naudotojas nori pakeisti nustatymus.	1.1. Vykdomas PA 7 Keisti nustatymus.

**12 lentelė** PA 7 Keisti nustatymus

<b>Tikslas/uždavinys.</b> Pakeisti tam tikrus programėlės nustatymus.	
<b>Aprašymas.</b> Leisti naudotojui atlikti tam tikrus programėlės nustatymus.	
<b>Prieš-sąlyga</b>	Naudotojas yra nustatymų lange.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Naudotojas nori pakeisti nustatymą.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	

Naudotojo veiksmai		Sistemos reakcija
1. Naudotojas paspaudžia duomenų bazės eksportavimo pasirinkimą.		1.1. Naudotojas yra atvaizduojamas langas, kuriame galima pakeisti duomenų bazės eksportavimo failą vietą bei pavadinimą.
2. Naudotojas nurodo norimą failo pavadinimą ir išsaugojimo vietą.		2.1. Programėlė išsaugo duomenis į nurodytą failą. 2.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Tam tikras programėlės nustatymas pakeistas.	
<b>Alternatyvūs scenarijai</b>		
1. Naudotojas pasirenka programėlės temų keitimo pasirinkimą. 2. Naudotojas pasirenka norimą programėlės temą.		1.1 Programėlė atvaizduoja galimas pasirinkti temas. 2.1 Programėlė atnaujina temą į naudotojo pasirinktą. 2.2 Baigiamas alternatyvus scenarijus.
1. Naudotojas pasirenka programėlės matavimo vienetų keitimo pasirinkimą. 2. Naudotojas pasirenka norimą matavimo vienetą.		1.1 Programėlė atvaizduoja galimus pasirinkti matavimo vienetus. 2.1 Programėlė atnaujina visos programos matavimo vienetų sistemą. 2.2 Baigiamas alternatyvus scenarijus.
1. Naudotojas pasirenka privatumo politikos arba terminų ir sąlygų pasirinkimą.		1.1 Programėlė nukreipia atitinkamai į privatumo politikos arba terminų ir sąlygų internetinio puslapio langą. 1.2 Baigiamas alternatyvus scenarijus.
1. Naudotojas paspaudžia atsijungti.		1.1 Programėlė atsijungia nuo sistemos.

### 13 lentelė PA 8 Peržiūrėti įspėjimų sąrašą

<b>Tikslas/uždavinys.</b> Peržiūrėti visus jutiklių įspėjimus.		
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas naudotojui peržiūrėti visus jutiklių įspėjimus.		
<b>Prieš-sąlyga</b>	Naudotojas yra prisijungęs ir yra pagrindiniame lange.	
<b>Aktorius</b>	Naudotojas.	
<b>Sužadavimo sąlyga</b>	Naudotojas nori peržiūrėti jutiklių įspėjimus.	
<b>Veiklos taisyklės</b>	-	
<b>Pagrindinis scenarijus</b>		
Naudotojo veiksmai		Sistemos reakcija
1. Naudotojas inicijuoja įspėjimų peržiūrėjimą, sąveikaudamas su viršuje dešinėje esančia įspėjimų piktograma.		1.1. Programėlė kreipiasi į lokalią duomenų bazę ir išgauna jutiklių įspėjimus. 1.2. Naudotojas yra perkeliamas į jutiklių įspėjimų langą ir jame yra atvaizduojami jutiklių įspėjimai (jutiklio pavadinimas, laikas, įspėjamasis pranešimas). 1.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui atvaizduoti jutiklių įspėjimai.	
<b>Alternatyvūs scenarijai</b>		
1. Naudotojas inicijuoja įspėjimų peržiūrėjimą, sąveikaudamas su viršuje dešinėje esančia įspėjimų piktograma.		1.1. Programėlė kreipiasi į lokalią duomenų bazę įspėjimų gavimui ir grąžina neigiamą rezultatą. 1.2. Naudotojui yra atvaizduojamas klaidos pranešimas.

1.3. Baigiamas alternatyvus scenarijus.

14 lentelė PA 9 Trinti įspėjimus

<b>Tikslas/uždavinys.</b> Ištrinti vieną ar visus gautus aliarminius pranešimus.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas naudotojui ištrinti jam nereikalingus aliarminius pranešimus.	
<b>Prieš-sąlyga</b>	Naudotojas yra prisijungęs ir yra pagrindiniame lange.
<b>Aktorius</b>	Naudotojas.
<b>Sužadinimo sąlyga</b>	Naudotojas nori ištrinti aliarminius pranešimus.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. Naudotojas nuspaudžia įspėjimą.	1.1. Naudotojui yra atvaizduojamas patvirtinimo langas.
2. Naudotojas patvirtina pasirinkimą paspausdamas trynimo mygtuką.	2.1. Programėlė pašalina įspėjimą iš programos. 2.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojo profilio informacija atnaujinta.
<b>Alternatyvūs scenarijai</b>	
1. Naudotojas paspaudžia viršuje esančią trynimo piktogramą.	1.1. Naudotojui yra atvaizduojamas patvirtinimo langas.
2. Naudotojas patvirtina pasirinkimą paspausdamas trynimo mygtuką.	2.1. Programėlė pašalina įspėjimus iš programos. 2.2. Baigiamas alternatyvus scenarijus

15 lentelė PA 13 Gauti aliarminius pranešimus

<b>Tikslas/uždavinys.</b> Naudotojui gauti įspėjamąjį pranešimą.	
<b>Aprašymas.</b> Šis panaudojimo atvejis leisti naudotojui gauti įspėjamąjį pranešimą apie jutiklio kritinius pokyčius.	
<b>Prieš-sąlyga</b>	Naudotojo programėlė veikia fone.
<b>Aktorius</b>	Naudotojas ir „Firebase“.
<b>Sužadinimo sąlyga</b>	„Firebase“ nori išsiųsti kritinę reikšmę.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Naudotojo veiksmai</b>	<b>Sistemos reakcija</b>
1. -	1.1. „Firebase“ gauna kritinę reikšmę. 1.2. „Firebase“ išsiunčia naudotojui įspėjamąjį pranešimą.
2. Naudotojo telefone parodomas įspėjamasis pranešimas.	2.1. Iš „Cloud Messaging Service“ eilės yra pašalinimas pranešimas. 2.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Naudotojui atvaizduotas įspėjamasis pranešimas.
<b>Alternatyvūs scenarijai</b>	
1. -	1.1. „Cloud Messaging Service“ neveikia, todėl „Firebase“ negali išsiųsti pranešimo 1.2. Sistemos pranešimų žurnale užfiksuojama kritinė klaida. 1.3. Baigiamas alternatyvus scenarijus.

2. -	2.1. „Firebase“ gauna kritinę reikšmę ir bando siųsti naudotojui. 2.2. Naudotojo telefonas neturi interneto ryšio, todėl žinutė laikoma saugykloje iki kol interneto ryšys atsiras ir vėl bus bandoma siųsti. 2.3. Baigiamas alternatyvus scenarijus.
------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

16 lentelė PA 14 Peržiūrėti naudotojus

<b>Tikslas/uždavinys.</b> Administratoriui peržiūrėti visus sistemoje esančius naudotojus.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas administratoriui peržiūrėti sistema besinaudojančius naudotojus.	
<b>Prieš-sąlyga</b>	Administratorius prisijungia.
<b>Aktorius</b>	Administratorius ir Serveris.
<b>Sužadinimo sąlyga</b>	Administratorius nori peržiūrėti naudotojus.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Administratoriaus veiksmai</b>	<b>Sistemos reakcija</b>
1. Administratorius prisijungia prie sistemos.	1.1. Programėlė kreipiasi į serverį išgauti naudotojus ir atvaizduoja juos administratoriaus pagrindiniame naudotojų lange. 1.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Administratoriui atvaizduoti visi sistemos naudotojai.
<b>Alternatyvūs scenarijai</b>	
-	1.1. Serveris kreipiasi į „Firebase“, kuris grąžina neigiamą rezultatą. 1.2. Programėlė administratoriui atvaizduoja klaidos pranešimą. 1.3. Baigiamas alternatyvus scenarijus.
2. Administratorius nori sukurti naudotoją.	2.1. Vykdomas PA 15 Sukurti naudotoją.
3. Administratorius nori ištrinti naudotoją.	3.1. Vykdomas PA 16 Ištrinti naudotoją.

17 lentelė PA 15 Sukurti naudotoją

<b>Tikslas/uždavinys.</b> Administratoriui sukurti naudotoją.	
<b>Aprašymas.</b> Šis panaudojimo atvejis yra skirtas sukurti sistemos naudotoją.	
<b>Prieš-sąlyga</b>	Administratorius yra prisijungęs ir yra naudotojų lange.
<b>Aktorius</b>	Administratorius, Serveris ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Administratorius nori sukurti naudotoją.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Administratoriaus veiksmai</b>	<b>Sistemos reakcija</b>
1. Administratorius paspaudžia pridėjimo piktogramą.	1.1. Programėlė atvaizduoja naudotojo pridėjimo lango vaizdą.
2. Administratorius užpildo duomenis ir paspaudžia sukūrimo mygtuką.	2.1. Programėlė kreipiasi į Serverį su duomenimis. 2.2. Serveris kreipiasi į „Firebase“ ir sukuria naudotoją. 2.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Sukuriamas naujas naudotojas.

<b>Alternatyvūs scenarijai</b>	
1. Administratorius užpildo duomenis ir paspaudžia sukūrimo mygtuką.	1.1. Serveris kreipiasi į „Firebase“, kuris grąžina neigiamą rezultatą dėl nenumatytų klaidų arba dėl to, kad toks naudotojas jau egzistuoja. 1.2. Programėlė administratoriui atvaizduoja klaidos pranešimą. 1.3. Baigiamas alternatyvus scenarijus.
2. Administratorius užpildo ne visus duomenis ir paspaudžia sukūrimo mygtuką.	2.1. Programėlė administratoriui atvaizduoja klaidos pranešimą. 2.2. Baigiamas alternatyvus scenarijus.

**18 lentelė** PA 16 Ištrinti naudotoją

<b>Tikslas/uždavinys.</b> Administratoriui ištrinti naudotoją.	
<b>Aprašymas.</b> Šis panaudojimo atvejis yra skirtas administratoriui pašalinti naudotoją iš sistemos.	
<b>Prieš-sąlyga</b>	Administratorius yra prisijungęs ir yra naudotojų lange.
<b>Aktorius</b>	Administratorius, Serveris ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Administratorius nori ištrinti naudotoją.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Administratoriaus veiksmai</b>	<b>Sistemos reakcija</b>
1. Administratorius nuspaudžia naudotoją.	1.1. Programėlė atvaizduoja naudotojo ištrynimo patvirtinimo langą.
2. Administratorius patvirtina paspausdamas patvirtinimo mygtuką.	2.4. Programėlė kreipiasi į Serverį su ištrynimo patvirtinimu. 2.5. Serveris kreipiasi į „Firebase“ ir ištrina naudotoją. 2.1. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Ištrinamas naudotojas iš sistemos.
<b>Alternatyvūs scenarijai</b>	
1. Administratorius patvirtina ištrynimo langą paspausdamas patvirtinimo mygtuką.	1.1. Serveris kreipiasi į „Firebase“, kuris grąžina neigiamą rezultatą dėl nenumatytų klaidų. 1.2. Programėlė administratoriui atvaizduoja klaidos pranešimą. 1.3. Baigiamas alternatyvus scenarijus.

**19 lentelė** PA 26 Siųsti duomenis

<b>Tikslas/uždavinys.</b> Išsiųsti esamus jutiklio duomenis.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas išsiųsti esamus jutiklio nuskaitytus duomenis ir išsaugoti juos duomenų bazėje.	
<b>Prieš-sąlyga</b>	Jutiklius yra autentifikuotas ir veikia.
<b>Aktorius</b>	Jutiklius, „Firebase“ ir „AWS IoT Core“.
<b>Sužadinimo sąlyga</b>	Jutiklius nori kas tam tikrą laiką siųsti duomenis.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Jutiklio veiksmai</b>	<b>Sistemos reakcija</b>

1. Mikrovaldiklis kas tam tikrą laiką nuskaityti duomenis iš daviklio ir sukuria siuntimo užklausą į duomenų perdavimo brokerį.	1.1. Brokeris priima užklausą ir patalpina rezultata duomenų eilėje. 1.2. Baigiamas panaudos atvejis.
<b>Po-sąlyga</b>	Duomenys patalpinti duomenų brokerio eilėje.
<b>Alternatyvūs scenarijai</b>	
1. Jutiklius negali nuskaityti duomenų iš daviklio dėl tam tikrų priežasčių. 2. Jutiklius sumirksi geltona spalva du kartus. 3. Baigiamas alternatyvus scenarijus.	
1. Jutiklius nuskaityti duomenis iš daviklio ir daro siuntimo užklausą į sistemą.	1.1. Sistema priima užklausą, tačiau negali išsaugoti duomenų ir grąžina neigiamą rezultatą.
2. Jutiklius sumirksi raudona spalva du kartus. 3. Baigiamas alternatyvus scenarijus.	-

20 lentelė PA 27 Priimti duomenis

<b>Tikslas/uždavinys.</b> Priimti duomenis iš jutiklio.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas nuskaityti duomenis iš brokerio duomenų eilės ir išsaugoti juos duomenų bazėje.	
<b>Prieš-sąlyga</b>	Serveris yra užprenumeravęs visus duomenų brokerio galutinius taškus.
<b>Aktorius</b>	Serveris ir „AWS IoT Core“.
<b>Sužadinimo sąlyga</b>	Duomenų brokeris patalpina duomenis eilėje.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Serverio veiksmi</b>	
1. Serverio užprenumeruotas taškas yra pažadinamas. 2. Serveris išsaugo duomenis duomenų bazėje. 3. Baigiamas panaudojimo atvejis.	
<b>Po-sąlyga</b>	Duomenys priimti ir išsaugoti duomenų bazėje.
<b>Alternatyvūs scenarijai</b>	
1. Nutrūksta brokerio taškų prenumerata. 2. Serveris bando užprenumeruoti visus taškus. 3. Serveris užprenumeruoja visus taškus. 4. Baigiamas alternatyvus scenarijus.	
5. Nutrūksta prisijungimas prie duomenų bazės. 6. Serveris bando prisijungti prie duomenų bazės. 7. Serveris prisijungia prie duomenų bazės. 8. Baigiamas alternatyvus scenarijus.	

21 lentelė PA 28 Generuoti ataskaitą

<b>Tikslas/uždavinys.</b> Sugeneruoti vandens vartojimo ataskaitą.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas sugeneruoti vandens naudojimo visų ypatybių ataskaitą ir perduoti naudotojui.	
<b>Prieš-sąlyga</b>	Serveris yra veiklus.
<b>Aktorius</b>	Serveris ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Laikas pasisuka ties 7 valanda ir 0 minučių.



<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Serverio veiksmi</b>	<b>„Firebase“ reakcija</b>
1. Serveris išgauna visus pagal kontekstą skirtus dienų duomenis ir sugeneruoja ataskaitą. 2. Serveris išsiunčia duomenis į „Firebase“.	2.1. „Firebase“ priima duomenis, gautus iš Serverio, ir perduoda juos naudotojui pranešimu. 2.2. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	Vandens vartojimo ataskaita išsiųsta naudotojui.
<b>Alternatyvūs scenarijai</b>	
-	1. „Firebase“ negali sėkmingai priimti duomenų. 2. Serveryje fiksuojama klaida. 3. Baigiamas alternatyvus scenarijus.

22 lentelė PA 29 Įvertinti vandens naudojimo anomaliją

<b>Tikslas/uždavinys.</b> Sukurti mašininio mokymo modelį anomaliam vandens naudojimui nustatyti.	
<b>Aprašymas.</b> Šis panaudojimo atvejis yra skirtas kas tam tikrą laiką įvykdyti sukurtą mašininio mokymo algoritmą anomaliam vandens naudojimui nustatyti ir įspėti naudotoją.	
<b>Prieš-sąlyga</b>	Serveris yra veikiantis.
<b>Aktorius</b>	Serveris.
<b>Sužadinimo sąlyga</b>	Serveris nori kas tam tikrą laiką analizuoti duomenis.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Serverio veiksmi</b>	<b>„Firebase“ reakcija</b>
1. Serveris kas minutę inicijuoja duomenų analizę mašininio mokymo modeliui sukurti. 2. Serveris įvertina gautus mašininio mokymo rezultatus. 3. Serverio įvertintas rezultatas yra teigiamas. 4. Baigiamas panaudojimo atvejis.	-
<b>Po-sąlyga</b>	Naudotojo mašininio mokymo modelis yra sukurtas ir įvykdytas.
<b>Alternatyvūs scenarijai</b>	
1. Serveris įvertina gautus mašininio mokymo rezultatus. 2. Serverio įvertintas rezultatas yra neigiamas. 3. Serveris kreipiasi į „Cloud Messaging Service“ aliarminiam pranešimui išsiųsti.	3.1. Cloud Messaging Service“ išsiunčia žinutę nurodytam naudotojui. 3.2. Naudotojo įrenginys išsaugo duomenis lokaliaje duomenų bazėje. 3.3. Baigiamas alternatyvus scenarijus.

23 lentelė PA 30 Siųsti aliarminį pranešimą

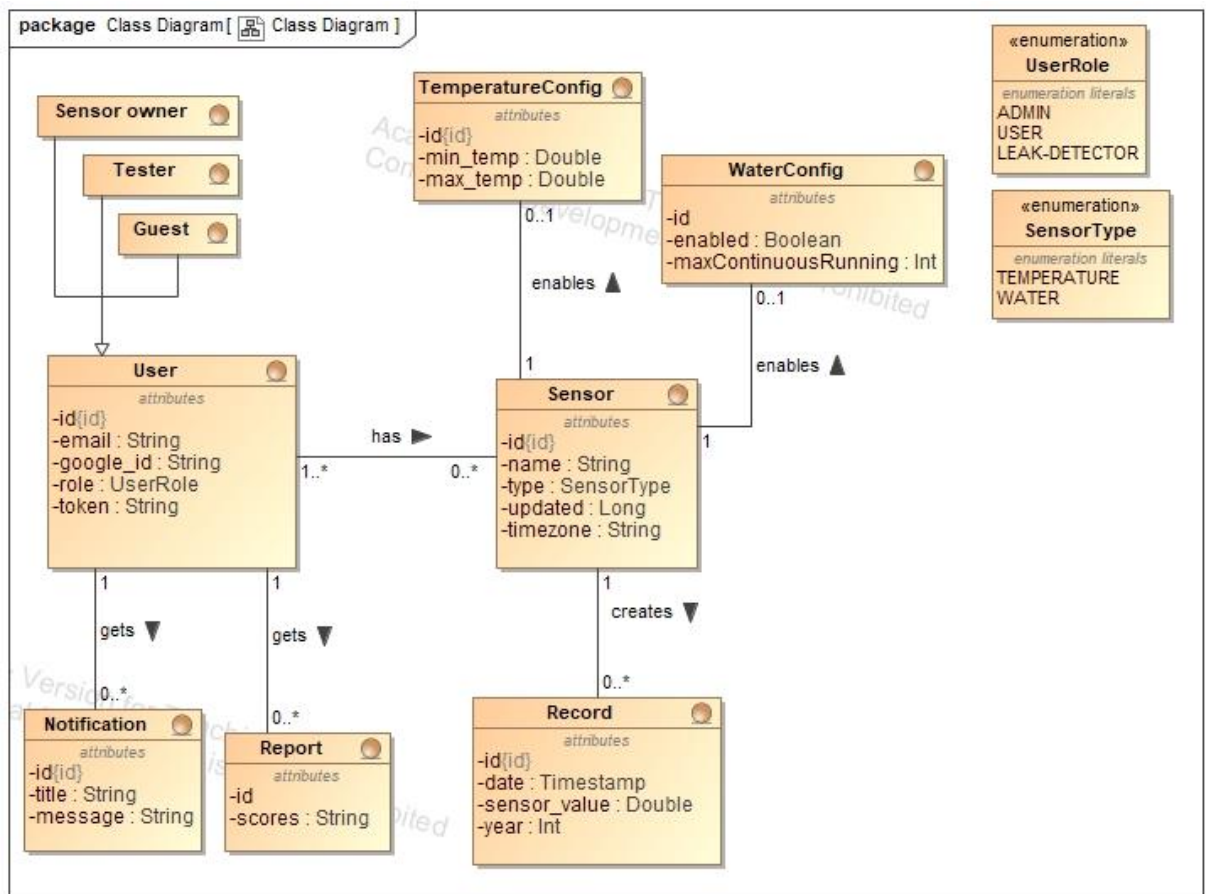
<b>Tikslas/uždavinys.</b> Išsiųsti naudotojui aliarminį pranešimą apie kritinę reikšmę.	
<b>Aprašymas.</b> Šis panaudojimo atvejis skirtas išsiųsti aliarminį pranešimą naudotojui apie kritinę reikšmę ir išsaugoti lokaliaje duomenų bazėje.	
<b>Prieš-sąlyga</b>	Serveris veikia ir turi išsaugotus naudotojų aliarminių pranešimų raktus.
<b>Aktorius</b>	Serveris ir „Firebase“.
<b>Sužadinimo sąlyga</b>	Serveris gauna kritinę reikšmę.
<b>Veiklos taisyklės</b>	-
<b>Pagrindinis scenarijus</b>	
<b>Serverio veiksmi</b>	<b>Sistemos reakcija</b>

1. Serveris gauna reikšmę iš jutiklio arba Serverio ir nustato, jog tai yra kritinė reikšmė. 2. Serveris kreipiasi į „Cloud Messaging Service“ aliarmui pranešimui išsiųsti.	2.1. „Cloud Messaging Service“ išsiunčia žinutę nurodytam naudotojui“. 2.2. Naudotojo įrenginys išsaugo duomenis lokaliaje duomenų bazėje. 2.3. Baigiamas panaudojimo atvejis.
<b>Po-sąlyga</b>	
<b>Alternatyvūs scenarijai</b>	
-	1.1. „Cloud Messaging Service“ neveikia, todėl „Firebase“ negali išsiųsti pranešimo 1.2. Sistemos pranešimų žurnale užfiksuojama klaida. 1.3. Baigiamas alternatyvus scenarijus.
-	1.4. „Firebase“ gauna kritinę reikšmę ir bando siųsti naudotojui. 1.5. Naudotojo telefonas neturi interneto ryšio, todėl žinutė laikoma saugykloje iki kol interneto ryšys atsiras ir vėl bus bandoma siųsti. 1.6. Baigiamas alternatyvus scenarijus.

## 2.1.9. Funkciniai reikalavimai ir reikalavimai duomenims

### 2.1.9.1. Reikalavimai duomenims

21 pav. yra pateikta sistemos esybių ryšių diagrama.



21 pav. Sistemos esybių ryšių diagrama

Toliau bus pateiktas duomenų žodynas (duomenų modelio specifikacija):

24 lentelė Duomenų žodynas

Esybė Naudotojas			
Atributo pav.	Aprašas	Galimos reikšmės	Ar privalomas?
id	Unikalus identifikacijos kodas.	Unikali raidžių, skaičių ir simbolių „-“ bei „_“ kombinacija, kur simbolis negali būti pradžioje, ilgis – 36 simboliai.	Taip
email	Programos naudotojo esamas bei naudojamas elektroninis paštas.	Unikalus tekstas, atitinkantis teisingus el. pašto reikalavimus.	Taip
google_id	Programos naudotojo „Google“ identifikacijos kodas, jei registruojamasi naudojant „Google“.	Unikali raidžių, skaičių ir simbolių „-“ bei „_“ kombinacija, kurios ilgis ne mažesnis nei 25, tačiau ne didesnis nei 100 simbolių.	Taip
role	Programos naudotojo rolė.	Enumeratoriaus reikšmė, žyminti tekstą, ne didesnę nei 20 simbolių.	Taip

token	Programos naudotojo prisijungimo žetonas.	Unikali raidžių, skaičių ir simbolių „-“ bei „_“ kombinacija, kurios ilgis 152 simboliai.	Taip
<b>Esybė Jutiklius</b>			
Atributo pav.	Aprašas	Galimos reikšmės	Ar privalomas?
id	Unikalus identifikacijos kodas.	Unikali raidžių, skaičių ir simbolių „-“ bei „_“ kombinacija, kur simbolis negali būti pradžioje, ilgis – 32 simboliai.	Taip
name	Jutiklio nustatytas žmogui suprantamas pavadinimas.	Tekstas, kuris turi būti netrumpesnis nei 1 raidė bei ne ilgesnis nei 20.	Taip
type	Jutiklio tipas.	Enumeratoriaus reikšmė, žyminti tekstą, ne didesnį nei 20 simbolių.	Taip
updated	Jutiklio paskutinis duomenų atnaujinimas.	Skaičius, kuris negali būti didesnis nei 20 simbolių.	Ne
timezone	Jutiklio laiko zona.	Tekstas, kurio ilgis gali būti ne didesnis nei 100 simbolių.	Taip
<b>Esybė Įrašas</b>			
Atributo pav.	Aprašas	Galimos reikšmės	Ar privalomas?
id	Unikalus identifikacijos kodas.	Unikalus teigiamas skaičius, kurio ilgis ne didesnis nei 19 skaitmenų.	Taip
date	Įrašo sukūrimo laikas.	Skaičius, kuris negali būti didesnis nei 20 simbolių.	Taip
sensor_value	Jutiklio reikšmė.	Skaičius, kuris negali būti didesnis nei 20 simbolių.	Taip
year	Įrašo sukūrimo metai.		Taip
<b>Esybė Pranešimas</b>			
Atributo pav.	Aprašas	Galimos reikšmės	Ar privalomas?
id	Unikalus identifikacijos kodas.	Unikalus teigiamas skaičius, kurio ilgis ne didesnis nei 19 skaitmenų.	Taip
title	Įspėjamojo pranešimo titulinis pavadinimas.	Tekstas, kuris turi būti netrumpesnis nei 2 raidės bei ne ilgesnis nei 20.	Taip
message	Įspėjamojo pranešimo žinutė.	Tekstas, kuris turi būti netrumpesnis nei 2 raidės bei ne ilgesnis nei 32.	Taip

<b>Esybė Ataskaita</b>			
<b>Atributo pav.</b>	<b>Atributo pav.</b>	<b>Galimos reikšmės</b>	<b>Ar privalomas?</b>
id	Unikalus identifikacijos kodas.	Unikalus teigiamas skaičius, kurio ilgis ne didesnis nei 19 skaitmenų.	Taip
scores	Vandens suvartojimo įvertinimo reikšmės.	Tekstas, kurio ilgis gali būti ne didesnis nei 100 simbolių.	Taip
<b>Esybė Temperatūros Nustatymas</b>			
<b>Atributo pav.</b>	<b>Aprašas</b>	<b>Galimos reikšmės</b>	<b>Ar privalomas?</b>
id	Unikalus identifikacijos kodas.	Unikalus teigiamas skaičius, kurio ilgis ne didesnis nei 19 skaitmenų.	Taip
max_temp	Maksimaliai leidžiama temperatūra prieš įspėjant naudotoją.	Sveikas skaičius, ne mažesnis nei -50, tačiau ne didesnis nei 150.	Taip
min_temp	Minimaliai leidžiama temperatūra prieš įspėjant naudotoją.	Sveikas skaičius, ne mažesnis nei -50, tačiau ne didesnis nei 150.	Taip
<b>Esybė Vandens Nustatymas</b>			
<b>Atributo pav.</b>	<b>Aprašas</b>	<b>Galimos reikšmės</b>	<b>Ar privalomas?</b>
id	Unikalus identifikacijos kodas.	Unikalus teigiamas skaičius, kurio ilgis ne didesnis nei 19 skaitmenų.	Taip
enabled	Ar parametras yra įjungtas.	„True“ arba „False“ reikšmės.	Taip
maxContinuousRunning	Didžiausia vandens tekėjimo leidimo reikšmė.	Sveikas skaičius, ne didesnis nei 100.	Taip

## 2.1.10. Reikalavimai sistemos išvaizdai

25 lentelė Išvaizdos reikalavimas 1

<b>Reikalavimas #:</b>	IR-1	<b>Reikalavimo tipas:</b>	V10a	<b>PA/FR #:</b>	3, 4
<b>Aprašymas:</b>	Sistemos logotipas turi būti atvaizduotas visų naudotojų programėlių sąrašė.				
<b>Pagrindimas:</b>	Sistemos logotipas užtikrins, kad sistemos naudotojai žino kokią programėlę naudoja bei leis dar labiau įsiminti šį logotipą, taip užtikrinant įmonės patikimumą.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu bus teisingai nustatytas sistemos logotipas, tenkinantis tuos pačius dydžio bei formos reikalavimus.				
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b>	4		
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b>	Nėra		

**Istorija:** Užregistruotas 2022 vasario 20 d.

## 26 lentelė Stiliaus reikalavimas 1

<b>Reikalavimas #:</b>	SR-1	<b>Reikalavimo tipas:</b>	V10b	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Programėlė turėtų pasižymėti paprastomis spalvomis, nebūti per daug spalvinga.				
<b>Pagrindimas:</b>	Programėlė turėtų atitikti tam tikrus profesionalumo bei rimtumo atitikmenis, tuo pačiu metu neturėtų būti nuobodi žvelgiant iš naudotojo pusės.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Bus atlikta naudotojų apklausa, kurios metu ne mažiau nei 70 procentų apklaustųjų pasakys, kad programėlė nepasižymi per daug spalvotu dizainu ir atitinka naudotojo lūkesčius.				
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b>	4		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 20 d.				

## 2.1.11. Reikalavimai panaudojamumui

### 27 lentelė Naudojimosi paprastumo reikalavimas 1

<b>Reikalavimas #:</b>	NP-1	<b>Reikalavimo tipas:</b>	V11a	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Programėlės naršymo keliai turi būti lengvi bei suprantami.				
<b>Pagrindimas:</b>	Naudotojai norėtų naršyti per programą lengvai bei efektyviai.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Bus atlikta naudotojų apklausa, kurios metu ne mažiau nei 80 procentų apklaustųjų pasakys, kad naršymo keliai yra lengvi bei suprantami.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

### 28 lentelė Naudojimosi paprastumo reikalavimas 2

<b>Reikalavimas #:</b>	NP-2	<b>Reikalavimo tipas:</b>	V11a	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Programėlė turi leisti naudotojui išvengti klaidų.				
<b>Pagrindimas:</b>	Naudotojai nenorėtų padaryti nenumatytų klaidų, taip panaikinant ar sugadinant jų esamą informaciją.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu įvedimo laukai turi atitikti keliamus reikalavimus, naudotojui turi būti pateiktas patvirtinimo langas veiksmui patvirtinti.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

### 29 lentelė Naudojimo paprastumo reikalavimas 3

<b>Reikalavimas #:</b>	NP-3	<b>Reikalavimo tipas:</b>	V11a	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Programėle naudotis turi būti galimybė be jokių apmokymų.				

<b>Pagrindimas:</b>	Naudotojai norėtų patys pradėti naudotis programėle be jokių apmokymų taip sutaupant laiko ir pastangų.		
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.		
<b>Atitikimo kriterijus:</b>	Bus atlikta naudotojų apklausa, kurios metu ne mažiau nei 75 procentai apklaustųjų pasakys, kad naudotis programėle nereikėjo jokių apmokymų ir patys sugebėjo ją pasinaudoti.		
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b>	,
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b>	Nėra
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.		

### 30 lentelė Personalizavimo reikalavimas 1

<b>Reikalavimas #:</b>	PKK-2	<b>Reikalavimo tipas:</b>	V11b	<b>PA/FR #:</b>	1-9
<b>Aprašymas:</b>	Programėlės sąsaja turi būti anglų kalba.				
<b>Pagrindimas:</b>	Anglų kalba yra viena iš populiariausių kalbų visame pasaulyje, todėl dauguma naudotojų supras, kas parašyta programėlėje.				
<b>Šaltinis:</b>	Interneto puslapis				
<b>Atitikimo kriterijus:</b>	Testavimo metu visi teksto žodžiai atitiks anglišką atitikmenį anglų žodyne.				
<b>Užsakovo pasitenkinimas:</b>	3	<b>Užsakovo nepasitenkinimas:</b>	2		
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 20 d.				

### 31 lentelė Personalizavimo reikalavimas 2

<b>Reikalavimas #:</b>	PKK-2	<b>Reikalavimo tipas:</b>	V11b	<b>PA/FR #:</b>	1-9
<b>Aprašymas:</b>	Programėlė turi prisitaikyti prie dienos ir nakties režimo dizainų.				
<b>Pagrindimas:</b>	Dauguma naudotojų norėtų skirtingų režimu metu matyti skirtingą dizainą, taip saugojant akis.				
<b>Šaltinis:</b>	Internetinės svetainės apklausa				
<b>Atitikimo kriterijus:</b>	Testavimo metu skirtingi dizainai atspindės skirtingas spalvas bei piktogramas, nurodytas reikalavimuose, priklausomai nuo pasirinkto laiko režimo.				
<b>Užsakovo pasitenkinimas:</b>	3	<b>Užsakovo nepasitenkinimas:</b>	1		
<b>Prioritetas:</b>	Žemas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 20 d.				

### 32 lentelė Prieinamumo reikalavimas 1

<b>Reikalavimas #:</b>	PN-1	<b>Reikalavimo tipas:</b>	V11e	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Programėlė turi būti pritaikyta naudotis neįgaliesiems.				
<b>Pagrindimas:</b>	Turi būti suteiktos visos teisės bei paslaugos neįgaliesiems naudotis šia programėle be jokių sunkumų.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu bus įvertinti šie aspektai: <ul style="list-style-type: none"> <li>• programėlė spalvos turi atitikti skirtingą spalvų spektrą, nurodytą reikalavimuose, žmonėms, kurie neskiria spalvų;</li> <li>• tekstas turi atitikti tam tikrą kontrasto lygį: jeigu tekstas mažesnis nei 18 pt arba pajuodintas ir mažesnis nei 14 pt, spalvų kontrastas turi būti bent 4.5:1, kitiems dydžiams - bent 3.0:1;</li> </ul>				

		• kiekvienas naudotojo sąsajos elementas turi būti unikalus
<b>Užsakovo pasitenkinimas:</b>	3	<b>Užsakovo nepasitenkinimas:</b> 2
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b> Nėra
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.	

## 2.1.12. Reikalavimai vykdymo charakteristikoms

33 lentelė Užduočių vykdymo reikalavimas 1

<b>Reikalavimas #:</b>	RUVG-1	<b>Reikalavimo tipas:</b> V12a	<b>PA/FR #:</b> 8, 10
<b>Aprašymas:</b>	Įspėjamasis pranešimas naudotojui turi ateiti greičiau nei per 5 sekundes.		
<b>Pagrindimas:</b>	Naudotojui yra svarbu gauti įspėjamąjį pranešimą apie per aukštą temperatūrą greitai.		
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.		
<b>Atitikimo kriterijus:</b>	Testavimo metu bus išsiųstas pranešimas, o įspėjamasis signalas ateis greičiau nei per 5 sekundes.		
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b> 5	
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b> Nėra	
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.		

34 lentelė Tikslumo reikalavimas 1

<b>Reikalavimas #:</b>	RT-1	<b>Reikalavimo tipas:</b> V12c	<b>PA/FR #:</b> 4, 5, 6
<b>Aprašymas:</b>	Naudotojui duomenys turi būti apvalinami iki vieną skaičių po kablelio tikslumu.		
<b>Pagrindimas:</b>	Naudotojui yra svarbu matyti detalią informaciją.		
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.		
<b>Atitikimo kriterijus:</b>	Testavimo metu temperatūros atvaizdavimo tekstas po kablelio turės vieną skaičių.		
<b>Užsakovo pasitenkinimas:</b>	3	<b>Užsakovo nepasitenkinimas:</b> 3	
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b> Nėra	
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.		

## 2.1.13. Reikalavimai veikimo sąlygoms

35 lentelė Veikimo sąlygos reikalavimas 1

<b>Reikalavimas #:</b>	VSR-1	<b>Reikalavimo tipas:</b> V13	<b>PA/FR #:</b> 9
<b>Aprašymas:</b>	Jutiklius nesiųsdamas duomenų turėtų „užmigti“ kietuoju miegu, taip taupant elektrą ar bateriją.		
<b>Pagrindimas:</b>	Naudotojams yra svarbu, kad jutiklius nenaudotų daugiau elektros negu būtų galima sutaupyti.		
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.		
<b>Atitikimo kriterijus:</b>	Testavimo metu bus atliktas patikrinimas, ar jutiklius, nesiųsdamas duomenų, nenaudoja tiek pat elektros kiek siųsdamas.		
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b> 3	
<b>Prioritetas:</b>	Vidutinis	<b>Konfliktai:</b> Nėra	
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.		



## 2.1.14. Reikalavimai sistemos priežiūrai

Reikalavimų sistemos priežiūrai nėra numatyta.

## 2.1.15. Reikalavimai saugumui

36 lentelė Prieigos reikalavimas 1

<b>Reikalavimas #:</b>	PR-1	<b>Reikalavimo tipas:</b>	V15a	<b>PA/FR #:</b>	1-8
<b>Aprašymas:</b>	Visa programėlės prieiga reikalauja identifikacijos rakto bei slaptažodžio.				
<b>Pagrindimas:</b>	Užtikrinant, kad visos funkcijos bus atliekamos autorizuotų naudotojų užtikrins, kad nebus atliekama nereikalingų bei nepageidaujamų operacijų su nenumatytais duomenimis.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Atlikus testavimą visos funkcijos bus atliekamos naudojant autorizuotu naudotojo (naudojant specialų žetoną) ir kitu atveju bus parodoma klaida.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

37 lentelė Prieigos reikalavimas 2

<b>Reikalavimas #:</b>	PER-2	<b>Reikalavimo tipas:</b>	V15a	<b>PA/FR #:</b>	Nėra
<b>Aprašymas:</b>	Visose sistemos dalyse prieigų raktai ir duomenys neturi būti talpinami viešai prieinamoje kodo vietoje ir turi būti patalpinti atskirame faile, kuris yra tik programuotojo įrenginyje.				
<b>Pagrindimas:</b>	Yra svarbu, kad prisijungimo raktai nebūtų paviešinti ir prieinami trečiosioms šalims patalpinus kodą tam tikrose saugyklose.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Atlikus testavimą visi prieigos raktai bus patalpinti atskirame faile.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

38 lentelė Vientisumo reikalavimas 1

<b>Reikalavimas #:</b>	VR-1	<b>Reikalavimo tipas:</b>	V15b	<b>PA/FR #:</b>	Visi
<b>Aprašymas:</b>	Programėlė tikrins įvedamų duomenų teisingumą.				
<b>Pagrindimas:</b>	Svarbu, kad duomenys, kurie bus naudojami atitikmeniui rasti ar į duomenų bazę įrašyti, atitiktų teisingumą, taip užtikrinant efektyvesnį bei saugesnį sistemos darbą.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu įvedami duomenys atitiks nurodytą teisingumą.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

39 lentelė Privatumo reikalavimas 2

<b>Reikalavimas #:</b>	PR-2	<b>Reikalavimo tipas:</b>	V15c	<b>PA/FR #:</b>	Visi
<b>Aprašymas:</b>	Programėlė neturi galėti rašyti duomenų į telefoną, išskyrus naudotojui leidus.				
<b>Pagrindimas:</b>	Bet kokių leidimų pažeidimas be naudotojo leidimo gali iššaukti administracinį pažeidimą.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Bandant rašyti duomenis į telefoną atsidarys leidimo patvirtinimo langas ir kurį tik patvirtinus duomenys bus įrašyti į telefoną (išskyrus atvejus, kai naudotojas buvo anksčiau leidęs ir pažymėjęs prisiminimo punktą).				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

#### 40 lentelė Savisaugos reikalavimas 1

<b>Reikalavimas #:</b>	SR-1	<b>Reikalavimo tipas:</b>	V15e	<b>PA/FR #:</b>	Visi
<b>Aprašymas:</b>	Sistema privalo būti apsaugota nuo įsilaužimų.				
<b>Pagrindimas:</b>	Yra be galo svarbu, kad duomenys ir pati sistema liktų saugi nuo įsilaužėlių, taip apsaugant naudotojų duomenis.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu bus atliekami įvairūs įsilaužimo būdai, o testavimo rezultatai turi grąžinti teigiamą rezultatą – į sistemą įsilaužti nepavyko.				
<b>Užsakovo pasitenkinimas:</b>	5	<b>Užsakovo nepasitenkinimas:</b>	5		
<b>Prioritetas:</b>	Aukštas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

### 2.1.16. Kultūriniai-politiniai reikalavimai

#### 41 lentelė Kultūrinis rinkos reikalavimas 1

<b>Reikalavimas #:</b>	KRR-1	<b>Reikalavimo tipas:</b>	V16a	<b>PA/FR #:</b>	4-7
<b>Aprašymas:</b>	Programėlė turėtų išskirti matavimo vienetų šalims, kurios naudoja skirtingus matavimo vienetus – vienos Celsijų, kitos – Farenheitą.				
<b>Pagrindimas:</b>	Yra svarbu, kad tam tikros šalies gyventojai galėtų lengvai suprasti nurodytus matavimo vienetų reikšmes.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Testavimo metu, priklausomai nuo regiono, bus atvaizduojamas skirtingas matavimo vieneto simbolis bei reikšmė.				
<b>Užsakovo pasitenkinimas:</b>	3	<b>Užsakovo nepasitenkinimas:</b>	1		
<b>Prioritetas:</b>	Žemas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 21 d.				

### 2.1.17. Teisiniai reikalavimai

Teisiniai reikalavimai nėra numatyti.

### **2.1.18. Atviri klausimai (problemos)**

Atvirų problemų ir klausimų nėra numatyta.

### **2.1.19. Egzistuojantys sprendimai**

#### **2.1.19.1. Prieinamos sistemos**

Toliau bus pateiktos prieinamos sistemos:

1. Duomenų realiu laiku sinchronizavimui gali būti panaudota tiek „Back4App“, „Parse“ arba „AWSAmplify“ platformos.
2. Serverio dalies realizavimui gali būti panaudota ir „NodeJS“, o mašininio mokymo modeliams kurti ir „Java“ programavimo kalba naudojant „Visual Studio Code“ programavimo aplinką.

#### **2.1.19.2. Prieinami komponentai**

Toliau bus pateikti prieinami komponentai:

1. Saugiam duomenų perdavimui iš programėlės į serverį būtų galima panaudoti atviro kodo biblioteką „Ktor“.

#### **2.1.19.3. Kopijuotini sprendimai**

Kita namų techninių įrenginių kampanija turi susikūrusi savo bendro pobūdžio temperatūros bei vandens stebėjimo sistemą, tačiau programėlė neatvaizduoja istorinių duomenų. Vis dėlto, nusipirkdami grafinės sąsajos programinį kodą mes galėtume sutaupyti apie 70 % laiko kuriant grafinę sąsają.

### **2.1.20. Naujos problemos**

#### **2.1.20.1. Problemos diegimo palinkai**

Poveikio diegimo aplinkai nėra numatyta.

#### **2.1.20.2. Įtaka jau instaliuotoms sistemoms**

Poveikio esamoms sistemoms nėra numatyta.

#### **2.1.20.3. Neigiamas vartotojų nusiteikimas**

Kadangi pagrindiniai programėlės naudotojai nėra labai gerai išmanantys bei prisitaikę prie technologijų, jiems gali būti sunku prisitaikyti prie įmantrių programėlės elementų bei animacijų, dėl ko jiems gali nepatikti naudotis programėle.

#### **2.1.20.4. Kliudantys diegimo aplinkos apribojimai**

Suplanuotos naudoti „Firebase“ duomenų bazės apkrovos elementai gali nesugebėti aptarnauti labai didelio klientų kiekio vienu metu, negu yra nustatyta reikalavimuose, dėl ko gali pasimatyti prastesnis našumas tam tikrais momentais.

## **2.1.20.5. Galimos naujos sistemos sukeltos problemos**

Kitų potencialių problemų nėra numatyta.

## **2.1.21. Uždaviniai**

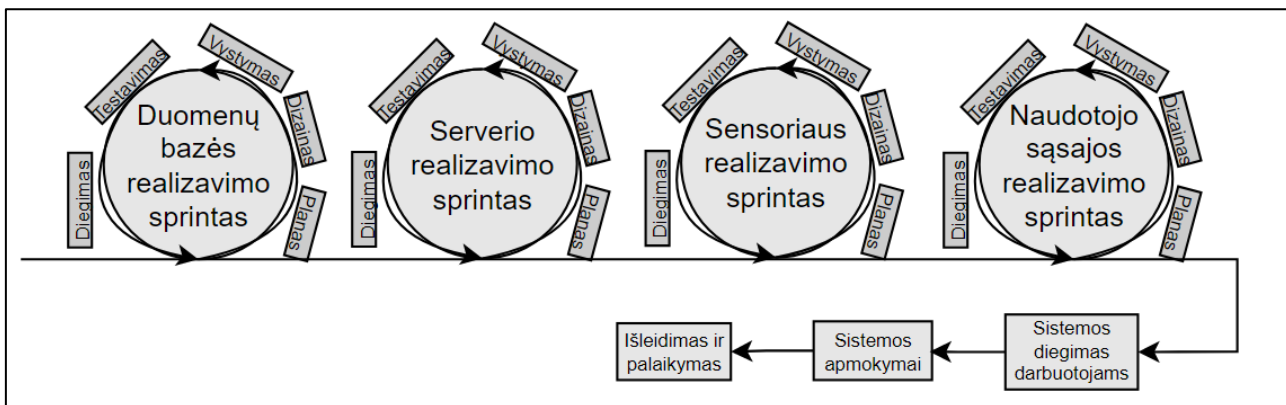
### **2.1.21.1. Sistemos kūrimo procesas**

Mobiliosios vandens stebėjimo programėlės kūrimo procesas bus vykdomas remiantis „Agile“ metodologija, kurios pagrindinė idėja yra sistemos padalinimas į keletą arba daugiau dalių ir kiekvienos atskiros dalies suplanavimas, įgyvendinimas, testavimas ir diegimas. Taip bus siekiama sumažinti projekto riziką klaidų atsiradimui iš anksto jas numatant, taip pat bus palaikomas didesnis ryšys su užsakovu sistemos kūrimo metu.

### **2.1.21.2. Detalus kūrimo planas**

**22 pav.** yra pateikta sistemos kūrimo proceso poskyryje minėta metodika, kuri bus taikoma toliau projekte realizuojanti bei diegiant sistemą. Sistemos kūrimo eigoje bus dirbama iteraciniu principu, kur kiekvienas iš išskirtų etapų (duomenų bazės, serverio, jutiklio realizavimo bei naudotojo sąsajos realizavimo) bus kuriamas tokia eiga: pirmiausia bus sukuriamas planas, tada sukurtas tam tikras prototipas, po to plano realizavimas, testavimas, o galiausiai diegimas. Perėjus visus etapus vyks sistemos mokymai bei diegimai esamiems darbuotojams. Galiausiai viską sėkmingai atlikus programėlė bus išleista ir toliau bus vykdomas jos palaikymas bei atnaujinimas. Toliau bus aptariama, kas įeina į kiekvieną etapą:

1. duomenų bazės realizavimas apima duomenų modelio transformavimą į duomenų bazės schemą ir jos įgyvendinimą duomenų talpinimo tikslu;
2. serverio realizavimas apima visus panaudojimo atvejus, tačiau bus įgyvendinama tik su „back-end“ susiję dalykai - duomenų priėmimas, saugojimas, persiuntimas bei mašininio mokymo algoritmo realizavimas;
3. jutiklio realizavimas apima 9 (-tą) panaudojimo atvejį - duomenų išsiuntimas, kur jutiklius siųs duomenis į duomenų brokerį tolimesniam duomenų apdorojimui;
4. naudotojo sąsajos realizavimo metu bus įgyvendinama tik su „front-end“ susiję dalykai - duomenų atvaizdavimas, įspėjamųjų signalų priėmimas bei naudotojo veiksmų apdorojimas;
5. įvykdžius visus etapus sistema bus diegiama darbuotojų pastatuose/namuose;
6. sėkmingai atlikus sistemos testavimą ir tikrinimą darbuotojų namuose vyks sistemos apmokymai;
7. galiausiai sistema bus išleista ir vykdomas jos palaikymas.



22 pav. Sistemos kūrimo procesas

## 2.1.22. Pritaikymas

### 2.1.22.1. Migravimo į naują produktą reikalavimai

Iš pradžių sistema bus įdiegta tik esamiems darbuotojams užtikrinant, kad sistema veikia taip, kaip tikėtasi, ir tik po to ji bus išleidžiama į rinką naudotis visiems.

### 2.1.22.2. Duomenų transformavimo reikalavimai

Duomenų transformavimo reikalavimų nėra numatyta.

## 2.1.23. Rizikų įvertinimas

Toliau bus pateiktos galimos projekto rizikos, jų tikimybės, poveikis bei galimas sprendimas.

1. Nebus laiku spėta pabaigti sistemos.
  - **Rizikos tikimybė:** Vidutinė.
  - **Poveikis:** Žemas.
  - **Galimas sprendimas:** Pasamdyti daugiau darbuotojų arba sumažinti sistemos numatytų panaudojimo atvejų kiekį.
2. Sistema kainuos daugiau negu numatyta apribojimuose.
  - **Rizikos tikimybė:** Vidutinė.
  - **Poveikis:** Vidutinis.
  - **Galimas sprendimas:** Naudoti pigesnius jutiklius arba sumažinti darbuotojų ar sistemos numatytų panaudojimo atvejų kiekį.
3. Nedidelė produkto kokybė.
  - **Rizikos tikimybė:** Žema.
  - **Poveikis:** Aukštas.
  - **Galimas sprendimas:** Skirti daugiau laiko kuriant sistemą arba sumažinti sistemos numatytų panaudojimų atvejų kiekį.
4. Prastas mašininio mokymo algoritmas.
  - **Rizikos tikimybė:** Maža.

- **Poveikis:** Aukštas.
- **Galimas sprendimas:** Naudoti paprastą algoritmą taip sumažinant projekto kokybę.

#### 2.1.24. Kaina

Projekto biudžetas, pradedant nuo 2021 m. lapkričio 24 d. yra pateiktas **42 lentelėje**.

**42 lentelė.** Projekto biudžetas

Išlaidos	Vienetas	Vienetų skaičius	Vieneto kaina, Eur	Viso, Eur
<b>1. Žmonių ištekliai</b>				
Projekto vadovas	Mėnesis	12 * 0.2	1900	4560
Programuotojas	Mėnesis	12 * 0.1	3000	3600
<i>Iš viso žmonių išteklių</i>				8160
<b>2. Kelionės</b>	Vienetas			
Konferencijos	Vienetas	2	1000	2000
<i>Iš viso kelionės</i>				2000
<b>3. Įranga ir prekės</b>				
„IntelliJ“	Licencija metams	1	499	499
<i>Iš viso Įranga ir prekės</i>				499
<b>4. Biuro išlaikymas</b>				
Elektros, šildymo, telefono, nuomos išlaidos	Mėnesis	12	220	2640
<i>Iš viso biuro išlaikymas</i>				2640
<b>5. Viso tiesioginiai projekto kaštai</b>				11499

#### 2.1.25. Vartotojo dokumentacija ir apmokymas

Toliau bus pateikti reikalavimai naudotojų dokumentacijai:

##### 1. Techninė specifikacija.

- **Skirta:** Programuotojams, inžinieriams, užsakovui.
- **Palaikymas:** Esamos IT sistemos autorius, kuris kūrė šią specifikaciją, turės atlikti ir atnaujinimo darbus.
- **Dokumentacijos pateikimo forma:** „DOC“ arba „DOCX“ formatu.

##### 2. Naudotojo vadovas.

- **Skirta:** Naudotojams.
- **Palaikymas:** Tie, kurie kurs naujas versijas, turės atnaujinti naudotojo vadovą.
- **Dokumentacijos pateikimo forma:** „DOC“ arba „DOCX“ formatu.

##### 3. Diegimo (instaliacijos) vadovas.

- **Skirta:** Naudotojams.

- **Palaikymas:** Vadovą turės atnaujinti techninės komandos nariai, atlikę bet kokius atnaujinimus, susijusius su diegimu.
- **Dokumentacijos pateikimo forma:** „DOC“ arba „DOCX“ formatu.

## 2.1.26. Perspektyviniai reikalavimai

43 lentelė Perspektyvinis reikalavimas 1

<b>Reikalavimas #:</b>	PR-1	<b>Reikalavimo tipas:</b>	V26	<b>PA/FR #:</b>	N/A
<b>Aprašymas:</b>	Programėlė turėtų būti pritaikyta ir „iOS“ operacinę sistemą turintiems įrenginiams.				
<b>Pagrindimas:</b>	Nors „iOS“ operacinę sistemą turinčių naudotojų rinka nėra didelė, tačiau vis dėlto tai padidintų esamų naudotojų ratą taip padarant programėlę dar labiau populiariesne, leidžiančia atrasti dar daugiau potencialių idėjų.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Programėlė turi veikti ir „iOS“ operacinėje sistemoje.				
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b>	1		
<b>Prioritetas:</b>	Žemas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 22 d.				

44 lentelė Perspektyvinis reikalavimas 2

<b>Reikalavimas #:</b>	PR-2	<b>Reikalavimo tipas:</b>	V26	<b>PA/FR #:</b>	N/A
<b>Aprašymas:</b>	Sistema turėtų palaikyti daugiau jutiklių tipų.				
<b>Pagrindimas:</b>	Didesnė jutiklių palaikymo įvairovė leistų pritraukti daugiau naudotojų.				
<b>Šaltinis:</b>	Užsakovas L. Kulikovas.				
<b>Atitikimo kriterijus:</b>	Sistema palaiko ir drėgmės bei oro kokybės jutiklius.				
<b>Užsakovo pasitenkinimas:</b>	4	<b>Užsakovo nepasitenkinimas:</b>	1		
<b>Prioritetas:</b>	Žemas	<b>Konfliktai:</b>	Nėra		
<b>Istorija:</b>	Užregistruotas 2022 vasario 22 d.				

## 2.1.27. Idėjos ir sprendimai

Toliau bus pateiktos idėjos ir sprendimai:

1. Mašininio mokymo integravimas į temperatūros stebėjimo jutiklius galbūt galėtų patobulinti naudotojų resursų sunaudojimo pasiūlymų efektyvumą.

## 2.2. Architektūros specifikacija

### 2.2.1. Architektūros pateikimas

Visa šios sistemos architektūra yra pateikta naudojantis UML modeliavimo įrankiu, naudojantis „Magic Draw“ modeliavimo programa, kur kiekvienas iš vaizdų yra pateiktas tokiomis diagramomis: panaudos atvejai – panaudos atvejų diagrama, statinis sistemos vaizdas – klasių diagrama, dinaminis sistemos vaizdas – sekų bei veiklos diagramomis ir galiausiai išdėstymo vaizdas – diegimo diagrama.

## 2.2.2. Architektūros tikslai ir apribojimai

Toliau bus pateikti architektūriniai tikslai ir apribojimai:

- kiekvieno naudotojo suvartoti duomenys apie vandenį ir temperatūrą privalo būti apsaugoti nuo trečiųjų šalių;
- į sistemą privalo būti nesunku integruoti naują jutiklio daviklį;
- naudotojas savo duomenis turės galėti peržiūrėti „Android 5.0“ ar naujesnę operacinę sistemą turinčiame įrenginyje;
- naudotojo išpėjamieji pranešimai „Firebase Cloud Messaging“ paslaugoje turėtų būti saugomi 7 dienas, jeigu jie negalėtų pasiekti naudotojo dėl tam tikrų priežasčių.

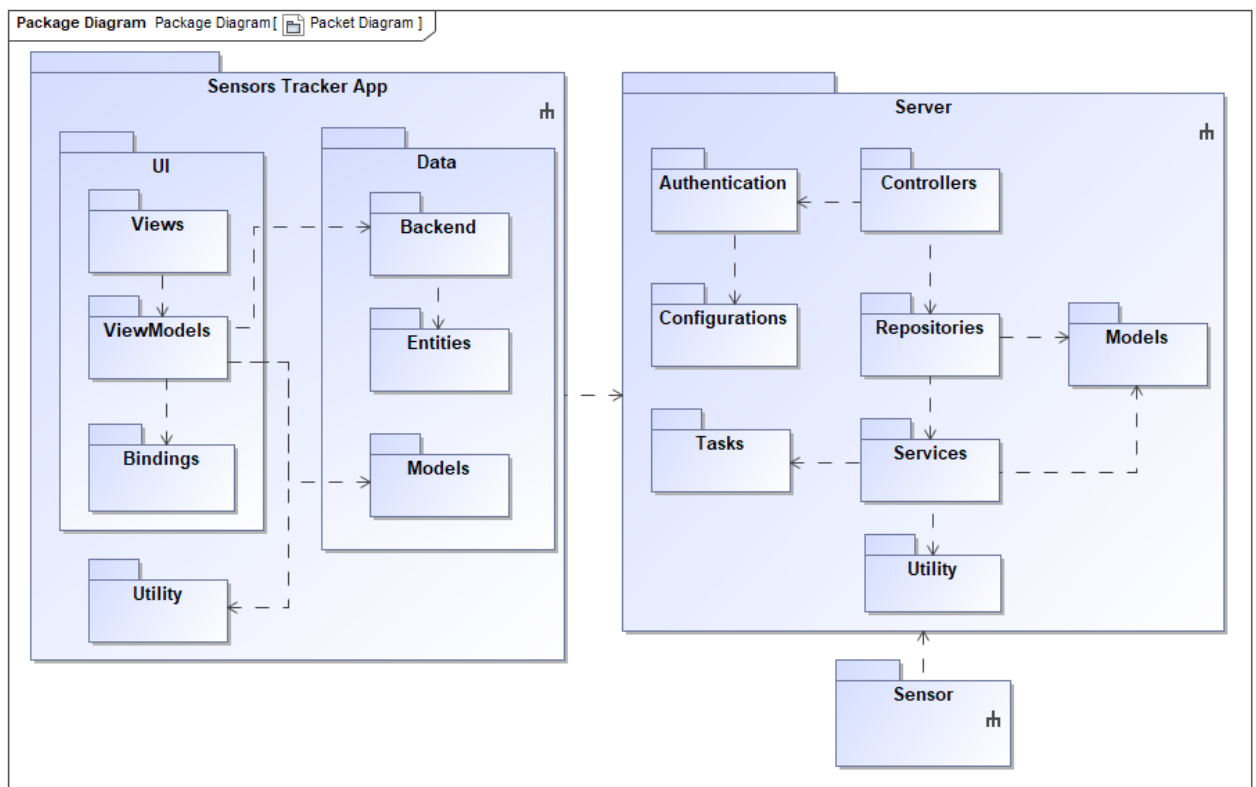
## 2.2.3. Panaudojimo atvejų vaizdas

Naudotojo, administratoriaus ir serverio panaudojimo atvejų diagramos yra pateiktos atitinkamai **18 pav.**, **19 pav.** ir **20 pav.**

## 2.2.4. Sistemos statinis vaizdas

### 2.2.4.1. Apžvalga

**23 pav.** yra pavaizduota sistemos paketų diagrama.



**23 pav.** Sistemos paketų diagrama

### 2.2.4.2. Paketų detalizavimas

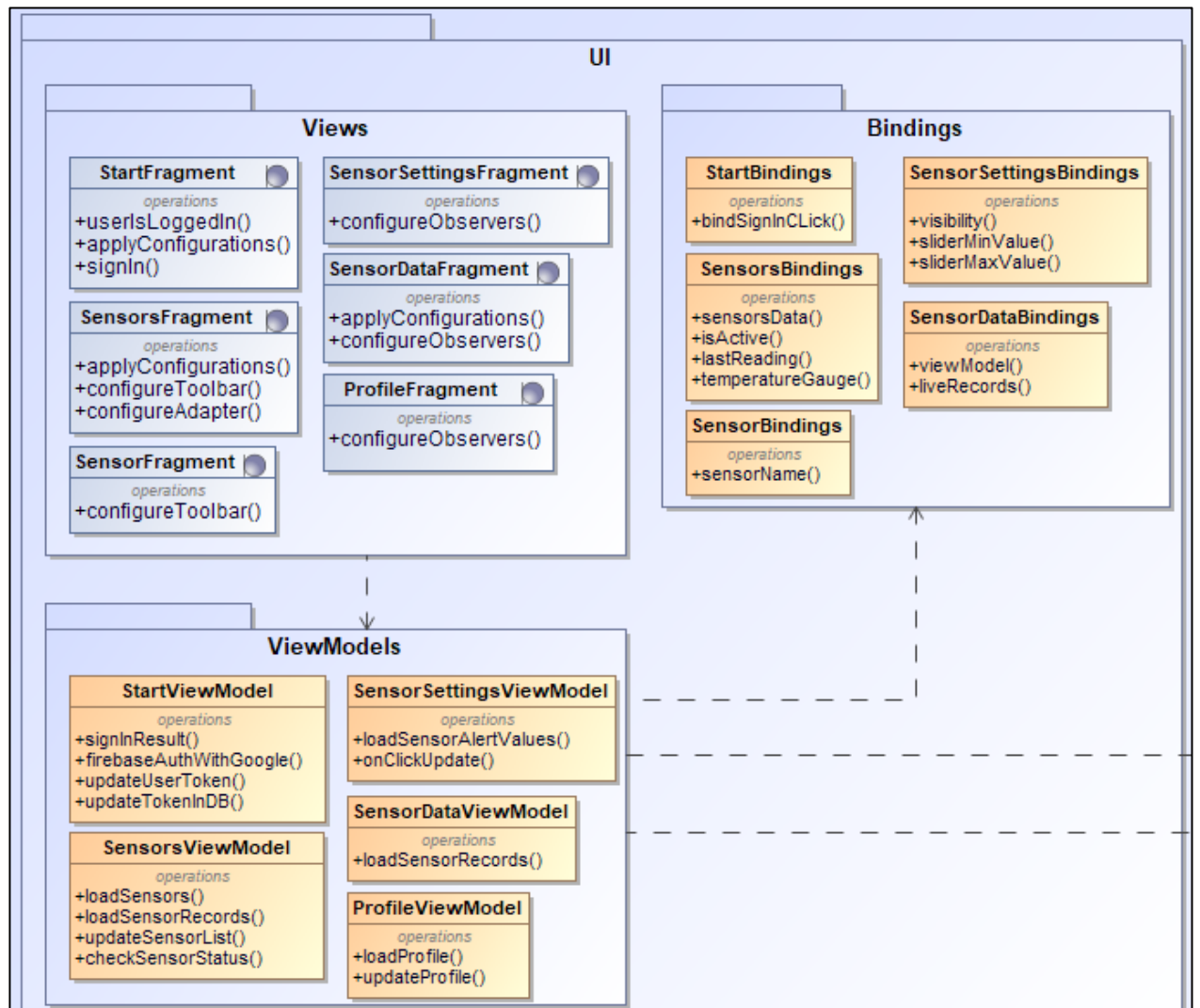
Kiekvienam paketui (mobiliosios aplikacijos, serverio bei jutiklio) yra pateikiamas jo trumpas aprašymas ir detali paketų-klasių diagrama.



### 2.2.4.2.1. Mobilios aplikacijos paketų diagramos

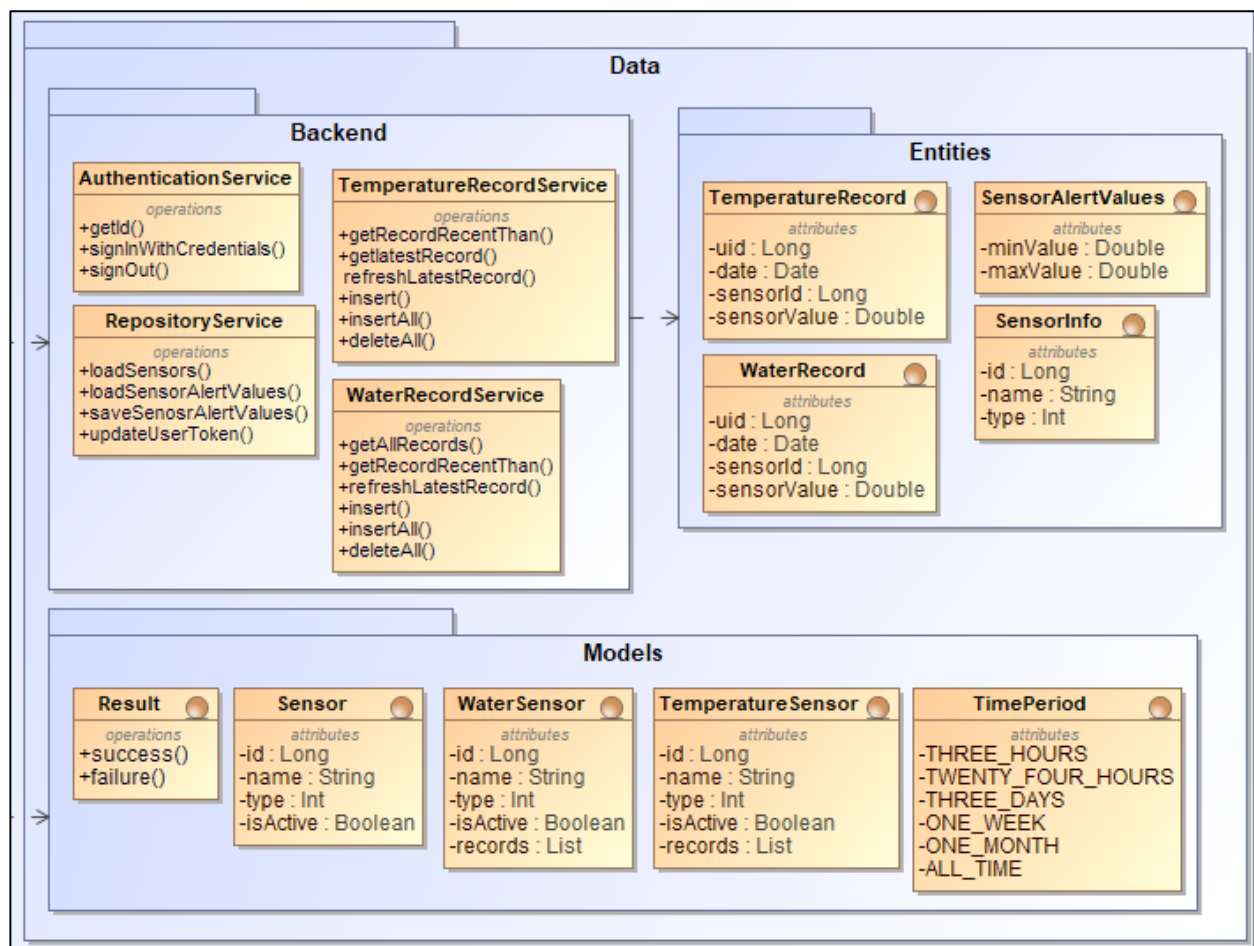
Šiame poskyryje yra pateiktos paketų diagramos, kurios yra susijusios su mobiliąja aplikacija.

**24 pav.** yra pateikta paketų diagrama, kuri yra atsakinga už duomenų atvaizdavimą naudotojui.



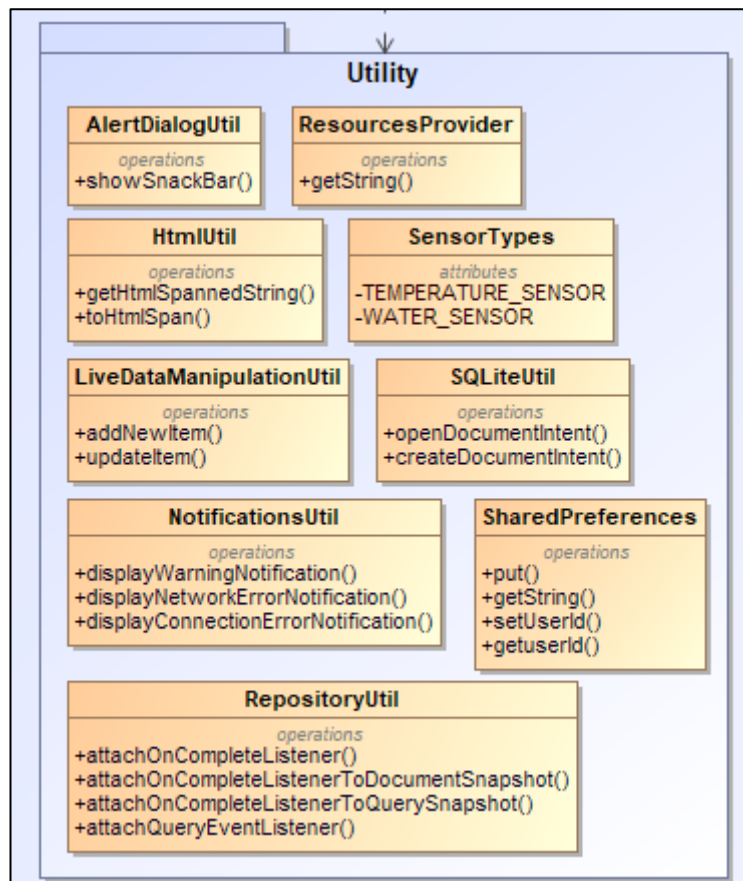
**24 pav.** Mobiliosios aplikacijos vartotojo sąsajos paketų diagrama

**25 pav.** yra pateikta paketų diagrama, kuri yra atsakinga už duomenų išsaugojimą bei sinchronizavimą tiek su serveriu, tiek su „Firebase“ paslauga.



25 pav. Mobiliosios aplikacijos duomenų paketų diagrama

26 pav. yra pateikta paketų diagrama, kuri talpina pagalbines klases, atsakingas už: įspėjamuosius pranešimus, lokalią duomenų bazę, „LiveData“ duomenų manipuliavimą bei duomenų saugyklą.

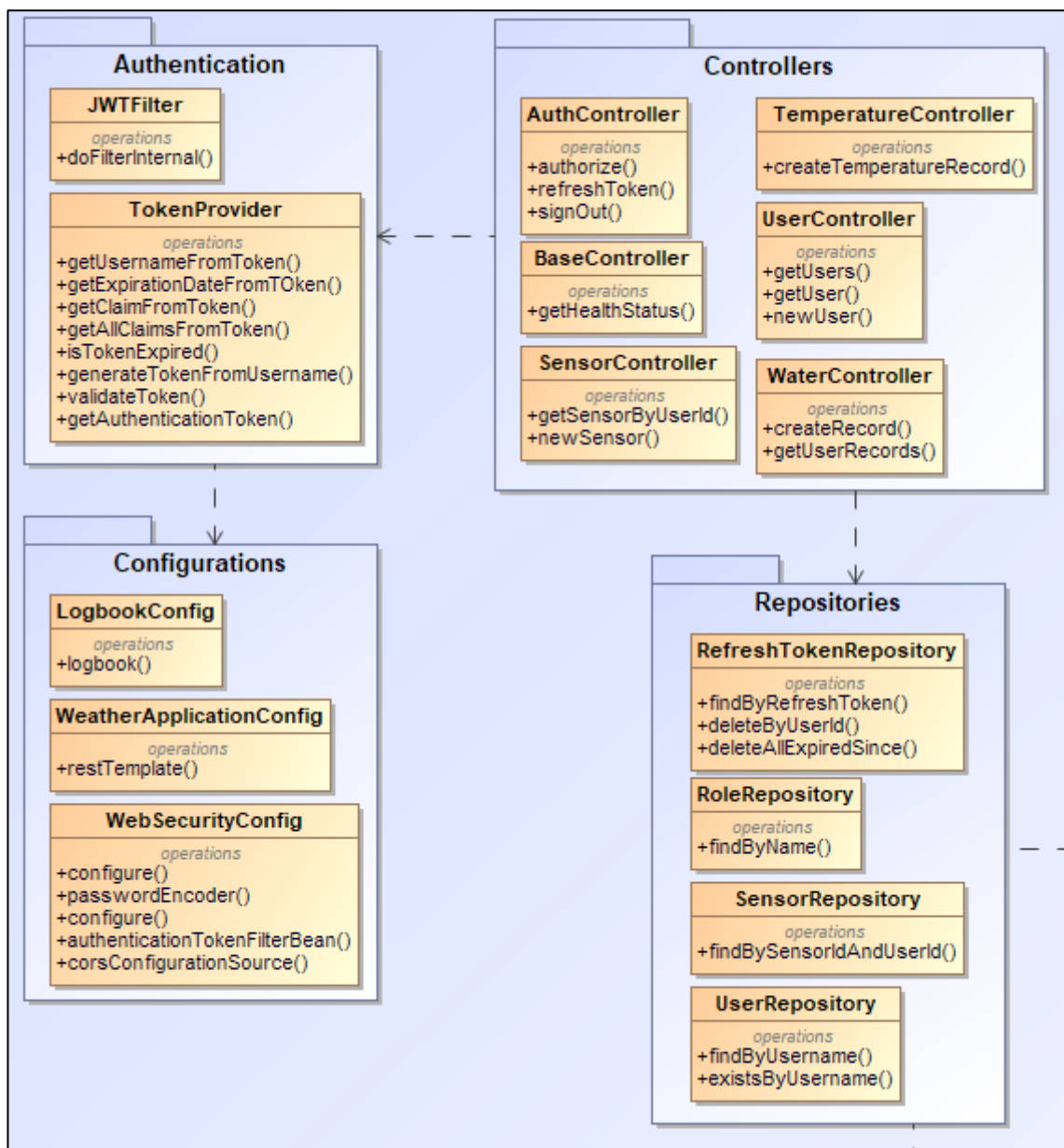


26 pav. Mobiliosios aplikacijos pagalbinių klasių paketų diagrama

#### 2.2.4.2.2. Serverio paketų diagramos

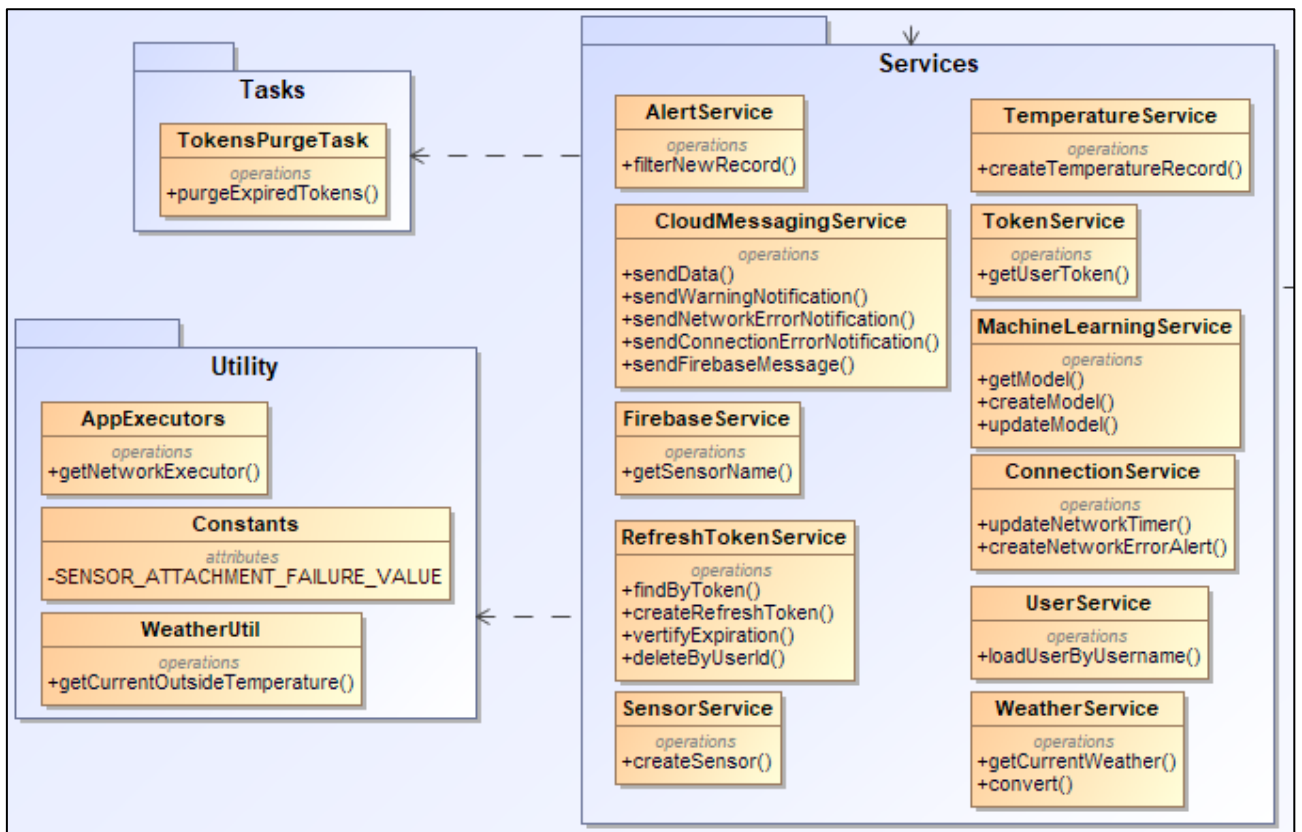
Šiame poskyryje yra pateiktos paketų diagramos, kurios yra susijusios su serveriu.

27 pav. yra pateikta paketų diagrama, kuri yra atsakinga už autorizaciją bei duomenų išgavimą ir išsaugojimą.



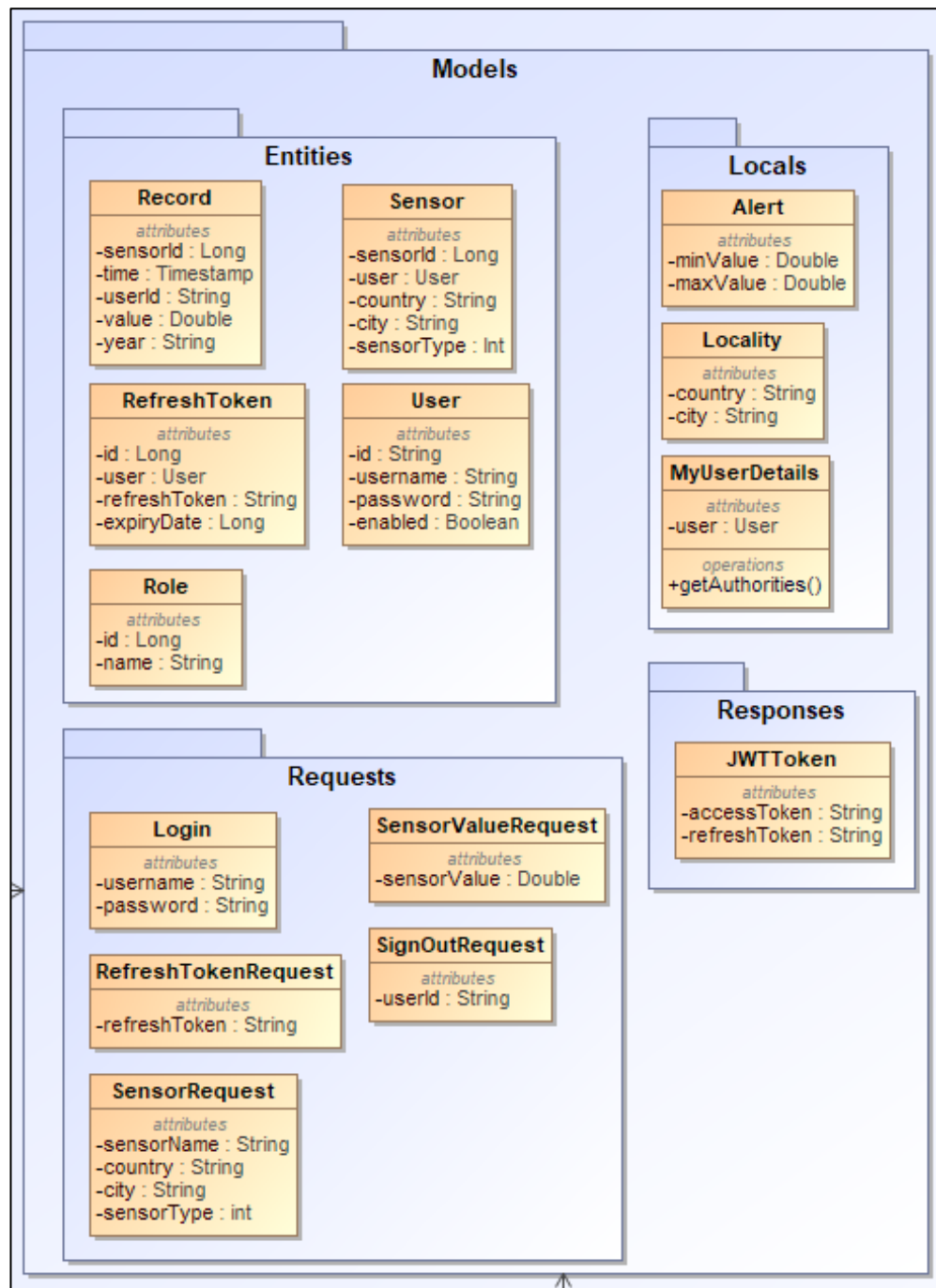
27 pav. Serverio autorizacijos bei duomenų gavimo paketų diagrama

28 pav. yra pateikta paketų diagrama, kuri yra atsakinga už servigus bei pagalbines klases.



28 pav. Serverio servisų bei pagalbinių klasių paketų diagrama

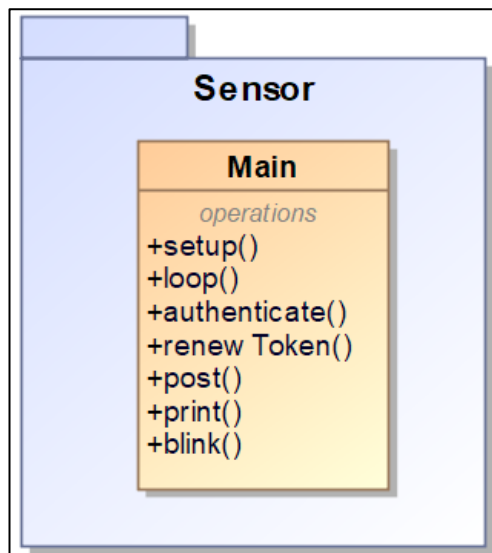
29 pav. yra pateikta paketų diagrama, kuri yra atsakinga už modelių talpinimą.



29 pav. Serverio modelių paketų diagrama

### 2.2.4.3. Jutiklio paketų diagrama

30 pav. yra pateikta paketų diagrama, kuri yra atsakinga už jutiklio pagrindinių metodų talpinimą.



30 pav. Jutiklio paketų diagrama

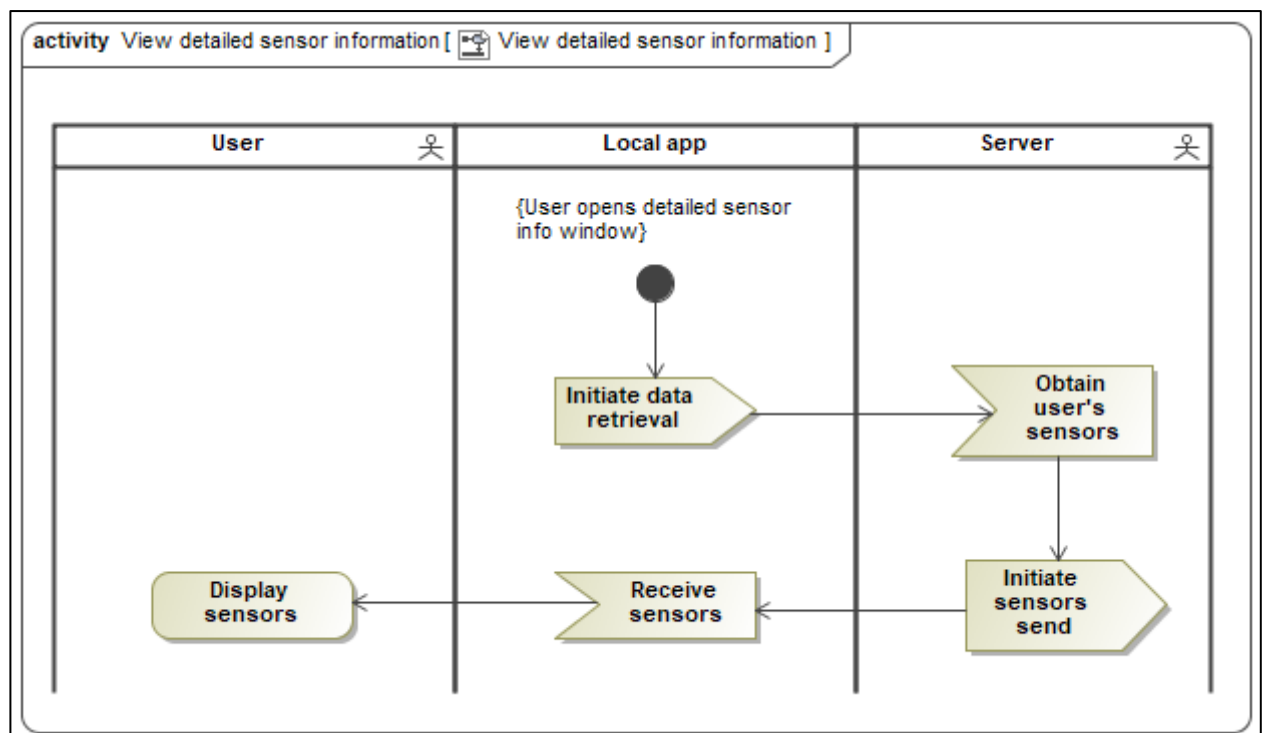
## 2.2.5. Sistemos dinaminis vaizdas

Šiame skyriuje yra pateikiamos sekų, būsenų bei veiklos diagramos.

### 2.2.5.1.1. Veiklos diagramos

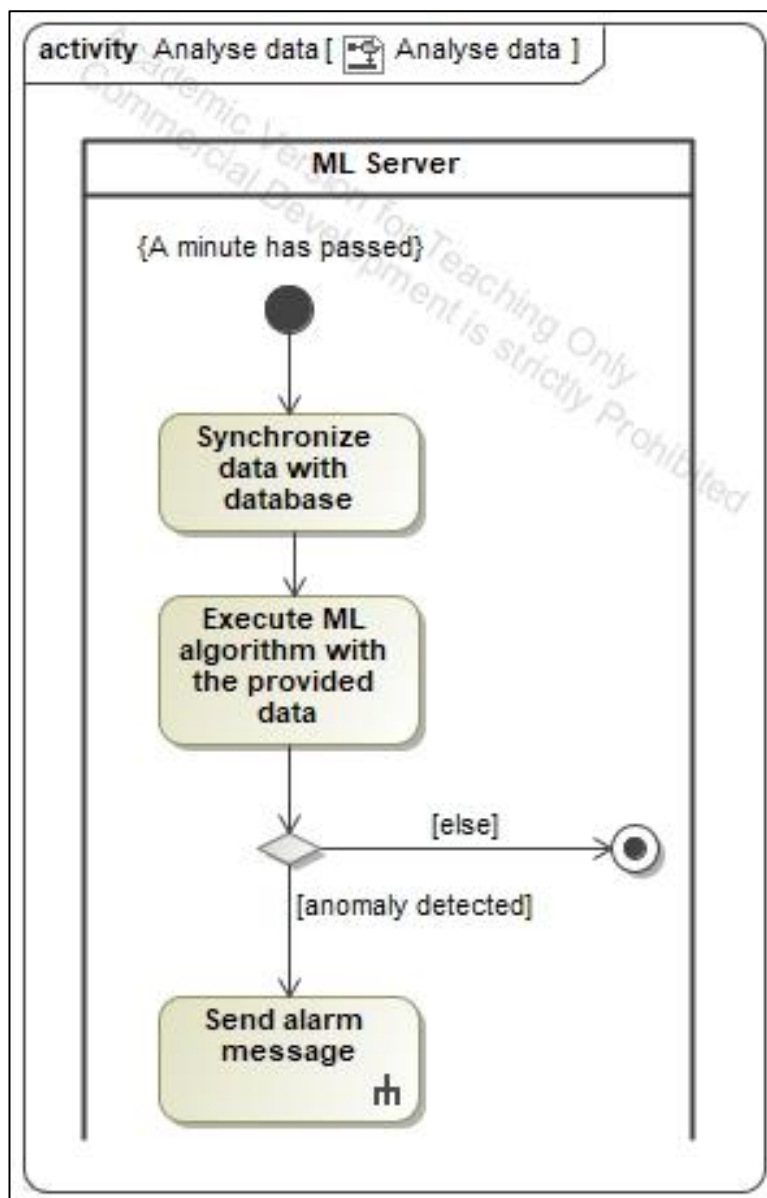
Šiame poskyryje yra pateikiamos veiklos diagramos (žr. nuo 31 pav. iki 33 pav.).

31 pav. yra pateikta detalios jutiklio informacijos peržiūros veiklos diagrama.



31 pav. Detalios jutiklio informacijos peržiūros veiklos diagrama

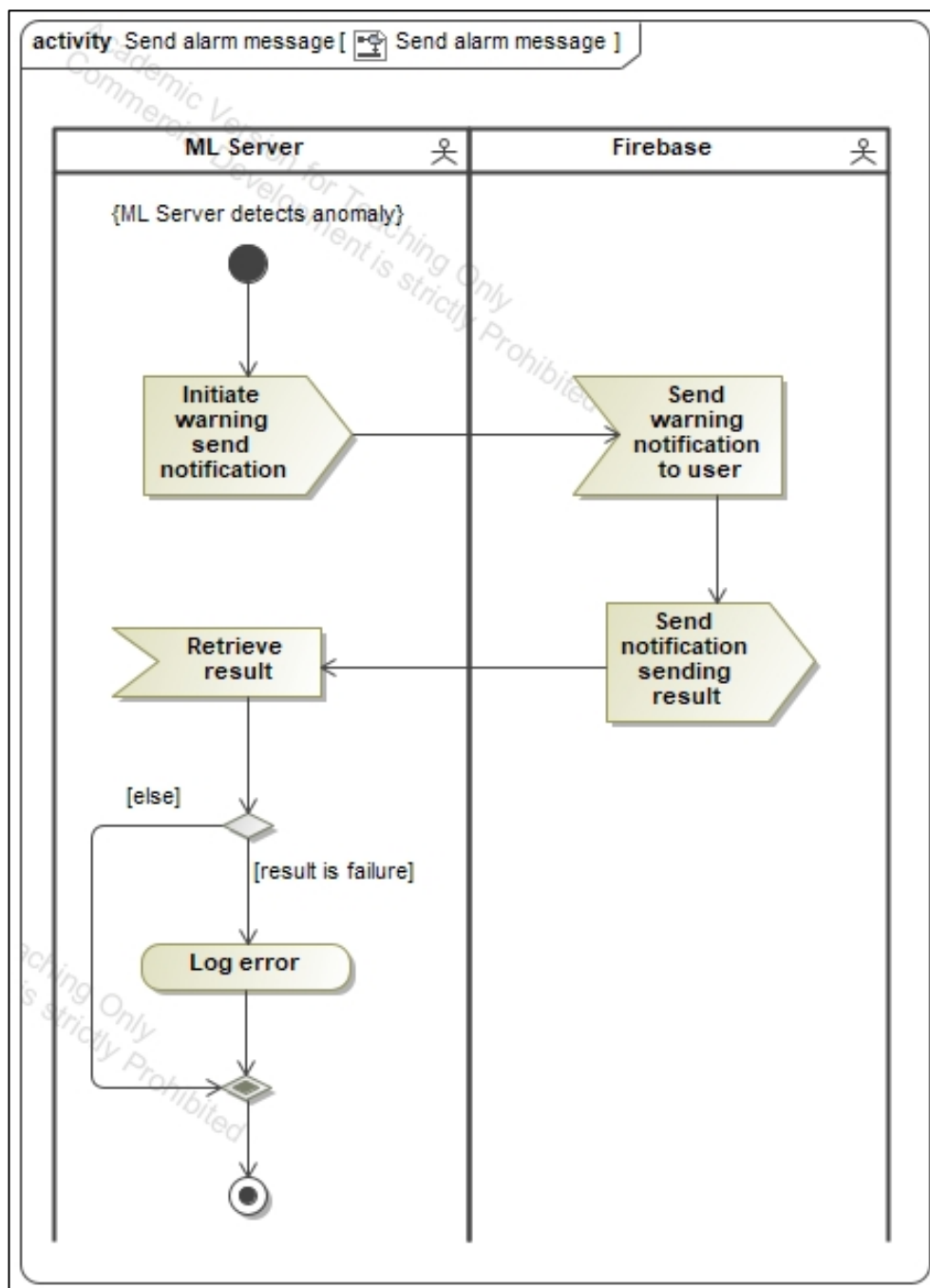
32 pav. yra pateikta mašininio mokymo duomenų analizės veiklos diagrama.



32 pav. Mašininio modelio duomenų analizės veiklos diagrama

33 pav. yra pateikta įspėjamojo pranešimo siuntimo veiklos diagrama.



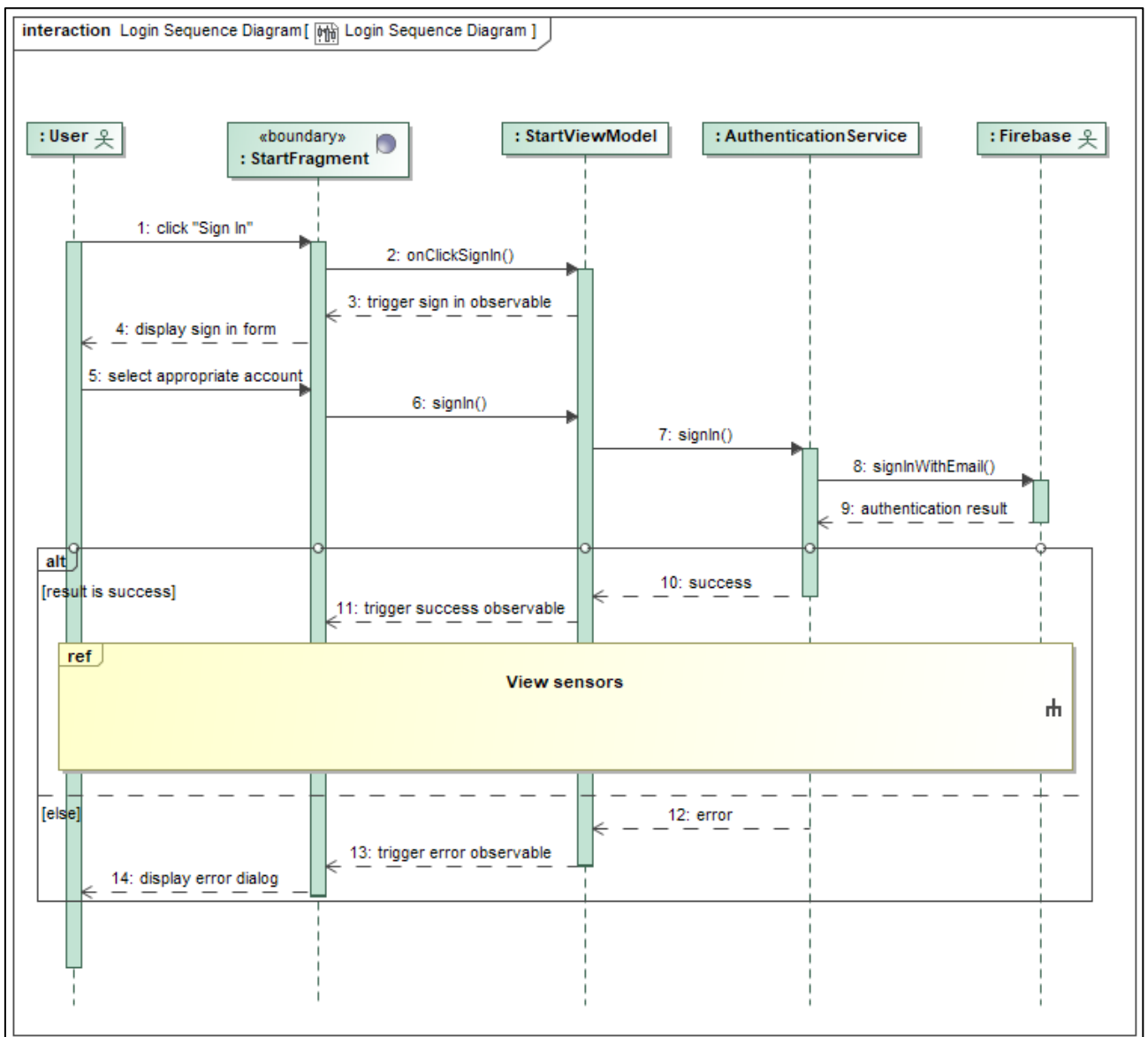


33 pav. Išpėjamojo pranešimo siuntimo veiklos diagrama

### 2.2.5.2. Sekų diagramos

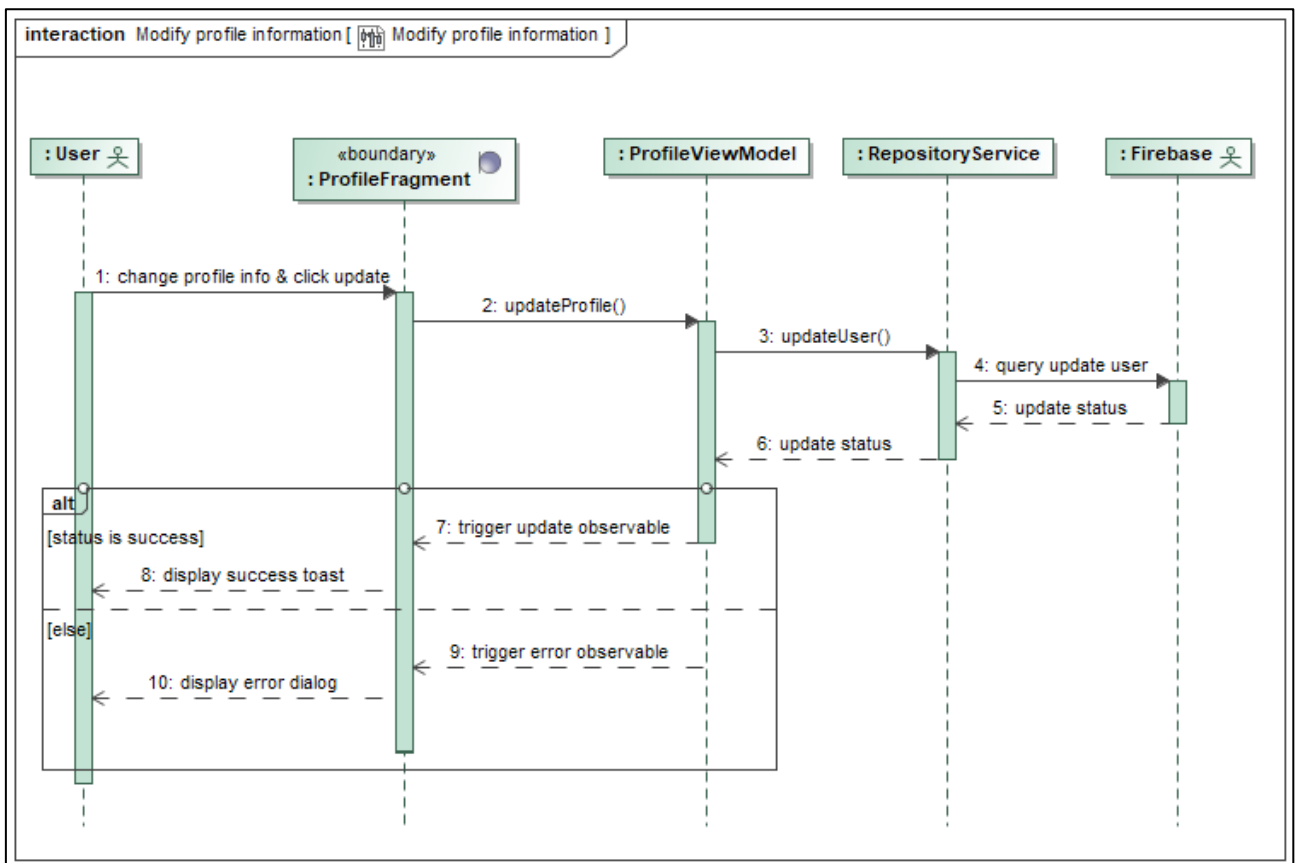
Šiame poskyryje bus pateiktos sekų diagramos (žr. nuo 34 pav. iki 41 pav.).

34 pav. yra pavaizduota prisijungimo sekų diagrama.



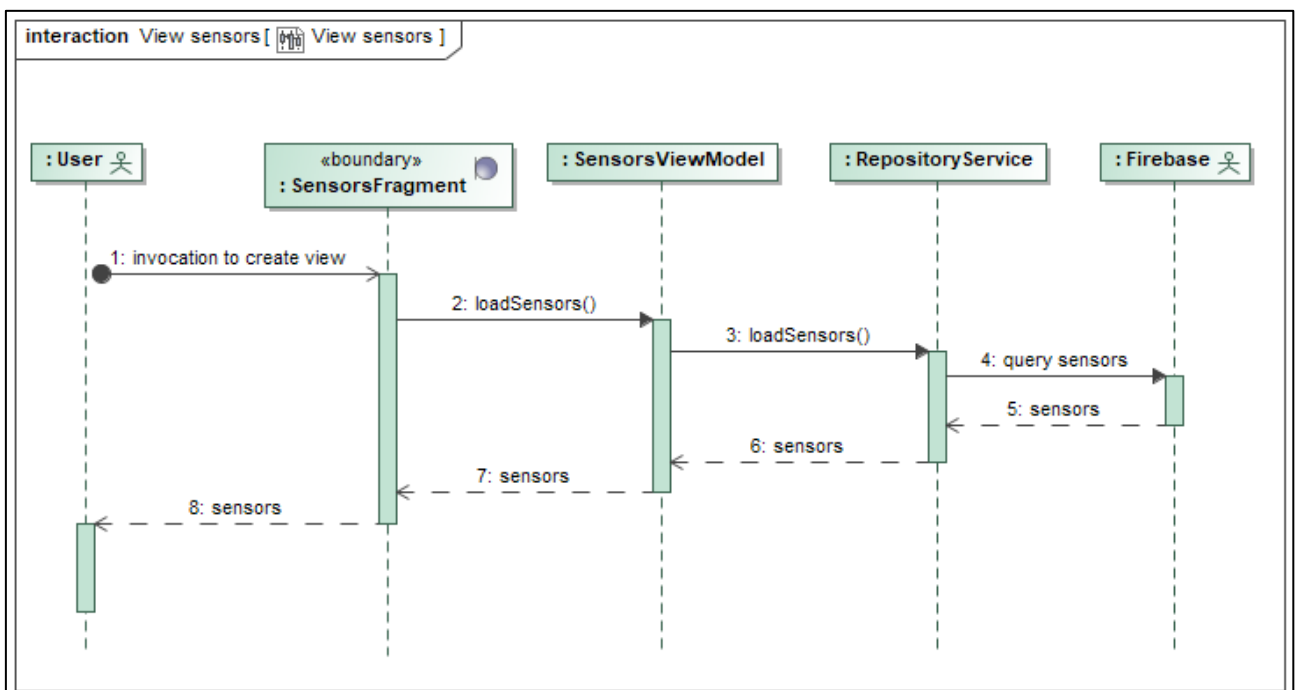
34 pav. Prisijungimo sekų diagrama

35 pav. yra pateikta profilio informacijos koregavimo sekų diagrama.



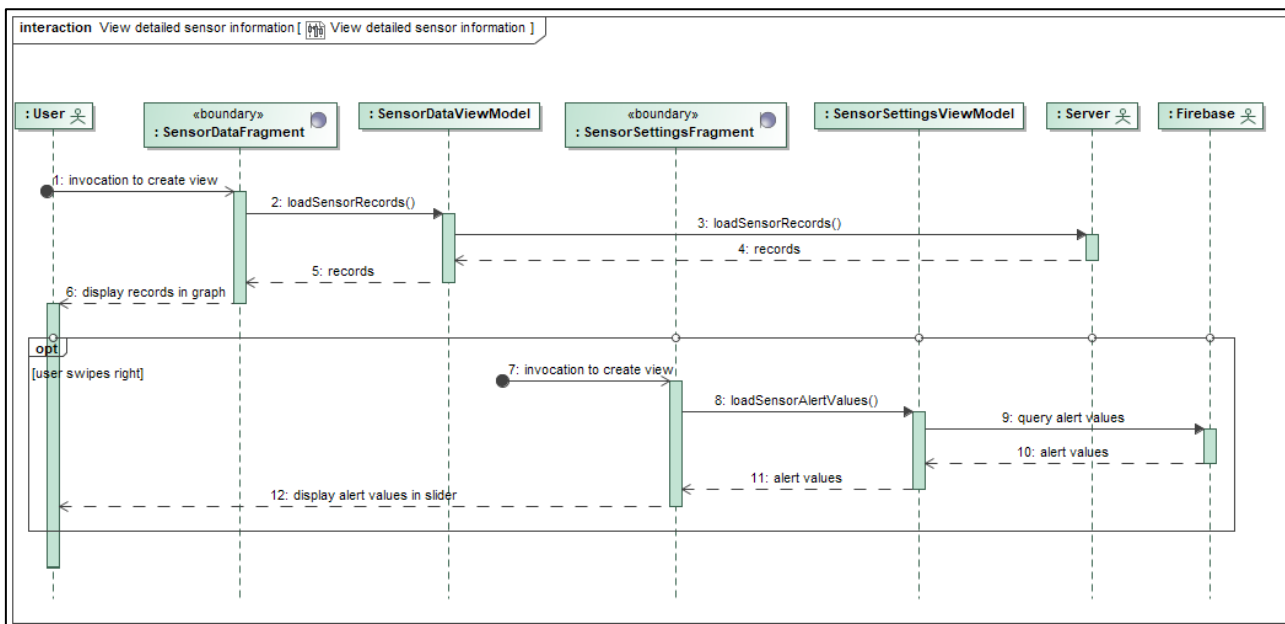
35 pav. Profilio informacijos koregavimo sekų diagrama

36 pav. yra pateikta jutiklių peržiūros sekų diagrama.



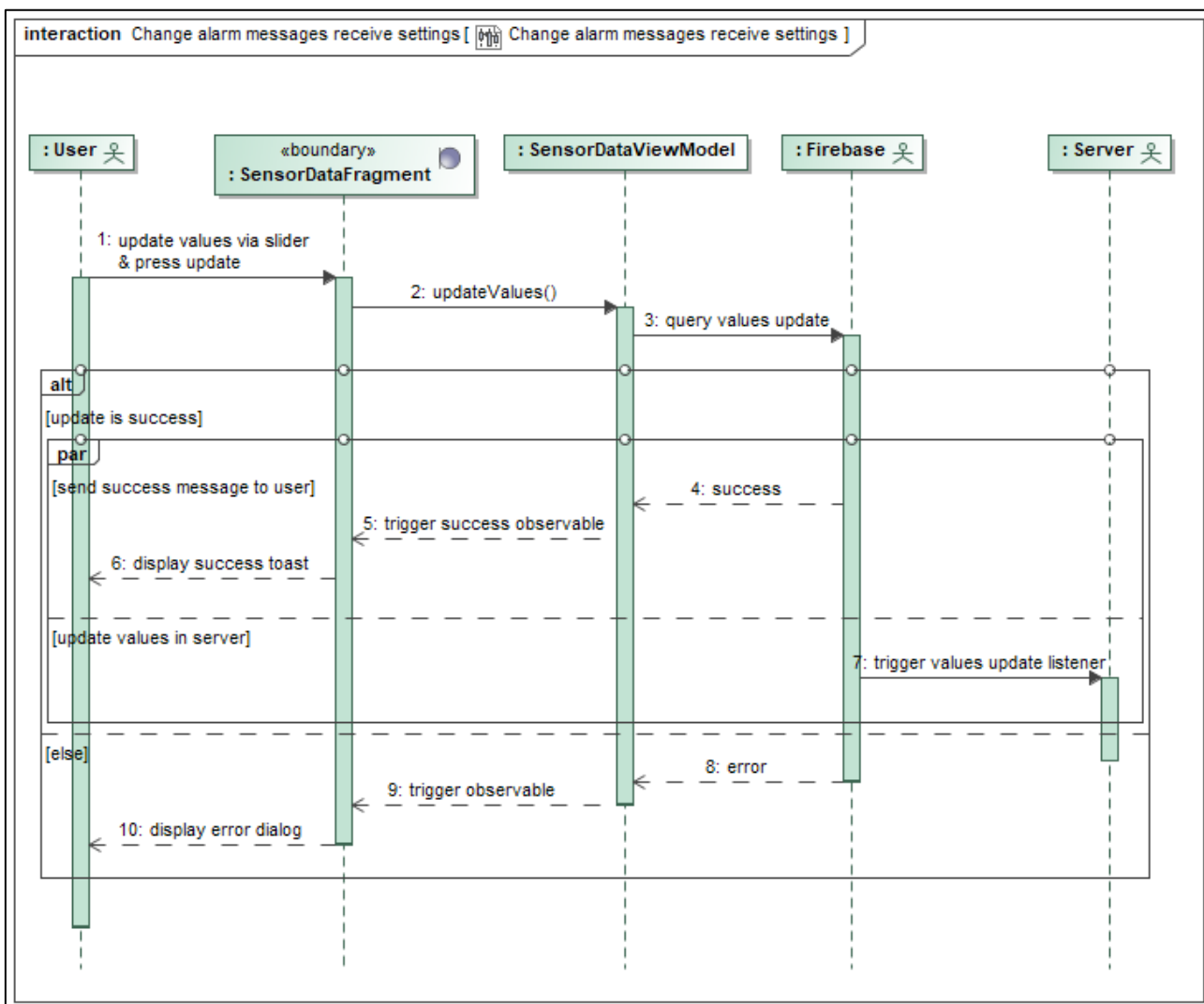
36 pav. Jutiklių peržiūros sekų diagrama

37 pav. yra pateikta detalios jutiklio informacijos peržiūros sekų diagrama.



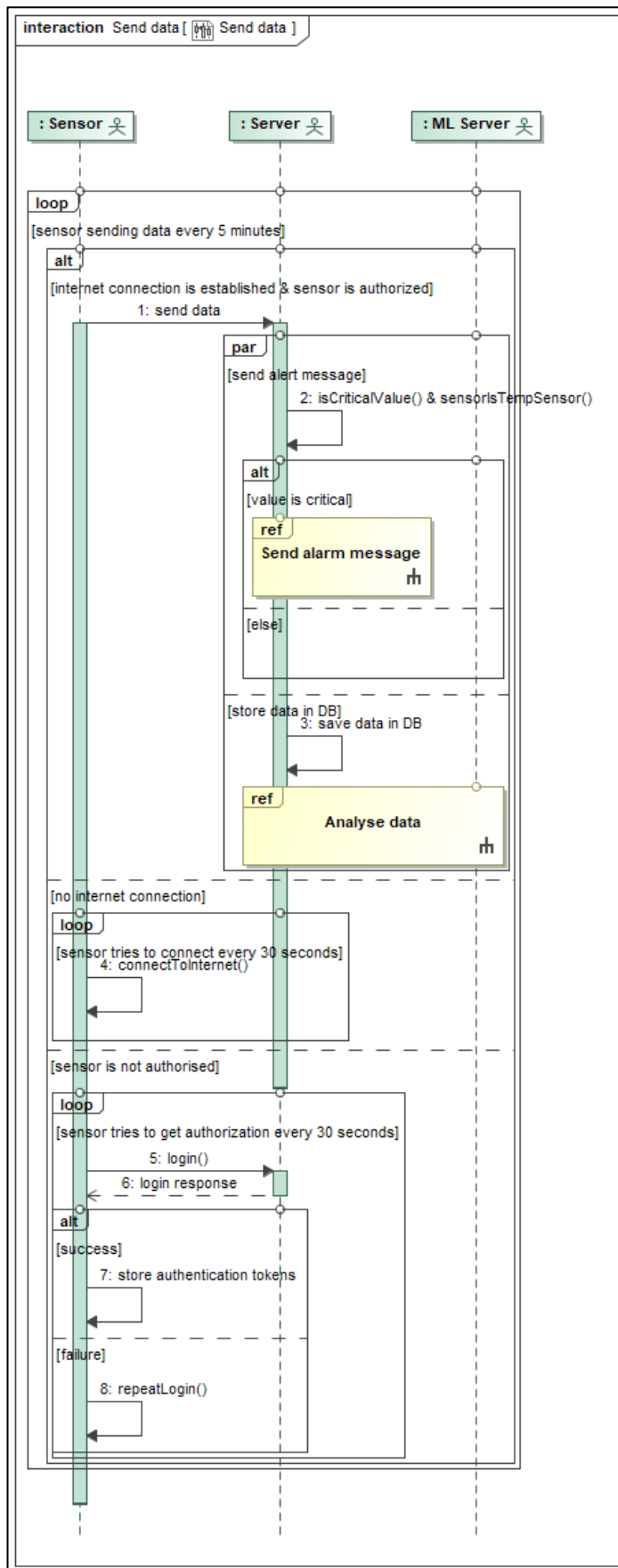
37 pav. Detalios jutiklio informacijos peržiūros sekų diagrama

38 pav. yra pateikta išspėjamųjų pranešimų nustatymų keitimo sekų diagrama.



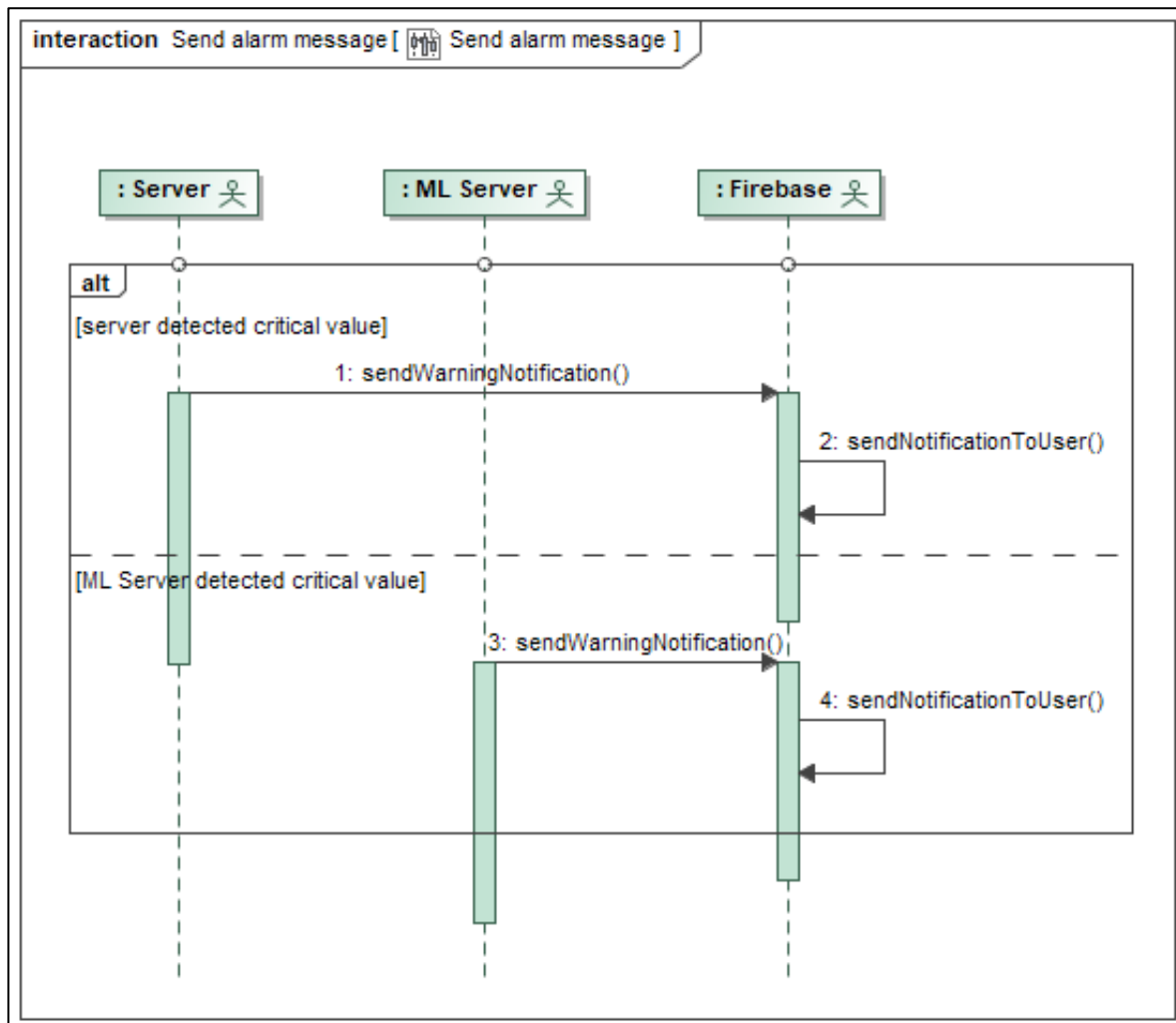
38 pav. Išspėjamųjų pranešimų nustatymų keitimo sekų diagrama

39 pav. yra pateikta jutiklio duomenų siuntimo sekų diagrama.



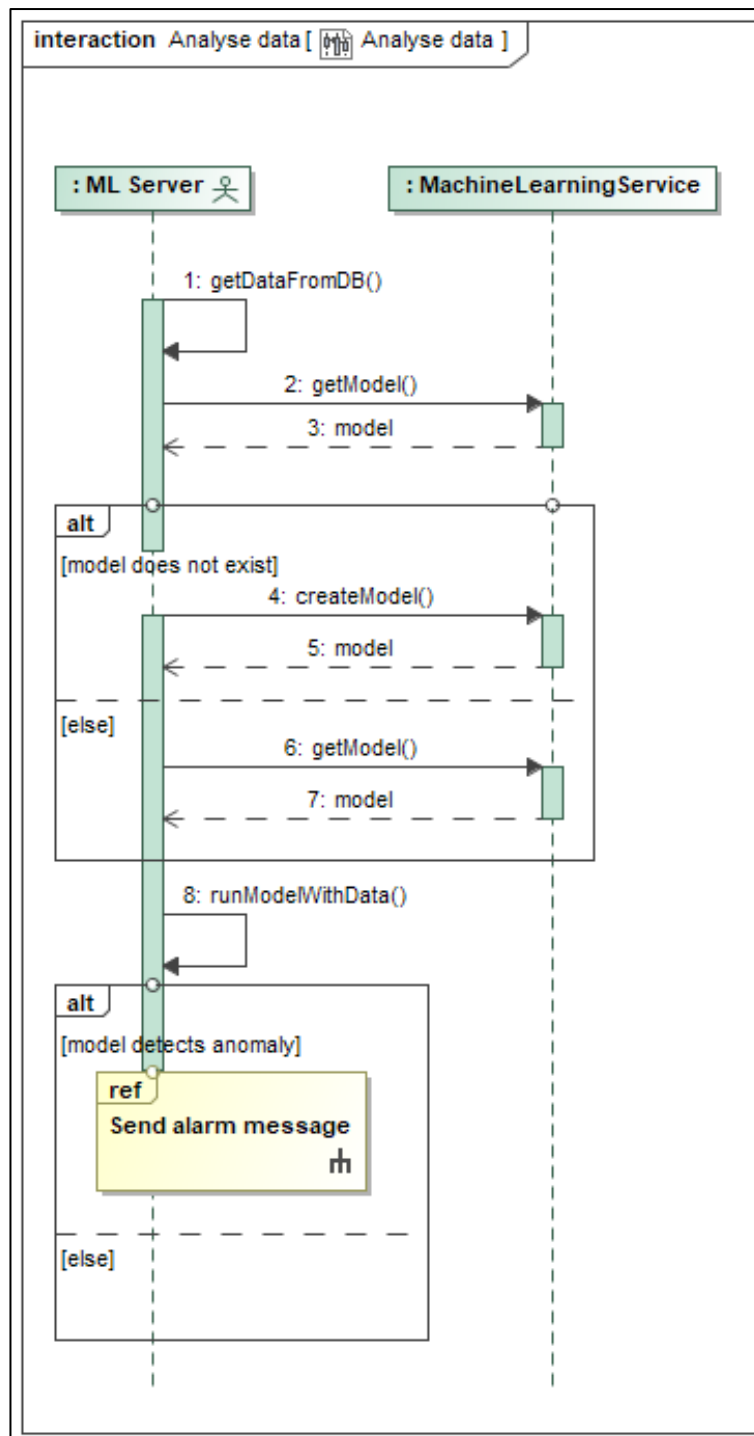
39 pav. Jutiklio duomenų siuntimo sekų diagrama

40 pav. yra pateikta įspėjamojo pranešimo siuntimo sekų diagrama.



40 pav. Įspėjamojo pranešimo siuntimo sekų diagrama

41 pav. yra pateikta mašininio mokymo duomenų analizės sekų diagrama.

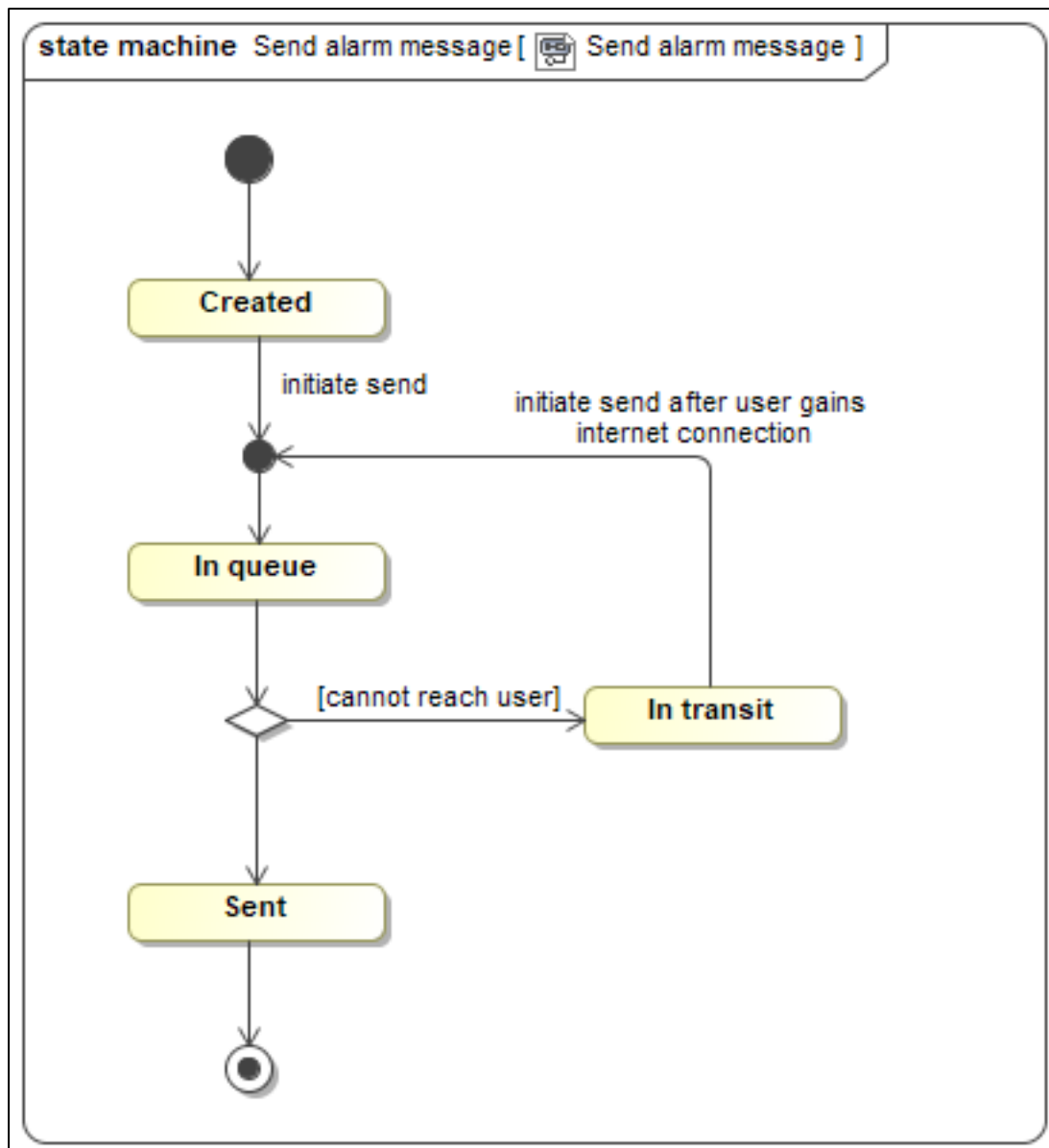


41 pav. Mašininio mokymo duomenų analizės sekų diagrama

### 2.2.5.3. Būsenų diagramos

Šiame poskyryje yra pateikiamos būsenų diagramos.

42 pav. yra pateikta išspėjamojo pranešimo siuntimo būsenų diagrama.



42 pav. Įspėjamojo pranešimo siuntimo būsenų diagrama

### 2.2.6. Išdėstymo vaizdas

Sistemos išdėstymo vaizdas yra pateiktas 16 pav.

45 lentelėje yra pateikta kiekvienos aplinkos reikalavimai bei konfigūracijos nustatymai.

45 lentelė Aplinkų reikalavimai bei konfigūracijos nustatymai

Aplinka	Minimalūs reikalavimai korektiškam veikimui	Esami konfigūracijos nustatymai
Vandens tėkmės jutiklius „YF-B5“	-	<ul style="list-style-type: none"> <li>- Įvestis nuo 3.0V iki 5.5V.</li> <li>- Temperatūros diapazonas nuo -55°C iki +125°C laipsnių.</li> <li>- 0.5°C paklaida nuo -10°C iki +85°C laipsnių.</li> <li>- 3 laidų sąsaja: raudonas – „VCC“, juodas – „GND“, geltonas – „DATA“.</li> </ul>



Temperatūros jutiklius „DS18B20“	-	<ul style="list-style-type: none"> <li>- Darbinė įtampa: 4,5 – 24 V DC.</li> <li>- Darbinė srovė: &lt; 15 mA.</li> <li>- Atitinka „RoHS“ standartą.</li> <li>- Vandens srauto diapazonas: 0-30 L/min.</li> </ul>
Mikrovaldiklis „ESP-WROOM-32“	-	<ul style="list-style-type: none"> <li>- Įvestis nuo 5V (su „microUSB“).</li> <li>- „Flash“ atmintis: 4 MB.</li> <li>- Įmontuotas „Wi-Fi“ modulis.</li> <li>- Įmontuotas „Bluetooth“ daviklis.</li> <li>- Darbinė temperatūra: -40°C - +125°C.</li> <li>- 2 mygtukai – įgalinti ir įjungti.</li> </ul>
Serveris / Mašininio mokymo serveris	<ul style="list-style-type: none"> <li>- 1 CPU.</li> <li>- 2 RAM.</li> </ul>	<ul style="list-style-type: none"> <li>- 1 CPU.</li> <li>- 2 RAM.</li> </ul>
„Firebase“ paslauga	-	<ul style="list-style-type: none"> <li>- Aktyvuota „Firebase Cloud Messaging“ paslauga.</li> <li>- Aktyvuota „Firebase Authentication“ paslauga.</li> <li>- Aktyvuota „Firebase Cloud Storage“ paslauga.</li> </ul>
Programėlė	<ul style="list-style-type: none"> <li>- „Android“ 5.0 ar naujesnė operacinė sistema.</li> </ul>	-

### 2.2.7. Kokybė

Esamos sistemos architektūra leidžia tiek lengvai prijungti kitų tipų jutiklių prie mikrovaldiklio, tiek sinchronizuoti skirtingų tipų mikrovaldiklių su serveriu. Tai ateityje leis nesunkiai įmontuoti, pavyzdžiui, oro taršos ar drėgmės matavimo daviklius prie sistemos.

Serverio ir programėlės programinio kodo suskirstymas į paketus padidina programinio rašymo efektyvumą bei kokybę, leidžia atskirti tam tikrų klasių priklausomybę, o ateityje leis daryti pakitimus bei atnaujinimus be jokių sunkumų.

Duomenų pernašai naudojant „HTTPS“ protokolą yra užtikrinamas duomenų saugumas, kadangi duomenys bus užšifruoti.

Kadangi duomenys yra saugomi išorinėje duomenų bazėje, pateikta architektūra leis nesunkiai integruoti skirtingų operacinių sistemų, pavyzdžiui, „Windows“ ar „iOS“, veikimą prie sistemos. Taip bus galima stebėti visą reikalingą informaciją ir gauti išpėjamuosius pranešimus ne tik „Android“ programėlėje.

## 2.3. Testavimo medžiaga

### 2.3.1. Testavimo elementai

Sąrašas elementų, kurie turės būti ištestuojami:

- išmanioji programėlė, versija **1.0.0**;
- serveris ir nuotėkio aptikimo algoritmas, versija **1.0.0**;
- mikrovaldiklis, versija **1.0**.

## 2.3.2. Programinės įrangos rizikos klausimai

Galimos šio projekto rizikos ir kaip į jas turėtų būti atsižvelgta:

- lokali duomenų bazė atlieka svarbų vaidmenį užtikrinant duomenų integralumą, todėl duomenų saugojimas joje turi būti gerai patikrintas;
- serverio duomenų bazės apsauga bei prieiga privalo būti aprašyta bei verifikuota;
- įspėjamojo signalo išsiuntimas naudotojui – svarbu užtikrinti, kad pranešimas bus išsaugotas eilėje, kai naudotojas neturės interneto, ir bus išsiųstas internetui atsiradus;
- sugedus ar kitaip sutrikus sklandžiam mikrovaldiklio veikimui naudotojas turi būti apie tai informuotas.

## 2.3.3. Testuotinos funkcijos

Sąrašas funkcijų, kurios bus atliekamos testavimo metu iš naudotojo perspektyvos, kartu su rizikos lygiu:

- prisijungimas bei atsijungimas (**aukštas**);
- jutiklių atvaizdavimas bei realaus laiko duomenų atnaujinimas (**aukštas**);
- temperatūros bei suvartoto vandens jutiklių istorinių duomenų atvaizdavimas (**vidutinis**);
- įspėjamojo signalo gavimas, kai jutiklius pasiekia kritinę reikšmę, yra aptinkamas vandens nuotėkis arba mikrovaldiklis tampa neveiklus (**aukštas**);
- įspėjamųjų pranešimų peržiūrėjimas ir trynimasis (**vidutinis**);
- vandens sunaudojimo ataskaitų peržiūrėjimas ir trynimasis (**aukštas**);
- matavimo vienetų nustatymo pakeitimas ir įgalinimas (**vidutinis**);
- jutiklio duomenų išsiuntimas ir išsaugojimas (**aukštas**);
- nuotėkio aptikimo algoritmo (modelio) įvertinimas (**aukštas**);

## 2.3.4. Netestuotinos funkcijos

Sąrašas funkcijų, kurios nebus tiesiogiai testuojamos, tačiau gali būti kitų testų netiesioginis rezultatas:

- lokalių duomenų eksportas į failą – tai yra žemos rizikos funkcija, kurios veikimas nėra kritinė reikšmė sistemos veikimui bei funkcionavimui;
- atskiras duomenų išsiuntimo, gavimo bei išsaugojimo testavimas – šio testavimo rezultatas bus gautas atliekant įspėjamojo signalo gavimo testavimą.

## 2.3.5. Strategija

### 2.3.5.1. Testavimo lygiai

Išmanaus namo vandens ir temperatūros monitoringo ir nuotėkio aptikimo sistemos testavimo planą sudaro penki pagrindiniai lygiai, kurie bus vykdomi toliau pateikta tvarka:

1. **vienetų testai** – bus siekiama užtikrinti, kad duomenys yra apdorojami teisingai mobiliojoje aplikacijoje, serveryje bei mikrovaldiklyje;
2. **integraciniai testai** – leis užtikrinti, kad mobiliojoje aplikacijoje naršymo keliai yra teisingai, o serveryje – kad servisų klasės veikia tinkamai;

3. **pasiekiamumo testai** – leis užtikrinti, kad programėlės elementai atitinka neįgalių žmonių keliamus reikalavimus;
4. **saugumo testai** – bus siekiama įsitikinti, kad serverio išteklių gali būti prieinami tik tam tikram autorizuotam asmeniui;
5. **priėmimo testai** – galiausiai bus siekiama patvirtinti, kad sistema veikia taip, kaip ir užsakovas tikėjosi.

Visus testus atliks sistemos kūrėjas Lukas Kulikovas, o patvirtins projekto vadovas Šarūnas Packevičius.

Toliau bus išsamiai aprašoma kiekvieno testo testavimo strategija bei procedūros tokia eilės tvarka, kokia jie ir bus atliekami.

#### **2.3.5.1.1. Vienetų testavimas**

Vienetų testai apima visus tris sistemos artefaktus – mobiliąją aplikaciją, serverį bei mikrovaldiklį.

##### **Mobiliosios aplikacijos vienetų testai**

Mobiliosios aplikacijos vienetų testų metu bus testuojamos pagalbinės ir „ViewModel“ (toliau – „VM“) klasės. Pagalbinės klasės apima įvairių matavimo vienetų konvertavimą į kitus ar manipuliavimą „LiveData“ duomenimis. „VM“ klasių pagrindinis tikslas yra apdoroti bei paruošti duomenis atvaizdavimui. Visa vienetų testų dokumentacija bei duomenys privalo būti aprašyti bei pateikti elektronine forma.

Šiems testams sukurti bus pasitelkiami netikri objektai (angl. *mock objects*), kurie leis sukurti tam tikrą scenarijų ir taip ištestuoti kuo daugiau galimų atvejų. Šiam tikslui bus pasitelkta netikrų objektų kūrimo biblioteka „MockK“. Kadangi „VM“ klasių paskirtis yra be galo svarbi, bus siekiama padengti minimaliai 95 % šių klasių, o tam išmatuoti bus pasitelkiama „Android Studio“ kūrimo platformoje esantis įrankis „Code Coverage“. Visi vienetų testai bus aprašyti anksčiau minėtoje kūrimo platformoje.

Toliau **43 pav.** yra pateiktas vienetų testų pavyzdys, o **44 pav.** – vienetų testų rezultatų pavyzdys.

```

@Test
fun `Updates notification click event on long click notification`() {
    val context = mockk<Context>()
    val view = View(context)

    viewModel.onLongClickNotification(view, id: 0)
    assertNotNull(viewModel.notificationClickEvent.value)
}

@Test
fun `Deletes specified notification and hides loading dialog`() = runTest { this: TestScope
    viewModel.loadingDialog.observeForever { }
    viewModel.onClickDelete(id: 0)

    advanceUntilIdle()

    coVerify { notificationsService.delete(id: 0) }
    assertEquals( expected: false, viewModel.loadingDialog.value?.peekContent())
}

```

43 pav. Mobiliosios aplikacijos vienetų testų pavyzdys

✓ Test Results	7 sec 934 ms
✓ com.example.sensors_tracker.ui.NotificationsViewModelUnitTests	7 sec 934 ms
✓ Updates notification click event on long click notification	7 sec 284 ms
✓ Loads all notifications and updates mediator	287 ms
✓ Deletes all notifications and hides loading dialog	333 ms
✓ Deletes specified notification and hides loading dialog	30 ms

44 pav. Mobiliosios aplikacijos vienetų testų rezultatų pavyzdys

## Serverio vienetų testai

Serverio vienetų testų metu bus testuojamos servisų klasės. Šios klasės užtikrina užklauso priėmimą iš valdiklio bei duomenų koregavimą, įrašymą ir ištrynimą, todėl svarbu užtikrinti, kad duomenys bus manipuluojami teisingai.

Šiems testams sukurti bus taip pat pasitelkiami netikri objektai (angl. *mock objects*), kurie leis sukurti tam tikrą scenarijų ir taip ištestuoti kuo daugiau galimų atvejų. Kadangi servisų klasės atlieka labai svarbų vaidmenį, bus siekiama ištestuoti ne mažiau kaip 90 % visų galimų metodų. Išmatavimo rezultatams pamatyti ir fiksuoti bus naudojamas „IntelliJ“ kūrimo platformoje esantis „Code Coverage“ įrankis.

### 2.3.5.1.2. Integracinis testavimas

#### Mobiliosios aplikacijos integracinis testavimas

Mobiliosios aplikacijos integracinio testavimo metu bus siekiama ištestuoti visus programėlės naršymo kelius – ar programėlės naudotojui paspaudus tam tikrą mygtuką ar vietą ekrane jis bus perkeltas į atitinkamą langą.

Šiam testavimui bus panaudota automatinė testavimo priemonė „Espresso“. „Espresso API“ leidžia užtikrinti ir sąveikauti su programėle kaip kiekvienas naudotojas galėtų su ja sąveikauti ir tikėtis tam tikro rezultato.

**45 pav.** yra pateiktas integracinių testų pavyzdys.

```
@Test
fun testNavigationToSensorSettings() {
    onView(
        allOf(
            childAtPosition(childAtPosition(withId(R.id.recyclerView), position: 0), position: 1),
            isDisplayed()
        )
    ).perform(click())

    Thread.sleep( millis: 2000)

    onView(withId(R.id.sensorFragment)).check(matches(isDisplayed()))
    onView(withId(R.id.view_pager)).perform(swipeLeft())
    onView(withId(R.id.tab_layout)).check(matches(not(isDisplayed())))
}
```

**45 pav.** Mobiliosios aplikacijos integracinių testų pavyzdys

### Serverio integracinis testavimas

Serverio integracinio testavimo metu bus siekiama užtikrinti, kad valdiklių bei servisų klasės teisingai bendradarbiauja tarpusavyje esant skirtingoms sąlygoms – tam bus pilnai paleidžiama sistema savo kontekste ir mažiau dėmesio kreipiant į netikrus objektus.

Šiam testavimui bus panaudotas „JUnit“ karkasas bei „MockK“ duomenų imitavimo biblioteka.

#### 2.3.5.1.3. Pasiiekiamumo testavimas

Pasiiekiamumo testavimo metu bus tikrinama, ar mobiliąja programėle galės naudotis visi asmenys, net ir neįgalieji – tai yra ar tam tikri elementai, esantys lange, atitinka reikalavimus žmonėms, kurie turi negalią. Šie reikalavimai apima elementų spalvas, kontrastus ir jų dydžius.

Šiam testavimui bus panaudotas automatinis analizės įrankis „Accessibility Scanner“. Jisai nuskenuos ekraną ir pateiks galimus pasiūlymus programėlės patobulinimui. Šis įrankis bus panaudotas nuskenuoti visus programėlės egzistuojančius langus, taip užtikrinant, kad visi asmenys, net ir neįgalieji, galės naudotis programėle be jokių sunkumų.

#### 2.3.5.1.4. Saugumo testavimas

Saugumo testavimo metu bus testuojami visi serverio valdiklių metodai bandant pasiekti visus serverio taškus (angl. *endpoints*) ir užtikrinti, kad juos gali pasiekti tik tam tikrą teisę turintis personalas.

#### 2.3.5.1.5. Priėmimo testavimas

Priėmimo testavimo metu bus siekiama užtikrinti, kad sukurta sistema atitinka užsakovo poreikius. Vykdam šį testavimą bus peržiūrima specifikacija bei visas funkcionalumas. Radus tam tikrus

neatitikimus ar klaidas sistemoje privalės būti atitikti pakeitimai tol, kol sistema atitiks užsakovo poreikius.

Toliau nuo **46 lentelės** iki **58 lentelės** bus pateikti galimi priėmimo testavimo scenarijai visiems panaudos atvejams užtikrinant, kad sistema veikia taip, kaip ir tikėtasi:

**46 lentelė** Prisijungimo testavimo scenarijus

<b>Scenarijus</b>	Naudotojo prisijungimas prie mobiliosios programėlės.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono ir interneto ryšio.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspusti „Sign In“.</li> <li>3. Pasirinkti savo „Google“ naudotoją.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
	<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
	Naudotojas yra registruotas sistemoje.	Naudotojas prijungiamas prie sistemos, jam atvaizduojami jo jutikliai.
	Naudotojas nėra registruotas prie sistemos.	Naudotojui atvaizduojamas prisijungimo klaidos pranešimas.

**47 lentelė** Profilio informacijos keitimo testavimo scenarijus

<b>Scenarijus</b>	Naudotojo profilio informacijos keitimas.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono ir yra prisijungęs prie sistemos.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspusti profilio piktogramą.</li> <li>3. Pakeisti norimą informaciją.</li> <li>4. Paspusti išsaugoti.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
	<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
	Naudotojas įvedė reikalavimus atitinkančius duomenis.	Duomenys yra išsaugomi ir naudotojui yra pateikiamas sėkmės pranešimas.
	Naudotojas įvedė reikalavimus neatitinkančius duomenis.	Naudotojui yra pateikiamas klaidos pranešimas.

**48 lentelė** Jutiklių peržiūros testavimo scenarijus

<b>Scenarijus</b>	Naudotojo jutiklių peržiūra.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
	<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
	Naudotojas turi jam priskirtus jutiklius.	Naudotojui atvaizduojami jam priskirti jutikliai.
	Naudotojas neturi nei vieno priskirto jutiklio.	Naudotojui atvaizduojamas tuščias sąrašas.

**49 lentelė** Jutiklių filtravimo testavimo scenarijus

<b>Scenarijus</b>	Naudotojo jutiklių filtravimas.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspausti filtravimo piktogramą.</li> <li>3. Pasirinkti norimus filtrus.</li> <li>4. Paspausti filtruoti.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>	
Naudotojas turi du temperatūros ir vieną vandens jutiklių, o pasirenko filtruoti tik temperatūros jutiklius.	Naudotojui atvaizduojami tik temperatūros jutikliai.	
Naudotojas turi du temperatūros ir vieną vandens jutiklių, o pasirenko filtruoti tik vandens jutiklius.	Naudotojui atvaizduojami tik vandens jutikliai.	
Naudotojas turi du temperatūros ir vieną vandens jutiklių, kur tik vienas temperatūros jutiklius yra aktyvus, o pasirenko filtruoti tik aktyvius jutiklius.	Naudotojas yra atvaizduojamas vienas aktyvus temperatūros jutiklius.	
Naudotojas turi du temperatūros ir vieną vandens jutiklių, kur tik vienas temperatūros jutiklius yra aktyvus, o pasirenko filtruoti tik neaktyvius jutiklius.	Naudotojui yra atvaizduojami vienas neaktyvus temperatūros ir vienas neaktyvus vandens jutiklius.	
Naudotojas turi du temperatūros ir vieną vandens jutiklių, o pasirenko rikiuoti pagal tam tikrą kategoriją.	Naudotojui išrikiuojama pagal jo pasirinktą kategoriją (pagal pavadinimą ar paskutinį duomenų atnaujinimą ir didėjančia arba mažėjančia tvarka).	

**50 lentelė** Jutiklio peržiūrėjimo testavimo scenarijus

<b>Scenarijus</b>	Naudotojo jutiklio peržiūrėjimas.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Pasirinkti norimą jutiklių.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>	
Naudotojas turi bent vieną jutiklių.	Naudotojui atidaromas pasirinkto jutiklio istorinių duomenų langas.	

**51 lentelė** Jutiklio nustatymų keitimo testavimo scenarijus

<b>Scenarijus</b>	Naudotojo jutiklio nustatymų keitimas.	
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.	
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Pasirinkti norimą jutiklių.</li> <li>3. Pasirinkti nustatymų kategoriją.</li> <li>4. Pakeisti norimą nustatymą.</li> </ol>	
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>		
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>	

Naudotojas pasirinkęs temperatūros jutiklių ir pasirenka minimalias ir maksimalias jam norimas reikšmes.	Reikšmės išsaugomos duomenų bazėje ir naudotojui atvaizduojamas sėkmės pranešimas.
Naudotojas pasirinkęs vandens jutiklių ir įrašo norimą didžiausią galimą vandens tekėjimo trukmę.	Reikšmė išsaugoma duomenų bazėje ir naudotojui atvaizduojamas sėkmės pranešimas.
Naudotojas pasirinkęs vandens jutiklių, tačiau įrašo reikalavimus neatitinkančią reikšmę.	Naudotojui yra atvaizduojamas klaidos pranešimas.

**52 lentelė** Įspėjimų peržiūros testavimo scenarijus

<b>Scenarijus</b>	Naudotojo jutiklių įspėjimų peržiūra.
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspausti įspėjimų piktogramą.</li> </ol>
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
Naudotojas turi gautų įspėjimų.	Naudotojui atvaizduojami gauti įspėjimai.
Naudotojas neturi gautų įspėjimų.	Naudotojui atvaizduojamas tuščias įspėjimų sąrašas.

**53 lentelė** Nustatymų peržiūros testavimo scenarijus

<b>Scenarijus</b>	Programėlės nustatymų peržiūra.
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspausti nustatymų piktogramą.</li> </ol>
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
-	Naudotojui atvaizduojami visi nustatymai.

**54 lentelė** Matavimo sistemos keitimo testavimo scenarijus

<b>Scenarijus</b>	Programėlės matavimo sistemos keitimas.
<b>Prielaidos</b>	Naudotojas turi prieigą prie mobilaus telefono, interneto ryšio ir yra prisijungęs prie sistemos.
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Atsidaryti mobiliąją programėlę.</li> <li>2. Paspausti nustatymų piktogramą.</li> <li>3. Paspausti ant matavimo sistemos nustatymo.</li> <li>4. Pasirinkti norimą matavimo sistemą.</li> </ol>
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
Naudotojas yra pasirinkęs metrinę sistemą, o pasirenka imperinę.	Naudotojui duomenys yra atvaizduojami pagal imperinę sistemą.
Naudotojas yra pasirinkęs imperinę sistemą, o pasirenka metrinę.	Naudotojui duomenys yra atvaizduojami pagal metrinę sistemą.



Naudotojas yra pasirinkęs metrinę sistemą, o pasirenka sistemos, kai sistemos nėra metrinė.	Naudotojui duomenys yra atvaizduojami pagal imperinę sistemą.
Naudotojas yra pasirinkęs imperinę sistemą, o pasirenka sistemos, kai sistemos nėra imperinė.	Naudotojui duomenys yra atvaizduojami pagal metrinę sistemą.

**55 lentelė** Duomenų siuntimo testavimo scenarijus

<b>Scenarijus</b>	Jutiklio duomenų siuntimas į duomenų perdavimo platformą.
<b>Prielaidos</b>	-
<b>Žingsniai</b>	1. Kas 1 minutę (vandens jutikliui) arba kas 5 minutes (temperatūros jutikliui) duomenys išsiunčiami į jutikliui priskirtą tašką (angl. <i>endpoint</i> ).
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
Jutiklius prisijungęs prie interneto.	Duomenys išsiunčiami į „AWS IoT“ duomenų perdavimo brokerį.
Jutiklius nėra prisijungęs prie interneto.	Jutiklius prisijungia prie interneto ir išsiunčia duomenis.
Jutiklius nėra prisijungęs prie duomenų perdavimo brokerio.	Jutiklius prisijungia prie duomenų perdavimo brokerio ir išsiunčia duomenis.

**56 lentelė** Duomenų priėmimo testavimo scenarijus

<b>Scenarijus</b>	Duomenų priėmimas serveryje.
<b>Prielaidos</b>	Serveris yra aktyvus.
<b>Žingsniai</b>	1. Serveris užprenumeruoja visus jutiklių duomenų gavimo taškus. 2. Serveris gauna duomenis iš jutiklio.
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
-	Duomenys išsaugomi duomenų bazėje ir yra įvertinamas galimas vandens nuotėkis arba įspėjamasis signalas.

**57 lentelė** Galimo vandens nuotėkio įvertinimo testavimo scenarijus

<b>Scenarijus</b>	Vandens nuotėkio aptikimas pagal nustatymą algoritmą.
<b>Prielaidos</b>	Serveris yra aktyvus ir yra sukurtas vandens nuotėkio algoritmas.
<b>Žingsniai</b>	1. Serveris gauna duomenis iš jutiklio. 2. Serveris, panaudojant algoritmą ir duomenis, nustato, ar šiuo metu yra vandens nuotėkis.
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
Gauti duomenys nėra tikrojo vandens nuotėkio rezultatas.	-
Gautis duomenys yra vandens nuotėkio rezultatas.	Serveris išsiunčia naudotojui įspėjamąjį pranešimą apie galimą vandens nuotėkį.

**58 lentelė** Įspėjamojo signalo siuntimo testavimo scenarijus

<b>Scenarijus</b>	Įspėjamojo signalo siuntimas.
-------------------	-------------------------------

<b>Prielaidos</b>	Serveris yra aktyvus.
<b>Žingsniai</b>	<ol style="list-style-type: none"> <li>1. Serveris gauna duomenis iš jutiklio.</li> <li>2. Serveris patikrina, ar duomenys atitinka naudotojo nustatytus nustatymus.</li> </ol>
<b>Testų rezultatai priklausomai nuo pradinių duomenų</b>	
<b>Pradiniai duomenys</b>	<b>Tikėtinas rezultatas</b>
Duomenys neatitinka naudotojo nustatytų nustatymų.	Serveris išsiunčia naudotojui įspėjimą apie kritinę jutiklio reikšmę.
Duomenys atitinka naudotojo nustatytą reikšmę.	-

### 2.3.5.2. Testavimo įrankiai, aplinka ir resursai

59 lentelėje yra pateikti testavimo įrankiai, o 60 lentelėje – testavimo aplinkos.

59 lentelė Testavimo įrankiai ir aplinka

	<b>Mobilioji aplikacija</b>	<b>Serveris</b>	<b>Mikrovaldiklis</b>
<b>Aplinka</b>	„Android Studio“ kūrimo platforma.	„Intelij“ kodo redaktorius.	„Visual Studio Code“ kodo redaktorius.
<b>Kodo padengimo nustatymas</b>	„Android Studio Code Coverage“	„Intelij Code Coverage“	Integruotas kodo redaktoriaus funkcionalumas.
<b>Vienetų testai</b>	„JUnit“ karkasas ir „MockK“ duomenų imitavimo biblioteka.	„JUnit“ karkasas ir „MockK“ duomenų imitavimo biblioteka.	-
<b>Integraciniai testai</b>	„Espresso“ automatinė testavimo priemonė ir „Android UI Automator“ karkasas.	„JUnit“ karkasas ir „MockK“ duomenų imitavimo biblioteka.	-
<b>Pasiekiamumo</b>	Automatinis analizės įrankis „Accessibility Scanner“.	-	-
<b>Saugumo testai</b>	-	„JUnit“ karkasas, „MockK“ duomenų imitavimo ir „ch4mpy“ prieigos teisių imitavimo biblioteka.	-
<b>Priėmimo testavimas</b>	+	+	+

60 lentelė Testavimo aplinkos

<b>Įrenginio pavadinimas</b>	<b>Procesorius</b>	<b>Darbinė atmintis</b>	<b>Vaizdo plokštė</b>	<b>Išorinė atmintis</b>	<b>Operacinė sistema</b>
Kompiuteris (HD1-DRT2)	AMD Athlon II X4 635 2.90 GHz	8 GB	NVIDIA GeForce GT 530	400 GB SSD 1TB HDD	Windows 10
Telefonas Samsung Galaxy A3 (2017)	Octa-core 1.6 GHz Cortex-A53	2 GB	Mali-T830 MP1	16 GB	Android 8.0 (Oreo)

Telefonas Sony Xperia Z3	Quad-core 2.5 GHz Krait 400	3 GB	Adreno 330	16 GB	Android 6.0 (Marshmallow)
Mikrovaldiklis ESP-WROOM-32	Dual-core Tensilica LX6 240 MHz	4 MB	-	-	-
Mikrovaldiklis ESP8266	Tensilica L106 32-bit RISC	32 KiB	-	-	-

### 2.3.5.3. Susitikimai

Sistemos kūrėjų komanda, kurią sudaro sistemos kūrėjas bei projekto vadovas, susitiks kas savaitę įvertinti atliktą darbą bei nustatyti galimas problemas kaip galima anksčiau. Visa informacija apie galimas problemas turi būti dokumentuota rašytine arba elektronine forma. Taip pat galimi papildomi susitikimai iškilus tam tikroms kritinėms problemoms.

### 2.3.5.4. Priemonės ir rodikliai (metrikos)

Sąrašas elementų, kurie bus surenkami iš vienetų testų bei integracinių testų, dokumentuojami bei pateikiami tolimesniam projekto tobulinimui:

- artefakto defektas bei rizikos lygis;
- defekto kilmė (programinis kodas arba naudotojo sąsaja);
- laikas, praleistas tvarkant defektą.

### 2.3.6. Elemento patenkinimo / nepatenkinimo kriterijai

61 lentelėje yra pateikti kiekvieno testavimo kiekvienam sistemos artefaktui patenkinimo kriterijai.

61 lentelė Elemento patenkinimo kriterijai

	Mobilioji aplikacija	Serveris	Mikrovaldiklis
<b>Vienetų testai</b>	Tiek „ViewModel“, tiek pagalbinės klasės turi būti padengtos bent 90 %.	Servisų klasės turi būti padengtos bent 90 %.	-
<b>Integraciniai testai</b>	Visi navigavimo keliai turi būti padengti 100 %.	Visi metodai tarp valdiklių ir servisų klasių turi būti padengti 100 %.	-
<b>Pasiekiamumo</b>	„Accessibility Scanner“ įrankio rezultatai visuose languose bus teigiami.	-	-
<b>Saugumo testai</b>	-	Valdiklių klasės turi būti padengtos 100 % (autorizavimo metu).	-
<b>Priėmimo testavimas</b>	Užsakovas patvirtina, kad sistema atitinka jo lūkesčius ir veikia taip, kaip ir tikėjosi.		

### 2.3.7. Testavimo rezultatai

Kaip šio testavimo rezultatas bus pateikta tokia informacija:

- testavimo plano dokumentas;
- testavimo atvejai;
- įrankiai ir jų išvestys;
- testavimo žinutės bei gautos reikšmės ar ataskaitos;
- tam tikro atrasto defekto aprašymas.

Testavimo metu gautos padengimo reikšmės bus saugomos „Microsoft Excel“ faile, kur bus aprašoma testo pavadinimas, padengimas, laikas, atrastos klaidos, įėjimo bei išėjimo duomenys.

### 2.3.8. Aplinkos poreikiai

Toliau bus pateikti šio testavimo plano specialūs reikalavimai:

- privalo būti pateikti testavimo duomenys kiekvienam testavimo scenarijui pagal panaudos atvejį;
- testavimo metu privalu panaudoti įvairių API lygių „Android“ emuliatorius ar realius įrenginius užtikrinant, kad programėlė veiks įvairiuose mobiliuose įrenginiuose.

### 2.3.9. Personalo ir mokymo poreikiai

Yra numatyta, kad testavimo eigą atliks bei prižiūrės sistemos kūrėjas kartu su projekto vadovu. Visą sistemos testavimą nuo pradžios iki pabaigos atliks vienintelis sistemos kūrėjas Lukas Kulikovas, o papildomų mokymų atlikti nereikės.

### 2.3.10. Atsakomybė

Už testų aprašymo bei įgyvendinimo kokybę, testų strategijos parinkimą, reikalingų elementų panaudojimą, testavimo reikiamus mokymus bei atitinkamus sprendimus yra atsakingas sistemos kūrėjas.

### 2.3.11. Testavimo rezultatai ir išvados

62 lentelė Testavimo rezultatai

	Mobilioji aplikacija	Serveris	Mikrovaldiklis
<b>Vienetų testai</b>	Parašyta <b>75 testai</b> ir padengta <b>95 %</b> „ViewModel“ bei <b>100 %</b> pagalbinių klasių.	Parašyta <b>87 testai</b> ir padengta <b>100 %</b> visų valdymo bei servisų klasių.	-
<b>Integraciniai testai</b>	Parašyta <b>6 testai</b> ir padengti <b>visi navigacijos keliai</b> .	Parašyta <b>60 testų</b> ir padengta <b>87 %</b> servisų klasių.	-
<b>Pasiekiamumo testai</b>	Visuose languose panaudojus „Accessibility Scanner“ įrankį ir sutvarkius gautus pasiūlymus įrankis grąžina, kad langai yra tinkamai paruošti naudotis neįgaliesiems.	-	-
<b>Saugumo testai</b>	-	Parašyta <b>40 testų</b> ir padengta <b>100 %</b> valdymo klasių.	-
<b>Pasiekiamumo testai</b>	Visi sistemos panaudojimo atvejai buvo įvertinti teigiamai tiek su užsakovu, tiek su testuotojais ir sistema įvertinta kaip atitinkanti visus užsakovo lūkesčius.		

### **2.3.12. Testavimo išvados:**

1. Užtikrinant stabilų visos sistemos darbą buvo nuspręsta testavimo metu apimti visus sistemos artefaktus: mobiliąją programėlę, serverį bei jutiklius.
2. Siekiant neprikaištingo visų sistemos artefaktų veikimo, buvo nuspręsta atlikti vienetų, integracinius, pasiekiamumo, saugumo bei priėmimo testus.
3. Išanalizavus programinės įrangos testavimo įrankius, buvo nuspręsta naudoti „JUnit“ karkasą, „MockK“ duomenų imitavimo biblioteką, „Espresso“ automatinę testavimo priemonę, „Android UI Automator“ ir „Unity“ karkasą, „ch4mpu“ prieigos teisių imitavimo biblioteką bei automatinį analizės įrankį „Accessibility Scanner“.

### **3. Tiriamoji dalis**

Šiame skyriuje pateikiama projekto kokybės vertinimo proceso procedūrų tyrimas bei išvados.

#### **Apklausų anketos.**

Kiekybiniam ir kokybiniam projekto įvertinimui bus pasitelkiami (1) ekspertai ir (2) privačių namų savininkai, kadangi bus vykdomas kokybinis ir kiekybinis vertinimas, vertinant sistemos efektyvumą bei funkcionalumą. Į eksperto poziciją yra įtraukiamas projekto vadovas, o namų savininkų pasirinkimas yra parenkamas atsižvelgiant į jų amžių bei turimą kompetenciją naudotis išmaniuoju įrenginiu.

#### **Duomenų surinkimas**

Kokybinio bei kiekybinio vertinimo rezultatų gavimas bus atliekamas pilnai sukūrus sistemą. Iš viso suplanuota surinkti dviejų tipų klausimynų rezultatus – ekspertų bei privačių namų savininkų klausimyno duomenis.

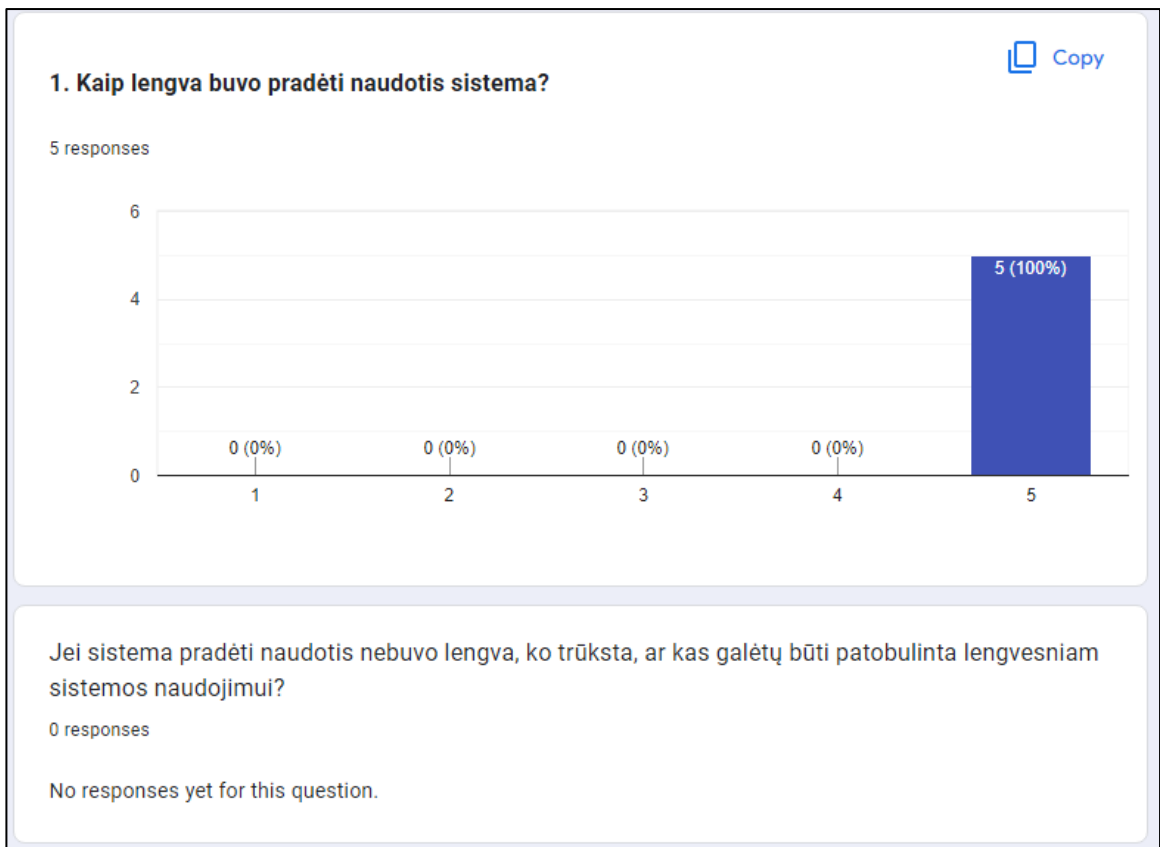
#### **Apklausa**

Prieš pateikiant klausimyną ekspertams bei pastatų savininkams bus pristatyta ši išmanaus namo vandens ir temperatūros monitoringo ir nuotėkio aptikimo sistema. Taip pat, sistema bus įdiegta vertintojų namuose su galimybe susipažinti su sistemos veikimu bei prietaisais. Tada, ekspertams bei naudotojams išbandžius sistemą bei apibūdinus sistemos vertinimo kriterijus, vertintojams bus pateikti klausimynai.

Klausimynai ekspertams bei vartotojams yra pateikiami prieduose atitinkamai **1 priedas** ir **2 priedas** prieduose.

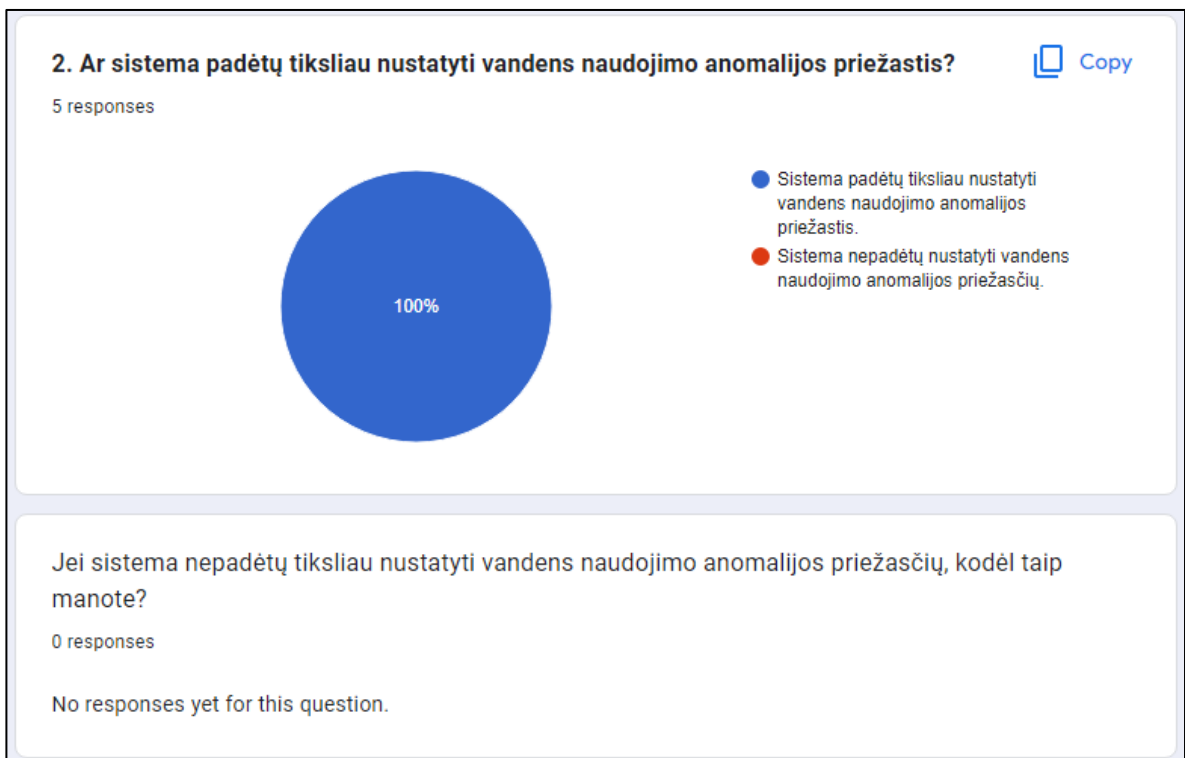
#### **3.1. Kiekybinis tyrimas**

Atlikus bendrą naudotojų apklausą, toliau yra apibendrinami gauti apklausų rezultatai:



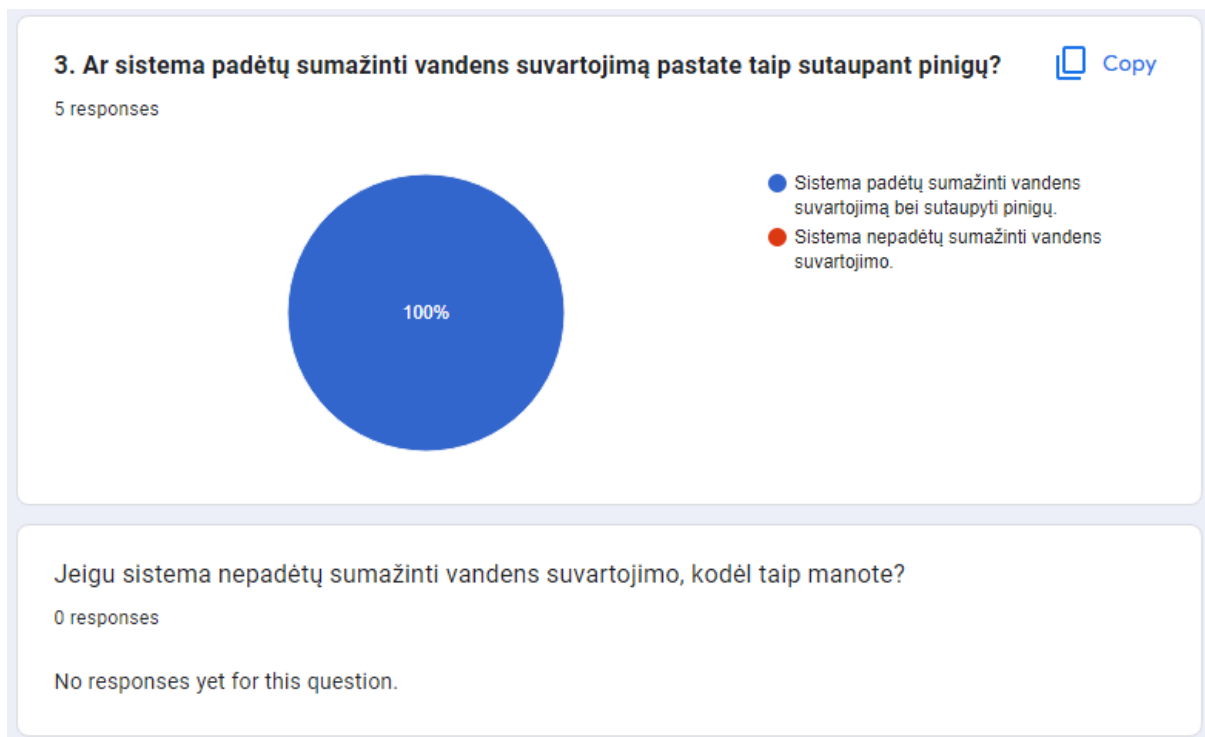
**46 pav.** Sistemos lengvumo naudojimo vertinimas

Visi penki apklausti naudotojai įvertino, kad sistema naudotis buvo lengva ir nereikėjo jokių papildomų mokymų ja naudotis.



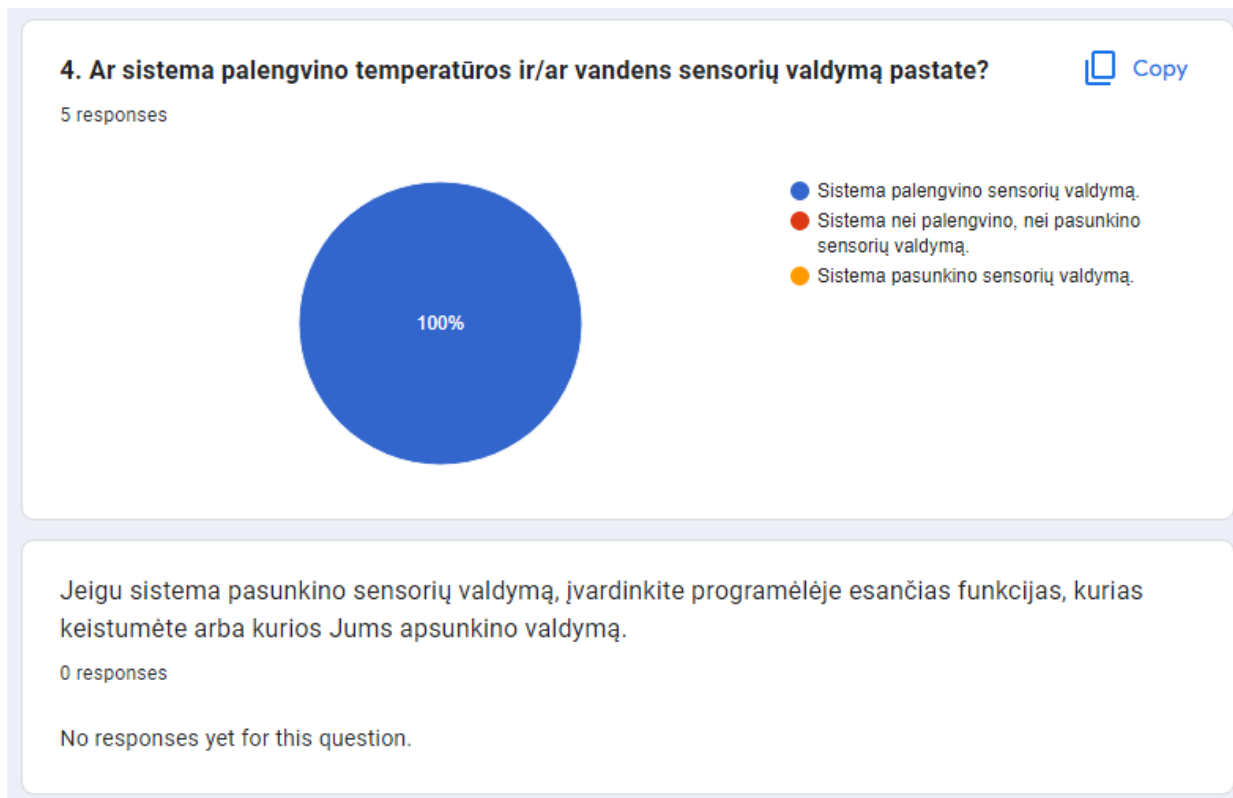
**47 pav.** Sistemos padėjimo įvertinti anomalijos priežastis vertinimas

Visi penki apklausti naudotojai įvertino, kad sistema tikrai padėtų tiksliau nustatyti vandens naudojimo anomalijos priežastis.



**48 pav.** Sistemos padėjimo sumažinti vandens suvartojimą pastate vertinimas

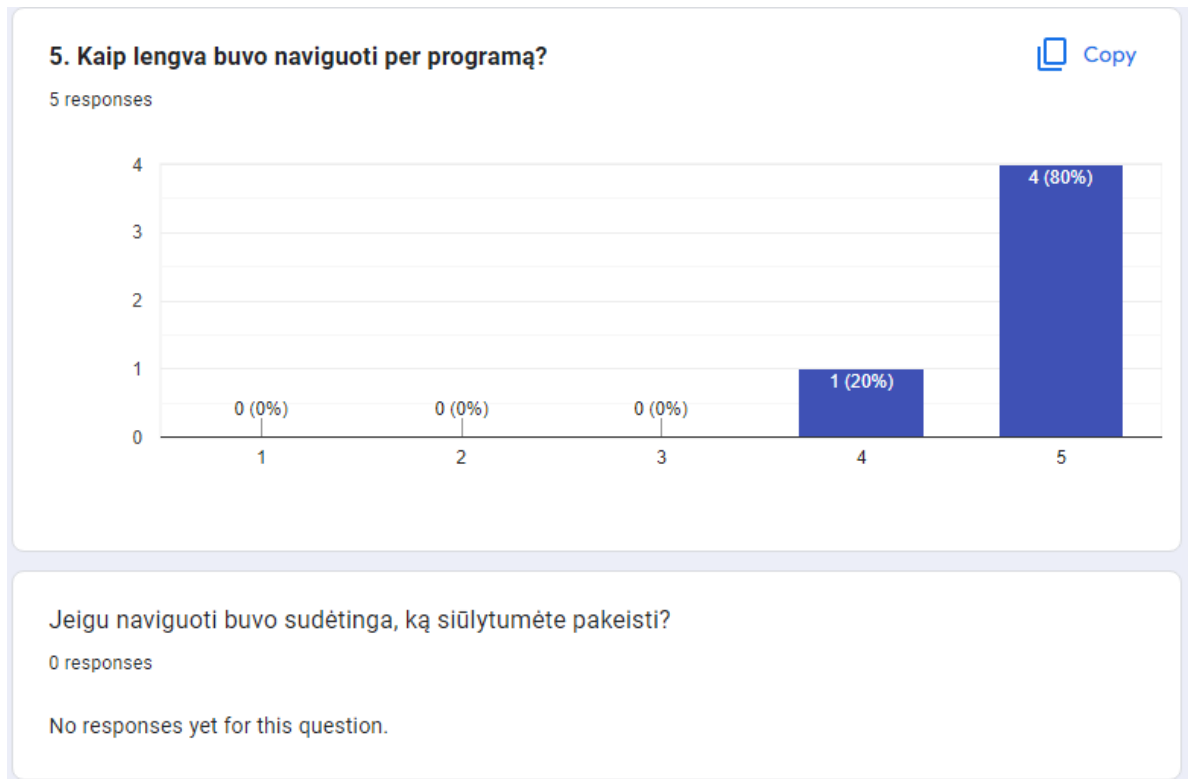
Visi penki apklausti naudotojai įvertino, kad sistema padėtų sumažinti vandens suvartojimą pastate ir taip galėtų sutaupyti pinigų.





**49 pav.** Sistemos palengvinimo jutiklių valdymo pastate vertinimas

Visi penki apklausti naudotojai įvertino, kad sistema palengvino temperatūros ir/ar vandens jutiklių valdymą pastate.



**50 pav.** Sistemos navigavimo lengvumo vertinimas

Visi penki apklausti naudotojai pakankamai gerai įvertino sistemos navigavimą ir kad jiems buvo nesudėtinga orientuotis programėlėje.



**51 pav.** Sistemos išbaigtumo vertinimas

Didžioji dalis naudotojų įvertino, kad sistemos funkcionalumas tenkina jų lūkesčius ir jokio papildomo funkcionalumo nereikia, o vienas žmogus išreiškė norą turėti papildomą galimybę redaguoti jutiklių pavadinimą.



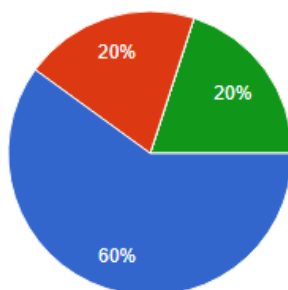
**52 pav.** Sistemos veikimo vertinimas

Visi penki apklausti naudotojai įvertino, kad sistema veikė sklandžiai ir greitai, nebuvo išsijungusi ar kitaip sutrikusi.

**8. Ar programėlės prenumeratos kaina (privatiems - 4.99 €/mėn., verslui - 19.99 €/mėn.) yra tinkama ir taptumėte/liktumėte mūsų klientu ir toliau?**



5 responses



- Programėlės kaina yra adekvati ir tikrai tapčiau/likčiau klientu.
- Programėlės kaina yra šiek tiek per didelė, tačiau klientu tapčiau/likčiau.
- Programėlės kaina yra per didelė, klientu tikrai netapčiau/nelikčiau.
- Programėlės apmokestinimas neatrodo tinkamas.

**Jeigu programėlės apmokestinimas neatrodo tinkamas, kaip siūlytumėte jį pakeisti?**

1 response

Būtų gerai tarkim sumokėti 50 eur. visam laikui ir nereiktų mokėti kas mėnesį.

**53 pav.** Sistemos kainos vertinimas

Trys naudotojai įvertino, kad sistemos kaina yra adekvati, vienas naudotojas – šiek tiek per didelė, o kitas – norėtų sumokėti vienintelę sumą visam laikui, kad nereiktų mokėti pinigų kas mėnesį.

### 3.2. Kokybinis tyrimas

Toliau bus pateiktas galutinis sistemos kokybės įvertinimas, kuris yra išskirstomas į tris kategorijas: mobiliosios aplikacijos, serverio bei valdiklio. Šis tyrimas atspindi eksperto įvertinimo bei testavimo gautus rezultatus.

Mobiliajai aplikacija įvertinti bus naudojami tokie parametrai:

Nr.	Parametras	Aprašymas
1.	Funkcionalumas	Sistemoje esančių funkcijų gausa.
2.	Išplečiamumas	Kaip nesunkiai galima įdėti naują funkcionalumą į sistemą.
3.	Panaudojamumas	Paprastumas išmokti naudotis sistema.
4.	Patvarumas	Sistemos atsilaikymas nuo sukeltų klaidų.
5.	Patikimumas	Sistemos atsparumas klaidoms.
6.	Susiejamumas	Integracijos su kita sistema lengvumas.

Serveriui įvertinti bus naudojami tokie parametrai:

Nr.	Parametras	Aprašymas
-----	------------	-----------

1.	Išplečiamumas	Kaip nesunkiai galima įdėti naują funkcionalumą į sistemą.
2.	Patvarumas	Sistemos atsilaikymas nuo sukeltų klaidų.
3.	Patikimumas	Sistemos atsparumas klaidoms.
4.	Susiejamumas	Integracijos su kita sistema lengvumas.

Valdikliui aplikacija įvertinti bus naudojami tokie parametrai:

Nr.	Parametras	Aprašymas
1.	Išplečiamumas	Kaip nesunkiai galima įdėti naują funkcionalumą į sistemą.
2.	Patvarumas	Sistemos atsilaikymas nuo sukeltų klaidų.
3.	Patikimumas	Sistemos atsparumas klaidoms.
4.	Susiejamumas	Integracijos su kita sistema lengvumas.

Mobiliosios aplikacijos įvertinimas:

Nr.	Parametras	Įvertinimas	Paiškinimas
1.	Funkcionalumas	4.9/5	Didžioji dalis apklaustų naudotojų įvertinto sistemą kaip išbaigtą su pilnu funkcionalumu.
2.	Išplečiamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai praplėsti programinės įrangos funkcionalumą.
3.	Panaudojamumas	5/5	Visi penki apklausti naudotojai įvertinto, kad sistema buvo lengva pradėti naudoti.
4.	Patvarumas	5/5	Visi apklaustieji nurodė, kad sistema veikė nepriekaištingai ir greitai, nebuvo nutrūkęs jos sklandus veikimas.
5.	Patikimumas	5/5	Visi apklaustieji nurodė, kad sistema nenusidūrė su klaidomis.
6.	Susiejamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai pridėti naujų modulių.

Serverio įvertinimas:

Nr.	Parametras	Įvertinimas	Paiškinimas
1.	Išplečiamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai praplėsti programinės įrangos funkcionalumą.
2.	Patvarumas	96/100	Atlikti 187 testai aprėpė 96 procentus kodo įrodant, kad ji yra patvari.
3.	Patikimumas	96/100	Atlikti 187 testai aprėpė 96 procentus kodo įrodant, kad ji yra atspari klaidoms.
4.	Susiejamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai pridėti naujų modulių.

Valdiklio:

Nr.	Parametras	Įvertinimas	Paiškinimas
1.	Išplečiamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai praplėsti programinės įrangos funkcionalumą.
2.	Patvarumas	100 <sup>2</sup> /100	Sistema veikė pas naudotoją (testuotoją) 3 mėnesius, o testuotojas nenurodė, kad valdiklis būtų susidūręs su klaidomis.
3.	Patikimumas	100 <sup>3</sup> /100	Sistema veikė pas naudotoją (testuotoją) 3 mėnesius, o testuotojas nenurodė, kad valdiklis sugestų.
4.	Susiejamumas	100/100	Ekspertinis vertinimas parodė, kad sistemos struktūra ateityje leis nesunkiai pridėti naujų modulių.

---

<sup>2</sup> Valdiklio patvarumas buvo įvertintas atsižvelgiant į naudotojo išsakytus rezultatus.

<sup>3</sup> Valdiklio patikimumas buvo įvertintas atsižvelgiant į naudotojo išsakytus rezultatus.

## 4. Eksperimentinė dalis

Šiame skyriuje yra pateikiamas įvairių mašininio mokymo algoritmų palyginimas, siekiant išsiaiškinti, kuris algoritmas efektyviausiai gali nustatyti anomalinį vandens naudojimą. Tam pasiekti yra panaudojami skirtingi scenarijai bei parametrai, taip pat yra įvertinami svarbūs statistiniai parametrai. Visi naudoti neprižiūravimo mašininio mokymo modeliai CBLOF, COPOD, ECOD, HBOS, IF, KNN, LOF, OCSVM ir PCA yra viešai prieinami „GitHub“ platformoje<sup>4</sup>. Taip pat naudotas dalinai prižiūravimo mašininio mokymo modelis SSDO, kuris yra irgi yra viešai prieinamas „GitHub“ platformoje<sup>5</sup>.

### 4.1. Duomenys

Modeliams įvertinti yra naudojami išmaniųjų vandens skaitiklių duomenys, gauti iš DAIAD, kuriuos galima rasti „GitHub“ platformoje<sup>6</sup>. Duomenų rinkinyje pateikiami 92 Alikantės namų ūkių valandiniai matavimai. Kiekvienas pastato vandens matavimo laikotarpis pradedamas nuo 2016 m. kovo 1 d. ir baigiasi 2018 m. vasario 28 d. Vidutiniškai vienam namų ūkiui tenka 7108 matavimų, o iš viso duomenų kiekis yra sudarytas iš 653,954 įrašų. Iš visų duomenų rinkinyje esančių įrašų nuokrypiai (angl. *outliers*) sudaro 322 įrašus [24]. Kadangi šie nuokrypiai yra laikoma tipiniu naudotojo elgesiu, todėl modelio prognozė šiam nuokrypiui kaip anomalinė bus laikoma klaidinga.

Kadangi DAIAD tyrimų duomenyse yra tik valandinis suvartojimas, į duomenų rinkinį yra įtraukiami ir išmaniojo vandens jutiklio surinkti duomenys iš esamos sistemos kūrėjo namų. Šiuose duomenyse yra minutinės trukmės matavimai, prasidedantys 2022 m. rugpjūčio 13 d. ir pasibaigiantys 2023 m. vasario 26 d. Iš viso šiame duomenų rinkinyje yra 258,496 įrašai.

### 4.2. Duomenų ir scenarijų paruošimas

Norint įvertinti kiekvieną modelį, turi būti naudojami teisingi ir kuo tikroviškesni duomenys, todėl buvo laikomasi šio proceso:

1. Pirmajame etape duomenys buvo paruošiami. Tai apėmė trūkstumų reikšmių užpildymą nuliniu vandens suvartojimu ir dienų, kuriose vanduo nebuvo naudojamas, pašalinimą.
2. Antrajame etape programos vykdymo metu buvo sukuriami ir į duomenų rinkinį įtraukiami vandens nuotėkio scenarijai. Kiekvienas vertinimas buvo atliekamas 5 kartus ir kiekvieną kartą buvo įtraukiami nauji atsitiktiniai vandens nuotėkio scenarijai. Neprižiūravimo modelio atveju buvo parodoma, o iš dalies prižiūravimo – apmokoma 80 % duomenų, o testuojami paskutiniai 20 %. Į mokymo duomenis atsitiktinai buvo įtraukiamas anomalijų skaičius režiuose [0; 3], o į testavimo duomenis – režiuose [1; 3]. Kiekvienas scenarijus galėjo sutapti, apimti tuos pačius scenarijus ir tęstis tarp 5 ir 180 minučių. Visi galimi scenarijai yra pateikti **63 lentelėje**.

**63 lentelė.** Vandens nuotėkio scenarijai

Scenarijus	Mažiausias srauto greitis (ml/min)	Didžiausias srauto greitis (ml/min)
Lašantis vamzdis	33,1	35,9
Lašantis vandens kranas	539,8	573,2

<sup>4</sup> <https://github.com/yzhao062/pyod>

<sup>5</sup> <https://github.com/Vincent-Vercruyssen/anomatools>

<sup>6</sup> [https://github.com/DAIAD/data/blob/master/swm\\_trialA\\_clean.zip](https://github.com/DAIAD/data/blob/master/swm_trialA_clean.zip)

Nutrūkusi magistralė	3671,9	3899
Atidarytas vandens kranas	7343,7	7798
Prakiuręs klozetas	15434	16389
Sugedęs lauko purkštuvai	45898,1	48737,2

Visuose eksperimentuose yra naudojami skirtingi testavimo parametrai, kurie apima duomenų, slenkančio lango, dimensijų dydį ir skirtingų duomenų susietumo lygius. Jeigu eksperimento aprašymas nenurodo kitaip, visuose testuose yra naudojama paskutinės 90 dienų nuo duomenų surinkimo pabaigos, 3 valandų tiek lango pločio, tiek slenkančio lango dydis, 1 vandens suvartojimo vidurkio dimensija ir visi duomenys, nepriklausomai nuo konteksto.

### 4.3. Įverčiai

Kiekvienam modeliui įvertinti bus apskaičiuojamas tikslumas, atspėjamumas, F1-balas, AUC-ROC ir AUC-PR įverčiai bei ED-balas.

**Tikslumas** – tai metrika, kuria yra vertinama modelio teigiamų prognozių, kurios iš tikrųjų yra teisingos, dalis. Jos apibrėžtis yra tikslų teigiamų prognozių ir visų teigiamų prognozių santykis, kuris yra išreiškiamas formule:

$$\text{Tikslumas} = \frac{TP}{TP + FP}$$

čia TP – teisingų teigiamų prognozių skaičius (angl. *True Positive*),

FP – neteisingų teigiamų prognozių skaičius (angl. *False Negative*).

**Atspėjamumas** – tai rodiklis, pagal kurį yra vertinama, kiek faktinių teigiamų atvejų modelis gali nustatyti. Jis nustatomas bendrą teigiamų atvejų skaičių padalinant iš faktinių teigiamų prognozių dalies ir yra išreiškiamas formule:

$$\text{Jautrumas} = \frac{TP}{TP + FN}$$

čia FN – neteisingų neigiamų prognozių skaičius (angl. *False Negative*).

**F1-balas** – tai vertinimo metrika, kuri apibrėžiama kaip precizijos (P) ir jautrumo (R) harmoninis vidurkis, kuris yra išreiškiamas formule:

$$F_1 = \frac{2PR}{P + R}$$

**AUC-ROC** – tai statistinis rodiklis, naudojamas vertinant, kaip gerai veikia dvejetainiai klasifikavimo modeliai. AUC-ROC svyruoja nuo 0 iki 1, kur 1 reiškia nepriekaištingą veikimą, o 0,5 – atsitiktinį spėjimą.

**AUC-PR** – tai taip pat yra vertinimo metrika, naudojama įvertinti, kaip gerai veikia dvejetainiai klasifikavimo modeliai. AUC-PR daugiausia dėmesio skiria modelio precizijai ir jautrumui, o ne

tikrojo teigiamojo ir klaidingai teigiamo rezultatų rodikliams, kaip AUC-ROC. AUC-PER svyruoja nuo 0 iki 1, kur 1 reiškia puikius rezultatus, o 0,5 – ne geresnį rezultatą nei atsitiktinis spėjimas.

**ED-balas** – tai ankstyvo aptikimo įvertis, pagal kurią yra teigiamai skatinami aptikimai, esantys netoli nuotėkio pradžios laiko  $t_f$ , o atlygis mažėja, kai aptikimas tolsta. Aptikimo laikas apibrėžiamas kaip anksčiausias laiko žingsnis per gedimo lango trukmę, kai algoritmas užregistravo aptikimą. Mūsų bandymuose, kaip ir Vercryussen'o bandymuose, sėkmingas aptikimas registruojamas tada, jei algoritmas generuoja aptikimus, kurie išlieka bent 75 % laiko. Ankstyvojo aptikimo rezultatas apskaičiuojamas pirmiausiai nustatant pirmojo aptikimo apibrėžtame gedimo lange uždelsimą, kuris pateikiamas kaip  $x = t_d - t_f$ , ir tada šiam aptikimui taikant tokią sigmoidinę funkciją [19]:

$$\sigma(x) = \frac{2}{1 + e^{\left(\frac{\alpha}{T_w} * x\right)}}$$

čia  $\sigma$  – reikšmė, kuri apibrėžiama taip, kad  $\sigma(x) \approx 0$ , o  $\sigma(0) = 1$  bet kokioms  $\alpha < \infty$  reikšmėms,

$T_w$  – gedimo lango trukmė [19].

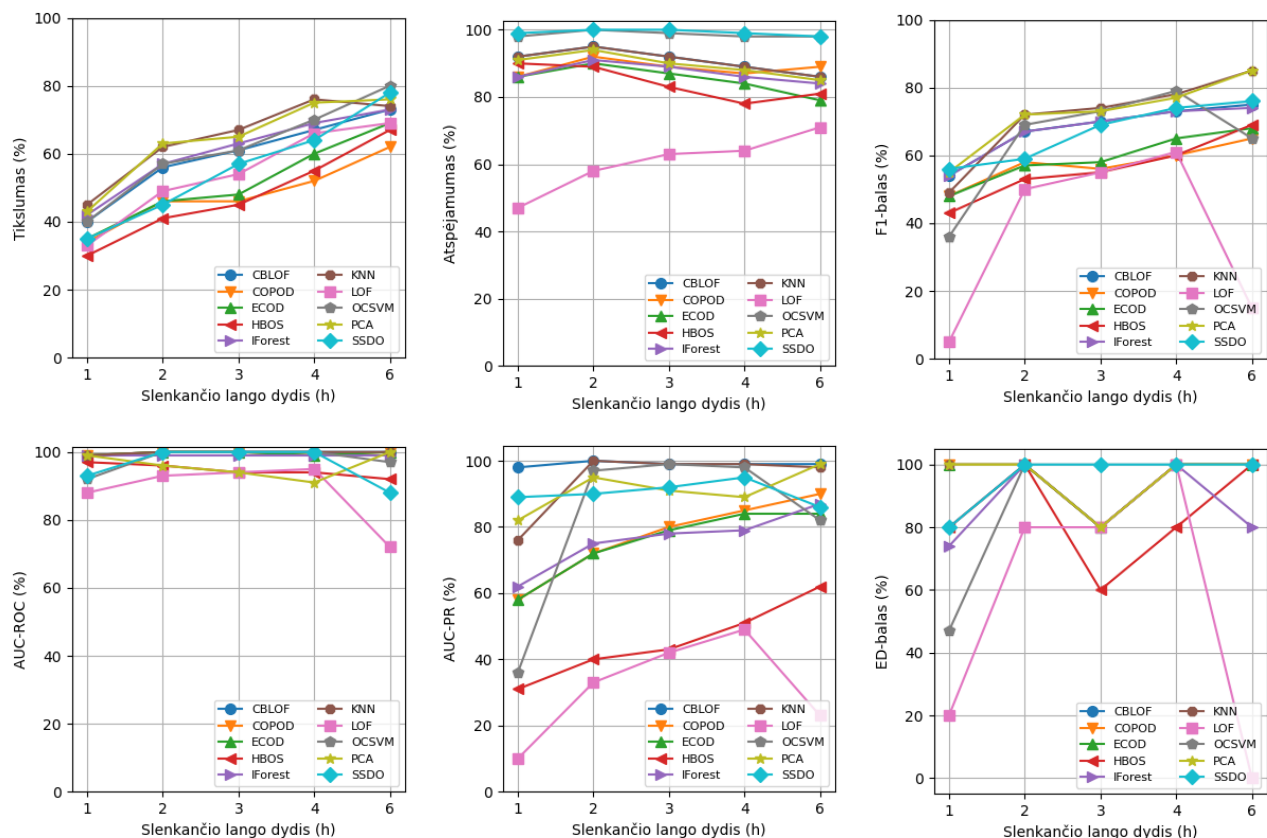
Bandymuose  $\sigma$  reikšmė buvo pasirinkta 6. Aptikimas, įvykęs už gedimų lango ribų, į rezultatą nėra įtraukiamas.

#### 4.4. Modelių palyginimas skirtingų slenkančių langų režiuose

Šio eksperimento metu buvo testuojamas modelių efektyvumas esant skirtingo slenkančio lango dydžiui. Pirmiausia modeliai buvo testuojami naudojant DAIAD duomenis (valandų režiuose), kuriuose lango plotis yra 3 valandos, o slenkančio lango dydis atitinkamai 1, 2, 3, 4 ir 6 valandos. Paskui modeliai buvo testuojami naudojant sistemos kūrėjo duomenimis (minučių režiuose), kuriuose lango plotis yra 3 valandos, o slenkančio lango dydis atitinkamai 1, 5 ir 10 minučių. Diagramose abscisės ašyje yra žymimas slenkančio lango dydis, o ordinatės – modelio įvertis pagal atitinkamai nurodytą kriterijų



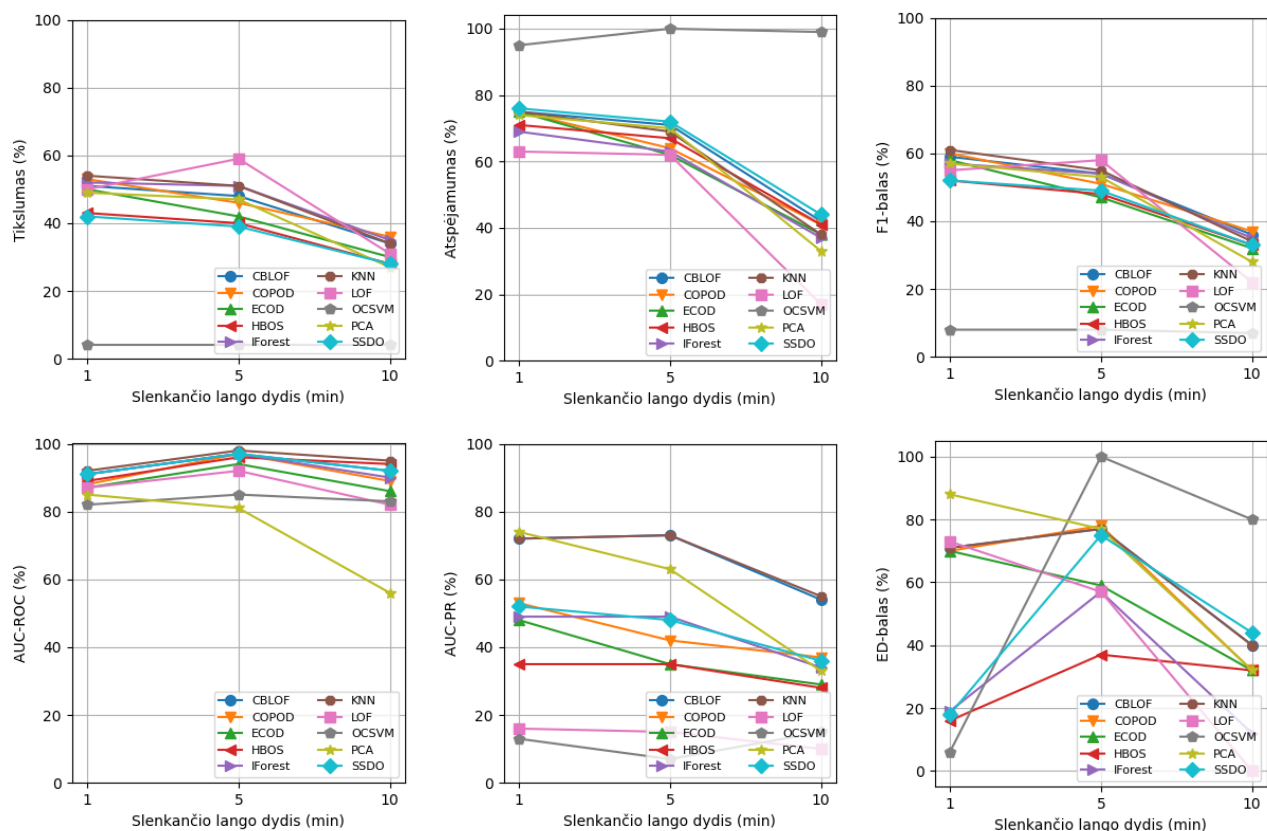
## Modelių palyginimas skirtingų valandų režiuose



54 pav. Modelių palyginimas skirtingų valandų režiuose

Skirtingų valandų režijų eksperimentas parodė, kad mažėjant slenkančio lango dydžiui nuo 6 h iki 1 h tikslumas suprastėjo dvigubai, tačiau atspėjamumas daugumoje modelių išliko tas pats. Taip pat žvelgiant į ED-balą dauguma modelių prastai sugebėjo anksti aptikti vandens nuotėkį esant 1 h slenkančiam langui. Dalinai apmokytas SSDO modelis sugebėjo pasireikšti geriau negu dauguma neapmokytų mašininio mokymo modelių.

## Modelių palyginimas skirtingų minučių režiuose



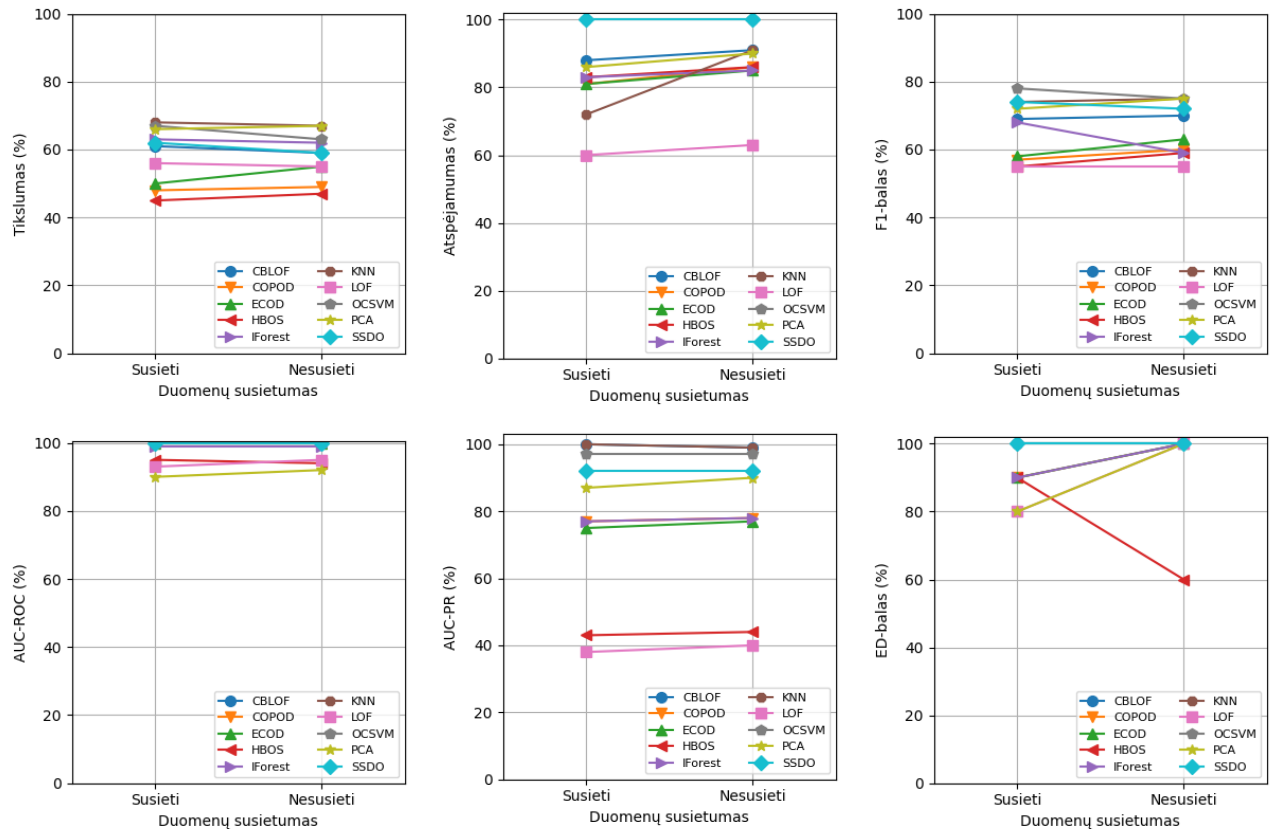
55 pav. Modelių palyginimas skirtingų minučių režiuose

Skirtingų minučių režijų eksperimentas parodė, kad mažėjant slenkančio lango dydžiui nuo 10 min. iki 5 min. tikslumas ir atspėjiamumas pagerėjo daug daugiau negu keičiant lango dydį nuo 5 min. iki 1 min. Dalinai apmokytas SSDO modelis nesugebėjo pateikti geresnio tikslumo lyginant su neapmokytais mašininio mokymo modeliais, tačiau pateikė beveik geriausią atspėjiamumą.

### 4.5. Modelių palyginimas naudojant skirtingus kontekstus

Šio eksperimento metu buvo testuojamas modelių efektyvumas esant skirtingam kontekstui. Diagramose nesusieti duomenys žymi tai, kad apmokymui ir testavimui buvo imami visi duomenys (nuo pirmadienio iki sekmadienio), o susieti – pirmiausia buvo imamos darbo dienos (nuo pirmadienio iki penktadienio), o tada savaitgalio dienos (šeštadienis ir sekmadienis), o gauti įverčių rezultatai (darbo ir savaitgalio dienų) buvo sudedami ir padalinami iš dviejų (vidurkis).

Diagramose abscisės ašyje yra žymima, ar rezultatas gautas apskaičiuavus susietus ar nesusietus duomenis, o ordinatės – modelio įvertis pagal atitinkamai nurodytą kriterijų.



56 pav. Modelių palyginimas skirtinguose kontekstuose

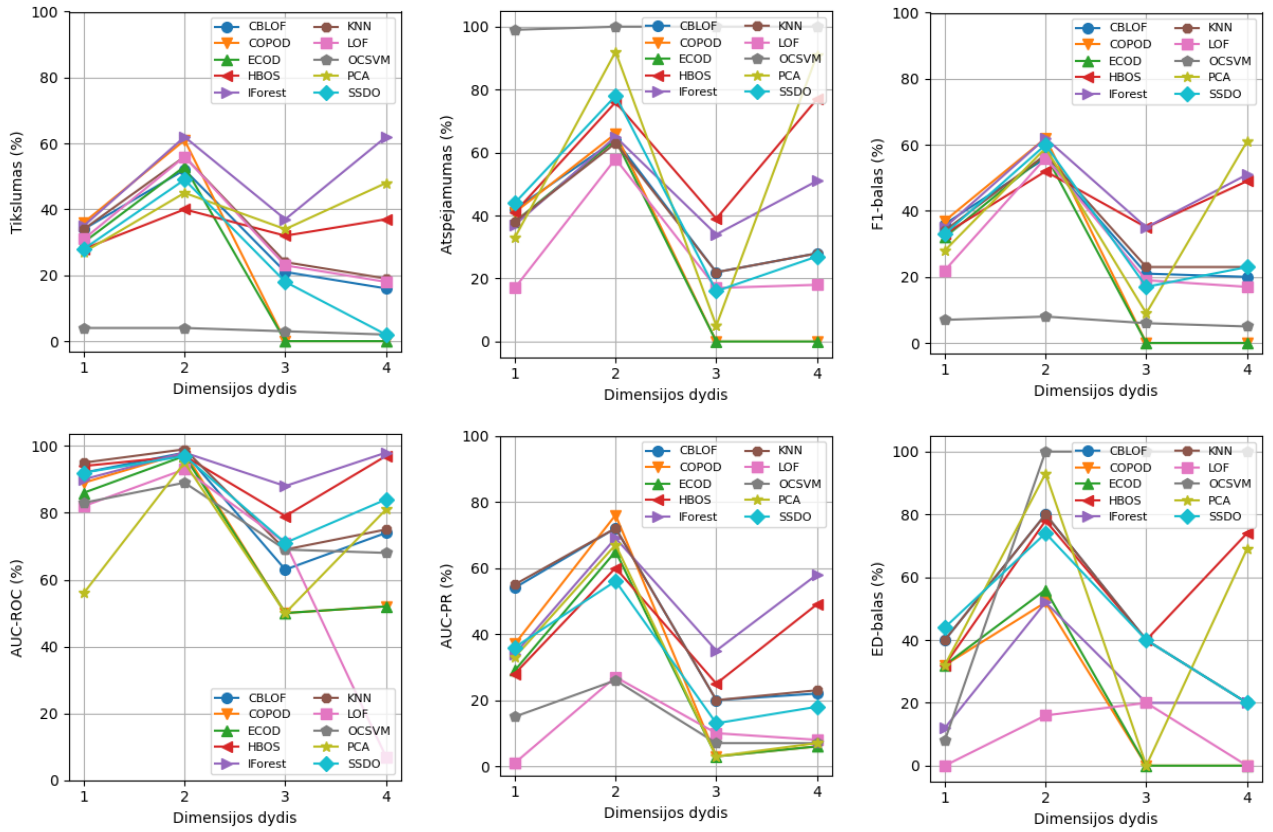
Skirtingų kontekstų eksperimentas parodė, kad naudojant susietus duomenis atspėjiamumas sumažėja, o tikslumas padidėja labai nedaug. Tačiau naudojant nesusietus duomenis dauguma modelių sugebėjo aptikti vandens nuotėkį daug greičiau, negu naudojant susietus.

#### 4.6. Modelių palyginimas naudojant skirtingas duomenų dimensijas

Šio eksperimento metu buvo testuojamas modelių efektyvumas esant skirtingoms duomenų dimensijoms. Diagramoje abscisės ašyje pažymėtos dimensijos yra išreiškiamos taip:

- viena dimensija – duomenų vidurkis;
- dvi dimensijos – duomenų vidurkis ir ilgiausias vandens bėgimo laikotarpis (minutėmis);
- trys dimensijos – duomenų vidurkis, minimaliausias ir maksimaliausias vandens suvartojimas trijų valandų laikotarpyje;
- keturios dimensijos – trečioji (aukščiau nurodyta) dimensija pridėdant ilgiausią vandens bėgimo laikotarpį (minutėmis).

Diagramos ordinatės ašyje yra žymimas modelio įvertis pagal atitinkamai nurodytą kriterijų.



57 pav. Modelių palyginimas skirtingose dimensijose

Skirtingų dimensijų eksperimentas parodė, kad naudojant dvi dimensijas (vidurkį ir ilgiausią vandens bėgimo laikotarpį) buvo pasiekti geriausi bendri rezultatai – modeliai sugebėjo pateikti geriausią tikslumą, atspėjamumą, o taip pat ir greičiausiai aptikti vandens nuotėkį.

## Išvados

1. Atlikus literatūros analizę buvo pastebėta, kad kiti autoriai mašininio mokymo modeliui vertinti pasitelkė tik tikslumo metriką ir neįvertino svarbiausių vertinimo reikšmių: precizijos, atspėjamumo, F1 bei ED balų, siekiant surasti efektyviausią mašininio mokymo modelį nustatant vandens nuotėkį.
2. Atlikus anomalinio vandens naudojimo rinkos analizę buvo pastebėta, kad rinkoje trūksta išmanių sprendimų, susijusių su vandens nuotėkio aptikimu, kadangi dauguma egzistuojančių sprendimų nėra „protingi“, nesuteikia galimybės stebėti vandens sunaudojimo ir dažnai montuojami po kiekvienu buitiniu įtaisu.
3. Sistemos kūrimas, remiantis „Agile“ metodologija, leido kurti sistemą per nedideles iteracijas, kurios leido greitai ir efektyviai atlikti reikalingus pakeitimus ir prisitaikyti prie galimai besikeičiančių reikalavimų. Visa tai leido sumažinti projekto nepasisekimo riziką, o taip pat įtraukti ir sistemos užsakovą į sistemos kūrimo procesą, jam suteikiant galimybę nuolatos sekti visą projekto eigą.
4. Atliktas sistemos kiekybinis vertinimas įrodė, kad sistema yra lengva naudotis, ji padėtų tiksliau nustatyti vandens naudojimo anomalijos priežastis ir sumažintų vandens suvartojimą pastate. Taip pat naudotojų įvertinimas parodė, kad sistema yra funkcionali, veikia greitai ir sklandžiai.
5. Atlikti sistemos testai ir kokybinis įvertinimas parodė, kad sistema yra visiškai realizuota ir atitinka aukštus įvertinimo kriterijus – projektas yra funkcionalus, lengvai išplečiamas, patvarus, patikimas bei lengvai integruojamas su kitomis sistemomis.
6. Atlikti eksperimentai skirtingų valandų režiuose parodė, kad kai sistema renka vandens suvartojimo duomenis kas valandą, dauguma modelių pateikia nepakitusių ir gerą atspėjamumą, nepriklausomai nuo slenkančio lango dydžio, tačiau tikslumas suprastėja dvigubai renkantis nuo 6 h iki 1 h slenkančio lango dydį. Taip pat dauguma modelių pateikė gerą ankstyvą nuotėkio aptikimą, nepriklausomai nuo lango dydžio. Vadinasi, siekiant nustatyti vandens nuotėkį, kai yra galimas tik valandinis suvartojimas, ir renkantis kuo greitesnį (valandinį) slenkantį langą, dauguma modelių sugebės nustatyti visas vandens anomalijas, tačiau dažnu atveju pateiks neteisingą aliarminį pranešimą.
7. Atlikti eksperimentai skirtingų minučių režiuose parodė, kad kai sistema renka vandens suvartojimo duomenis kas minutę, visi modeliai veikė daug geriau, kai buvo mažesnis slenkančio lango dydis tiek iš atspėjamumo, tiek iš tikslumo įvertinių. Iš rezultatų galima spręsti, kad naudojant PCA mašininio mokymo modelį ir naudojant minutinį slenkantį langą bus galima nustatyti vandens nuotėkį apie 78 % kartų, taip pat ir labai greitai nustatyti vandens anomalinio naudojimo pradžią ir laiku sustabdyti nepageidaujamas pasekmes.
8. Atlikti eksperimentai skirtingų duomenų susietumų lygiuose parodė, kad naudojant susietus ar nesusietus duomenis skirtumas nėra labai didelis, tačiau daugumoje atvejų naudojant nesusietus duomenis modeliai sugebėjo aptikti vandens nuotėkį daug greičiau, negu naudojant susietus. Vadinasi, sistemoje modeliui išmokyti, siekiant greičiau aptikti vandens nuotėkį, geriau naudoti nesusietus duomenis.
9. Atlikti eksperimentai naudojant skirtingus dimensijos dydžius parodė, kad naudojant dvi dimensijas (vandens suvartojimo vidurkį bei ilgiausią vandens bėgimo laikotarpį) buvo pasiekti geriausi rezultatai, o PCA modelis pateikė geriausius rezultatus – sugebėjo nustatyti vandens nuotėkį apie 95 % laiko ir taip efektyviai nustatyti anomalijos pradžią.

## Literatūros sąrašas

1. *10 Quick Facts about Water Leaks*. Interaktyvus. 2020. Prieiga per: <https://thedetectiongroup.com/10-quick-facts-about-water-leaks/>. [žiūrėta 2021-11-25].
2. *Fix a Leak Week*. Interaktyvus. 2017. Prieiga per: <https://www.epa.gov/watersense/fix-leak-week>. [žiūrėta 2021-11-25].
3. *Water Leak Detection & Facts - Flologic System Against Water Leak Damage*. Interaktyvus. Prieiga per: <https://www.flologic.com/water-leak-facts>. [žiūrėta 2021-11-27].
4. *Water Damage by the Numbers*. Interaktyvus. Prieiga per: <https://www.waterdamagedefense.com/pages/water-damage-by-the-numbers>. [žiūrėta 2021-11-25].
5. COELHO, João; GLÓRIA, André ir SEBASTIÃO, Pedro. *Precise Water Leak Detection using Machine Learning and Real-Time Sensor Data*, 2020. Prieiga per: <https://www.mdpi.com/2624-831X/1/2/26>.
6. FUENTES, Henry ir MAURICIO, David. Smart Water Consumption Measurement System for Houses using IoT and Cloud Computing. *Environmental Monitoring and Assessment*, 2020, t. 192, nr. 9. pp. 602. ISSN 1573-2959. Prieiga per: <https://doi.org/10.1007/s10661-020-08535-4>.
7. SARANGI, Anish K. *Smart Water Leakage and Theft Detection using IoT*. 2020. Prieiga per: <https://www.doi.org/10.1109/I4Tech48345.2020.9102701>.
8. MONI, Nayna A; SIGWENI, Boyce; MANGWALA, Mmoloki ir KOLOBE, Lone. *Water Leak Detection from Irrigation Pipelines in Botswana using Vibration Interpretation Technique*. 2019. Prieiga per: <https://www.doi.org/10.1109/AFRICON46755.2019.9133829>.
9. BOUDHAOUIA, Aida ir WIRA, Patrice. *Water Consumption Analysis for Real-Time Leakage Detection in the Context of a Smart Tertiary Building*. 2018. Prieiga per: <https://www.doi.org/10.1109/ICASS.2018.8651976>.
10. AMORA, Dennis J. A et al. *Design of a Household Consumption Based Water Leak Detection System Utilizing Machine Learning Algorithm*. 2022. Prieiga per: <https://www.doi.org/10.1109/IET-ICETA56553.2022.9971564>.
11. JI, Mingeun; YI, Gangman [ORCID: 0000-0003-4664-5325] ir JUNG, Jaehee [ORCID: 0000-0002-0932-3039]. *Central Prediction System for Time Series Comparison and Analysis of Water Usage Data*. 2020. Prieiga per: <https://www.doi.org/10.1109/ACCESS.2019.2963373>.
12. PATABENDINGE, Samitha; CARDELL, Rachel O ir WANG, Rui. Detection and Interpretation of Anomalous Water use for Non-Residential Customers. *Environmental Modelling & Software*, 2018, t. 100. pp. 291-301. DOI 10.1016/j.envsoft.2017.11.028. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S1364815217305285> ISSN 1364-8152.
13. ISMAIL, Heba; ELABYAD, Rawan ir DYAB Arwa. *Smart Residential Water Leak and Overuse Detection System using Machine Learning*. 2022. Prieiga per: <https://www.doi.org/10.1109/AICCSA56895.2022.10017508>.

14. WOOK, Lee C ir DO, Yoo G. Development of Leakage Detection Model and its Application for Water Distribution Networks using RNN-LSTM. *Sustainability*, 2021, t. 13, nr. 16. DOI 10.3390/su13169262. Prieiga per: <https://www.mdpi.com/2071-1050/13/16/9262> ISSN 2071-1050.
15. BLÁZQUEZ, Ane G; CONDE, Angel; MORI, Usue ir LOZANO, Jose A. Water Leak Detection using Self-Supervised Time Series Classification. *Information Sciences*, 2021, t. 574. pp. 528-541. DOI 10.1016/j.ins.2021.06.015. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S002002552100606X> ISSN 0020-0255.
16. WANG, Wenlu; CHEN, Pengfei; XU, Yibin ir HE, Zilong. *Active-MTSAD: Multivariate Time Series Anomaly Detection with Active Learning*. 2022. Prieiga per: <https://www.doi.org/10.1109/DSN53405.2022.00036>.
17. DAS, Shubhomoy et al. Discovering Anomalies by Incorporating Feedback from an Expert. *ACM Transactions on Knowledge Discovery Data*. 2020, t. 14, nr. 4. Prieiga per: <https://doi.org/10.1145/3396608>.
18. ZHU, Yanqiao ir YANG Kai [ORCID: 0000-0002-5983-198X]. *Tripartite Active Learning for Interactive Anomaly Discovery*. 2019. Prieiga per: <https://www.doi.org/10.1109/ACCESS.2019.2915388>
19. VINCENT, Vercruyssen et al. *Semi-Supervised Anomaly Detection with an Application to Water Analytics*. 2018. Prieiga per: <https://www.doi.org/10.1109/ICDM.2018.00068>.
20. ZESE, Riccardo [ORCID: 0000-0001-8352-6304]; BELLODI, Elena [ORCID: 0000-0002-3717-3779]; LUCIANI, Chiara ir ALVISI, Stefano [ORCID: 0000-0002-5690-2092]. *Neural Network Techniques for Detecting Intra-Domestic Water Leaks of Different Magnitude*. 2021. Prieiga per: <https://www.doi.org/10.1109/ACCESS.2021.3111113>.
21. FAN, Xudong; ZHANG, Xijin ir YU, Xiong. Machine Learning Model and Strategy for Fast and Accurate Detection of Leaks in Water Supply Network. *Journal of Infrastructure Preservation and Resilience*. 2021, t. 2, nr. 1. pp. 10. Prieiga per: <https://doi.org/10.1186/s43065-021-00021-6> ISSN 2662-2521.
22. GRIZHNEVICH, Alex. *IoT Architecture Explained: Building Blocks and how they Work*. Apr 1, 2018. Prieiga per: <https://www.scnsoft.com/blog/iot-architecture-in-a-nutshell-and-how-it-works>. [žiūrėta 2021-11-21]
23. DIAS, Pedro. *Using IoT to Detect Water Leakages with Powel*. 2016. Prieiga per: <https://microsoft.github.io/techcasestudies/iot/2016/11/29/Powel.html>. [žiūrėta 2021-11-21]
24. ATHANASIOU, Spiros et al. *Trials Evaluation and Social Experiment Results*. 2017. Prieiga per: [http://www.daiad.eu/wp-content/uploads/2017/11/D7.3\\_Trials\\_Evaluation\\_v1.0.pdf](http://www.daiad.eu/wp-content/uploads/2017/11/D7.3_Trials_Evaluation_v1.0.pdf). [žiūrėta 2023-02-26]

## Priedai

### 1 priedas Klausimynas ekspertams

1. **Ar sistema atitinka saugumo standartus – yra vykdomas teisingas vartotojų autentifikavimas ar/ir autorizavimas?**

Sistema atitinka saugumo standartus.

2. **Ar sistemos struktūra ateityje leis praplėsti programinės įrangos funkcionalumą ar naujų modulių kūrimą/pridėjimą?**

Sistemos struktūra leis praplėsti programinės įrangos funkcionalumą.

3. **Ar projekto dokumentacija yra aiški ir suprantama?**

Projekto dokumentacija yra aiški ir suprantama.

4. **Ar testavimo atvejai ir naudotos technologijos jiems atlikti atitinka naujausius standartus ir geriausias praktikas?**

Testavimo atvejai ir naudotos technologijos atitinka naujausius standartus.

5. **Ar sistema yra tolerantiška vartotojo klaidoms?**

Sistema yra tolerantiška vartotojo klaidoms.

6. **Ar programėlė tausoja įrenginio resursus ir neapsunkina jo veikimo?**

Sistema tausoja įrenginio resursus.

7. **Identifikuokite dalykus, kurie buvo atlikti nesėkmingai ar blogai, o jų veikimas turėtų būti pagerintas sėkmingam projekto veikimui.**

-



**8. Identifikuokite dalykus, kurie buvo atlikti sėkmingai ar gerai, o jų veikimas ateityje galėtų būti dar labiau patobulintas.**

Mašininio mokymo algoritmo pritaikymas anomaliniam vandens naudojimui nustatyti.

**9. Kaip vertinate projekto rizikos valdymą?**

Teigiamai.

**10. Kaip vertinate projekto sėkmę šiuolaikinėje rinkoje?**

Teigiamai.

## 2 priedas Klausimynas vartotojams

**1. Kaip lengva buvo pradėti naudotis sistema?**

**Naudotojo vertinimas** (pažymi numerį skalėj tarp 1 ir 5)

- Labai lengva**
- Lengva
- Šiek tiek sunku
- Sunku
- Labai sunku

**Komentaras** (privaloma, jeigu nebuvo lengva išmokti).

-

**2. Ar sistema padėtų tiksliau nustatyti vandens naudojimo anomalijos priežastis ir vietą?**

**Naudotojo vertinimas** (pažymėti „X“)

- Sistema padėtų tiksliau nustatyti vandens naudojimo anomalijos priežastis.**
- Sistema nepadėtų nustatyti vandens naudojimo anomalijos priežasčių.

**Komentaras** (privaloma, jeigu nepadėtų).

-

**3. Ar sistema padėtų sumažinti vandens suvartojimą pastate taip sutaupant pinigų?**

**Naudotojo vertinimas** (pažymėti „X“)

- Sistema padėtų sumažinti vandens suvartojimą bei sutaupyti pinigų.**
- Sistema nepadėtų sumažinti vandens suvartojimo.

**Komentaras** (privaloma, jeigu nepadėtų).

-

<b>4. Ar sistema palengvino temperatūros ir/ar vandens jutiklių valdymą name?</b>
<b>Naudotojo vertinimas</b> (pažymėti „X“)
<input type="checkbox"/> <b>Sistema palengvino jutiklių valdymą.</b> <input type="checkbox"/> Sistema nei palengvino nei pasunkino jutiklių valdymą. <input type="checkbox"/> Sistema pasunkino jutiklių valdymą.
<b>Komentaras</b> (privaloma, jeigu sistema nepalengvino jutiklių valdymo). -

<b>5. Kaip lengva buvo naviguoti per programą?</b>
<b>Naudotojo vertinimas</b> (pažymi numerį skalėje tarp 1 ir 5)
<input type="checkbox"/> <b>Labai lengva</b> <input type="checkbox"/> Lengva <input type="checkbox"/> Nei lengva, nei sunku <input type="checkbox"/> Sunku <input type="checkbox"/> Labai sunku
<b>Komentaras</b> (privaloma, jeigu buvo sudėtinga naviguoti). -

<b>6. Ar programėlė atrodo išbaigta ir funkcionalumas tenkina Jūsų lūkesčius?</b>
<b>Naudotojo vertinimas</b> (pažymėti „X“)
<input type="checkbox"/> <b>Programėlės funkcionalumas tenkina mano lūkesčius, jokio papildomo funkcionalumo nereikia.</b> <input type="checkbox"/> Programėlėje trūksta tam tikro funkcionalumo, kuris galėtų teigiamai paveikti programos naudojimą. <input type="checkbox"/> Programėlė atrodo skurdi, funkcionalumas nedidelis ir manęs netenkina.
<b>Komentaras</b> (privaloma, jeigu programėlė netenkina lūkesčių – ko trūksta programėlėje?). -

<b>7. Ar programėlės prenumeratos kaina (privatiems - 6.99 €/mėn., verslui - 19.99 €/mėn.) yra tinkama ir taptumėte/liktumėte mūsų klientu ir toliau?</b>
<b>Naudotojo vertinimas</b> (pažymėti „X“)
<input type="checkbox"/> <b>Programėlės kaina yra adekvati ir tikrai tapčiau/likčiau klientu.</b> <input type="checkbox"/> Programėlės kaina yra šiek tiek per didelė, bet klientu tapčiau/likčiau. <input type="checkbox"/> Programėlės kaina yra per didelė, klientu tikrai netapčiau/nelikčiau. <input type="checkbox"/> Programėlės apmokestinamas neatrodo tinkamas.
<b>Komentaras</b> (privaloma, jeigu kaina yra neadekvati ar nenorėtumėt tapti/likti klientu).

# Anomalous Water Use Detection Using Machine Learning

Lukas Kulikovas<sup>1</sup> and Šarūnas Packevičius<sup>1</sup>

<sup>1</sup>*Faculty of Informatics, Kaunas University of Technology, Kaunas, Lithuania*

## Abstract

Water is an essential resource that is necessary for human life, agriculture, and industry. Numerous countries confront water shortages and inefficient water usage. Anomalous water usage detection is an important task in the efficient management of water resources and the prevention of water leaks. In this publication, we present a comparison between various machine learning models to detect unusual patterns in water usage data. All the machine learning models were tested on a real-world water usage dataset. The performance of each model was evaluated by accuracy, precision, recall, F1-score, ROC AUC, and MAE scores. The results indicate that PCA outlier detector can accurately detect uncommon patterns in water usage data. Our results outlined in this paper might be utilized by either individual homeowners or water utility corporations to detect water leaks more quickly and hence minimize water wastage.

## Keywords

Anomalous water use detection, unsupervised learning, semi-supervised learning.

## 1. Introduction

Water is a limited and invaluable resource, which plays a crucial role in supporting every life on the Earth. As the population grows and urbanization develops, so will the need for water. With increased concerns about water shortage and its wasteful use, proper water management has emerged as a significant worldwide challenge.

According to UNESCO, global water consumption has increased by about 1% per year since the 1980s, driven by growing populations and changing habits of water consumption [1]. In accordance with Burek et al. [2], worldwide water use will likely continue to grow at 1% yearly rate, culminating in an increase of 20 to 30% above current levels by 2050. Approximately 2.2 billion people do not have access to safe drinking water, roughly 4.2 billion people face acute water scarcity for at least one month each year, and around three billion individuals do not have access to basic handwashing facilities [1].

In the USA a typical household uses about 138 gallons (~ 522 liters) of water every day, where the toilet flush accounts for the majority of this use (24%), followed by faucets (20%), showers (20%), clothes washers (16%), leaks (13%), baths (3%), others (3%) and dishwashers (2%) [3]. Even while leaks account for just 13% of total home water usage, approximately 1 trillion gallons (3.785 trillion liters) of water can be wasted by residential leaks in the United States, with the average household's leaks accounting for nearly 10,000 gallons (~37,854 liters) of water every year [4].

The “Leaving No One Behind” report highlights the importance of improving water resources management and how it is crucial to address various problems, such as poverty, health, food security and environmental sustainability [5]. Water leaks are a sign of larger concerns, which come from outdated infrastructure, poor maintenance procedures and inefficient water use management [5]. As recommended by World Water Assessment Programme (WWAP), optimization of water resource management could aid in the prevention of water leaks, which include frequent pipe monitoring, maintenance, enhanced metering, and leak identifications systems [5].



Because water resources are limited, we must manage them effectively and responsibly to assure that it will be available for our future generations. To enhance water metering and existing water management systems, we compare a number of machine learning models to see how they perform under different water usage conditions. We begin by presenting relevant research on the topic of water leak anomaly detection, then we analyze various machine learning models and select the best one for detecting water usage abnormalities.

## 2. Related Work

This interesting yet difficult field, known as anomaly or outlier detection, has been acknowledged and thoroughly investigated by a plethora of research over the years. Han et al. [6] scrutinized how, in the last two decades, developed anomaly detection algorithms perform with regard to varying levels of supervision, different types of abnormalities and noisy and polluted data. Boukerche et al. [7] present a taxonomy of newly created outlier identification algorithms and approaches for high-dimensional data, data streams, big data, and little labeled data, followed by an overview of benefits and limitations for each algorithm. Wang et al. [8] presented a comprehensive and organized review of the progress of outlier detection methods. Chandola et al. [9] presented basic anomaly detection techniques, followed by an overview of the advantages and disadvantages of each technique. Campos et al. [10] conducted an extensive experimental study on the performance of a representative set of standard k-nearest neighborhood-based methods for unsupervised outlier detection.

Various techniques have been developed, which have been applied to a variety of real-life scenarios, that include intrusion detection systems, fraud detection, medical anomaly diagnosis, anomaly detection in wireless sensor networks and urban traffic flow [7]. One of the real-life scenarios is an anomalous water use detection system, which has the main objective of detecting unusual water use.

It is critical to distinguish between the numerous types of anomaly detection algorithms used to identify water leaks. Where physics and predefined expert rules seem appropriate to use, a traditional anomaly detection approach might be applied. This strategy, however, may not be appropriate in various situations, and labeled data may be difficult to get in this context. Furthermore, because anomalies are usually infrequent and unexpected, unsupervised learning algorithms have grown in prominence for their ability to detect abnormalities in unlabeled data. Where there is a possibility to obtain some labeled data points, semi-supervised techniques are used to train the algorithm, whereas fully supervised methods use solely labeled data. Active learning models are another way that uses expert or user input to categorize data and enhance algorithm accuracy. To further understand what research has been done and how different approaches function, the next paragraph will discuss traditional, unsupervised, semi-supervised, active learning, and fully supervised anomaly detection algorithms.

### 2.1. Traditional Anomaly Detection Methods

Sarangi [11] presented a technique for detecting water leaks and theft that is based on the concept of conservation of mass, which states that mass cannot be generated or destroyed. The author provided a solution to stop water theft by installing two sensors, where one of them detects the amount of water flow coming into the pipeline, the other one – water leaving the pipeline. If the difference in data collected by both sensors exceeds a specified limit defined to minimize false alarms, the microcontroller will send an alarm for further investigation of that location. Similarly, if there is a pipeline burst or water theft, the difference between the two measurements will be large.

Moni et al. [12] discussed how their approach could benefit farmers to detect water leaks. For this, they collected leak and no-leak vibration data from a pipeline using an accelerometer in three linear axes – x, y, and z. The results suggested that the root mean square (RMS) error, which in this case represents the difference between the measured vibration data and the expected vibration data for each condition, was always smaller when there was a leak. Final evaluation results indicated 87.9% accuracy when there was a leak and 96.3% accuracy when there was not.

Boudhaouia and Wira [13] proposed a general solution for collecting and managing water consumption by a non-intrusive approach which works at any measurement point from a water distribution system. The suggested approach is based on three parameters: maximum daily load curve,

minimum night flow (MNF), and a non-null time period calculated from water flow rate (PWNC). During nighttime, minimum night flow parameter is used. The algorithm considers the fact that average day flow rate is different from zero and night water use must be close to zero, otherwise it is classified as a small leak. During daytime, maximum daily load curve and MFN are used. The maximum daily load curve reflects a maximum water usage threshold. The detection is performed by comparing the current consumption to the specified limit. If the current water usage exceeds or is at the given threshold, accordingly a big leakage or risk of leakage is reported. Finally, a period without null consumption parameter is defined, which gives an overview of how long the water may run before it is considered a leak. Authors claim that their proposed procedure detected all water leakages.

## 2.2. Unsupervised Water Anomaly Detection

Ji et al. [14] compared linear regression, ARIMA and additive regression models using the data of Osan City to find the best method for water leak detection. First, authors calculated Watson value for the data, which resulted in  $2.83E-05$ . This indicated that the linear regression model was unfit for the provided dataset. For this reason, time series model ARIMA and additive regression Prophet models were proposed. The ARIMA model resulted in an accuracy rate of 64% with an average MAE for all houses of 39,230.88. Fbprophet provided a smaller MAE of 17,635.15 and smaller accuracy rate of 46% considering yearly trend, and accuracy rate was 65% with slightly higher MAE of 23,566.54 without considering yearly trend.

Fuentes and Mauricio [15] presented a smart water consumption measurement system, which involves house data collection, analyzation, and leakage alert functionality. Authors explain their 4-scenario algorithm, which involves negative trend evaluation, last 24-hour consumption evaluation, similar consumption evaluation and historical data process. Authors extract historical data into different features, apply the k-NN algorithm to obtain a list of the consumptions that are closest to ( $K = 4$ ) and apply Tchebysheff theorem for confidence interval construction. Outstandingly, the algorithm demonstrated accuracy, recall, precision, and F1-score as 100%, surpassing the rest of the leak detection algorithms.

Patabendige et al. [16] developed a context aware anomaly detection algorithm that takes the relevant context for each day into account, applies the k-NN algorithm together with Gaussian error function to transform the outlier score into a probability value. The system also generates an anomaly score for each day together with a rationale that describes what could have caused an unusual water use and reports it to the user.

## 2.3. Semi-supervised Water Anomaly Detection

Lee et al. [17] built RNN-LSTM (Recurrent Neural Networks-Long Short-Term Memory) deep learning model. Authors applied a model on the actual leakage data and the leak was recognized at most points immediately after the accident. Also, this model resulted in good performance and showed more than 90% accuracy. Authors also mention that the model is highly scalable.

Pang et al. [17] proposed a deep reinforcement learning-based approach that enables an end-to-end optimization of the detection of both labeled and unlabeled data. This approach learns the known abnormalities by automatically interacting with an anomaly-biased simulation environment, while continuously extending the learned abnormality to novel classes of anomaly by actively exploring anomalies in the unlabeled data. The authors demonstrated how experiments on 48 real-world datasets proved that their model outperforms state-of-the-art competing methods.

Blázquez-García et al. [18] proposed a self-supervised water leak detection method based on a self-supervised classification of flow time series, called Self-Supervised Leak Detector (SSLD). This algorithm does not require external class labels and instead uses labels that have been assigned to artificially generated data. In the first step of their self-supervised framework, a self-labeled training set is generated. Later, the classifier is trained to learn the mapping between input and its corresponding label. Authors concluded that proposed SSLD method obtains the best trade-off between detecting the majority of the detectable leaks and providing a low FPR. Also, the provided model is purely data-driven and therefore does not require in-depth knowledge about the dynamics of the series.

## 2.4. Active Learning Approach for Water Anomaly Detection

Numerous authors proposed various active-learning-based algorithms that can interactively query user's response to label data with the desired outputs. Wang et al. [19] proposed an active anomaly detection framework Active-MTSAD. Das et al. [20] proposed the Active Anomaly Discovery (AAD) algorithm. Zhu and Yang [21] proposed the tripartite active learning method. Vercruyssen et al. [22] proposed a novel constrained-clustering-based approach for anomaly detection that works in both unsupervised and semi-supervised setting. Active learning approach strategy starts with unsupervised learning where most important unlabeled instances are selected and then provided to the expert or user. Later, the model gets updated with new labels so it can achieve higher performance. This type of learning could adapt to the user's needs and provide better results. As authors mention, their experiments demonstrated how active learning models outperform most methods for domain-specific anomaly detection [19].

## 2.5. Supervised Water Anomaly Detection

Ismail et al. [23] proposed a comparison between four machine learning classification models. First, human annotators generated a ground truth dataset for water consumption. Secondly, the data was normalized using the z-score. Later, the data was applied to four machine learning models, including Decision Tree, k-NN, Naïve Bayes, and Random Forest. Authors concluded that Random Forest machine learning model gives the highest overall accuracy of 87%, precision of 75% and recall of 83 % compared to other three classification models.

Amora et al. [24] designed a Bidirectional LSTM (BiLSTM) machine learning model and compared it with Gated recurrent units (GRU) and Autoregressive models. The suggested method has two iterative loops, in which the outer loop optimizes the batch size, number of input time steps, and number of output units in each LSTM. Also, Hyperactive library and mean square error are used in this simulation. The inner loop of the suggested method updates each weighting element in the BiLSTM using the traditional Adam optimizer. According to the results gathered by authors, BiLSTM outperforms the GRU and Autoregression models when detecting water leak.

A supervised learning approach in water anomaly detection most of the time is a challenge due to lack of labels. Zese et al. [25] applied several supervised machine learning techniques for the automatic detection of leakages. Authors demonstrated how convolutional neural network models were the best in detecting both the presence and absence of water leaks. Overall results showed that the model was able to classify water leaks with accuracy, precision, recall, F-measure, and AUC ROC ranging from 92% to 99% [25]. Fan et al. [26] demonstrated how artificial neural networks (ANN) can accurately classify leaking versus non-leaking scenarios. However, it requires a balanced dataset under both leaking and non-leaking conditions. Nonetheless, authors claim that their model detected leaks in pipes with 100% accuracy [26].

## 3. The Data

In this study, we use SWM (smart water meter) time-series (Trial A) data from the DAIAD Trials, which is available on GitHub<sup>1</sup>, and we also include one of our water use data, which was gathered from a smart water sensor in one person's home. Our water consumption data is also available on GitHub<sup>2</sup>.

The DAIAD dataset contains hourly water consumption measurements for ninety-two households in Alicante. Each time-series starts at 1/3/2016 and ends at 28/2/2018. On average, there are 7108 measurements per household, which in total is 653,954 records. The outliers amount to 322 out of the total records in the dataset [27]. As they are a typical user behavior and not the water leak, thus any prediction as anomalous on the dataset's outlier would be considered false positive.

---

<sup>1</sup> [https://github.com/DAIAD/data/blob/master/swm\\_trialA\\_clean.zip](https://github.com/DAIAD/data/blob/master/swm_trialA_clean.zip)

<sup>2</sup> <https://github.com/LukaLike/water-consumption-data>

Our time-series dataset contains minutely measurements, starting on 8/13/2022 and ending at 3/2/2023. In total, this dataset contains 263,811 records. This dataset has no outlier and any prediction as anomalous on this dataset would also be considered false positive.

## 4. Methodology

In this section, we present a comparison of different outlier detection models to see which one is the most efficient in detecting anomalous water use. Different scenarios are used to achieve this, and important statistical parameters are evaluated. In these experiments we include CBLOF, COPOD, ECOD, HBOS, IForest, KNN, LOF, OCSVM, and PCA outlier detector models that are publicly available on GitHub platform<sup>1</sup>. The semi-supervised SSDO model that is used in these experiments is also publicly available on the GitHub platform<sup>2</sup>.

To extensively analyze various machine learning models for detecting anomalous water use, we first prepared the data. This included reading timestamp and consumption values from the datasets. To ensure the quality of the data, preprocessing was applied, which included filling missing values and ignoring days with no consumption. Afterwards, different leak scenarios were added to the datasets. Throughout the data preparation process, we ensured that any modifications made to the dataset were performed only on the data that was temporarily stored in memory. This enabled us to maintain the original dataset intact for future runs. Subsequently, a feature and label matrix were built.

Since multiple tests were done, several feature matrices and their sizes were created, including 2, 3, or 4-dimensional feature matrices with mean, min, max, or longest non-zero water flowing duration, and varying window range sizes. The hourly datasets were separated into 1-, 2-, 3-, 4-, and 6-hour window sizes, while the minutely datasets were divided into 1-, 5-, and 10-minute window sizes. In this regard, labels matrices were created, which included values indicating whether or not a given window range had a water leak. Because none of the datasets had any unusual water use points, only the values generated by the water leak function generator were marked as anomalous. Finally, the data was separated into training and testing sets and submitted to all outlier detection algorithms.

In the following sections, we are going to further explain how we prepared the data and scenarios, and how we completed the evaluation of the models.

### 4.1. Data and Scenarios Preparation

To evaluate each model, correct and most realistic data must have been used. For this, the following process was followed:

1. In the first step, we prepared the data. This included filling the missing values with zeros and removing the days that had not used any water.
2. Secondly, water leak scenarios were generated and included into the dataset at the program's runtime. Each assessment was performed 5 times and each time new random water leak scenarios were included. The model was fit on the first 80% and tested on the last 20% of the data. In the training randomly [0; 3], and in the testing data – [1; 3] leak scenarios were included. Each scenario had a chance to overlap, contain the same scenarios, and last between 5 and 180 minutes. All of the possible scenarios are shown in **Table 1**.

**Table 1**  
Water leak scenarios

Scenario	Minimum flowing speed (ml/min)	Maximum flowing speed (ml/min)
Dripping pipe	33.1	35.9
Dripping faucet	539.8	573.2
Broken mainline	3671.9	3899

<sup>1</sup> <https://github.com/yzhao062/pyod>

<sup>2</sup> <https://github.com/Vincent-Vercruyssen/anomatools>

Flowing faucet	7343.7	7798
Leaking toilet	15434	16389
Broken sprinkler	45898.1	48737.2

## 4.2. Evaluation

To evaluate each model's performance, we calculate accuracy, precision, recall, F1-score, ED-score, AUC ROC, and AUC PR values.

### Precision

It is a metric that assesses the proportion of a model's positive predictions that are actually correct. Its definition is the proportion of accurate positive predictions to all positive predictions.

$$Precision = \frac{TP}{TP + FP'} \quad (1)$$

where:

TP: True Positive, the number of correct positive predictions

FP': False Positive, the number of incorrect positive predictions

### Recall

It is a metric that assesses how many actual positive instances a model can identify. It is determined by dividing the total number of positive occurrences by the proportion of actual positive predictions.

$$Recall = \frac{TP}{TP + FN'} \quad (2)$$

where:

FN': False Negative, the number of incorrect negative predictions

### F1-score

It is an evaluation metric that is defined as the harmonic mean of the precision P and recall R.

$$F_1 = \frac{2PR}{P + R} \quad (3)$$

### AUC-ROC: Area Under the Receiver Operating Characteristic Curve

It is a statistic used to assess how well binary classification models perform. The AUC-ROC ranges from 0 to 1, with a score of 1 denoting flawless performance and a score of 0.5 denoting no improvement over a random guessing.

### AUC-PR: Area Under the Precision-Recall Curve

It is also a metric used to assess how well binary classification models perform. The AUC-PR concentrates on the model's precision and recall rather than the true positive and false positive rate, like AUC-ROC does. At various levels, it calculates the precision and recall trade-off. The AUC-PR scales from 0 to 1, with a score of 1 denoting perfect performance and a score of 0.5 denoting no better than a random guessing.

### ED-score: Early Detection score

It is a score that rewards detections that are close to the fault start time  $t_f$ , with the reward decreasing as the detection moves further away. The detection time is defined as the earliest time step within the fault window duration at which the algorithm registered a detection. In our tests, as in Vercruyssen's, a successful detection is recorded if the algorithm generates detections that persist for at least 75% of the time. The early detection score is calculated by first determining the delay of the first detection in the defined fault window, given by  $x = t_d - t_f$ , and then applying the following sigmoid function to this detection [22]:



$$\sigma(x) = \frac{2}{1 + e^{\left(\frac{\alpha}{T_w} * x\right)}}, \quad (4)$$

where:

$\sigma$  – value that is defined such that  $\sigma(x) \approx 0$ , while  $\sigma(0) = 1$  for any values of  $\alpha < \infty$ ,  
 $T_w$  – fault window duration [22].

In our tests,  $\alpha$  was selected as 6. A detection that occurs outside the fault window is not included in the score.

## 5. Analyzes and Results

In the following sections, we present machine learning models comparisons in different scenarios, which include time windows, contexts, and data dimensions. The default parameter of each evaluation strategy is fixed 3 hour, contextual, 1 dimension (mean), non-overlapping window interval. For each strategy, only one default parameter will be changed or if stated otherwise.

### 5.1. Models Evaluation on Different Time Windows

During this experiment, we evaluated the performance of the models at different sliding window sizes. First, the models were tested using DAIAD data (in hourly frames) with a window width of 3 hours and a sliding window size of 1, 2, 3, 4, and 6 hours, respectively. The models were then tested using data from the system’s developer (in minute frames) with a window width of 3 hours and a sliding window size of 1, 5, and 10 minutes, respectively. In the graphs, the  $x$  axis indicates the size of the sliding window, and the  $y$  axis indicates the model’s performance for the specified criterion.

#### Comparison of the models in different hourly time frames

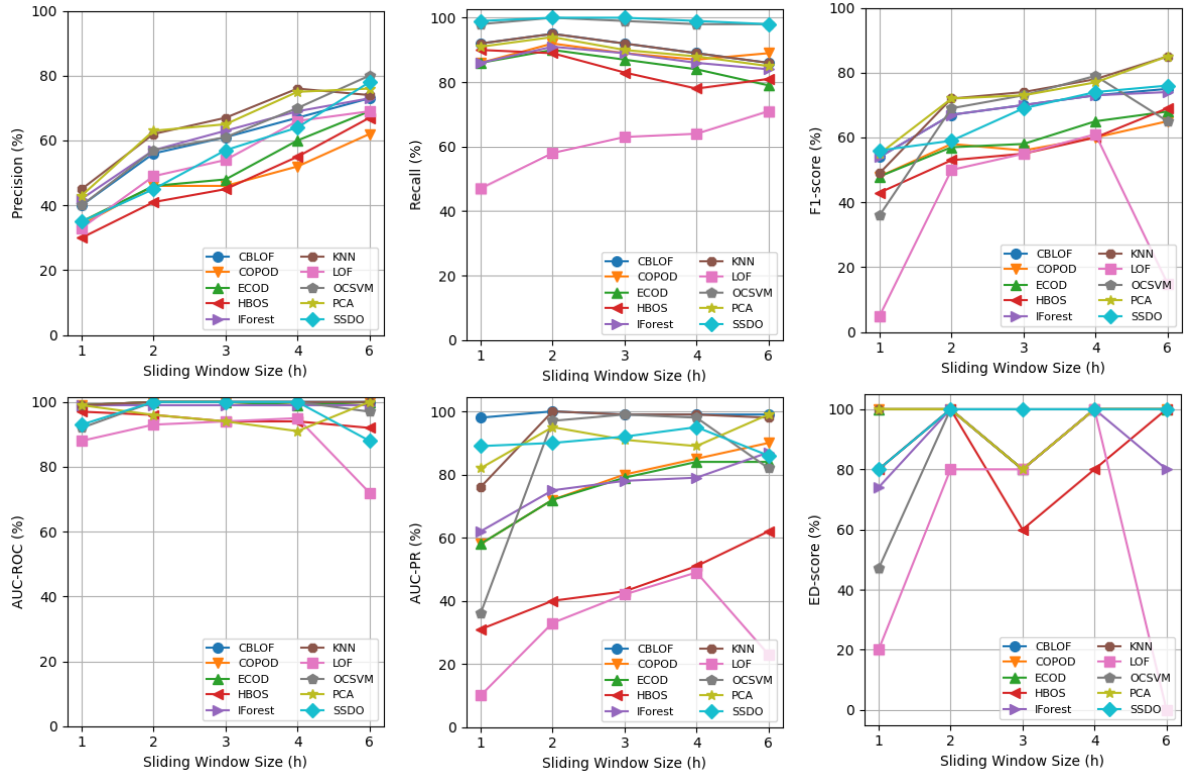


Figure 1: Comparison of models in different hourly frames

The experiment with different hourly frames showed that the precision decreased 2 times as the sliding window decreased from 6h to 1h, but the recall remained the same in most of the models. Also looking at the ED-score, most models performed poorly in terms of early detection of water leakage with a 1h rolling window. The semi-supervised SSDO model was able to perform better than most of the unsupervised machine learning models.

### Comparison of the models in different minutely time frames

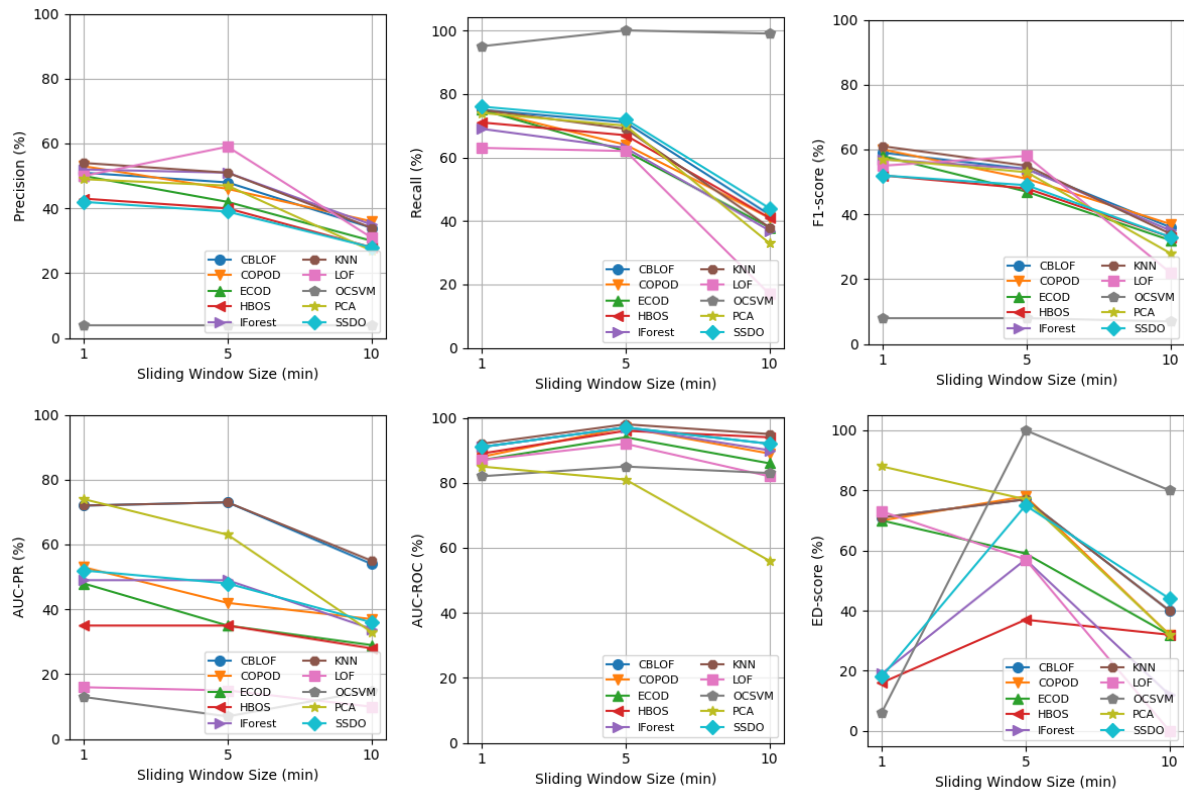


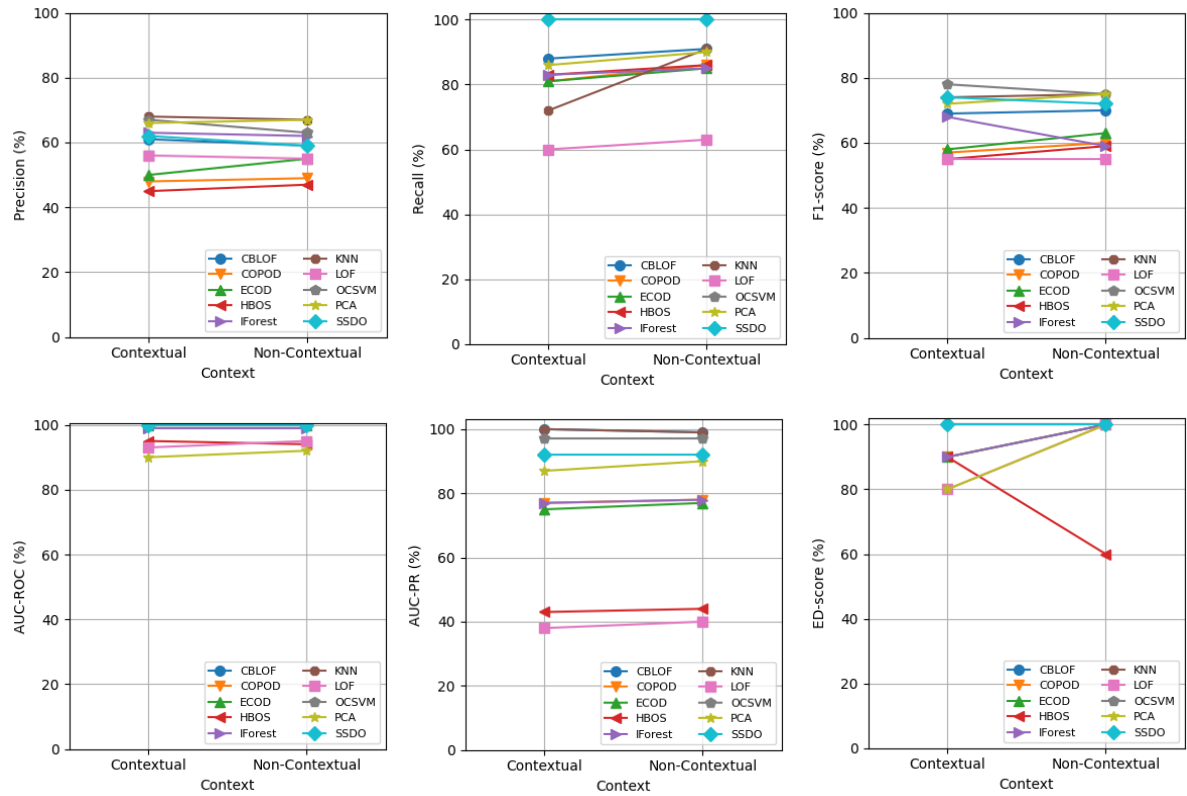
Figure 2: Comparison of models in different minutely frames

The experiment with different minute frames showed that decreasing the size of the rolling window from 10 minutes to 5 minutes improved accuracy and recall much more than changing the window size from 5 minutes to 1 minute. The partially trained SSDO model was not able to provide better accuracy compared to the unsupervised models but provided in the best recall values.

## 5.2. Models Evaluation on Different Context

During this experiment, we tested the performance of the models in different contexts. In the graph, not contextual data indicates that for training and testing, all data (from Monday to Sunday) was taken, while contextual data indicates that during different testing stages working days (from Monday to Friday) and weekend days (Saturday and Sunday) were taken, and the resulting estimates were summed and divided by two.

The graphs on the  $x$  axis indicate whether the result is contextual or not, and on the  $y$  axis is displayed model's performance for the specified criterion.



**Figure 3:** Comparison of models in different contexts

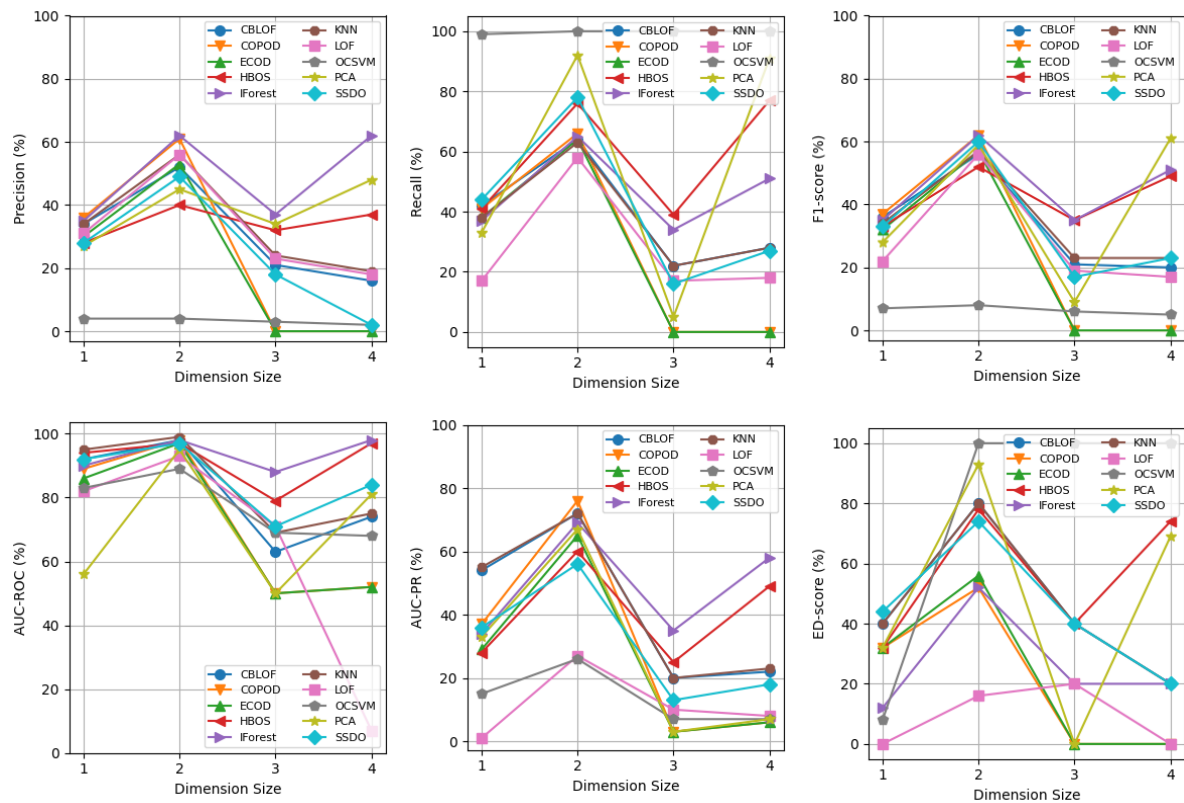
This experiment showed that the use of contextual data reduces the precision and improves the recall only marginally. However, when using non-contextual data, most models were able to detect water leakage much faster than when using contextual data.

### 5.3. Models Evaluation on Different Data Dimensions

During this experiment, we evaluated the performance of the models on different data dimensions. In the graphs, the dimensions marked on the  $x$  axis are expressed as follows:

- One dimension includes the mean of the data.
- Two dimensions include the mean of the data and the longest water running period (in minutes).
- Three dimensions include the average of the data, the minimum and the maximum water consumption over a three-hour period.
- Four dimensions include the third dimension plus the longest water running period (in minutes).

On the  $y$  axis the performance of the model is displayed for the specified criterion.



**Figure 4:** Comparison of models in different data dimensions

The experiment on different data dimensions showed that using two dimensions (average and longest water running period) gives the best overall results – the models were able to provide the best precision, recall and also provide the fastest water leakage detection rate.

## 6. Conclusion

In this work, we compared unsupervised CBLOF, COPOD, ECOD, HBOS, IForest, KNN, LOF, OCSVM, PCA and semi-supervised SSSO outlier detectors on water usage data and how they perform in detecting anomalous water usage. Experiments performed in different minute frames showed that when the system collects water consumption data every minute, all models perform much better with a smaller sliding window size, both in terms of precision and recall estimates. The results thus suggest that using the PCA outlier detector with a minute sliding window will be able to detect water leakage approximately 78% of the time, as well as to detect the onset of anomalous water use very quickly in order to stop unwanted consequences in time. Experiments performed at different levels of context showed that the difference between contextual and non-contextual data is not incredibly significant, but in most cases the models were able to detect leakage much faster using non-contextual data. Experiments performed using different dimension values indicated that using two dimensions (average water consumption and longest water running period) models were able to give the best results, and overall, the PCA model produced the best outcome, being able to detect water leakage 95% of the time and efficiently identify the start of the anomaly.

## 7. References

- [1] UNESCO. The United Nations World Water Development Report 2021: Valuing Water, Mar 22, 2021.
- [2] P. Burek, Y. Satoh, G. Fischer, M. T. Kahil, A. Scherzer, et al. Water Futures and Solutions: World Water Scenarios Report, 2016. URL: <https://pure.iiasa.ac.at/id/eprint/13008/1/WP-16-006.pdf>.

- [3] WWAP (UNESCO World Water Assessment Programme). The United Nations World Water Development Report 2019: Leaving no One Behind, 2019.
- [4] W. B. DeOreo, P. Mayer, B. Dziegielewski, J. Kiefer. Residential End Uses of Water, Version 2 Executive Report, 2016. URL: [https://www.circleofblue.org/wp-content/uploads/2016/04/WRF\\_REU2016.pdf](https://www.circleofblue.org/wp-content/uploads/2016/04/WRF_REU2016.pdf).
- [5] Epa.Gov, Fix a Leak Week. URL: <https://www.epa.gov/watersense/fix-leak-week>.
- [6] S. Han, X. Hu, H. Huang, M. Jiang, Y. Zhao. ADBench: Anomaly Detection Benchmark, 2022. doi:10.48550/arXiv.2206.09426.
- [7] A. Boukerche, L. Zheng, O. Alfandi. Outlier Detection: Methods, Models, and Classification. ACM Computing Surveys, 2020, vol. 53, no. 3. doi:10.1145/3381028.
- [8] H. Wang, M. J. Bah, M. Hammad. Progress in Outlier Detection Techniques: A Survey, 2019. doi:10.1109/ACCESS.2019.2932769.
- [9] V. Chandola, A. Banerjee, V. Kumar. Anomaly Detection: A Survey. ACM Computing Survey, 2009, vol. 41, no. 3. doi:10.1145/1541880.1541882.
- [10] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, et al. On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical Study. Data Mining and Knowledge Discovery, 2016, vol. 30, no. 4. pp. 891-927. doi:10.1007/s10618-015-0444-8.
- [11] A. K. Sarangi. Smart Water Leakage and Theft Detection using IoT, 2020. doi:10.1109/I4Tech48345.2020.9102701.
- [12] N. A. Moni, B. Sigweni, M. Mangwala, L. Kolobe. Water Leak Detection from Irrigation Pipelines in Botswana using Vibration Interpretation Technique, 2019. doi:10.1109/AFRICON46755.2019.9133829.
- [13] A. Boudhaouia, P. Wira. Water Consumption Analysis for Real-Time Leakage Detection in the Context of a Smart Tertiary Building, 2018. doi:10.1109/ICASS.2018.8651976.
- [14] D. J. A. Amora, M. A. V. Janapin, B. A. M. Calayag, C. L. P. P. Rioflorido, N. M. Estur, et al. Design of a Household Consumption Based Water Leak Detection System Utilizing Machine Learning Algorithm, 2022. doi:10.1109/IET-ICETA56553.2022.9971564.
- [15] H. Ismail, R. Elabyad, A. Dyab. Smart Residential Water Leak and Overuse Detection System using Machine Learning, 2022. doi: 10.1109/AICCSA56895.2022.10017508.
- [16] M. Ji, G. Yi, J. Jung. Central Prediction System for Time Series Comparison and Analysis of Water Usage Data, 2020. doi:10.1109/ACCESS.2019.2963373.
- [17] H. Fuentes, D. Mauricio. Smart Water Consumption Measurement System for Houses using IoT and Cloud Computing. Environmental Monitoring and Assessment, 2020, vol. 192, no. 9. pp. 602. doi:10.1007/s10661-020-08535-4.
- [18] S. Patabendige, R. C. Oliver, R. Wang, W. Liu. Detection and Interpretation of Anomalous Water use for Non-Residential Customers. Environmental Modelling & Software, 2018, vol. 100. pp. 291-301. doi:10.1016/j.envsoft.2017.11.028.
- [19] G. Pang, A. van den Hengel, C. Shen, L. Cao. Toward Deep Supervised Anomaly Detection: Reinforcement Learning from Partially Labeled Anomaly Data. Virtual Event, Singapore ed. New York, NY, USA: Association for Computing Machinery, 2021. doi:10.1145/3447548.3467417.
- [20] A. B. García, A. Conde, U. Mori, J. A. Lozano. Water Leak Detection using Self-Supervised Time Series Classification. Information Sciences, 2021, vol. 574. pp. 528-541. doi:10.1016/j.ins.2021.06.015.
- [21] W. Wang, P. Chen, Y. Xu, Z. He. Active-MTSAD: Multivariate Time Series Anomaly Detection with Active Learning, 2022. doi:10.1109/DSN53405.2022.00036.
- [22] S. Das, W. K. Wong, T. Dietterich, A. Fern, A. Emmott. Discovering Anomalies by Incorporating Feedback from an Expert. ACM Transactions on Knowledge Discovery from Data, 2020, vol. 14, no. 4. doi:10.1145/3396608.
- [23] Y. Zhu, K. Yang. Tripartite Active Learning for Interactive Anomaly Discovery, 2019. doi:10.1109/ACCESS.2019.2915388.
- [24] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Bäumer. Semi-Supervised Anomaly Detection with an Application to Water Analytics. IEEE, 2018. doi:10.1109/ICDM.2018.00068.
- [25] R. Zese, E. Bellodi, C. Luciani, S. Alvisi. Neural Network Techniques for Detecting Intra-Domestic Water Leaks of Different Magnitude, 2021. doi:10.1109/ACCESS.2021.3111113.

- [26] X. Fan, X. Zhang, X. B. Yu. Machine Learning Model and Strategy for Fast and Accurate Detection of Leaks in Water Supply Network. *Journal of Infrastructure Preservation and Resilience*, 2021, vol. 2, no. 1. pp. 10. doi:10.1186/s43065-021-00021-6.
- [27] S. Athanasiou, G. Giannopoulos, Y. Kouvaras, P. Chronis, G. Hatzigeorgakidis, et al. *Trials Evaluation and Social Experiment Results*, 2017. URL: [http://www.daiad.eu/wp-content/uploads/2017/11/D7.3\\_Trials\\_Evaluation\\_v1.0.pdf](http://www.daiad.eu/wp-content/uploads/2017/11/D7.3_Trials_Evaluation_v1.0.pdf).