



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

Elvinas Jakubcevičius

**NUTOLUSIOS ORŲ STOTELĖS TEIKIAMŲ DUOMENŲ TIKSLUMO IR
GREITAVEIKOS TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Lekt. dr. Dominykas Barisas

KAUNAS, 2017

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

**NUTOLUSIOS ORŲ STOTELĖS TEIKIAMŲ DUOMENŲ TIKSLUMO IR
GREITAVEIKOS TYRIMAS**

Baigiamasis magistro projektas
Programų sistemų inžinerija (kodas M4046N21)

Vadovas

Lekt. dr. Dominykas Barisas

Recenzentas

Doc. dr. Šarūnas Packevičius

Projektą atliko

Elvinas Jakubcevičius

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

(Fakultetas)

ELVINAS JAKUBCEVIČIUS

(Studento vardas, pavardė)

PROGRAMŲ SISTEMŲ INŽINERIJA (M4046N21)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto

„Nutolusios orų stotelės teikiamų duomenų tikslumo ir greitaveikos tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. Gegužės 19 d.
Kaunas

Patvirtinu, kad mano, **Elvino Jakubcevičiaus**, baigiamasis projektas tema „Nutolusios orų stotelės teikiamų duomenų tikslumo ir greitaveikos tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

TURINYS

1. Įvadas	10
1.1. Dokumento paskirtis	10
1.2. Santrauka	10
2. Analitinė dalis	11
2.1. Egzistuojantys sprendimai	11
2.2. Orų prognozavimas	11
2.2.1. Naudojant neuroninį tinklą	12
2.2.2. Slankiojančio lango metodas	14
2.2.3. Naudojantis kitų sistemų teikiamomis orų prognozėmis	18
2.3. Ligų nustatymas	20
2.3.1. Linijinis ligos progresas	20
2.3.2. Policiklinis ligos progresas	21
2.3.3. Maksimalus ligos progresas	21
2.4. Stotelės aparatūrinė įranga	22
2.4.1. Jutikliai	22
2.5. Išvados	23
3. Projektinė dalis	25
3.1. Sistemos paskirtis	25
3.2. Sistemos panaudojimo atvejai	26
3.2.1. Sistemos panaudos atvejų diagrama	26
3.2.2. Panaudojimo atvejų scenarijai	27
3.3. Funkciniai reikalavimai	36
3.4. Nefunkciniai reikalavimai	39
3.4.1. Reikalavimai sistemos išvaizdai	39
3.4.2. Reikalavimai panaudojamumui	40
3.4.3. Reikalavimai vykdymo savybėms	41
3.4.4. Reikalavimai saugumui	43
3.5. Naudotos technologijos ir metodologijos	44
3.6. Sistemos architektūros modelis	45
3.6.1. Skirstymas į paketus	45
3.7. Duomenų vaizdas	51
3.8. Išdėstymo vaizdas	52
3.9. Naudojami trečiųjų šalių komponentai	53
3.10. Autonominės meteorologinės stotelės konstrukcija	53
4. Tyrimo dalis	54
4.1. Tyrimo tikslas	54
4.2. Tyrimo aprašymas	54
4.3. Tyrimo eiga	54

4.3.1. Orų parametrų tikslumas.....	54
4.3.2. Prognozės tikslumas	57
4.4. Tyrimo išvados	59
5. Eksperimentinė dalis.....	61
5.1. Eksperimentinio tyrimo tikslas	61
5.2. Eksperimentinio tyrimo aprašymas	61
5.3. Eksperimentinio tyrimo eiga.....	62
5.4. Eksperimentinio tyrimo rezultatai	62
5.5. Eksperimentinio tyrimo išvados	67
6. Išvados	68
7. Literatūra.....	69
8. Terminų ir santrumpų žodynas	70

LENTELIŲ SĄRAŠAS

2.2.1 lentelė Algoritmo prognozių tikslumo rezultatai	18
2.2.2 lentelė Oro sąlygos būseną pagal saulės radiaciją	19
3.2.1 lentelė PA 1.1 Gauti konkrečios stotelės tam tikro laikotarpio duomenis	27
3.2.2 lentelė PA 1.2 Gauti konkrečios stotelės paskutinius duomenis	27
3.2.3 lentelė PA 1.3 Gauti visas sistemoje užregistruotas stoteles	28
3.2.4 lentelė PA 2 Žiūrėti informaciją grafikų pagalba	28
3.2.5 lentelė PA 3 Žiūrėti stotelės informaciją lentelių pagalba	29
3.2.6 lentelė PA 4 orų prognozės peržiūra	29
3.2.7 lentelė PA 5 Peržiūrėti stotelės būvimo vietą žemėlapyje	30
3.2.8 lentelė PA 6 Užregistruoti nuosavą stotelę	30
3.2.9 lentelė PA 7 Peržiūrėti visas sistemoje užregistruotas stoteles	31
3.2.10 lentelė PA 8 Išjungti/Ijungti stotelę	31
3.2.11 lentelė PA 9 Redaguoti stotelės profilį	32
3.2.12 lentelė PA 10 Sudaryti augalų ligų modelius	32
3.2.13 lentelė PA 11 Pasirinkti stotelės sekamų augalų ligų modelius	33
3.2.14 lentelė PA 12 Trinti stotelę iš sistemos	33
3.2.15 lentelė PA 13 Nustatyti individualius perspėjimus	34
3.2.16 lentelė PA 14 Redaguoti visų stotelių profilius	34
3.2.17 lentelė PA 15 Trinti stoteles iš sistemos	35
3.2.18 lentelė PA 16 Peržiūrėti prisijungimų prie sistemos statistiką	35
3.3.1 lentelė Laikotarpio pasirinkimas grafikų generavime	36
3.3.2 lentelė Filtravimo ir rikiavimo lentelėse pasirinkimas	36
3.3.3 lentelė Sistemos sąsaja	36
3.3.4 lentelė Orų prognozių peržiūra	36
3.3.5 lentelė Stebėti duomenis grafiko pagalba	37
3.3.6 lentelė Stebėti duomenis lentelėje	37
3.3.7 lentelė Žiūrėti stotelės būvimo vietą	37
3.3.8 lentelė Užregistruoti stotelę	37
3.3.9 lentelė Žiūrėti visų stotelių duomenis	37
3.3.10 lentelė Stotelės informacijos redagavimas	38
3.3.11 lentelė Ligos modelio sudarymas	38
3.3.12 lentelė Sekti ligos modelį	38
3.3.13 lentelė Stotelės trynimas	38
3.3.14 lentelė Nustatyti perspėjimus	38
3.3.15 lentelė Administravimas	39
3.4.1 lentelė Išvaizda	39
3.4.2 lentelė Stilius	39
3.4.3 lentelė Naudojimosi paprastumas	40
3.4.4 lentelė Naudojimosi paprastumas	40
3.4.5 lentelė Vartotojui skirtos konfigūravimo priemonės	40
3.4.6 lentelė Mokymosi reikalavimai	40
3.4.7 lentelė Supratingumas ir mandagumas	41
3.4.8 lentelė Prieinamumas neįgaliems	41
3.4.9 lentelė Užduočių vykdymo greitis	41
3.4.10 lentelė Reikalavimai tikslumui	41
3.4.11 lentelė Patikimumas	42
3.4.12 lentelė Pasiiekiamumas	42
3.4.13 lentelė Reikalavimai apdorojamų duomenų apimtims	42
3.4.14 lentelė Reikalavimai išplečiamumui	42
3.4.15 lentelė Reikalavimai produkto ilgaamžiškumui	43
3.4.16 lentelė Prieigos reikalavimai	43

3.4.17 lentelė Vientisumo (integralumo) reikalavimai	43
3.4.18 lentelė Reikalavimai privatumui	43
3.4.19 lentelė Audito reikalavimai.....	44
3.4.20 lentelė Reikalavimai savisaugai nuo išorinių grėsmių.....	44
4.3.1 lentelė Prognozių tyrimo rezultatai.....	59
5.4.1 lentelė Langų kiekis – 3, lango dydis 12 d. tikslumas	62
5.4.2 lentelė Langų kiekis – 5, lango dydis - 12 d. tikslumas.....	62
5.4.3 lentelė Langų kiekis – 8, lango dydis - 7 d. tikslumas.....	63
5.4.4 lentelė Langų kiekis – 11, lango dydis - 4 d. tikslumas.....	63
5.4.5 lentelė Langų kiekis – 13, lango dydis - 2 d. tikslumas.....	64

PAVEIKSLŲ SĄRAŠAS

2.1.1 pav. „iMetos“ sistemos langas	11
2.2.1 pav. W_1 atvaizduoja pirmąjį langą, W_2 antrąjį langą	14
2.2.2 pav. Slankiojančio lango algoritmas	15
2.2.3 pav. Maksimalios temperatūros algoritmo prognozių grafikas [12].....	16
2.2.4 pav. Minimalios temperatūros algoritmo prognozių grafikas [12].....	16
2.2.5 pav. Drėgmės algoritmo prognozių grafikas [12].....	17
2.2.6 pav. Kritulių kiekio algoritmo prognozių grafikas [12].....	17
2.3.1 pav. Linijinis ligos progreso grafikas	20
2.3.2 pav. Policiklinis ligos progreso grafikas.....	21
2.3.3 pav. Maksimalaus lygio ligos progreso grafikas	21
2.4.1 pav. „SparkFun“ orų plokštė „Arduino“ mikrokontroleriui	22
2.4.2 pav. Vėjo krypties, greičio ir kritulių jutikliai	22
2.4.3 pav. Dirvožemio temperatūros ir drėgmės jutiklis.....	23
3.2.1 pav. Panaudos atvejų diagrama.....	26
3.5.1 pav. Serverio dalies paketų diagrama	45
3.5.2 pav. Meteorologinės stotelės paketų diagrama	45
3.5.3 pav. „HTTP“ paketas	46
3.5.4 pav. „Jobs“ paketas	47
3.5.5 pav. „Exceptions“ paketas	47
3.5.6 pav. „Services“ paketas.....	48
3.5.7 pav. „Repositories“ paketas	48
3.5.8 pav. „Models“ paketas	49
3.5.9 pav. „DataBase“ paketas.....	49
3.5.10 pav. „Services“ paketas.....	50
3.5.11 pav. „Models“ paketas	50
3.5.12 pav. „DataBase“ paketas.....	51
3.6.1 pav. Serverio dalies duomenų bazės vaizdas	51
3.6.2 pav. Autonominės meteorologinės stotelės duomenų bazės vaizdas.....	52
3.7.1 pav. Išdėstymo diagrama	52
3.9.1 pav. Sukurta autonominė meteorologinė stotelė.....	53
4.3.1 pav. Temperatūros matavimų grafikas.....	55
4.3.2 pav. Dirvožemio temperatūros matavimų grafikas	55
4.3.3 pav. Kritulių matavimų grafikas	56
4.3.4 pav. Vėjo greičio matavimų grafikas	56
4.3.5 pav. Dirvožemio drėgmės matavimų grafikas	57
4.3.6 pav. Temperatūros prognozės grafikas	57
4.3.7 pav. Slėgio prognozės grafikas	58
4.3.8 pav. Vėjo greičio prognozės grafikas	58
4.3.9 pav. Vėjo krypties prognozių grafikas.....	59
5.4.1 pav. Temperatūros prognozių tikslumo grafikas	64
5.4.2 Slėgio prognozių tikslumo grafikas	65
5.4.3 pav. Vėjo greičio prognozių tikslumo grafikas.....	65
5.4.4 pav. Vėjo krypties prognozių tikslumo grafikas	66
5.4.5 pav. Algoritmo vykdymo trukmė.....	67

Jakubcevičius, Elvinas. *Research Based on the Data Provided by the Autonomic Meteorological Station, the Accuracy and Performance of the Weather Forecast*. Master's thesis in Software engineering / supervisor lect. dr. Dominykas Barisas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Technological sciences area, informatics engineering field

Key words: weather station, weather forecasting, plant disease model

Kaunas, 2017. 70 p.

SUMMARY

Throughout the modern and developing XXI century electronic devices are becoming more and more popular and there are increasing numbers of areas where they are established. Most of electronic devices act a huge role while making human's life easier and they also allow to solve casual problems in a more quick and efficient way.

Casual activities, such as work or leisure time always were, and until now are dependable from the weather conditions. Because of this factor scientific field, involving weather forecasting, is considered as very old and extensive. In the ancient ages there were basic methods applied in order to forecast the weather, but nowadays these tasks are carried out by the super machines. There are complex math methods, algorithms, information provided by the satellites, and many more. This field of science is still under development, and rarely but sometimes mistakes or errors occur while forecasting the weather.

Despite the fact that there are numbers of professional meteorological stations, that allow us to gather actual data, it is not enough of it especially in farming. Meteorological station is undoubtedly useful for the farmers, because it allows to gather actual data in the farmed area, so consequently farmers are able to plan their time and funds in a more efficient way. For example, when considering the changes of the temperature and the level of moisture in the soil, farmer is able to prospect whether it is necessary or not to resow the crop.

During this research there was analysis performed basing on the functionality of existing autonomic meteorological stations, also on the methods and algorithms of weather forecasting, and principles by which plant diseases are determined.

In this paper there is analytical part presented, where the procedure of projecting meteorological station system is being described. Therefore, analysis was performed on the accuracy of the data, provided by the system, while in the same way experiments, by which the system was being improved, were revealed.

1. ĮVADAS

1.1. Dokumento paskirtis

Šis dokumentas yra skirtas aprašyti sukurtos sistemos architektūrą, bei atlikti jos analizę. Taip pat dokumente yra atliekama teorinė analizė susijusi su orų prognozavimu, ligų nustatymu, bei meteorologinės stotelės aparatūrine įranga. Eksperimento dalyje yra pateikti atlikto tyrimo rezultatai, bei jų analizė.

1.2. Santrauka

Vis labiau populiarėjant išmaniems įrenginiams, didėja ir jų panaudojimo sritys. Daugumos išmaniųjų įrenginių tikslas ir užduotis yra palengvinti žmogaus gyvenimą bei sutaupyti jo laiko spendžiant kasdienes problemas.

Žmogaus veikla (darbas, ūkis, laisvalaikis ir t.t.) buvo ir yra priklausoma arba dalinai priklausoma nuo oro sąlygų. Dėl šios priežasties mokslas apimantis orų prognozavimą yra labai senas ir platus. Senovėje buvo taikomi elementarūs būdai prognozuoti orams, tačiau šiais laikais ši užduotį atlieka „super“ kompiuteriai. Naudojami sudėtingi matematiniai modeliai, algoritmai, palydovų teikiama informacija ir t.t. Šis mokslas ir toliau yra plėtojamas, nors ir retai tačiau atsiranda ir didelėse sistemose klaidų prognozuojant orus.

Nepaisant to, kad yra pastatyta nemažai profesionalių meteorologinių stočių iš kurių galima gauti faktinius duomenis visgi to neužtenka, o ypatingai ūkyje. Meteorologinė stotelė ūkininkams yra tikras pagalbininkas. Žinant tikslius faktinius orus stotelės buvimo vietoje (t.y. dirbamuose laukuose) leidžia ūkininkams planuoti savo laiką bei taupyti lėšas. Pavyzdžiui spendžiant iš dirvožemio temperatūros ir drėgmės (būtent ūkininko dirbamuose laukuose) pokyčių ūkininkas gali daryti išvadas apie pasėlių būseną – reikia juos pavasarį persėti ar ne.

Šio darbo metu buvo atlikta analizė egzistuojančių autonominių meteorologinių stotelių sprendimų, orų prognozavimo metodų ir algoritmų bei augalo ligų nustatymo principų. Buvo suprojektuota, surinkta ir suprogramuota autonominė meteorologinė stotelė kartu su sistema, kuri geba pateikti duomenis vartotojui realiu laiku iš stotelės, patikslinti orų prognozę stotelės būvimo vietoje, bei perspėti apie galimai gresiančią augalo ligą.

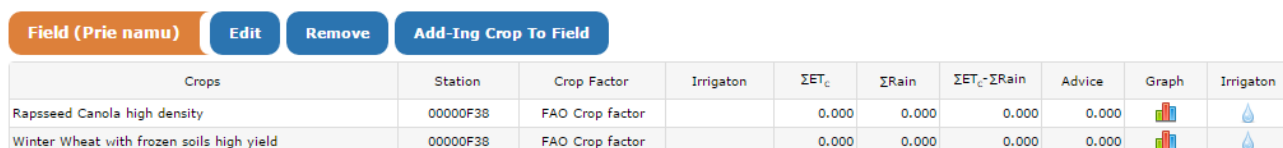
Šiame dokumente pateikiama projekcinė dalis, kurioje pateikiamas meteorologinės stotelės ir sistemos projektavimo eiga. Taip pat pateikti stotelės teikiamų duomenų ir prognozių tikslumo tyrimai, bei eksperimentai kuriais buvo bandoma sistemą patobulinti.

2. ANALITINĖ DALIS





2.1. Egzistuojantys sprendimai

Orų prognozavimo sistemų pasaulyje yra begalės. Sunku išsirinkti, kuri yra geriausia. Visos sistemos naudoja ne tik matematinius modelius ar algoritmus bet ir papildomus informacijos šaltinius: duomenis ir nuotraukas iš palydovų, kitų aplinkinių meteorologinių stočių duomenis, žmogaus įžvalgas ir t.t. Surasti, kuri sistema naudoja tik istorinius duomenis internete neteko. Yra tik tokius metodus ir mokslinius tyrimus aprašantys straipsniai, tačiau veikiančios sistemos surasti nepavyko.

Viena rimčiausių ir visame pasaulyje pripažintų sistemų augalų ligoms nustatyti, prognozuoti orus ir ne tik yra „iMetos“ sistema, kurią sukūrė „Pessl Instruments“. Prieš 30 metų buvo paleista į rinką pirmoji stotelė, kuri per visą gyvavimo laikotarpį buvo išstbulinta. 1994 jie pirmą kartą pritaikė augalų ligų modelius, todėl dabar tie modeliai ir metodikos yra labai gerai išvystyti ir stotelių savininkams pateikia didelio patikimumo prognozes [1].



The screenshot shows the iMetos system interface. At the top, there are four buttons: "Field (Prie namu)" in orange, and "Edit", "Remove", and "Add-Ing Crop To Field" in blue. Below the buttons is a table with the following columns: Crops, Station, Crop Factor, Irrigaton, ΣET_c , $\Sigma Rain$, $\Sigma ET_c - \Sigma Rain$, Advice, Graph, and Irrigaton. The table contains two rows of data:

Crops	Station	Crop Factor	Irrigaton	ΣET_c	$\Sigma Rain$	$\Sigma ET_c - \Sigma Rain$	Advice	Graph	Irrigaton
Rapseed Canola high density	00000F38	FAO Crop factor		0.000	0.000	0.000	0.000		
Winter Wheat with frozen soils high yield	00000F38	FAO Crop factor		0.000	0.000	0.000	0.000		

2.1.1 pav. „iMetos“ sistemos langas

2.1.1 paveiksle matome, jog vartotojas prie namų pastatęs stotelę ir pasirinkęs augalų ligų modelius gali stebėti jam būdingus parametrus. Sistema pati informuos vartotoją apie gresiančią augalo ligą. Ši sistema ir yra patraukli, nes visą darbą atlieka pati sistema, o ne vartotojas.

Sistema ne tik prognozuoja augalų ligas, ji turi ir daug kitų funkcijų. Duomenų peržiūros grafikus, orų prognozavimą, susietą su kitomis prognozes teikiančiomis sistemomis, bei daug kito naudingo vartotojui funkcionalumo.

2.2. Orų prognozavimas

Orų prognozavimas buvo labai populiaris sritis tyrimams pastarąjį dešimtmetį.

Dabartinės sistemos, kurios prognozuoja orus, naudoja daugybę informacijos šaltinių, tokių kaip: palydovinė informacija, duomenys iš kelių meteorologinių stočių, nuotraukomis iš palydovų ir t.t. Orų prognozavimas yra svarbus mokslas, kuris lydi žmogų nuo pat jo atsiradimo.

Yra metodų, kurie naudodami vien tik istorinius duomenis gali gauti orų prognozę („Like-Yesterday“ modelis), iš istorinių duomenų su kintamu vidurkio reguliavimu („ARMA“ modeliai) arba naudojantis statistikos funkcijas ir istoriniais duomenimis („Bin“ metodas).

Galima prognozuoti neuroniniais tinklais, bei anksčiau minėtais metodais. Jie visi suteiks orų prognozę, tačiau ne viskas priklauso tik nuo metodų. Šių metodų tikslumą veikia ir tai, kokio tikslumo duomenis pateiksime metodams bei kurioje pasaulio vietoje norime prognozuoti orus, kadangi klimato charakteristikų skirtumai įtakoja prognozes. Todėl yra labai sunku nustatyti, kuris metodas tam tikram klimatui yra geriausias.

Orų prognozavimas yra daugiausiai susijęs su oro sąlygomis netolimoje ateityje. Yra skirtingi būdai prognozuoti orus - nuo paprasto dangaus stebėjimo iki sudėtingų matematinių modelių. Prognozuoti orus yra esminis įvairių sistemų funkcionalumas. Kai kurios iš jų yra skirtos klimatui stebėti, sausras aptikti, sunkioms oro sąlygoms nuspėti, agrokultūrai ir produkcijai, planavimui energijos industrijoje, aviacijos industrijoje ir t.t. Įvairiais atvejais yra skirtingas poreikis prognozuoti tam tikrus orų parametrus. Parametrai ir jų diapazonas svyruoja priklausomai nuo vietovės. Orų parametrai bet kurią dieną turi sąryšį su orų būseną, kuri egzistavo tuo pačiu metu laiku ankstesniais metais ar ankstesnę savaitę.

Galima naudoti statistinį modelį, kuris suprojektuotas taip [2], kad gali numatyti kritulius ir temperatūrą pagal senus sukauptus duomenis panaudojant neuroninį tinklą. Naudojant dirbtinio ir neuroninio tinklo kombinaciją su genetiniu algoritmu galima gauti gana tikslią prognozę [3]. Geresnis būdas, kuris naudoja dirbtinį neuroninį tinklą su Levenberg Marquardt algoritmu, buvo pasiūlytas Isa et al. [4]. Be anksčiau paminėtų metodų „Fuzzy“ logika taip pat buvo naudojama orams prognozuoti. Duomenys yra skirstomi į tris „Fuzzy“ rinkinius, kurie gali būti prognozuojami pagal paprastas „Fuzzy“ taisykles. [5]

Visi anksčiau paminėti metodai naudoja istorinius duomenis. Iš to seka, kad duomenys turi būti tikslūs ir išmatuoti būtent toje vietovėje, kurioje norima atlikti orų prognozes. Šios technologijos naudoja dirbtinius neuroninius tinklus (ANN), kuriais buvo tikimasi pateikus tam tikrus duomenis, gauti teisingą rezultatą. Vis tik ryšys tarp duomenų matematiškai nebuvo apibrėžtas. Taip pat šie metodai turi trūkumą, kadangi nėra žinoma, kiek duomenų reikės, kad neuroninis tinklas galėtų apskaičiuoti teisingus rezultatus. Kitos, regresiją naudojančios technologijos yra išvardintos [6] [7].

Yra pagrindiniai būdai kaip galima prognozuoti orus naudojant istorinius duomenis:

- Naudojant neuroninį tinklą;
- Slankiojančio lango metodas;
- Naudojantis kitų sistemų teikiamomis orų prognozėmis;

2.2.1. Naudojant neuroninį tinklą

Orų prognozavimas šiais laikais tapo ne vien tik žmogaus darbas. Tai yra įvairiausių jutiklių bei galingų kompiuterių darbas. Vertingiausia orų prognozavimo informacija yra informacija apie tam tikrą vietovę, kurioje norima išprognozuoti orus. Kadangi orų kitimas priklauso nuo vietovės.

Prognozuojant temperatūrą reikia suprasti, jog prognozavimas kinta laiko atžvilgiu, pavyzdžiui, vienos valandos minimali ir maksimali temperatūra gali būti visos dienos režiai. Todėl labai keblu atlikti tiksliai, kas valandinę orų prognozę. Orų prognozavimo istorijoje šiam uždaviniui spręsti buvo panaudota daug neuroninių modelių.

Kaur [8] ir Maqsood [9] aprašo modelį, pagal kurį prognozuojama valandinė temperatūra, vėjo greitis ir santykinė oro drėgmė 24 valandoms į priekį. Testavimas ir bandymai buvo atlikti atskirai žiemą, pavasarį, vasarą ir rudenį. Šie autoriai palygino daugiasluoksnį neuroninį tinklą (MPL), „Elman“ rekurentinį neuroninį tinklą (ERNN), spindulinio pagrindo funkcinį tinklą (RBFN) ir „Hopfield“ tinklą (HFM). Autoriai pasiūlė vieną paslėptą sluoksnį su 72 neuronais MPL tinkle ir 2 paslėptus sluoksnius su 180 neuronais RBFN, kaip optimalias architektūras. RBFN turi geriausią greitaveiką. RBFN ir MLP turi panašų tikslumą, o MLP mokymosi laikas sueikvoja daugiau laiko.

Tyrimas, kurį atliko Sanjay Mathur [10] buvo sukonzentruotas į didžiausią ir mažiausią temperatūrų prognozavimą ir reliatyvų drėgmės prognozavimą naudojant laiko analizes. Buvo lyginami tiesioginiai ir statistiniai duomenys bei periodai. Minimaliai/maksimaliai temperatūros prognozei prognozuoti geriausias yra 15 savaičių duomenų laikotarpis. Per šias 15 savaičių klaidos tikimybė buvo tik 3 %. Šiuo rezultatu buvo įrodyta, jog naudojant sukauptus duomenis ir neuroninius tinklus - galima prognozuoti orus.

Neuroninio tinklo modelis

Tipinis neuroninio tinklo modelis orų prognozavimui turi ne mažiau kaip tris sluoksnius: sluoksnis duomenų įvedimui, paslėptas sluoksnis bei išvedimo sluoksnis [11]. Paslėptų sluoksnių ir neuronų skaičių pasirinkimas yra eksperimentinis dalykas. Neuroninio tinklo mokymui yra pasirenkamas Levenberg-Marquardt algoritmas. Literatūroje jis laikomas standartiniu algoritmu. Kai mokymas yra baigiamas, tinklui duodami duomenys, sudaryti iš maksimalios temperatūros (pirmų metų, antrų metų, ..., dešimtų metų). Kiekviename iteraciniame žingsnyje paklaidos signalas aktyvuoja mechanizmą, kuris patikslina svorius ir nutolimus neurone. Korekcijos vyksta iki to laiko, kol neuroninis tinklas grąžina rezultatus arčiausiai tikrosios temperatūros.

Duomenų bandymais ir rinkiniais yra apmokomas, bei testuojamas neuroninis tinklas. Rezultatai įrašomi ir vėliau lyginami, siekiant patikrinti, kaip gerai neuroninis tinklas veikia prognozuodamas temperatūrą.

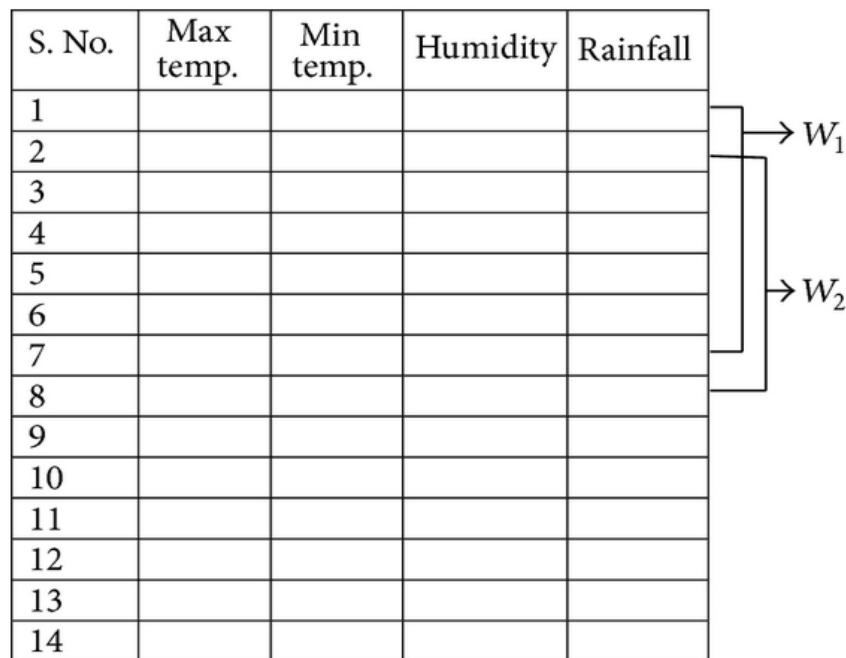
Tyrimai parodė, jog padidinus neuronų/sluoksnių skaičių, didėja greitaveika. Padidinus istorinių duomenų kiekį, greitaveika taip pat didėja. Šis metodas buvo iširtas prognozuojant aukščiausią temperatūrą, tačiau jį galima taikyti ir žemiausiai temperatūrai rasti, taip pat drėgmei, vėjo greičiui ir kitiems parametrų nustatyti, naudojant tą patį modelį su apmokymu ir paklaidos

nustatymu. Papildomos priemonės šio modelio optimizacijai gali būti įvairios makro ir mikro aplinkos faktoriai. Todėl yra būtina aplinkos analizė, kurioje norima naudoti šį metodą.

2.2.2. Slankiojančio lango metodas

Šis metodas siūlo vienos dienos prognozę. Norint, kad šis algoritmas veiktų, jam reikia prieš tai esančių 7 dienų ir dviejų savaitių to pačio laikotarpio tik praėjusių metų duomenų imties. Tarkime mums reikia sužinoti 2015 rugpjūčio 19d. orų prognozę. Tuomet šiam metodui reikia duomenų nuo rugpjūčio 12d iki rugpjūčio 18d, taip pat pastarųjų metų nuo rugpjūčio 12d iki rugpjūčio 25d. Šių duomenų pagalba algoritmas skaičiuos kiekvienos dienos šiame laikotarpyje prognozių galimas reikšmes. Šiose reikšmėse bus atsižvelgiama kokia yra maksimali, bei minimali temperatūra, drėgmė, bei kritulių kiekis. Taigi einamų metų duomenų dydis bus atvaizduojamas matricoje, kurios dydis – 7×4 . Praėjusiems metams matricos dydis bus – 14×4 [12].

Pirmas žingsnis bus išskaidyti matricą, kurios dydis 14×4 į „slankiojančius langus“. Taigi 8 slankiojančių langų dydžiai bus 7×4 (kiekvienas). Pagrindinė slankiojančių langų idėja apteikta 2.2.1 paveiksle.



2.2.1 pav. W_1 atvaizduoja pirmąjį langą, W_2 antrąjį langą

Sekančiame žingsnyje reikia palyginti praeitų metų duomenis (langus) su šių metų duomenimis. Labiausiai atitinkantis „langas“ yra pažymimas, jog gali teigti prognozę. Geriausiai atitinkančio lango atpažinimui yra naudojamas „Euklido“ atstumo metodas [13]. Priežastis, kodėl buvo pasirinktas „Euklido“ atstumo metodas yra ta, jog jis gerai parodo lango ir variacijų panašumą. Toliau yra kintamieji, kurie naudojami oro sąlygų prognozėms:

1. Vidurkis - vidurkis dienos oro sąlygų, tai yra: maksimali ir minimali temperatūra, drėgmė ir kritulių kiekis. Po to kai yra rodikliai sudedami atskirai jie yra padalinami iš dienų skaičiaus.

$$\text{vidurkis} = \text{parametrų suma} / \text{dienų skaičiaus}$$
2. Variacija - paskaičiuoti variacija kiekvienos dienos panaudojus kiekvieno parametro skirtumą. Tai gali pasakyti, kaip kitos dienos oras yra susijęs su vakarykščiu oru.
3. Euklido atstumas - jis lygina duomenų variacijas einamųjų ir praėjusių metų.

Šiais rodikliais mes galime matematiškai modeliuoti apibrėžtas priklausomybes.

Slankiojantis langas naudojamas prognozuoti „n“ orų kondicijas ($W_1, W_2, W_3, W, \dots, W_n$).

Tai pavaizduota žemiau apseiktame algoritme:

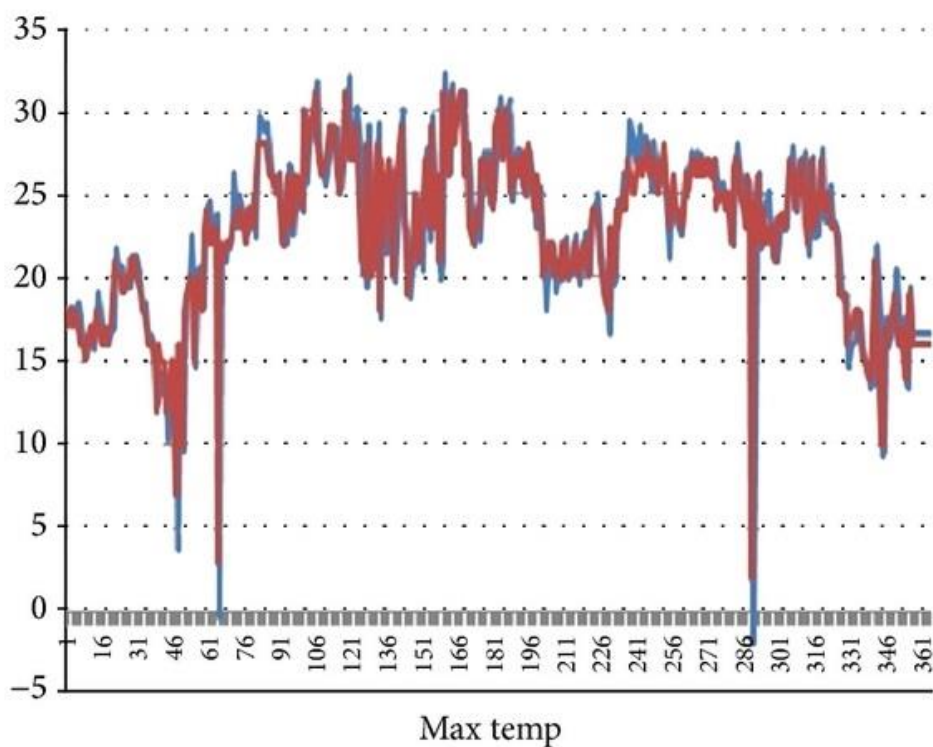
Step 1. Take matrix “CD” of last seven days for current year’s data of size 7×4 .
Step 2. Take matrix “PD” of fourteen days for previous year’s data of size 14×4 .
Step 3. Make 8 sliding windows of size 7×4 each from the matrix “PD” as $W_1, W_2, W_3, \dots, W_8$
Step 4. Compute the Euclidean distance of each sliding window with the matrix “CD” as $ED_1, ED_2, ED_3, \dots, ED_8$
Step 5. Select matrix W_i as
 $W_i = \text{Correponding_Matrix}(\text{Min.}(ED_i))$
 $\forall i \in [1, 8]$
Step 6. For $k = 1$ to n
 (i) For WC_k compute the variation vector for the matrix “CD” of size 6×1 as “VC”.
 (ii) For WC_k compute the variation vector for the matrix “PD” of size 6×1 as “VP”.
 (iii) $\text{Mean}_1 = \text{Mean}(\text{VC})$
 (iv) $\text{Mean}_2 = \text{Mean}(\text{VP})$
 (v) Predicted Variation “V” = $(\text{Mean}_1 + \text{Mean}_2)/2$
 (vi) Add “V” to the previous day’s weather condition in consideration to get the predicted condition.
Step 7. End

2.2.2 pav. Slankiojančio lango algoritmas

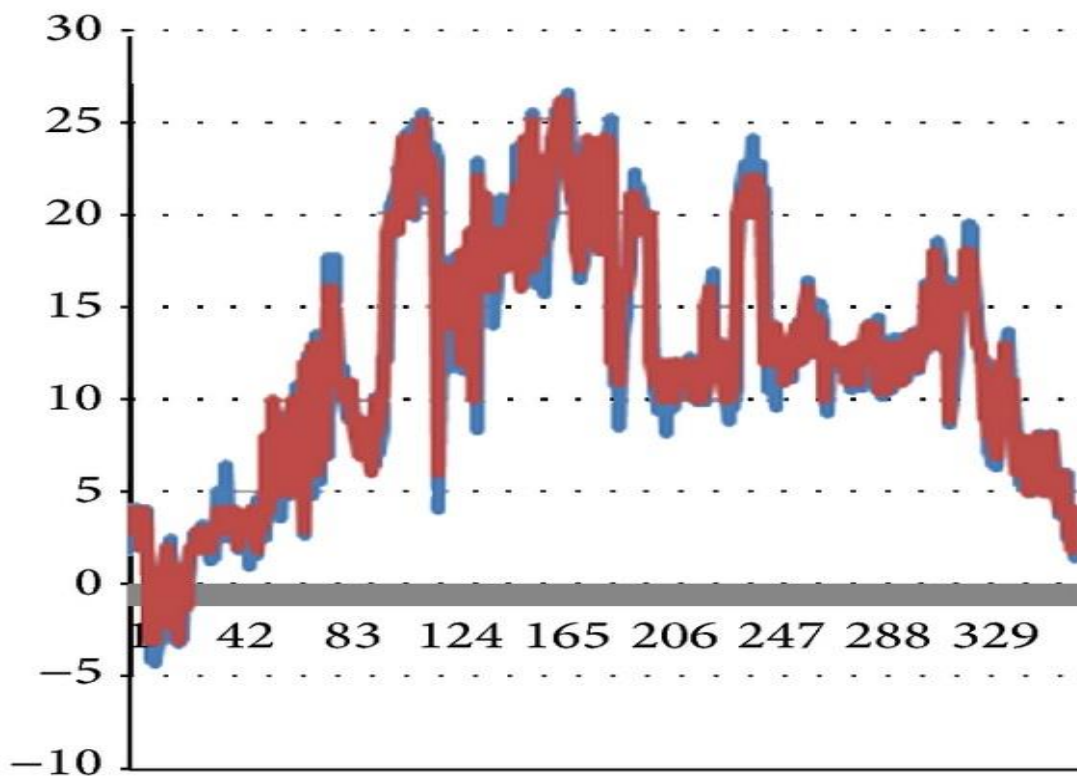
Naudojant slankiojančių langų metodika, pagrindinė logika yra ta, jog oro sąlygos vyraujančios tam tikrą dienos trukmę šiais metais, nebūtinai gali būti tokia pat ir praėjusių metų dienoje. Pavyzdžiui orų sąlygos vyraujančios pirmoje savaitėje 2015 metų gali neegzistuoti pirmąją savaitę 2014 metais. Panašios orų sąlygos galėjo būti ir praėjusiais metais, tačiau nebūtinai tą pačią savaitę. Didžiausia tikimybė yra rasti panašias orų sąlygas yra dviejų savaičių intervale.

Šis algoritmas buvo ištestuotas su duomenų aibe nuo 2006 m. iki 2010 m. Algoritmas buvo įgyvendintas ir ištestuotas su „Matlab“ įrankiu. Taigi, algoritmas atsižvelgdamas į praėtų metų duomenis gali prognozuoti orus. Šis algoritmas buvo bandomas trejus metus. Galima padaryti išvadas, jog algoritmo mokymasis yra kontroliuojamas. T.y. jei duomenis paduosime tikslius, jis mums prognozuos tikslesnius orus, jei duomenys bus iškraipyti - algoritmas teiks klaidingus rezultatus.

Žemiau pateiktuose paveiksluose [12] galime matyti, kaip šiam algoritmui sekėsi prognozuoti orus. Raudona linija reiškia algoritmo nuspėjamą prognozę, o mėlynas kokia iš tikrųjų tą dieną orų parametrai buvo.

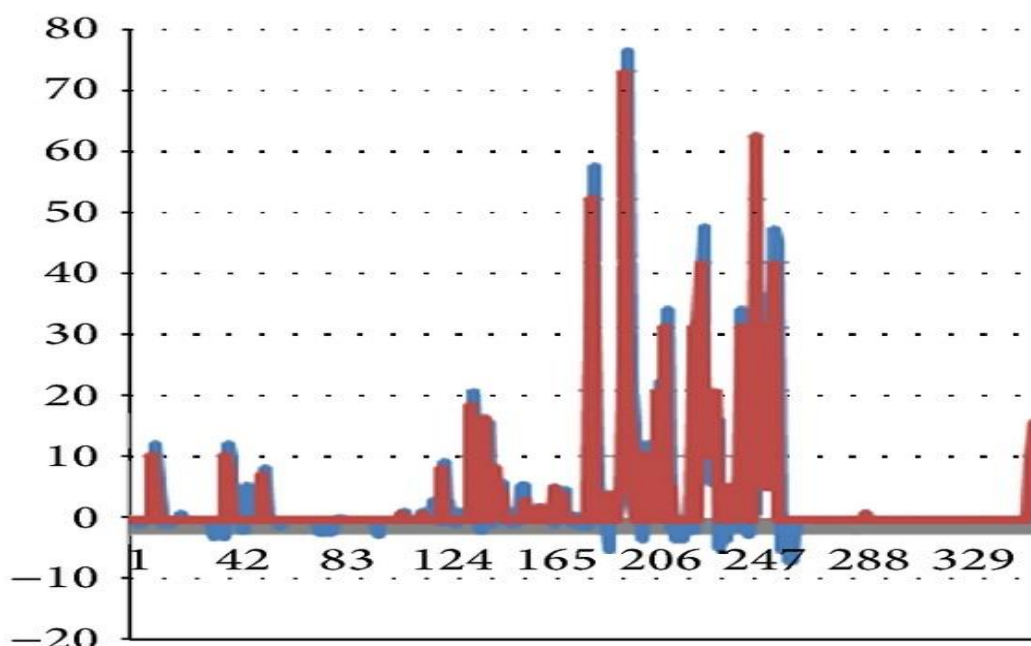


2.2.3 pav. Maksimalios temperatūros algoritmo prognozių grafikas [12]

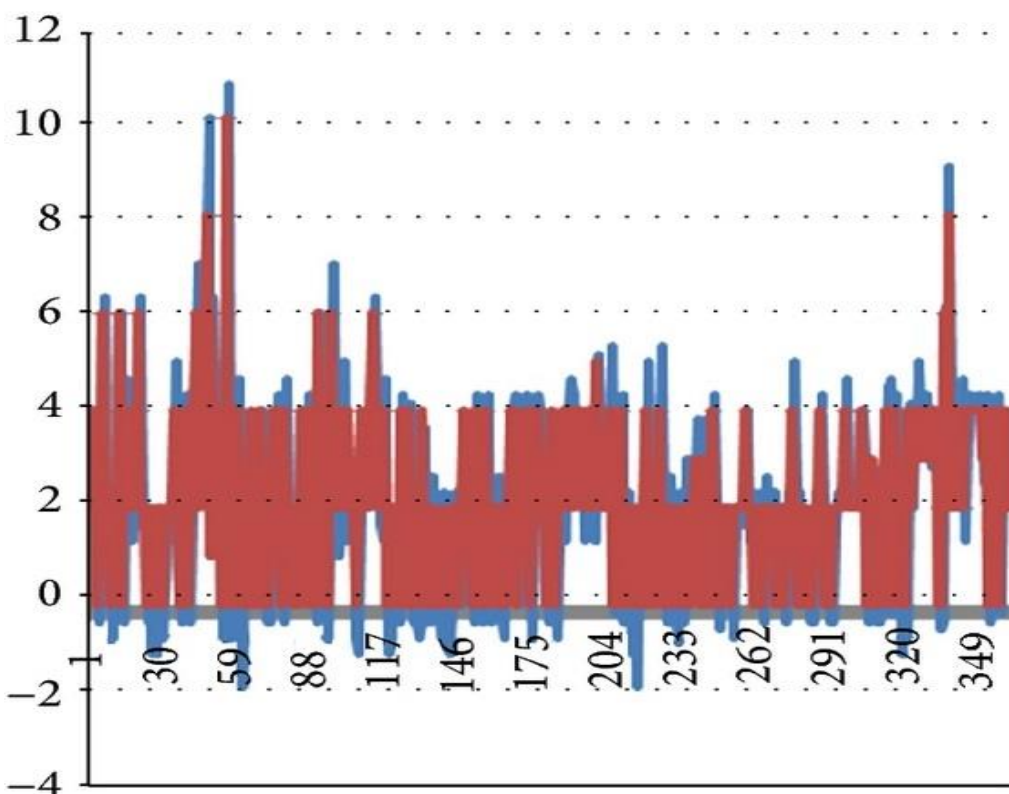


2.2.4 pav. Minimalios temperatūros algoritmo prognozių grafikas [12]

Iš temperatūros grafikų matome, kad algoritmas gana tiksliai išprognozuoja temperatūrą. Didžiausias skirtumas tarp tikros ir išprognozuotos temperatūros yra vos keli laipsniai.



2.2.5 pav. Drėgmės algoritmo prognozių grafikas [12]



2.2.6 pav. Kritulių kiekio algoritmo prognozių grafikas [12]

Visi šie grafikai parodo, kokių tikslumu šis algoritmas veikia. Visų grafikų duomenis apskaičiavus ir išvedus tikslumą, juos pateikiame lentelėje.

2.2.1 lentelė Algoritmo prognozių tikslumo rezultatai

Mėnesis	Išprognuozuotų orų tikslumas
Sausis	97.24%
Vasaris	88.94%
Kovas	92.25%
Balandis	98.17%
Gegužė	93.49%
Birželis	99.02%
Liepa	87.19%
Rugpjūtis	78.83%
Rugsėjis	79.23%
Spalis	94.35%
Lapkritis	98.78%
Gruodis	95.73%

2.2.3. Naudojantis kitų sistemų teikiamomis orų prognozėmis

Šis būdas yra naudojamas pasitelkiant kelių tarptautinių orų prognozavimo sistemomis.

Temperatūros prognozavimas

Naudojant šį būdą, reikia išnagrinėti stotelės buvimo vietą. Tai svarbu todėl, kad duomenys yra papildomai renkami iš aplinkinių sistemų, kurios teikia orų prognozes. Išnagrinėjus aplinkinių meteorologinių stočių teikiamus duomenis ir jų tikslumą, pasirenkamos kelios stotys. Taip pat yra atsižvelgiama į žemės reljefą. Žiūrima iš kurių pusių yra kalnai arba žemumos, tai suteiks informacijos apie vėjo greičio ir krypties toje vietovėje būvimą.

Išanalizavus vietovę ir duomenis teikiamas sistemas, galima apskaičiuoti preliminarią temperatūrą. Temperatūros prognozės yra skaičiuojamos pagal formulę:

$$\hat{T}_i = \frac{\sum_{i=1}^4 w_i (a_i \hat{X}_{ti} + b_i)}{\sum_{i=1}^4 w_i}$$

Į formulę įeina praeitos dienos temperatūra, bei trejų kitų pasirinktų stočių ir asmeninės stotelės duomenys. a ir b yra paklaidos, kurios paskaičiuotos atsirenkant meteorologines stotis. w yra svoris, kuris nustatomas pagal kiekvieno šaltinio tinkamumą prognozavimui. W_i yra mūsų stotelės duomenys.

Naudojantis kitų sistemų teikiamais duomenimis, visada gali atsitikti taip kad sistema mums reikiamų duomenų nepateiks. Tuomet šis metodas klaidingai paskaičiuos reikiamus rezultatus. Patartina turėti keletą papildomai išanalizuotų stočių, kurias reikalui esant galima būtų panaudoti ekstra atveju.

Toliau pritaikius formulę galime gauti preliminarią temperatūros prognozę.

Prognozavimas pagal saulės radiaciją

Trumpo laiko tarpo prognozių teikimas pagal saulės radiacijos lygį iš aplinkinių sistemų yra daug sudėtingesnis, nei temperatūros prognozavimas. Saulės radiacija, kitaip nei temperatūra, nėra pateikiama orų prognozavimo sistemose. Tačiau sistemos teikia tokius duomenis, kaip slėgį, drėgmę, kritulius, debesų kiekį ir t.t.

Pagal šiuos parametrus yra sudaryta priklausomybė su saulės radiacija, ji pateikta žemiau esančioje lentelėje.

2.2.2 lentelė Oro sąlygos būseną pagal saulės radiaciją

Saulės radiacijos lygis	Oro apibūdinimas	Debesuoto ir giedro dangaus santykis
11	Giedra/Saulėta	>0.95
10	Protarpiais saulėta	0.85~0.95
9	Dalinai debesuota	0.75~0.85
8	Vidutinio lygio debesys	0.65~0.75
7	Žemo lygio debesys	0.55~0.65
6	Rūkas	0.45~0.55
5	Dulksna	0.35~0.45
4	Lengvas lietus/sniegas	0.25~0.35
3	Šlapdriba	0.15~0.25
2	Didelis lietus/sniegas	0.05~0.15
1	Naktis	<0.05

Stotelėje turint saulės radiacijos jutiklį, šiuo principu galima vadovautis. Dažnai yra naudojami kelių orų parametrų rodikliai, norint išpranašauti orus. Saulės, slėgio, temperatūros ir vėjo

greičio ir krypties rodiklių informacija galima panaudoti orų prognozavimui, kadangi su šiais duomenimis galima atlikti prognozę. Vienas iš pavyzdžių būtų: jei žiemą saulėta (giedra), šalta ir pučia šiaurinis vėjas, galima spėti, jog naktis bus 30% šaltesnė nei diena.

2.3. Ligų nustatymas

Augalas kaip ir gyvas organizmas, gali susirgti įvairiomis ligomis. Tačiau ne tik ligos lemia augalo augimą. Yra kenkėjai, bei nepalankios oro sąlygos.

Esant neigiamai temperatūrai dauguma augalų žūsta, o esant šiltam ir drėgnam orui, yra palanku visiems mikroorganizmams daugintis. Iš čia atsiranda visokie kenkėjai, kurie tiesiog sugraužia augalą.

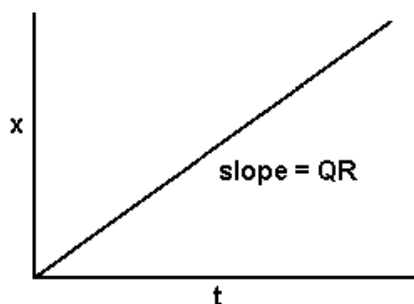
Norint apsisaugoti nuo visų šių faktorių, yra būtinas augalo stebėjimas. Kiekvienas augalas turi aibę savo kenkėjų, nepalankių oro sąlygų ar ligų. Todėl kiekvienam augalui yra sudaromas atskiras modelis. Modelis, kuris saugo visą informaciją, apie augalui nepalankias sąlygas.

Modeliai kiekvienam augalui ir kenkėjams yra labai dideli ir smulkmeniški. Žmogui numatyti, kad augalas gali neišgyventi, yra labai sunku. Tai pastebima kuomet jau augalas būna stipriai pažeistas. Todėl čia į pagalbą žmogui ateina kompiuteris, kuris galės numatyti apie tokių įvykių galimybę.

Dažniausiai modeliai seka augalo augimo aplinką, kuri įtakoja ligų ir kitų neigiamų faktorių atsiradimą. Aplinka ir laikas yra pagrindiniai susiję faktoriai ligų nustatyme. [14]

2.3.1. Linijinis ligos progresas

Jei ligos progresas yra linijinis, tai ligos pažangos kreivė yra pastovi. Tuomet galime apibūdinti epidemiją su linijine ligos eiga, naudojanti diferencialinę lygtį: $dx/dt = QR$, kur dx yra labai maža ligos priaugio dalis, dt yra labai mažas laiko tarpas, Q yra pagrindinio kultūros sėjimo suma, R yra proporcingumo konstanta, kuri parodo vieno vieneto užsikrėtimą. R priklauso nuo daugelio veiksnių, tokių kaip ligos agresyvumo, augalo jautrumo ir t.t. Integravę diferencialinę lygtį gauname: $x = QRt$. Ligos progresą galime stebėti žemiau pateiktame grafike.

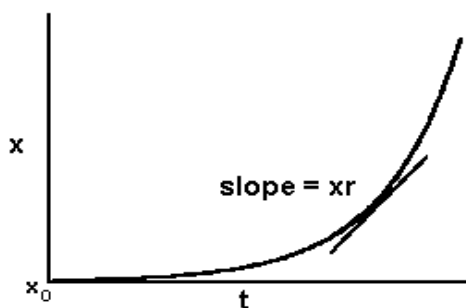


2.3.1 pav. Linijinis ligos progreso grafikas

2.3.2. Policiklinis ligos progresas

Tai yra toks pats ligos progresas kaip ir linijinis, tačiau užkrečiamų augalų kiekis didėja nuo to, kiek kultūroje yra užkrėstų augalų. Tai yra, kuo daugiau augalų bus užkrėsta, tuo sparčiau užsikrės kiti.

Šį progresą galime aprašyti lygtimi: $dx/dt = xr$. Čia x yra dimensija nuo nulio iki vieneto, r yra pastovus ir priklauso nuo ligos progresyvumo. Tuomet $x = x_0 e^{rt}$, kur x_0 yra proporcija nuo epidemijos pradžios iki laiko t . Tuomet apkrėstų augalų kiekis, priklausomybė nuo laiko grafikas yra:

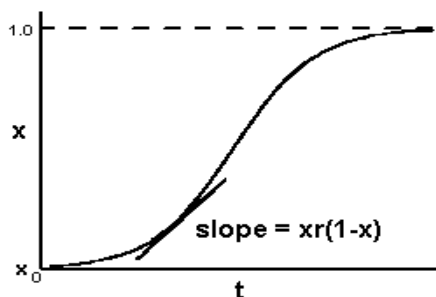


2.3.2 pav. Policiklinis ligos progreso grafikas

2.3.3. Maksimalus ligos progresas

Anksčiau išvardinti ligos progreso modeliai yra begaliniai, ko realiame gyvenime būti negali, nes kiekviename lauke yra ribotas skaičius augalų. Pastarieji modeliai taip pat yra taikomi, kuomet laukai yra dideli, ir pravartu žinoti ligos progreso būseną.

Maksimalaus ligos progreso lygtis yra $dx/dt = xr(1-x)$. Čia kintamieji yra naudojami tie patys kaip ir praeitose lygtyse. Grafikas šio modelio atrodo taip:



2.3.3 pav. Maksimalaus lygio ligos progreso grafikas

2.4. Stotelės aparatūrinė įranga

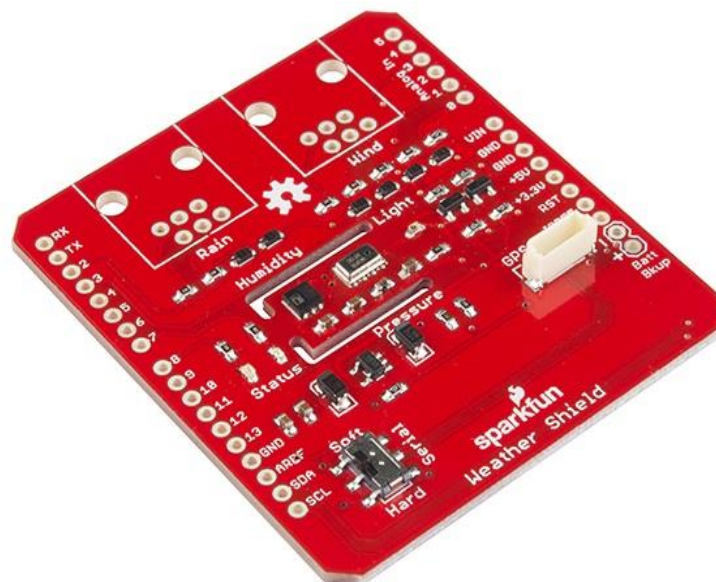
Kadangi stotelė autonominė, reikia pasverti jos visas galimybes. Natūralu, jog laukuose neturėsime jokių maitinimo šaltinių, tuomet belieka vienintelė išeitis - naudoti saulės energiją.

Stotelė turėtų būti mobili, ją galima bus perstatyti, todėl negalime turėti didelių saulės baterijų, o naudojant neuroninius tinklus ir algoritmus, kompiuterio apkrovos bus didelės. Todėl vienintelė išeitis, stotelėje tiesiog rinkti duomenis iš aplinkos, o visus skaičiavimus atlikti serveryje.

Pasirinkome „Raspberry Pi“ mini kompiuterį kartu su „Arduino Uno R3“ mikrokontroleriu.

2.4.1. Jutikliai

Labiausiai atitinkantis poreikius yra „SparkFun“ siūloma orų stebėjimo plokštė [15] (2.4.1 pav.). Šioje plokštėje yra įmontuoti drėgmės, temperatūros, saulės radiacijos, slėgio jutikliai. Taip pat galimybė prijungti GPS, kritulių, bei vėjo jutiklius. Taip pat pasirinkome su šia plokšte siūlomus, kritulių, vėjo greičio ir krypties jutiklius.



2.4.1 pav. „SparkFun“ orų plokštė „Arduino“ mikrokontroleriui



2.4.2 pav. Vėjo krypties, greičio ir kritulių jutikliai



2.4.3 pav. Dirvožemio temperatūros ir drėgmės jutiklis

Plokštės pavaizduotos (2.4.1 pav.), papildomų jutiklių pavaizduotų [16] (2.4.2 pav.), bei dirvožemio temperatūros ir drėgmės (2.4.3 pav.) jutiklių pagalba mes turėsime visus mums reikiamus duomenis, norint išspręsti šio darbo uždavinį. Jutikliai komunikuoja per I2C protokolą. Aukščiau paiminėta plokštė yra suderinama su „Arduino“ mikrokontroleriu.

2.5. Išvados

Atlikus analizę buvo rasta daug įvairių problemos sprendimo būdų. Ne visi jie atitiko mūsų reikalavimams. Kadangi mūsų tikslas yra sugebėti lokaliai, išprognuoti orus, bei augalų ligas. Analizėje pateikėme pačius labiausiai mūsų darbui tinkamus metodus, bei algoritmus.

Atliekant analizę buvo pastebėta, jog ši mokslo kryptis yra išties labai plati. Taip yra todėl, kad žmonės jau nuo pat atsiradimo bandė prognozuoti orus, kad apsaugotų save ar galėtų planuoti savo darbus ar laisvalaikį. Šis mokslas yra be galo sunkus, kadangi ne retai tiek sistemos, tiek žmonės negali numatyti ir išprognuoti orų 100% tikslumu. Yra sistemos, kurios prognozuoja net mėnesiui į priekį, tačiau tos prognozės nėra tikslios. Tiksliausios prognozės vis dar yra 1-2 parų į priekį.

Atikus analizę ir įvertinus visus metodų privalumus, orų prognozėms mes pasirinkome slankiojančio lango metodą, kurį detalai ir išanalizavome. Šis metodas iš visų išnagrinėtų metodų orų prognozavime mums tiko nes mes turime tik istorinius duomenis. Be to algoritmas atliktame tyrime parodė gan aukštus rezultatus. Šį algoritmą modifikuosime ir pabandysime jį padaryti lankstesnį. Sistemoje sukursime funkcionalumą, kurio pagalba galima bus patvirtinti prognozę (jei ji atitiks realybę). Jei vartotojas nepatvirtins prognozės, sekantį kartą prognozuojant orus tuo pat metu bus naudojamas skirtingas slankiojančio lango dydis. Taip pabandysime pritaikyti algoritmą kintamam ir turinčiam keturis metų laikus Lietuvos klimatui.

Ligų prognozavimui pasirinkome neuroninį tinklą su ligos progreso stebėjimu. Sujungus neuroninį tinklą ir šiuos progresus, galėsime tiksliau nustatysi apie galimos ligos ar kenkėjų atsiradimą. Taip pat sukursime ligų modelius, kurių pagalba galima bus dar tiksliau išprognuoti

galimą kenkėjų ar ligos atsiradimą. Šie modeliai savyje laikys informaciją apie orų charakteristikas ir laikotarpius, būdingus ligos/kenkėjų atsiradimui. Apie augalų ligas yra daug informacijos internete, kur mokslininkų atliktuose tyrimuose yra išdėstomos priežastys, dėl kurių augalai užsikrečia tam tikromis ligomis.

3. PROJEKTINĖ DALIS

3.1. Sistemos paskirtis

Orų stotelių pasaulyje yra nemažai. Lietuvoje jų yra apie keliolika. Tačiau visiems yra žinoma, kad oras yra permainingas dalykas ir tos permainos atsispindi net mažose distancijose. Dažnai orų tinklalapiai teikia ne realius tam tikrai vietai orus, bet prognozuojamus tam tikrai apskričiai. Todėl žmonės, kuriems yra svarbu gauti tikslias oro sąlygas, privalo statyti orų stotelę savo norimoje vietoje.

Kasdieniniame gyvenime kartais nepakanka vien termometro. Norima sužinoti koks yra slėgis, drėgmė, vėjo greitis arba jo kryptis. Peržiūrėti orų kitimą tam tikrame laikotarpyje ir t.t. Stotelė yra aktuali žmonėms, kurie užsiima ūkine veikla arba dirbantiems uždarose patalpose, bet tuo pačiu metu jiems aktualu kas vyksta lauke (tarkime jie džiovina savo dirbinius atviroje erdvėje ar užsiima kita veikla).

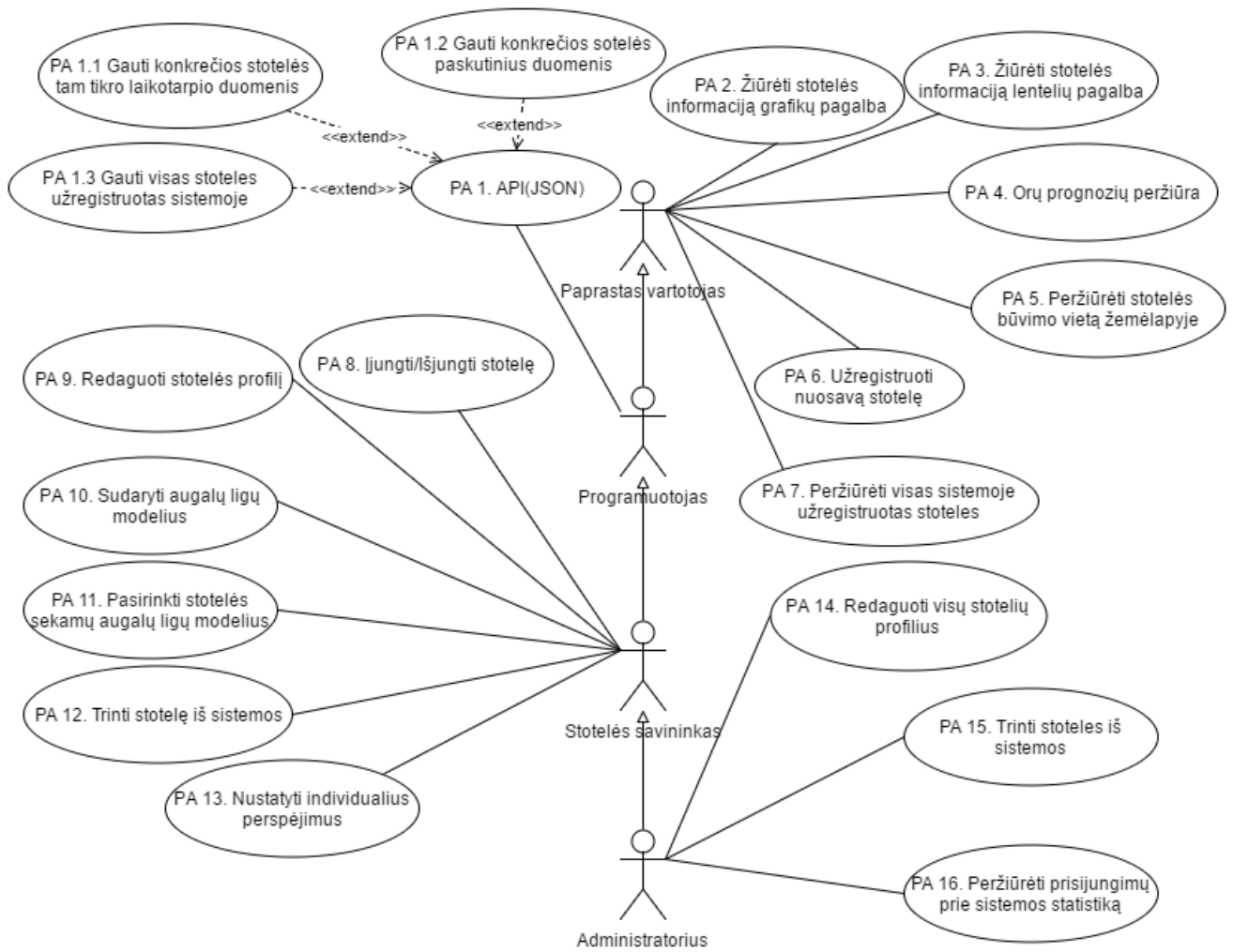
Ūkininkystėje tai yra tikras pagalbininkas, kuris sutaupys daug laiko ir pastangų. Ūkininkas žinodamas kas vyksta jo laukuose gali priimti sprendimus, net nežengdamas iš namų. Tiesiog prisijungęs iš kompiuterio/mobilaus įrenginio prie sistemos matys visą jam aktualią informaciją. Stotelei pranašaujant orus ir ligų atsiradimo tikimybes ūkininkams dar labiau pagelbėja. Ypač svarbus yra ligų atsiradimo prognozavimas, kuris gali padėti apsaugoti augalus.

Tačiau ne tik ūkininkystėje šios stotelės yra aktualios. Sklaidymas, buriavimas ir kitos sporto šakos, kur yra svarbūs tiksloje vietoje vėjo greitis bei kryptis. Kasdieniniame naudojime - gauti perspėjimus apie prasidėjusį liūtį, kad vartotojas laiku spėtų uždaryti automobilio langus ar susirinkti skalbinius ir t.t.

Sistemos sukūrimas, kurioje galima registruoti neribotą skaičių stotelių, suteiks informaciją visiems interneto vartotojams. Entuziastai ir kiti stotelių savininkai registruos savo stoteles, kurios teiks informaciją į serverį. Tokiu būdu jos padengs didelę dalį Lietuvos ploto, o tai suteiks įvairiems interneto vartotojams aktualią informaciją šalia jų buvimo vietos.

3.2. Sistemos panaudojimo atvejai

3.2.1. Sistemos panaudos atvejų diagrama



3.2.1 pav. Panaudos atvejų diagrama

3.2.2. Panaudojimo atvejų scenarijai

3.2.1 lentelė PA 1.1 Gauti konkrečios stotelės tam tikro laikotarpio duomenis

Panaudos atvejis „PA 1.1 Gauti konkrečios stotelės tam tikro laikotarpio duomenis”	
Aprašymas	Gauti konkrečios stotelės tam tikro laikotarpio duomenis.
Prieš sąlyga	Vartotojas nori gauti konkrečios stotelės tam tikro laikotarpio duomenis JSON formatu.
Aktorius	Programuotojas, paprastas vartotojas, Stotelės savininkas, Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas įrašo į HTTP užklausą atitinkamus parametrus.2. Kreipiasi į serverį.3. Gauną atsakymą iš serverio JSON formatu.4. Duomenis panaudoja savo sistemoje ar kitaip juos apdoroja.5. Baigiamas PA.	
Po sąlyga	Vartotojas API pagalba gauna konkrečios stotelės duomenis.

3.2.2 lentelė PA 1.2 Gauti konkrečios stotelės paskutinius duomenis

Panaudos atvejis „PA 1.2 Gauti konkrečios stotelės paskutinius duomenis”	
Aprašymas	Gauti konkrečios stotelės paskutinius duomenis API pagalba.
Prieš sąlyga	Vartotojas, kuris nori gauti tam tikros stotelės duomenis JSON formatu.
Aktorius	Programuotojas, paprastas vartotojas, Stotelės savininkas, Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas įrašo į HTTP užklausą atitinkamus parametrus.2. Kreipiasi į serverį.3. Gauną atsakymą iš serverio JSON formatu.4. Duomenis panaudoja savo sistemoje ar kitaip juos apdoroja.5. Baigiamas PA.	
Po sąlyga	Vartotojas API pagalba gauna konkrečios stotelės paskutinius duomenis.

3.2.3 lentelė PA 1.3 Gauti visas sistemoje užregistruotas stoteles

Panaudos atvejis „PA 1.3 Gauti visas sistemoje užregistruotas stoteles”	
Aprašymas	Gauti visas sistemoje užregistruotas stoteles ir jų informaciją
Prieš sąlyga	Vartotojas nori gauti visas sistemoje užregistruotas stoteles ir jų informaciją JSON formatu.
Aktorius	Programuotojas, paprastas vartotojas, Stotelės savininkas, Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas įrašo į HTTP užklausą atitinkamus parametrus. 2. Kreipiasi į serverį. 3. Gauną atsakymą iš serverio JSON formatu. 4. Duomenis panaudoja savo sistemoje ar kitaip juos apdoroja. 5. Baigiamas PA. 	
Po sąlyga	Vartotojas API pagalba gauna visų stotelių užregistruotų sistemoje informaciją.

3.2.4 lentelė PA 2 žiūrėti informaciją grafikų pagalba

Panaudos atvejis „PA 2 žiūrėti informaciją grafikų pagalba”	
Aprašymas	Stotelės informacijos peržiūra grafikų pagalba.
Prieš sąlyga	Vartotojas nori peržiūrėti pasirinktos stotelės informaciją grafikų pagalba.
Aktorius	Paprastas vartotojas, Stotelės savininkas, Administratorius.
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas pasirenka norimą žiūrėti stotelę sistemoje. 2. Pasirenka kokią informaciją nori peržiūrėti (vėjo greitį/kryptį, slėgį, temperatūrą, drėgmę ir t.t) 3. Pasirenka laikotarpį. 4. Spaudžia generuoti grafiką. 5. Baigiamas PA. 	
Po sąlyga	Vartotojas peržiūrėjo norimos stotelės informaciją grafiko(-ų) pagalba.

3.2.5 lentelė PA 3 žiūrėti stotelės informaciją lentelių pagalba

Panaudos atvejis „PA 3 Žiūrėti stotelės informaciją lentelių pagalba”	
Aprašymas	Stotelės duomenų peržiūrėjimas lentelių pagalba.
Prieš sąlyga	Vartotojo, nori peržiūrėti pasirinktos stotelės informaciją lentelės pagalba.
Aktorius	Paprastas vartotojas, Stotelės savininkas, Administratorius.
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas pasirenka norimą žiūrėti stotelę sistemoje.2. Pasirenka lentelę.3. Jeigu reikia rikiuoja/filtruoja duomenis joje.4. Baigiamas PA.	
Po sąlyga	Vartotojas peržiūrėjo stotelės informaciją lentelių pagalba.

3.2.6 lentelė PA 4 orų prognozės peržiūra

Panaudos atvejis „PA 4 orų prognozės peržiūra”	
Aprašymas	Orų prognozės peržiūra.
Prieš sąlyga	Vartotojas nori peržiūrėti pasirinktos stotelės apylinkės orų prognozę.
Aktorius	Paprastas vartotojas, Stotelės savininkas, Administratorius.
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas pasirenka norimą stotelę.2. Pasirenka orų prognozes.3. Peržiūri prognozes.4. Baigiamas PA.	
Po sąlyga	Vartotojas peržiūrėjo stotelės orų prognozę.

3.2.7 lentelė PA 5 Peržiūrėti stotelės būvimo vietą žemėlapyje

Panaudos atvejis „PA 5 Peržiūrėti stotelės būvimo vietą žemėlapyje”	
Aprašymas	Žiūrėti stotelės vietą žemėlapyje
Prieš sąlyga	Vartotojas nori peržiūrėti kurioje vietoje žemėlapyje yra stotelė.
Aktorius	Paprastas vartotojas, Stotelės savininkas, Administratorius.
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas atidaro sistemoje esantį žemėlapi. 2. Susiranda ją žemėlapyje arba pasinaudoja paieška. 3. Baigiamas PA. 	
Po sąlyga	Vartotojas peržiūrėjo stotelės būvimo vietą žemėlapyje.

3.2.8 lentelė PA 6 Užregistruoti nuosavą stotelę

Panaudos atvejis „PA 6 Užregistruoti nuosavą stotelę”	
Aprašymas	Vartotojo stotelės registracija.
Prieš sąlyga	Vartotojo, kuris nori užregistruoti savo stotelę sistemoje, registracija.
Aktorius	Paprastas vartotojas, Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas atidaro registracijos formą. 2. Vartotojas suveda reikiamą registracijai informaciją. 3. Vartotojas spaudžia mygtuką „Registruotis”. 4. Baigiamas PA. 	
Po sąlyga	Vartotojas užregistruoja savo stotelę sistemoje.

3.2.9 lentelė PA 7 Peržiūrėti visas sistemoje užregistruotas stoteles

Panaudos atvejis „PA 7 Peržiūrėti visas sistemoje užregistruotas stoteles”	
Aprašymas	Visų sistemoje užregistruotų stotelių peržiūra.
Prieš sąlyga	Vartotojas nori peržiūrėti visas užregistruotas sistemoje stoteles.
Aktorius	Paprastas vartotojas, Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas atidaro pagrindinį sistemos langą.2. Pasirenka meniu punktą “Visos stotelės sistemoje”.3. Peržiūri sąrašą.4. Baigiamas PA.	
Po sąlyga	Vartotojas peržiūrėjo visas sistemoje užregistruotas stoteles.

3.2.10 lentelė PA 8 Išjungti/Ijungti stotelę

Panaudos atvejis „PA 8 Išjungti/Ijungti stotelę”	
Aprašymas	Stotelės įjungimas arba išjungimas.
Prieš sąlyga	Stotelės savininkas nori įjungti arba išjungti stotelę
Aktorius	Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none">1. Vartotojas įjungia arba išjungia pagrindinį stotelės bloką mygtuko pagalba.2. Baigiamas PA.	
Po sąlyga	Stotelės savininkas įjungė arba išjungė stotelę.

3.2.11 lentelė PA 9 Redaguoti stotelės profilį

Panaudos atvejis „PA 9 Redaguoti stotelės profilį”	
Aprašymas	Stotelės profilio redagavimas.
Prieš sąlyga	Stotelės savininkas nori redaguoti stotelės informaciją.
Aktorius	Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas atidaro stotelės redagavimo formą. 2. Redaguoja informaciją. 3. Spaudžia mygtuką išsaugoti. 4. Baigiamas PA. 	
Po sąlyga	Stotelės savininkas paredagavo stotelės informaciją.

3.2.12 lentelė PA 10 Sudaryti augalų ligų modelius

Panaudos atvejis „PA 10 Sudaryti augalų ligų modelius”	
Aprašymas	Augalo ligos modelio sudarymas.
Prieš sąlyga	Vartotojas nori sudaryti augalo ligos modelį.
Aktorius	Stotelės savininkas, Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Atidaromas ligos modelio sudarymo langas. 2. Suvedama visa informacija, apie augalo ligos atsiradimo priežastį. 3. Suvedami parametrai, prie kurių liga gali atsirasti. 4. Išsaugo ligos modelį. 5. Baigiamas PA. 	
Po sąlyga	Vartotojas sukuria augalo ligos modelį.

3.2.13 lentelė PA 11 Pasirinkti stotelės sekamų augalų ligų modelius

Panaudos atvejis „PA 11 Pasirinkti stotelės sekamų augalų ligų modelius”	
Aprašymas	Galimos ligos pasirinkimas
Prieš sąlyga	Vartotojas pagal auginamą augalą pasirenka ligos modelį.
Aktorius	Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas pagal augalą pasirenka ligos modelį(savo arba sistemoje egzistuojantį) 2. Pasirenka perspėjimo būdą(SMS, el. paštu, mobilioji aplikacija) 3. Išsaugo pakeitimus. 4. Baigiamas PA. 	
Po sąlyga	Vartotojas pasirenka ligų modelius.

3.2.14 lentelė PA 12 Trinti stotelė iš sistemos

Panaudos atvejis „PA 12 Trinti stotelė iš sistemos”	
Aprašymas	Stotelės trynimasis iš sistemos.
Prieš sąlyga	Vartotojas nori panaikinti stotelę sistemoje.
Aktorius	Stotelės savininkas, Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Vartotojas atidaro stotelės profilio redagavimo langą. 2. Spaudžia mygtuką “Naikinti”. 3. Patvirtina savo veiksmą. 4. Baigiamas PA. 	
Po sąlyga	Stotelė ištrinta iš sistemos.

3.2.15 lentelė PA 13 Nustatyti individualius perspėjimus

Panaudos atvejis „PA 13 Nustatyti individualius perspėjimus”	
Aprašymas	Vartotojo nustatymai perspėjimams apie įvykius.
Prieš sąlyga	Vartotojas nori nustatyti perspėjimus.
Aktorius	Stotelės savininkas
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Atidaro stotelės profilio redagavimo langą. 2. Pasirenka perspėjimo sąlygas(ligų aptikimas, stotelės GPRS ryšio praradimas, senkanti stotelės baterija) 3. Išsaugo pakeitimus. 4. Baigiamas PA. 	
Po sąlyga	Vartotojas nustato perspėjimo sąlygas.

3.2.16 lentelė PA 14 Redaguoti visų stotelių profilius

Panaudos atvejis „PA 14 Redaguoti visų stotelių profilius”	
Aprašymas	Stotelių profilių redagavimas.
Prieš sąlyga	Administratorius pastebėjęs neatitikimus redaguoja sistemos stotelių profilius.
Aktorius	Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Atidaro visų stotelių sąrašą. 2. Pasirenką redaguojamą stotelę. 3. Atidaro jos profilio formą. 4. Redaguoja. 5. Išsaugo pakeitimus. 6. Baigiamas PA. 	
Po sąlyga	Administratorius poredagavo stotelės(-ių) profilį(-ius).

3.2.17 lentelė PA 15 Trinti stoteles iš sistemos

Panaudos atvejis „PA 15 Trinti stoteles iš sistemos”	
Aprašymas	Stotelių trynimasis iš sistemos.
Prieš sąlyga	Administratorius dėl tam tikrų aplinkybių nori ištrinti stotelę.
Aktorius	Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Atidaro visų stotelių sąrašą. 2. Pasirenką redaguojamą stotelę. 3. Atidaro jos profilio formą. 4. Spaudžia mygtuką “Naikinti”. 5. Patvirtina savo veiksmą. 6. Baigiamas PA. 	
Po sąlyga	Administratorius ištrynė stotelę iš sistemos.

3.2.18 lentelė PA 16 Peržiūrėti prisijungimų prie sistemos statistiką

Panaudos atvejis „PA 16 Peržiūrėti prisijungimų prie sistemos statistiką”	
Aprašymas	Prisijungimų ir kreipimusi į serverį statistikos peržiūra.
Prieš sąlyga	Administratorius nori peržiūrėti prisijungimų istoriją ir kreipimus į serverį.
Aktorius	Administratorius
Pagrindinių įvykių srautas	
<ol style="list-style-type: none"> 1. Atidaro statistikos langą. 2. Pasirenka vartotoją arba prisijungimą pagal IP adresą. 3. Peržiūri vartotojo arba pagal IP adresą veiksmus. 4. Baigiamas PA. 	
Po sąlyga	Administratorius peržiūrėjo sistemos prisijungimų ir kreipimusi statistiką.

3.3. Funkciniai reikalavimai

3.3.1 lentelė Laikotarpio pasirinkimas grafikų generavime

Reikalavimo numeris	1
Aprašymas	Laikotarpio pasirinkimas grafikų generavime
Pagrindimas	Galima matyti duomenis grafiko pagalba, tik to laikotarpio, kuris labiausiai domina vartotoją
Atlikimo kriterijus	Vartotojas pasirenka laikotarpį, kurio duomenis nori matyti grafike.
Vartotojo patenkinimas: 3	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Žiūrėti informaciją grafikų pagalba

3.3.2 lentelė Filtravimo ir rikiavimo lentelėse pasirinkimas

Reikalavimo numeris	2
Aprašymas	Filtravimo ir rikiavimo lentelėse pasirinkimas
Pagrindimas	Galima matyti duomenis lentelės pagalba, tik tuos kuriuos išfiltravo arba išrikiavo
Atlikimo kriterijus	Vartotojas pasirenka filtrus ir rikiavimą.
Vartotojo patenkinimas: 3	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Žiūrėti informaciją lentelių pagalba

3.3.3 lentelė Sistemos sąsaja

Reikalavimo numeris	3
Aprašymas	Sistemos sąsaja
Pagrindimas	Kad sistemos turimus duomenis ir funkcionalumu galėtų naudotis kitos sistemos, bei įrenginiai.
Atlikimo kriterijus	Kitos sistemos prisijungia prie kuriamos sistemos per API.
Vartotojo patenkinimas: 3	Vartotojo nepatenkinimas: 1
Panaudos atvejai	API(JSON)

3.3.4 lentelė Orų prognozių peržiūra

Reikalavimo numeris	4
Aprašymas	Orų prognozių peržiūra
Pagrindimas	Galima stebėti patikslintas orų prognozes stotelės būvimo vietoje.
Atlikimo kriterijus	Vartotojas atidaro prognozių langą ir pasirenka datą.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 4
Panaudos atvejai	Orų prognozių peržiūra

3.3.5 lentelė Stebėti duomenis grafiko pagalba

Reikalavimo numeris	5
Aprašymas	Stebėti duomenis grafiko pagalba
Pagrindimas	Galima peržiūrėti duomenis grafiko pagalba, taip yra patogiau analizuoti orų permainas.
Atlikimo kriterijus	Vartotojas atidaro grafiką ir pasirenka datą. Įjungia arba išjungia grafikus, kuriuos nori/nenori matyti.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 3
Panaudos atvejai	Peržiūrėti stotelės informaciją grafikų pagalba.

3.3.6 lentelė Stebėti duomenis lentelėje

Reikalavimo numeris	6
Aprašymas	Stebėti duomenis lentelėje
Pagrindimas	Galima peržiūrėti duomenis lentelėje.
Atlikimo kriterijus	Vartotojas atidaro lentelę ir pasirenka jam reikalingą rykiavimą.
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 2
Panaudos atvejai	Žiūrėti stotelės informaciją lentelės pagalba.

3.3.7 lentelė Žiūrėti stotelės būvimo vietą

Reikalavimo numeris	7
Aprašymas	Žiūrėti stotelės būvimo vietą
Pagrindimas	Galima peržiūrėti kurioje vietoje yra stotelė.
Atlikimo kriterijus	Vartotojas atsidaro stotelę ir peržiūri kurioje vietoje ji yra.
Vartotojo patenkinimas: 3	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Peržiūrėti stotelės būvimo vietą žemėlapyje

3.3.8 lentelė Užregistruoti stotelę

Reikalavimo numeris	8
Aprašymas	Užregistruoti stotelę
Pagrindimas	Vartotojas turi turėti teisę užregistruoti stotelę ir naudotis sistema.
Atlikimo kriterijus	Vartotojas suveda visus reikiamus laukus stotelės užregistravimui.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 4
Panaudos atvejai	Užregistruoti nuosavą stotelę.

3.3.9 lentelė Žiūrėti visų stotelių duomenis

Reikalavimo numeris	9
Aprašymas	Žiūrėti visų stotelių duomenis
Pagrindimas	Visi vartotojai gali stebėti sistemoje užregistruotų stotelių duomenis
Atlikimo kriterijus	Vartotojas peržiūri visas stoteles ir jų duomenis.
Vartotojo patenkinimas: 3	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Peržiūrėti visas sistemoje užregistruotas stoteles

3.3.10 lentelė Stotelės informacijos redagavimas

Reikalavimo numeris	10
Aprašymas	Stotelės informacijos redagavimas
Pagrindimas	Vartotojas turi galėti pakeisti stotelės informaciją.
Atlikimo kriterijus	Vartotojas atsidaręs stotelės profilį, pakeičia informaciją.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 3
Panaudos atvejai	Redaguoti stotelės profilį

3.3.11 lentelė Ligos modelio sudarymas

Reikalavimo numeris	11
Aprašymas	Ligos modelio sudarymas
Pagrindimas	Ligos modelio sudarymas pagal asmeninius augalus.
Atlikimo kriterijus	Vartotojas sukuria ligos modelį, bei visus jo kriterijus.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 4
Panaudos atvejai	Sudaryti augalų ligų modelius

3.3.12 lentelė Sekti ligos modelį

Reikalavimo numeris	12
Aprašymas	Sekti ligos modelį.
Pagrindimas	Vartotojas turi galėti stotelei priskirti, kurią ligą/ligas sekti.
Atlikimo kriterijus	Vartotojas atsidaręs ligos modelį, priskiria jam stoteles.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 4
Panaudos atvejai	Sudaryti augalų ligų modelius

3.3.13 lentelė Stotelės trynimas

Reikalavimo numeris	13
Aprašymas	Stotelės trynimas
Pagrindimas	Vartotojas turi galėti panaikinti stotelę iš sistemos.
Atlikimo kriterijus	Vartotojas atsidaręs stotelės redagavimo langą, ištrina stotelę.
Vartotojo patenkinimas: 2	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Trinti stotelę iš sistemos

3.3.14 lentelė Nustatyti perspėjimus

Reikalavimo numeris	14
Aprašymas	Nustatyti perspėjimus
Pagrindimas	Vartotojas turi galėti nustatyti, kokiomis oro sąlygomis jis turi būti perspėtas.
Atlikimo kriterijus	Vartotojas atsidaręs stotelės profilį, nustato kokiomis aplinkybėmis jį reikia perspėti žinute.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 3
Panaudos atvejai	Nustatyti individualius perspėjimus.

3.3.15 lentelė Administravimas

Reikalavimo numeris	15
Aprašymas	Administravimas
Pagrindimas	Turi būti administravimo profilis, kur administratorius galėtų valdyti sistemą.
Atlikimo kriterijus	Vartotojas su administratoriaus teisėmis, turi teise redaguoti informaciją sistemoje, bei peržiūrėti prisijungimus ir veiksmus sistemoje.
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Redaguoti stotelių profilius, Trinti stoteles iš sistemos, Peržiūrėti prisijungimų prie sistemos statistiką.

3.4. Nefunkciniai reikalavimai

3.4.1. Reikalavimai sistemos išvaizdai

3.4.1 lentelė Išvaizda

Reikalavimo numeris	17
Aprašymas	Išvaizda turi būti minimalistinė. Pagrindinis sistemos fonas turi būti baltos spalvos, mygtukai aiškūs ir su paveikslukais.
Pagrindimas	Sistema turi būti lengva naudotis. Visi sistemos mygtukai neturi kelti dviprasmybių.
Atlikimo kriterijus	Vartotojams patogiu, kai sistema yra aiški.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.2 lentelė Stilius

Reikalavimo numeris	18
Aprašymas	Sistemoje neturi būti naudojamos pereinamos ir neryškios spalvos. 12 šrifto dydis, Calibri stiliumi.
Pagrindimas	Spalvos ir šriftas neturi varginti vartotojo akių ir leisti naudotis sistema ilgą laiką.
Atlikimo kriterijus	Vartotojus sistema neturi veikti psichologiškai. Vartotojas turi galėti ilgą laiką naudotis sistema.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.2. Reikalavimai panaudojamumui

3.4.3 lentelė Naudojimosi paprastumas

Reikalavimo numeris	19
Aprašymas	Produktu turi galėti naudotis bet kuris interneto vartotojas. Vartotojo sąsaja turi būti neperkrauta ir lengvai suprantama. Neturiu reikėti papildomų mokymų, kad naudotis šia sistema. Sistema turi būti panaši į kitas internetines svetaines, kad vartotojams apsilankius iš karto būtų viskas aišku.
Pagrindimas	Vartotojui nereikės papildomų mokymųsi darbui su sistema, kadangi aplinka jau jam bus pažįstama.
Atlikimo kriterijus	Prisijungus prie sistemos turi būti viskas aišku.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.4 lentelė Naudojimosi paprastumas

Reikalavimo numeris	20
Aprašymas	Visi mygtukų paveikslukai/užrašai turi būti aiškūs ir trumpi. Matyti kiekvieno puslapio antraštę, užvedus pelę ant svarbaus lango elemento matyti papildomą paaiškinimą.
Pagrindimas	Sistema tampa aiškesnė ir lengvesnė naudoti, suteikia daugiau pagalbos vartotojui. Nereikia papildomai skaityti dokumentacijos, kad naudotis sistema.
Atlikimo kriterijus	Naudotis sistema turi būti lengva ir aišku.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.5 lentelė Vartotojui skirtos konfigūravimo priemonės

Reikalavimo numeris	21
Aprašymas	Vartotojas turi galėti nusistatyti paskyros nustatymus: perspėjimų nustatymai, profilio redagavimas, stotelės informacijos redagavimas.
Pagrindimas	Sistema lankstesnė, prisitaiko prie vartotojo poreikių.
Atlikimo kriterijus	Vartotojas nusistato paskyros nustatymus pagal savo poreikius.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	Redaguoti stotelės profilį; Pasirinkti stotelės sekamų augalų ligų modelius; Mobilioje aplikacijoje nustatyti perspėjimus

3.4.6 lentelė Mokymosi reikalavimai

Reikalavimo numeris	22
Aprašymas	Neturi reikėti jokių mokymų, kad naudotis šia sistema.
Pagrindimas	Sutaupo vartotojui laiko ir pastangų.
Atlikimo kriterijus	Vartotojui pradėjus naudotis sistema nereikia jokių mokymų.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.7 lentelė Supratingumas ir mandagumas

Reikalavimo numeris	23
Aprašymas	Sistema turi būti lengvai suprantama. Klaidų ir kitų sistemos sutrikimų pranešimai turi būti aiškūs ir su paaiškinimu kaip vartotojas gali klaidą išspręsti.
Pagrindimas	Lengvesnis sistemos naudojimas, kuomet klaidos aiškios ir su pagalba, kaip tas klaidas išspręsti.
Atlikimo kriterijus	Vartotojas gavęs klaidos pranešimą aiškiai suvokia, kas atsitiko.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.8 lentelė Prieinamumas neįgaliesiems

Reikalavimo numeris	24
Aprašymas	PĮ gali būti eksploatuojama arba konfigūruojama daltonizmu sergančių žmonių.
Pagrindimas	Vartotojai su negalia turi galimybe naudotis sistema.
Atlikimo kriterijus	Neįgalus vartotojas gali naudotis sistema be jokių trikdžių.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.3. Reikalavimai vykdymo savybėms

3.4.9 lentelė Užduočių vykdymo greitis

Reikalavimo numeris	25
Aprašymas	Sistemoje visi veiksmai turi vykti vartotojui priimtiniu greičiu(1-2 sekundės). Apie ilgesnį veiksmą(grafikų generavimas), vartotojai turi būti informuojami, jog sistema dirba.
Pagrindimas	Vartotojas gauna atsakymus iš sistemos jam priimtiniu greičiu ir yra informuojamas apie ilgesnius veiksmus. Taip vartotojas žinos, jog sistema veikia, o ne „užlūžo“.
Atlikimo kriterijus	Atlikdamas veiksmą vartotojas mato, ką sistema veikia tuo metu.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.10 lentelė Reikalavimai tikslumui

Reikalavimo numeris	26
Aprašymas	Duomenis, kuriuos suveda vartotojas turi būti išsaugoti lygiai tokie pat. Stotelės teikiami duomenų tikslumas priklausys nuo jutiklių paklaidos.
Pagrindimas	Stotelės duomenų tikslumas turi būti neiškraipytas. Paklaida priklausys tik nuo jutiklių.
Atlikimo kriterijus	Vartotojui yra labai svarbus duomenų tikslumas. Vartotojas turi būti informuotas, kur ir kokios paklaidos sistemoje yra.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.11 lentelė Patikimumas

Reikalavimo numeris	27
Aprašymas	Prisijungimas prie paskyros turi būti apsaugotas. Kiekviena stotelė turi turėti savo raktą, kurį siunčia kartu su užklausa į serverį.
Pagrindimas	Sistema turi būti apsaugota nuo neteisėto įsilaužimo ir duomenų iškraipymo.
Atlikimo kriterijus	Vartotojui yra svarbu turėti neiškraipytus ir teisingus duomenis.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.12 lentelė Pasiekiamumas

Reikalavimo numeris	28
Aprašymas	Sistema turi būti pasiekama bet kuriuo metu. Apie sistemos trikdžius turi būti pranešama iš anksto.
Pagrindimas	Sistema turi būti pasiekama bet kuriuo metu, kas leis vartotojui sistema naudotis be trikdžių.
Atlikimo kriterijus	Jei vartotojas bando prisijungti prie sistemos, o ji nepasiekama, jis turi būti informuotas, dėl kokios priežasties sistema nėra pasiekama.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 3
Panaudos atvejai	-

3.4.13 lentelė Reikalavimai apdorojamų duomenų apimtims

Reikalavimo numeris	29
Aprašymas	Sistema turi apdoroti visų stotelių duomenis.
Pagrindimas	Visų stotelių duomenys turi būti saugojami serveryje. Serveris turi sugebėti apdoroti visų užregistruotų stotelių duomenis.
Atlikimo kriterijus	Vartotojas turi matyti visų stotelių duomenis.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 3
Panaudos atvejai	-

3.4.14 lentelė Reikalavimai išplečiamumui

Reikalavimo numeris	30
Aprašymas	PĮ turi būti suprojektuota taip, jog būtų galimybė pridėti neribotą stotelių kiekį. Taip sistema turės daug duomenų, kurie padės prognozuoti orus. Sistemoje taip pat turi būti galimybė keisti ir pridėti pagrindinius algoritmus(oru, bei augalų ligų prognozavimui).
Pagrindimas	Sistema turi turėti galimybę būti plečiamai.
Atlikimo kriterijus	Vartotojui naudinga matyti kuo daugiau kitų stotelių duomenų, bei gauti tikslias prognozes.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 2
Panaudos atvejai	-

3.4.15 lentelė Reikalavimai produkto ilgaamžiškumui

Reikalavimo numeris	31
Aprašymas	PĮ turi būti suprojektuota taip, kad galima būtų laisvai pridėti naują funkcionalumą, bei algoritmus, nekeičiant sistemos nuo pagrindų. Taip bus užtikrintas sistemos palaikomumas ir ilgaamžiškumas.
Pagrindimas	Sistema turi būti suprojektuota į ateitį.
Atlikimo kriterijus	Vartotojai rinkdamiesi produktą atkreipia dėmesį į produkto ilgaamžiškumą.
Vartotojo patenkinimas: 5	Vartotojo nepatenkinimas: 2
Panaudos atvejai	-

3.4.4. Reikalavimai saugumui

3.4.16 lentelė Prieigos reikalavimai

Reikalavimo numeris	39
Aprašymas	Visi interneto vartotojai turės galimybę žiūrėti stotelių teikiamus duomenis. Tik stotelės savininkas galės kurti ligos modelius ir nusistatyti perspėjimus iš savo stotelės. Administratorius galės matyti visų stotelių informacija ir ją redaguoti.
Pagrindimas	Didesnio entuziastų rato pritraukimas, leidžiant jiems stebėti stotelių duomenis.
Atlikimo kriterijus	Visi interneto vartotojai galės naudotis sistema. Tik jie turės skirtingas teises.
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.17 lentelė Vientisumo (integralumo) reikalavimai

Reikalavimo numeris	40
Aprašymas	Visi į sistemą įvedami duomenys ir informacija prieš tai turi būti validuojama.
Pagrindimas	Kad sistema veiktų be trikdžių, būtina validuoti įvedamus duomenis.
Atlikimo kriterijus	Vartotojui įvedus informaciją į sistema, jie turi būti validuojami.
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.18 lentelė Reikalavimai privatumui

Reikalavimo numeris	41
Aprašymas	Prisijungimo ir vartotojų asmeniniai duomenys turi būti privatus
Pagrindimas	Vartotojo duomenys turi būti privatus. Informaciją perimantys tretieji asmenys gali ją panaudoti kenkejiškiems tikslams
Atlikimo kriterijus	Vartotojo suvesta informacija yra privati(el.paštas, prisijungimo vardai, Facebook ir Google paskyros).
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.19 lentelė Audito reikalavimai

Reikalavimo numeris	42
Aprašymas	Visi sistemos įvykiai turi būti saugomi.
Pagrindimas	Lengva atsekti, kas padarė pakeitimą.
Atlikimo kriterijus	Vartotojas pakeitęs (pvz: augalo ligos modelį) informaciją, ją iš karto matys kiti vartotojai(galės pažiūrėti kas padarė pakeitimą).
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.4.20 lentelė Reikalavimai savisaugai nuo išorinių grėsmių

Reikalavimo numeris	43
Aprašymas	Produktas turi užkirsti kelią neautorizuotiems prisijungimams. Turi galėti prisijungti tik autorizuoti asmenys ir sistemos.
Pagrindimas	Tik autorizuoti prisijungimai prie sistemos leis apsaugoti sistemos duomenis.
Atlikimo kriterijus	Stotelės per API kreipsis tik su raktais. Vartotojai save autentifikuos kitų sistemų paskiromis.
Vartotojo patenkinimas: 4	Vartotojo nepatenkinimas: 1
Panaudos atvejai	-

3.5. Naudotos technologijos ir metodologijos

Kuriant programinę įrangą buvo skiriamas didelis dėmesys lengvam sistemos plečiamumui ir palaikomumui gerinti. Tam tikslui sistemos kūrimo buvo naudojamos įvairios programinės įrangos kūrimo metodologijos.

Serverio pusė buvo programuojama naudojant „Laravel 5.2“ „Php“ programavimo kalbos karkasą. Šis karkasas įgyvendina MVC (Model-View-Controller) kūrimo ideologiją. Vien šio karkaso panaudojimas leido sistemą projektuoti labiau plečiamą. Serverio pusėje buvo rašomi testai su „PHPUnit“ įrankio pagalba, kurie leidžia atlikti pakeitimus sistemoje nebijant, kad bus paveiktos kitos sistemos dalys ir tai nebus pastebėta.

Visi serverio programinės įrangos sluoksniai bus atskirti, tai leis bet kuriuo metu esant poreikiui pakeisti vieną dalį, nekludant kitos.

Projektuojant sistemą buvo atsižvelgiama į tai, jog sistema naudosis įvairūs įrenginiai (personalinis kompiuteris, stotelės bei mobilūs įrenginiai). Todėl buvo nuspręsta realizuoti API, per kurį visi įrenginiai bus susieti tarpusavyje. Projekte naudojama MySQL duomenų bazė. Komunikavimui tarp duomenų bazės ir sistemos naudojome ORM ideologiją.

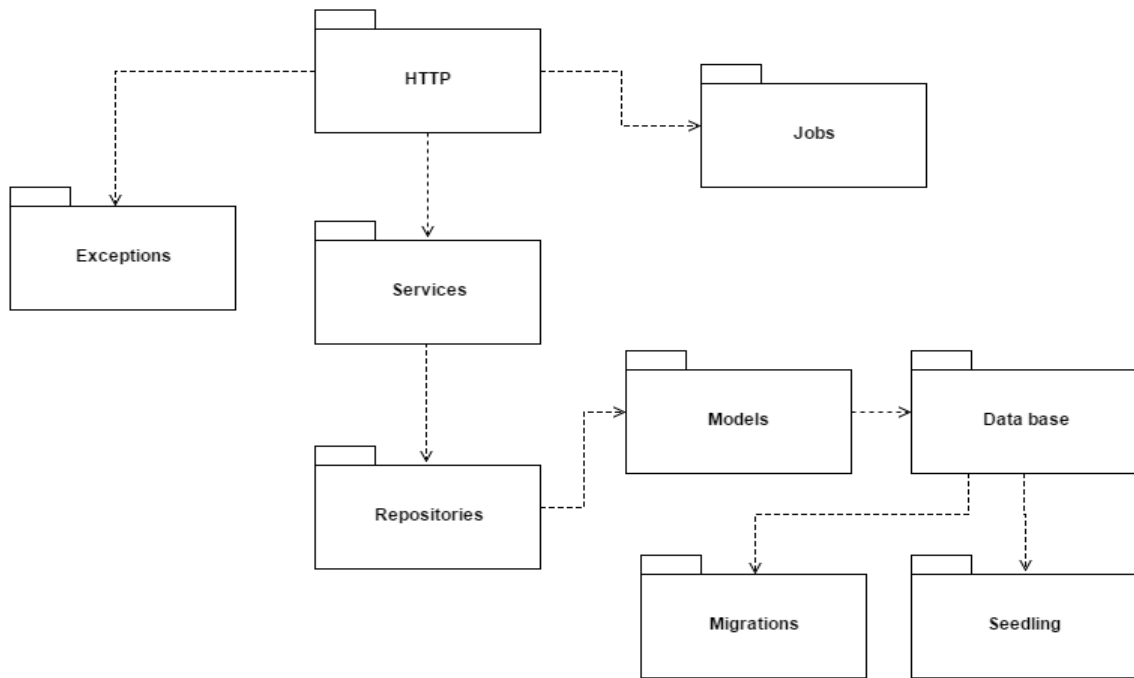
Sistemos atvaizdavimui buvo naudojamas „Angular Js“ JavaScript kalbos karkasas. Tai leido visiškai atskirti kliento ir serverio puses, bei sukurti interaktyvią vartotojo sąsają. Pasinaudojus „Ionic“ karkaso pagalba sukūrėme mobilią aplikaciją.

Stotelė buvo programuojama C++ ir Python kalbomis. Stotelę suprojektavome taip, kad ilgainiui bus galima pridėti papildomų jutiklių, bei išplėsti stotelės funkcionalumą. Prisijungimui prie serverio naudojome USB modemą su SIM kortele. Duomenų bazė – MySQL.

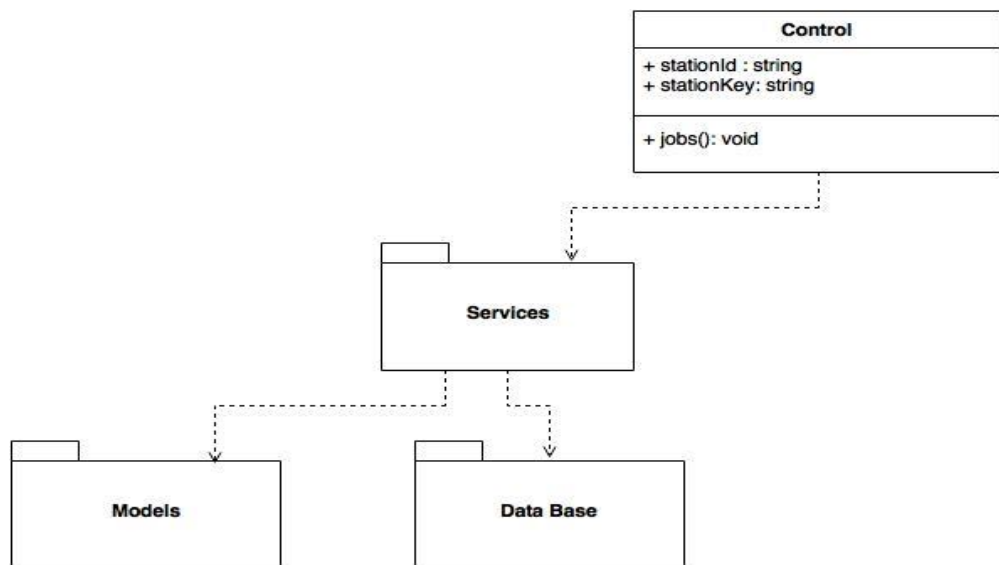
3.6. Sistemos architektūros modelis

3.6.1. Skirstymas į paketus

3.6.1.1. Sistemos skirstymas į paketus



3.6.1 pav. Serverio dalies paketų diagrama

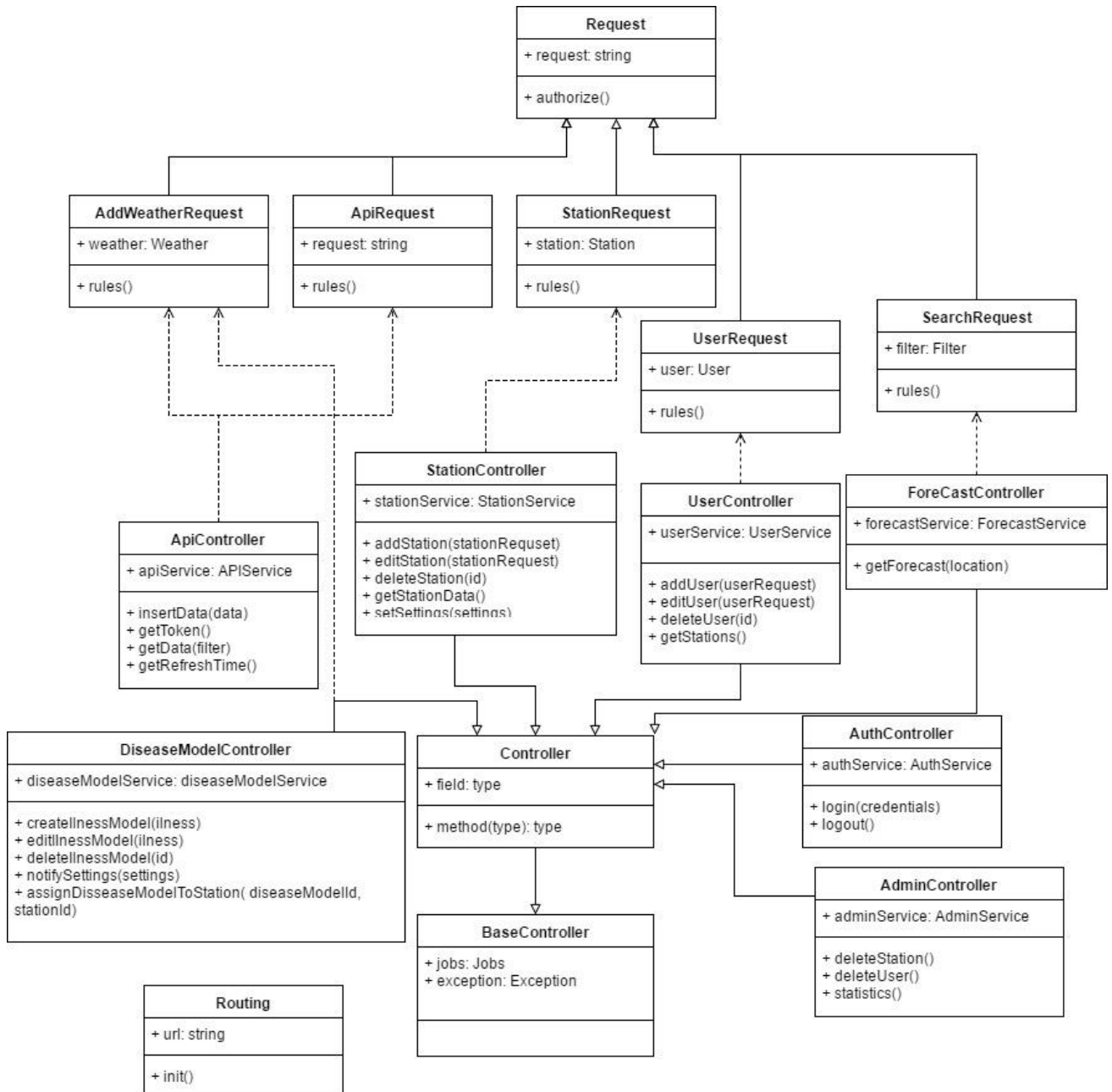


3.6.2 pav. Meteorologinės stotelės paketų diagrama

Control – tai klasė, kuri inicijuojama „Raspberry Pi“ paleidimo metu. Ši klasė yra kaip valdiklis visai stotelei. Joje inicijuojami visi procesai, kurie gyvuos kol stotelė bus įjungta.

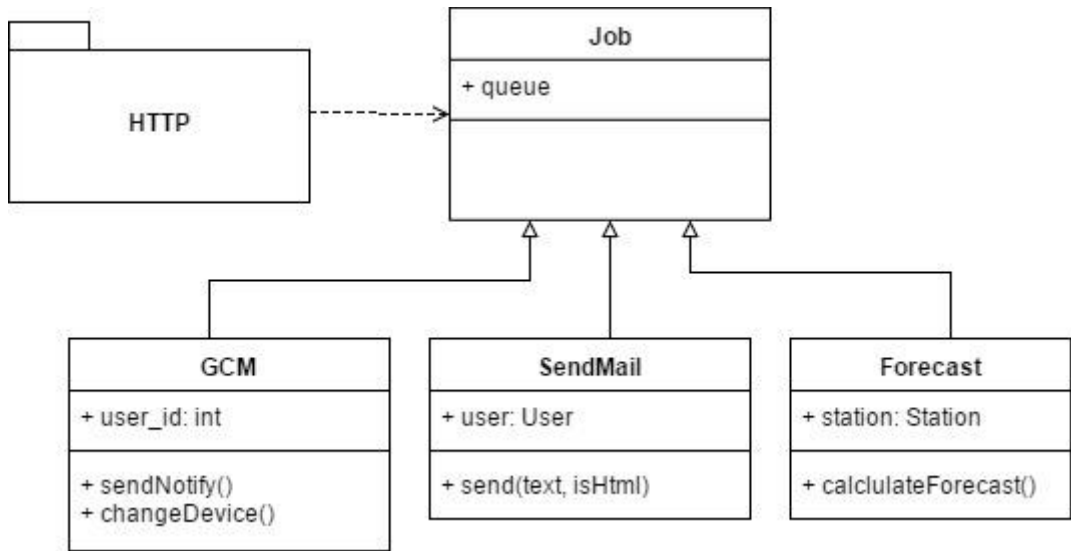
3.6.1.2. Sistemos paketų detalizavimas

Serverio dalies paketai:



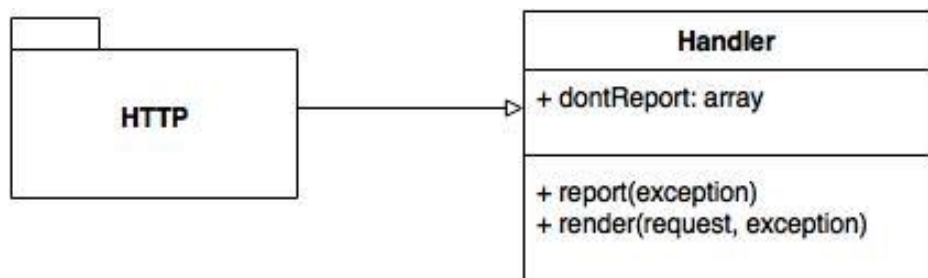
3.6.3 pav. „HTTP“ paketas

Siunčiant užklausas į serverį (iš stotelės ar kitų įrenginių), už jų apdorojimą atsakingas šis paketas. Jis apdoroja užklausas, bei gražina rezultatus JSON formatu. Šis paketas jokios logikos neturi, jis tik gauna duomenis iš užklausų, juos validuoja ir atiduoda servisams tolesniam jų apdorojimui.



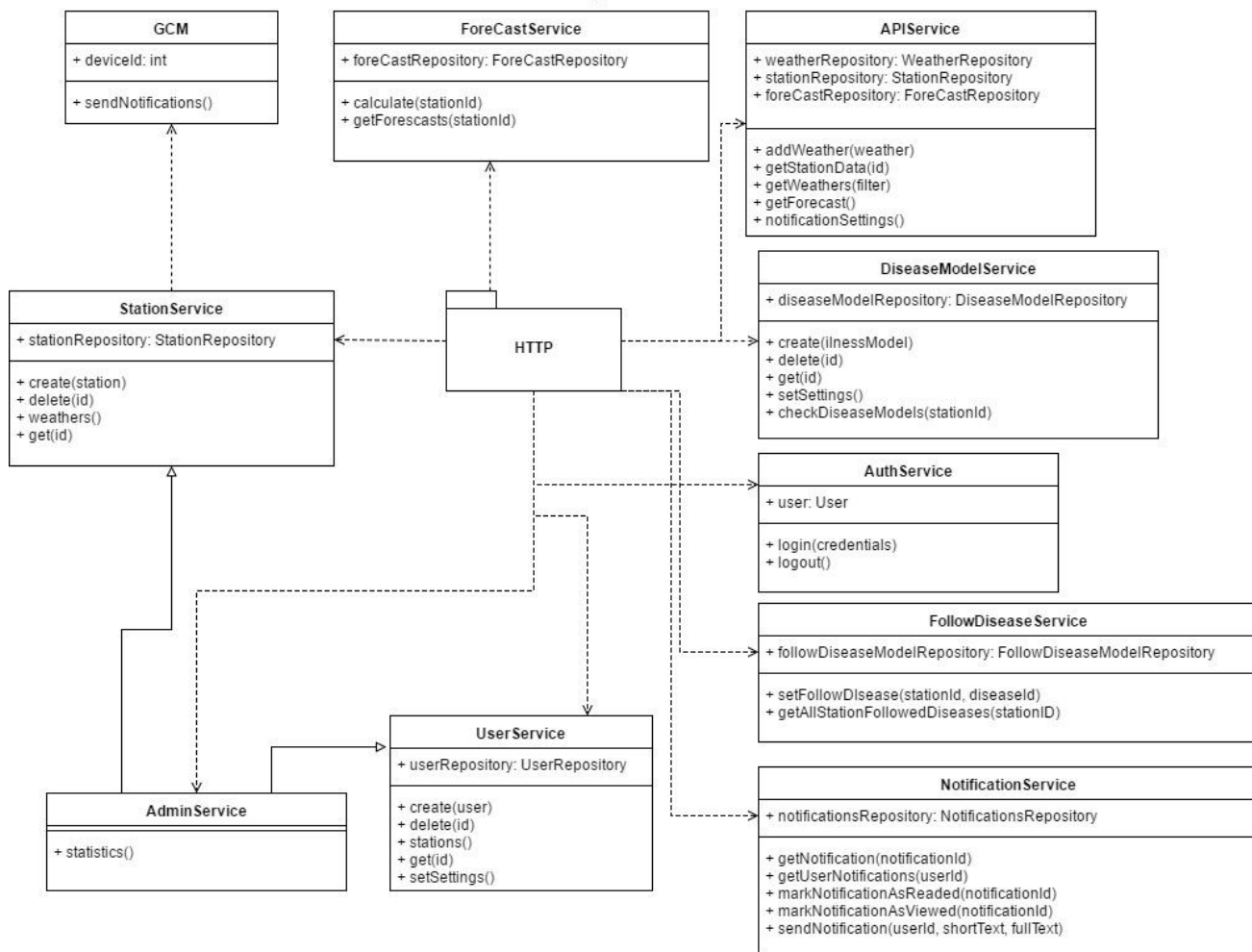
3.6.4 pav. „Jobs“ paketas

Šis paketas yra atsakingas už darbus. Šiuo atveju bus sukuriama eilė, iš elektorinių laiškų siuntimo, orų prognozavimo bei perspėjimų išsiuntimo mobiliems įrenginiams. Atsilaisvinęs serveris iš eilės vykdys visus šiuos darbus. Orų prognozavimas inicijuojamas kiekvienos dienos vidurnaktį.



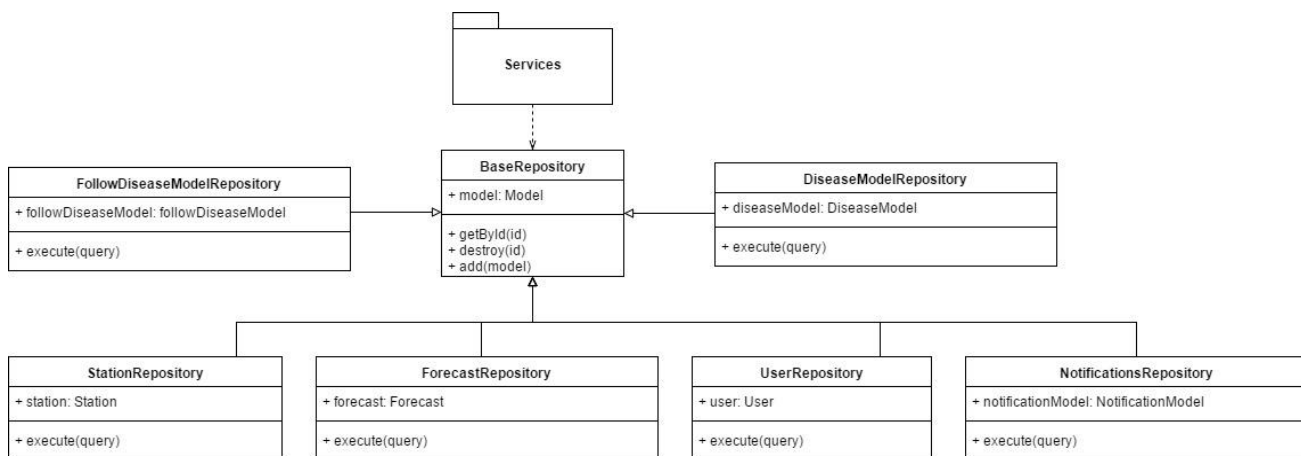
3.6.5 pav. „Exceptions“ paketas

3.5.5 paveiksle paketas kuris atsakingas už išimčių apdorojimą.



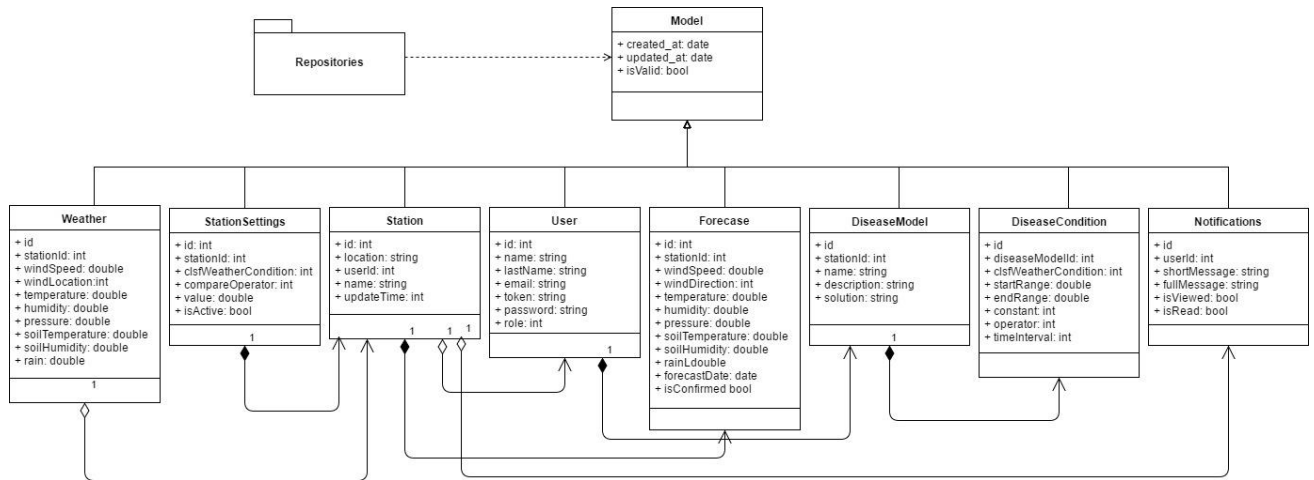
3.6.6 pav. „Services“ paketas

Aukščiau pateiktame „Services“ pakete yra įgyvendinta visa sistemos logika. Atskiri servais norėdami gauti duomenis iš duomenų bazės kreipiasi į saugyklas, kurios pateiktos žemiau esančiame paveiksle.



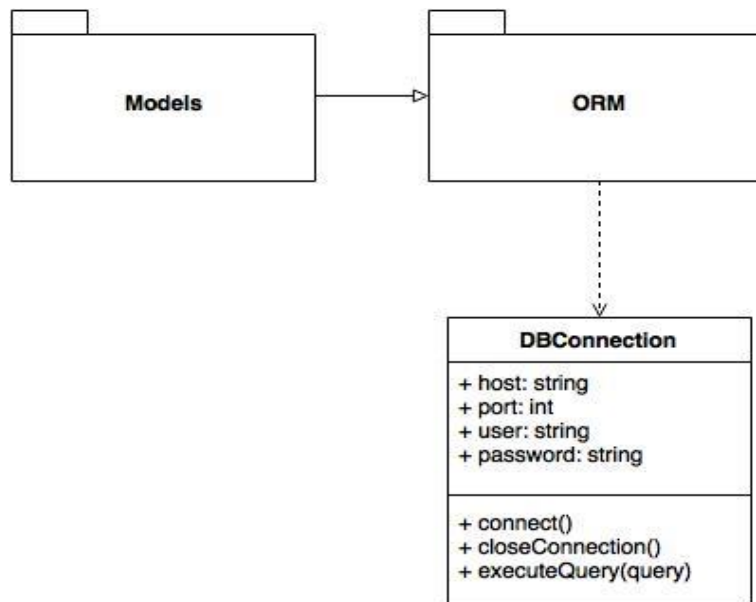
3.6.7 pav. „Repositories“ paketas

Šiame pakete realizuojamos visos užklausos į duomenų bazę, per esybes (modelius). Servisuose repositorijos inicijuojamos pasitelkiant DI (Dependency Injection).



3.6.8 pav. „Models“ paketas

Šį paketą naudoja repositorijos, kurios kuria užklausas. Modeliai turi atributus, tokius pat, kaip ir duomenų bazės lentelės. Tarp modelių yra panaudoti agregavimo ir kompozicijos ryšiai.



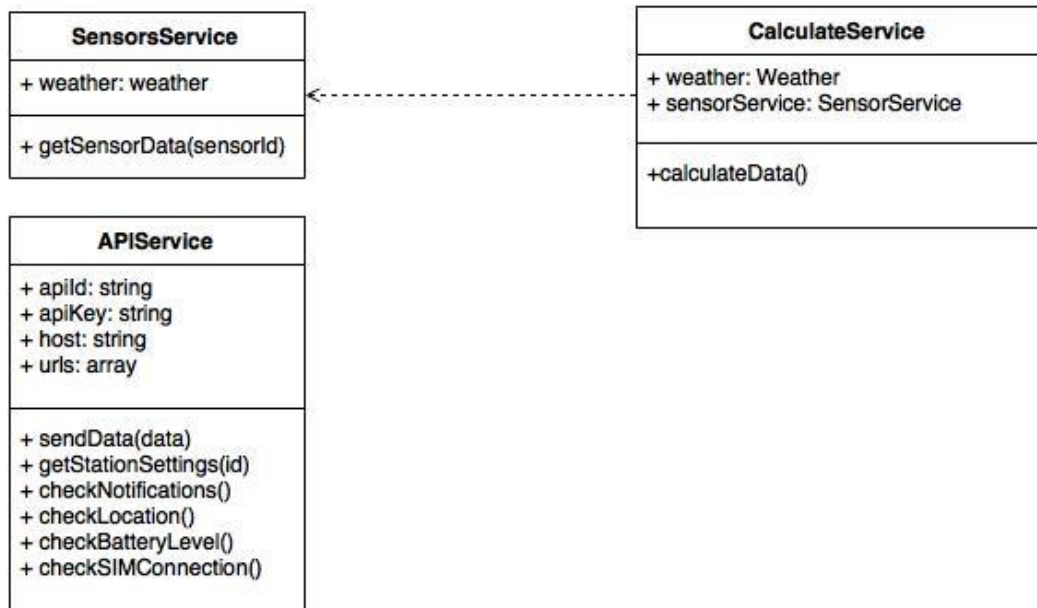
3.6.9 pav. „DataBase“ paketas

Modeliai analogiški duomenų bazės esybėms, todėl per ORM (Object-relational mapping) sluoksnį galima manipuluoti duomenų bazės įrašais. Tai labai patogu, kuomet visa aplikacija yra suprojektuota objektiškai.

Migrations – tai paketas kuriame yra klasės, kurios kuria duomenų bazės esybes. Buvo naudojama „code first“ ideologija. Todėl laukų pridėjimas ar jų redagavimas kas kartą pridėdavo po naują klasę. Naujas pakeitimas duomenų bazėje iššaukdavo naujos klasės kūrimą.

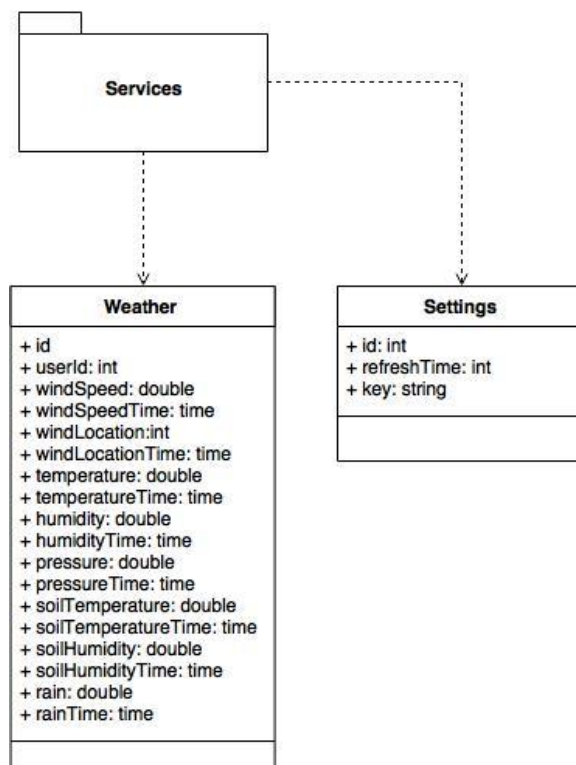
Seedling paketas saugo ir įkelia bazinius duomenis į duomenų bazę.

Stotelės paketai:



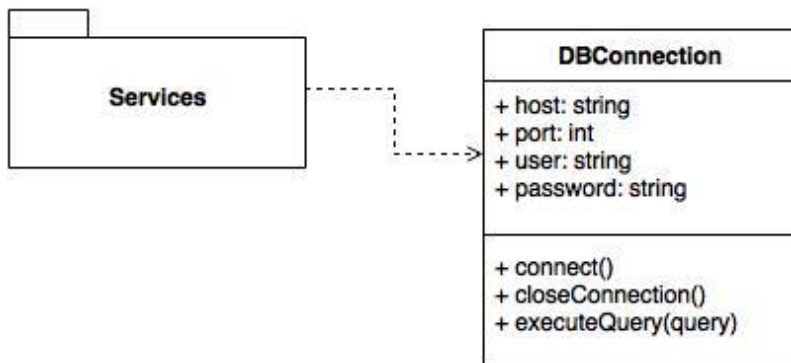
3.6.10 pav. „Services“ paketas

Šiame pakete realizuota visa stotelės logika. Duomenų paėmimui iš jutiklių, jų apdorojimas ir saugojimas lokaliaje duomenų bazėje. Užklausų suformavimas į serverį, bei serverio atsakų apdorojimas.



3.6.11 pav. „Models“ paketas

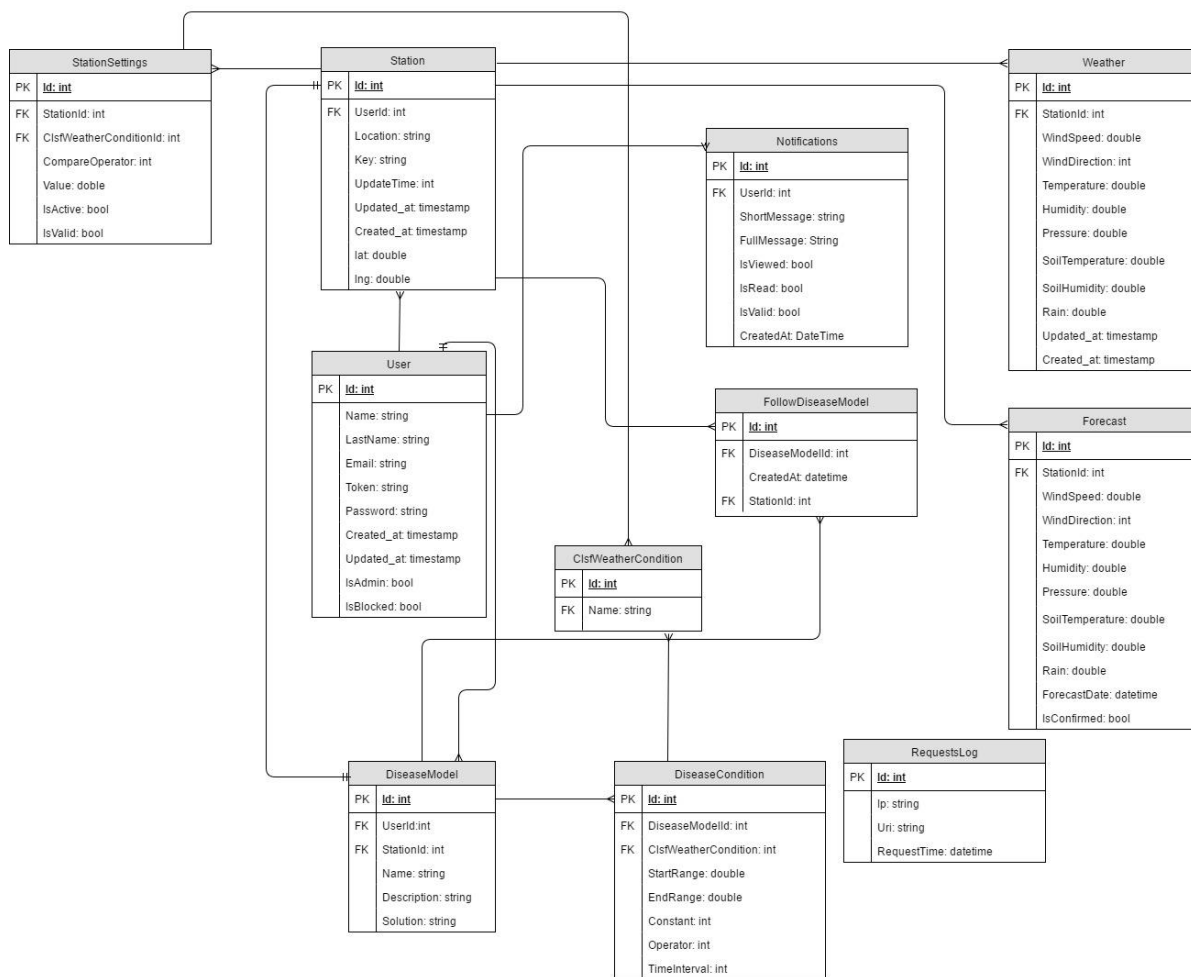
3.5.11 paveiksle pavaizduotos duomenų bazės esybės, kuriomis manipuluos servaisi.



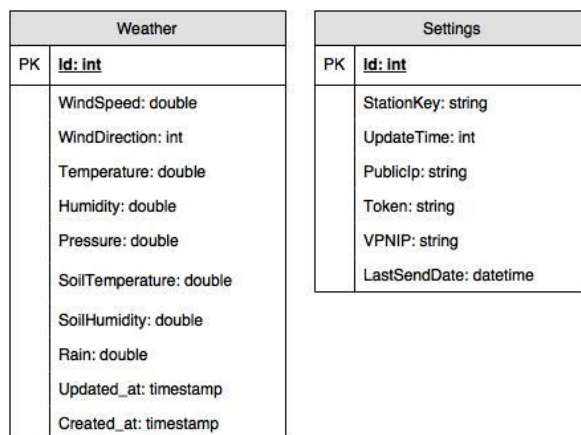
3.6.12 pav. „DataBase“ paketas

Šiame pakete yra prisijungimo prie duomenų bazės, bei užklausų vykdymo klasė.

3.7. Duomenų vaizdas



3.7.1 pav. Serverio dalies duomenų bazės vaizdas



3.7.2 pav. Autonominės meteorologinės stotelės duomenų bazės vaizdas

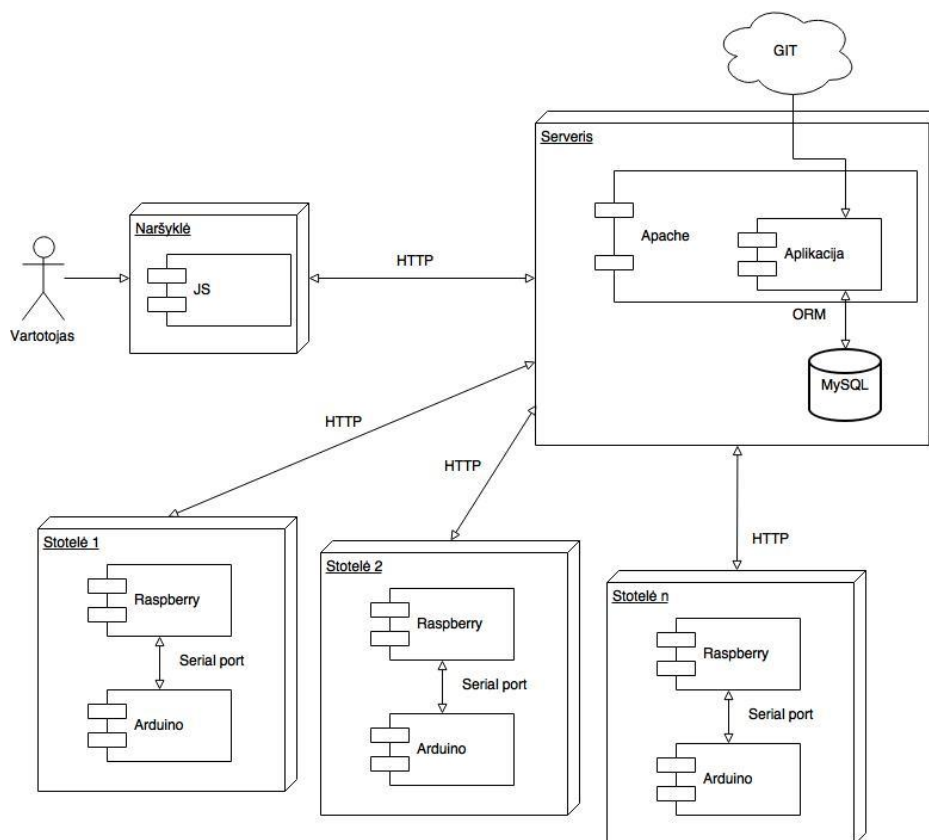
3.8. Išdėstymo vaizdas

Vartotojas turi turėti naršyklę su JavaScript kalbos palaikymu ir interneto prieiga.

Serveryje turi būti įdiegtas „Apache“ web serveris, bei MySQL duomenų bazė. Serveris turi būti dedikuotas ir jame įrašytas „OpenVPN“ servisas, bei „Capistrano“, kurio dėka programinės įrangos kodas bus atnaujinamas tiesiai iš versijavimo sistemos.

Stotelė turi turėti GSM modulį su interneto prieiga, jame turi būti įrašyta MySQL duomenų bazė.

Žemiau pateiktame 3.7.1 pav. galite matyti išdėstymo diagramą.



3.8.1 pav. Išdėstymo diagrama

3.9. Naudojami trečiųjų šalių komponentai

- Jwt – Raktu grįstas autentifikavimas sistemoje;
- Highcharts – biblioteka skirta grafikams braižyti;

3.10. Autonominės meteorologinės stotelės konstrukcija



3.10.1 pav. Sukurta autonominė meteorologinė stotelė

4. TYRIMO DALIS

4.1. Tyrimo tikslas

Šio tyrimo tikslas yra patikrinti koks sukurtos meteorologinės stotelės teikiamų duomenų, bei orų prognozavimo algoritmo tikslumas. Nustatyti meteorologinės stotelės ir sistemos problemines sritis. Atrasti aparatūrinės, bei programinės įrangos tobulinimo galimybes.

4.2. Tyrimo aprašymas

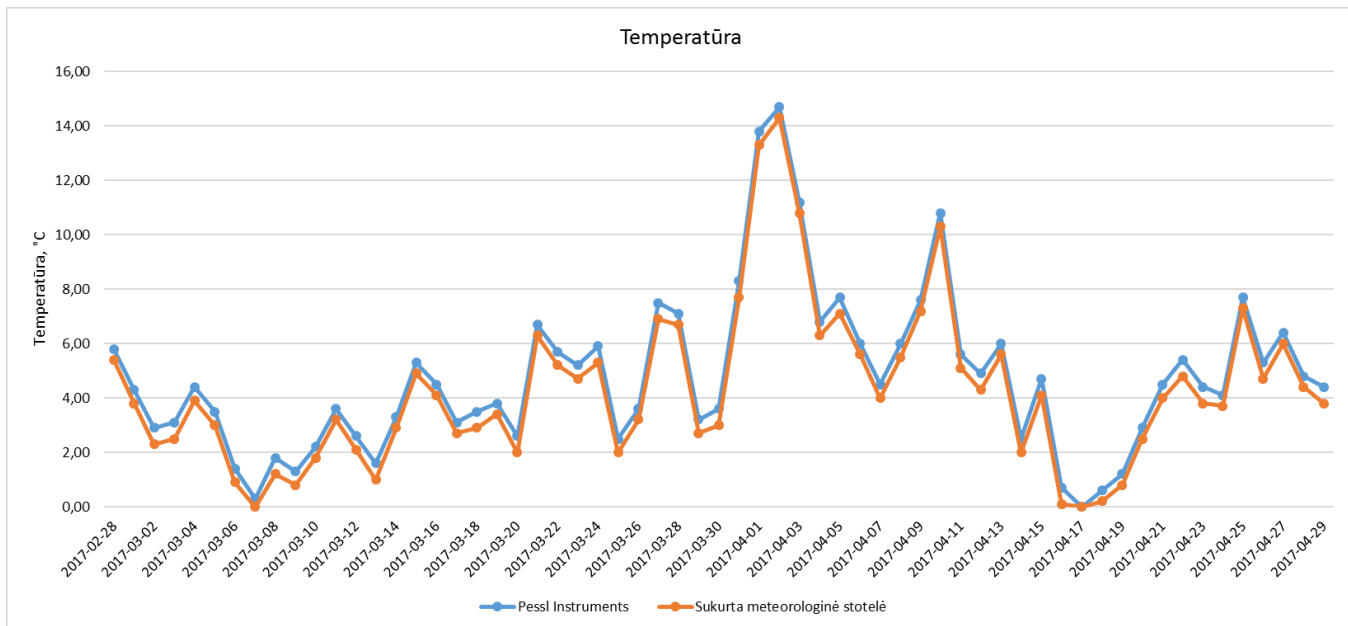
Norint atlikti šį tyrimą, reikia turėti sukurtos autonominės meteorologinės stotelės sukauptus duomenis, apskaičiuotas prognozes, bei faktinius tiriamos vietos orų parametrus. Jei gautus duomenis palyginę pastebėsime nesutapimų, reikės išsiaiškinti jų atsiradimo priežastį, bei rasti sprendimus jiems ištaisyti.

4.3. Tyrimo eiga

Pirma reikia įsitikinti ar stotelė veikia tinkamai. Ar ji siunčia duomenis nustatytu intervalu, ar yra pilnai autonominė bei patikrinti jos teikiamų duomenų teisingumą. Šiam tikslui, stotelė buvo pastatyta Biržų rajone dvejiems mėnesiams šalia profesionalios „Pessl Instruments“ meteorologinės stotelės. Stotelė kaupė duomenis nuo kovo 1d. iki balandžio 27d. Lyginsime šios atkarpos 5 rodiklių duomenis: oro temperatūrą, dirvožemio temperatūrą 30 cm gylyje, vėjo greitį, kritulius, bei dirvožemio drėgmę 30 cm gylyje.

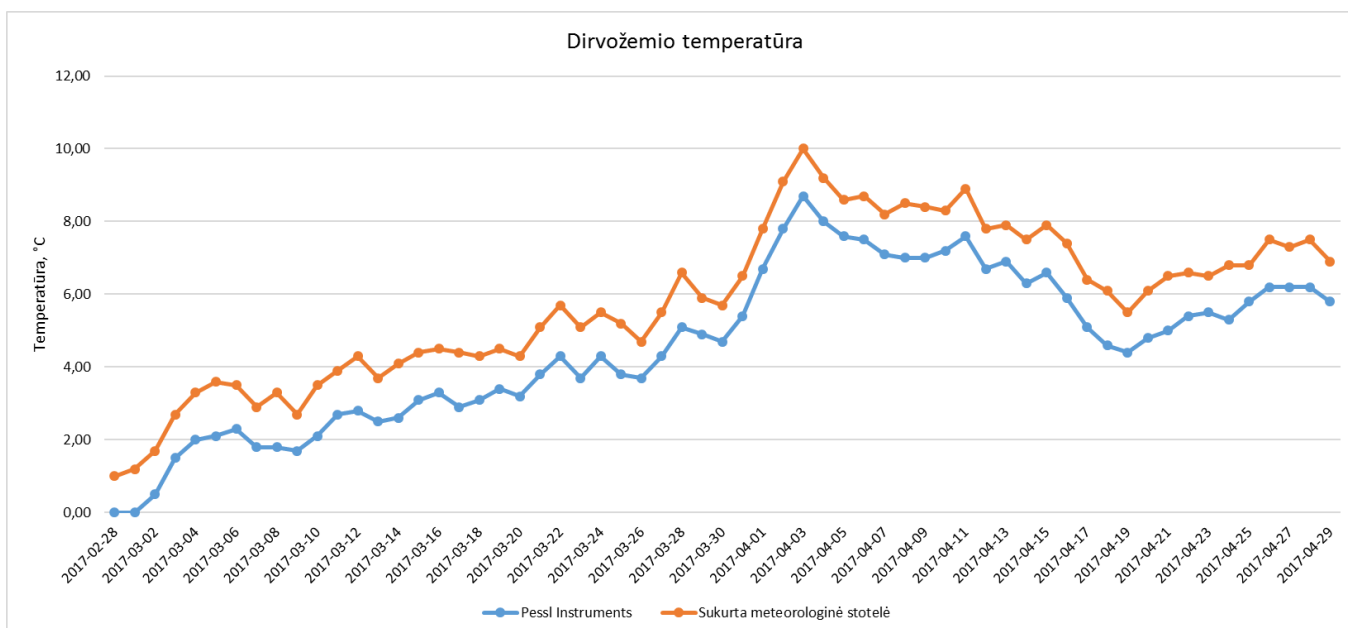
4.3.1. Orų parametrų tikslumas

Pagal atliktų matavimų duomenis nubraižėme žemiau pateiktus grafikus, kuriuose pavaizduoti „Pessl Instruments“ ir mūsų sukurtos meteorologinės stotelės vidutiniai išmatuoti paros duomenys



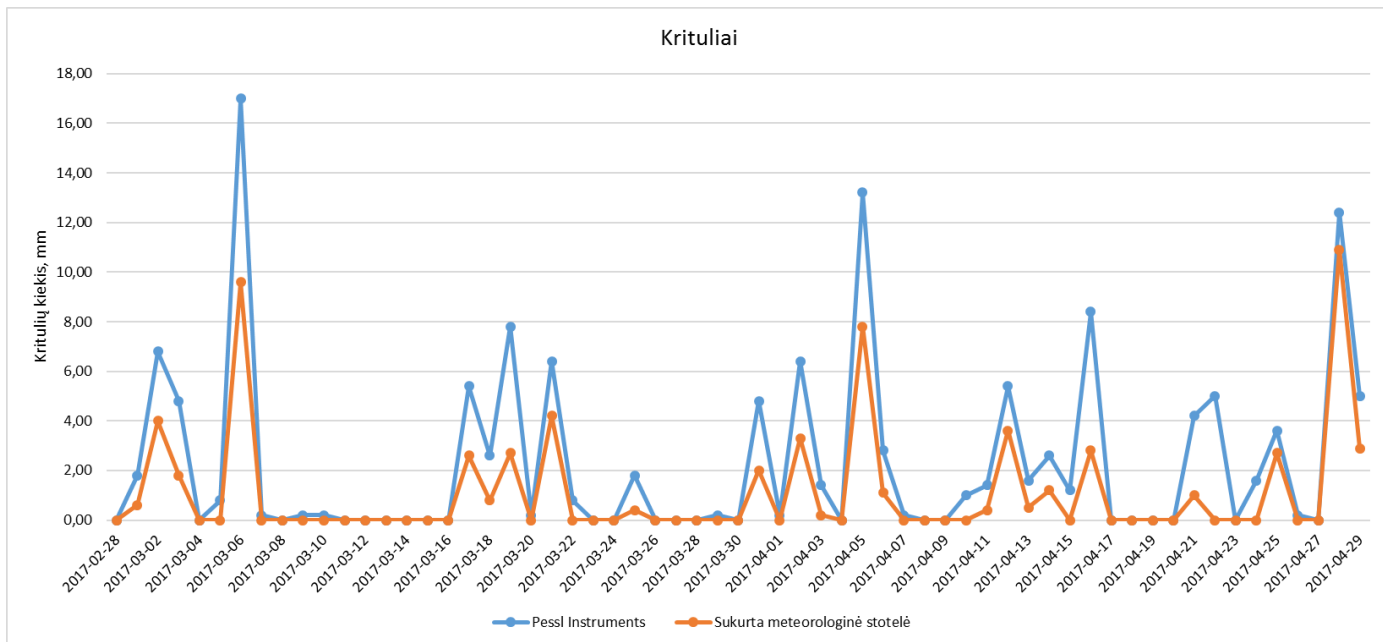
4.3.1 pav. Temperatūros matavimų grafikas

4.3.1 pav. pateiktame temperatūros matavimo grafike matome, kad sukurta stotelė matuoja temperatūra su tam tikru nukrypimu nuo profesionalios stotelės. Atidžiau pažiūrėjus į grafiką matome, kad visi duomenys yra mažesni maždaug pusės laipsnio pagal Celsijų (~0.4 - 0.6).



4.3.2 pav. Dirvožemio temperatūros matavimų grafikas

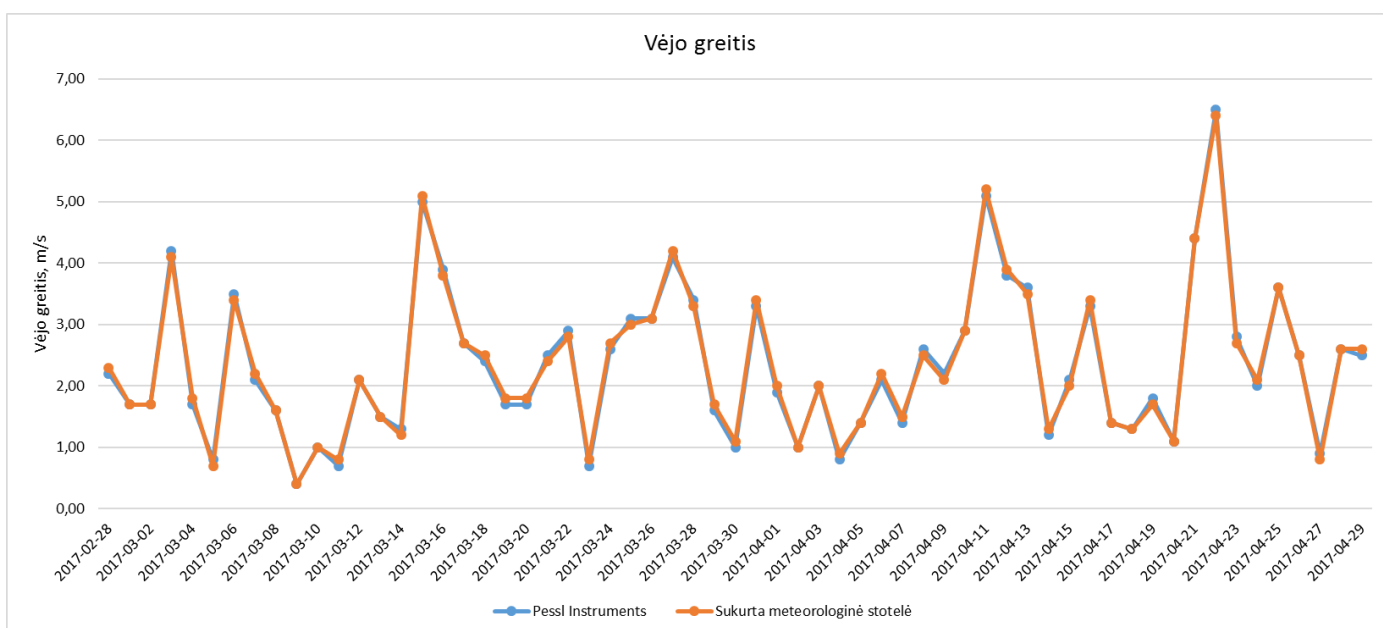
Dirvožemio temperatūros matavimo grafike (4.3.2 pav.) matome, kad sukurtoji stotelė rodo rodmenys yra didesni 1 - 1,5 laipsniu pagal Celsijų nei „Pessl Instruments“ stotelėje.



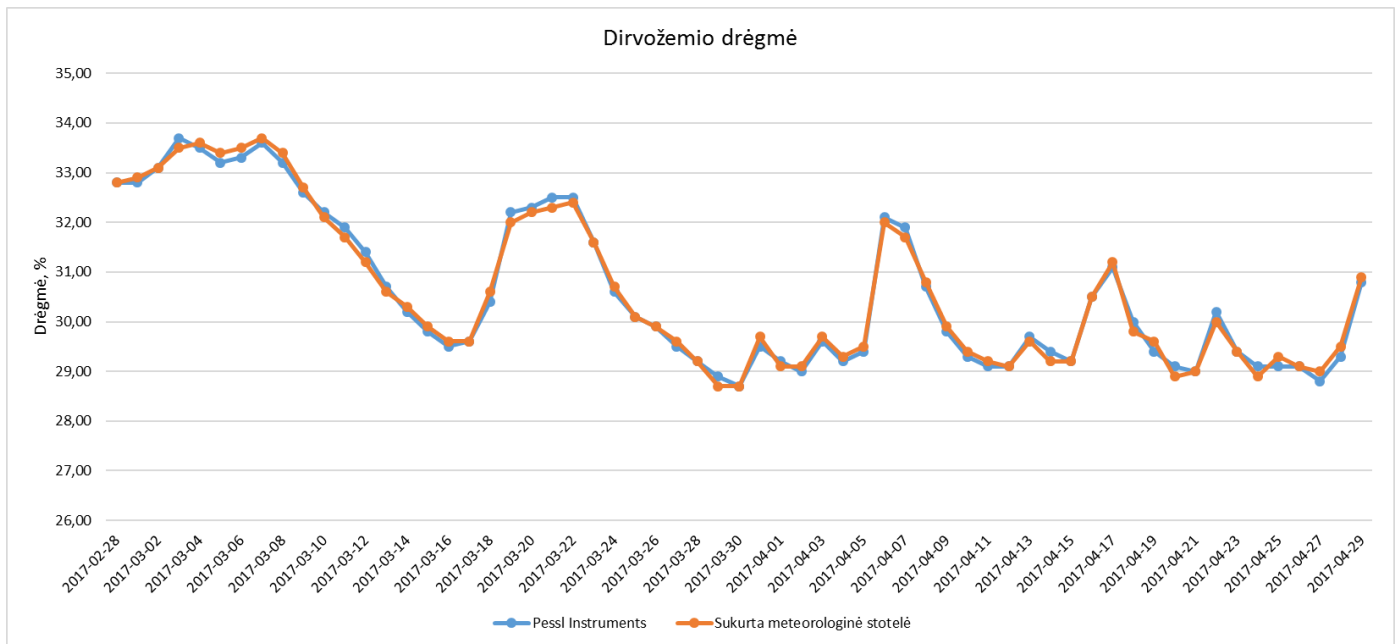
4.3.3 pav. Kritulių matavimų grafikas

Iš aukščiau pateikto kritulių kiekio grafiko (4.3.3 pav.) matome, kad mūsų sukurta stotelė kritulius matuoja klaidingai. Kai kuriais atvejais skirtumas yra dvigubas arba stotelė apskritai neužfiksuoja kritulių.

Žemiau pateikti grafikai yra vėjo greičio ir dirvožemio drėgmės. Iš jų matyti, kad stotelė gana tiksliai matuoja šiuos parametrus. Vėjo greitis skiriasi vos 0,1 m/s tikslumu, o dirvožemio drėgmė svyruoja apie 0,2 %.



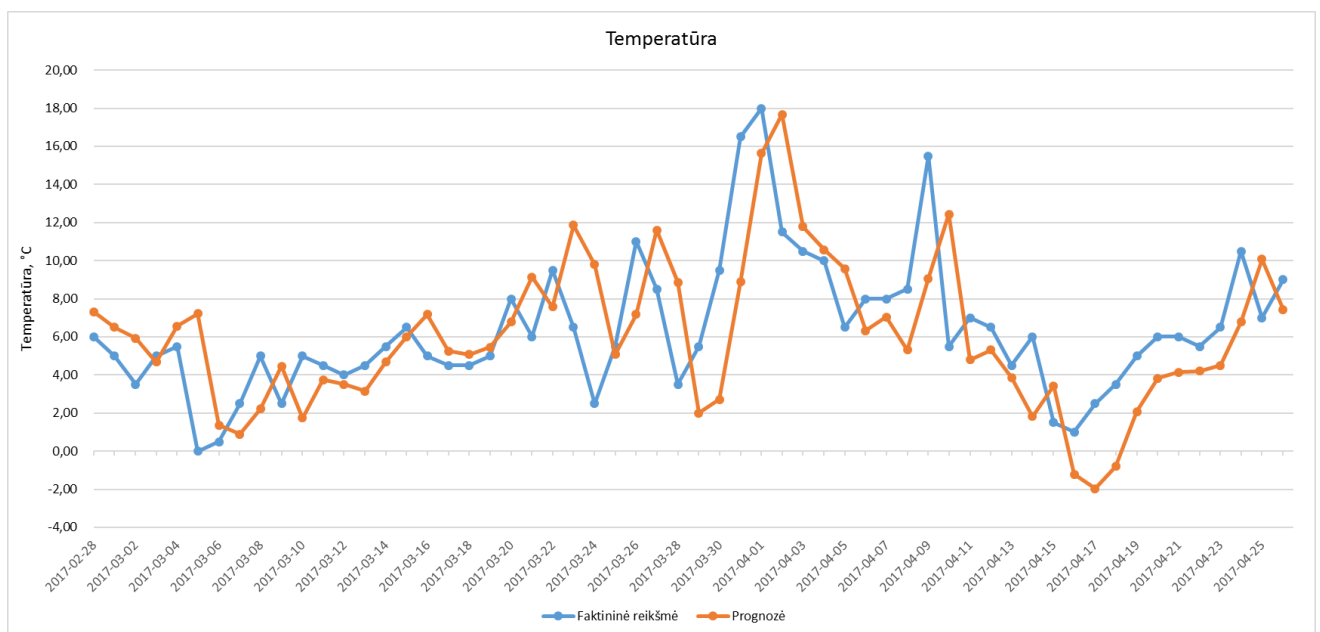
4.3.4 pav. Vėjo greičio matavimų grafikas



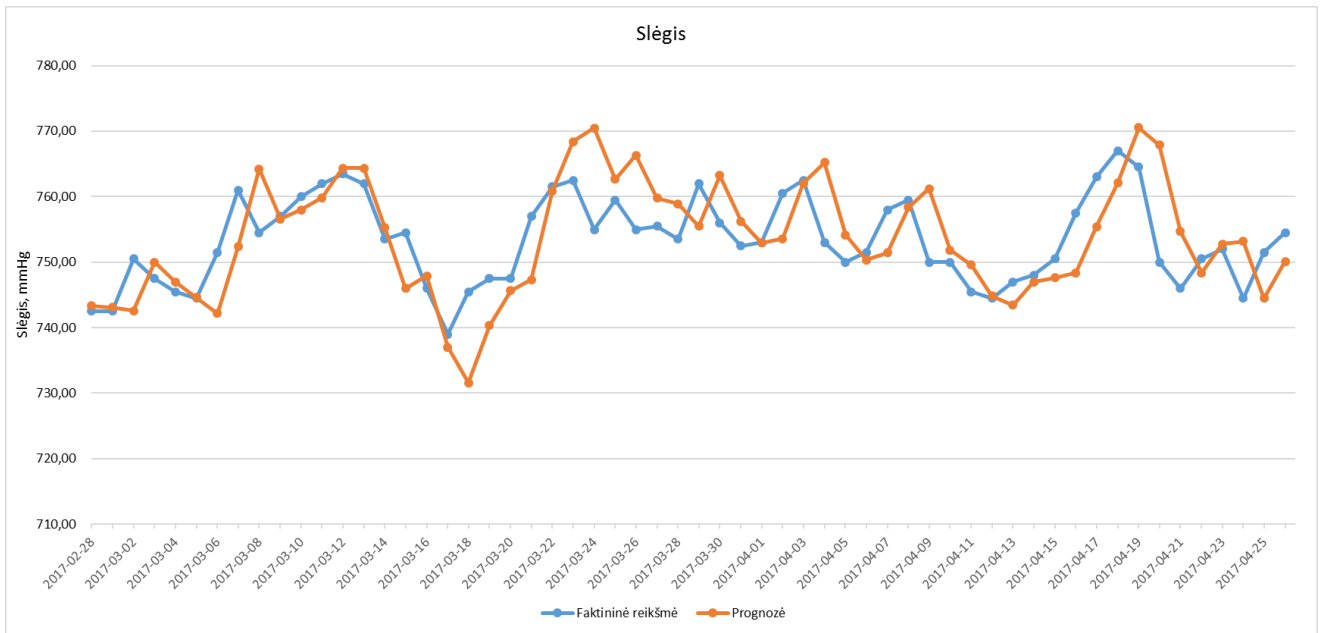
4.3.5 pav. Dirvožemio drėgmės matavimų grafikas

4.3.2. Prognozės tikslumas

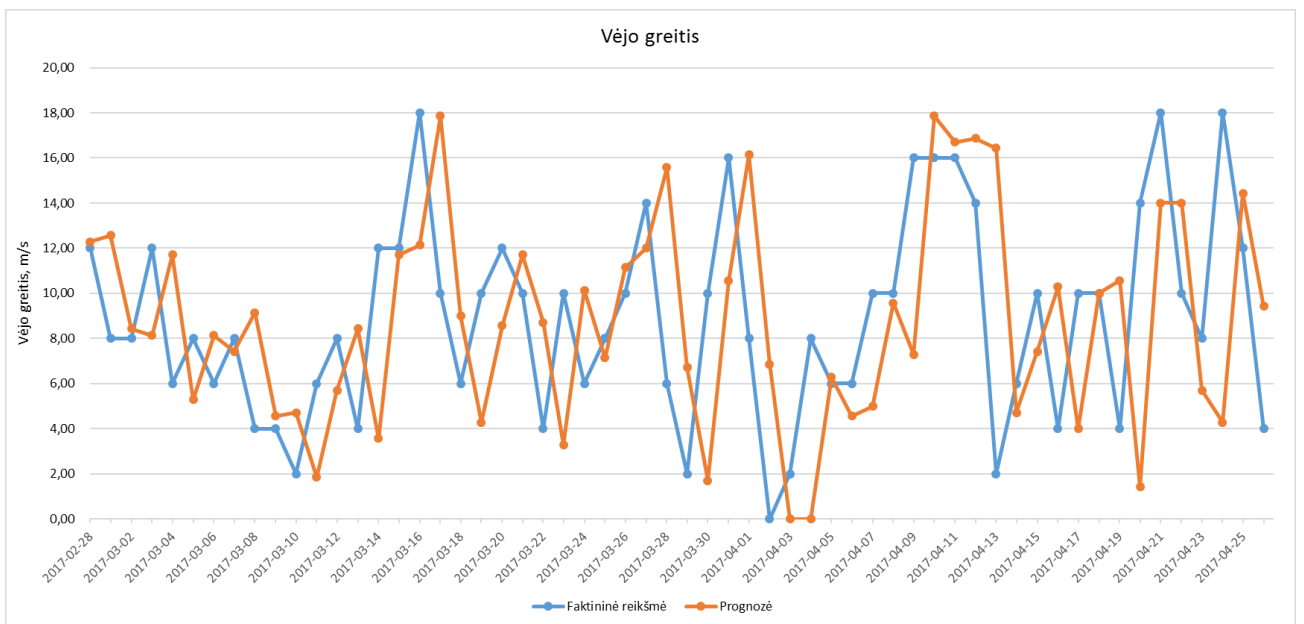
Atliekant šį tyrimą buvo svarbu turėti tikslus istorinius duomenis (mažiausiai vienerių metų senumo). Buvo kreiptasi į „Meteo“ ir „Gismeteo“ sistemas su užklausa duomenims gauti. Faktiniai istoriniai duomenys buvo paimti iš „Gismeteo“ sistemos. Šiais duomenimis operuojantis, buvo atliktas orų prognozių tikslumo tyrimas. Tyrėme keturis parametrus: temperatūrą, slėgį, vėjo greitį ir kryptį. Žemiau pateiktuose grafikuose palyginome kiekvieną rodmenį atskirai.



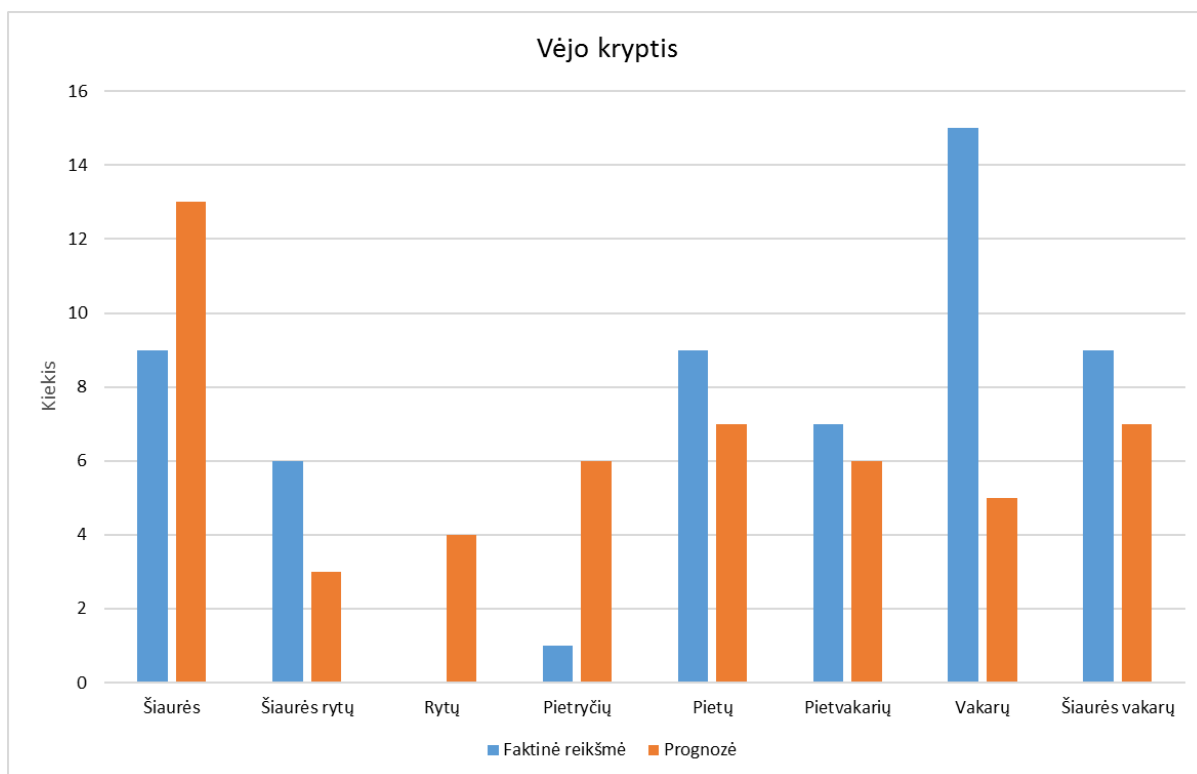
4.3.6 pav. Temperatūros prognozės grafikas



4.3.7 pav. Slēģio prognozēs grafikas



4.3.8 pav. Vējo greiċio prognozēs grafikas



4.3.9 pav. Vėjo kryptių prognozių grafikas

Pagal gautus rezultatus, sunku įvertinti algoritmo sudėtingumą. Todėl buvo nuspręsta paskaičiuoti šio algoritmo tikslumą pasinaudoję formule (1):

$$\left(1 - \frac{\sum |A-F|}{\sum A}\right) * 100\%; \quad (1)$$

čia A – faktinė reikšmė, F – išprognuota reikšmė. Pritaikius šią formulę, gavome rezultatus, kuriuos pateikėme žemiau esančioje lentelėje.

4.3.1 lentelė Prognozių tyrimo rezultatai

	Temperatūra	Slėgis	Vėjo greitis	Vėjo kryptis
Kovas	90 %	99 %	97 %	16 %
Balandis	83 %	99 %	93 %	33 %

4.4. Tyrimo išvados

Išanalizavus gautus duomenis matome, kad neatitikimų yra. Pastebėta, kad stotelės teikiamų duomenų tikslumas svyruoja panašiu dydžiu. Pavyzdžiui oro temperatūra svyruoja su 0.2 - 0.4 °C paklaida, dirvos temperatūros rodmenys yra 1 - 1.5 °C didesni. Rezultatuose nėra jokių didesnių šuolių, todėl galima matyti jog tai yra jutiklių paklaidos, kadangi projekte buvo naudojama ne patys kokybiškiausi jutikliai. Patys naudotų jutiklių gamintojai skelbia apie didelę galimą paklaidą. Pagal

gautus rezultatus galima matyti, jog kritulių jutiklis rodo nepatikimus duomenis. Greičiausiai jis yra nekokybiškas arba brokuotas.

Kadangi dauguma jutiklių teikia duomenis su tam tikra paklaida, buvo nuspręsta patobulinti sistema įdiegiant kalibravimo funkcionalumą. Tuomet kiekvieną stotelę galima bus kalibruoti, sumažinant rodiklių skirtumą iki minimumo, taip pasiekiant geresnius rezultatus. Kalibravimas vyktų panašiu principu: pastatant prie patikimos ir sertifikuotos stotelės ir lyginant rodmenų skirtumus.

Išanalizavę prognozių grafikus, bei pasiskaičiavę rezultatų tikslumą gavome gan aukšto patikimumo prognozes. Slėgio prognozavimas buvo net 99% tikslumu. Žinoma, slėgis nėra labai kintamas reiškinys, todėl jį išprognozuoti yra lengviau. Temperatūra, kovo mėnesyje buvo išprognozuota 90% tikslumu, o balandį šiek tiek mažiau - 83%. Vėjo greitis taip pat buvo išprognozuotas dideliu tikslumu, kovą - 97%, balandį - 93%. Kaip ir buvo tikėtasi prasčiausiai buvo išprognozuota vėjo kryptis. Vėjo kryptis yra labai nepastovus reiškinys, todėl jam prognozuoti vien istorinių duomenų nepakanka, reikia ir palydovų informacijos apie ciklonų padėtį.

Atliekant šį tyrimą taip pat buvo pastebėtos ir pačios autonominės stotelės trūkumai. Pora kartų buvo išsikrovusi stotelės baterija. Taip įvykdavo tuomet, kai saulės nebūdavo 6-7 dienas iš eilės. Buvo nuspręsta padidinti akumuliatoriaus talpą, bei pakeisti iš „Raspberry Pi“ į „Raspberry Pi Model A+“, kuris eikvoja mažiau energijos.

5. EKSPERIMENTINĖ DALIS

5.1. Eksperimentinio tyrimo tikslas

Šio darbo eksperimentinio tyrimo tikslas yra patikrinti ar keičiant „slankiojančio lango“ algoritmo lango dydį ir jų kiekį galima pagerinti prognozių tikslumą. Taip pat palyginti ar įtakoja lango dydis, bei jų kiekis algoritmo greitaveiką. Palyginti faktinius duomenis, originalaus, bei modifikuoto algoritmo išprognuotus orų parametrus.

5.2. Eksperimentinio tyrimo aprašymas

Atliekant literatūros analizę buvo analizuojami galimi orų prognozavimo būdai. Kadangi turėjome ribotą resursų kiekį, buvo pasirinktas sprendimas patikslinti orų prognozę operuojantis vien tik istoriniais duomenimis. Todėl pasirinkome „slankiojančio lango“ algoritmą.

Šio algoritmo pagrindinis principas lyginti einamų dienų orų parametrus su praėjusių metų pačio laikotarpio duomenimis. Rasti mažiausiai skirtumų turintį intervalą pasitelkiant Euklido atstumo skaičiavimo metodika ir tuomet paskaičiuoti skirtumą tarp šių dviejų duomenų imčių. Gautą skirtumą pridedame prie šios dienos parametrų reikšmių ir gauname ateinančios dienos prognozę.

Kaip matome, algoritmo tikslumą labiausiai įtakoja atitinkančių duomenų imties radimas. Sistemoje įgyvendinome prognozės patvirtinimo funkcionalumą. T.y. jei prognozė atitiko realybę tuomet vartotojas patvirtina tai. Tuomet sekantį kartą bus naudojami tie patys algoritmo parametrai, priešingu atveju bus mažinamas arba didinamas lango dydis ir bandoma pagerinti algoritmo patikimumą.

Eksperimente norime iširti prie kokių konfigūracijų algoritmas teikia tikslesnes prognozes ir nuo ko tai gali priklausyti. Pasirenkant „slankiojančio lango“ dydį ir kiekį buvo panaudota tiesinė regresija.

Šiame eksperimentiniame tyrimo palyginsime penkias skirtingas lango dydžio ir kiekių kombinacijas:

- Langų kiekis: 3; Lango dydis: 12 dienų;
- Langų kiekis: 5; Lango dydis: 10 dienų;
- Langų kiekis: 8; Lango dydis: 7 dienų; - standartinė algoritmo kombinacija
- Langų kiekis: 11; Lango dydis: 4 dienų;
- Langų kiekis: 13; Lango dydis: 2 dienų;

Maksimalus lango dydis 12 dienų pasirinktas todėl, kad Lietuvos klimatas turi 4 metų laikus. Taigi klimatas yra kintamas ir analizuoto didesnę imti nei 12 dienų tiesiog nėra prasmės.

5.3. Eksperimentinio tyrimo eiga

Prieš atliekant šį eksperimentą buvo paimti dviejų metų istoriniai duomenys iš „Gismeteo“ sistemos. Operuojantis šiais duomenimis buvo leidžiamas algoritmas su skirtingomis konfigūracijomis ir prognozuojami praėjusių metų orai. Kadangi jau žinome kokie buvo faktiniai orų parametrai, mes juos galime palyginti ir paskaičiuoti kokių tikslumu algoritmas teikia orų prognozes.

Taip pat kas kartą inicijuojant algoritmą buvo skaičiuojamas jo veikimo laikas su skirtingomis konfigūracijomis.

5.4. Eksperimentinio tyrimo rezultatai

Žemiau pateiktose lentelėse yra algoritmų su skirtingomis konfigūracijomis prognozių tikslumai. Skaičiuojant tikslumą buvo taikoma formulė (1), kuri yra pateikta 4.3.2 skyrelyje.

5.4.1 lentelė Langų kiekis – 3, lango dydis 12 d. tikslumas

		Prognozės tikslumas, %			
Mėnuo		Temperatūra	Slėgis	Vėjo greitis	Vėjo kryptis
2016	Balandis	96,35	99,96	84,19	6,90
	Gegužė	98,45	99,91	89,62	16,13
	Birželis	97,14	99,88	95,18	10,34
	Liepa	99,43	99,96	94,11	26,67
	Rugpjūtis	93,14	99,89	98,10	25,81
	Rugsėjis	99,40	99,89	84,69	6,67
	Spalis	99,54	99,93	91,87	41,94
	Lapkritis	18,04	99,89	97,22	10,00
	Gruodis	68,30	99,97	97,32	12,90
2017	Sausis	33,02	99,92	75,45	16,13
	Vasaris	20,19	99,13	85,57	17,86
	Kovas	86,88	99,69	90,04	16,13
	Balandis	92,02	99,83	99,15	22,22

5.4.2 lentelė Langų kiekis – 5, lango dydis - 12 d. tikslumas

		Prognozės tikslumas, %			
Mėnuo		Temperatūra	Slėgis	Vėjo greitis	Vėjo kryptis
2016	Balandis	99,15	99,89	83,60	3,45
	Gegužė	99,94	99,97	91,89	22,58
	Birželis	97,25	99,91	98,91	17,24
	Liepa	99,52	99,93	95,72	20,00
	Rugpjūtis	94,75	99,91	96,63	22,58
	Rugsėjis	96,91	99,81	88,22	16,67
	Spalis	93,54	99,88	92,32	32,26
	Lapkritis	20,43	99,83	97,72	16,67
	Gruodis	90,70	99,91	94,41	12,90

2017	Sausis	44,81	99,87	76,88	6,45
	Vasaris	21,83	99,18	86,38	21,43
	Kovas	87,25	99,71	89,88	6,45
	Balandis	89,18	99,86	97,87	11,11

5.4.3 lentelē Langu kiekis – 8, lango dydis - 7 d. tikslumas

		Prognozēs tikslumas, %			
Mēnuo		Temperatūra	Slēgis	Vējo greitis	Vējo kryptis
2016	Balandis	98,73	99,77	88,50	17,24
	Gegužē	95,13	99,97	97,30	16,13
	Birželis	98,42	99,98	97,46	17,24
	Liepa	95,41	99,91	93,17	26,67
	Rugpjūtis	99,39	99,93	98,34	16,13
	Rugsējis	94,13	99,75	90,31	20,00
	Spalis	93,90	99,99	91,43	32,26
	Lapkritis	37,17	99,99	99,78	20,00
	Gruodis	37,87	99,93	98,18	6,45
2017	Sausis	64,09	99,93	81,74	16,13
	Vasaris	24,47	99,49	86,36	10,71
	Kovas	90,66	99,98	97,00	12,90
	Balandis	83,59	99,90	95,85	18,52

5.4.4 lentelē Langu kiekis – 11, lango dydis - 4 d. tikslumas

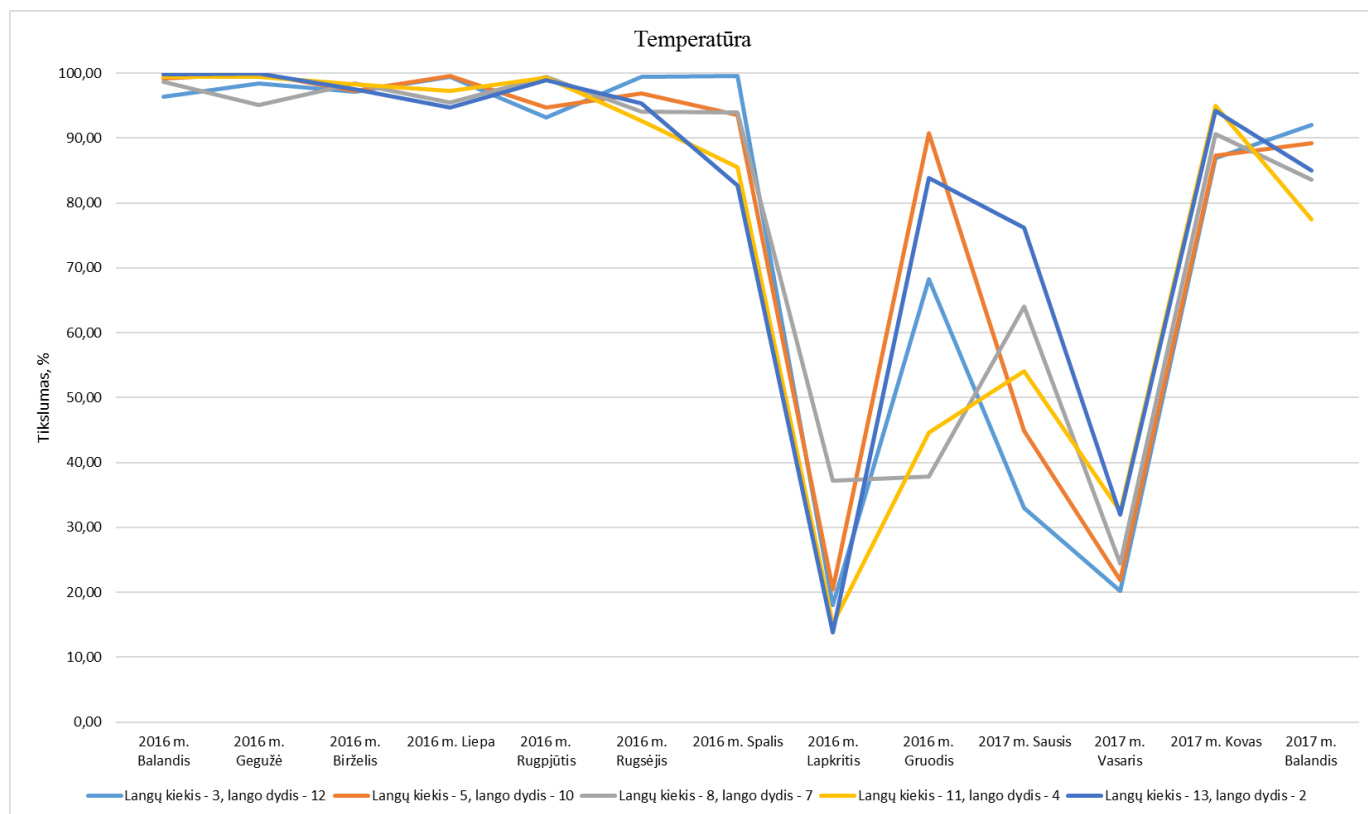
		Prognozēs tikslumas, %			
Mēnuo		Temperatūra	Slēgis	Vējo greitis	Vējo kryptis
2016	Balandis	99,44	99,78	90,07	20,69
	Gegužē	99,42	99,99	95,99	16,13
	Birželis	98,23	99,95	97,72	10,34
	Liepa	97,24	99,81	95,27	3,33
	Rugpjūtis	99,33	99,81	94,48	19,35
	Rugsējis	92,63	99,80	89,73	16,67
	Spalis	85,45	99,80	96,90	25,81
	Lapkritis	15,07	99,95	98,40	16,67
	Gruodis	44,57	99,83	94,05	3,23
2017	Sausis	54,07	99,89	86,82	16,13
	Vasaris	32,44	99,85	91,67	14,29
	Kovas	95,02	99,87	99,80	16,13
	Balandis	77,41	99,92	92,34	7,41

5.4.5 lentelė Langų kiekis – 13, lango dydis - 2 d. tikslumas

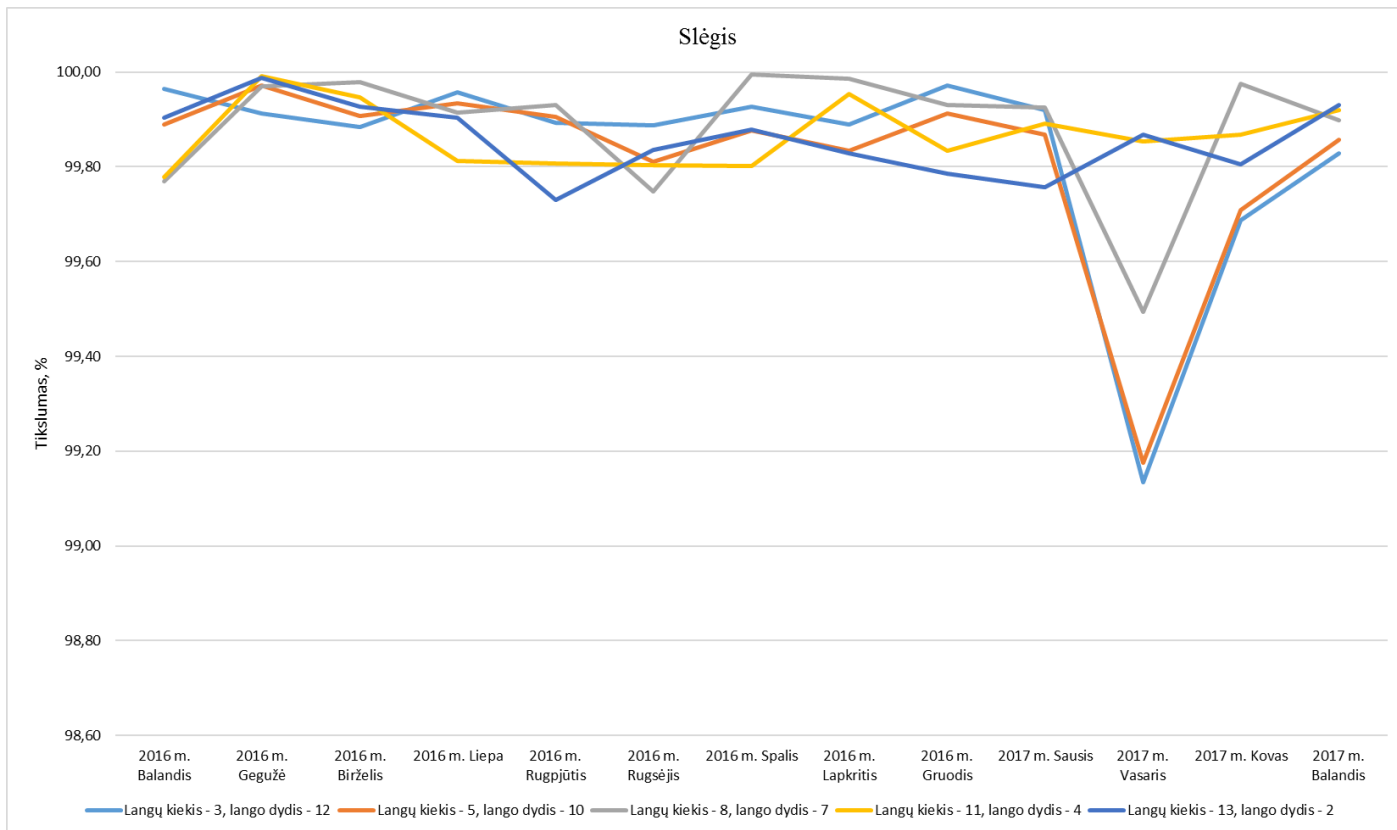
	Mėnuo	Prognozės tikslumas, %			
		Temperatūra	Slėgis	Vėjo greitis	Vėjo kryptis
2016	Balandis	99,84	99,90	99,82	31,03
	Gegužė	99,96	99,99	90,33	22,58
	Birželis	97,47	99,93	90,22	6,90
	Liepa	94,67	99,90	95,05	16,67
	Rugpjūtis	98,94	99,73	95,16	12,90
	Rugsėjis	95,38	99,84	99,26	20,00
	Spalis	82,63	99,88	97,65	25,81
	Lapkritis	13,87	99,83	98,00	16,67
	Gruodis	83,78	99,78	96,21	12,90
2017	Sausis	76,12	99,76	88,87	9,68
	Vasaris	31,95	99,87	91,26	7,14
	Kovas	94,25	99,80	92,58	19,35
	Balandis	85,00	99,93	98,64	3,70

Paskaičiavę algoritmo tikslumą su skirtingais langų dydžiais sunku įvertinti kuris yra tikslesnis. Todėl buvo nuspręsta gautus duomenis atvaizduoti grafike.

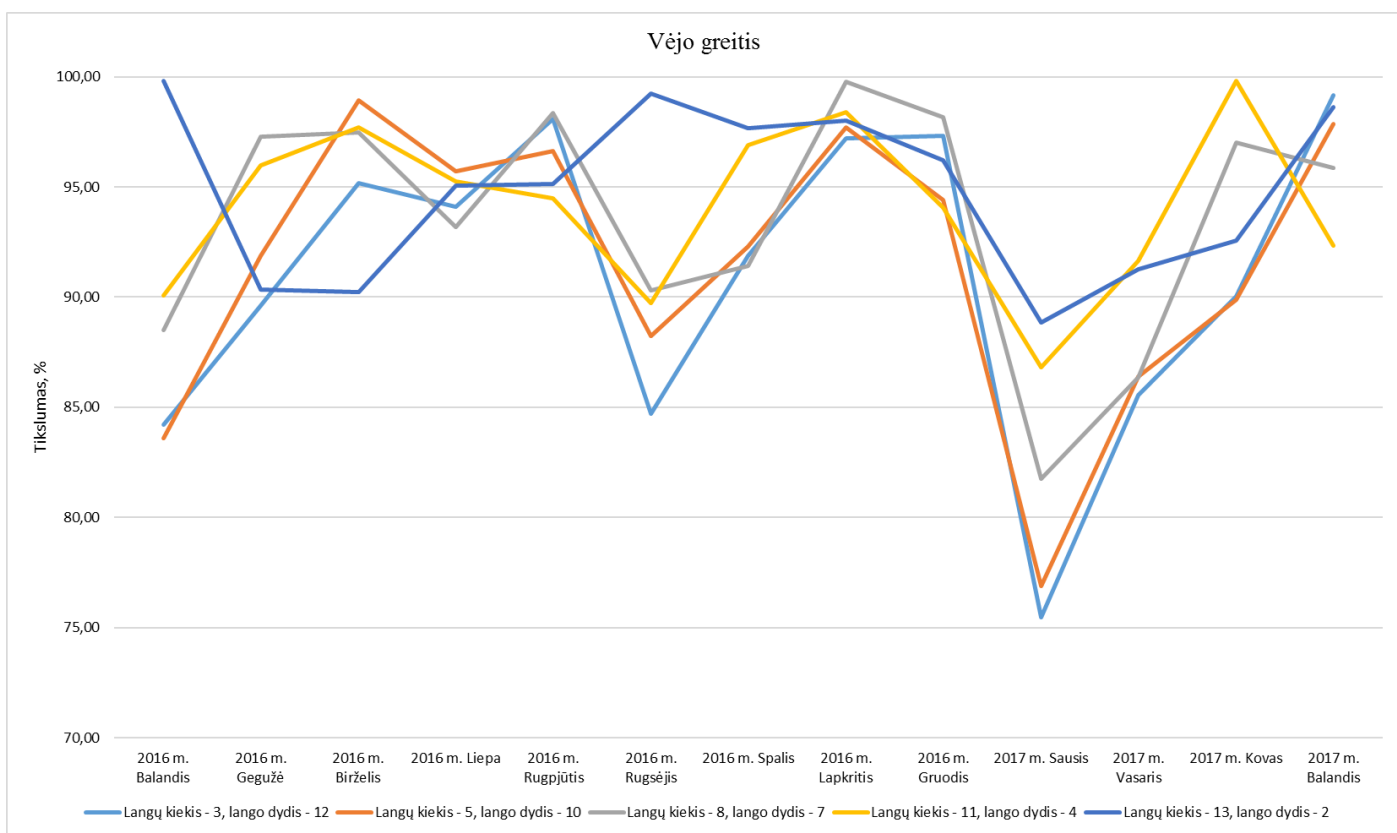
Žemiau pateiktuose grafikuose yra pavaizduoti eksperimentinio tyrimo metu gauti rezultatai.



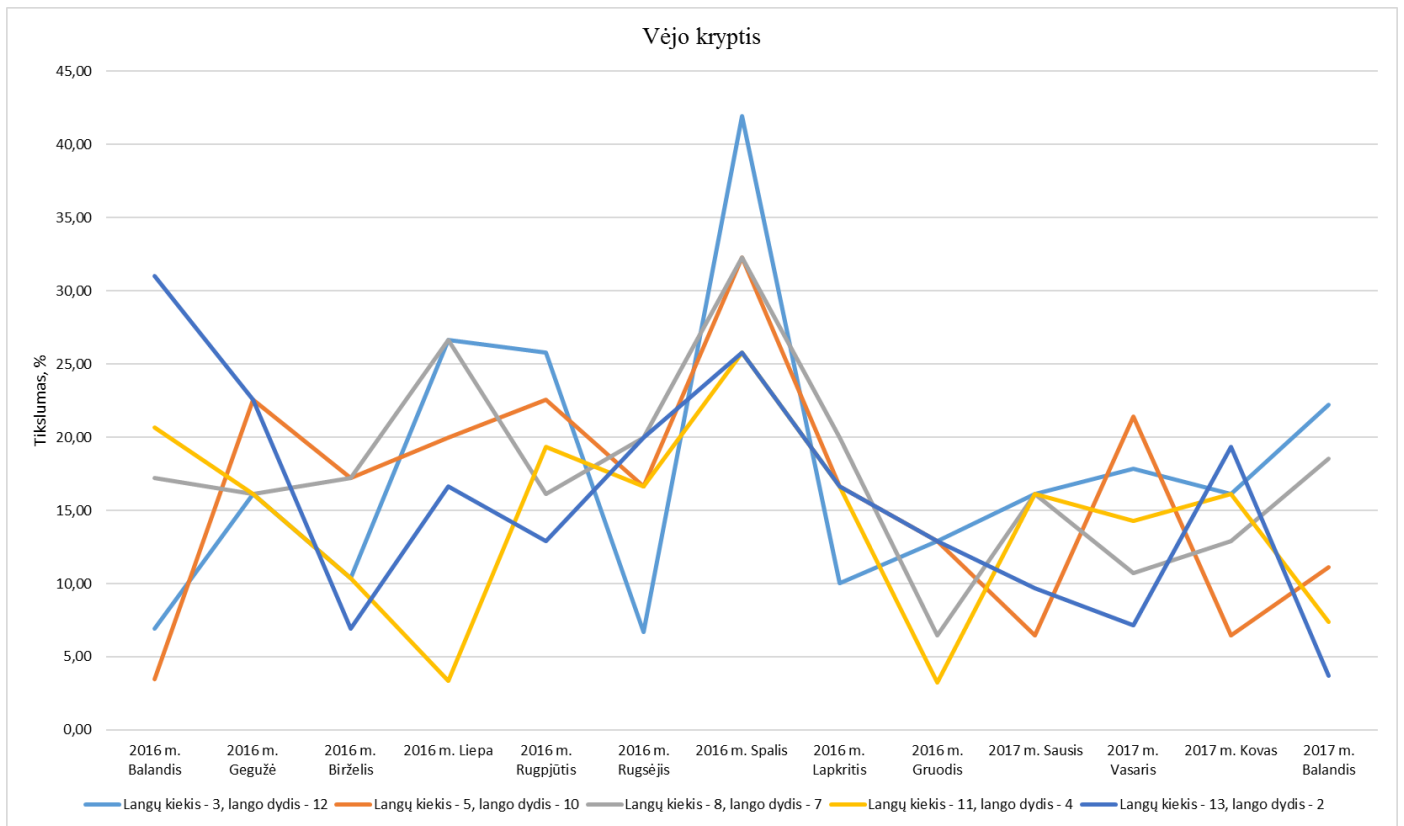
5.4.1 pav. Temperatūros prognozių tikslumo grafikas



5.4.2 Slėgio prognozių tikslumo grafikas



5.4.3 pav. Vėjo greičio prognozių tikslumo grafikas

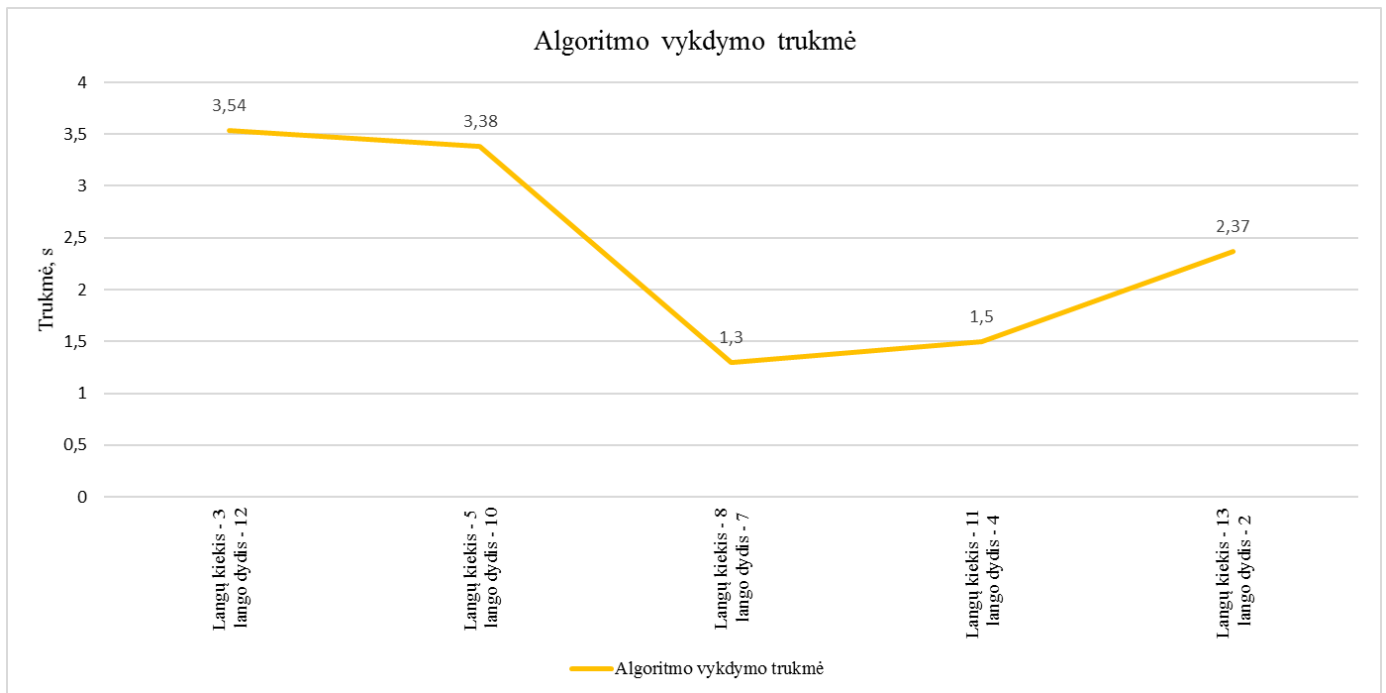


5.4.4 pav. Vėjo krypties prognozių tikslumo grafikas

Kaip matome iš grafikų ir lentelių geriausiai yra išprognozuotas slėgis, apie 99% tikslumu. Slėgis nėra stipriai kintantis reiškinys, todėl jo prognozavimo tikslumas yra aukštas. Visgi iš 5.4.2 paveikslo matome, kad optimaliausią prognozę pateikė algoritmas su originalia konfigūracija t.y. su 8 langais, kurių dydis 7 dienos.

Iš vėjo greičio ir temperatūros grafikų matome, kad keičiantis metų laikams (t.y. pereinant iš vieno metų laiko į kitą) tikslesnes prognozes pateikia algoritmas su mažesniu langu bei didesniu jų kiekiu. O esant nusistovėjusiems orams geresnius rezultatus rodo algoritmas su didesniu langu. Visgi temperatūros prognozavimo tikslumas yra apie ~80%. Vasarį algoritmui prastai sekėsi prognozuoti, greičiausiai todėl, kad šiemet vasario mėnuo buvo šaltesnis, nei praeitų metų vasario mėnuo. Vėjo greičio prognozavimo tikslumas yra apie 90%, tai yra labai geras rodiklis.

Prasčiausiai algoritmui sekėsi prognozuoti vėjo kryptį. Tiksliausia prognozė buvo 2016 m. spalio mėnesiui ir ji siekė vos 40%. Čia geriausius rezultatus taip pat parodė algoritmas su standartine konfigūracija. Vėjo kryptis yra kintamas reiškinys ir sunkiai išprognozuojamas pasitelkiant vien tik istorinius duomenis, tokiam reiškiniui išprognozuoti reikia daugiau informacijos (palydovų teikiamų duomenų ir t.t.).



5.4.5 pav. Algoritmo vykdymo trukmė

Atliekant eksperimentą buvo matuojama algoritmo greitaveika. 5.4.5 paveiksle pateiktas grafikas, kuriame atvaizduotas vidutinis algoritmo veikimo laikas su skirtingomis konfigūracijomis. Nors algoritmas neapkrauna sistemos ir gali laisvai veikti prie bet kurios konfigūracijos visgi matome, kad esant dideliame langų kiekiui ar dideliame langui algoritmo veikimo laikas didėja. Lyginant su optimaliausiu variantu (kai langų kiekis 8, o dydis 7 dienos), veikimo laikas padidėja apie 2-3 kartus.

5.5. Eksperimentinio tyrimo išvados

Atlikę eksperimentą pastebėjome, kad langų kiekis ir jų ilgis dienomis turi įtaką prognozėms. Vykdomo laikas nors ir nėra labai didelis ir neapraunantis sistemos, tačiau greičiausiai veikia su 8-iais langais, kurių dydis yra 7 dienos. Ši langų kiekio ir dydžio kombinacija taip pat ir optimaliausiai prognozuoja orus.

Įsigilinę į rezultatus pastebėjome tendenciją, kad mėnesiais kuomet keičiasi metų laikai, geresnius rezultatus teikia algoritmas su mažesniu lango dydžiu, o metų laikų viduryje – algoritmas su didesniu langu.

Taigi Lietuvos klimatui prognozuojant šiuo algoritmu reiktų rinktis tokias konfigūracijas: mėnesiuose kuomet keičiasi metų laikai reiktų naudoti daugiau „langų“ su mažesniu dienų skaičiumi – tuomet algoritmas pastebės didesnius rodiklių svyravimus, o metų laikų viduryje, kuomet oro rodikliai nusistovi – naudoti didesnius „langus“.

6. IŠVADOS

1. Analizės metu buvo išsiaiškinta kokių technologijų, aparatūrinės įrangos, strategijų reikės, kad įgyvendinti autonominę meteorologinę stotelę. Kad taupyti stotelės energiją buvo nuspręsta daugiau skaičiavimo ir logikos perkelti į serverio pusę, bei įgyvendinti stotelės konfigūravimo galimybę (keisti siunčiamų duomenų dažnį).
2. Analizės metu buvo analizuojami skirtingi prognozavimo algoritmai ir metodikos. Iš visų peržiūrėtų algoritmų/metodikų buvo pasirinktas slankiojančio lango metodas, kadangi jis yra paremtas prognozavimu naudojant istorinius duomenis. Šis metodas labiausiai tiko pagal mūsų turimus resursus. Ši algoritmą patobulinome ir įgyvendinome funkcionalumą kurio pagalba algoritmas atsižvelgdamas į vartotojo pasitenkinimą keičia algoritmo parametrus – taip dar pagerindamas orų prognozių tikslumą.
3. Analizės metu taip pat buvo išanalizuoti esami sprendimai rinkoje. Buvo nuspręsta įgyvendinti augalų ligų modelius ir ligų sekimo funkcionalumą. Taip pat patogesniai perspėjimų gavimui, bei duomenų peržiūrai buvo nuspręsta sukurti mobilią aplikaciją.
4. Projektuojant sistemą, buvo nustatyti funkciniai ir nefunkciniai reikalavimai sistemos reikalavimai, kurie buvo detalizuoti pagal užsakovo reikalavimus. Apibrėžti panaudojimo atvejai.
5. Atlikus sukurtos meteorologinės stotelės ir sistemos tyrimą buvo pastebėta, kad kai kurie jutikliai turi netoleruotiną paklaidą. Kadangi stotelėje buvo naudojami ne pakankamai aukštos kokybės jutikliai, buvo nuspręsta sukurti funkcionalumą, kurio pagalba galima būtų stoteles kalibruoti ir sumažinti paklaidos dydį iki minimumo.
6. Tyrimo metu buvo pastebėti autonominės stotelės trūkumai. Esant daugiau nei 7 niūrioms (be saulės) dienoms iš eilės išsikraudavo baterija. Buvo nuspręsta padidinti baterijos talpą, bei pakeisti „Raspberry Pi“ kompiuterį į labiau ekonomiškesnį – „Raspberry Pi Model A+“.
7. Atliktas eksperimentinis tyrimas, atskleidė jog keičiant slankiojančio algoritmo konfigūraciją su tais pačiais duomenimis įtakoja prognozių tikslumą. Tikslesnėms prognozėms Lietuvoje reiktų pasirinkti šias konfigūracijas: mėnesiuose kuomet keičiasi metų laikai reiktų naudoti daugiau „langų“ su mažesniu dienų skaičiumi – tuomet algoritmas pastebės didesnius rodiklių svyravimus, o viduriniaisiais metų laikų mėnesiais, kuomet oro rodikliai nusistovi – naudoti didesnius „langus“.

7. LITERATŪRA

- [1] „Pessl Instruments,“ [Tinkle]. Available: http://metos.at/page/docs/EN_Catalogue_optimised.pdf. [Kreiptasi 01 12 2015].
- [2] L. L. Lai, H. Braun, Q. P. Zhang et al., „Intelligent weather forecast,“ įtraukta *Proceedings of the International Conference on Machine Learning and Cybernetics*, Shanghai, China, 2004, p. 4216–4221.
- [3] J. Gill, B. Singh, and S. Singh, „Training back propagation neural networks with genetic algorithm for weather forecasting,“ įtraukta *Proceedings of the 8th IEEE International Symposium on Intelligent Systems and Informatics*, 2010, p. 465–469.
- [4] I. S. Isa, S. Omar, Z. Saad, N. M. Noor, and M. K. Osman, „Weather forecasting using photovoltaic system and Neural Network,“ įtraukta *Proceedings of the 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, 2010, p. 96–100.
- [5] L. Zuoyong, C. Zhenpei, and L. Jitao, „A model of weather forecast by fuzzy grade statistics,“ įtraukta *Fuzzy Sets and Systems*, 1988, p. 275–281.
- [6] R. Collobert and S. Bengio, „Support vector machines for large-scale regression problems,“ *Journal of Machine Learning Research*, t. 1, p. 143–160, 2001.
- [7] P.-S. Yu, S.-T. Chen, and I.-F. Chang, „Support vector regression for real-time flood stage forecasting,“ *Journal of Hydrology*, t. 328, p. 704–716, 2006.
- [8] Amanpreet Kaur, J K Sharma, and Sunil Agrawal, „Artificial neural networks in forecasting maximum and minimum relative humidity,“ *International Journal of Computer Science and Network Security*, pp. 197-199, 2011.
- [9] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham, „An ensemble of neural networks for weather forecasting,“ įtraukta *Neural Comput & Applic*, 2004, p. 112–122.
- [10] Rosmina Bustami, Nabil Bessaih, Charles Bong, and Suhaila Suhaili, „Artificial neural network for precipitation and water level predictions of bedup river,“ *IAENG International Journal of Computer Science*, pp. 228-233, 2007.
- [11] Kumar Abhishek, M.P. Singh, Saswata Ghosh, Abhishek Anand, „Weather Forecasting Model using Artificial Neural Network,“ *Procedia Technology*, t. 4, pp. 311-318, 2012.
- [12] Piyush Kapoor and Sarabjeet Singh Bedi, „Weather Forecasting Using Sliding Window Algorithm,“ *International Scholarly Research Notices*, p. 2, 2013.
- [13] Subhash Lele and Joan T. Richsmeiter, „Euclidean matrix analysis,“ *American journal of physical anthropology*, pp. 415-427, 1991.
- [14] „Modelling Disease Progress,“ APS, [Tinkle]. Available: <http://www.apsnet.org/edcenter/advanced/topics/EpidemiologyTemporal/Pages/ModellingProgress.aspx>. [Kreiptasi 02 12 2015].
- [15] „SparkFun Weather Shield,“ SparkFun Electronics, [Tinkle]. Available: <https://www.sparkfun.com/products/13956>. [Kreiptasi 10 12 2015].
- [16] „Weather Meters,“ SparkFun Electronics, [Tinkle]. Available: <https://www.sparkfun.com/products/8942>. [Kreiptasi 10 12 2015].
- [17] Nik J. Cunniffe, Britt Koskella, C. Jessica E. Metcalf, Stephen Parnell, Christopher A. Gilligan, Tim R. Gottwald, „Thirteen challenges in modelling plant diseases,“ *Epidemics*, t. 10, pp. 6-10, 2015.

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

ARMA (Autoregressive-move-average model) – autoregresijos judėjimo vidurkio modelis

Bin (Binary number) – dvejetainė skaičiavimo sistema

Fuzzy (Fuzzy logic) – tai daugiareikšmė logika, kurioje teisingos reikšmės iš parametų aibės gali būti bet kokia reikšmė iš realių skaičių nuo 0 iki 1.

ANN (Artificial Neural Network) – dirbtinis neuroninis tinklas

ERNN (Elman Recurrent Neural Network) – Elmano periodinis neuroninis tinklas

MPL (Multilayer Perceptron Networks) – daugiasluoksnis Perceptronų (kompiuterio modelis ar kompiuterizuota mašina, kuri imituoja žmogaus smegenų gebėjimą atpažinti ir atskirti) tinklas

HFM (Hopfield Model) – Hopfield'o modelis

PĮ – programinė įranga

API (Application Programming Interface) - tai sąsaja, kurią suteikia kompiuterinė sistema, biblioteka ar programa tam, kad programuotojas per kitą programą galėtų pasiekti jos funkcionalumą ar apsikeistų su ja duomenimis.

GPRS (General Packet Radio Service) - mobiliojo ryšio technologija, skirta duomenų perdavimui GSM ir D-AMPS tinkluose.

HTTP (The Hypertext Transfer Protocol) – užklauso - atsakymo protokolas, jungiantis klientą ir serverį.

GCM (Google Cloud Messaging) – technologija, skirta siųsti pranešimams iš serverio į mobilųjį telefoną.

ORM (Object-relational Mapping) – programavimo technika, kai duomenys konvertuojami į objektus.

GPS (Global Positioning System) - globali padėties nustatymo sistema

JSON (JavaScript Object Notation) - yra atviro standarto formatas, perduodantis duomenų objektus, sudarytus iš atributo ir reikšmės porų, lengvai skaitomame tekste.

DI (Dependency injection) - tai PĮ kūrimo šablonas, kuris apsprendžia priklausomybes.

Raspberry Pi – mini kompiuteris.

Arduino - mikrokontroleris.