

VILNIAUS UNIVERSITETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ KATEDRA

Anomalių aptikimas Interneto Prieigos Stebėsenos Sistemos įrašuose

Detection of anomalies in Internet Access Monitoring System data

Bakalauro darbas

Atliko:	Jokūbas Rusakevičius	(parašas)
Darbo vadovas:	asist. dr. Vytautas Valaitis	(parašas)
Darbo recenzentas:	j. asist. Linas Petkevičius	(parašas)

Vilnius – 2019

Santrauka

Glaustai aprašomas darbo turinys: pristatoma nagrinėta problema ir padarytos išvados. Santraukos apimtis ne didesnė nei 0,5 puslapio. Santraukų gale nurodomi darbo raktiniai žodžiai.

Raktiniai žodžiai: raktinis žodis 1, raktinis žodis 2, raktinis žodis 3, raktinis žodis 4, raktinis žodis 5

Summary

Santrauka anglų kalba. Santraukos apimtis ne didesnė nei 0,5 puslapio.

Keywords: keyword 1, keyword 2, keyword 3, keyword 4, keyword 5

TURINYS

IVADAS	5
Problematika	5
1. MEDŽIAGOS DARBO TEMA DĖSTYMO SKYRIAI	6
1.1. Poskyris	6
2. DUOMENŲ ANALIZĖS EKSPERIMENTAS NAUDOJANT „MACROBASE SQL“ MO- DULĮ	7
2.1. Duomenų rinkinys	7
2.1.1. Eksperimentui pritaikyti duomenys	7
2.1.1.1. Duomenų aprašymas	7
2.1.1.2. Duomenų paruošimas	10
2.2. Eksperimento aplinkos paruošimas	12
2.3. Eksperimento vykdymo eiga	12
2.3.1. Konfiguracija	12
REZULTATAI IR IŠVADOS	13
SANTRUMPOS	14
PRIEDAI	14
1 priedas. IPSS matavimų koordinatės 2014–2018 metais	15
2 priedas. Pradinių IPSS CSV failų apdorojimo „Python“ skriptas	20
3 priedas. Atrinktų IPSS CSV failų duomenų papildymo „Python“ skriptas	22
4 priedas. Eksperimentinio palyginimo rezultatai	27

Įvadas

Įvade nurodomas darbo tikslas ir uždaviniai, kuriais bus įgyvendinamas tikslas, aprašomas temos aktualumas, apibrėžiamas tiriamasis objektas akcentuojant neapibrėžtumą, kuris bus išspręstas darbe, aptariamoms teorinėms darbo prielaidoms bei metodikai, apibūdinami su tema susiję literatūros ar kitokie šaltiniai, temos analizės tvarka, darbo atlikimo aplinkybės, pateikiama žinių apie naudojamus instrumentus (programas ir kt., jei darbe yra eksperimentinė dalis). Darbo įvadas neturi būti dėstyimo santrauka. Įvado apimtis 2–4 puslapiai.

Problematika

TODO

1. Medžiagos darbo tema dėstymo skyriai

Medžiagos darbo tema dėstymo skyriuose išsamiai pateikiamos nagrinėjamos temos detalės: pradiniai duomenys, jų analizės ir apdorojimo metodai, sprendimų įgyvendinimas, gautų rezultatų apibendrinimas.

Medžiaga turi būti dėstoma aiškiai, pateikiant argumentus. Tekste dėstomas trečiuoju asmeniu, t.y. rašoma ne „aš manau“, bet „autorius mano“, „autoriaus nuomone“. Reikėtų vengti informacijos nesuteikiančių frazių, pvz., „...kaip jau buvo minėta...“, „...kaip visiems žinoma...“ ir pan., vengti grožinės literatūros ar publicistinio stiliaus, gausių metaforų ar panašių meninės išraiškos priemonių.

Skyriai gali turėti poskyrius ir smulkesnes sudėtines dalis, kaip punktus ir papunkčius.

1.1. Poskyris

Citavimo pavyzdžiai: cituojamas vienas šaltinis [**PvzStraipsnLt**]; cituojami keli šaltiniai [**PvzStraipsnEn; PvzKonfLt; PvzKonfEn; PvzKnygLt; PvzKnygEn; PvzElPubLt; PvzElPubEn; PvzMagistrLt; PvzPhdEn**].

2. Duomenų analizės eksperimentas naudojant „MacroBase SQL“ modulį

2.1. Duomenų rinkinys

Šiame skyriuje aprašyti eksperimentui pritaikyti IPSS duomenų rinkiniai.

2.1.1. Eksperimentui pritaikyti duomenys

Eksperimentui buvo naudojami Lietuvos Respublikos ryšių reguliavimo tarnybos administruojamos Interneto prieigos stebėsenos sistemos (IPSS) atliekami belaidės interneto prieigos duomenų perdavimo spartos kontrolinių matavimų 2014–2018 metų rezultatai. Matavimai atlikti operatorių UAB „Bitė Lietuva“, AB „Telia Lietuva“, UAB „TELE2“ ir AB Lietuvos radijo ir televizijos centro (LRTC) judriojo ryšio tinkluose visoje Lietuvos teritorijoje. Matavimų paskirtis yra stebėti ir įvertinti teikiamų interneto prieigos paslaugų kokybę, kaip to reikalauja Europos Sąjungos direktyvos, taip pat supažindinti visuomenę su tokių matavimų rezultatais. Įrašai pateikiami CSV formato failais. Iš viso gauti 3 failai skirti trimis skirtingoms technologijoms: 3G, LTRE ir WiMAX. Viena CSV failu priklausomai nuo failui priskirtos technologijos buvo apytiksliai 136000, 157000 ir 18000 įrašų eilučių, vidutiniškai 104000. Vieno failo dydis vidutiniškai 8,5MB.

2.1.1.1. Duomenų aprašymas

Kiekvienas IPSS įrašų failas prasideda antraštės (angl. *header*) eilute, kuri susideda iš failo saugomų įrašų sąrašo laukų (angl. *fields*) pavadinimų. IPSS įrašų sąrašų laukai ir jų paaiškinimas:

- **Bendri visiems failams laukai:**

Data ir laikas – matavimo įrašo fiksavimo data ir laikas, užrašoma „yyyy-MM-dd hh:mm:ss“ formatu.

Platuma – matavimo įrašo fiksavimo koordinatų platuma užrašoma dešimtainiais laipsniais (angl. *decimal degrees*).

Ilguma – matavimo įrašo fiksavimo koordinatų ilguma užrašoma dešimtainiais laipsniais (angl. *decimal degrees*).

Operatorius – matavimo įrašo operatoriaus skaitinis kodas arba pavadinimas (vieno iš jau minėtų operatorių: „Bitė Lietuva“, „Telia Lietuva“, „TELE2“ ir LRTC).

RSSI – signalo stiprumo indikatorius (angl. *received signal strength indicator*).

Sparta kbit/s – sveikuoju skaičiumi nurodomas užfiksuotas matavimo greitis kilobitais per sekundę.

- **3G failui išskirtiniai laukai:**

Celės id – matavimą atlikusio įrenginio identifikacijos kodas.

Ryšio tinklas – nurodo matuojamo ryšio tinklą (visi įrašai šiame faile ryšio tinklo lauke turi reikšmę „3G“).

3G ryšio technologija – nurodo matavimo metu naudotą ryšio technologiją (vieną iš: „DC-HSPA+“, „HSDPA“, „HSPA“, „HSPA+“, „HSUPA“ arba „WCDMA“).

- **LTE failui išskirtiniai laukai:**

Celės id – matavimą atlikusio įrenginio identifikacijos kodas.

RSRP – angl. *reference signals received power* panašiai kaip RSSI signalo stiprumo identifikatorius LTE tinklams.

RSRQ – angl. *reference signals received quality* signalo kokybės laukas (šiam dokumente šis laukas reikšmės neturi – „NULL“).

SINR – signalo ir trukdžių bei triukšmo santykis (angl. *signal-to-interference-plus-noise ratio*) (šiam dokumente šis laukas reikšmės neturi – „NULL“).

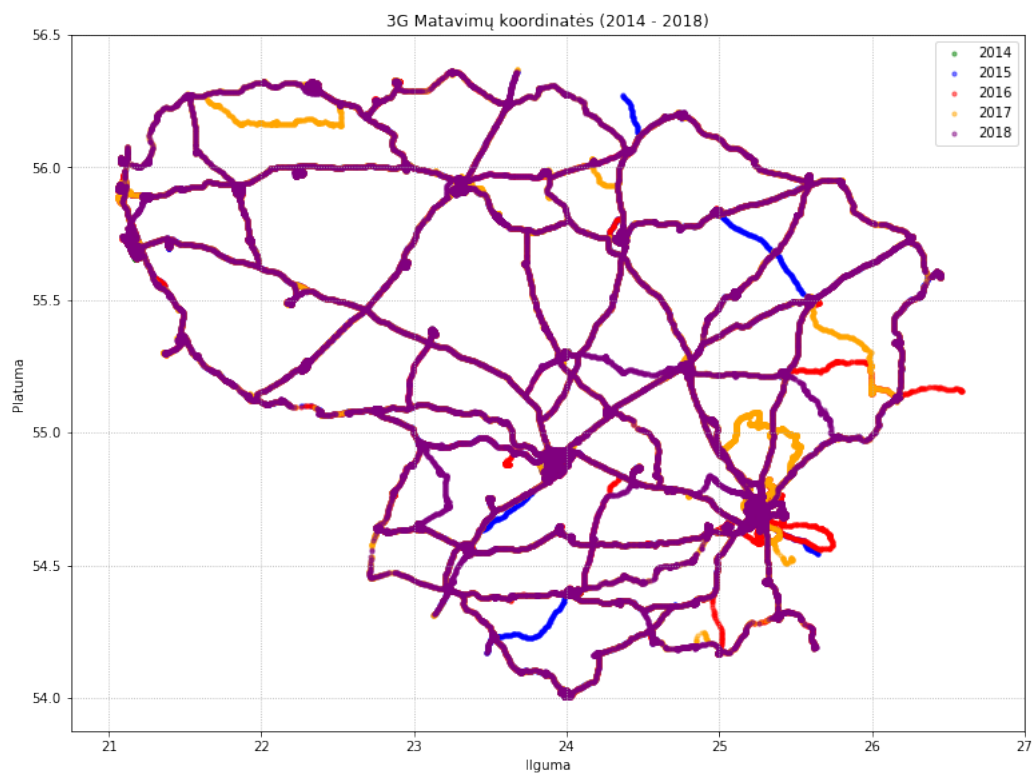
Ryšio tinklas – nurodo matuojamo ryšio tinklą (visi įrašai šiame faile ryšio tinklo lauke turi reikšmę „LTE“).

- **WiMAX failui išskirtiniai laukai:**

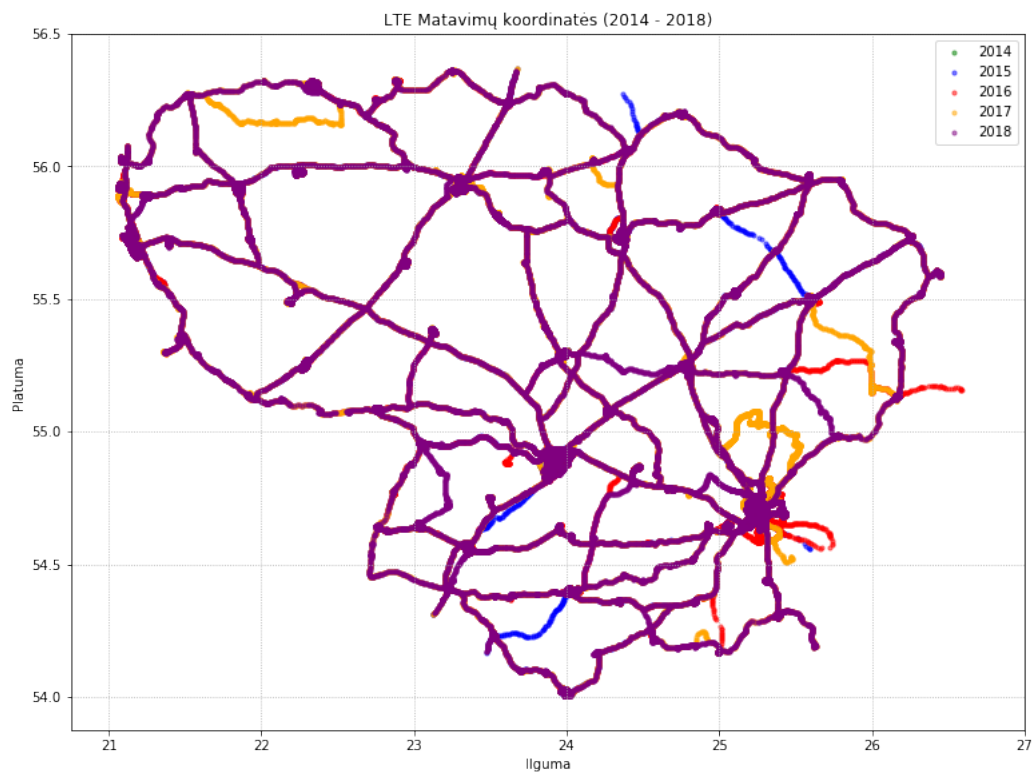
Bazinės stoties id – matavimą atlikusios stotelės MAC adresas.

CINR – trikdžių ir triukšmo santykis (angl. *carrier to interference-plus-noise ratio*).

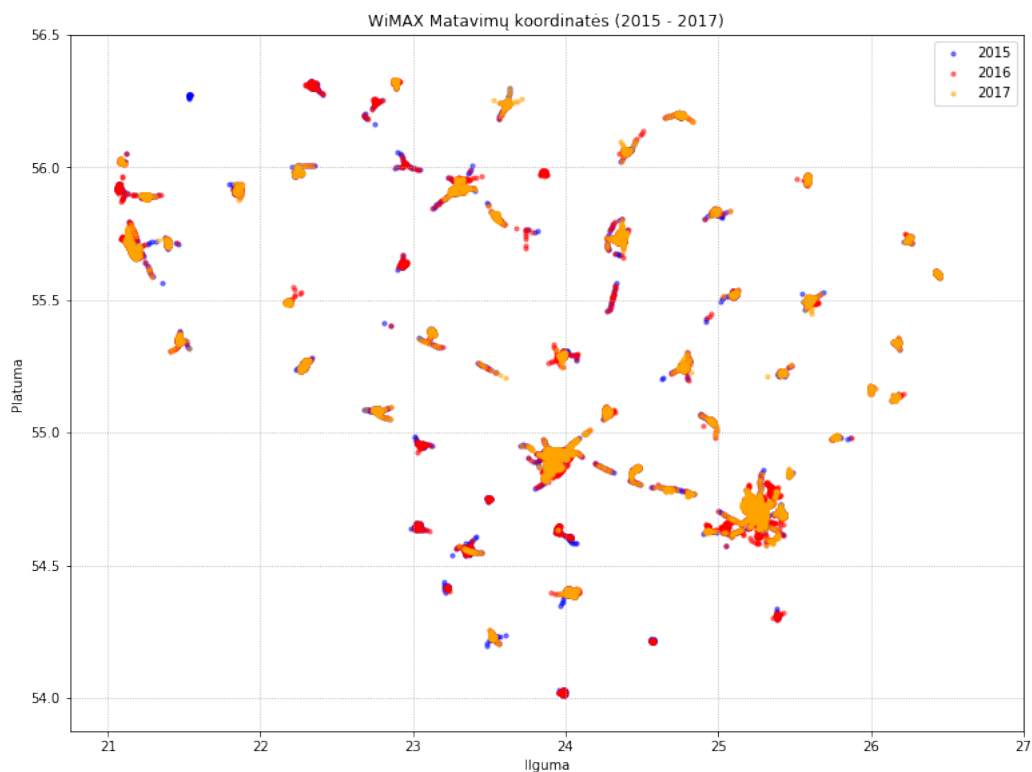
Ryšio technologija – nurodo matuojamo ryšio tinklą (visi įrašai šiame faile ryšio technologijos lauke turi reikšmę „WiMAX“).



1 pav. 3G Matavimai 2014–2018 metais.



2 pav. LTE Matavimai 2014–2018 metais.



3 pav. WiMAX Matavimai 2015–2017 metais.

Aiškiau išskirtus įrašų koordinacių duomenis galima rasti Priede nr. 1.

2.1.1.2. Duomenų paruošimas

Eksperimentui atlikti buvo atrinkti laukai, kurie turi reikšmes visose įrašų sąrašo eilutėse bei taip pat buvo atmesti pasikartojančios informacijos laukai (ryšio tinklo ir ryšio technologijos laukai, kurie nurodo, koks tinklo tipas iš trijų tiriamųjų yra naudojamas faile). Taip pat buvo nuspręsta datos ir laiko lauką pakeisti tik metų lauku. Taigi, 3G tinklui atrinkti šie pradiniai laukai: metai, ilguma, platuma, operatorius, celės id, rssi, 3G ryšio technologija ir sparta; iš LTE duomenų atrinkti laukai: metai, ilguma, platuma, operatorius, celės id, rssi, rsrp, sparta; iš WiMAX duomenų atrinkti laukai: metai, ilguma, platuma, operatorius, bazinės stoties id, rssi, cirn, sparta.

Eksperimentui atlikti buvo reikalingas CSV formato failai, kuriuose nebūtų lietuviškų raidžių ir būtų naudojami vieno žodžio arba vienos simbolių eilutės antraštės pavadinimai (pavyzdžiui, tarpai gali būti pakeičiami „_“ simboliu), todėl buvo realizuoti „Python“ skriptas, kuris nuskaito duotųjų duomenų CSV failus ir juos apdorojus, grąžina naujus CSV failus. Skriptas kurtas ir leistas „Jupyter Notebook“ aplinkoje. Skripte galima pasirinkti kelią iki duotųjų CSV failų ir kelią iki naujai sukuriamų CSV failų. Šis skriptas skirtas pervadinti antraštes, atskirti metus ir atrinkti nereikalingus laukus. Skriptai atlieka šiuos veiksmus 3 kartus kiekvienai ryšio

technologijai (3G, LTE, WiMAX) (Priedas nr. 2):

1. Nuskaito CSV įrašų failą pateiktą nurodytu keliu.
2. Sukuria naują CSV failą nurodytų keliu ir pavadinimu.
3. Įrašo į naujai sukurtą CSV failą, naują antraštę, su naujais angliškais vieno žodžio (simbolių eilutės) pavadinimais.
4. Vykdo ciklą, kuris skaitant po vieną eilutę iš pradinio failo įrašo tik reikalingus duomenis į naują failą bei iš datos ir laiko paima tik metus. Taip pat kiekviena eilutė yra skaičiuojama.
5. Išvedamas galutinis eilučių skaičius.

Atrinkus duomenis iš pradinių duomenų buvo sukurtas antras skriptas skirtas išgauti daugiau informacijos iš koordinacių. Kadangi ilgumos ir platumos laukai yra realiosios reikšmės, jomis yra nepatogu naudotis. Tam buvo sukurtas dar vienas skriptas, kuris suapvalina koordinacių laukus iki pasirinkto skaičiaus po kableliu taip sumažinant unikalių reikšmių kiekį. Šios reikšmės buvo pridėtos į joms sukurtus naujus laukus. Papildomai originalios koordinacių reikšmės buvo panaudotos išgauti kiekvieno koordinacių taško aukštį virš jūros lygio. Ši bei suapvalinta iki dešimčių metrų reikšmės buvo pridėtos į papildomus joms sukurtus laukus. Taigi naujas „Python“ skriptas nuskaito prieš tai sukurtus CSV failus, gauna aukščio virš jūros lygio reikšmes, suapvalina koordinacių bei aukščio reikšmes ir sukuria naujus CSV failus pasirinktu adresu. Skriptai atlieka šiuos veiksmus 3 kartus kiekvienai ryšio technologijai (3G, LTE, WiMAX) (Priedas nr. 3):

1. Nuskaito CSV įrašų failą pateiktą nurodytu keliu.
2. Iš nuskaitytų duomenų atskiriama antraštė nuo įrašų.
3. Iš antraštės surandami metų, platumos ir ilgumos laukų indeksai.
4. Pasinaudojus gautais indeksais, sukuriama visų ilgumos ir platumos reikšmių masyvai.
5. Vykdomas ciklas, kuris visus įrašus padalina į pasirinkto žingsnio dydžio dalis.
6. Vykdomas ciklas cikle, kuris visas ilgumos ir platumos reikšmes sujungia į „string“ tipo kintamuosius, kuriuose ilguma ir platumą yra atskirtos kableliu „“.
7. Kiekvienam žingsniui, iš platumos ir ilgumos bendrų kintamųjų sukuriamas vienas „string“ kintamasis, kur kiekvienas bendras koordinacių kintamasis yra atskirtas vertikaliu brūkšniu „|“.
8. Vykdomas ciklas, kuris kiekvienai žingsnio koordinacių kiekio „string“ kintamajam sukuria URL užklausą „Jawg.io“ API, kuri susideda iš koordinacių kintamojo, pagrindinio URL ir vartotojo prieigos žymės (angl. *access token*).
9. Kiekvienam žingsniui yra siunčiama HTTP užklausa, kuri grąžina JSON duomenų struktūrą su koordinatėmis ir jų aukščiais virš jūros lygio.
10. Iš JSON duomenų struktūros atskiriamos aukščio virš jūros lygio reikšmės ir jos yra patal-

pinamos į masyvą atitinkamu indeksu.

11. Vykdomas ciklas, kuris kiekvienam nuskaitytam įrašui sukuria naują įrašą su suapvalintomis iki pasirinkto skaičiaus po kablelio reikšmėmis bei aukščio virš jūros lygio ir suapvalinta iki dešimčių aukščio virš jūros lygio reikšmėmis. Šie įrašai sujungiami su atitinkamais įrašais ir sukuriamas naujas masyvas su šiais įrašais.
12. Sukuriamas naujas CSV failas nurodytu keliu ir pavadinimu.
13. Įrašoma nauja antraštė su papildytais pavadinimais.
14. Vykdomas ciklas, kuris kiekvieną naujai papildytą įrašą įveda į naują CSV failą.

Eksperimentui atlikti buvo nuskaityti ir apdoroti apie 300000 įrašų, vidutiniškai 100000 per failą.

TODO: pridėti citatą

2.2. Eksperimento aplinkos paruošimas

2.3. Eksperimento vykdymo eiga

2.3.1. Konfiguracija

Rezultatai ir išvados

Rezultatų ir išvadų dalyje išdėstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas įdiegta), toliau pateikiamos išvados (daromi nagrinėtų problemų sprendimo metodų palyginimai, siūlomos rekomendacijos, akcentuojamos naujovės). Rezultatai ir išvados pateikiami sunumeruotų (gali būti hierarchiniai) sąrašų pavidalu. Darbo rezultatai turi atitikti darbo tikslą.

Santrumpos

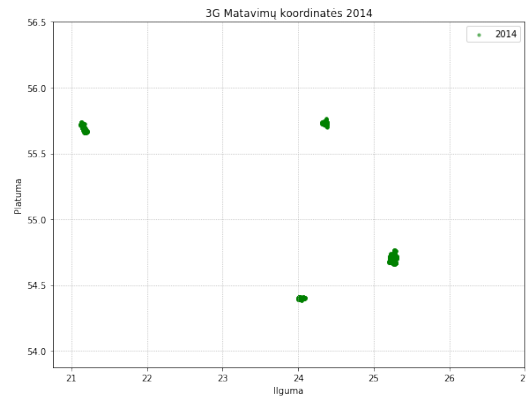
- CSV
- 3G
- LTE
- MAC adresas
- IPSS
- URL
- API
- HTTP
- JSON

Sąvokų apibrėžimai ir santrumpų sąrašas sudaromas tada, kai darbo tekste vartojami specialūs paaiškinimo reikalaujantys terminai ir rečiau sutinkamos santrumpos.

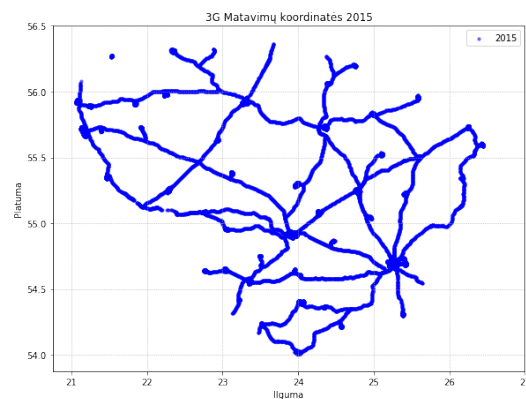
Priedas nr. 1

IPSS matavimų koordinatės 2014–2018 metais

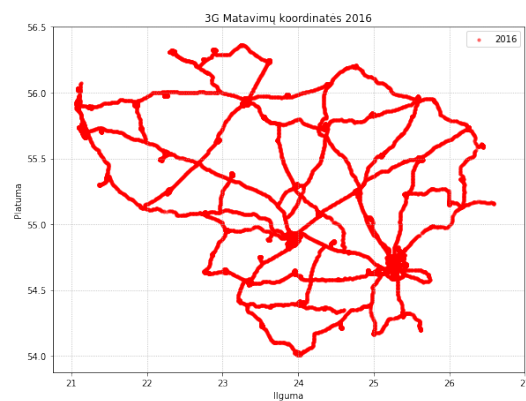
IPSS 3G ryšio technologijos matavimai 2014–2018 metais. Viso 136003 įrašai.



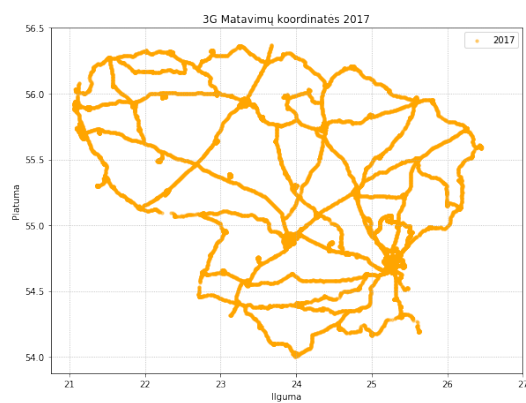
4 pav. 3G Matavimai 2014 metais. Viso – 2245 matavimai.



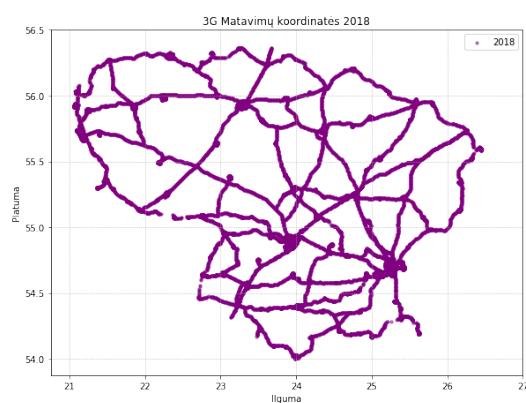
5 pav. 3G Matavimai 2015 metais. Viso – 19427 matavimai.



6 pav. 3G Matavimai 2016 metais. Viso – 34769 matavimai.

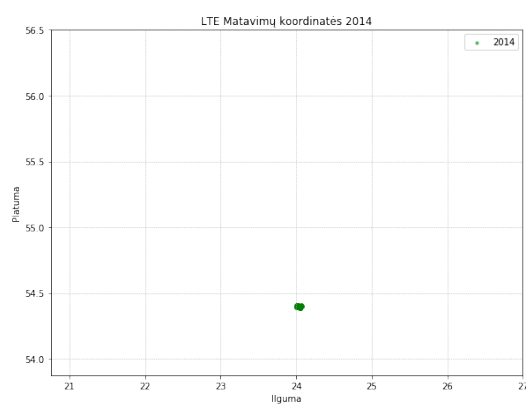


7 pav. 3G Matavimai 2017 metais. Viso – 38789 matavimai.

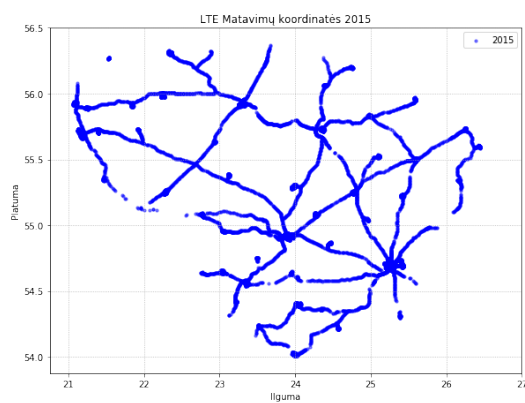


8 pav. 3G Matavimai 2018 metais. Viso – 40773 matavimai.

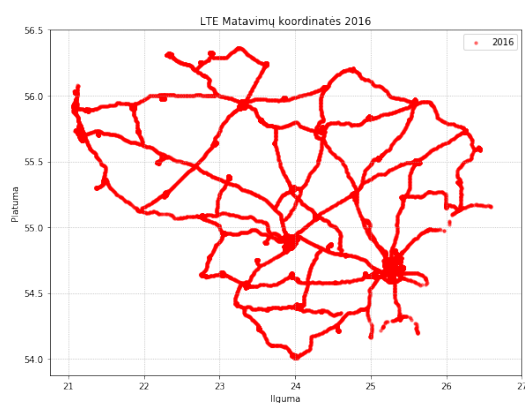
IPSS LTE ryšio technologijos matavimai 2014–2018 metais. Viso 157522 įrašai.



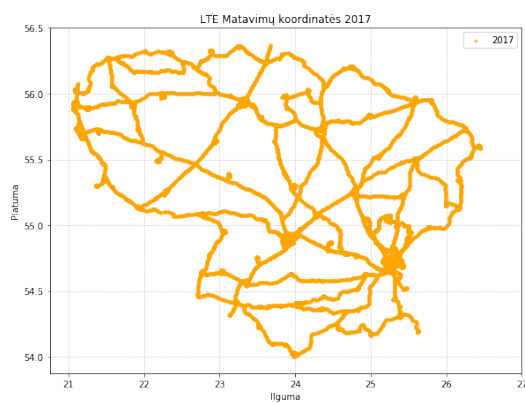
9 pav. LTE Matavimai 2014 metais. Viso – 232 matavimai.



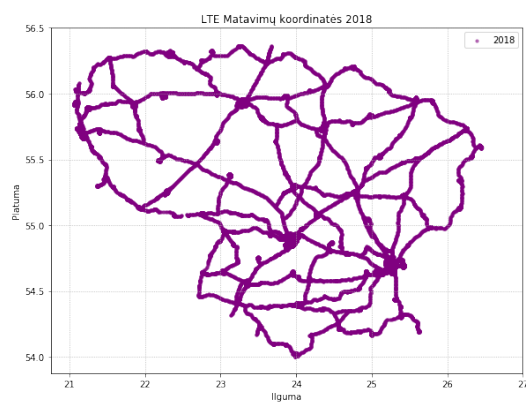
10 pav. LTE Matavimai 2015 metais. Viso – 13962 matavimai.



11 pav. LTE Matavimai 2016 metais. Viso – 40738 matavimai.

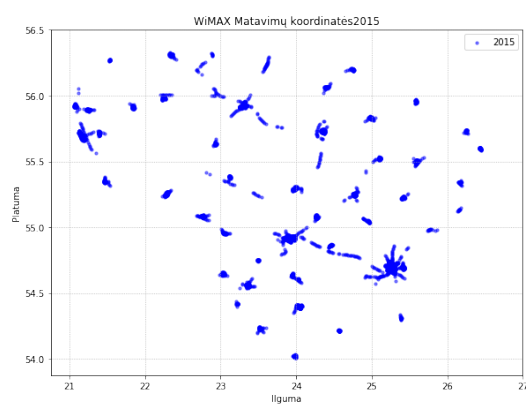


12 pav. LTE Matavimai 2017 metais. Viso – 49672 matavimai.

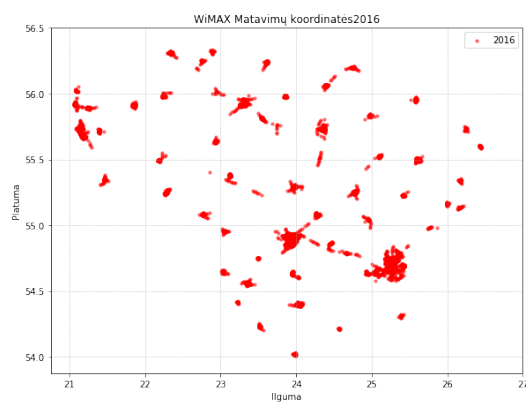


13 pav. LTE Matavimai 2018 metais. Viso – 52918 matavimai.

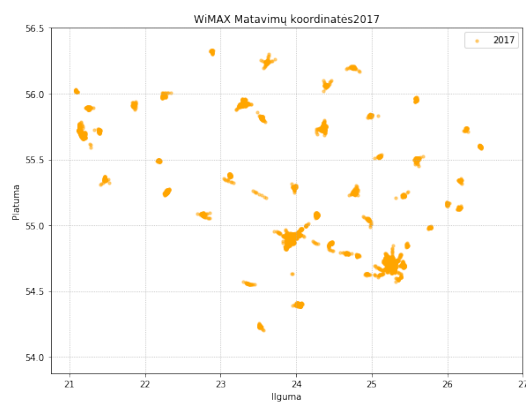
IPSS WiMAX ryšio technologijos matavimai 2014–2018 metais. Viso 18166 įrašai.



14 pav. WiMAX Matavimai 2015 metais. Viso – 4307 matavimai.



15 pav. WiMAX Matavimai 2016 metais. Viso – 7653 matavimai.



16 pav. WiMAX Matavimai 2017 metais. Viso – 6206 matavimai.

Priedas nr. 2

Pradinių IPSS CSV failų apdorojimo „Python“ skriptas

„Python“ skriptas skirtas iš duotųjų IPSS duomenų CSV failų atrinkti nereikalingus laukus ir pervadinti lietuviškus antraštės pavadinimus į vieno žodžio (vienos simbolių eilutės) angliškus pavadinimus bei sugeneruoti naujus CSV failus:

```
import csv

file_3G_csv = 'Data/Raw/Matavimai-3G.csv'
file_LTE_csv = 'Data/Raw/Matavimai-LTE.csv'
file_WiMAX_csv = 'Data/Raw/Matavimai-WiMAX.csv'
new_file_3G_csv = 'Data/Processed/measurements-3G.csv'
new_file_LTE_csv = 'Data/Processed/measurements-LTE.csv'
new_file_WiMAX_csv = 'Data/Processed/measurements-WiMAX.csv'

with open(new_file_3G_csv, mode='w') as new_3G_csv_file:
    writer_3G = csv.writer(new_3G_csv_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
    writer_3G.writerow(['year', 'latitude', 'longitude', 'provider', 'cell_id', 'speed', 'technology', 'rsssi'])
    with open(file_3G_csv, mode='r') as csv_file:
        csv_reader = csv.DictReader(csv_file)
        line_count = 0
        for row in csv_reader:
            year = row['data ir laikas'][:4]
            latitude = row['platuma']
            longitude = row['ilguma']
            provider = row['operatorius']
            cell_id = row['ėcels id']
            rsssi = row['rsssi']
            technology = row['3G šryio technologija']
            speed = row['sparta kbit/s']
            writer_3G.writerow([year, latitude, longitude, provider, cell_id, speed, technology, rsssi])
            line_count += 1
        print(f'Processed {line_count} 3G CSV file lines.')

with open(new_file_LTE_csv, mode='w') as new_LTE_csv_file:
    writer_LTE = csv.writer(new_LTE_csv_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
    writer_LTE.writerow(['year', 'latitude', 'longitude', 'provider', 'cell_id', 'speed', 'technology', 'rsssi'])
    with open(file_LTE_csv, mode='r') as csv_file:
        csv_reader = csv.DictReader(csv_file)
        line_count = 0
        for row in csv_reader:
```

```

        year = row['data ir laikas'][:4]
        latitude = row['platuma']
        longitude = row['ilguma']
        provider = row['operatorius']
        cell_id = row['ècels id']
        rssi = row['rssi']
        rsrp = row['rsrp']
        speed = row['sparta kbit/s']
        writer_LTE.writerow([year, latitude, longitude, provider, cell_id, rssi, rsrp, speed])
        line_count += 1

    print(f'Processed {line_count} LTE CSV file lines.')

with open(new_file_WiMAX_csv, mode='w') as new_WiMAX_csv_file:
    writer_WiMAX = csv.writer(new_WiMAX_csv_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
    writer_WiMAX.writerow(['year', 'latitude', 'longitude', 'base_station_id', 'rssi', 'rsrp', 'speed'])
    with open(file_WiMAX_csv, mode='r') as csv_file:
        csv_reader = csv.DictReader(csv_file)
        line_count = 0
        for row in csv_reader:
            year = row['data ir laikas'][:4]
            latitude = row['platuma']
            longitude = row['ilguma']
            base_station_id = row['èbazins stoties id']
            rssi = row['rssi']
            cinr = row['cinr']
            speed = row['sparta kbit/s']
            writer_WiMAX.writerow([year, latitude, longitude, base_station_id, rssi, cinr, speed])
            line_count += 1

    print(f'Processed {line_count} WiMAX CSV file lines.')

```

Priedas nr. 3

Atrinktų IPSS CSV failų duomenų papildymo „Python“ skriptas

„Python“ skriptas skirtas jau atrinktiems CSV failams pridėti papildomus laukus: aukščio virš jūros lygio (gaunamas iš koordinacių), suapvalintų koordinacių reikšmių bei suapvalintos aukščio reikšmės. Šiems duomenims yra sugeneruojamas nauji CSV failai:

```
import csv
import numpy as np
import pandas as pd
import urllib.parse as up
import urllib.request as ur
import json

new_file_3G_csv = 'Data/Processed/measurements-3G.csv'
new_file_LTE_csv = 'Data/Processed/measurements-LTE.csv'
new_file_WiMAX_csv = 'Data/Processed/measurements-WiMAX.csv'
more_data_file_3G_csv = 'Data/Processed/more-measurements-3G.csv'
more_data_file_LTE_csv = 'Data/Processed/more-measurements-LTE.csv'
more_data_file_WiMAX_csv = 'Data/Processed/more-measurements-WiMAX.csv'
token = "XPIYEbsLssXaMRcPolV9txdQyrZ0LrOTA5Hn9bq6SW7jvnb92rsKQZbx2augAEpL"
main_url = "https://api.jawg.io/elevations?"
request_step = 100
round_digits = 1
new_X_3G = []
new_X_LTE = []
new_X_WiMAX = []

read_3G = pd.read_csv(new_file_3G_csv)
X_3G = np.array(read_3G)
y_3G = np.array(read_3G.columns)
lat_3G = np.where(y_3G == 'latitude')[0][0]
long_3G = np.where(y_3G == 'longitude')[0][0]
lats_3G = X_3G[0:, lat_3G]
longs_3G = X_3G[0:, long_3G]
prev_r = 0
len_3G = len(X_3G)
locations_3G = []
for r in range(request_step, len_3G + request_step, request_step):
    locs_3G = []
    for i in range(prev_r, r):
        if i >= len_3G:
```



```

len_LTE = len(X_LTE)
locations_LTE = []
for r in range(request_step, len_LTE + request_step, request_step):
    locs_LTE = []
    for i in range(prev_r, r):
        if i >= len_LTE:
            break
        locs_LTE.append(str(lats_LTE[i]) + ',' + str(longs_LTE[i]))
    prev_r = r
    locations_LTE.append("|".join(locs_LTE))
elevation_LTE = np.array([])
for loc in locations_LTE:
    params = {'locations': loc, 'access-token': token}
    full_url = main_url + up.urlencode(params)
    request = ur.urlopen(full_url)
    contents = request.read()
    request.close()
    jdata = json.loads(contents)
    el = np.array([j['elevation'] for j in jdata])
    elevation_LTE = np.concatenate((elevation_LTE, el))
new_X_LTE = []
for i in range(len_LTE):
    item = X_LTE[i]
    new_item = []
    new_item.append(round(item[lat_LTE], round_digits))
    new_item.append(round(item[long_LTE], round_digits))
    new_item.append(elevation_LTE[i])
    new_item.append(round(elevation_LTE[i] / 10, 0) * 10)
    complete_item = np.append(np.array(item), np.array(new_item))
    new_X_LTE.append(complete_item)
new_X_LTE = np.array(new_X_LTE)
with open(more_data_file_LTE_csv, mode='w') as new_LTE_csv_file:
    writer_LTE = csv.writer(new_LTE_csv_file, delimiter=',', quotechar='\"', quoting=csv.QUOTE_MINIMAL)
    writer_LTE.writerow(['year', 'latitude', 'longitude', 'altitude', 'lat_1', 'long_1'])
    for item in new_X_LTE:
        writer_LTE.writerow([item[0], item[1], item[2], item[10], item[8], item[9], item[3], item[4], item[5], item[6], item[7]])

read_WiMAX = pd.read_csv(new_file_WiMAX_csv)
X_WiMAX = np.array(read_WiMAX)
y_WiMAX = np.array(read_WiMAX.columns)

```



```

lat_WiMAX = np.where(y_WiMAX == 'latitude')[0][0]
long_WiMAX = np.where(y_WiMAX == 'longitude')[0][0]
lats_WiMAX = X_WiMAX[0:, lat_WiMAX]
longs_WiMAX = X_WiMAX[0:, long_WiMAX]
prev_r = 0
len_WiMAX = len(X_WiMAX)
locations_WiMAX = []
for r in range(request_step, len_WiMAX + request_step, request_step):
    locs_WiMAX = []
    for i in range(prev_r, r):
        if i >= len_WiMAX:
            break
        locs_WiMAX.append(str(lats_WiMAX[i]) + ',' + str(longs_WiMAX[i]))
    prev_r = r
    locations_WiMAX.append("|".join(locs_WiMAX))
elevation_WiMAX = np.array([])
for loc in locations_WiMAX:
    params = {'locations': loc, 'access-token': token}
    full_url = main_url + up.urlencode(params)
    request = ur.urlopen(full_url)
    contents = request.read()
    request.close()
    jdata = json.loads(contents)
    el = np.array([j['elevation'] for j in jdata])
    elevation_WiMAX = np.concatenate((elevation_WiMAX, el))
new_X_WiMAX = []
for i in range(len_WiMAX):
    item = X_WiMAX[i]
    new_item = []
    new_item.append(round(item[lat_WiMAX], round_digits))
    new_item.append(round(item[long_WiMAX], round_digits))
    new_item.append(elevation_WiMAX[i])
    new_item.append(round(elevation_WiMAX[i] / 10, 0) * 10)
    complete_item = np.append(np.array(item), np.array(new_item))
    new_X_WiMAX.append(complete_item)
new_X_WiMAX = np.array(new_X_WiMAX)
with open(more_data_file_WiMAX_csv, mode='w') as new_WiMAX_csv_file:
    writer_WiMAX = csv.writer(new_WiMAX_csv_file, delimiter=',', quotechar='\"', quoting=csv.QUOTE_MINIMAL)
    writer_WiMAX.writerow(['year', 'latitude', 'longitude', 'altitude', 'lat_1', 'long_1'])
    for item in new_X_WiMAX:

```

```
writer_WiMAX.writerow([item[0], item[1], item[2], item[9], item[7], item[8], item[3], item[4], item[5], item[6], item[10], item[11], item[12], item[13], item[14], item[15], item[16], item[17], item[18], item[19], item[20], item[21], item[22], item[23], item[24], item[25], item[26], item[27], item[28], item[29], item[30], item[31], item[32], item[33], item[34], item[35], item[36], item[37], item[38], item[39], item[40], item[41], item[42], item[43], item[44], item[45], item[46], item[47], item[48], item[49], item[50], item[51], item[52], item[53], item[54], item[55], item[56], item[57], item[58], item[59], item[60], item[61], item[62], item[63], item[64], item[65], item[66], item[67], item[68], item[69], item[70], item[71], item[72], item[73], item[74], item[75], item[76], item[77], item[78], item[79], item[80], item[81], item[82], item[83], item[84], item[85], item[86], item[87], item[88], item[89], item[90], item[91], item[92], item[93], item[94], item[95], item[96], item[97], item[98], item[99]])
```

Priedas nr. 4

Eksperimentinio palyginimo rezultatai

1 lentelė. Lentelės pavyzdys

Algoritmas	\bar{x}	σ^2
Algoritmas A	1.6335	0.5584
Algoritmas B	1.7395	0.5647