

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Aleksandr Aleksandrov 185396IAIB
Maria Pazeikova 185713IAIB
Anastassia Lobatšjova 185736IAIB

**KAUGSEIRE VAATLUSE ANDMETE
PÕHINEVA STATISTILISE ANALÜÜSI
ALGORITMIDE RAKENDAMINE JA
VISUALISEERIMINE ANOMAALIAATE
TUVASTAMISEKS**

Bakalaureusetöö

Juhendaja: Evelin Halling, PhD
Martin Simon, MSc

Tallinn 2021

Autorideklaratsioon

Kinnitame, et oleme koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Aleksandr Aleksandrov, Maria Pazeikova, Anastassia Lobatšjova

18.05.2021

Annotatsioon

Arendatud lahenduse eesmärk oli luua prorotüüp, millega klient analüüsib, katsetab lahendusi, nõ "sandbox" seireandmete peale tehtava objekti ja muutuste tuvastuse ning sellega tehtava aegrea analüüsiks. Lõputöö raames oli vaja rakendada statistilise analüüsi algoritmid, võrrelda nende töötamine ja efektiivsuse *mock-up* anmestikute peal ja otsustada, milline nendest algoritmidest visualiseerida veebipõhise paaneli peal.

Lõputöös on kirjeldatud terve töö protsess, kasutatavad tehnoloogiad, abivahendid algoritmide rakendamiseks ja visualiseerimiseks, anomaalse liikumise tähendus ning on välja toodud erinevate algoritmide tulemuste võrdlemine.

Töö tulemuseks on veebipõhine paneel, kus terve analüüsi protsess ja tulemus on hästi jälgitav. Paneeli peal on võimalik vaadata kõik huvipakkuva objektid maailmakaardil, ning vaadata nende liikumise trajektoore. Lisaks sellele on võimalik vaadata iga kindla objekti informatsiooni kasutades graafiku ja kaardi, näiteks liikumise kiirus kindlal ajahetkel, kas liikumine oli anomaalne või mitte ja kindla objekte erinevaid trajektoore. Veebipõhine paneel on testitud kaasjuhendaja abil.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 7 peatükki, 14 joonist, 1 tabel.

Abstract

Application and Visualization of Statistical Analysis Algorithms Based on Remote Sensing Observation Data for Anomalies Detection.

The aim of developed solution was to create a prototype with which the customer analyzes and tests the solutions, so-called “sandbox” monitoring data for object and time series analysis. In the dissertation, it was necessary to apply the algorithms of statistical analysis, compare their operation and efficiency on mock-up data sets, and decide which of these algorithms to visualize on a web-based panel.

The thesis describes the whole work process, the technologies used and the tools for implementing and visualizing the algorithms and compares the results of different algorithms. The meaning of anomalous movement is described in the work.

The result of the work is a web-based panel, where the entire analysis process and result can be well monitored. On the panel it is possible to view all the objects of interest on the world map, and view their trajectories. In addition, it is possible to view the information of each specific object using a graph and a map, for example, the speed of movement at a particular point in time, whether the movement was anomalous or not, and different objects in different trajectories. The web-based panel has been tested with the help of a co-tutor.

The thesis is in Estonian and contains 27 pages of text, 7 chapters, 14 figures, 1 table.

Lühendite ja mõistete sõnastik

Backend	Veebisaidi sisemise osa rakendamine
Frontend	Kliendipoolse kasutajaliides
Timestamp	Kuupäev ajaga
Mock-up	Andmed, mis on väga sarnased tulevikus kasutatavate andmetega
AI	Närvivõrgu moodul
DBSCAN	Density-Based Spatial Clustering of Applications with Noise – andmete klastramise algoritm
FFT	Fast Fourier Transform – kiirendatud diskreetne Fourieri teisendusalgoritm
Unit testid	Tarkvara testimise tüüp, kus testitakse tarkvara üksusi või komponente.

Sisukord

1 Sissejuhatus.....	9
1.1 Ülesande püstitus	9
2 Tehnoloogiate valik	11
2.1 Andmete analüüsi tehnoloogiad.....	11
2.1.1 Flask.....	11
2.1.2 Sklearn	11
2.2.3 Pandas	11
2.2.4 Numpy.....	11
2.2 Kasutajaliidese tehnoloogiad	12
2.2.1 React	12
2.2.2 Chart.js.....	12
2.2.3 React Leaflet	12
2.3 Docker.....	12
2.4 PostgreSQL.....	13
3 Projekti kirjeldus.....	14
4 Valideerimine.....	15
5 Statistiline analüüs	16
5.1 Aegrida.....	16
5.2 Anomalia.....	16
5.3 Töö mock-up andmetega.....	16
5.3.1 Andmeanalüüsi eelprotsess	17
5.3.2 Anomaaliate otsingu võimalused	17
5.3.3 Anomaalia tuvastamiseks algoritmide rakendamine	17
5.3.4 Kotkaste liikumise andmestiku tulemused.....	22

5.3.5 Busside liikumiste andmestiku tulemused	25
5.4 Objekti liikumise suuna määramine.....	26
5.4.1 Asimuut väärtus	27
5.4.2 Suuna määramine.....	27
5.5 Töö dünaamilise andmetega	27
6 Kasutajaliides	28
6.1 Prototüüp.....	28
6.2 Praegune UI paneel	28
6.2.1 Peamine koduleht.....	28
6.2.2 Graafiku ja analüüsi vaated.....	29
6.2.3 Tabeli vaade ja raporti komponent	29
7 Kokkuvõte.....	31
Edasine töö.....	32
Kasutatud allikad	33
Lisa 1 – “Funkstioon kauguse arvutamiseks kahe GPS punktide vahel”	35
Lisa 2 – “Funktsioon asimuut väärtuse arvutamise kahe GPS punktide vahel”	36
Lisa 3 – Kasutajaliidese prototüüp.....	37
Lisa 4 – Kasutajaliidese peamine leht.....	38
Lisa 5 – Kasutajaliidese graafiku ja analüüsi vaade	40
Lisa 6 – Kasutajaliidese tabeli ja raporti component.....	42
Lisa 7 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	43

Jooniste loetelu

Joonis 1. Arhitektuuri diagramm	14
Joonis 2. Aegrida	16
Joonis 3. DBSCAN, eps=0.3, min_samples=10	18
Joonis 4. FFT	19
Joonis 5. Vasakul isoleetitud xi, paremal isoleeritud x0.....	20
Joonis 6. Anomaaliate otsing manuaalselt	22
Joonis 7. DBSCAN'i tulemus, eps=0.3, min_samples=10	23
Joonis 8. Isoleeritud metsa tulemus. Contamination=0.001	23
Joonis 9. Isoleeritud metsa tulemus. Contamination=0.1	24
Joonis 10. Anomaaliate tuvastamine standardhälve meetodiga.....	24
Joonis 11. FFT algoritmi tulemus	25
Joonis 12. DBSCAN'i tulemus, eps=0.3, min_samples=10	25
Joonis 13. Isoleeritud metsa tulemus, contamination=0.003	26
Joonis 14. Anomaaliate tuvastamine standardhälve meetodiga.....	26

1 Sissejuhatus

Tänapäeval andmeanalüüs on väga võimekas tööriist. Iga aasta andmemaht suureneb ja analüüsides neid andmeid võib leida väärtusliku informatsiooni.

Üks kõige tähtsamatest ülesandest andmeanalüüsis on anomaaliate tuvastamine. Anomaaliate otsimiseks nimetatakse probleemiks, mis on suunatud objektide otsimisele, mille käitumine ei ole sarnane ülejäänud andmetega. Anomaaliate tuvastamise praktiliste ülesannete näited on: finantspettus, anomaaliad meditsiini näitudes (EKG uuring) ning mootoririke tuvastus. Lähtudes loetletud valdkondadest, kus kasutatakse anomaaliate otsimist ja tuvastamist, ei saa selle ülesande olulisust eitada. Vaadeldava ülesande olulisuse määrab asjaolu, et anomaalia annab olulist teavet süsteemi seisundi kohta, see võib olla signaal tegutsemiseks, näiteks täiendava kontrolli läbiviimiseks.

Anomaaliate otsing ja tuvastamine on keeruline ülesanne, kuna kõige tihedamini andmestikes andmed pole sildistatud ehk eelnevalt puudub lisainformatsioon mis andmed oli fikseeritud nagu anomaalsed, aga milliseid andmeid peetakse normaalseks. Seepärast anomaaliate tuvastamine vaadatakse nagu järelevalveta õppimine. Järelevalveta õppimine tähendab seda, et algoritm võtab andmed ja peab ise leidma millise tunnuse järgi andmed erinevad üksteisest. Samuti tekib raskus algoritmi töö kontrollimisega.

Selles töös anomaaliate tuvastamine on tehtud eesmärgiga, et saada ülevaade objektide liikumisest. Ülevaade all mõistetakse, et soovitakse teada iga objekti trajektoori, liikumise suunda, tihedamaid punkte ning keskmist kiirust. Dokumendis on kirjeldatud tehtud analüüsi tulemused, kus analüüsiti kaks erinevat andmestikku, mis kirjeldavad objektide liikumist. Lõputöö raames oli proovitud rakendada masinõpe algoritme ning statistilist meetodit andmeanalüüsis.

Töö jooksul oli rakendatud rakendustarkvara, mis koosneb UI panelist ning backendist.

1.1 Ülesande püstitus

Lõputöö eesmärk on rakendada rakendustarkvara, mis võimaldab anda mugava ja arusaadava ülevaade objektide liikumisest vähendades maksimaalselt inimteguri. Idee paikneb selles, et kõik vajalik töö ja analüüs teeks arvuti.

Kuna lõputöö lõpptoode kasutamise eesmärk on salastatud, meeskonnale ei olnud teada millise andmetega oleks vaja töötada.

Seepärast oli otsustatud kasutada niinimetatud *mock-up* andmeid, mis on väga sarnased nendega, mida kavatsetakse kasutada lõpuks.

Lõputöö raames seati järgmised eesmärgid ja nõuded:

1. Tutvumine mock-up andmekogumitega
2. Andmete asukohtade visualiseerimine kasutajaliidese peal
3. Andmete statistiline analüüs: kõige sagedasemad asukohad
4. Kõige sagedasemate asukohtade visualiseerimine
5. Objekti trajektoori visualiseerimine
6. Liikumise anomaaliade aegridade analüüs
7. Objekti suunde määramamine GPS punktide vahel
8. Närvivõrgu integreerimine
9. Anomaalia esitusviisi integreerimine kaardil ja graafikuna
10. Töö testimine ja dokumenteerimine

2 Tehnoloogiate valik

Mugava arendamise protsessi jaoks oli otsustatud töö jagada kaheks osaks. Back-endi osa on realiseeritud kasutades Python programmeerimis keelt ja front-endi osas kasutatakse React'i raamistik, mis samal ajal on realiseeritud kasutades JavaScript programmeerimis keelt. All toodud tehnoloogiad olid valitud kuna kõigil meeskonnaliikmetel oli nende kasutamise kogemus.

2.1 Andmete analüüsi tehnoloogiad

Andmeanalüüsi osas oli kasutatud Python keel ning seoses sellega abiks olid juba tutavad teed, mida kasutatakse andmeteaduse valdkonnas.

2.1.1 Flask

Flask on lõputöö põhiline back-endi raamistik, mis võimaldab genereerida JSON objekte kasutades Pythonis oleva andmetüübi ja edastada olulist informatsiooni UI paneelile. Selleks, et laiendada Flaski back-end serverit eduka informatsiooni analüüsimiseks kasutatakse erinevaid Pythoni raamistikud. See võimaldab rakendada anüüsi ja edastada olulist informatsiooni UI paneelile mugavam ja kiiremini [1].

2.1.2 Sklearn

Sklearn on masinõppe teek Python keele jaoks. Sklearn võimaldab realiseerida regressiooni, klassifatsiooni ja muu masinõppe ülesanded. Samuti teek võimaldab teha andmete eeltöötlust - andmestikute jagamine treenimis-ja testimiandmestikuteks, standardiseerime.

2.2.3 Pandas

Pandas on teek, mis võimaldab efektiivselt töötada andmestikega. Teegiga saab filtreerida, otsida, eemaldada veergusid, kirjutada ja lugeda faile.

2.2.4 Numpy

Numpy on teek, mis töötab ndarray massividega. Teek võimaldab rakendada põhjalikke matemaatilisi funktsioone, Fourier'i teisendusi, loogiliseid operatsioone ja muud algoritme.

Lõputöös Numpy aitab FFT algoritmi ja statistilise meetodi rakendamisel. Samuti tuli kasuks Numpy sissefunktsioonid, mis arvutavad välja aritmeetilist keskmist.

2.2 Kasutajaliidese tehnoloogiad

2.2.1 React

React on lõputöö põhiline front-endi raamistik, kus toimub analüüsi tulemuse töötlemine ja visualiseerimine graafiku ja kaartide peal. Reactis kasutatakse sisseehitatud JavaScripti meetodeid selleks, et kuvata kindla komponendi kindlal ajal. Arendamise protsess on lihtsustatud tänu informatsiooni edastamisele komponentide vahel. On võimalik edastada olulist informatsiooni praegusest komponendist selle alama komponentidele. Lisaks sellele Reactis on väga lihtne raamistikute integreerimine [2].

2.2.2 Chart.js

Chart.js on Reacti raamistik graafikute loomiseks. Raamistiku kasutades on võimalik luua üheksa erinevat graafikut erinevate sise parameetritega [3]. Lõputöö raames kasutatakse kolm graafiku varianti - "Line chart", "Bar chart" ja "Bubble chart". Esimene kasutatakse objekti kiiruse visualiseerimiseks kindlal ajal, teine anomaaliade visualiseerimiseks ja kolmas kõige tihedam objekti valitava suuna visualiseerimiseks.

2.2.3 React Leaflet

Kuna kaart on UI paneeli peamine tööriist, oli vaja leida mugav ja lihtsasti implementeeriv raamistik, mis samal ajal oleks võimalik ka kasutada Reacti peal. Oli valitud React Leaflet. Selline raamistik aitab implementeerida Leaflet kaardid Reacti rakenduse peal. Lisaks sellele raamistik võimaldab komponentide muutmist. Näiteks raamistiku abil saab kaardi peale lisada oma jooned, kaardi kihid jne [4].

2.3 Docker

Selleks et vähendada korduvaid, igapäevaseid seadistamise ülesanded ja teha kogu arendustsükli jooksul rakenduse kiireks ja lihtsaks oli otsustatud kasutada Dockerit [5]. Kuna lõputöö rakendust on vaja edastada ka kliendile testimiseks siis seda on lihtsam teha Dockerit image'na, kuna kliendile ei ole vaja projekti konfigureerida või seadistada, vaid on võimalik kasutada ainuke fail, kus toimub terve rakenduse genereerimine.

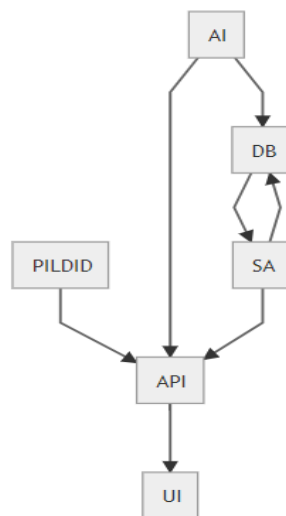
2.4 PostgreSQL

PostgreSQL on relatiivne andmebaas, mis laiendab SQL keelt koos paljude funktsioonidega, mis turvaliselt salvestavad ja suurendavad kõige keerukamate andmete töökoormusi [6]. Antud lõputöö raames PostgreSQL on kasutatud logini süsteemi jaoks, kasutajate salvestamiseks.

3 Projekti kirjeldus

Rakendustarkvara koosneb mitmetest osast. Back-end, kus toimub peamine andmete analüüs, UI paneli algoritmide tulemuse visualiseerimiseks, AI moodul ning andmebaas analüüsi tulemuse ja informatsiooni hoidmiseks. Igal osal on oma ülesanded ja eesmärgid.

AI moodul kujundab endast närvivõrgu, mis tuvastab objekte. AI mooduli eesmärk tuvastada objekte ning arvutada pildilt objekti koordine. Backendis toimub statistiline andmeanalüüs ning UI paneeliga suhtlus ja UI panelil toimub saadud tulemuseid visualiseerimine.



Joonis 1. Arhitektuuri diagramm

Andmeanalüüs annab informatsioon nii iga objekti kohta, kui ka terve ülevaade objektidest. Kui vaadeldakse ühte objekti, siis esialgselt tahetakse saada objekti trajektoori, mis koosneb GPS punktidest (laius ja pikkus) ning soovitatakse jälgida kiiruseid, millega oli läbitud kaugust iga kahe GPS punktide vahel. Samuti on plaanis, et ülevaades oleks võimalik näha liikumise suunda GPS punktide vahel ning statistiliseid näiteid nagu keskmine kiirus ja kõige tihedam liikumise suund terve marsruudi jooksul. Kõikide objektide puhul tehakse vaadelda kõike tihedamaid punkte, mis oli külastatud objektidega ning väljendada nende trajektoore. Analüüsis peab olema ka rakendatud algoritm, mis saab efektiivselt leida anomaaliaid objekti kiiruse põhjal. Teades anomaalseid kiiruseid on soov eristada UI panelil kaks trajektoori: esimene on objekti terve marsruut ja teine on marsruut, kus algoritm pidi kiirust anomaalseks nende GPS punktide vahel.

Järgnevalt on kirjeldatud iga osa eesmärgid ja nende implementeerimine.

4 Valideerimine

Lõputöö käigus oli pidev suhtlus kliendiga. Klient avaldas oma soove kasutajaliidese kohta ja tema ootus analüüsi tulemusest.

Nagu enne oli mainitud lõputöö tõelised eesmärgid on salastatud ning ei olnud teada meeskonnale. Kuna klient soovis rakendust oma andmestikuga testida, ei olnud ka viisi, kuidas saab sellel juhul Unit teste kirjutada. Seepärast tööprotsessis peamine rõhk oli kliendi esindaja manuaalsele testimisele ja suhtlemisele. Iga nädal oli korraldatud koosolek, kus kliendi esindaja avaldas oma arvamust ja soove UI paneeli kohta ja analüüsist.

Lõputöös ei olnud saavutatud kõike eesmärke. Lõputöö raames ei olnud tehtud AI mooduli ühendamine. Vaatamata sellele teised eesmärgid oli saavutatud. Järgnevalt on toodud kliendi esindaja sõnad tehtud töö eest: „Töö on tehtud väga hästi. Tudengid olid väga aktiivsed esimestest kohtumistest peale ja hästimotiveeritud.

See tagas olukorra, kus kliendi poolt tõstatatud küsimused ja funktsionaalsused lahendati üpris kiiresti ja vajalikul tasemel. Loodud süsteem ületas mõnevõrra kliendi ootuseid. UI paneeli prototüüp on tänaseks kliendil aktiivselt kasutuses ja kliendi andmeanalüüsi meeskond on testimas oma back-end lahendusi selle abil.“

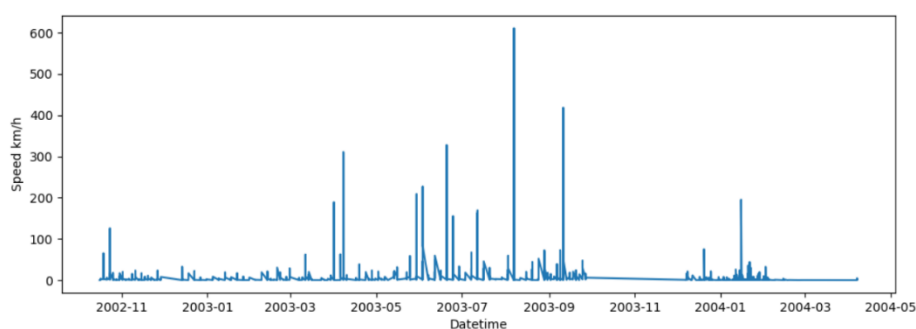
5 Statistiline analüüs

Statistilise analüüsi moodul on pühendatud andmete analüüsile, mis võtab endasse anomaaliatuvastamine, objektide liikumise suunda määramine kahe GPS punkti vahel ning leida objekti keskmist kiirust. Statilise analüüsi modulis toimuvad kõik arvutused ja valmistatakse andmeid edasi saatmiseks UI modulise väljastamiseks.

Statistilise analüüsi moodul on vaheetapp AI ja UI moodulite vahel

5.1 Aegrida

Statistikas nimetatakse aegridadeks andmeid, mis on järjestikuliselt mõõdetud teatud ajaintervallide kaupa. Ajaintervallid on sageli võrdsed (aasta, tund, päev), kuid see ei pruugi alati nii olema. Uuritavast objektist tuleb dünaamiline ettekujutus, kus x teljel on aeg, aga y teljel on uuritava objekti tunnus [7]. Lõputöö raames aegreas uuritakse objekti kiirust.



Joonis 2. Aegrida

5.2 Anomalia

Anomaaliad on andmemustrid, mis ei vasta täpselt määratletud normaalse käitumise kontseptsioonile. Anomaaliatuvastamine on väga tähtis etapp andmeanalüüsis. Anomaaliatuvastamine ning andmeanalüüs saab anda väärtuslikku informatsiooni paljudes valdkondades nagu turvalisus, arstiabi jne.

5.3 Töö mock-up andmetega

Enne AI mooduli ühendamist töötleti *mock-up* andmeid, mis olid võetud avalikestest allikast. Selline juurdeminek projekti tööle andis head arusaamist mida ja kuidas on vaja analüüsida ning mis viisil rakendada algoritme.

Töötati kahe andmestikuga, mis kirjeldasid erinevate objektide liikumist. Esimene andmestik koosnes kotkaste liikumisest, aga teine kujundas endast busside liikumine. Need andmestikud on erinevad, aga veerud, millele peamine rõhk oli tehtud analüüsis, langevad kokku.

5.3.1 Andmeanalüüsi eelprotsess

Anomaalia leidmisel orienteeruti kiiruse järgi. Sellist veergu ei olnud esialgsetes andmestikes. Kiiruse leidmiseks oli kasutatud $v = S/t$ valem, kus S - läbitud kaugus kahe GPS punktide vahel ja t - selle kauguse läbimisele kulutatud aeg.

Kulutatud aeg oli välja arvatud antud timestamp vahe leidmisel ning kuna objektide liikumise andmed oli fikseeritud Maal, siis kauguse leidmiseks oli vaja arvestada sellega ja kasutada erivalemit allikast [8]. (Lisa 1)

Ühikud:

- Aeg on välja arvatud tundides
- Kiirus on välja arvatud km/t

5.3.2 Anomaaliade otsingu võimalused

Andmehõive analüüs oli jagatud kolmeks osaks:

- Esimesena oli proovitud leida anomaaliat lähtudes kaineist mõistusest (kotkaste andmestiku varal oli leitud, et kotkas ei saa lennata kiiremini, kui 320 km/h. Busside andmestiku juhul oli vaja arvestada ka sellega, kus bussid liiguvad (näiteks linnas või väljaspool linna), sest lubatud bussi kiirus erineb sõltuvalt sellest, kus asub bussi marsruut. Kui arvutamisel saadud kiirus oli suurem, kui piirkiirus, mida oli paika pandud, siis peeti seda anomaalseks liikumiseks. Näiteks, kui kotkas lendas mingis punktis kiiremini, kui 320 km/h, siis pidasime selle punkti anomaaliaks).
- Teisena oli proovitud rakendada masinõpe algoritme, mis suudaks tuvastada anomaaliad aegridades. Anomaaliade tuvastamiseks oli proovitud kasutada 3 algoritmi. Need on FFT (Fast Fourier transform), DBSCAN ja Isoleeritud Metsa algoritm.
- Kolmandal etapil oli kasutatud anomaaliade leidmiseks aegridades standarthälvet.

5.3.3 Anomaalia tuvastamiseks algoritmide rakendamine

Kuna üks peamiseks lõputöö eesmärgiks on vähendada inimteguri, siis on vaja leida efektiivset algoritmi, mis suudab automaatselt tuvastada anomaaliaid. Järgnevalt selles peatükis on detailsemalt kirjeldatud algoritmid, mis kasutati lõputöö käigus ja algoritmide töötamise tulemusi.

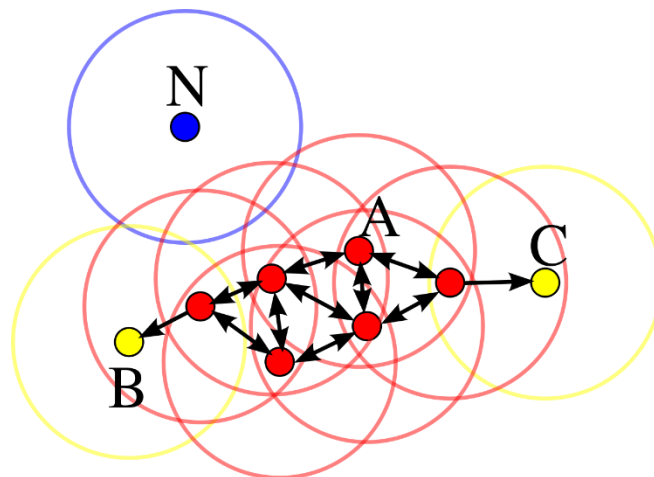
5.3.3.1 DBSCAN

DBSCAN - tiheduspõhine klasterdamisalgoritm, mida kasutatakse masinõppes

Algoritm jaotab andmepunktid etteantud parameetrite põhjal klastritesse. Algoritm paigutab piisavalt tihedalt paiknevad andmepunktid samasse klastrisse, samas kui kaugelt paiknevaid punkte loetakse müraks [9].

DBSCAN algoritm vajab kahte parameetrit:

- **eps**: määrab, kui lähedased punktid peaksid olema üksteisele, et neid saaks pidada klasteri osaks. See tähendab, et kui kahe punkti vaheline kaugus on selle väärtusega (eps) väiksem või võrdne, loetakse neid punkte naabriteks. [9]
- **min_samples**: tiheda piirkonna moodustamiseks vajame vähemalt min_samples punkti [9].



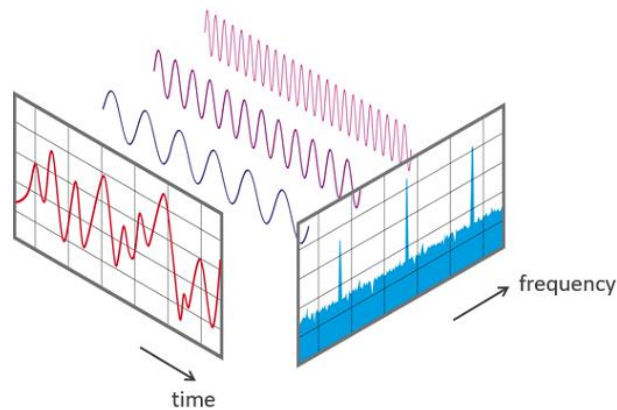
Joonis 3. DBSCAN, eps=0.3, min_samples=10

DBSCAN parameetreid ei saa häälestada, aga eksisteerivad reeglid, mille põhjal saab leida optimaalseid parameetreid.

- Kui andmestikus on rohkem kui 2 atribuuti, siis min_samples parameetrit saab valida valemi abil $\text{min_samples} = 2 * \text{atribuutide_arv}$ [10].
- Eps parameetri jaoks saab rakendada NearestNeighbors algoritm, mis võimaldab leida iga punkti jaoks kaugust lähima naabrini [11]. Saadud tulemuste põhjal joonistatakse graafik ja otsitakse eps parameetrit.

5.3.3.2 FFT

FFT - algoritm, mis on kõige populaarsem signaalide töötlemiseks [12]. FFT kasutatakse muusikatööstuses, inseneerias, andmeteaduses ja teistest valdkondades.



Joonis 4. FFT

Algoritmi põhiidee koosneb sellest, et iga mittelineaarne funktsioon võib kujutada siinuslainete summana. Fourier teisendus teisendab aja domeeni sageduse domeeniks. Samuti on olemas pöördteisendus, mis võimaldab teisendada sageduse domeenist aja domeenisse. Pärast algoritmi rakendamist on võimalik vaadata sagedusspektrit. Sagedusspekteril võib vaadata amplituudi või faasi, mis saab anda informatsiooni riketest ja anomaaliast [13].

5.3.3.3 Isoleeritud mets

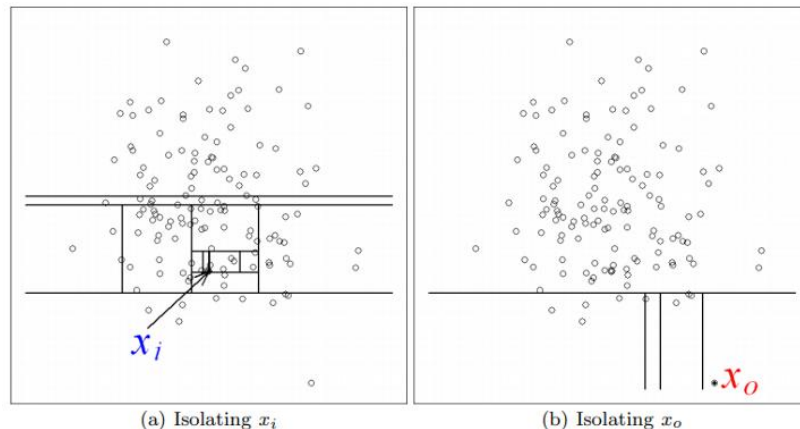
Isoleeritud Mets (Isolation Forest) on anomaaliade tuvastamise järelevalveta õppealgoritm, mis töötab anomaaliade isoleerimise põhimõttel, mitte tavalisemate tavapunktide profileerimise tehnikate asemel.

Isolation Forest 'isoleerib' vaatlused, valides juhuslikult mingi funktsiooni ja valides seejärel juhusliku jaotuse valitud tunnuse maksimaalse ja minimaalse väärtuse vahel. Kuna rekursiivset jaotamist saab kujutada puustruktuuriga, on valimi eraldamiseks vajalike jagunemiste arv võrdne tee pikkusega juur- ja lõpp-sõlme vahel. Juhuslik jaotamine annab anomaaliade jaoks märgatavalt lühemad teed. Seega, kui juhuslike puude mets toodab konkreetsete proovide jaoks ühiselt lühemat rada, on need suure tõenäosusega anomaaliad.

Isoleeritud Mets kasutab binaarseid puid punktirühmade eraldamiseks, kasutades juhuslikku tunnusruumi rakendavat juhuslikku künnist, kuni see jõuab lehtedeni, kus igal on üks punkt. Ja siis genereeritakse igale proovirühmale mitu puud, et saada puudemets, kus punkthinnang on metsa puude keskmiseks teepikkuseks mõõdetud.

Selle algoritmi tööpõhimõte on see, et andmekogumit jagatakse edasi, kuni iga punkt on üks, välised punktid jagunevad algoritmi täitmise varasemates etappides, samas kui tavalised

punktid nõuavad rohkem partitsioone. Iga osa genereeritakse juhuslikult, eraldi puude genereeritakse erinevate jaotiste komplektidega.



Joonis 5. Vasakul isoleeritud x_i , paremal isoleeritud x_0

Piltidelt on selgelt näha, et x_0 (anomaalia punkt) puruneb kiiremini kui x_i (tavapunkt).

Isoleeritud metsa põhiprintsiip on see, et heitkoguseid on vähe. Pärast puude loomist on iga punkti määramiseks kindel valem. See valem näeb välja selline:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

$h(x)$ tee pikkus

$E(h(x))$ on $h(x)$ keskmine isolatsioonipuude hulgast

$c(n)$ ebaõnnestunud binaarse puu pikkuse otsing

N väliste sõlmede arv

Isolatsioonimetsa iseloomustab vajadus väärtuste mõõtkavas mõõtmisruumi puudumise järele, see on tõhus meetod, kui väärtuste jaotust ei saa eeldada, sellel on väike arv parameetreid, mis muudab selle meetodi üsna usaldusväärseks ja lihtne optimeerida, sobib hästi andmete voogesituseks ja töötab andmekogumitega väga hästi [14].

Hüpperparameetrid isolatsiooni metsa rakendamiseks [15]:

- `n_estimators`: int, default=100
Baashinnangute arv ansamblis.
- `max_samples`: "auto", int või float, default="auto"
Iga põhihinnangu väljaõpetamiseks X -st võetud proovide arv.
- `contamination`: 'auto' või float, default='auto'
Andmekogumi saastatuse suurus, st kõrvalekallete osakaal andmekogumis.
Kasutatakse sobitamisel proovide skoori künnise määramiseks.

- `max_features`: int või float, default=1.0
Funktsioonide arv, mida saab tõmmata X-st iga baashinnangu trennimiseks.
- `bootstrap`: bool, default=False
Kui väärtus on True, sobivad üksikud puud asendusvalimiks valitud treeningandmete juhuslikesse alamhulkadesse. Kui väärtus on False, siis tehakse proov asenduseeta.
- `n_jobs`: int, default=None
Paralleelselt töötavate töökohtade arv nii sobivuse kui ka ennustamise jaoks.
- `random_state`: int, RandomState instance või None, default=None
Kontrollib tunnuse ja jagatud väärtuste valiku pseudo-juhuslikkust iga hargnemissammu ja iga selle metsa puu jaoks.
- `verbose`: int, default=0
Kontrollib puu ehitamise protsessi paljusust.
- `warm_start`: bool, default=False
Kui see on seatud väärtusele True, kasutatakse eelmise call-i lahendust, et see sobiks ja lisaks ansamblisse rohkem hinnanguid, vastasel juhul võetakse täiesti uut metsa.

Isolatsioonimetsa algoritmi parameetreid saab automaatselt häälestada kui andmed on sildistatud. Sellise töö jooksul sellist veergu ei olnud andmestikes. Seepärast põhineti reeglitele, mida oli leitud ava allikast [16]:

- `n_estimators`: 100 väärtus on kõige optimaalne enamiku juhtudel
- `contamination`: on vaja proovida leida katsete ja eksimuse viisil
- `max_features`: on vaja määrata parameetri arvu nii suur, kui see on iga olukorras.

5.3.3.4 Standardhälve

Standardhälve näitab väärtuste ümber hajuvust. Standardhälbe saab leida ruutjuurt dispersioonist arvutamisel. Kui standardhälve on piisavalt suur, siis saab pidada väärtust anomaaliaks, kuna see on keskmisest kaugel. Samuti ka vastupidi, kui standardhälve on väike, siis vastaja vastus on keskmisest lähedal [17].

Anomaaliate tuvastamine toimub võrdlemise teel. Kui väärtus (sellisel juhul kiirus) on suurem või võrdne kui summa kolm korda standardhälbest ning valimi keskmise väärtusest, siis peetakse sellist väärtust anomaaliaks [11].

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - m)^2}{n}}$$

S - standardhälve

x – iga tunnuse väärtus

m – valimi keskvärtus

n – valimi mahu suurus

Anomaaliate tuvastamine toimub võrdlemise teel. Kui väärtus (sellisel juhul kiirus) on suurem või võrdne kui summa kolm korda standardhälbest ning valimi keskmise väärtusest, siis peetakse sellist väärtust anomaaliaks [11].

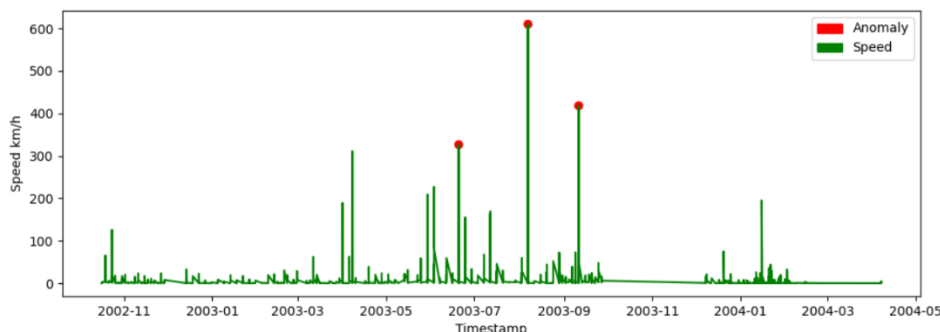
5.3.4 Kotkaste liikumise andmestiku tulemused

Kotkaste andmestikul on 23194 rida, 14 veergu ja 24 unikaalset id [17]. Järgnevalt on toodud algoritmide töötamise tulemused ja graafikud. Graafikutel on kotka (id=37305) kiirus ja leitud anomaaliaid. Roheline on kiirus ja punased punktid on anomaaliad. Järgnevalt on toodud analüüsi tulemused.

5.3.4.1 Anomaaliate otsing manuaalselt viisil

Nagu enne oli mainitud anomaaliate otsing oli tehtud kahe viisiga. Põhines Google artiklil [19], välja arvatud kiirus eelprotsessil oli võrreldud 320 arvuga ja juhul kui kiirus oli suurem kui 320, siis peeti sellist kiirus ja GPS punkte anomaaliaks.

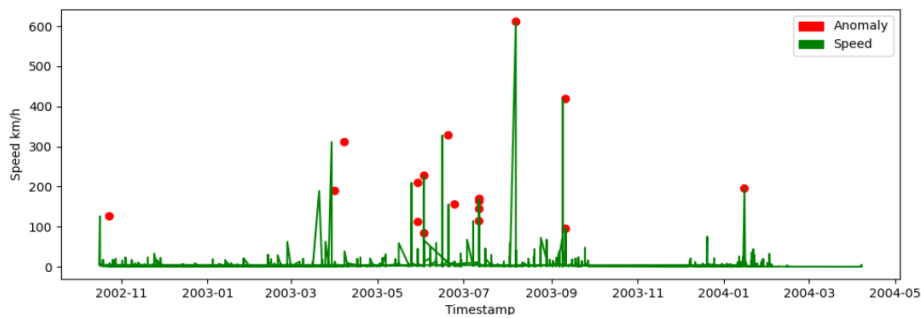
Kokkuvõtteks oli leitud 86 anomaalset punkti terve andmestiku jaoks.



Joonis 6. Anomaaliate otsing manuaalselt

5.3.4.2 DBSCAN

DBSCAN algoritmi rakendamisel oli saadud tulemuseks 480 anomaaliat terve andmestiku jaoks. Ühe kotka jaoks (id=37305) oli leitud 17 anomaaliat



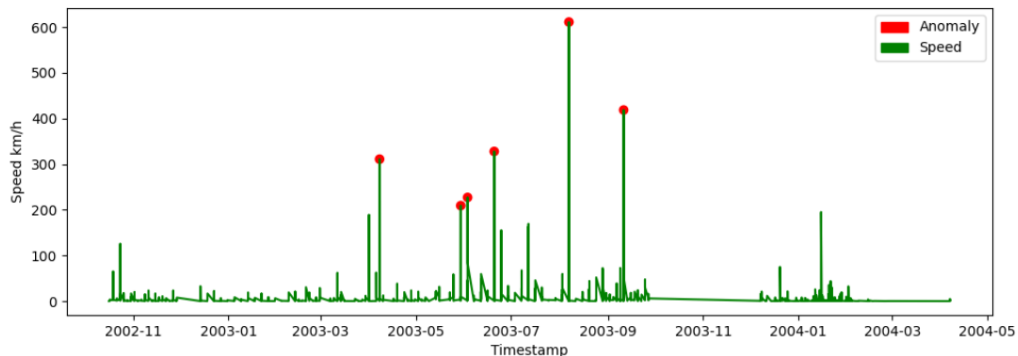
Joonis 7. DBSCAN'i tulemus, eps=0.3, min_samples=10

5.3.4.3 Isoleeritud mets

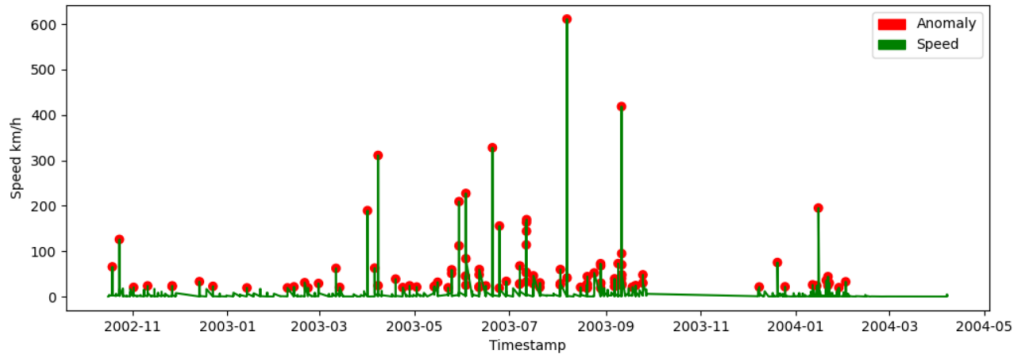
Isoleeritud metsa algoritmi rakendamisel oli saadud tulemuseks 3351 anomaaliat terve andmestiku jaoks, kui implementeerida ilma hüperparameetriteta.

Tööprotsessis oli raske reguleerida algoritmi contamination parameetrit. See on väga tundlik parameeter ning leida optimaalset on väga keeruline. Kui panna liiga suur, siis algoritm eksib väga palju ja peab anomaaliaks peaaegu kõigi kiiruseid

Sõltuvalt contamination parameetrist, oli leitud 6-103 anomaaliat ühe kotka jaoks (id=37305)



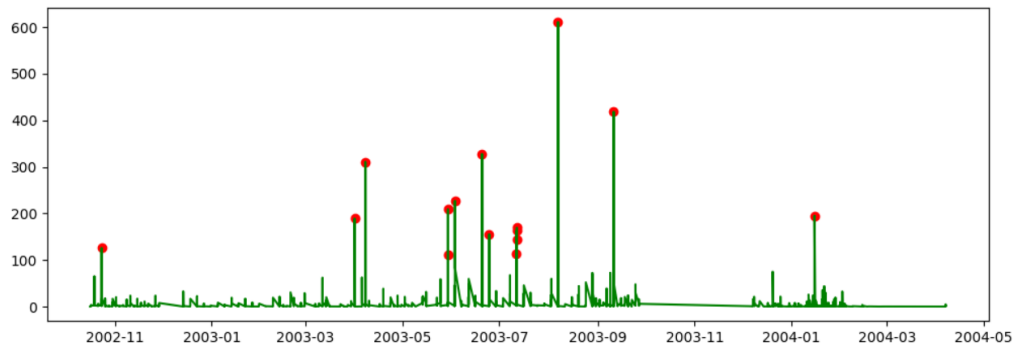
Joonis 8. Isoleeritud metsa tulemus. Contamination=0.001



Joonis 9. Isoleeritud metsa tulemus. Contamination=0.1

5.3.4.4 Standardhälve

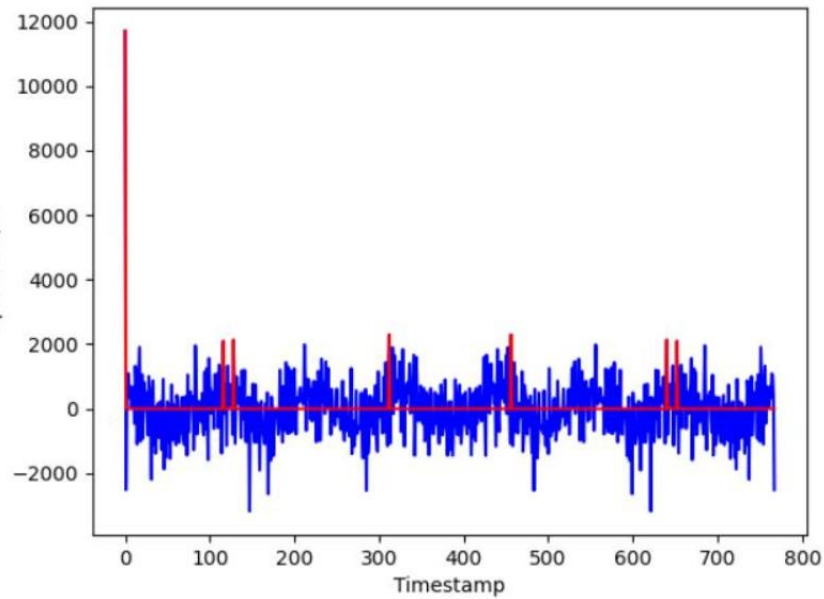
Kasutades standardhälvet oli saadud kokku 415 anomaaliat. Ühe kotka jaoks (id=37305) tulemus oli 15 anomaaliat.



Joonis 10. Anomaaliat tuvastamine standardhälve meetodiga

5.3.4.5 FFT

FFT rakendamisel toimub esialgsete väärtuste konverteerimine sageduseks ning filtreerimisel ja tagasi konverteerimisel see annab vale tulemusi (Näiteks graafikul on nähtav, et on olemas 7 anomaaliat ja tagasi konverteerimisel algoritm tagastab list 100+elementidest. Samuti esialgsed väärtused on nüüd kompleksarvud). Meeskonnas oli otsustatud loobuda FFT algoritmi kasutamisest. Graafikult on raske lugeda anomaaliaid.



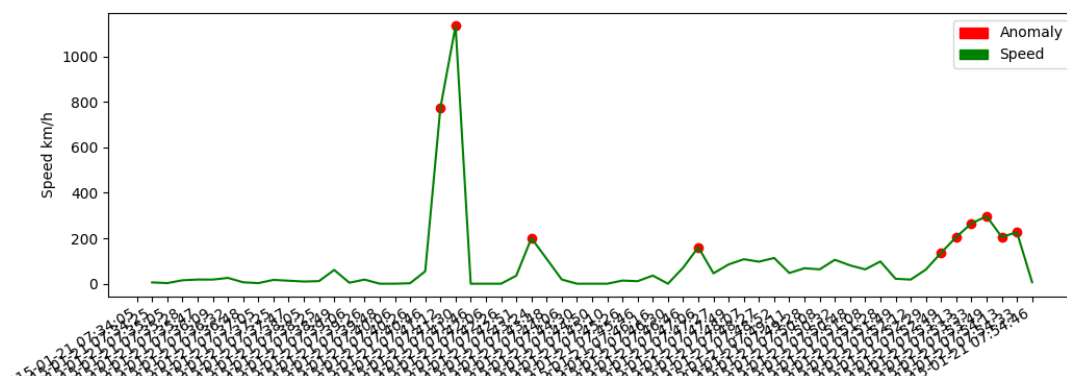
Joonis 11. FFT algoritmi tulemus

5.3.5 Busside liikumiste andmestiku tulemused

Busside andmestikul on 9 veergu ja 162 unikaalset id. Järgnevalt on toodud algoritmide töötamise tulemused ja graafikud. Graafikutel on bussi (id=58) kiirus ja leitud anomaaliaid. Roheline on kiirus ja punased punktid on anomaaliad.

5.3.5.1 DBSCAN

DBSCAN algoritmi rakendamisel oli saadud tulemuseks 32 anomaaliat terve andmestiku jaoks. Ühe bussi jaoks identifikaatori numbriga 58 oli leitud 10 anomaaliat järgmiste parameetritega:

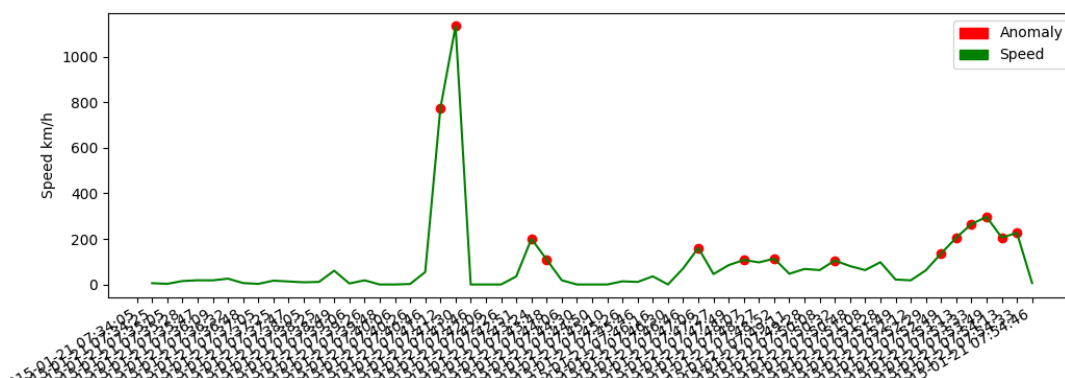


Joonis 12. DBSCAN'i tulemus, eps=0.3, min_samples=10

5.3.5.2 Isoleeritud mets

Isoleeritud metsa algoritmi rakendamisel oli saadud tulemuseks 55 anomaaliat terve andmestiku jaoks.

Ühe bussi jaoks identifikaatori numbriga 58 oli leitud 14 anomaaliat.

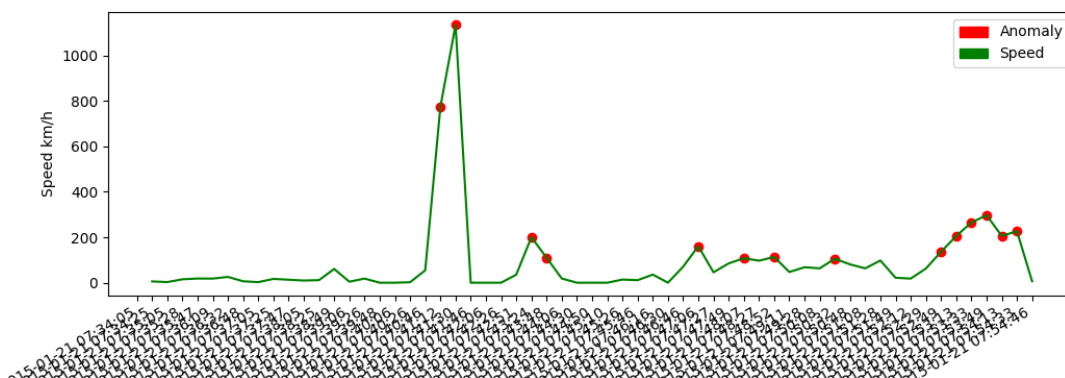


Joonis 13. Isoleeritud metsa tulemus, contamination=0.003

5.3.5.3 Standardhälve

Kasutades standardhälvet oli saadud kokku 55 anomaaliat.

Ühe bussi jaoks identifikaatori numbriga 58 tulemus oli 14 anomaaliat.



Joonis 14. Anomaaliat tuvastamine standardhälve meetodiga

5.4 Objekti liikumise suuna määramine

Ülevaade objektide liikumisest võib saada mitte ainult kiiruse ja anomaaliat alusel. Projekti käigus oli proovitud tuvastada objekti liikumise suunda iga kahe GPS punktide vahel. Suuna määramiseks arvutati Asimuut väärtust ja selle põhjal vaadeti, kuhu liikus objekt.

5.4.1 Asimuut väärtus

Asimuut on päripäeva nurk, mis on loodud 2 joonega. Esimene joon ühendab objekti praeguse positsiooni (punkt A) põhjasuunaga ning teine joon ühendab objekti praeguse positsiooni (punkt A) lõpliku punktiga (punkt B). Mõõdetakse 0-360 kraadini. Saadud kraadi väärtuse põhjal võib teha järeldust objekti liikumise suunast [20].

Asimuut väärtuse arvutamiseks kasutati skripti, mis võimaldas arvutada kraadi väärtust kahe GPS punkti vahel [21]. (Lisa 2)

5.4.2 Suuna määramine

Pärast asimuut väärtuste arvutamist saab määrata liikumist. Selle jaoks kasutati tabelit, mis oli loodud allika [22] alusel.

Tabel 1. Kompassi suunad asimuudi väärtuste põhjal

North	0	South	180
North-northeast	22.5	South-southwest	202.5
Northeast	45	Southwest	225
East-northeast	67.5	West-southwest	247.5
East	90	West	270
East-southeast	112.5	West-northwest	292.5
Southeast	135	Northwest	315
South-south	157.5	North-northwest	337.5

5.5 Töö dünaamilise andmetega

Antud etapi eesmärgiks oli ühendada statistilise analüüside mooduli AI mooduliga. AI moodul kujundab endast närvivõrgu, mis tuvastab objekte piltidel ja saadab JSON statistilise analüüsi moodulise. Lõputöö käigus ei jõudnud integreerida AI moodulit süsteemi.

6 Kasutajaliides

Analüüsi andmete visualiseerimiseks ja haldamiseks oli otsustatud arendada praktilise ja mugava veebipõhise paneeli.

6.1 Prototüüp

Enne kasutajaliidese arendamise protsessi algust oli kaalutud järele paneeli nii disaini osa kui ka peamine funktsionaalsus. Kõik komponendid pidi olema ilusti ja maksimaalselt loogiliselt kuvatud veebilehe peal. Selleks oli tehtud esmane paneli prototüüp võttes arvesse tuleva kasutuskogemust.

Prototüübil oli illustreeritud ainult üks vaade. Peamine koduleht, kus pidi toimuma terve analüüsi protsess. Sellel lehel suurem osa ruumist katab kaart, kus eeldatakse, et huvipakkuvad objektid ja nende liikumise tüübid ja trajektoorid on kuvatud selle kaardi peal. Kaardist üleval pool on statistiline informatsioon, näiteks kui palju objekte oli sellel ajahetkel eelmisel aastal või näiteks kui õige oli tehtud analüüüs objektide andmetega. Paremalt pool on erinevad graafikud, mis näitavad analüüsi tulemusi, näiteks kui imeline objektide liikumise tüüp on. Kodulehe allpool olid erinevad kastid inim ja ajaressurssi jaoks, näiteks praegused probleemid mis on otseselt seotud UI paneeliga, mitu inimest tegelevad selle projektiga. (Lisa 3)

6.2 Praegune UI paneel

UI paneeli arendamise protsessi jooksul selgus, et mõned komponendid olid vaja täiesti kustutada või asendada teiste rohkem mugavama komponentidega. Seepärast praegune UI paneel välja näeb teistmoodi ja tema funktsionaalsus erineb sellest, mis oli illustreeritud prototüübi peal.

Praegusel UI paneelil on neli komponenti - peamine kaardi vaade, graafikute ja analüüsi vaade, tabeli vaade ja raporti komponent. Kasutades UI paneeli oleva navigatsiooni riba üleval serval on võimalik nende komponentide vahel liikuda.

6.2.1 Peamine koduleht

Peamisel kodulehel terve ruum on kaetud kaardiga. Kuna kaart on selle kasutajaliidese peamine tööriist, siis seaded huvipakkuva objektidega, nende liikumise tüübi ja trajektooriumidega olid pandud otse kaardi peale. Kaardil vasakul poolel on abiline informatsiooni kast. Selle kasti

kasutades saab inimene rohkem informatsiooni objektide kogumi kohta. Näiteks millise objektiga tegu on (informatsiooni boxil on “Object of interest”), kui palju need objektid on praegu kuvatud kaardi peal (informatsiooni boxil on “Total count”), on võimalik objektid ja nende sageli külastatud kohta peida või kuvata kaardist kasutades nuppu “SHOW” klõpsamist. Kaardi paremal üleval nurgal on nupp kaardi kihtide vahetamiseks ja objektide trajektooride kuvamiseks või peitmiseks.

Kaardi peal objektid on kuvatud vastava ikooniga ja objektide sageli külastatud kohad on illustreeritud gps punktina. Lisaks sellele, kaardi peal on võimalik vaadata objektide erinevaid trajektoore - roheline värviga on objektide üldine trajektoor ja punase värviga on objektide imeline trajektoor (kui näiteks mingil ajahetkel objekti kiirus oli suurem kui tavaline või objekt muutis järsult liikumissuunda). Kaardi saab suuremaks või vähemaks muuta kasutades “+” ja “-” nupud kaardi üleval vasakul nurgal. (Lisa 4)

6.2.2 Graafiku ja analüüsi vaated

Selleks, et objektide analüüsi tulemused ilusti ja loogiliselt näidata UI paneeli peale, oli tehtud graafiku ja analüüsi vaade. See vaade on kõige oluline ja selle vaate peal toimub UI paneeli peamine funktsionaalsus. Vaade kasutades saab inimene kõige rohkem informatsiooni objekti kohta - objekti kiirus kindlal ajahetkel, imeline kiirus, üldine trajektoor, imeline trajektoor mis on arvutatud inimese abiga ja trajektoor mis statistiline analüüs arvutas imeliseks trajektoorigs. Vaate üleval poolel on aja ja kiiruse sõltuvuse graafik, kus roheline värviga on objekti üldine kiirus ajahetkel ja punased punktid on aeg, kus objekti kiirus oli suurem kui tavaline. Vaade keskel on objekti suunagraafik, kus iga suuna kohta on arv, mis näitab kui tihti objekt ühes või teises suunas liikunud. Allpool on kaart, kus on kuvatud objekti kaks kohustuslikut ja üks optionaalne trajektoorigid – roheline värviga on kohustuslik üldine objekti trajektoor, punase värviga on kohustuslik anomaalne trajektoor mis on statistilise analüüsi tulemus ja kollase värviga on optionaalne objekti anomaalne trajektoor, mis on arvutatud inimese abiga. Kui kaardi peal kollast trajektoori ei ole siis see tähendab, et inimese arvates anomaaliast liikumist ei olnud. (Lisa 5)

6.2.3 Tabeli vaade ja raporti komponent

Mitme objekti informatsiooni manipuleerimiseks, analüüsimiseks ja raporteetmiseks on tehtud tabeli ja rapordi komponendid. Tabeli all on seade, kui mitu objektid on vaja kuvata tabelis. Praegu on võimalik kuvata viis, kümme või viisteist objekti tabelis. Iga objekti vasakul poolel on selle objekti tehtud raportite aruannete ajalugu. Uue rapordi koostamiseks on vaja klõpsata

pluss nupule. Tekib uus aken, kus objekti tüübi ja nimi on eelsisestatud. Rapordi komponendis on võimalik lisada kommentaare, mis annaksid detailne ülevaade raportist. Rapordite aruannete ajalugu ei ole veel imlementeeritud. Eeldatakse, et raporti koostamise teade peaks tulema emailile. Lisaks, iga objekti kohta tabelis on esitatud id, tüüp, nimi ja koordinaadid. (Lisa 6)

7 Kokkuvõte

Lõputöö raames olid implementeeritud mitmed erinevad algoritmid anomaaliade liikumise tuvastamiseks. Selleks, et teada, milline algoritm on kõige efektiivsem oli võetud Internetist kaks andmestikku, kus vaadeldavad objektid oleksid sarnased tuleviku kasutatava andmetega. Mõlemale andmestiku peal oli proovitud rakendada statistilise analüüsi algoritme Pythoni programmeerimis keelega, kuid tulemuste visualiseerimiseks UI paneeli peal oli võetud ainult üks nendest. Arendamise protsessi jooksul selgus millist algoritmi on kõige mõistlikum ja parem kasutada olemasoleva andmestikuga. Meskonnaga oli otsustatud kasutada standardhälvet lõpliku algoritmina, kuna see on hästi implementeeriv ja põhinedes tehtud katsete põhjal annab häid tulemusi. Samuti kasutades standardhälvet pole vaja häälestada parameetreid, mis põhjustab implemnteerimise lihtsust.

Lõputöös ei olnud realiseeritud AI mooduli ühendamist, kuna see on teise meeskonna lõputöö teema ja antud töö kirjutamise jooksul nad ei jõudnud seda valmis teha.

Objektide andmete manipuleerimiseks ja visualiseerimiseks on tehtud UI paneel. Paneelil on neli komponenti - peamine kaardi vaade, graafikute ja analüüsi vaade, tabeli vaade ja raporti komponent. Kasutades UI paneelil oleva navigatsiooni riba üleval serval on võimalik nende komponentide vahel liikuda. Andmete haldamiseks paneeli peal on kõik vajalikud tööriistad - kaardid, graafikud ja tabelid. Kaardi peal objektid on kuvatud vastava ikooniga ja objektide sageli külastatud kohad on illustreeritud GPS punktidenä. Lisaks sellele, kaardi peal on võimalik vaadata objektide erinevaid trajektoore - roheline värviga on objektide üldine trajektoor ja punase värviga on objektide anomaalne trajektoor. Paneele kasutades saab inimene ka kindla objekti andmete rohkem informatsiooni - anomaalia liikumise trajektoor ja üldine objekti trajektoor, tema kiirus kindlal ajahetkel. Kõik andmed on illustreeritud graafikute ja kaartide peal. Iga objekti kohta saab koostada raporti ja eeldatakse, et raport läheb otse e-mailile. Raporti salvestamine andmebaasis ja saatmine e-mailile ei olnud realiseeritud.

Edasine töö

Lõputöö raames oli leitud kõige efektiivsem algortim andmete tootlemiseks. Objektid ja statilise analüüsi tulemused on kuvatud UI paneeli peal, ning on võimalik vaadata rohkem informatsiooni kindla objekti kohta. Samuti UI paneeli saab edasi areneda. Lõputöö raames ei olnud kuvatud UI paneeli peal automaatne anomaaliade kontrollimise tulemust. Antud töö jooksul kogutud teadmised ja tulemused võivad osutada kasulikuks järgmiste analüüsiga seotud projektide tegijatele ning ilmtingimata aitavad Falconers OÜ kiirendada uurimisprotsessi ja rakendada saadud teadmised oma toote edasi arendamiseks.

Praeguseks hetkeks kõik analüüsi protsessid on tehtud *mock-up* andmetega. Tulevikus *mock-up* andmete asemel peaks olema võimalus kasutada objektid mis on tuvastatud närvivõrgu kasutades. Lõputöö raames on pandud hästi tugev alus selle idee edasi arendamiseks.

Kasutatud allikad

- [1] Flask web development one drop at a time, Pallets, 2010. [Võrgumaterjal]. Available: <https://flask.palletsprojects.com/en/1.1.x/> [Kasutatud 31 märts 2021]
- [2] React. A JavaScript library for building user interfaces, Facebook, 2021. [Võrgumaterjal]. Available: <https://reactjs.org/> [Kasutatud 3 aprill 2021]
- [3] Chart.js, Chart.js contributors, 2021. [Võrgumaterjal]. Available: <https://www.chartjs.org/docs/master/> [Kasutatud 5 aprill 2021]
- [4] Introduction, Paul Le Cam and contributors, 2021. [Võrgumaterjal]. Available: <https://react-leaflet.js.org/docs/start-introduction/> [Kasutatud 14 aprill 2021]
- [5] Accelerate how you build, share and run modern applications, Docker, 2021. [Võrgumaterjal]. Available: <https://www.docker.com/> [Kasutatud 22 aprill 2021]
- [6] What is PostgreSQL, The PostgreSQL Global Development Group, 2021. [Võrgumaterjal]. Available: <https://www.postgresql.org/about/> [Kasutatud 5 mai 2021]
- [7] Aegrea esmaanalüüs, Liina-Mai Tooding, 2020. [Võrgumaterjal]. Available: <http://samm.ut.ee/aegrea-esmasanal%C3%BC%C3%BCs>. [Kasutatud 26 märts 2021]
- [8] GitHub. Fail “gps_distances.py”, José Aniceto, 2018. [Võrgumaterjal]. Available: <https://gist.github.com/jAniceto/98c195ffe714d25a6a65f00f979f145a>. [Kasutatud 5 veebruar 2021]
- [9] How DBSCAN works and why should we use it?, Kelvin Salton do Prado, 2017. [Võrgumaterjal]. Available: <https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80>. [Kasutatud 5 märts 2021]
- [10] DBSCAN Parameter Estimation Using Python, Tara Mullin, 2020. [Võrgumaterjal]. Available: <https://medium.com/@taramullin/dbscan-parameter-estimation-ff8330e3a3bd> [Kasutatud 4 mai 2021]
- [11] DBSCAN Python Example: The Optimal Value For Epsilon (EPS), Cory Maklin, 2019. [Võrgumaterjal]. Available: <https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc> [Kasutatud 30 aprill 2021]
- [12] On-line vibration monitoring and diagnosing of a multi-megawatt wind turbine gearbox, C.Zhu, ... Z. Liu, 2014. [Võrgumaterjal]. Available: <https://www.sciencedirect.com/topics/engineering/fast-fourier-transform>. [Kasutatud 10 märts 2021]
- [13] Understanding the Fourier Transform by example, Ritchie Vink, 2017. [Võrgumaterjal]. Available: <https://www.ritchievink.com/blog/2017/04/23/understanding-the-fourier-transform-by-example/>

- [14] Anomaly Detection – Part 1, Raviteja Arava, 2020. [Võrgumaterjal]. Available: <https://medium.com/analytics-vidhya/anomaly-detection-part-1-acf1a993b573> [Kasutatud 28 aprill 2021]
- [15] Isolation forest, scikit-learn developers (BSD License), 2020. [Võrgumaterjal]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html> [Kasutatud 24 aprill 2021]
- [16] Multivariate Outlier Detection with Isolation Forests, Teun Cuijpers, 2018. [Võrgumaterjal]. Available: <http://www.extended-cognition.com/2018/11/15/multivariate-outlier-detection-with-isolation-forests/>. [Kasutatud 24 aprill 2021]
- [17] Kirjeldav statistika, Kadri Rootalu, 2014. [Võrgumaterjal]. Available: <http://samm.ut.ee/kirjeldav-statistika>. [Kasutatud 16 märts 2021]
- [18] GitHub. Repository “movement”, Abdishakur Hassan, 2019. [Võrgumaterjal]. Available: <https://github.com/shakasom/movement>. [Kasutatud 7 veebruar 2021]
- [19] How Fast Does The Golden Eagle Fly?, Eleanor Blaine, 2017. [Võrgumaterjal]. Available: <https://animals.mom.com/how-to-train-a-dog-to-walk-with-a-leash-13582930.html>. [Kasutatud 14 märts 2021]
- [20] Azimuth Calculator, Bogna Szyk, 2021. [Võrgumaterjal]. Available: <https://www.omnicalculator.com/other/azimuth> [Kasutatud 7 mai 2021]
- [21] GitHub. File “compassbearing.py”, Jérôme Renard, 2020. [Võrgumaterjal]. Available: <https://gist.github.com/jeromer/2005586>. [Kasutatud 19 märts 2021]
- [22] Measuring Directions With “points of a compass” and Azimuth. [Võrgumaterjal]. Available: <https://web.ics.purdue.edu/~braile/eas100/Azimuth.pdf> [Kasutatud 24 märts 2020]

Lisa 1 – “Funktsioon kauguse arvutamiseks kahe GPS punktide vahel”

Järgnev kood on loodud allika [8] materjali põhjal

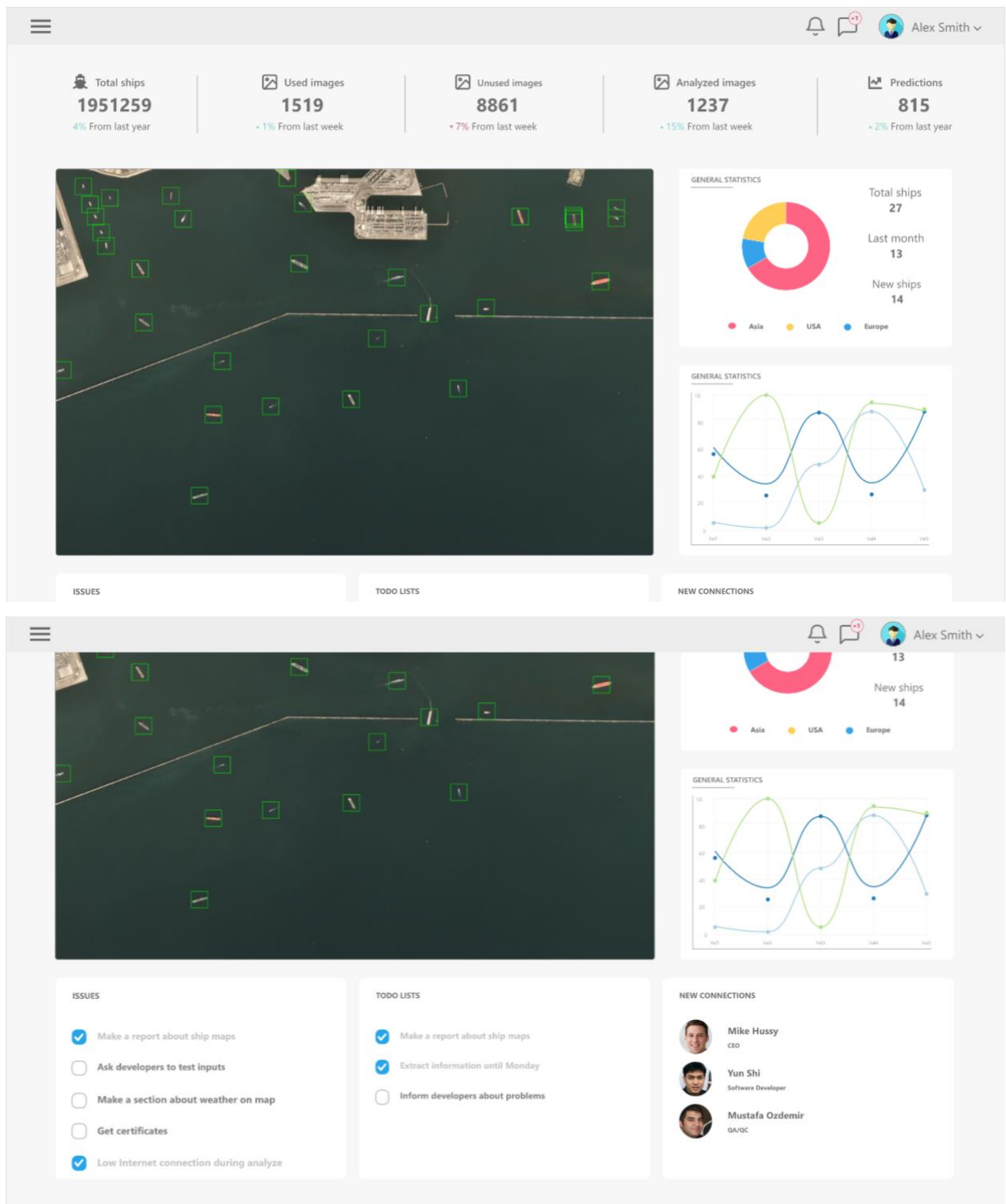
```
def gps_points_calc(lon1, lon2, lat1, lat2):  
    R = 6372800  
    phi1, phi2 = math.radians(lat1), math.radians(lat2)  
    dphi = math.radians(lat2 - lat1)  
    dlambd = math.radians(lon2 - lon1)  
  
    a = math.sin(dphi / 2) ** 2 + \  
        math.cos(phi1) * math.cos(phi2) * math.sin(dlambd / 2) ** 2  
    return 2 * R * math.atan2(math.sqrt(a), math.sqrt(1 - a)) / 1000
```

Lisa 2 – “Funktsioon asimuut väärtuse arvutamise kahe GPS punktide vahel”

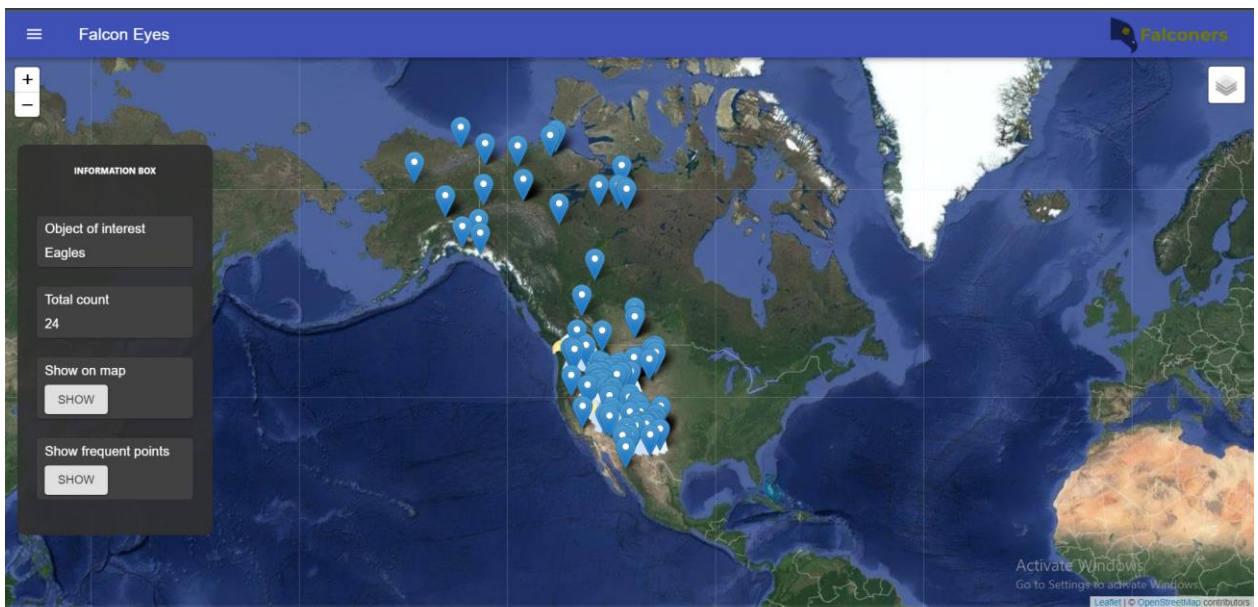
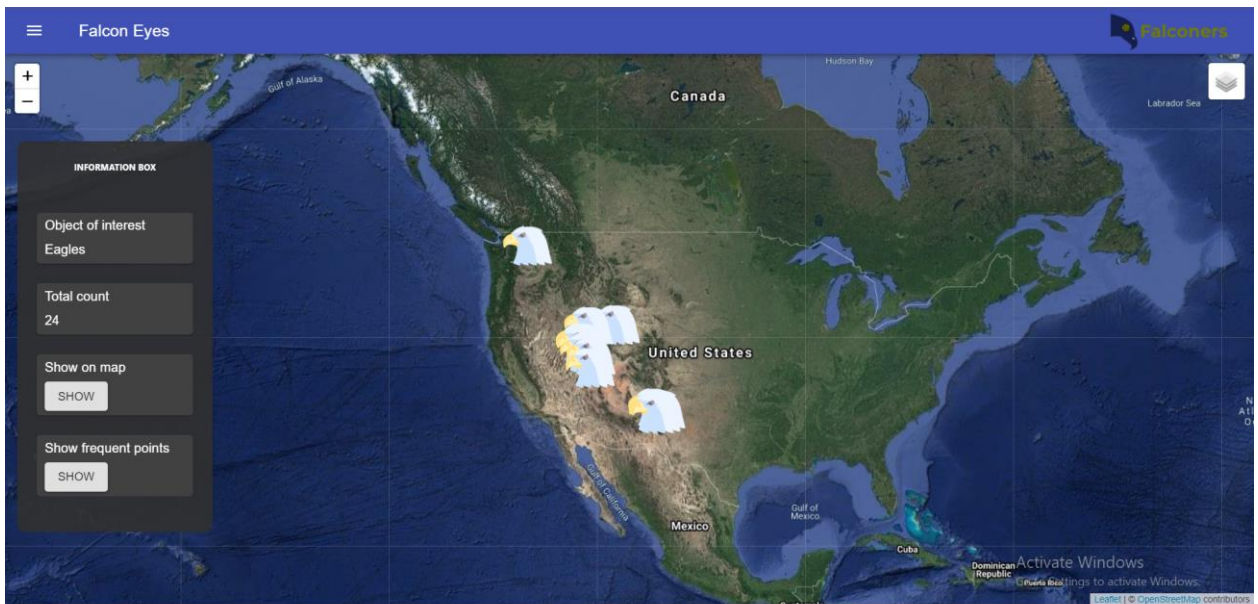
Järgnev kood on võetud [20] allikast

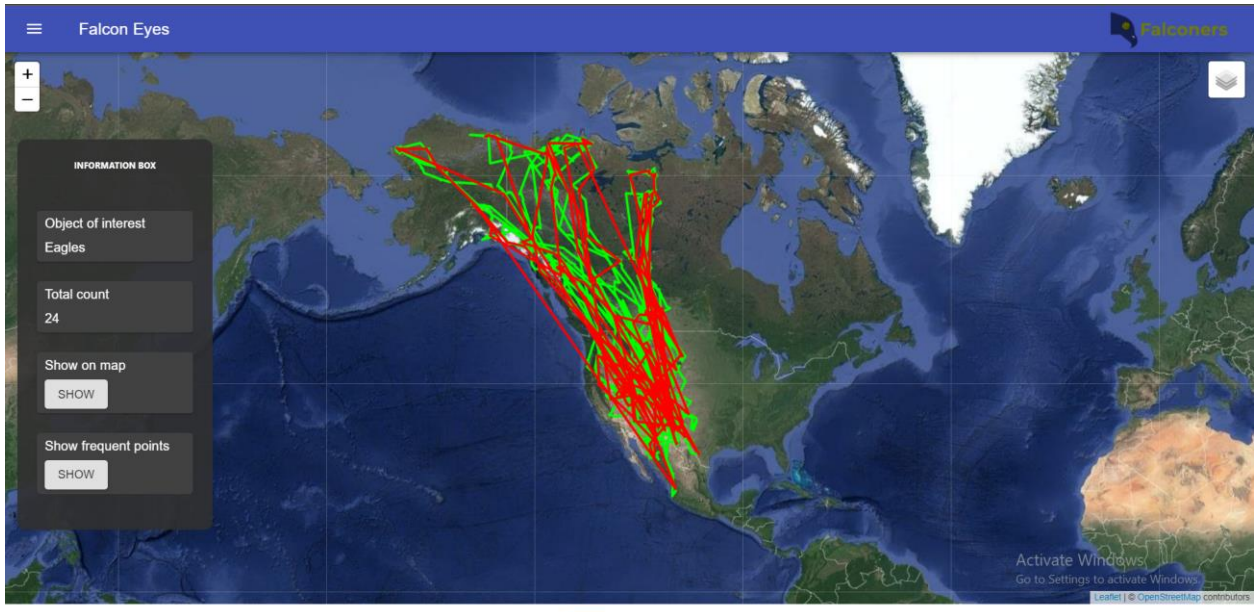
```
def calculate_initial_compass_bearing(points):
    azimuth = []
    for i in range(len(points) - 1):
        lat1 = math.radians(points[i][1])
        lat2 = math.radians(points[i + 1][1])
        diff_long = math.radians(points[i + 1][0] - points[i][0])
        x = math.sin(diff_long) * math.cos(lat2)
        y = math.cos(lat1) * math.sin(lat2) - (math.sin(lat1)
                                                * math.cos(lat2)
                                                * math.cos(diff_long))
        initial_bearing = math.atan2(x, y)
        initial_bearing = math.degrees(initial_bearing)
        compass_bearing = (initial_bearing + 360) % 360
        azimuth.append(compass_bearing)
    return azimuth
```

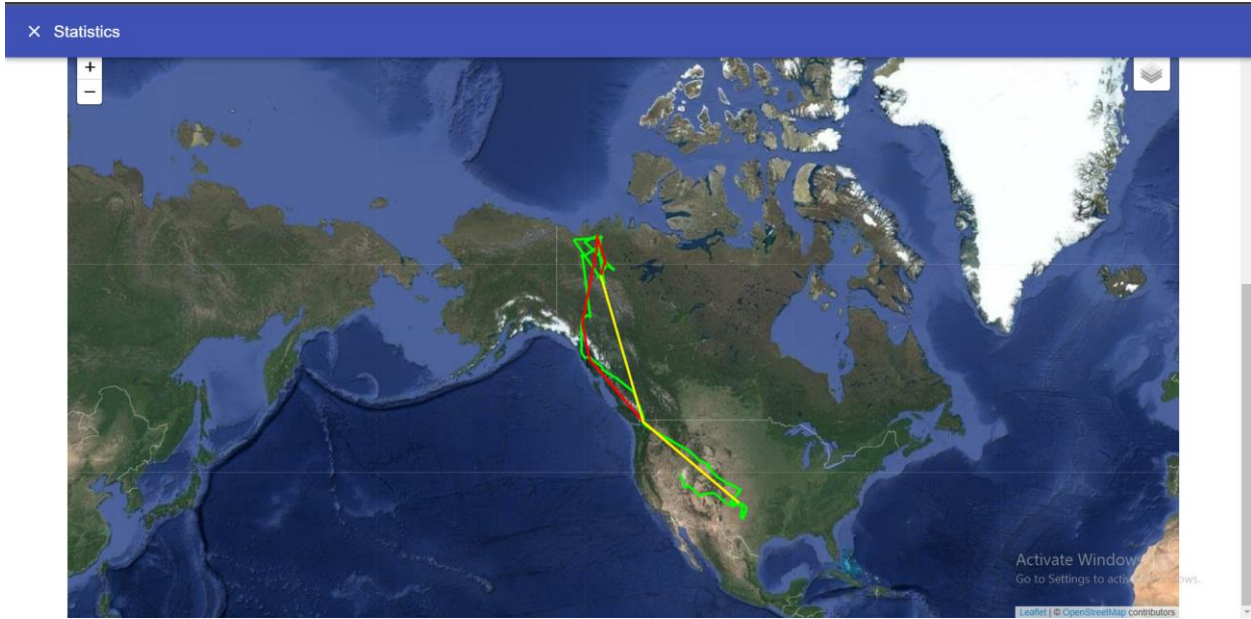
Lisa 3 – Kasutajaliidese prototüüp



Lisa 4 – Kasutajaliidese peamine leht







Lisa 6 – Kasutajaliidese tabeli ja raporti component

Falcon Eyes				
	Id	Type	Name	Coordinates
History	11515	HawkWatch International Golden Eagles	Aquila chrysaetos/1	34.704-106.411
History	18549	HawkWatch International Golden Eagles	Aquila chrysaetos/2	34.911-105.945
History	20948	HawkWatch International Golden Eagles	Aquila chrysaetos/3	40.424-114.271
History	21166	HawkWatch International Golden Eagles	Aquila chrysaetos/4	41.905-113.464
History	21168	HawkWatch International Golden Eagles	Aquila chrysaetos/5	42.025-110.589
History	21259	HawkWatch International Golden Eagles	Aquila chrysaetos/6	48.02-120.094
History	21260	HawkWatch International Golden Eagles	Aquila chrysaetos/7	40.495-114.35799999999999
History	21839	HawkWatch International Golden Eagles	Aquila chrysaetos/8	34.699-106.399
History	26353	HawkWatch International Golden Eagles	Aquila chrysaetos/9	34.69-106.426
History	26359	HawkWatch International Golden Eagles	Aquila chrysaetos/10	34.522-106.197

Rows per page: 10 1-10 of 24

Activate Windows
Go to Settings to activate Windows.

Falcon Eyes				
	Id	Type	Name	Coordinates
History	11515	HawkWatch International Golden Eagles	Aquila chrysaetos/1	34.704-106.411
Previous reports				
2020-07-05	https://es.lipsum.com/			
2020-01-05	https://es.lipsum.com/			
2020-09-05	https://es.lipsum.com/			
2020-03-05	https://es.lipsum.com/			
History	18549	HawkWatch International Golden Eagles	Aquila chrysaetos/2	34.911-105.945
History	20948	HawkWatch International Golden Eagles	Aquila chrysaetos/3	40.424-114.271
History	21166	HawkWatch International Golden Eagles	Aquila chrysaetos/4	41.905-113.464
History	21168	HawkWatch International Golden Eagles	Aquila chrysaetos/5	42.025-110.589
History	21259	HawkWatch International Golden Eagles	Aquila chrysaetos/6	48.02-120.094
History	21260	HawkWatch International Golden Eagles	Aquila chrysaetos/7	40.495-114.35799999999999
History	21839	HawkWatch International Golden Eagles	Aquila chrysaetos/8	34.699-106.399
History	26353	HawkWatch International Golden Eagles	Aquila chrysaetos/9	34.69-106.426
History	26359	HawkWatch International Golden Eagles	Aquila chrysaetos/10	34.522-106.197

New report ✕

Object type
HawkWatch International Golden Eagles

Object name
Aquila chrysaetos/1

Useful details

SUBMIT REPORT

Activate Windows
Go to Settings to activate Windows.

Lisa 7 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Meie, Aleksandr Aleksandrov, Maria Pazekova, Anastassia Lobatšjova

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "KAUGSEIRE VAATLUSE ANDMETE PÕHINEVA STATISTILISE ANALÜÜSI ALGORITMIDE RAKENDAMINE JA VISUALISEERIMINE ANOMAALIAATE TUVASTAMISEKS, mille juhendaja on Evelin Halling ja kaasjuhendaja on Martin Simon
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Oleme teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

18.05.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.