

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Alan Parik 213074IAIB

**ÜJUVATE OBJEKTIDE JA ÕLIREOSTUSTE
TRIIVPROGNOOSI SÜSTEEMI ARENDAMINE EESTI
MEREALA JAOKS**

Bakalaureusetöö

Juhendaja: Siim Pärt
MsC

Kaasjuhendaja: Evelin Halling
PhD

Tallinn 2024

Autorideklaratsioon

Kinnitan, et olen 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.

Autor: Alan Parik

27.05.2024

Annotatsioon

Käesolev bakalaureusetöö keskendub ujuvate objektide ja õlireostuste triivprognoosi süsteemi arendamisele Läänemere ala jaoks. Selle eesmärk on luua veebirakendus, mis võimaldab ennustada ujuvate objektide, nagu näiteks paat või puidujäätmed, ja õlireostuste liikumist ja selle muutumist, nii mere pinnal, kui ka pinna all.

Veereostused võivad tekitada pikaajalisi kahjustusi maailma merekeskonnale ja Eesti riigi majandusele. Läänemeri on üks maailma kõige tundlikumaid merekeskkondi, kus õlireostused ja ujuvad objektid võivad põhjustada ökosüsteemile ja mereorganismidele kahju ning ohustada ka inimeste tervist.

Probleem seisnes vajaduses arendada tööriista, mis võimaldaks teadlastel modelleerida merereostuse liikumisi ja arenguid. Selle probleemi lahendamiseks arendas töö autor veebirakendust, mis võimaldab kasutajal modelleerida reostuste triivprognoose Läänemere piirkonnas. Rakendus põhineb *OpenDrift* raamistikul ja on loodud eesmärgiga anda kasutajatele vahend, millega nad saavad paremini mõista ja ennustada ujuvate objektide ja õlireostuste käitumist merekeskkonnas.

Antud töö tulemusena sai valmis veebirakendus, mis suudab ennustada ujuvate objektide ja õlireostuste triivimist Läänemere piirkonnas, kasutades selleks *MSI Leeway* ja *MSI OpenOil* mudeleid. Lisaks rakenduse jaoks töötati välja uus skaleeritav ja modulaarne arhitektuur. Uue arhitektuuri eesmärk on muuta rakendus paindlikumaks ja laiendatavamaks, võimaldades teistel arendajatel rakendust lihtsamalt kohandada erinevate mudelite kasutamiseks või laiendada selle funktsionaalsust vastavalt vajadustele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 7 peatükki, 16 joonist, 8 tabelit.

Abstract

Development of a Drift Forecast System for Floating Objects and Oil Spills for the Estonian Sea Area

This thesis focuses on developing a drift forecast system for floating objects and oil spills for the Baltic Sea area. Its aim is to develop a web application that predicts the movement and changes of floating objects, such as boats or wood debris, and oil pollution, both on the sea surface and underwater.

Water pollution can cause long-term damage to the marine environment and the Estonian economy. The Baltic Sea is one of the world's most sensitive marine environments, where oil spills and floating objects can harm ecosystems and water organisms, as well as endanger human health.

The problem lay in the need to develop a tool that would allow scientists to model the movements and developments of marine pollution. To address this problem, the author of the thesis developed a web application that allows users to model drift forecasts for pollutions in the Baltic Sea region. The application is based on the *OpenDrift* framework and is designed to provide users with a tool to better understand and predict the behavior of floating objects and oil pollution in the marine environment.

As a result of this work, a web application was created that can predict the drift of floating objects and oil pollution in the Baltic Sea region, using the *MSI Leeway* and *MSI OpenOil models*. Additionally, a new scalable and modular architecture was developed for the application. The goal of the new architecture is to make the application more flexible and extensible, allowing other developers to more easily adapt the application for use with different models or to expand its functionality according to their needs.

The thesis is written in Estonian and is 29 pages long, including 7 chapters, 16 figures and 8 tables.

Lühendite ja mõistete sõnastik

API	Rakendusliides (<i>Application Programming Interface</i>)
IDE	Integreeritud programmeerimiskeskond (<i>Integrated Development Environment</i>)
MSI	Meresüsteemide instituut (<i>Department Of Marine Systems</i>)
STW	SeaTrack Web
CI/CD	Pidev kooste/Pidev integratsioon (<i>Continuous integration/Continuous delivery</i>)
GPLv2	GNU Üldine avalik litsents versioon 2 (<i>GNU General Public License version 2</i>)
NetCDF	Võrgu ühine andmevorm (<i>Network Common Data Form</i>)
OWP	Õli ilmastikumõjud (<i>Oil weathering processes</i>)
HELCOM	Läänemere merekeskkonna kaitse komisjon (<i>Baltic Marine Environment Protection Commission</i>)
WSGI	Veebiserveri lüüsi liides (<i>Web Server Gateway Interface</i>)
CLI	Käsurealiides (<i>Command-line interface</i>)
AJAX	Asünkroonne JavaScript ja XML (<i>Asynchronous JavaScript And XML</i>)

Sisukord

1	Sissejuhatus	9
1.1	Probleem	9
1.2	Eesmärk	10
1.3	Valideerimine	10
2	Taust	12
2.1	Merekeskkonna reostusprobleemid	12
2.1.1	Õlireostused	12
2.1.2	Ujuvad objektid	14
2.2	Rakenduse protoüübi analüüs	15
2.2.1	OpenDrift	15
2.2.2	NetCDF	19
2.2.3	Flask	19
2.2.4	AJAX	20
3	Uue arhitektuuri väljatöötamine	21
3.1	Olemasolev funktsionaalsus	21
3.2	Uus arhitektuur	22
4	Rakenduse arendamine	25
4.1	Docker'i seadistamine	25
4.2	CI/CD seadistamine	25
4.2.1	Testimine	25
4.3	Mitmete OpenDrift mudelite kasutamine	27
4.4	Rakenduse ülevaade	27
4.5	Leitud vigade ja ebamugavuste parandamine	33
4.5.1	Genereetud piltide ja GIF-ide salvestamine	33
4.5.2	API päringud ja lehekülgede "props" atribuudid	33
4.5.3	Lähtekoodis fikseeritud konstandid	33
5	Tulemuste valideerimine	35
5.1	Rakenduse tulemuste korrektsus	35
5.2	Kasutajamugavus ja andmete ülevaatlikkus	35
5.3	Rakenduse testimine erinevates kasutusjuhtumites	35
6	Edasiarendamise võimalused	40

7 Kokkuvõte	41
Kasutatud kirjandus	43
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	46
Lisa 2 – models.json mudelite konfiguratsiooni faili näide	47
Lisa 3 – MSI Leeway mudeli .mustache konfiguratsiooni näide	49
Lisa 4 – MSI OpenOil mudeli .mustache konfiguratsiooni näide	51

Jooniste loetelu

1	<i>Aastatel 1998–2022 HELCOMi riikide poolt, seirelendudega, tuvastatud ebaseaduslikud õlireostused [10].</i>	13
2	<i>OpenDrift LagrangianArray moodul ja seosed kliimamudelite objektidega. [23]</i>	16
3	<i>Peamised õlireostust mõjutavad transpordi- ja ilmastikuprotsessid (OWP) [9].</i>	17
4	<i>OpenDrift-i simulatsiooni moodulite struktuur ja kasutatavad mudelid [23].</i>	18
5	<i>Rakenduse prototüübi mudelite valikud.</i>	21
6	<i>Rakenduse prototüübi arhitektuur.</i>	22
7	<i>Rakenduse uus arhitektuur.</i>	23
8	<i>CI/CD skeem.</i>	26
9	<i>Mudelite valiku vaade.</i>	28
10	<i>Uue triivprognoosi projekti loomine.</i>	28
11	<i>Uue projekti sisendparameetrite väärtused.</i>	29
12	<i>Õlireostuse simulatsiooni tulemus.</i>	30
13	<i>Õlireostuse simulatsiooni vertikaalne vaade.</i>	31
14	<i>Õlireostuse simulatsiooni horisontaalne vaade.</i>	31
15	<i>Simulatsiooni graafikud.</i>	32
16	<i>MSI Leeway ja MSI OpenOil mudelite simulatsioonide kiiruse sõltuvus "osakeste arv" sisendparameetrist.</i>	39

Tabelite loetelu

1	<i>Keskmiste (7–700 tonni) ja suurte (>700 tonni) õlireostuste arv aastatel 1970–2020 [1].</i>	14
2	<i>MSI Leeway mudeli testimine erinevate reaostuste raadiustega.</i>	36
3	<i>MSI Leeway mudeli testimine erinevate osakeste arvuga.</i>	36
4	<i>MSI Leeway mudeli testimine erinevate objektidega.</i>	36
5	<i>MSI OpenOil mudeli testimine erinevate õlireostuste raadiustega.</i>	37
6	<i>MSI OpenOil mudeli testimine erinevate osakeste arvuga.</i>	37
7	<i>MSI OpenOil mudeli testimine erinevate õli kogustega.</i>	38
8	<i>MSI OpenOil mudeli testimine erinevate õli tüübidega.</i>	38

1. Sissejuhatus

Merekeskkonna ettearvamatut iseloomu esitab olulisi väljakutseid sellistele juhtumitele nagu õlireostused ja otsingu- ning päästeoperatsioonide läbiviimine. Selliste intsidentide tõhus juhtimine nõuab aga täpseid ennustusvahendeid, mis suudavad simuleerida saasteainete liikumist ja levikut või prognoosida merel kadunud esemete või inimeste triivi.

Triivimudelid on kujunenud olulisteks vahenditeks nende väljakutsetega tegelemisel, pakkudes süsteemset lähenemist, integreerides andmeid okeanograafilistest ja meteoroloogilistest mudelitest ennustamiseks merre valgunud õli või objektide võimalikke liikumisi pinnal.

Triivimudelite kasutamine õlireostusele reageerimisel aitab optimeerida puhastusoperatsioone, ennustades õli levikut ja kontsentratsiooni, mis võimaldab reageerijatel ressursse tõhusamalt jaotada ja vähendada keskkonnakahjustusi. Päästeoperatsioonides on triivimudelid hädavajalikud otsinguala kitsendamiseks, mis on kriitiline arvestades selliste missioonide ajatundlikku olemust. Nende mudelite täpsus ja kasutajamugavus võib oluliselt mõjutada operatiivstrateegiate edu, mõjutades mitte ainult otseseid tulemusi, vaid ka pikaajalist keskkonnakaitset ja inimohutust.

1.1 Probleem

Tänapäeval on Läänemerd ümbritsevates riikides peamiseks objektide ja õlireostuse triivi prognoosimiseks kasutatud mudel SeaTrackWeb (STW) [1]. See mudel on aga kinnise lähtekoodiga ning kolmadad osapooled ei saa selle funktsionaalsust muuta või edasi arendada. Ka kasutajapoolne võimalus erinevate sisendparameetrite muutmiseks või nende seadistamiseks on antud mudeli puhul piiratud.

Lisaks on Eesti ligemale 4000 km pikkune liigendatud rannajoon esitatud väikse täpsusega ning isegi suuremaid saari (näiteks, Naissaar) pole arvestatud. STW annab sõltuvalt ilmaoludest üsna täpse prognoosi avamerel tekkiva või liikuva reostuse puhul, kuid rannikumeri pole piisavalt täpselt esitatud ning lisaks ei võimalda see kasutada sisendiks Eesti oma meremudeleid, mis on kõrgema resolutsiooniga. Peale selle, on STW ka tasuline.

Eestis on hiljuti kasutusele võetud kaks kohalikku kõrge resolutsiooniga operatiivmudelit: NEMO-EST05 [2] hüdrodünaamika ja SWAN-EST [3] lainete jaoks. Triivi

prognoosimiseks on vaja eelnevalt nimetatud kliimamudeleid integreerida modelleerimisüsteemi. STW-süsteem seda aga ei toeta. Seetõttu valiti alternatiivseks kasutamiseks ja edasiarendamiseks Norra Meteoroloogia Instituudis arendatav avatud lähtekoodiga Pythonil põhineva Lagrange'i osakeste trajektoori modelleerimise raamistik OpenDrift [4].

OpenDrifti mugavamaks ja lihtsamaks kasutamiseks seati üles selle vabavaraline ja veebipõhine kasutajaliides Adrift [5], mille edasiarendamisel antud töö põhineb. Püstitatud probleemi lahendamiseks arendatakse selle töö käigus edasi alternatiivset rakendust, mis STW asemel kasutab Adrifti.

1.2 Eesmärk

Antud töö esimene eesmärk on arendada rakendust olemasolevalt prototüübilt edasi, et kasutajal oleks lihtsalt ja mugavalt võimalik saada ülevaatlilikuid andmeid ujuvate objektide ja õlireostuste liikumiste kohta Läänemeres. Peamiseks ülesandeks on implementeerida mitmete mudelite kasutamise võimalust. Töö käigus otsustasid töö autor ja juhendaja kasutada *Leeway* (ujuvad objektid) ja *OpenOil* (õlireostused) mudeleid, kuna just need triivprognoosid on Eesti merealal ning Läänemeres kõige olulisemad.

Töö teiseks eesmärgiks on rakendusele skaleeritava ja modulaarse arhitektuuri loomine, mis võimaldaks teistel arendajatel lihtsasti lisada rakendusse uusi OpenDrifti mudeleid, kohandades rakenduse funktsionaalsust vastavalt nende vajadustele.

1.3 Valideerimine

Püstitatud eesmärkide saavutamiseks uuritakse olemasoleva rakenduse prototüübi arhitektuuri ja funktsionaalsust. Analüüsi käigus saadud teadmisi kasutatakse rakenduse edasiarendamisel ja uue funktsionaalsuse implementeerimisel, vastavalt ülesandepüstitusele. Leitud vead ja muud kasutamise ebamugavused tuuakse välja eraldi loeteluna ning seejärel tehakse vastavad parandused rakenduse funktsionaalsuses.

Nii rakenduse arendamise käigus kui ka selle lõpus viiakse läbi katsetusi seoses rakenduse tööga. Katsetuste käigus selgitatakse, kui ülevaatlikku informatsiooni rakendus kuvab, kui kiiresti rakendus töötab erinevates kasutusjuhtumites ning milline veebiliidese disain pakub kasutajale kõige mugavama töökogemuse.

Sellise valideerimise eesmärk on välja selgitada, kui kasutajasõbralik ja informatiivne on valmis rakendus nendele inimestele, kes reaalselt tegelevad õlireostuste ja päästeoperat-

sioonidega merel.

2. Taust

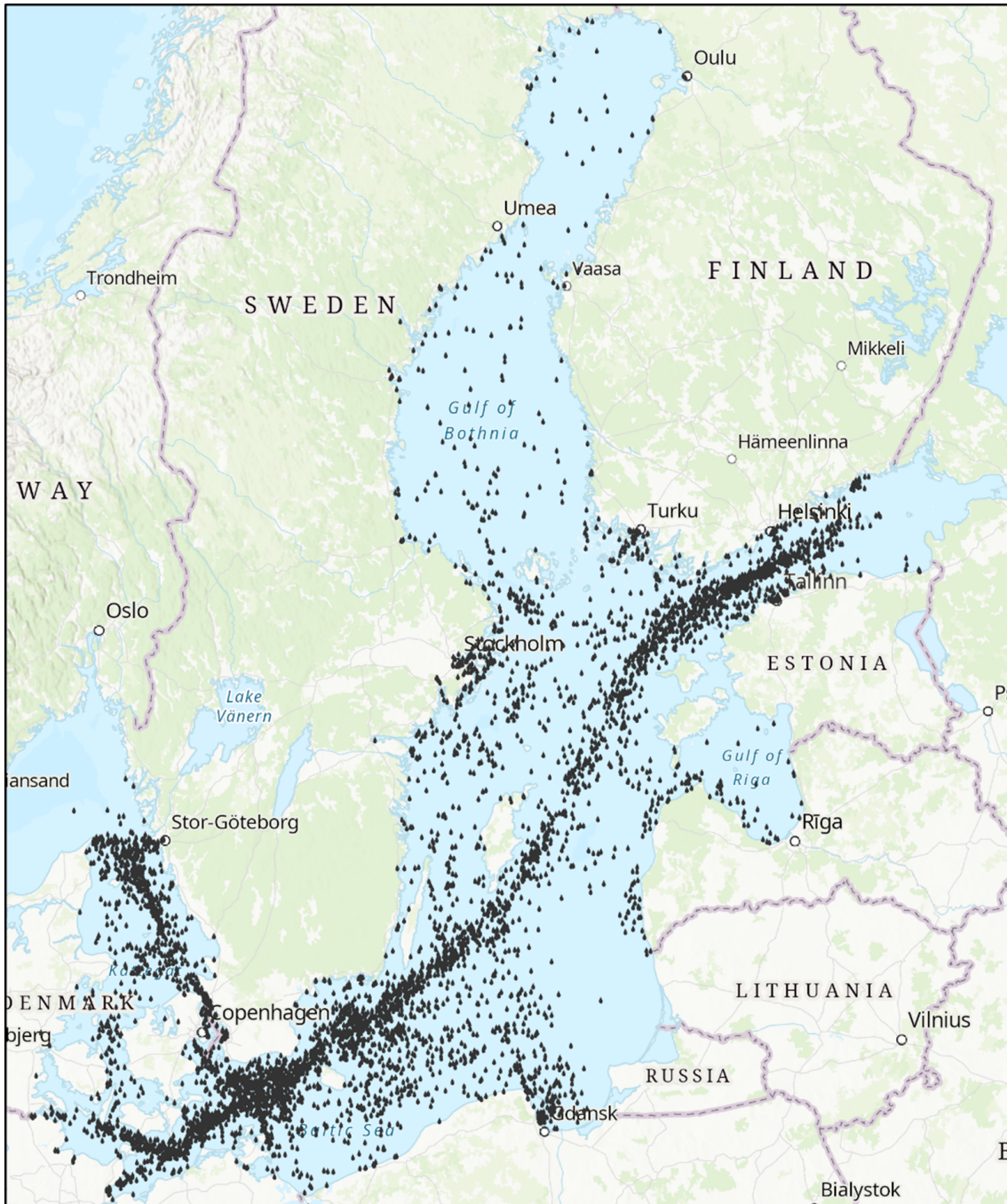
2.1 Merekeskkonna reostusprobleemid

Läänemeri on riimveeline sisemeri, millel on piiratud veevahetus [6, 7], keerukad rannajooned ja hooajaline jääkate. HELCOMi (Helsingi Komisjon) [8] andmetel moodustab suhteliselt väike Läänemeri kuni 15% maailma meretranspordist ja on samal ajal oma eriliste geograafiliste, klimatoloogiliste ja okeanograafiliste omaduste tõttu väga tundlik inimtegevuse keskkonnamõjude suhtes.

2.1.1 Õlireostused

Nafta ja naftatoodete reostused ehk üldisemalt õlireostused võivad pärineda maismaalt või otse merest. Need tekivad enamasti looduslikul eraldumisel, õli transportimisel, naftapuurimisel ja juhuslikult õlitankerite kokkupõrgetest või uppumistest, samuti transporditorustike ja naftapuurtornide rikete tõttu [9]. Suurem osa väiksemast merepõhisest naftareostusest pärineb laevandustegevusest või avamereplatvormidelt ehk nn operatiivsetest heidetest, mis võivad olla kas juhuslikud või tahtlikud.

Joonis 1. näitab aastatel 1998-2022 Läänemerel tuvastatud õlireostuste asukohti. Jooniselt on näha, et ka Eesti ja Soome merealadel (eriti Soome lahes) on nendel aastatel toimunud ning kindlaks tehtud suur hulk õlireostusi.



Joonis 1. Aastatel 1998–2022 HELCOMi riikide poolt, seirelendudega, tuvastatud ebaseaduslikud õlireostused [10].

Kuigi peaaegu 30% ülemaailmsest merekaubandusest on seotud naftaga ja selle toodetega ning igal aastal globaalse naftakaubanduse maht suureneb, on suurte naftareostuste arv aastatega vähenenud (Tabel 1) [11, 12].

Tabel 1. Keskmiste (7–700 tonni) ja suurte (>700 tonni) õlireostuste arv aastatel 1970–2020 [1].

Periood	7-700 tonni	> 700 tonni
1970 - 1979	543	245
1980 - 1989	360	94
1990 - 1999	281	77
2000 - 2009	149	32
2000 - 2019	45	18
2020	3	0
Kokku	1381	466

Teine oluline õlireostuse allikas on potentsiaalselt saastavad laevavrakid [13, 14]. Enamik neist vrakkidest pärineb II maailmasõjast ning olles olnud merepõhjas enam kui 75. aastat on neil korrosiooni tõttu üha kasvav struktuurikahjustuste ja lekete oht [15].

Tänapäeval on Läänemeres peamiseks õlireostuste allikaks laevade operatiivsed naftaheitmed [16] ning nagu ka mujal maailma ookeanides ja meredes, on potentsiaalselt saastavad vrakid esile kerkiv probleem [17].

Pärast naftareostuse avastamist on reostuse likvideerimisel üks peamisi murekohti reostuse trajektoori ja selle ruumilise jaotuse määramine. Naftareostuse liikumise ja arengu simuleerimiseks ja prognoosimiseks on üle maailma kasutusel mitmeid erinevaid mudeleid ja raamistikke [9, 18, 19].

2.1.2 Ujuvad objektid

Ujuvad objektid on kõik sellised esemed või materjalid, mis on võimelised püsima veepinnal ja liikuma veevoolude mõjul. Need võivad hõlmata erinevaid objekte alates looduslikest nagu jäämäed ja merejää kuni inimese poolt valmistatud nagu päästevestid, lastikonteinerid ja muud tüüpi mereaju. Ujuvate objektide hulka kuuluvad ka inimesed, kes on sattunud vette õnnetuste või muude põhjuste tõttu. [20]

Suurem osa õnnetustest juhtuvad just otse sadamates või nende läheduses, seega suureneb ka oht tekitada uusi õnnetusi. Võivad tekkida navigatsiooniohud laevadele ja muudele veesõidukitele, eriti öisel ajal või halva nähtavuse korral, kui vette sattunud ujuvaid objekte on raske märgata või neid ei ole võimalik radariga tuvastada. [21]

Lisaks tekib rannalähedaste õnnetuste korral oht kahjustada väikseid paate või isegi inimesi. Mõned ujuvad objektid võivad olla ohtlikud inimestele, kes viibivad veekogudes ujudes või sukeldumas, eriti kui need objektid on teravad või ohtlike ainetega saastunud.

Seetõttu tekib tõsine vajadus selliste lahenduste järele, mis võimaldavad päästjatel, keskonnakaitseametil ja piirivalveametil kiiresti reageerida ja ohtlikuid objekte merepinnalt eemaldada.

2.2 Rakenduse protoüübi analüüs

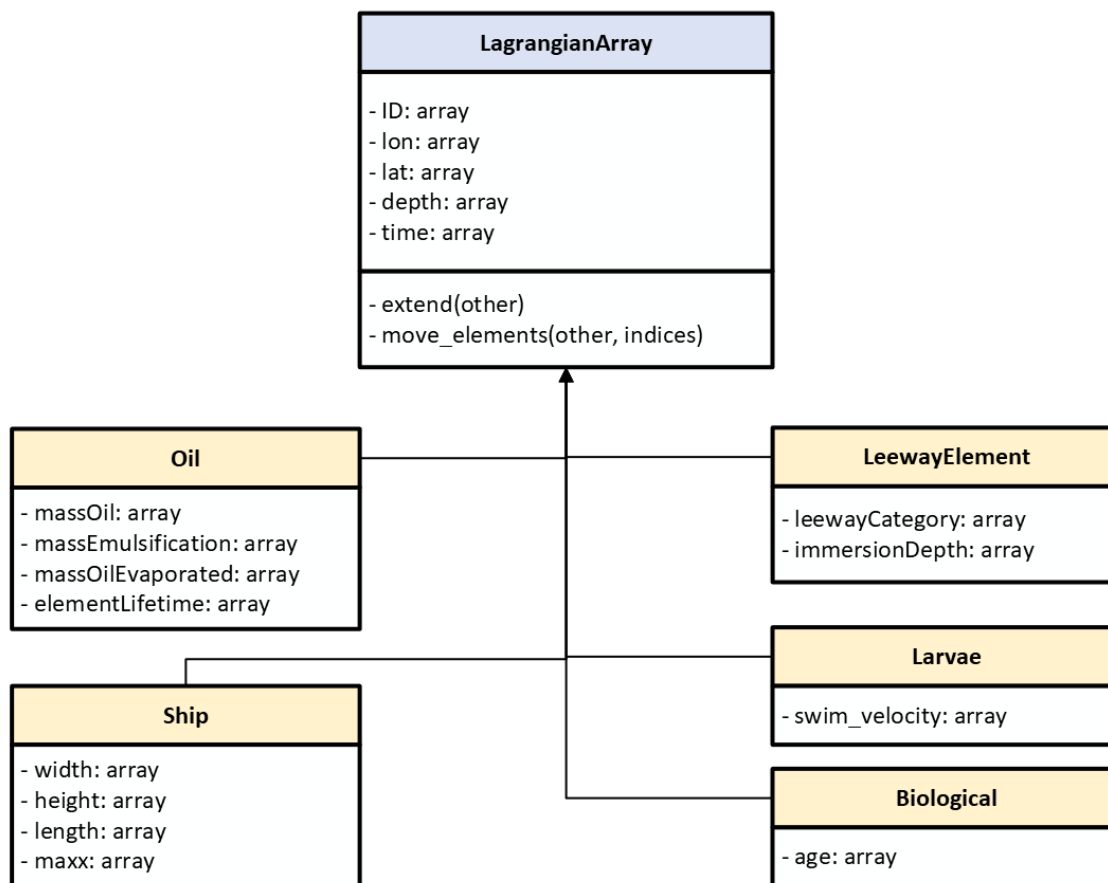
Antud töös arendatud veebirakendus põhineb olemasoleval prototüübil, Adrift, mis on loodud Iiri mereinstituudis [5]. Selles rakenduses merekeskonnaga seotud osakeste modelleerimised ja simulatsioonid põhinevad *OpenDrift* raamistikul, selle toimimiseks vajalike okeanograafiliste ja meteoroloogiliste mudelite sisenditel ning Network Common Data Form (netCDF) liidese funktsionaalsustel [5].

Enne edasiarendamisega alustamist viidi läbi antud prototüübi analüüs. Analüüsi eesmärkideks oli selgitada, milline funktsionaalsus on prototüübis olemas, milliseid vigu koodis esineb ning milliseid tehnoloogiaid antud prototüüp kasutab.

Veebitehnoloogiast on rakenduses kasutatud Flask ja AJAX. Flaskis on ehitatud eesrakendus ning API päringute jaoks on kasutatud AJAX.

2.2.1 OpenDrift

OpenDrift [4] on Pythonis programmeeritud raamistik, mis on mõeldud ookeanis ja atmosfääris olevate osakeste liikumistrajektooride modelleerimiseks. OpenDrift on avatud lähtekoodiga ning on kaitstud GNU üldise avaliku litsentsiga (GPLv2) [22], mis tagab, et töö autoril kui ka tarkvaraarendajal on õigus saada ligipääs OpenDrifti raamistiku lähtekoodile, et seda uurida, muuta ning vastavalt oma vajadustele arendada.



Joonis 2. OpenDrift LagrangianArray moodul ja seosed kliimamudelite objektidega. [23]

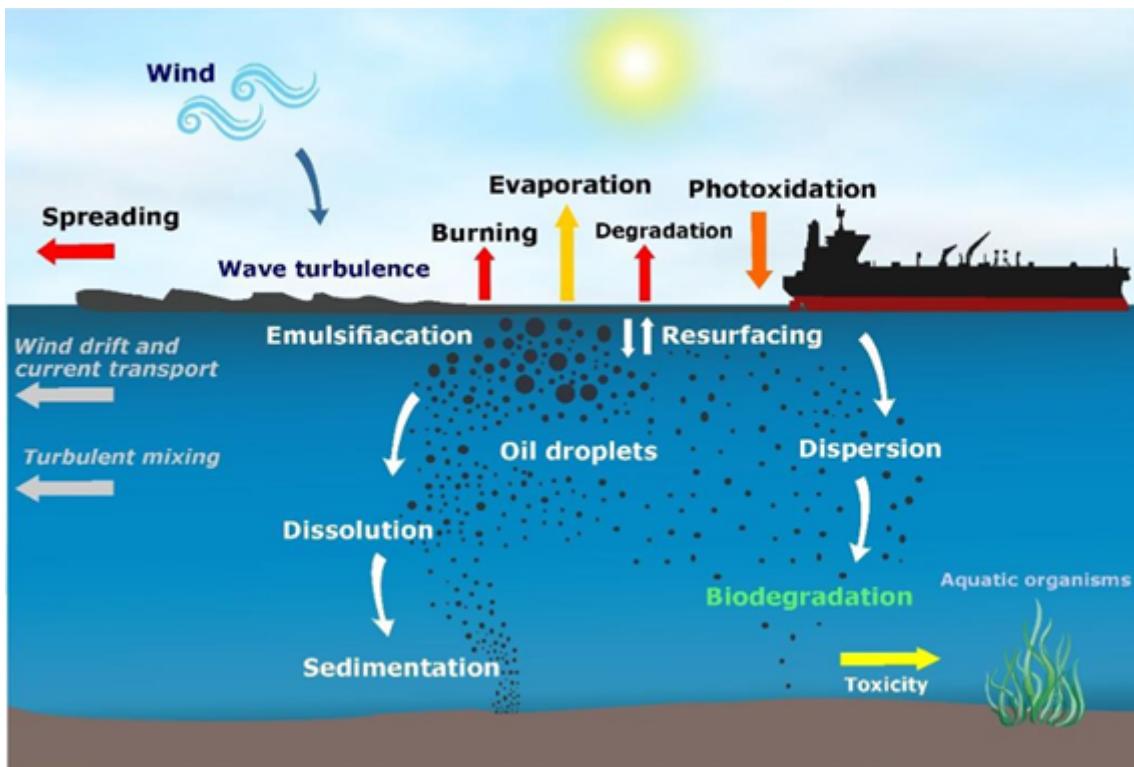
OpenDrift suudab modelleerida simulatsioone erinevate tüüpi osakestega. Iga osakese tüübi jaoks on OpenDriftis olemas eraldi mudelid: *OpenOil* (õlireostused), *Leeway* (ujuvad objektid), *PlastDrift* (mikroplastiku osakesed), *ChemicalDrift* (kemikaalide reostused) ja muud [24]. Iga mudel omakorda kasutab enda jaoks spetsiifilisi osakesi kirjeldavaid objekte. Näiteks *OpenOil* mudel kasutab "Oil" tüüpi objekte (Joonis 2). Sellel objektil on olemas spetsiifilised atribuudid: õli mass, õli emulsiooni mass (õli + vesi), aurunud õli mass, õli osakese eluiga.

OpenDrift on niinimetatud Lagrange'i osakeste triivi moodul, mis põhineb individuaalsete osakeste liikumise jälgimisel läbi aja ja ruumi, simuleerides nende trajektoore erinevate keskkonajõudude mõjul. Mudelisse sisestatakse osakesed, mis esindavad õli või muid aineid, ja iga osakese liikumist mõjutavad tuule põhjustatud veeliikumine, hoovused ja muud füüsikalised tegurid, mis pärinevad atmosfääri ja mere mudelist. Osakeste trajektoorid arvutatakse dünaamiliselt, võttes arvesse muutuvaid keskkonnatingimusi.

OpenDrifti mudelid vajavad sisendiks andmeid okeanograafilistest ja atmosfääriallikatest: tuule kiiruse ja suuna, hoovuse kiiruse ja suuna ning võimalusel ka laineindutseeritud triivi

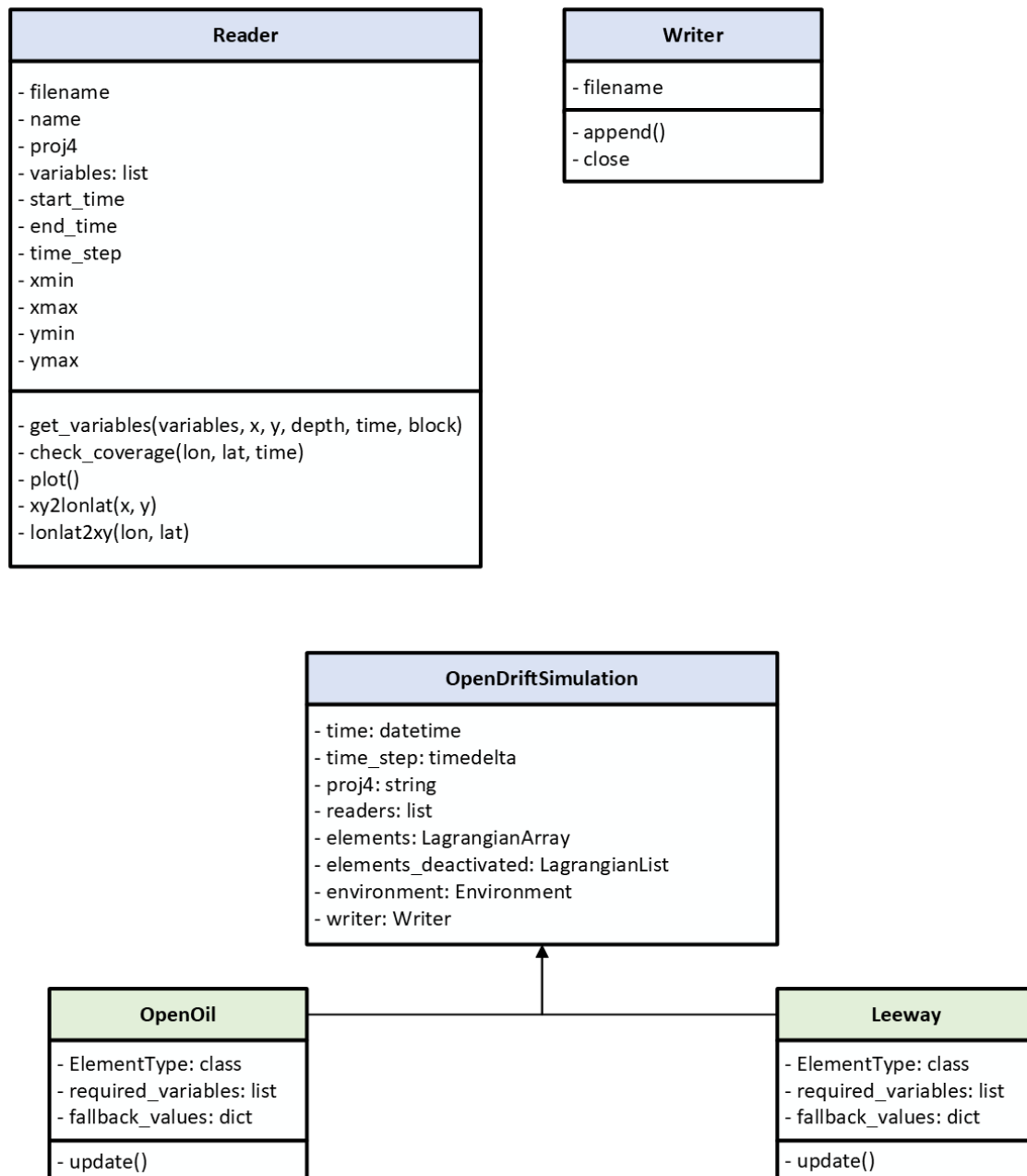
(Stokes'i triiv) andmeid. Antud prototüübi tööks piisab minimaalselt tuulte ja hoovuste andmetest.

Lisaks on võimalus lisada ka selliseid andmeid nagu merevee temperatuur ja soolsus, mis mõjutavad nii vee kui ka õli füüsikalisi omadusi, sealhulgas õli liikumist ja aurustumiskiirust. Vee sügavus, merevee ülesvoolu kiirus ja vertikaalne difusioon mõjutavad õliosakeste vertikaalset jaotumist veesambas. Kuna lained mängivad dispersiooniprotsessides olulist rolli, saab mudelitele ette anda ka lainete keskmise ja maksimaalse perioodi. Kuigi need muutujad ei ole põhitrajektoori modelleerimiseks olulised, parandavad need õliga seotud ilmastiku- ja transpordiprotsesside parameetrite täpsust [25] (Joonis 3).



Joonis 3. Peamised õlireostust mõjutavad transpordi- ja ilmastikuprotsessid (OWP) [9].

OpenDrifti raamistikus on olemas mitmeid mudeleid ja nende alamudeleid. Antud töö raames arendatavas rakenduses kasutatakse ainult kahte: *Leeway* ja *OpenOil*.



Joonis 4. *OpenDrift-i simulatsiooni moodulite struktuur ja kasutatavad mudelid [23].*

Joonis 4. kirjeldab, kuidas OpenDrift raamistiku moodulid on omavahel seotud ning kuidas on antud raamistik ülesehitatud.

- **Reader:** Loeb sisse simulatsiooniks vajalikuid keskkonnaandmeid (hoovuste ja tuulte andmed). Antud töö raames kasutakse andmede hoidmiseks .nc formaadi (netCDF failid).
- **Writer:** Kirjutab genereeritud tulemusi iga simulatsiooni etapi kaupa või simulatsiooni lõpus.
- **OpenDriftSimulation:** Peamine raamistiku moodul, mis vastutab simulatsiooni

genereerimise eest.

- Loeb *LagrangianArray* järjendist olevaid andmeid.
- Initsialiseerib valitud objektide osakesi vastavalt valitud mudelile.
- Alustab osakeste trajektooride arvutamisega kasutades `update()` meetodit, värskendades osakeske asukohti ja omadusi määratud ajaetapi kaupa (näiteks, iga minut, iga 10 minutit, iga tund).
- Osakeste omaduste värskendamiseks kasutatakse keskkonnamuutujaid (tuul, lained, hoovused jne).
- Deaktiveerib osakesi. Deaktiveerimine toimub, näiteks, kui osakese eluiga sai otsa või osake on aurunud.
- Kirjutab saadud tulemust `.json` faili (antud juhul `context.json`).

2.2.2 NetCDF

NetCDF [26] on C-keeles kirjutatud teek, mille baasil on ehitatud `netcdf4-python` liides. Antud rakenduses on NetCDF kasutatud suurte andmekogumite lugemiseks ja kirjutamiseks. NetCDF sobib eriti hästi mitmemõõtmeliste massiivide salvestamiseks, mida tavaliselt esineb teaduslikes andmekogumites. See võimaldab salvestada tõhusalt andmeid koos nendega seotud mõõtmete ja metaandmetega, kuid samaegselt pakkudes kiiret juurdepääsu andmete alamhulkadele andmede lugemise käigus

Antud töö raames on `netCDF` oluliseim omadus sealhulgas ka võimalus opereerida kliimamudelite andmetega, nagu hoovuste kiirused ja suunad, tuule kiirused ja suunad ning nendele vastavad koordinaadid, kuupäevad ja kellaajad.

2.2.3 Flask

Flask [27] on raamistik, mis on mõeldud veebirakenduste loomiseks. Antud raamistikul on olemas põhjalik dokumentatsioon, mis aitab kasutajal kiiresti tööle panna veebiprojekti esialgne variant. Flask kasutab Werkzeug WSGI (Web Server Gateway Interface) ja Click CLI (Command-line interface) tööriistakomplekte ning Jinja mallimootorit.

Werkzeug [28] on põhjalik WSGI veebirakenduste teek, mis seadistamisel ei jõusta pakettide sõltuvusi ning mis võimaldab arendajal iseseisvalt lisada rakendusse vajalikke pakette.

Jinja [29] on kiire, väljendusrikas ja laiendatav mallimootor, mis võimaldab kirjutada Pythoni süntaksiga sarnast koodi ning seejärel edastatada mallile andmed lõpliku dokumendi renderdamiseks.

Click [30] on Pythoni pakett käsurea liideste loomiseks. See on kergesti konfigureeritav, eesmärgiga muuta käsureatööriistade kirjutamise protsess kiireks.

2.2.4 AJAX

Antud rakenduses AJAX [31] (Asynchronous JavaScript And XML) on kasutatud API päringute saatmiseks serveri vastu ning veebilehtede asünkroonse värskendamiseks. See tähendab, et veebilehe osasid on võimalik uuendada ilma tervet lehte uuesti laadimata ehk kõik eesrakenduses renderdamised toimuvad ilma üksteist takistamata.

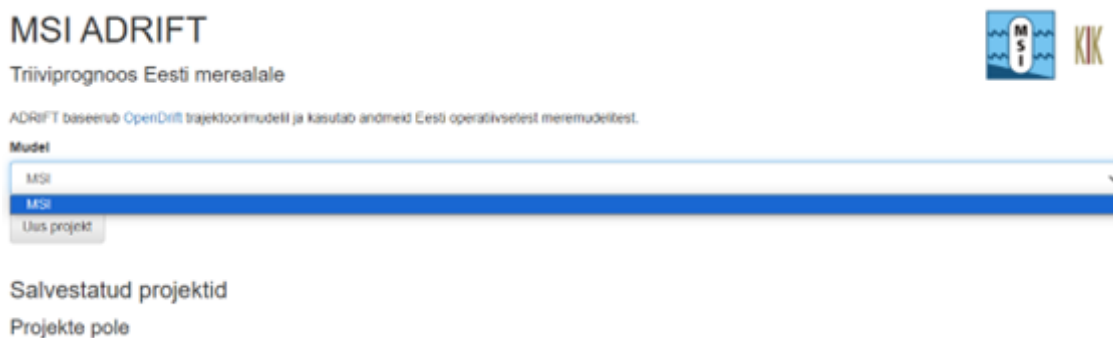
3. Uue arhitektuuri väljatöötamine

Prototüübi analüüs näitas, et olemasolev arhitektuur ei võimalda kasutajal lisada uusi mudeleid ega käivitada rakenduses erinevaid mudeleid samaaegselt. Seetõttu tekkis vajadus välja töötada uus arhitektuur, mis võimaldaks mitmete mudelite kasutamist ja nende samaaegset integreerimist rakendusse.

3.1 Olemasolev funktsionaalsus

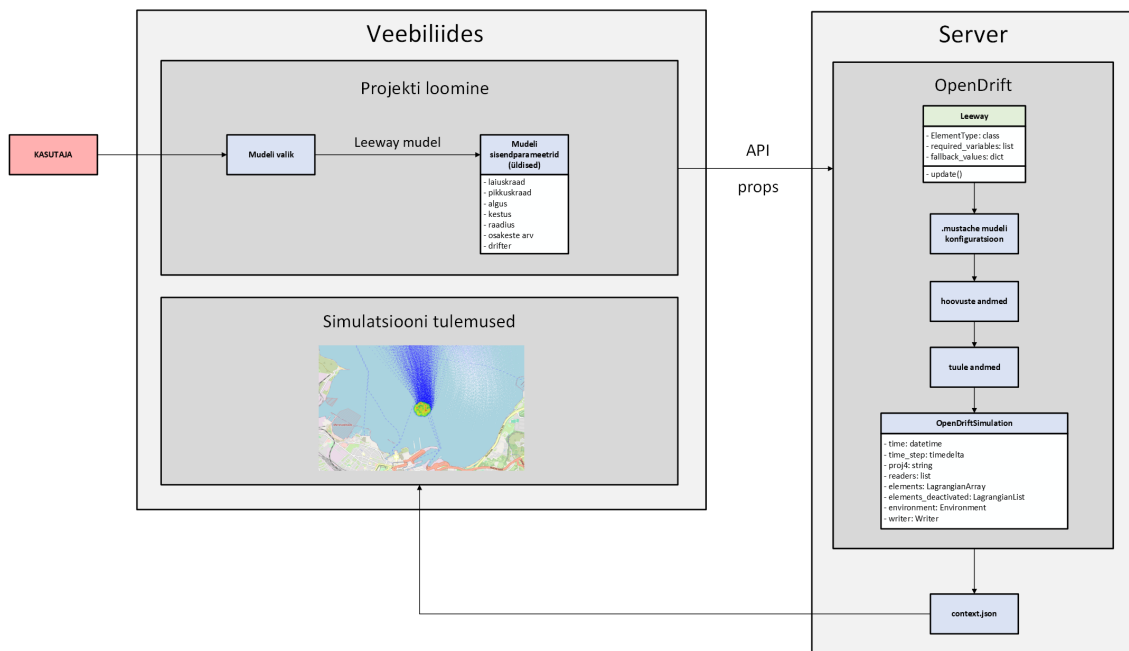
Esialgne rakendus võimaldas kasutajal prognoosida ujuvaid objekte Iiri merel, kasutades Iirimaa ilmastikuportaalidest saadud hoovuste ja tuule andmeid. Varasemalt oli antud rakendus üles seatud Tallinna Tehnikaülikooli meresüsteemide instituudi (MSI) serverisse ning katsetatud selle toimimist Eesti keskkonnamudelitega. Lisaks oli rakendus tõlgitud eesti keelde.

Analüüsi tulemustena selgus, et rakenduse arhitektuur ega selle esialgne funktsionaalsus ei ole piisavad antud töö eesmärkide saavutamiseks. Olemasolev rakendus oli seadistatud töötama ainult ühe konkreetse, *Leeway*, mudeliga (Joonis 5).



Joonis 5. Rakenduse prototüübi mudelite valikud.

Suurem osa prototüübi koodiplokkidest ja muutujatest olid nn "hardcoded" olekus, mis tähendab, et prototüüp ei võimaldanud kasutada erinevaid OpenDrifti mudeleid samaaegselt. Lisaks ei olnud kolmanda osapoole tarkvaraarendajal, olemasoleva arhitektuuri puhul, võimalust lihtsalt integreerida uue mudelitüübi kasutamist ega lisada uusi mudeli seadistamiseks vajalikke parameetreid. Selle tõttu tuli välja töötada uus rakenduse arhitektuur, säilitades samas olemasoleva rakenduse töötamise põhiprintsiibid.



Joonis 6. Rakenduse prototüübi arhitektuur.

Joonis 6 kirjeldab esialgse rakenduse arhitektuuri. Antud arhitektuuris on olemas veebiliides, kus saab kasutada ainult *Leeway* mudelit ning luua uut projekti ujuva objekti triivprognoosi jaoks. Kasutaja saab määrata reostuse alguskoha ning simulatsiooni alguse ja kestvuse ajad.

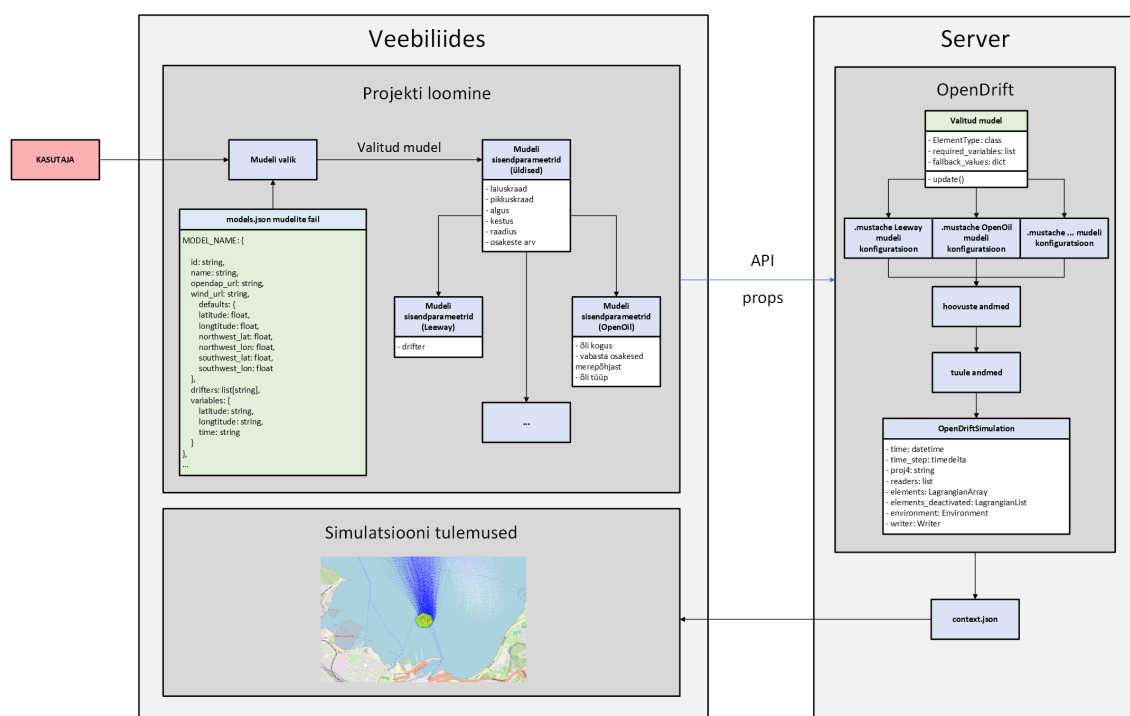
3.2 Uus arhitektuur

Uue arhitektuuri (Joonis 7) kavandamisel arvestati järgmiste põhimõtetega:

- **Modulaarsus:** Arhitektuur peab olema modulaarne, võimaldades erinevate tüüpi mudelite lisamist ja eemaldamist. Antud töös on kõige olulisem, et rakendus suudaks kasutada samaaegselt *Leeway* ja *OpenOil* mudeleid ilma nendevaheliste takistusteta.
- **Skaleeritavus:** Arhitektuur peab olema skaleeritav, võimaldades rakenduse laiendamist ja kohandamist vastavalt vajadustele ning uute mudelite integreerimist tulevikus.
- **Kasutajaliidese paindlikkus:** Arhitektuur peab võimaldama kasutajaliidese kohandamist vastavalt mudelitele ning nende nõuetele, ilma selleta, et peaks muutma olemasolevate rakenduste mooduleid.

Uue arhitektuuri väljatöötamisel kasutas töö autor nii enda kogemust rakenduse prototüübi kasutamises kui ka juhendajalt ja MSI teadlastelt saadud tagasisidet, kes olid eelnevalt samuti prototüüpi kasutanud ning oskasid anda asjakohaseid soovitusi vajalike funktsion-

aalsuste kohta rakenduse edasiarenduses.



Joonis 7. Rakenduse uus arhitektuur.

Uue arhitektuuri põhielemendid hõlmavad järgmist:

- **Mudelite juhtimissüsteem:** Juhtimissüsteem võimaldab kasutajal valida ja käivitada erinevaid mudeleid vastavalt eelistustele ja vajadustele. Uus süsteem on paindlik ja pakub tulevastele arendajatele võimaluse lihtsasti integreerida mitmeid erinevaid mudeleid, mida OpenDrifti raamistik pakub. Selleks on kasutatud models.json mudelite konfiguratsioon.
- **Kasutajaliides:** Kasutajaliides on veebipõhine ning võimaldab kasutajal lihtsalt rakendusega suhelda. Rakenduse arendamise käigus seadistati kasutajaliides vastavalt töö ülesandepüstitusele, et pakkuda kasutajale võimalust valida kahe mudeli vahel (*Leeway* ja *OpenOil*) ning kohandada sisendparameetreid vastavalt vajadustele.

Uus implementeeritud arhitektuur võimaldab arendajatel luua paindliku ja skaleeritava platvormi, mis aitab kasutajal lihtsalt ja efektiivselt analüüsida ujivate objektide ja õlireostuste liikumist Läänemeres. Lisaks võimaldab see integreerida mitmeid erinevaid mudeleid ja pakub kasutajatele paremat kasutuskogemust ning suuremat kontrolli sisendparameetrite üle.

Olulisim arhitektuuri muudatus seisneb selles, et mudelite tüübid ja arv ei ole rakenduses

enam piiratud. Uus kasutaja saab seadistada erinevaid mudeleid vastavalt enda vajadustele ilma selleta, et peaks rakenduse koodis suuri muudatusi tegema.

4. Rakenduse arendamine

Rakenduse arendamises kasutati GitLabi projektide haldussüsteemi ja JetBrains PyCharm IDE-d. Kõikides rakenduse arendamise etappides jälgiti tarkvara avatud-suletud printsiipi, eesmärgiga tagada seda, et teine arendaja saaks rakendusse sisse tuua uut funktsionaalsust ilma olemasolevate moodulite muutmiseta.

4.1 Dockeri seadistamine

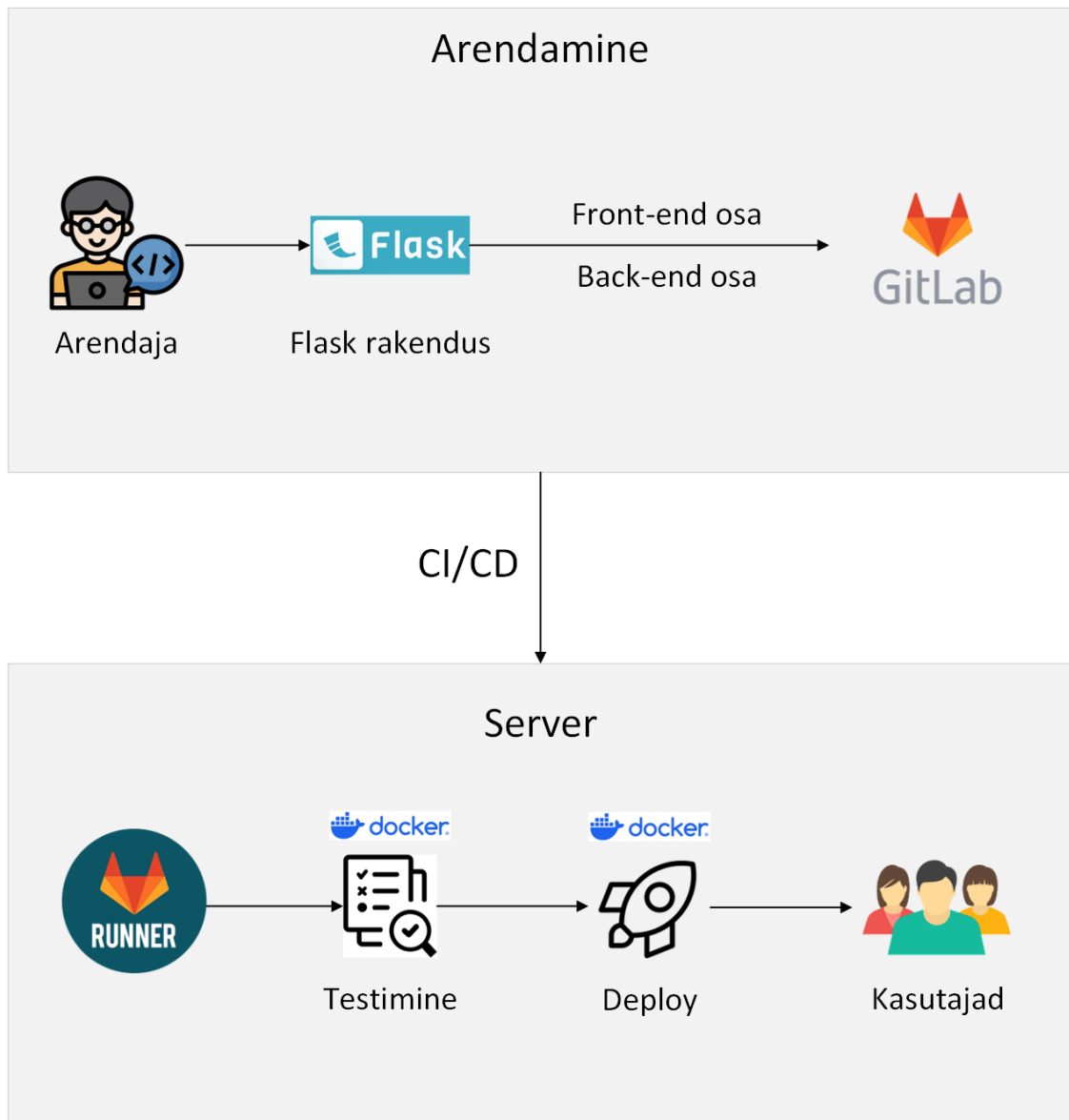
Prototüübi kaustas oli olemas eraldi "Dockerfile", mis võimaldas rakendust käivitada otse "docker" konteineris. Selline lahendus ei olnud mugav, sest iga rakenduse muudatuse korral tuli konteiner peatada, vana "docker pilt" eemaldada ja seejärel uus pilt ehitada. See võtab aga liiga palju aega. Antud probleemi lahendamiseks koostati "docker-compose" fail, mis tagas, et kõik rakenduse "docker pildi" uuendamiseks vajalikke rutiine teeb server automaatselt.

4.2 CI/CD seadistamine

Projekti arendamise ja testimise automatiseerimiseks seadistati CI/CD protsesse, kasutades GitLab Runnerit. See tagas, et rakenduse kõige uuem versioon on alati serveril saadaval. Rakendus pandi üles TalTechi serverisse, kuid vajadusel saab rakendust üle viia ka teise serverisse.

4.2.1 Testimine

CI/CD automatiseerimine hõlmab sealhulgas ka rakenduse testimist, mille käigus GitLab Runner käivitab eraldi "docker" konteineri, kus jooksutatakse Pythonis kirjutatud üksusteste. Antud testide eesmärk on valideerida rakenduse põhimooduli töötamise korraksust, milleks on Flaski *app.py* fail. Selle protsessi käigus testitakse meetodeid, mis vastutavad API päringute eest ning funktsioone, mis vastutavad simulatsioonide arvutamise eest. Juhul, kui vähemalt üks test ei tööta, ei lase GitLab Runner vigase rakenduse versiooni serverisse üles laadida.



Joonis 8. CI/CD skeem.

Joonis 8. näitab, kuidas antud töö rakenduse projektis oli seadistatud CI/CD automatiseerimine. Iga kord, kui arendaja tegi muudatusi rakenduse GitLabi repositooriumis, käivitas see sündmus CI/CD protseduuri. Selle protseduuri käigus jooksub GitLab Runner kahte "docker" konteinerit: esimene rakenduse testimiseks, mille käigus üksustestid valideerisid, et rakenduse põhiosades ei ole vigu tekkinud, ning seejärel teine konteiner, mis käivitas "docker pildi" uuendatud rakendusega ja laadis seda serverisse ülesse.

4.3 Mitmete OpenDrift mudelite kasutamine

Selleks, et kasutajal oleks võimalus kasutada mitut erinevat mudelit (antud töö raames *Leeway* ja *OpenOil* mudeleid), tuli eraldi implementeerida moodul, mis vastutaks selle eest, mis tüüpi mudeli kasutaja on projekti loomisel valinud. See oli ka antud töö peamine eesmärk ning üks keerukamaid väljakutseid.

Igal OpenDrifti mudelil on oma sisend- ja väljundparameetrite hulk. Eranditeks on sellised sisendparameetrid, nagu laiuskraad, pikkuskraad, algus, kestus, raadius ja osakeste arv, mida vajavad OpenDrifti kõik mudelid (Joonis 7). *Leeway* mudel vajab korrektseks tööks sisendparameetrina ujuva objekti tüüpi. *OpenOil* mudel, aga vajab selle asemel õli tüüpi, õli kogust ning võimalust vabastada osakesed merepõhjast. Nende parameetrite kasutamist, omistamist ja mudeli käivitamist tuleb eraldi seadistada konfiguratsioonifailides, mis on igal mudelil erinev. Selliseid konfiguratsioone hoitakse failides “.mustache” laiendiga.

Vana arhitektuuri puhul ei olnud rakenduses kohe algselt võimalik *OpenOil* mudeli kasutamist integreerida. Enne tuli refaktoreerida olemasolev kood, mis vastutas “.mustache” konfiguratsioonifailide haldamise eest.

Lahenduseks oli kasutada eraldi konfiguratsiooni faili – *models.json*, kus kasutaja saaks luua uue mudeli olemasoleva .json struktuuri järgi ja sealhulgas määrata ka mudeli tüübi (Joonis 7). Lisades uue mudeli, kasutaja saab määrata mudeli tüübi kujul “mudeliNimi_mudeliTüüp“. Näiteks, “MSI_leeway” mudeli puhul on tegemist *Leeway* mudeli tüübiga ja “MSI_openoil” mudeli puhul on tegemist *OpenOil* mudeli tüübiga. Sarnase reegli järgi saab kasutaja lisada piiramatu hulk uusi mudeleid, peaasi, et sellist tüüpi mudeleid leidub OpenDrift raamistikus.

Lisaks tekitati mudelite uued konfiguratsioonifailid: *leeway.py.mustache* (Lisa 3) ja *openoil.py.mustache* (Lisa 4). Tänu sellele ei kasuta kõik mudelid enam ühte ja sama konfiguratsiooni faili, vaid kasutaja saab seadistada iga mudeli tüübi eraldi.

4.4 Rakenduse ülevaade

Allpool on näidatud antud valmis rakenduse tavapärase kasutamisuutum.

1. Kasutaja soovib genereerida triivprognoosi Läänemeres tekkinud reostusele. Selleks on tal valik kahest mudelist: *MSI Leeway* (ujuvad objektid/reostused) ja *MSI OpenOil* (õlireostused). Võimalike mudelite valik sõltub sellest, kuidas oli seadis-

tatud models.json konfiguratsiooni fail (Joonis 9).

MSI ADRIFT

Triiviprognosis Eesti merealale



ADRIFT baseerub OpenDrift trajektoormudelil ja kasutab andmeid Eesti operatiivsetest meremudelitest.

Mudel

MSI OpenOil

Uus projekt

Salvestatud projektid

Projekte pole

Joonis 9. Mudelite valiku vaade.

2. Kasutaja valib mudeli *MSI OpenOil* ja vajutab nupule “Uus projekt”. Avaneb projekti seadistamise vaade (Joonis 10). Vastavalt rakenduse uuele arhitektuurile kuvatakse ekraanile üldised sisendparameetrid ning need, mis on spetsiifilised *OpenOil* mudeli jaoks (Joonis 7).

MSI ADRIFT

Triiviprognosis Eesti merelalale



Sisesta uue projekti alguskoht või liiguta märki kaardil

Laiuskraad

59.4443

Pikkuskraad

24.5594

Algus

Apr 16 3:00 AM

Kestus

16 hours

Raadius 400 meetrit



Osakeste arv

2500

Öli kogus (m³)

65

Vabasta osakesed merepõhjast



Algus (UTC) 2024-04-16T00:00:00Z

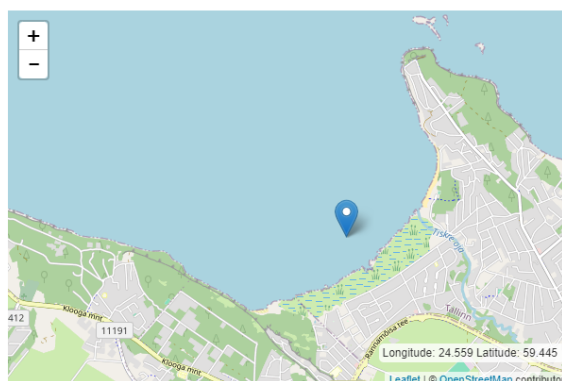
Lõpp (UTC) 2024-04-16T16:00:00Z

Objekt

1 GENERIC BUNKER C

Edasi

Loobu



Joonis 10. Uue triiviprognosisi projekti loomine.

3. Kasutaja seadistab sisendparameetreid. Antud juhul on sisendparameetrite väärtused järgmised: 1) Laiuskraad: 59,4443 – õlireostuse alguskoha laiuskraadi koordinaat. 2) Pikkuskraad: 24,5594 – õlireostuse alguskoha pikkuskraadi koordinaat. 3) Algus:

Apr 16 3.00 AM – õlireostuse alguse kuupäev ja kellaaeg. 4) Kestus: 16 tundi – õlireostuse kestus tundides. 5) Raadius: 400m – õlireostuse alguskoha raadius meetrites. 6) Osakeste arv: 2500 – simulatsiooni osakeste arv. Mida suurem on osakeste arv, seda suurem on simulatsiooni tulemuse resolutsioon, kuid seda rohkem aega võtab simulatsiooni arvutamine. 7) Õli kogus: 65 – mitu kuupmeetrit õli satub vette. 8) Vabasta osakesed merepõhjast: Jah – määrab ära, kas simulatsiooni alguses vabastatakse osakesed merepõhjast. 9) Objekt: GENERIC DIESEL – õli tüüp (Joonis 11).

Triiviproгноos Eesti merelalale

Sisesta uue projekti alguskoht või liiguta märki kaardil

Laiuskraad

59.4443

Pikkuskraad

24.5594

Algus

Apr 16 3:00 AM

Kestus

16 hours

Raadius 400 meetrit



Osakeste arv

2500

Õli kogus (m³)

65

Vabasta osakesed merepõhjast



Algus (UTC) 2024-04-16T00:00:00Z

Lõpp (UTC) 2024-04-16T16:00:00Z

Objekt

1 GENERIC BUNKER C

Edasi

Loobu

Joonis 11. Uue projekti sisendparameetrite väärtused.

4. Kasutaja vajutab nupule “Edasi” ja server hakkab simulatsiooni arvutama. Simulatsiooni tulemusena kuvatakse kasutaja ekraanile kaart, kus on kujutatud osakeste

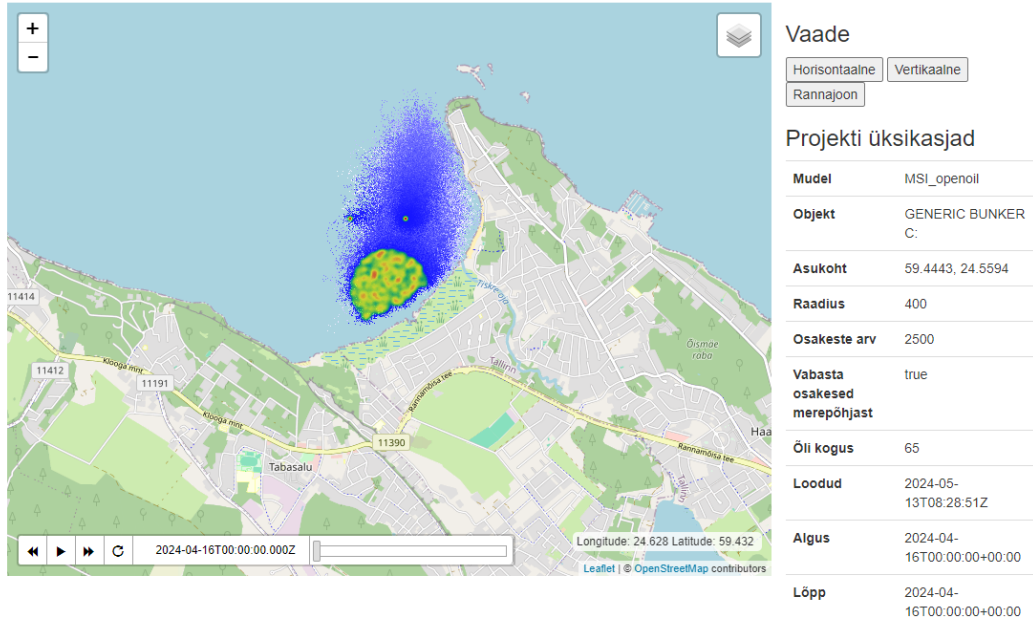
(õlireostuse) liikumise trajektoorid (Joonis 12).

MSI ADRIFT

Triiviprognos Eesti merealale



2024-04-16 00:00 GENERIC BUNKER C



Joonis 12. Õlireostuse simulatsiooni tulemus.

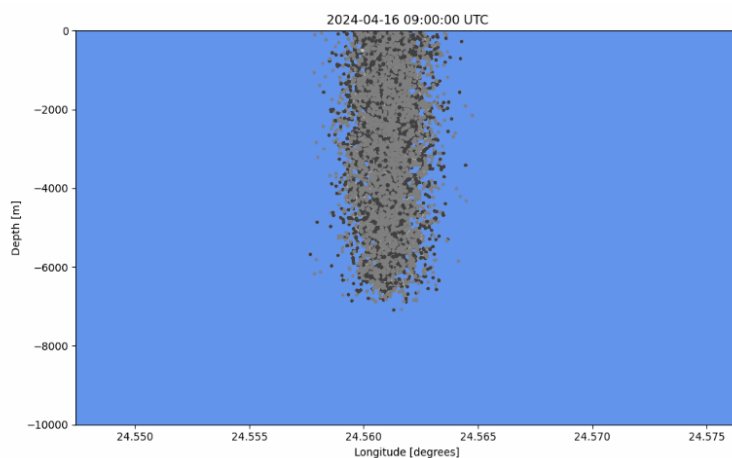
Kuna eelnevalt oli kasutaja teinud “Vabasta osakesed merepõhjust” valiku, siis on tal võimalus vaadata ka õlireostuse liikumise vertikaalset vaadet, vajutades nupule “Vertikaalne“ (Joonis 13).

MSI ADRIFT

Triiviprognosis Eesti merealale



2024-04-16 00:00 GENERIC BUNKER C



Vaade

Projekti üksikasjad

Mudel	MSI_openoil
Objekt	GENERIC BUNKER C:
Asukoht	59.4443, 24.5594
Raadius	400
Osakeste arv	2500
Vabasta osakesed merepõhjast	true
Õli kogus	65
Loodud	2024-05-13T08:28:51Z
Algus	2024-04-16T00:00:00+00:00
Lõpp	2024-04-16T00:00:00+00:00

Joonis 13. Õlireostuse simulatsiooni vertikaalne vaade.

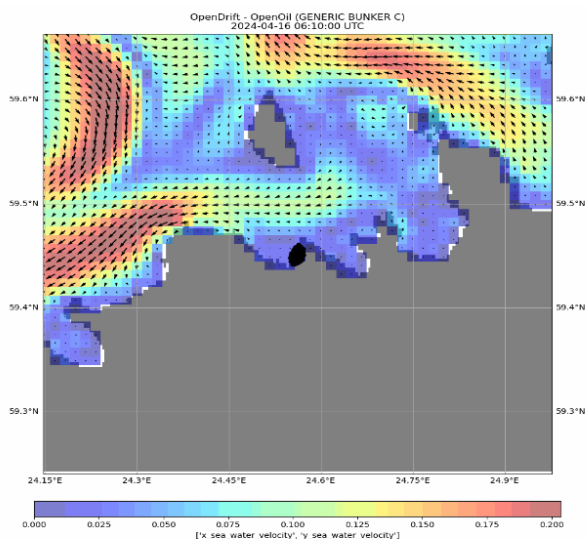
Kasutajal on ka võimalus vaadata simulatsiooni tulemust pealtvaatelt, kus lisaks on kujutatatud ka hoovuste liikumised, vajutades nupule “Rannajoon” (Joonis 14).

MSI ADRIFT

Triiviprognosis Eesti merealale



2024-04-16 00:00 GENERIC BUNKER C



Vaade

Projekti üksikasjad

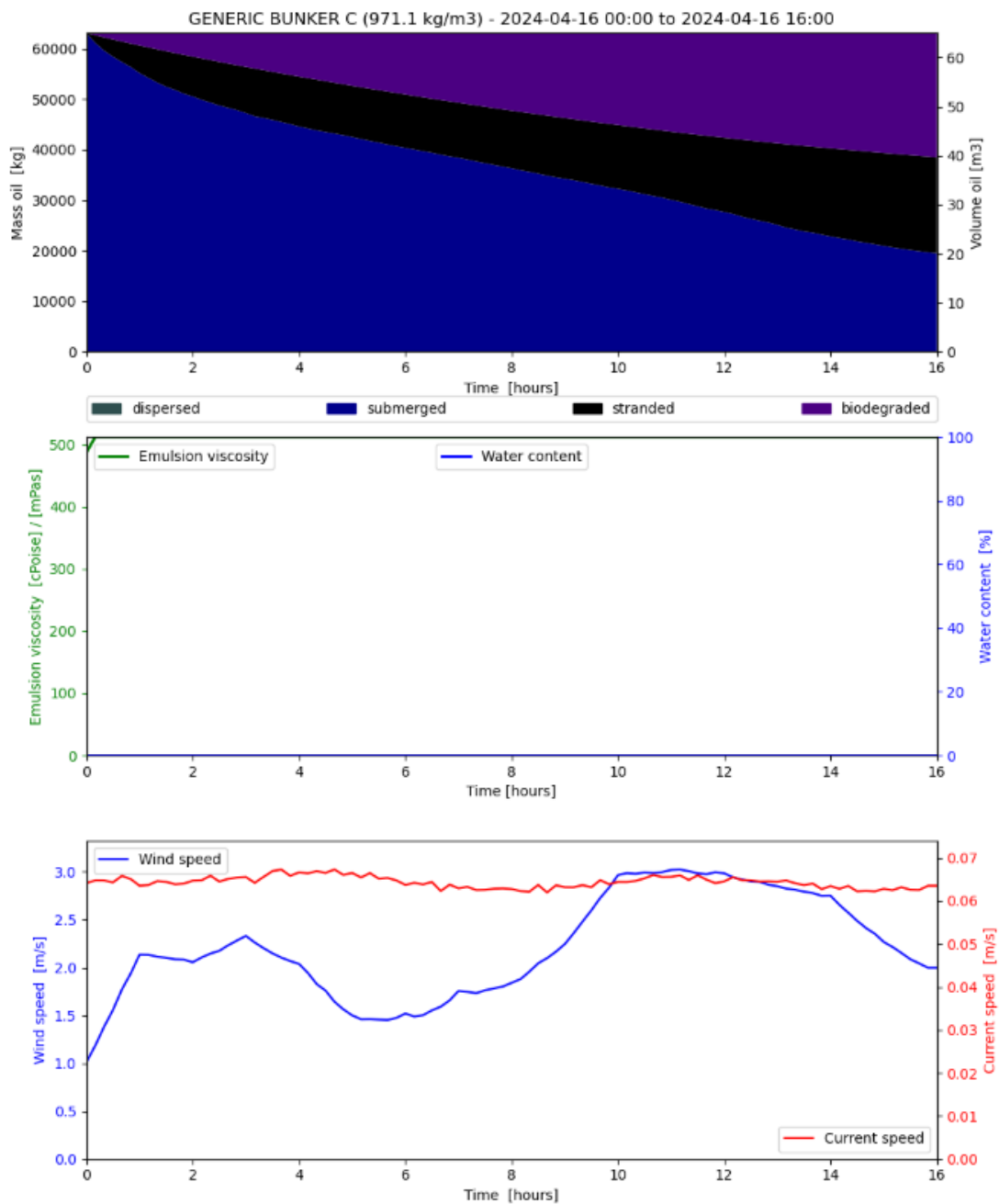
Mudel	MSI_openoil
Objekt	GENERIC BUNKER C:
Asukoht	59.4443, 24.5594
Raadius	400
Osakeste arv	2500
Vabasta osakesed merepõhjast	true
Õli kogus	65
Loodud	2024-05-13T08:28:51Z
Algus	2024-04-16T00:00:00+00:00
Lõpp	2024-04-16T00:00:00+00:00

Joonis 14. Õlireostuse simulatsiooni horisontaalne vaade.

Lisaks on kasutajal võimalus vaadata simulatsiooni graafikuid, mis kirjeldavad õli muutumist aja jooksul (Joonis 15), kus:

- Antud juhul ülemine graafik näitab, kui palju õli hajus, kui palju on sattunud veesambasse, kui palju jõudis rannikule ning kui palju sai lagundatud.
- Keskmine graafik näitab emulsiooni viskoossust ja vee kogust õli koguse sees.
- Alumine graafik näitab, kuidas on muutunud hoovuste ja tuulte kiirused simulatsiooni jooksul.

Graafikud



Joonis 15. Simulatsiooni graafikud.

4.5 Leitud vigade ja ebamugavuste parandamine

Rakenduse prototüübis leidsid tõsised ja vähem tõsised kasutamiseiga seotud ebamugavused ja muud vead, mille iseloomu ja parandmist antud peatükis kirjeldatakse.

4.5.1 Genereeritud piltide ja GIF-ide salvestamine

OpenDrift pakub mugavat liidest simulatsioonide tulemuste piltide ja GIF-ide kujul esitamiseks, kuid rakenduse prototüübis seda funktsionaalsust ei rakendatud.

Antud probleemi lahendamiseks tuli teha muudatusi otse mudelite „mustache“ konfiguratsioonifailides, kuhu olid lisatud meetodid simulatsiooni tulemuste genereerimiseks GIF-ide kujul. Lisaks laiendati ka OpenDrifti mudelite meetodeid, mis vastutavad tulemuste graafikute genereerimise eest. Selleks, et neid graafikuid saaks kasutaja veebilehel näha, kirjutati eraldi API käsud, mis tagastasid graafikuid, pildi kujul, konkreetse projekti jaoks.

4.5.2 API päringud ja lehekülgede “props” atribuudid

Rakenduse testimisel selgus, et prototüübi kliendipoolsete API päringute sees esines vigu, mis ei võimaldanud serveril õigesti töödelda kliendilt saadud käsked. Nendes päringutes ja nendega seotud veebilehtedel olid valesti seadistatud nn veebilehtede “props” atribuudid, mida alati edastatakse koos päringutega. Antud vea puhul ei olnud mõnedel GET päringutel kaasa antud mudeli nime, mis ei võimaldanud seda kasutada otse veebilehtedel. Arendamise käigus sai see parandatud.

4.5.3 Lähtekoodis fikseeritud konstandid

Esialgse *Leeway* mudeli “.mustache” konfiguratsioonis leidsid muutujate ja mudelite sisendparameetrite fikseeritud konstandid, mis esinesid koodis peamiselt mudelite konfiguratsioonide failis arusaamatute nimede all ning ilma selgitusteta. Seda tüüpi muutujate kasutamine muudab nende tähenduse ja eesmärgi mõistmise teistele programmeerijatele keeruliseks.

Lisaks ei olnud kasutajal võimalust muuta simulatsiooni kestust, osakeste arvu ja reostuse raadiust. Kuigi need parameetrid olid veebilehel nähtavad, ei mõjutanud nad simulatsiooni tulemust. Selle vea parandamiseks tuli samuti muuta olemasolevaid API päringuid ja nendega seotud veebilehtede “props” atribuute, kuhu lisati uued väljad, mis saavad edastada

vastavaid andmeid serverile.

5. Tulemuste valideerimine

5.1 Rakenduse tulemuste korrektsus

Rakenduse triivipronoosi tulemusi võib pidada korrektseteks, kuna enne rakenduse edasiarendamisega alustamist valideeriti OpenDrift Eesti mereala jaoks, kasutades kohalikke keskkonnamudeleid [25].

5.2 Kasutajamugavus ja andmete ülevaatlikus

Rakenduse kasutajamugavust ning uuenenud informatiivsust testisid uurimisgrupi liikmed, kes olid seotud esialgse prototüübi seadistamisega Eesti merealale ja kes on ka edasiarendatud rakenduse potentsiaalsed kasutajad. Tagasisidet kasutajamugavuse kohta anti töö autorile arendusprotsessi käigus jooksvalt, mille alusel tegi töö autor rakenduses vastavaid parandusi.

Rakendusest varem puudunud *OpenOil* mudel annab hea ülevaate potentsiaalse reostuse liikumise ning leviku kohta. Lisatud võimalus seadistada sisendparameetreid, nagu "osakeste arv", "õli kogus" ja "osakeste vabastamine merepõhjast", võimaldab kasutajal anda mudelisse täpsemaid andmeid ning saada reostuse liikumise ja muutuste kohta täpsemat informatsiooni. Näiteks võimaldab osakeste vabastamine merepõhjast uurida laevavrakkidest pärinevat õlireostust. Lisaks pakub laiendatud kasutajaliides võimalust mugavalt näha õli omaduste ja oleku muutuste graafikuid ning võimalust vaadata hoovuste liikumismustreid reostuse piirkonnas.

Võimalus hõlpsasti uusi OpenDrifti mudeleid rakendusse lisada on kasulik ka kasutajatele, kes ei ole informaatika valdkonnaga hästi kursis.

5.3 Rakenduse testimine erinevates kasutusjuhtumites

Antud peatükis mõõdetakse valmis rakenduse triivipronoosi simulatsioonide arvutamise kiirust. Esialgu testitakse *MSI Leeway* mudelit ja seejärel *MSI OpenOil* mudelit. Iga järgnev tabel näitab, kuidas iga mudeli sisendparameeter eraldi mõjutab mudeli simulatsiooni aega.

Mõlema mudeli katsetamisel kasutati sisendparameetriteks järgmiseid konstantseid väärtusi: laiuskraad 59,5318, pikkuskraad 24,6712, kestus 12 tundi.

Tabel 2. MSI Leeway mudeli testimine erinevate reaostuste raadiustega.

Raadius (m)	Osakeste arv	Objekt	Simulatsiooni aeg (sekund)
250	1000	Sailboat Mono-hull (Average)	13,70
500	1000	Sailboat Mono-hull (Average)	12,56
750	1000	Sailboat Mono-hull (Average)	13,00
1000	1000	Sailboat Mono-hull (Average)	13,80

Tabel 3. MSI Leeway mudeli testimine erinevate osakeste arvuga.

Raadius (m)	Osakeste arv	Objekt	Simulatsiooni aeg (sekund)
250	5000	Sailboat Mono-hull (Average)	30,45
250	10000	Sailboat Mono-hull (Average)	51,56
250	15000	Sailboat Mono-hull (Average)	70,04
250	20000	Sailboat Mono-hull (Average)	102,29

Tabel 4. MSI Leeway mudeli testimine erinevate objektidega.

Raadius (m)	Osakeste arv	Objekt	Simulatsiooni aeg (sekund)
250	1000	Person in Water, survival suit (face up)	12,14
250	1000	Life Raft, 4-14 person capacity (average)	12,26

Jätkub...

Tabel 4 – *Jätukub...*

Raadius (m)	Osakeste arv	Objekt	Simulatsiooni aeg (sekund)
250	1000	Life Raft, 15-50 person capacity (heavy loading)	14,22
250	1000	Medical waster (mean values)	10,74
250	1000	Medical waster (syringes, large)	12,46

Tabel 5. *MSI OpenOil mudeli testimine erinevate õlireaostuste raadiustega.*

Raadius (m)	Osakeste arv	Õli kogus (m³)	Õli	Simulatsiooni aeg (sekund)
250	1000	10	GENERIC DIESEL	38,00
500	1000	10	GENERIC DIESEL	39,12
750	1000	10	GENERIC DIESEL	39,44
1000	1000	10	GENERIC DIESEL	40,01

Tabel 6. *MSI OpenOil mudeli testimine erinevate osakeste arvuga.*

Raadius (m)	Osakeste arv	Õli kogus (m³)	Õli	Simulatsiooni aeg (sekund)
250	5000	10	GENERIC DIESEL	68,92
250	10000	10	GENERIC DIESEL	105,10
250	15000	10	GENERIC DIESEL	145,92
250	20000	10	GENERIC DIESEL	176,81

Tabel 7. *MSI OpenOil* mudeli testimine erinevate õli kogustega.

Raadius (m)	Osakeste arv	Õli kogus (m ³)	Õli	Simulatsiooni aeg (sekund)
250	1000	50	GENERIC DIESEL	38,07
250	1000	150	GENERIC DIESEL	39,73
250	1000	250	GENERIC DIESEL	38,46
250	1000	350	GENERIC DIESEL	38,77

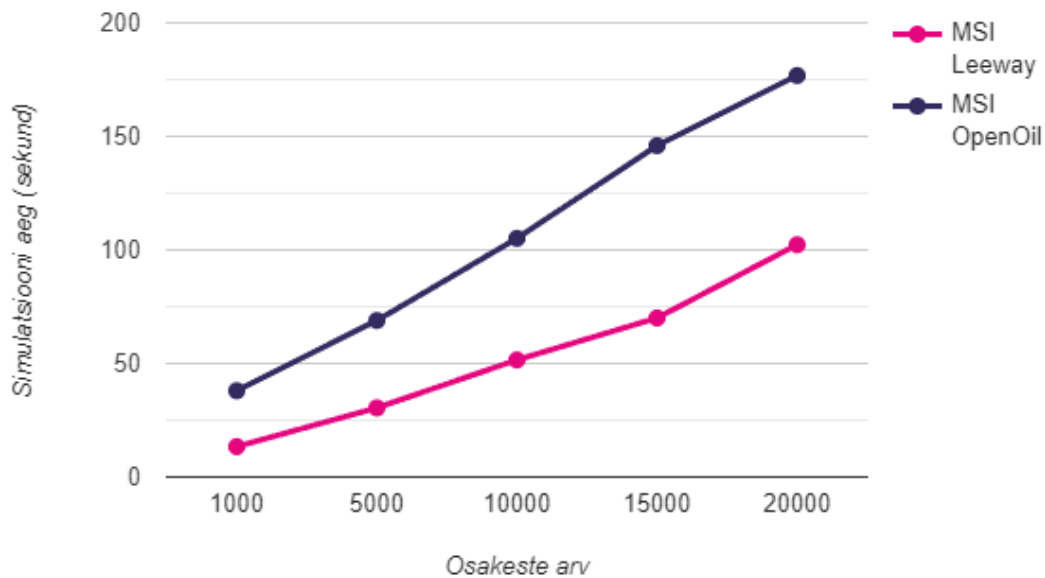
Tabel 8. *MSI OpenOil* mudeli testimine erinevate õli tüübiga.

Raadius (m)	Osakeste arv	Õli kogus (m ³)	Õli tüüp	Simulatsiooni aeg (sekund)
250	1000	10	ABU SAFAH, ARAMCO	38,42
250	1000	10	ARABIAN HEAVY, ARAMCO	40,98
250	1000	10	BUNKER C FUEL OIL	39,82
250	1000	10	DIESEL FUEL OIL (Alaska)	39,11

Tabelitest 2 ja 4 selgub, et *MSI Leeway* mudeli simulatsiooni kiirust ei mõjuta raadiuse suurus ega valitud ujuva objekti tüüp. Tabelitest 5, 7 ja 8 selgub, et *MSI OpenOil* mudeli simulatsiooni kiirust ei mõjuta raadiuse suurus, õli kogus ega valitud õli tüüp. Nende kõikide sisendparameetrite puhul jäävad mudelite simulatsiooni ajad üldjuhul samaks.

Tabelitest 3 ja 6 aga on näha, et nii *MSI Leeway* kui ka *MSI OpenOil* mudelite simulatsioonide kiirust mõjutab tõsiselt sisendparameetri "osakeste arv" väärtus.

Joonis 16 näitab, et antud juhul on *MSI Leeway* ja *MSI OpenOil* mudelite simulatsioonide kiirustel lineaarne sõltuvus "osakeste arv" sisenparameetrist. Peale selle on näha, et mudeli *MSI OpenOil* puhul võtab simulatsiooni arvutamine samade sisendparameetritega keskmiselt 25 sekundit rohkem aega võrreldes *MSI Leeway* mudeliga.



Joonis 16. MSI Leeway ja MSI OpenOil mudelite simulatsioonide kiiruse sõltuvus "osakeste arv" sisendparameetrist.

Saadud testimiste tulemustest lähtuvalt saab öelda, et rakenduse triivprognooosi simulatsioonide arvutamise aeg on aktsepteeritav mõlema mudeli puhul. Mõlemad mudelid annavad piisavalt ülevaadet reostuste triivi kohta kasutades 2000 osakesest, seega üldjuhul on soovitatav kasutada just sellise osakeste arvu. Täpsemate simulatsioonide genereerimiseks saab kasutada ka suurema osakeste arvu, kuid tuleb arvestada, et simulatsiooni aeg samuti hakkab suurenema.

6. Edasiarendamise võimalused

Juba praegu antud rakendus pakub kasutajale võimalust prognoosida ujuvate objektide ja õlireostuste liikumisi Läänemerel, kasutades selleks seadistatud mudeleid *MSI Leeway* ja *MSI OpenOil*. Siiski on mitmeid suundi, kuidas rakendust veelgi täiustada ja funktsionaalsust laiendada.

Üks oluline edasiarendamise võimalus on lisada rakendusse teisi OpenDrifti mudeleid, mis võimaldaksid kasutajal teha mitmekesisemaid prognoose. Näiteks, võiks implementeerida *ChemicalDrift* või *PlastDrift* mudelite kasutamist, et prognoosida keemiliste ainete või mikroplastiku reostuste liikumist ja arengut Läänemerel vastavalt. Tänu rakenduse uuele modulaarsele arhitektuurile oleksid sellised uuendused lihtsasti integreeritavad.

Teine võimalus on kasutajaliidese täiustamine, sealhulgas kasutusmugavuse parandamine, interaktiivsete elementide lisamine, graafika ja visualiseerimise integreerimine ning kasutajakogemuse optimeerimine erinevate seadmete ja platvormide jaoks.

Need on vaid mõned võimalikud suunad, kuidas rakendust edasi arendada. Kuna valmis rakendus on avatud lähtekoodiga ning asub avatud GitHub repositooriumis, siis iga arendaja saab soovi korral hakata antud rakendust edasi arendama vastavalt vajadustele.

7. Kokkuvõte

Käesoleva töö eesmärk oli arendada edasi rakendust, mis võimaldab prognoosida ujuvate objektide ja õlireostuste liikumist Läänemerel, kasutades selleks osakeste liikumistrajektoride modelleerimiseks mõeldud OpenDrift raamistikku ning selle mudeleid - *Leeway* ja *OpenOil*. Probleem seisnes vajaduses luua tööriist, mis pakuks kasutajale võimalust efektiivselt simuleerida erinevaid stsenaariume ning anda kasutajatele ülevaatlikku informatsiooni ujuvate objektide ja õlireostuste liikumise kohta Läänemeres, täpsemalt Eesti merealal, kasutades selleks kohalikke keskkonnamudeleid.

Eesmärkide saavutamiseks viidi esiteks läbi olemasoleva prototüübi põhjalik analüüs, mille käigus selgus, et prototüübi arhitektuur ei sobinud mitme mudeli samaaegseks kasutamiseks. Seetõttu töötati välja rakenduse uus arhitektuur, mis on skaleeritavam ja modulaarsem. Tänu uuele arhitektuurile saavad teised arendajad lisada rakendusse uusi OpenDrifti mudeleid ilma rakenduse olemasolevate moodulite muutmiseta. Lisaks parandati järgnenud arenduse käigus prototüübis esinevaid vigu ja muid ebamugavusi.

Töö tulemusena saadi valmis rakendus, mis suudab efektiivselt genereerida ujuvate objektide ja õlireostuste triivprognoosi Eesti merealal. Rakenduse arendamise käigus õpiti uusi veebitehnoloogiad, mitmeid aspekte merekeskkonna dünaamika ja reostuste leviku kohta, samuti rakenduse arendamise ja testimise protsesside kohta.

Peale selle, viidi läbi valmis rakenduse testimine erinevates kasutusjuhtumites. Selle käigus valideeriti rakenduse simulatsioonide arvutamise kiiruse sõltuvust mudelite erinevate sisendparameetrite väärtustest. Testimise tulemusena jõuti järelduseni, et üldjuhul võtab *MSI OpenOil* mudelil simulatsiooni arvutamine keskimiselt 25 sekundid rohkem aega võrreldes *MSI Leeway* mudeliga. Samuti selgus, et simulatsiooni osakeste arvu suurenemisega simulatsiooni aeg kasvab lineaarselt mõlemate mudelite puhul.

Antud töö võib olla kasulik mitmes valdkonnas. Näiteks merekeskkonna jälgimise ja kaitse valdkonnas, kus selline rakendus võib aidata paremini mõista ujuvate objektide ja reostuste arengut ning võtta vastu paremaid otsuseid reageerimisel erinevates olukordades. Samuti võib see olla kasulik mereõnnetuste ennetamise ja mereohutuse tagamise valdkonnas, kus selline prognoosimise võimekus võib aidata vähendada riske päästeoperatsioonides. Lisaks võib antud töö olla kasulik ka teadlastele ja õppeasutustele merekeskkonna dünaamika ja reostuste leviku uurimisel ning uute meetodite ja triivprognoosi rakenduste arendamisel

selles valdkonnas.

Kasutatud kirjandus

- [1] HELCOM. *SeaTrackWeb Oil Drift Modeling*. [Kasutatud: 05-05-2024]. URL: <https://helcom.fi/action-areas/response-to-spills/helcom-seatracweb-and-oil-drift-modeling/>.
- [2] ilmateenistus.ee. *NEMO-EST05 hoovuste mudel*. [Kasutatud: 04-05-2024]. URL: <https://www.ilmateenistus.ee/meri/mereprognoosid/hoovused/#layers/current>.
- [3] ilmateenistus.ee. *SWAN-EST lainete mudel*. [Kasutatud: 04-05-2024]. URL: <https://www.ilmateenistus.ee/meri/mereprognoosid/oluline-lainekorgus-ja-suund/#layers/waveheight>.
- [4] OpenDrift. *Introduction to OpenDrift*. [Kasutatud: 18-04-2024]. URL: <https://opendrift.github.io/>.
- [5] Irish Marine Institute. *ADRIFT rakenduse prototüübi GitHub repositoorium*. [Kasutatud: 12-02-2024]. URL: <https://github.com/IrishMarineInstitute/adrift/tree/master>.
- [6] B. Gustafsson. *Interaction between Baltic Sea and North Sea*. [Kasutatud: 03-05-2024]. 1997. URL: <https://link.springer.com/article/10.1007/BF02764031>.
- [7] P. Winsor, J. Rodhe, and A. Omstedt. *Baltic Sea ocean climate: an analysis of 100 yr of hydrographic data with focus on the freshwater budget*. [Kasutatud: 02-05-2024]. 2001. URL: <https://www.int-res.com/abstracts/cr/v18/n1-2/p5-15/>.
- [8] HELCOM. *The Baltic Marine Environment 1999–2002*. [Kasutatud: 27-03-2024]. 2003. URL: <https://helcom.fi/wp-content/uploads/2019/10/BSEP87.pdf>.
- [9] P. Keramea, K. Spanoudaki, and G. Zodiatis. *Oil Spill Modeling: A Critical Review on Current Trends, Perspectives, and Challenges*. [Kasutatud: 05-05-2024]. 2021. URL: <https://www.mdpi.com/2077-1312/9/2/181>.
- [10] HELCOM. *Helcom Metadata Catalogue - Detected Spills of Mineral Oil and Other Substances*. [Kasutatud: 02-05-2024]. URL: <https://metadata.helcom.fi/geonetwork/srv/eng/catalog.search#/metadata/345c9b95-6e9c-44a4-b02a-ee4304cccffc>.

- [11] NAS. *Oil in the Sea IV*. [Kasutatud: 08-05-2024]. 2022. URL: <https://nap.nationalacademies.org/catalog/26410/oil-in-the-sea-iv-inputs-fates-and-effects>.
- [12] J. Chen and W. Zhang. *Oil spills from global tankers: Status review and future governance*. [Kasutatud: 06-05-2024]. 2019. URL: <https://www.sciencedirect.com/science/article/abs/pii/S095965261931090X>.
- [13] J. Michel, T. Gilbert, and D. S. Etkin. *An Issue Paper Prepared for the 2005 International Oil Spill Conference: Potentially Polluting Wrecks in Marine Waters*. [Kasutatud: 06-05-2024]. 2005. URL: <https://meridian.allenpress.com/iosc/article/2005/1/1/138397/AN-ISSUE-PAPER-PREPARED-FOR-THE-2005-INTERNATIONAL>.
- [14] H. Landquist. *Evaluating the needs of risk assessment methods of potentially polluting shipwrecks*. [Kasutatud: 07-05-2024]. 2013. URL: <https://www.sciencedirect.com/science/article/pii/S0301479713000066>.
- [15] H. Landquist. *A fault tree model to assess probability of contaminant discharge from shipwrecks*. [Kasutatud: 07-05-2024]. 2014. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0025326X1400589X>.
- [16] HELCOM. *Maritime Activities in the Baltic Sea*. [Kasutatud: 27-03-2024]. 2018. URL: <https://www.helcom.fi/wp-content/uploads/2019/08/BSEP152-1.pdf>.
- [17] H. Landquist et al. *VRAKA—A Probabilistic Risk Assessment Method for Potentially Polluting Shipwrecks*. [Kasutatud: 05-05-2024]. 2016. URL: <https://www.frontiersin.org/articles/10.3389/fenvs.2016.00049/full>.
- [18] M. L. Spaulding. *State of the art review and future directions in oil spill modeling*. [Kasutatud: 11-05-2024]. 2017. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0025326X17300012>.
- [19] L. Hole. *Report on Existing Oil Spill Response Models*. [Kasutatud: 11-05-2024]. 2018. URL: https://app4sea.interreg-npa.eu/subsites/c4ca/DT3.2.1_Report_on_existing_oil_spill_response_models.pdf.
- [20] J. W. Till, I. Eisenman, and M. Ceroli. *How Winds and Ocean Currents Influence the Drift of Floating Objects*. [Kasutatud: 06-05-2024]. URL: <https://journals.ametsoc.org/view/journals/phoc/52/5/JPO-D-20-0275.1.xml>.

- [21] HELCOM. *Annual report on discharges observed during aerial surveillance in the Baltic Sea*. [Kasutatud: 27-03-2024]. 2022. URL: <https://helcom.fi/wp-content/uploads/2023/11/Aerial-surveillance-report-2022.pdf>.
- [22] Free Software Foundation. *GNU General Public License, version 2*. [Kasutatud: 05-05-2024]. URL: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>.
- [23] OpenDrift. *OpenDrift andmemudel*. [Kasutatud: 18-04-2024]. URL: https://opendrift.github.io/theory/data_model.html.
- [24] OpenDrift. *OpenDrift mudelid*. [Kasutatud: 18-04-2024]. URL: https://opendrift.github.io/choosing_a_model.html.
- [25] S. Pärt et al. *An ocean-wave-trajectory forecasting system for the eastern Baltic Sea: Validation against drifting buoys and implementation for oil spill modeling*. [Kasutatud: 04-05-2024]. 2023. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X23009311>.
- [26] netCDF. *Package netCDF4*. [Kasutatud: 06-05-2024]. URL: <https://unidata.github.io/netcdf4-python/>.
- [27] Pallets. *Flask*. [Kasutatud: 24-04-2024]. URL: <https://flask.palletsprojects.com/en/3.0.x/#user-s-guide>.
- [28] Pallets. *Werkzeug dokumentatsioon*. [Kasutatud: 27-04-2024]. URL: <https://werkzeug.palletsprojects.com/en/3.0.x/>.
- [29] Pallets. *Jinja dokumentatsioon*. [Kasutatud: 27-04-2024]. URL: <https://jinja.palletsprojects.com/en/3.1.x/>.
- [30] Pallets. *Click dokumentatsioon*. [Kasutatud: 27-04-2024]. URL: <https://click.palletsprojects.com/en/8.1.x/>.
- [31] W3 Schools. *AJAX dokumentatsioon*. [Kasutatud: 27-04-2024]. URL: https://www.w3schools.com/js/js_ajax_intro.asp.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Alan Parik

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Ujuvate objektide ja õlireostuste triivprognoosi süsteemi arendamine Eesti mereala jaoks”, mille juhendaja on Siim Pärt and Evelin Halling
 - 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. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

27.05.2024

¹Lihtlitsents 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 loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – models.json mudelite konfiguratsiooni faili näide

```
{
  "MSI_leeway": {
    "id": "MSI_leeway",
    "name": "MSI Leeway",
    "opendap_url": "./mat_test_nemo31.nc",
    "wind_url": "./mat_test_nemo31.nc",
    "defaults": {
      "latitude": 59.5318,
      "longitude": 24.6712,
      "northwest_lat": 60.725,
      "northwest_lon": 21.55,
      "southeast_lat": 56.941666666667,
      "southeast_lon": 30.35
    },
    "drifters": ["PIW-1", "PIW-6", "LIFE-RAFT-DB-10", "PERSON-POWERED-VESSEL-1", "PERSON-POWERED-VESSEL-2", "PERSON-POWERED-VESSEL-3", "FISHING-VESSEL-1", "SAILBOAT-1", "SAILBOAT-2", "OIL-DRUM", "CONTAINER-1", "SLDMB"],
    "variables": {
      "latitude": "lat",
      "longitude": "lon",
      "time": "time"
    }
  },
  "MSI_openoil": {
    "id": "MSI_openoil",
    "name": "MSI Openoil",
    "opendap_url": "./mat_test_nemo31.nc",
    "wind_url": "./mat_test_nemo31.nc",
    "defaults": {
      "latitude": 59.5318,
      "longitude": 24.6712,
      "northwest_lat": 60.725,
      "northwest_lon": 21.55,
      "southeast_lat": 56.941666666667,
      "southeast_lon": 30.35
    },
  },
}
```



```
"drifters": ["GULLFAKS, EXXON", "ARABIAN MEDIUM, API", "
  ALGERIAN CONDENSATE, STATOIL"],
"variables": {
  "latitude": "lat",
  "longitude": "lon",
  "time": "time"
}
}
```

Lisa 3 – MSI Leeway mudeli .mustache konfiguratsiooni näide

```
from opendrift.models.leeway import Leeway
from opendrift.readers import reader_netCDF_CF_generic
from opendrift.readers import reader_ROMS_native
from datetime import datetime, timedelta
import dateutil.parser

filename = '{{nc_output_path}}'

longitude = {{longitude}}
latitude = {{latitude}}

N = {{particles_amount}}

start_time = datetime('{{start_time_datetime}}')
end_time = datetime('{{end_time_datetime}}')

R = {{radius}}

model = Leeway(loglevel=0)
model.set_config('general:use_auto_landmask', True)

phys = reader_netCDF_CF_generic.Reader('{{opendap_url}}',
standard_name_mapping={'SSU': 'x_sea_water_velocity', 'SSV': 'y_sea_water_velocity'}
})

wind = reader_netCDF_CF_generic.Reader('{{wind_url}}',
standard_name_mapping={'u10': 'x_wind', 'v10': 'y_wind'}
})

model.add_reader([phys, wind])

object_type = {{object_type}} # {{object_type_key}} {{object_type_description}}

model.seed_elements(lon=longitude,
```

```
        lat=latitude ,
        number=N,
        radius=R,
        radius_type="uniform " ,
        time=start_time ,
        object_type=object_type
    )

model.run(end_time=end_time ,
        time_step=timedelta ( minutes=10) ,
        time_step_output=timedelta ( minutes=10) ,
        outfile=filename ,
        export_variables=[" trajectory " ,
            " time " ,
            " status " ,
            " age_seconds " ,
            " lon " ,
            " lat " ]
    )
```

Lisa 4 – MSI OpenOil mudeli .mustache konfiguratsiooni näide

```
from datetime import datetime, timedelta
from opendrift.readers import reader_netCDF_CF_generic
from opendrift.readers import reader_ROMS_native
from opendrift.models.openoil import OpenOil
import dateutil.parser
import os
import numpy as np

filename = '{{nc_output_path}}'

os.makedirs(f'{filename[:-10]}/plots ')

longitude = {{longitude}}
latitude = {{latitude}}
N = {{particles_amount}}
oil_amount = {{oil_amount}}
is_free_particles = {{is_free_particles}}
start_time = datetime('{{start_time_datetime}}')
end_time = datetime('{{end_time_datetime}}')
R = {{radius}}
depth = 0
duration = "{{duration}}"
duration_days = int(duration.split(" ")[0])
duration_hours = int(duration.split(" ")[2])
duration_minutes = int(duration.split(" ")[4])

model = OpenOil(loglevel=20, weathering_model='noaa')
model.set_config('general:use_auto_landmask', True)
model.set_config('processes:biodegradation', True)
model.set_config('biodegradation:method', 'half_time')

model.set_config('drift:vertical_mixing', True)

if is_free_particles == True:
    depth = 'seafloor'
    model.disable_vertical_motion()
```

```

phys = reader_netCDF_CF_generic.Reader('{{ opendap_url }}',
standard_name_mapping={'SSU': 'x_sea_water_velocity', 'SSV': '
    y_sea_water_velocity'
})

wind = reader_netCDF_CF_generic.Reader('{{ wind_url }}',
standard_name_mapping={'u10': 'x_wind', 'v10': 'y_wind'
})

custom_reader = reader_netCDF_CF_generic.Reader('{{ wind_url }}',
    standard_name_mapping={'SSU': 'x_sea_water_velocity', 'SSV': '
    y_sea_water_velocity', 'u10': 'x_wind', 'v10': 'y_wind'})

model.add_reader(custom_reader)

object_type = "{{ object_type_key }}"

model.seed_elements(lon=longitude,
                    lat=latitude,
                    z=depth,
                    number=N,
                    radius=R,
                    radius_type="uniform",
                    time=start_time,
                    biodegradation_half_time_slick=3,
                    biodegradation_half_time_droplet=1,
                    oil_type=object_type,
                    m3_per_hour=oil_amount,
                    diameter=1e-5
                )

model.run(duration=timedelta(hours=duration_hours),
          time_step=600,
          outfile=filename,
          export_variables=["trajectory",
                           "time",
                           "status",
                           "age_seconds",
                           "lon",
                           "lat",

```

```

        "z",
        "mass_oil",
        "density",
        "mass_evaporated",
        "mass_dispersed",
        "mass_biodegraded",
        "viscosity",
        "water_fraction",
        "x_wind",
        "y_wind",
        "x_sea_water_velocity",
        "y_sea_water_velocity",
        "sea_surface_wave_significant_height
    "
    ]
)

model.plot_oil_budget(filename=f"{filename[:-10]}/plots/
oil_budget")

if depth == 'seafloor':
    model.animation_profile(filename=f"{filename[:-10]}/plots/
vertical_movement.gif")
    print("vertical movement animation saved")

model.animation(fast=True, background=['x_sea_water_velocity', '
y_sea_water_velocity'], filename=f"{filename[:-10]}/plots/
coastline.gif")

```