

Određivanje pomorskih plovidbenih koridora zasnovano na statističkom učenju na zapisima automatskog sustava za identifikaciju plovila

Gamulin, Mihaela

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:264833>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-05**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Mihaela Gamulin

**ODREĐIVANJE POMORSKIH
PLOVIDBENIH KORIDORA
ZASNOVANO NA STATISTIČKOM
UČENJU NA ZAPISIMA
AUTOMATSKOG SUSTAVA ZA
IDENTIFIKACIJU PLOVILA**

Diplomski rad

Voditelji rada:
prof. dr. sc. Luka Grubišić
prof. dr. sc. Renato Filjar

Zagreb, veljača, 2020

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Kako znaš da je nešto nemoguće ako ne probaš?!

Sadržaj

Sadržaj	iv
Uvod	2
1 Automatski identifikacijski sustav	3
1.1 Uvod	3
1.2 Princip rada	5
1.2.1 Način komunikacije	5
1.2.2 Vrste AIS uređaja	7
1.3 AIS poruke	8
1.3.1 Sadržaj AIS poruke	8
1.3.2 Prikaz AIS podataka	9
2 Klasifikacija metoda strojnog učenja	11
2.1 Strojno učenje	11
2.2 Nenadgledano učenje	13
2.3 Inkrementalno učenje	13
2.4 Klasteriranje i algoritam DBSCAN	15
2.4.1 Algoritam klasteriranja zasnovan na gustoći opažanja sa šumom	17
2.4.2 Inkrementalni DBSCAN algoritam	22
3 Metoda izdvajanja prometnih ruta i detekcija anomalija	24
3.1 Prethodna istraživanja	24
3.2 Opis	25
3.3 Prometni model i otkrivanje prometnih spoznaja	26
3.3.1 Struktura metode TREAD	27
3.4 Primjena spoznaja	33
3.4.1 Klasifikacija ruta	33
3.4.2 Predviđanje ruta	42
3.4.3 Detekcija anomalija	42

3.5	Razmatranje obilježja TREAD metode	42
4	Programska izvedba	44
4.1	Uvod u R	44
4.2	Ulazni podaci	45
4.2.1	Struktura podataka	45
4.2.2	Pregled podataka	46
4.3	Rezultati izvedbe	51
5	Zaključak	59
	Bibliografija	60
A	Kod izvedbe nenadgledanog izdvajanja putnih točaka i ruta	66
B	Kod izvedbe primjene	77
C	Kod izvedbe grafičkih prikaza	86

Uvod

Razumijevanje uzoraka ponašanja plovila u operativnom radu ključno je za razvijanje svijesti o situacijama u pomorskom prometu, posebice za klasificiranje i predviđanje aktivnosti. Razvojem automatskog identifikacijskog sustava (eng. Automatic Identification System, AIS), omogućeno je praćenje putanja plovila, kao ulaza za postupke identifikacije uobičajenih obrazaca ponašanja, definicije koridora, detekcije neuobičajenih manevara koji mogu ukazivati na kritične situacije (gusarstvo, terorizam, krijumčarenje, problemi s brodom ili posadom), optimizaciju upravljanja plovidbenim tokovima, strategijsko planiranje, te povećanje učinkovitosti navigacije i smanjenje učinaka na okoliš. Navedeni elementi svijesti o situaciji mogu se izvesti iz AIS poruka korištenjem statističke analize i strojnog učenja na opaženim (AIS poruke) i pohranjenim podacima (prostorni okoliš, kontekst).

Iako je još uvijek zakonski propisano da samo veća plovila moraju imati AIS uređaj, sve je veći porast AIS uređaja na ostalim tipovima plovila, te se time povećava i količina podataka¹. Prikazani pristup omogućuje iskorištavanje AIS podataka za otkrivanje novih znanja i dobivanje uvida u obilježja, od analize prometa u lukama do vremenske i prostorne raspodjele plovidbenih koridora², u nastavku ruta.

U ovom radu primijenjena je metoda izdvajanja prometnih ruta i detekcije anomalija prozvana TREAD (eng. Traffic Route Extraction and Anomaly Detection), koja ima za cilj pretvoriti velike količine AIS podataka u znanje koje se može iskoristiti za klasificiranje i predviđanje ruta te detekciju anomalija, neovisno o broju uređaja iz kojih se dobivaju podaci, njihovim obilježjima i platformi. Metoda je zasnovana na nenadgledanom učenju koristeći pristup inkrementalnog učenja za izdvajanje znanja o pomorskom prometu, što omogućuje karakterizaciju pomorskog prometa bez *a priori*³ znanja, tj. koristeći isključivo izvorne AIS podatke.

¹Centar za pomorska istraživanja i eksperimentiranje (eng. Centre for Maritime Research and Experimentation, CMRE) trenutno prima prosječnu stopu od 600 milijuna AIS poruka po mjesecu iz više izvora, a stopa se povećava [24].

²Koridor ili plovidbeni put/ruta.

³lat. od prijašnjega, otprije.

U poglavlju 1. opisan je automatski identifikacijski sustav AIS, princip njegova rada i izgled njegovih poruka. U poglavlju 2. opisani se tipovi i metode strojnog učenja te određeni algoritmi potrebni za razvoj metode TREAD. U poglavlju 3. opisana je metodologija rada. Poglavlje je, osim prethodnih istraživanja i opisa, podijeljeno na dva velika dijela. Prvi dio predstavlja oblikovanje pomorskog modela i princip otkrivanja prometnih spoznaja/znanja, dok drugi dio predstavlja primjenu istih. Nastavno na 3. poglavlje, u poglavlju 4., predstavljena je izvedba metode u programskom okruženju za statističko računarstvo R.

Poglavlje 1

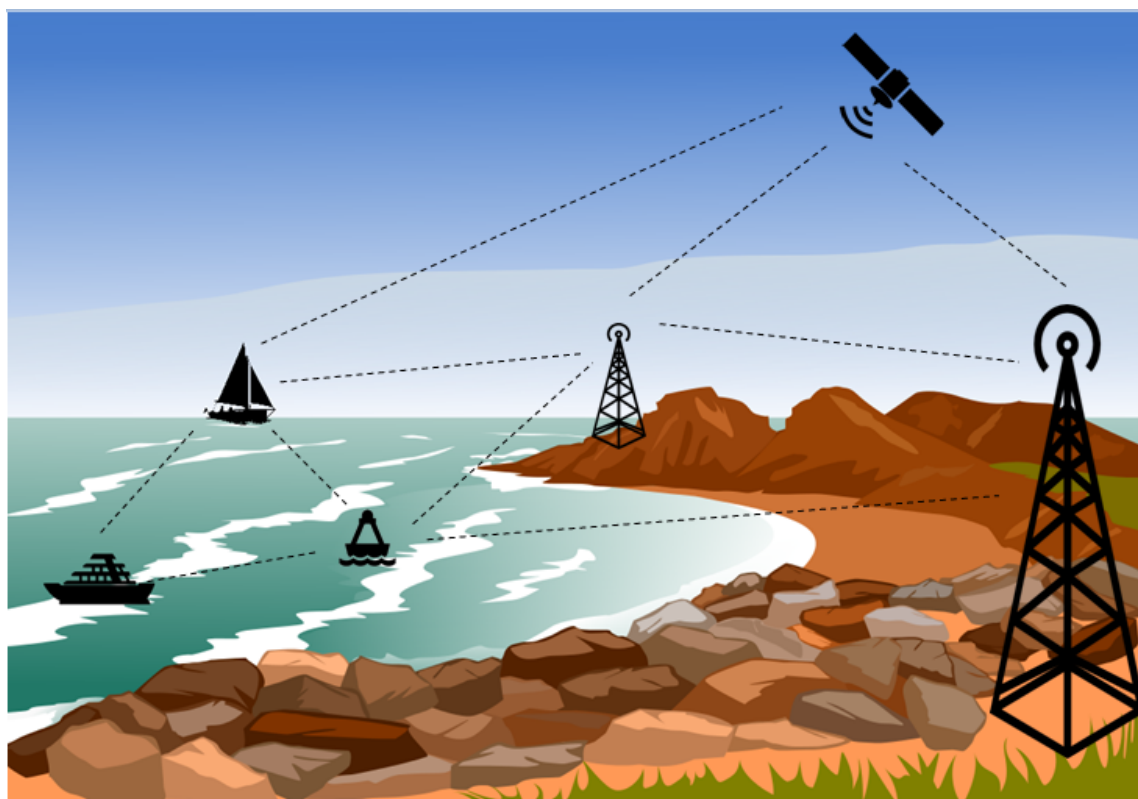
Automatski identifikacijski sustav

1.1 Uvod

Automatski identifikacijski sustav (eng. Automatic Identification System, AIS) je informacijsko-komunikacijski sustav za razmjenu navigacijskih informacija objekata izravno uključenih u pomorski promet kojeg je razvila Međunarodna pomorska organizacija (eng. International Maritime Organization, IMO¹). Pomoću AIS-a plovila postižu jednostavnu međusobnu razmjenu podataka o svom identitetu i trenutnom statusu kao što su položaj, brzina, itd. AIS se primjenjuje na svim plovnim putevima, ne samo u priobalju. Također se ostvaruje i razmjena podataka između specijalnih uređaja koji se nalaze na obalama (bazne stanice) i plutača. AIS poruke mogu se izmjenjivati i putem satelita, na prostorima daleko od obale (Slika 1.1).

Prije nastanka AIS-a, plovila i objekti koji su se nalazili u fizičkoj blizini, prikazivali bi se na radaru, a opis njihovog položaja bio je moguć preko radio-telefona. Prvobitno je AIS trebao služiti isključivo za komunikaciju brodova i obale, ali se učinilo zgodnim da brodovi mogu i međusobno komunicirati bez obzira na ljudsku interakciju.

¹Specijalizirana agencija Ujedinjenih naroda čija je glavna svrha razvoj i održavanje sveobuhvatnog regulatornog okvira za pomorsku plovidbu, te sprječavanje zagađenja uzrokovanog prometom brodova.



Slika 1.1: AIS - Automatski identifikacijski sustav

AIS je obavezan za plovila s masom većom od 300 bruto registarskih tona² (BRT) za međunarodna putovanja i većom od 500 bruto registarskih tona (BRT) za prijevoz tereta, te ribarska plovila duža od 15 metara [24].

Obaveza korištenja AIS-a vezana je za određene klase brodova i klase njihovog kretanja (ulazak u luku, krstarenje na otvorenom moru, sidrenje i slično). Na primjer, ukoliko kapetan smatra da kontinuirani rad uređaja može ugroziti sigurnost na brodu, taj isti uređaj se smije isključiti sve do ulaska u područje obaveznog javljanja, gdje o svom isključenju mora obavijestiti nadležno tijelo [1].

²Mjerna jedinica koja u pomorstvu označava volumen svih zatvorenih prostora.

1.2 Princip rada

AIS je sofisticirana radio tehnologija koja kombinira:

- globalni sustav za određivanje položaja (eng. Global Positioning System, GPS),
- područje vrlo visokih frekvencija³ (eng. Very High Frequency, VHF),
- tehnologije za obradu podataka

kako bi se omogućila razmjena informacija u definiranom formatu između različitih morskih entiteta.

AIS uređaj je radiokomunikacijska naprava (primopredajnik) koji radi, odnosno odašilje radijske poruke s AIS podacima (trenutni položaj, brzina, vrsta broda i tereta i dr.) na zajedničkoj međunarodnoj frekvenciji (AIS 1 - 161,975 [MHz] i AIS 2 - 162,025 [MHz]) koja pripada području vrlo visokih frekvencija (VHF). Obraduje više od četiri tisuće i petsto izvještaja o navigaciji u minuti, te ih ažurira svake dvije sekunde [13].

AIS uređaji prenose podatke na organiziran način, spajajući statičke podatke, kao što je jedinstveni identifikacijski broj pomorske mobilne postaje (eng. Maritime Mobile Service Identity, MMSI) i dinamičke podatke, kao što je trenutni položaj i brzina, dobivene pomoću brodskog prijammnika za satelitsku navigaciju (tradicionalno GPS, a danas i s drugim Globalnim Navigacijskim Satelitskim Sustavima, GNSS) [1, 2].

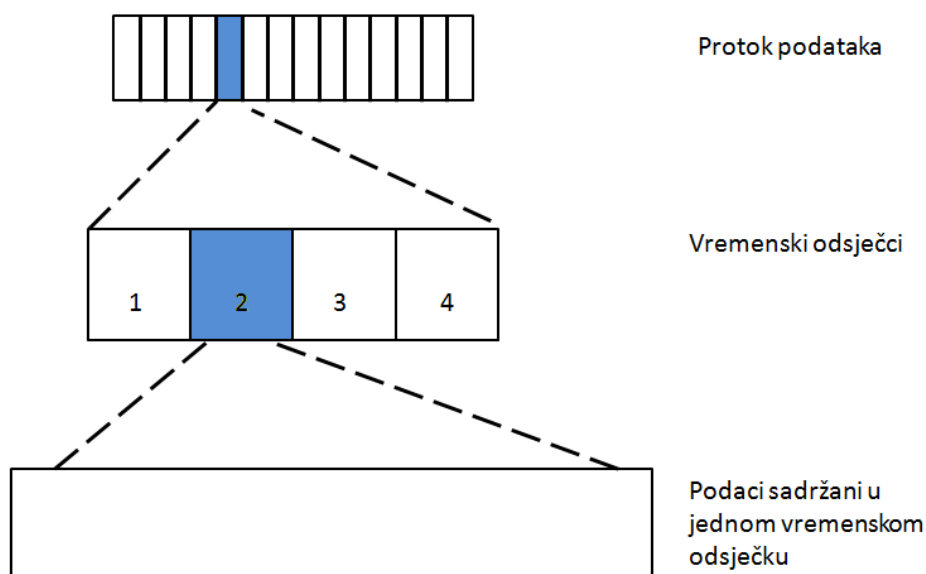
1.2.1 Način komunikacije

AIS uređaji međusobno komuniciraju korištenjem tehnike višestrukog pristupa s vremenskom raspodjelom (eng. Time Division Multiple Access, TDMA) (Slika 1.2). Ta metoda pristupa omogućuje da više korisnika dijeli isti frekvencijski kanal⁴ dijeljenjem signala u različitim vremenskim intervalima. Dodijeljeno frekvencijsko područje podjeli se na kanale, koji se dijele na velik broj vremenskih odsječaka⁵ [31].

³Područje elektromagnetskog zračenja između frekvencija od 30 [MHz] do 300 [MHz].

⁴Širina frekvencijskog pojasa (eng. bandwidth) komunikacijskog kanala koji se koristi u komunikaciji između predajne i prijemne strane.

⁵eng. time slot.



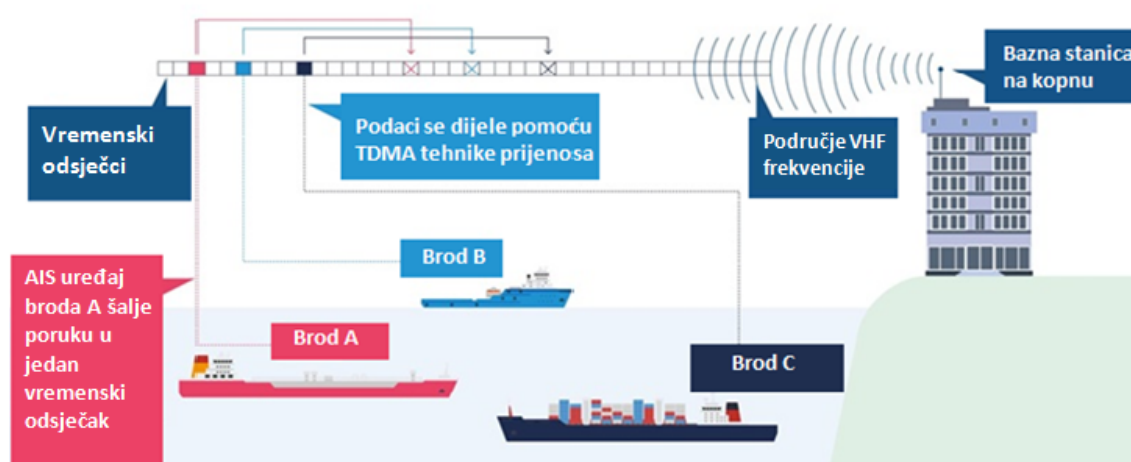
Slika 1.2: TDMA - Tehnika višestrukog pristupa s vremenskom raspodjelom

AIS koristi TDMA tehniku za dodjelu i dijeljenje raspoloživih zračnih valova na AIS frekvenciji, gdje su vremenski odsječci veličine 26,6 milisekundi [33].

Postoji više vrsta prijenosa podataka zasnovanih na TDMA tehnici koje se razlikuju po načinu pristupa vremenskom odsječku: samoorganizacijski višestruki pristup s vremenskom raspodjelom (eng. Self-Organized Time Division Multiple Access, SOTDMA); višestruki pristup s osjetom vala nosioca (eng. Carrier Sense Access Time Division Multiple Access, CSTDMA); nasumični višestruki pristup s vremenskom raspodjelom (eng. Random Access Time Division Multiple Access, RATDMA); fiksni višestruki pristup s vremenskom raspodjelom (eng. Fixed Access Time Division Multiple Access, FATDMA); i unaprijed najavljeni višestruki pristup s vremenskom raspodjelom (eng. Pre-Announced Time Division Multiple Access, PATDMA).

Osim međusobne komunikacije, AIS uređaji mogu komunicirati, tj. izmjenjivati podatke i s drugim računalnim uređajima kao što su na primjer radari i elektronski sustav za karte⁶ (eng. Electronic Chart Display And Information System, ECDIS) koristeći odgovarajuće standardne postupke (protokole) [2].

⁶Sustav koji se koristi u pomorskoj navigaciji kao alternativa papirnatim kartama.



Slika 1.3: Komunikacija AIS uređaja s TDMA tehnikom na zajedničkoj frekvenciji [2]

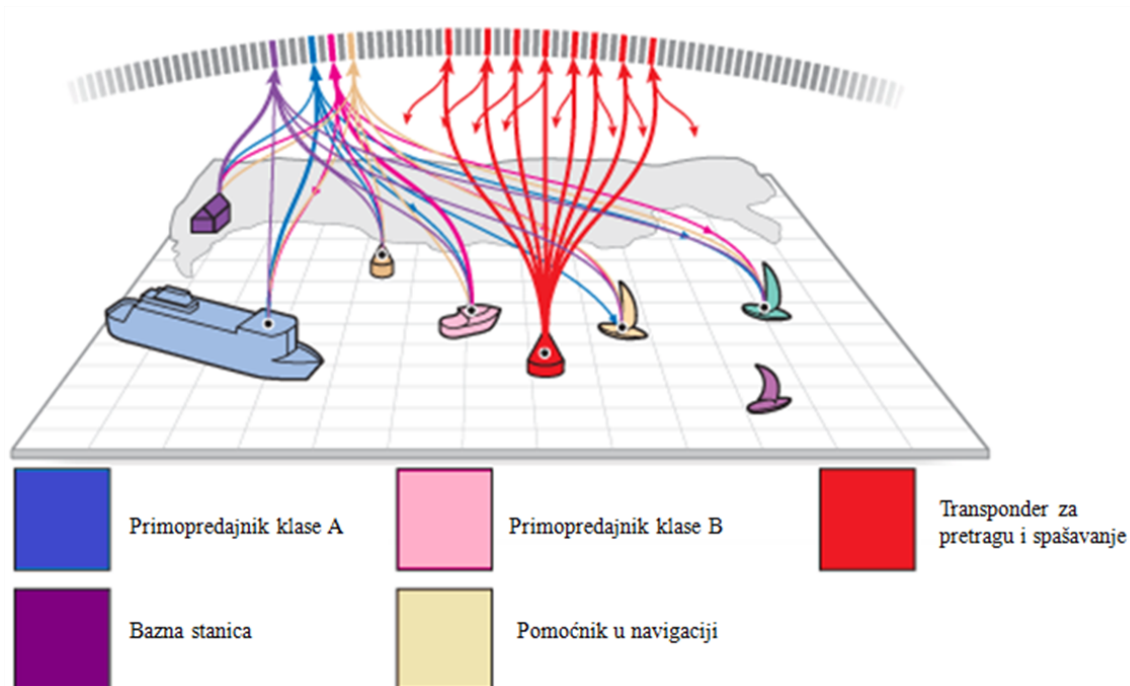
1.2.2 Vrste AIS uređaja

Vrste AIS uređaja [1]:

- primopredajnik klase A (eng. class A transceiver) - dizajniran za velika prekoceanska plovila na međunarodnim putovanjima,
- primopredajnik klase B (eng. class B transceiver) - dizajniran za manja plovila s manjim dometom u usporedbi s klasom A,
- pomoćnik u navigaciji (eng. Aids to Navigation, AtoN) - stacionarni uređaji dizajnirani za postavljanje u blizini kopna i/ili na plutačama,
- transponder⁷ za pretragu i spašavanje (eng. Search And Rescue Transponder, SART) - dizajniran za upozoravanje brodova da postoji brod za spašavanje koji treba pomoć,
- bazne stanice.

Navedeni uređaji i prikaz zauzimanja vremenskih odsječaka može se vidjeti na slici 1.4.

⁷Uređaj koji prima signale i odgovara na njih.



Slika 1.4: AIS uređaji i njihovo zauzimanje vremenskih odsječaka [2]

1.3 AIS poruke

1.3.1 Sadržaj AIS poruke

Dinamički podaci (eng. dynamic data) - dobivaju se automatski preko diferencijalnog GPS (eng. Differential Global Positioning System, DGPS)⁸ senzora spojenog na AIS uređaj, a neki od njih su:

- položaj plovila,
- UTC⁹ (eng. Coordinated Universal Time) vrijeme,
- brzina kretanja,
- smjer kretanja, ...

⁸Unaprjeđenje GPS-a koji omogućuje poboljšanu točnost, u rasponu operacija svakog sustava.

⁹Standardno vrijeme prema kojem se određuje vrijeme u drugim dijelovima svijeta.

Statički podaci (eng. static data) - unose se odmah tijekom izgradnje plovila, a mijenjaju po potrebi:

- broj međunarodne pomorske organizacije - IMO broj
- MMSI broj plovila¹⁰,
- ime plovila,
- dimenzije plovila (dužina, širina), ...

Podaci vezani uz putovanje (eng. voyage related data) - podaci koji se ručno unose prije svakog putovanja:

- vrsta tereta,
- odredište,
- predviđeno vrijeme dolaska u luku (eng. Estimated Time of Arrival, ETA).

Sigurnosni podaci (eng. short safety related messages)

Svaka od ovih skupina podataka ima svoj određeni interval javljanja. Statički podaci i podaci o plovidbi imaju interval javljanja otprilike svakih 5 minuta ili na zahtjev, dok interval za dinamičke podatke ovisi o brzini i statusu navigacije (npr. dok je brod usidren interval javljanja je otprilike 2 minute, a dok vozi brzinom većom od 23 čvora¹¹ interval javljanja je 2 sekunde [24]).

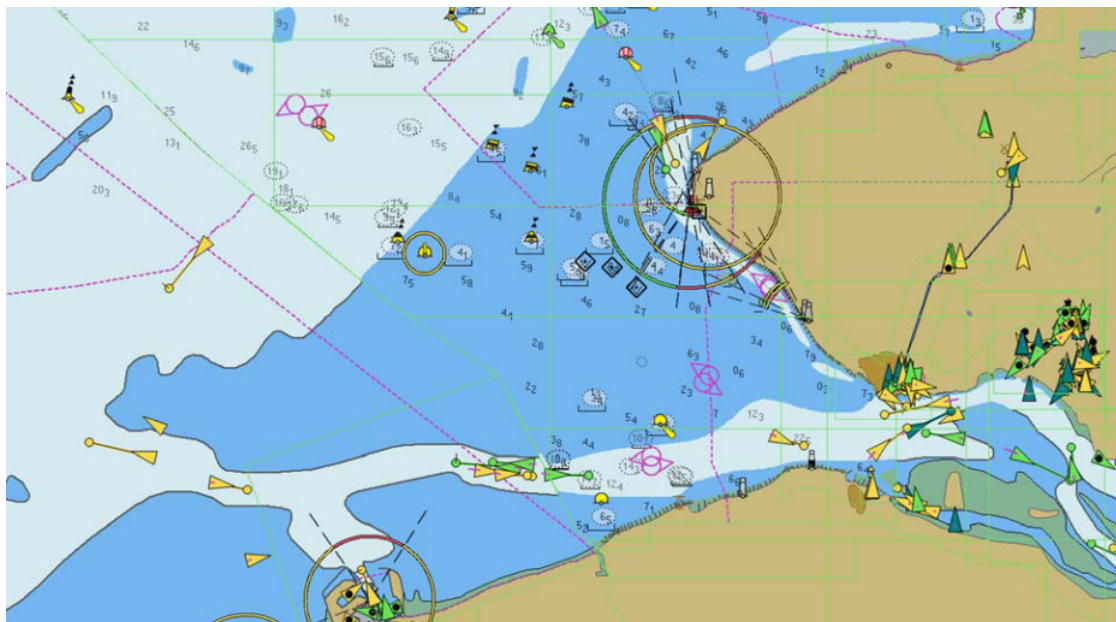
1.3.2 Prikaz AIS podataka

AIS informacije mogu se prikazivati na razne načine. Na slici 1.5 može se vidjeti prikaz podataka na AIS uređajima.

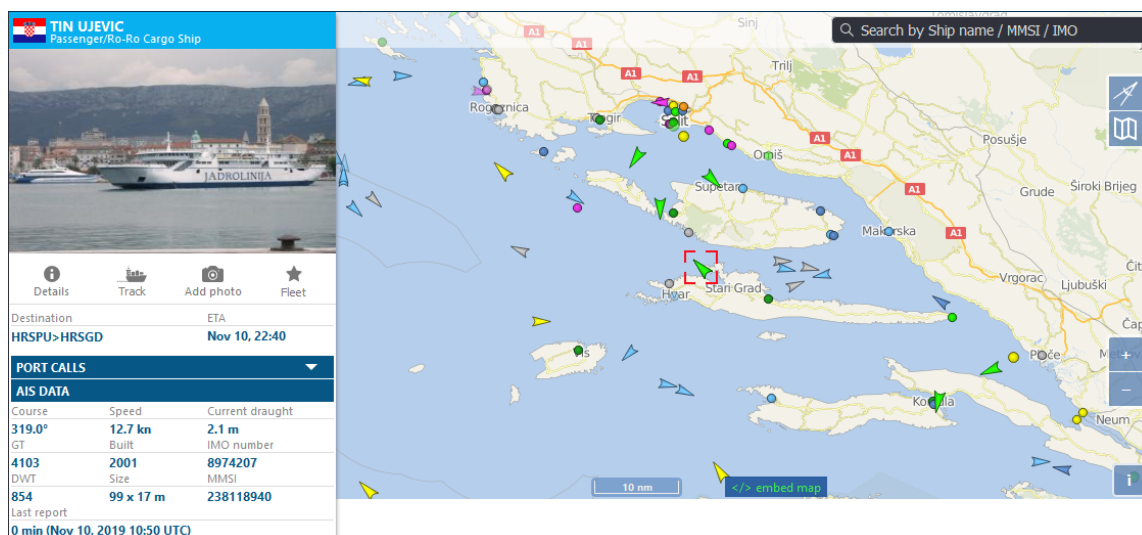
Prikaz AIS podataka je također dostupan i preko interneta. Tako se na slici 1.6 mogu vidjeti podaci o brodu Tin Ujević, njegov trenutni položaj, dinamički i statički podaci, podaci vezani uz putovanje i vrijeme zadnjeg javljanja njegovog AIS uređaja.

¹⁰Ovaj broj se nikad ne mijenja nakon što je dodijeljen nekom plovilu.

¹¹Mjerna jedinica za brzinu koja se često koristi u pomorstvu, zrakoplovstvu i meteorologiji.



Slika 1.5: Prikaz AIS podataka [1]



Slika 1.6: Prikaz AIS podataka u stvarnom vremenu [35]

Poglavlje 2

Klasifikacija metoda strojnog učenja

U ovom radu razmatrani su postupci izdvajanja prometnih ruta i detekcija anomalija. Navedeni problem može se razumjeti kao klasifikacijski, budući da elemente originalnoga skupa opažanja (u ovom slučaju ruta) želimo svrstati u skupine prema njihovim statističkim obilježjima. Za potrebe metodologije rada izabrana je metoda izdvajanja prometnih ruta i detekcije anomalija, zvana TREAD, koja će biti opisana u poglavlju 3. Metoda TREAD jedan je od postupaka strojnog/statističkog učenja na eksperimentalnim podacima. Strojno učenje omogućuje razvoj opisnog (interpretativnog) ili prognostičkog (prediktivnog) modela promatrane pojave ili procesa. U ovom poglavlju predstaviti će se pregledno najvažniji pristupi i metode strojnog učenja potrebne za razvoj metode TREAD.

2.1 Strojno učenje

Strojno učenje grana je umjetne inteligencije koja se bavi oblikovanjem algoritama koji svoju učinkovitost poboljšavaju na temelju empirijskih¹ podataka.

Definicija 2.1.1 (strojno učenje).

Strojno učenje predstavlja način programiranja računala kojim se postiže optimizacija nekog kriterija uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva.

Strojno učenje omogućava računalima da uče na sličan način kako to rade ljudi: stroj prikuplja znanje zasnovano na prošlom iskustvu. Umjesto da mu se stalno mora ažurirati softverski kod, on je, kako vrijeme prolazi, samostalno sposoban poboljšavati svoj rad. Algoritmi strojnog učenja grade matematički model zasnovan na uzorcima podataka, poznatim kao *training data* ili *learning data*, u nastavku podskup opažanja za učenje, kako bi se donijele odluke ili predviđanja. Podskup opažanja za učenje predstavlja početni skup podataka koji

¹Iskustvo, znanje stečeno iskustvom.

se koristi kao pomoć programu kako bi mogao naučiti primijeniti tehnologije i proizvesti sofisticirane rezultate.

Definicija 2.1.2 (učenje).

Računalni program uči iz iskustva E , s obzirom na neku klasu zadataka T i mjerom uspješnosti P , ako se njegov rad na zadacima u T , mjereno s P , poboljšava iskustvom E .

Primjer 2.1.3 (Problem prepoznavanja rukopisa).

- *zadatak T : prepoznavanje i klasificiranje rukopisa,*
- *mjera uspješnosti P : postotak točno klasificiranih riječi,*
- *podskup opažanja za učenje E : baza riječi pisanih rukom zajedno s klasifikacijom.*

Napomena 2.1.4. *U ovom radu podaci predstavljaju opažanja, ulazni podaci predstavljaju opažanja opisnih varijabli (prediktora), a izlazni podaci predstavljaju opažanja izlaznih varijabli (ishoda).*

Statističko učenje okvir je za strojno učenje. Bavi se uporabom statističkih modela za prepoznavanje obrazaca u opažanjima prediktora i prognozu ishoda.

Postupci strojnog učenja mogu se podijeliti u sljedeće tri osnovne skupine:

1. Nadgledano učenje (eng. supervised learning) - u ovoj inačici učenja računalu su poznata i opažanja opisnih varijabli (prediktora) i opažanja izlaznih varijabli (ishoda), zajedno s povratnim informacijama tokom njegovog učenja. Analizira se i točnost predviđanja računala tijekom samog učenja. Glavni cilj je omogućiti računalima da nauče preslikati ulaz u izlaz.
2. Nenadgledano učenje (eng. unsupervised learning) - za razliku od nadgledanog učenja ovdje se računalu predstavljaju opažanja svrstana u varijable, bez naznake koje su od njih prediktori, a koje ishodi. Zbog toga je u procesu nenadgledanog učenja potrebno najprije odrediti uzročno-posljedične odnose. Uglavnom se primjenjuje na podatke vezane za događaje, tj. promjene uzrokovane transakcijom (eng. transactional data).
3. Učenje pojačavanjem (eng. reinforcement learning) - ovo učenje zasniva se na donošenju odluke iza koje slijedi zakašnjela nagrada ili kazna (ovisno o uspješnosti odluke). Navedenim načinom razvija se iskustvo s kojim donošenje odluka biva sve uspješnije, u slučaju uspješnog učenja. Ova vrsta učenja koristi tri komponente: agenta, okruženje i akciju/djelovanje. Agent je onaj koji opaža svoje okruženje,

okruženje je ono s kojim agent ima interakciju i djeluje u tom okruženju. Glavni cilj učenja je pojačanje pronalaženja najboljeg mogućeg smjera.

Strojno učenje također je usko povezano s nekoliko drugih znanstvenih polja kao što su: rudarenje podataka (eng. data mining), optimizacija, statistika i sl. [3, 6, 16, 21].

2.2 Nenadgledano učenje

Nenadgledano učenje (učenje bez nadzora, nekontrolirano učenje) opisuje klasu problema koja uključuje korištenje modela za opisivanje ili izdvajanje odnosa u podacima.

Definicija 2.2.1 (nenadgledano učenje).

Nenadgledano učenje je vrsta strojnog učenja gdje su podaci dani bez ciljne vrijednosti, te je cilj naći pravilnost u njima.

U usporedbi s nadgledanim učenjem, nenadgledano učenje djeluje samo na opažanjima opisnih varijabli bez opažanja izlaznih ili ciljnih varijabli. Kao takvo, ono nema ispravljački model, već algoritam mora naučiti razumjeti podatke i sam pronaći rješenje. Dakle, nenadgledano učenje je učenje u kojemu za svako opažanje opisnih varijabli iz podskupa opažanja za učenje nije definirano opažanje izlazne varijable. Sustav samostalno traži uzorke među prediktorima. Postoji mnogo metoda nenadgledanog učenja, od kojih su glavne:

- klasteriranje (eng. clustering) - pronalaženje grupa u podacima,
- procjena gustoće (eng. density estimation) - uključuje sažimanje raspodjele/razdiobe podataka.

Takav oblik učenja pogodan je za pronalazak nepoznatih obrazaca u podacima i pomaže u pronalasku značajki korisnih za klasifikaciju [6, 9, 15]. Također može poslužiti i za identifikaciju međusobnih uzročno-posljedičnih veza, nakon čega je moguće prijeći u nadgledano učenje.

2.3 Inkrementalno učenje

Inkrementalno učenje igra ključnu ulogu u mnogim stvarnim aplikacijama gdje podaci dolaze s vremenom. Ono je metoda u kojoj se opažanja opisnih varijabli kontinuirano koriste kako bi se proširilo znanje postojećeg modela. Predstavlja dinamičnu metodu nadgledanog i nenadgledanog učenja koja se može primijeniti kada podskup opažanja za učenje postane dostupan postupno s vremenom.

Definicija 2.3.1 (inkrementalno učenje).

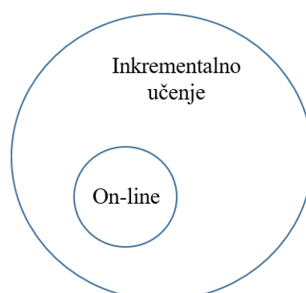
Inkrementalno učenje je paradigma strojnog učenja u kojoj se proces učenja odvija kad god se pojave novi primjeri i prilagođava se ono što je naučeno prema novim primjerima.

Algoritam inkrementalnog učenja je postupak koji generira niz modela h_1, h_2, \dots, h_t zasnovan na nizu podskupova opažanja za učenje s_1, s_2, \dots, s_t . Svaki $s_i = (x_i, y_i) \in \mathbb{R}^n \times \{1, \dots, C\}$ označava jedan podskup opažanja za učenje. Svaka $h_i : \mathbb{R}^n \times \{1, \dots, C\}$ je funkcija modela koja ovisi isključivo o h_{i-1} i nedavnih p primjeraka s_1, s_2, \dots, s_{i-p} , gdje je p strogo ograničen [18].

Pojam koji se javlja uz inkrementalno učenje je izravno povezano (uređajem podržano) učenje (eng. on-line learning). U literaturi je i dalje mnogo nejasnoća u vezi s definicijom inkrementalnog i izravno povezanog učenja. Neki ih autori upotrebljavaju kao sinonime, dok ih drugi razlikuju na različite načine. Algoritmi izravno povezanog učenja definiraju se kao algoritmi inkrementalnog učenja koji su dodatno ograničeni u složenosti modela i vremenu izvođenja, sposobni za beskrajno/cjelozivotno učenje na uređaju s ograničenim resursima. Također definira se razlika da se u pristupu izravno povezanog učenja model ažurira kako bi se prilagodio novim podacima te je moguće da model zaboravi prethodno naučene zaključke koje se nazivaju katastrofalnim smetnjama² (eng. chatastrophic interference), dok u inkrementalnom pristupu, čak i kad se model ažurira, prethodni zaključci nisu zaboravljeni. Dakle, izravno povezano učenje je uvijek inkrementalno, ali inkrementalno učenje ne mora biti izravno povezano (slika 2.1). Algoritmi za inkrementalno učenje suočavaju se sa sljedećim izazovima:

- model se mora postepeno prilagođavati, tj. h_{i+1} je izgrađen na osnovu h_i bez ponovnog učenja,
- model mora očuvati ranije stečena znanja bez učinka katastrofalnog zaboravljanja,
- model mora dopuštati samo ograničen broj od p primjera podskupova opažanja za učenje [18].

²Katastrofalne smetnje, poznate i kao katastrofalno zaboravljanje, su tendencije umjetne neuronske mreže da potpuno i naglo zaborave prethodno naučene informacije prilikom učenja novih informacija [19].



Slika 2.1: Odnos inkrementalnog i izravno povezanog učenja

2.4 Klasteriranje i algoritam DBSCAN

Klasteriranje je proces grupiranja sličnih cjelina zajedno. Cilj ove nenadgledane metode učenja je pronaći sličnosti u podatkovnim točkama³ i grupirati slične podatkovne točke zajedno. Taj proces daje uvid u temeljne obrasce različitih grupa [15].

Prvo se definiraju pojmovi potrebni za karakterizaciju dobrog klasteriranja [5].

Definicija 2.4.1 (udaljenost).

Duljina je osnovna fizikalna veličina kojom se opisuje prostorna udaljenost dviju točaka. Neka su P i Q iz \mathbb{R}^n . Udaljenost, u oznaci $\text{dist}(P, Q)$, je funkcija $\text{dist} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$.

Definicija 2.4.2 (metrički prostor).

Metrički prostor je neprazan skup X zajedno s funkcijom $\text{dist} : X \times X \rightarrow \mathbb{R}$ koja zadovoljava svojstva:

- (M1) $\text{dist}(P, Q) \geq 0$,
- (M2) $\text{dist}(P, Q) = 0 \iff P = Q$,
- (M3) $\text{dist}(P, Q) = \text{dist}(Q, P)$,
- (M4) $\text{dist}(P, R) \leq \text{dist}(P, Q) + \text{dist}(Q, R)$.

Funkcija dist naziva se funkcija udaljenosti ili metrika na X .

Mjere blizine karakteriziraju sličnost ili različitost između objekata na osnovi empirijskog ispitivanja.

³Podatkovna točka je diskretna jedinica informacija. U statističkom ili analitičkom kontekstu, podatkovna točka obično se izvodi iz mjerenja ili istraživanja i može se prikazati numerički i/ili grafički.

Definicija 2.4.3 (mjera sličnosti).

Mjera sličnosti između x, y iz skupa X , u oznaci $s(x, y)$, je funkcija $s : X \times X \rightarrow \mathbb{R}$ takva da za svaki x, y vrijedi:

$$(i) \quad 0 \leq s(x, y) \leq 1,$$

$$(ii) \quad s(x, x) = 1,$$

$$(iii) \quad s(x, y) = s(y, x).$$

Definicija 2.4.4 (mjera različitosti).

Mjera različitosti između x, y iz skupa X , u oznaci $d(x, y)$, je funkcija $d : X \times X \rightarrow \mathbb{R}$ takva da za svaki x, y vrijedi:

$$(i) \quad d(x, y) \geq 0,$$

$$(ii) \quad d(x, x) = 0,$$

$$(iii) \quad d(x, y) = d(y, x).$$

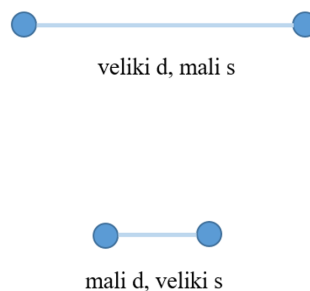
Sukladno s gornjim definicijama, stvari potrebne za dobro klasteriranje su [32]:

1. Mjera blizine:

- mjera sličnosti $s(x, y) \rightarrow$ velika ako su x i y slični,
- mjera različitosti $d(x, y) \rightarrow$ mala ako su x i y slični.

2. Funkcija kriterija za ocjenu uspješnosti klasteriranja.

3. Algoritam za računalno određivanje klastera.



Slika 2.2: Mjere blizine

Postoji više vrsta algoritama klasteriranja [12, 30]: algoritmi za razdjeljivanje, hijerarhijski algoritmi, algoritmi klasteriranja zasnovani na gustoći opažanja, algoritmi klasteriranja na osnovi rešetke⁴ i algoritmi klasteriranja zasnovani na modelu.

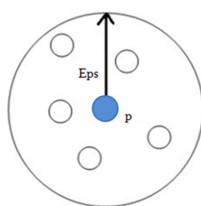
U nastavku se detaljnije obrađuje algoritam za klasteriranje zasnovan na gustoći opažanja koji se kasnije koristi u metodi ovog rada.

2.4.1 Algoritam klasteriranja zasnovan na gustoći opažanja sa šumom

Algoritam klasteriranja zasnovan na gustoći opažanja sa šumom (eng. Density-Based Spatial Clustering of Application with Noise, DBSCAN) je algoritam za klasteriranje koji se oslanja na pojam klastera⁵ koji se zasniva na gustoći i koji je osmišljen da otkrije nakupine proizvoljnih oblika. Pomaže pri identifikaciji klastera sastavljenih od objekata i pri identifikaciji objekata koji ne pripadaju niti jednom klasteru, koji se nazivaju šum (eng. noise). Glavni razlog prepoznavanja je taj da se u svakom klasteru nalaze objekti tipične gustoće koja je znatno veća nego izvan njega. Nadalje, gustoća unutar područja šuma niža je od gustoća u bilo kojem od klastera. Neka je D neka baza podatkovnih točaka, tj. objekata, gdje su podaci iz nekog metričkog k -dimenzionalnog prostora S . Za svaki objekt klastera, okolina određenog radijusa mora sadržavati minimalan broj objekata, tj. gustoća u okolini mora prelaziti neki prag. Oblik okoline određen je izborom funkcije udaljenosti između dva objekta p i q , u oznaci $dist(p, q)$. Na primjer, kada se koristi Manhattanska udaljenost u dvodimenzionalnom prostoru, okolina je pravokutnog oblika.

Definicija 2.4.5 (*Eps* okolina objekta).

Eps okolina objekta p , u oznaci $N_{Eps}(p)$, definirana je s $N_{Eps}(p) = \{q \in D \mid dist(p, q) \leq Eps\}$.



Slika 2.3: Eps okolina objekta p

⁴Dijeljenje podatkovnog prostora na ograničen broj ćelija.

⁵Nakupina srodnih članova.

Dakle, okolina određuje skup objekata udaljenog manje od Eps od promatranog objekta.

Postoje dvije vrste objekata u klasteru: objekti unutar klastera, odnosno jezgri objekti (eng. core objects) i objekti na rubu klastera, odnosno rubni objekti (eng. border objects). Neka je $MinPts$ oznaka za minimalan broj objekata. $MinPts$ označava koliko se objekata mora nalaziti unutar okoline promatranog objekta kako bi on mogao biti jezgri objekt. Zaključuje se kako je $MinPts$ minimalan broj objekata koji čini klaster.

Zahtjeva se da za svaki objekt p u klasteru postoji objekt q u klasteru tako da je p unutar Eps okoline objekta q i $N_{Eps}(q)$ sadrži barem $MinPts$ objekata. Slijedi precizna definicija.

Definicija 2.4.6 (izravna dohvatljivost).

Objekt p je izravno dohvatljiv (eng. directly density-reachable) iz objekta q s obzirom na Eps , $MinPts$ ako vrijedi:

1. $p \in N_{Eps}(q)$.
2. $|N_{Eps}(q)| \geq MinPts$ (uvjet jezgri objekta).

Drugim riječima, objekt p je izravno dohvatljiv iz objekta q ako se p nalazi u okolini objekta q i q čini jezgri objekt.

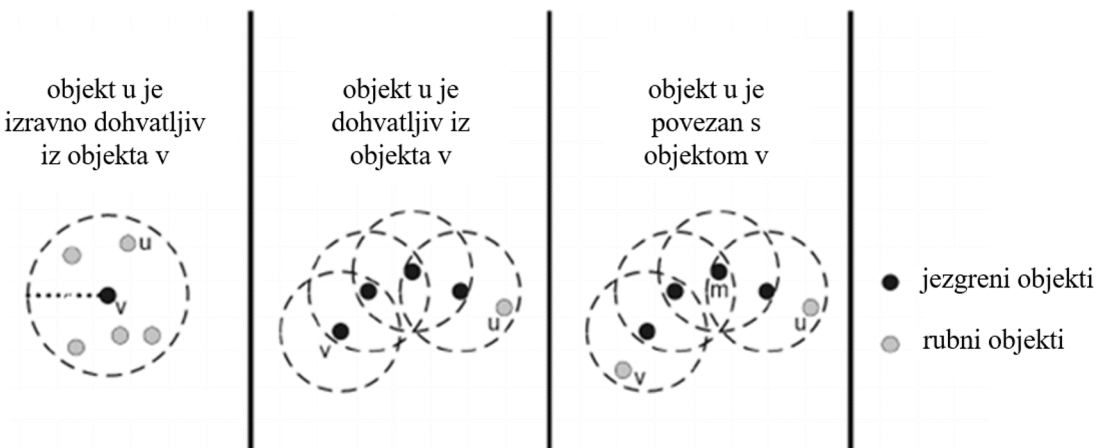
Definicija 2.4.7 (dohvatljivost).

Objekt p je dohvatljiv (eng. density-reachable) iz objekta q s obzirom na Eps , $MinPts$ ako postoji put preko objekata p_1, p_2, \dots, p_n , gdje je $p_1 = q$, $p_n = p$, takvih da je p_{i+1} izravno dohvatljiv iz p_i .

Problematični mogu biti rubni objekti klastera zbog mogućnosti da dva takva objekta istog klastera nisu dohvatljivi jedan iz drugog jer ne vrijedi uvjet jezgri objekta. Uvodi se još jedna relacija koja će pokriti relacije rubnih objekata.

Definicija 2.4.8 (povezanost).

Objekt p je povezan (eng. density-connected) s objektom q s obzirom na Eps , $MinPts$ ako postoji objekt o takav da su oba objekta p i q dohvatljivi iz objekta o s obzirom na Eps , $MinPts$.



Slika 2.4: Relacije dohvatljivosti i povezanosti

Vrijedi: izravna dohvatljivost \subseteq dohvatljivost \subseteq povezanost.

Nakon definiranih relacija slijede precizne definicije klastera i šuma s obzirom na gustoću.

Definicija 2.4.9 (klaster).

Neka je D baza objekata. Klaster C s obzirom na Eps , $MinPts$ je neprazan podskup od D koji zadovoljava:

1. $\forall p, q$: ako je $p \in C$ i q je dohvatljiv iz p s obzirom na Eps , $MinPts$, tada je $q \in C$.
2. $\forall p, q \in C$: p je povezan s q s obzirom na Eps , $MinPts$.

Definicija 2.4.10 (šum).

Neka su C_1, \dots, C_k klasteri baze D s obzirom na parametre Eps_i , $MinPts_i$, $i = 1, \dots, k$. Šum (eng. noise) je skup objekata baze D koji ne pripada niti jednom klasteru C_i , odnosno vrijedi da su šum svi objekti za koje vrijedi $\{p \in D \mid \forall i : p \notin C_i\}$.

Dakle, nedohvatljivi objekti označavaju se kao šum i ne grupiraju se [12].

Opis algoritma DBSCAN

Algoritam DBSCAN osmišljen je da otkrije klustere i šumove u prostornim bazama. Algoritam inicijalno prima skup podatkovnih točaka, tj. objekata D , veličinu okoline Eps i minimalan broj objekata za stvaranje klastera $MinPts$. Na početku su svi objekti u D neklasificirani.

1. Izvršavanje algoritma započinje proizvoljnim objektom koji je neklasificiran (nije još posjećen) - informacije o okolini preuzima iz parametra Eps .
2. Ako taj objekt sadrži $MinPts$ u svojoj Eps okolini započinje oblikovanje klastera. Inače, označen je kao šum.
3. Ako se utvrdi da je objekt jezgreni, tada su objekti unutar njegove Eps okoline također dio klastera. Tako se dodaju svi objekti pronađeni u Eps okolini, zajedno sa njihovim Eps okolinama, ako su i oni jezgreni objekti.
4. Gornji postupak se nastavlja sve dok se u potpunosti ne nađe povezan klaster.
5. Postupak se ponovo pokreće s novim objektom koji može biti dio novog klastera ili označen kao šum.

Napomena 2.4.11. *U nekim radovima umjesto pojma podatkovna točka ili objekt koristi se samo točka. Neovisno o nazivu, navedeni objekt je jedinica informacija definirana u metričkom prostoru.*

Algoritam 1 predstavlja pseudokod algoritma DBSCAN i vidljiv je na sljedećoj stranici.

Algoritam 1 DBSCAN algoritam [12]

Require: $D, Eps, MinPts$ ▷ Svi objekti u D su neklasificirani

- 1: **for all** objects o in D **do**
- 2: **if** o is unclassified **then**
- 3: call function *expand_cluster* to construct a cluster containing o
- 4: **end if**
- 5: **end for**

- 6: **function** EXPAND_CLUSTER($o, D, Eps, MinPts$)
- 7: retrieve the Eps neighborhood $N_{Eps}(o)$ of o
- 8: **if** $|N_{Eps}(o)| < MinPts$ **then** ▷ o nije jezgri objekat
- 9: mark o as noise
- 10: **return**
- 11: **else** ▷ o je jezgri objekat
- 12: select a new cluster_id and mark all objects in $N_{Eps}(o)$ with this current cluster_id
- 13: push all objects from $N_{Eps}(o) \setminus \{o\}$ onto the stack *seeds*
- 14: **while** seeds.empty() **do**
- 15: $currentObject := seeds.top()$
- 16: receive the Eps neighborhood $N_{Eps}(currentObject)$ of $currentObject$
- 17: **if** $|N_{Eps}(currentObject)| \geq MinPts$ **then**
- 18: select all objects in $N_{Eps}(currentObject)$ not yet classified or marked as noise
- 19: push the unclassified objects onto seeds and mark them with current cluster_id
- 20: **end if**
- 21: *seeds.pop*
- 22: **end while**
- 23: **return**
- 24: **end if**
- 25: **end function**

2.4.2 Inkrementalni DBSCAN algoritam

DBSCAN algoritam, opisan u prethodnom odjeljku, primjenjiv je na statičke baze, no u slučaju kad se baze stalno mijenjaju tokom vremena potrebna je inačica algoritma koja će raditi i s takvim podacima. U tu svrhu predstavljen je inkrementalni DBSCAN algoritam koji daje iste rezultate kao i neinkrementalna verzija [11].

Objekti u bazi mogu se s vremenom umetati, ali i brisati. Time inkrementalni DBSCAN algoritam ažurira klastere nakon svakog novog umetanja i brisanja podataka. Zbog prirode inkrementalnog DBSCAN algoritma, koji se zasniva na gustoći, umetanje ili brisanje objekta utječe samo na objekte unutar određene okoline.

Prepoznati objekti (eng. affected objects) su potencijalni objekti koji mogu promijeniti svoju pripadnost u klasteru nakon umetanja/brisanja objekta p . Ti objekti su definirani kao objekti u $N_{Eps}(p)$ plus svi ostali objekti koji su dohvatljivi iz objekata u $N_{Eps}(p)$. Slijedi precizna definicija.

Definicija 2.4.12 (prepoznati objekti).

Neka je D baza podataka objekata i neka je p neki objekt (ne nužno iz D). Skup prepoznatih objekata u D , nakon umetanja ili brisanja objekta p , definiran je kao $Affected_D(p) = N_{Eps}(p) \cup \{q \mid \exists o \in N_{Eps}(p) \wedge q >_{D \cup \{p\}} o\}$.

Pripadnost klasteru svih ostalih objekata koji nisu u skupu prepoznatih objekata se ne mijenja.

Slijedi precizna definicija objekata koji se moraju ažurirati, nakon umetanja ili brisanja.

Definicija 2.4.13 (objekti za umetanje/brisanje).

Neka je D skup svih objekata i p objekt za umetanje ili brisanje. Definiraju se sljedeći skupovi:

$$UpdSeed_{Ins} = \{q \mid q \text{ je jezgreni objekt u } D \cup \{p\}, \exists q' : q' \text{ je jezgreni objekt u } D \cup \{p\}, \text{ ali nije u } D \text{ i } q \in N_{Eps}(q')\},$$

$$UpdSeed_{Del} = \{q \mid q \text{ je jezgreni objekt u } D \setminus \{p\}, \exists q' : q' \text{ je jezgreni objekt u } D, \text{ ali nije u } D \setminus \{p\} \text{ i } q \in N_{Eps}(q')\}$$

Objekte $q \in UpdSeed_{Ins}$ zovemo objektima za umetanje, a objekte $q \in UpdSeed_{Del}$ objektima za brisanje.

Kada se radi umetanje objekta p u bazu D , razlikuju se sljedeći slučajevi:

1. (Šum) Ako je $UpdSeed_{Ins}$ prazan, tada se p smatra šumom.
2. (Kreiranje) Ako $UpdSeed_{Ins}$ sadrži samo jezgrene objekte koji nisu pripadali klasteru prije umetanja p , tada p kreira novi klaster s tim objektima.
3. (Apsorpcija) Ako $UpdSeed_{Ins}$ sadrži jezgrene objekte koji pripadaju samo jednom klasteru C , onda se p pridružuje tom klasteru.
4. (Spajanje) Ako $UpdSeed_{Ins}$ sadrži jezgrene objekte koji pripadaju u nekoliko klastera, tada p spaja sve te klastere u jedan klaster.

Slično, kada se radi brisanje objekta p iz baze D , razlikuju se sljedeći slučajevi:

1. (Uklanjanje) Ako je $UpdSeed_{Del}$ prazan, p se samo uklanja.
2. (Redukcija) Ako $UpdSeed_{Del}$ sadrži objekte koji su dohvatljivi jedni iz drugih, tada brisanjem objekta p neki objekti iz $N_{Eps}(p)$ postaju šumovi.
3. (Potencijalna podjela) Ako objekti u $UpdSeed_{Del}$ nisu izravno dohvatljivi jedni iz drugih, tada su ti objekti pripadali točno jednom klasteru C prije brisanja objekta p . Radi se provjera jesu li ti objekti bili povezani. Ovisno o postojanju povezanosti, razlikuju se situacije podjele i ne podjele [11].

Poglavlje 3

Metoda izdvajanja prometnih ruta i detekcija anomalija

3.1 Prethodna istraživanja

Primjena statističkih metodologija za dobivanje obrazaca kretanja iz skupa ruta primjenom nenadgledanog učenja predstavlja izazovan zadatak.

Uobičajeni pristup je podjela područja interesa u prostornu rešetku čije ćelije predstavljaju obilježja gibanja plovila koja uđu u tu ćeliju [14]. Iako je pristup pomoću rešetke učinkovit za nadzor manjih površina, njegova glavna ograničenja su proporcionalno povećanje kompleksnosti sa povećanjem veličine rešetke i potreba za predefiniranjem optimalne veličine ćelije.

Novi koncept u polju otkrivanja anomalija/iznimaka ustaljenih obrazaca u odvijanju pomorskog prometa predstavlja vektorski prikaz prometa, pri čemu se ruta/putanja plovila smatra skupom putova plovila koji povezuju točke planirane plovidbe brodova¹, u nastavku putna točka (eng. waypoint), što omogućuje kompaktnu reprezentaciju kretanja plovila. Tako na primjer u [23], putne točke su čvorovi u blizini kopnenih područja, a rute velikog kruga (eng. Great Circle route) predstavljaju oceanska putovanja. U područjima s velikim brojem promjena smjera plovila potrebno je dodatno uvesti prijelazne točke, tj. točke promjene smjera (eng. turning points) kako bi se točnije opisivale rute [34]. Korištenje udaljenosti kao podloge predstavlja nedostatak jer izaziva smanjenu djelotvornost kod razlikovanja međusobno bliskih točaka.

Također, identifikacija prijelaznih točaka u nereguliranim područjima pojavljuje se kao nedostatak vektorskog pristupa, gdje su promjene smjera plovila složenije, pa ih je prema tome teško klasificirati.

U zadnje vrijeme nastaju brojni radovi na temu detekcije anomalija. Postupci detekcije

¹Bilo koja mapirana referentna točka na ruti koju je lako prepoznati.

anomalija, prikazani u radovima poput [27], oslanjaju se na postupke procesiranja slika. Alternativni postupci, poput onih prikazanih u literaturi [36], primjenjuju vjerojatnosnu detekciju anomalija, gdje se udaljenost između ruta izražava vjerojatnošću.

Mnoge metode uključuju prethodnu analizu ruta, čime se omogućuje korištenje uobičajenih mjera udaljenosti, kao što je na primjer euklidska udaljenost koja zahtjeva podjednako raspoređene i usklađene rute.

3.2 Opis

Metoda TREAD - Traffic Route Extraction and Anomaly Detection - predstavljena je od strane G.Pallotte et al. [24]. Zasniva se na vektorizaciji pomorskog prometa, ali za razliku od prethodno predstavljenih postupaka objekti koji predstavljaju rute (eng. route objects), u nastavku rute, se izravno oblikuju iz vektora objekata koji predstavljaju plovila (eng. vessel objects), u nastavku plovila, čiji putovi spajaju izvedene putne točke. Konkretno, navedeni pristup se zasniva na prethodnom grupiranju putnih točaka i identificiranju ruta između njih. Za razliku od ostalih vektorskih reprezentacija, ruta obuhvaća promjene smjera bez eksplicitno izvedenih prijelaznih točaka. Kao što će se vidjeti u nastavku, tim pristupom je i dalje moguće točno odrediti obrasce ruta, ali i izdvojiti privremene informacije kao što je vremenska razdioba putovanja² te pridružiti povijesne obrasce ruta nekom plovilu.

Kako bi se savladala problematika razlikovanja putnih točaka koje su blizu jedna drugoj, koristi se DBSCAN algoritam, i to inkrementalna verzija predstavljena u poglavlju 2.4.2. Za otkrivanje obrazaca kretanja i njihovo predviđanje koristi se nenadgledano učenje, točnije inkrementalni pristup zasnovan na točkama, dok je detekcija anomalija izvedena pomoću kliznog vremenskog prozora (eng. sliding time window). Prethodna analiza, transformacija i vrednovanje AIS podataka integrirana je u arhitekturu koja pruža (programski) okvir (eng. framework) za analizu i razvoj modela uzoraka kretanja. Navedeni okvir oblikovan je objektnim modelom koji se sastoji od objekata ruta, plovila i putnih točaka. Govori se o plovilima, rutama i putnim točkama, iako se smatra da su oni objekti predstavljeni u obliku podatkovne točke.

U nastavku se najprije opisuje princip rada metode TREAD za otkrivanje pomorskih prometnih spoznaja/znanja, odnosno otkrivanje obrazaca kretanja (eng. knowledge discovery), te nakon toga korištenje tih spoznaja za klasificiranje i predviđanje ruta kao i za detekciju anomalija (eng. knowledge exploitation) [24].

²Razdioba putovanja po vremenskih intervalima.

3.3 Prometni model i otkrivanje prometnih spoznaja

TREAD metoda osmišljena je s ciljem razvoja statističkog modela pomorskog prometa zasnovanog na AIS podacima primjenom postupka nenadgledanog učenja.

Odabran je i prostorni okvir (eng. bounding box) koji odgovara promatranom prostoru od interesa.

Neka je s Vs označen skup objekata koji predstavljaju plovila, odnosno oznaka za listu plovila. Klasteriranje, tj. grupiranje, zasnovano na gustoći putnih točaka, pokrenuto od strane različitih objekata plovila Vs omogućuje oblikovanje putnih točaka, u oznaci WPs , koje označavaju jedno od sljedećeg unutar odabranog okvira:

(T1) stacionarna točka (eng. stationary point), u oznaci POs ,

(T2) ulazna točka (eng. entry point), u oznaci ENs , (3.1)

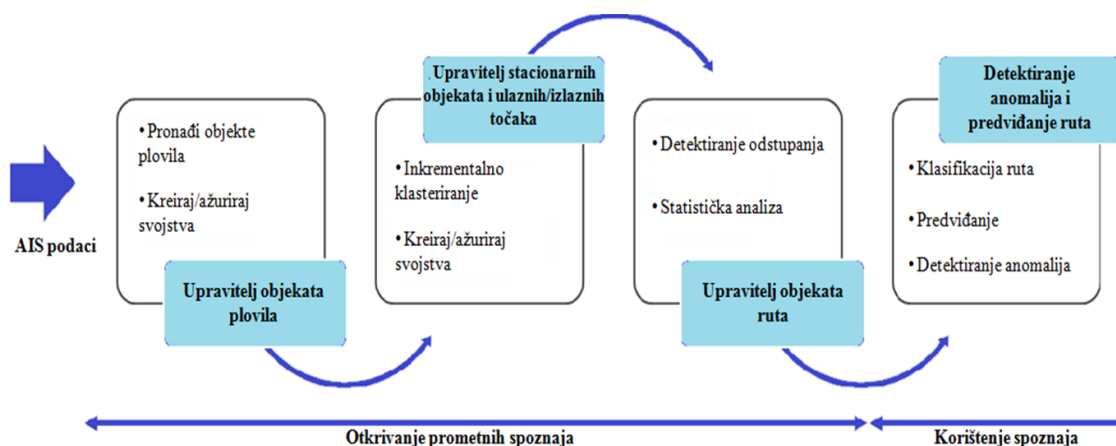
(T3) izlazna točka (eng. exit point), u oznaci EXs .

Povezivanje navedenih putnih točaka vodi do identifikacije i statističke karakterizacije ruta, oblikovanih kao objekti koji predstavljaju rute, u oznaci Rs .

Navedeni pristup u literaturi se naziva pristup zasnovan na točkama (eng. point-based approach), za razliku od pristupa zasnovanog na ruti, gdje se promet predstavlja cijelim rutama. Pristup zasnovan na točkama je jedan od kategorija pristupa strojnog i statističkog učenja u otkrivanju prometnih spoznaja, te svaka točka u (3.1) predstavlja podatkovnu točku/objekt, odnosno smatra se pojedinačnom točkom rute.

3.3.1 Struktura metode TREAD

Otkrivanje i iskorištavanje spoznaja o pomorskom prometu zasnovano na AIS podacima slijedi sljedeću arhitekturu³:



Slika 3.1: Struktura metode TREAD (prema [24])

Kao što se može vidjeti na slici 3.1, AIS podaci su analizirani s ciljem inkrementalnog učenja modela obrazaca kretanja plovila kroz procese nazvane upravitelji (eng. managers) koje aktiviraju određeni relevantni događaji. Upravitelj objekata plovila aktiviran je događajima zasnovanim na vremenskoj i prostornoj karakterizaciji plovila. Zatim se takvi događaji grupiraju, što dovodi do identifikacije putnih točaka kroz upravitelja stacionarnih objekata i ulazno/izlaznih točaka. Nakon što su saznate putne točke, iz njih se mogu izvesti rute. Nakon tog procesa otkrivanja znanja, ostvarene spoznaje mogu se iskoristiti u svrhu klasifikacije ruta, predviđanja i detekcije anomalija. U nastavku opisujemo detaljnije proces otkrivanja spoznaja kroz navedene upravitelje.

Upravitelj objekata plovila

Čim novo plovilo uđe u okvir/prostor koji se nadzire dolazi do detekcije te se inicijalizira upravljanje plovilima V_s . Lista plovila, V_s , ažurirana je prema sadržaju svake nove dekodirane AIS poruke. Sva plovila se identificiraju pomoću MMSI broja, u oznaci $V_s\{MMSI\}$, i u posebnoj varijabli sadrže statičke i dinamičke podatke. Statički podaci služe za potrebe identifikacije, dok dinamički podaci čine vektor stanja (položaj (koordinate), kurs (eng. Course Over Ground, COG), brzina (eng. Speed Over Ground, SOG)) i prijašnji obrasci.

³Arhitektonski model koji identificira funkcije sustava i njihove interakcije.

Ti se podaci postepeno ažuriraju svakom novom AIS porukom. Tokom provedbe, algoritam analizira i ažurira informacije o povijesti i statusu plovila.

Za svako određeno plovilo, $Vs\{MMSI\}$, ažurira se i vremenski označena povijest opaženih informacija vektora stanja (tj. parametara položaja i brzine), u oznaci $Vs\{MMSI\}.track$, u nastavku trag rute. Također, ažurira se i njegov status, koji može biti:

- izgubljen - ako objekt nije promatran neko vrijeme τ , gdje je τ vrijeme veće od maksimalne stope ažuriranja AIS poruka,
- stacionaran - ako objekt stoji,
- pokretan - ako se objekt pomiče, tj. plovi,
- tranzicije - promjene iz stajanja u kretanje plovila i obrnuto.

Ažuriranje statusa i praćenje tranzicija plovila omogućuje identifikaciju i ažuriranje putnih točaka WPs .

Upravitelj stacionarnih objekata i ulaznih/izlaznih točaka

Kao što je navedeno u (3.1), posebnu klasu putnih točaka predstavljaju stacionarne točke, POs . Primjer takvih točaka su luke ili platforme, koje su po svojim obilježjima nepomični objekti. Također, u stacionarne objekte spadaju plovila koja imaju brzinu manju od datog praga. Stacionarni događaji se prvo detektiraju promjenom brzine plovila s obzirom na posljednje opažanje unutar promatranog okvira. Parametri

Δ_t - posljednji promatrani vremenski interval,

Δ_{pos} - rezultirajući pomak položaja,

izračunavaju se empirijski kako bi se dobila prosječna brzina plovila:

$$\text{prosječna brzina} = \Delta_{pos} / \Delta_t.$$

Prosječnu brzinu potrebno je računati datom formulom zbog uočene nepouzdanosti vrijednosti parametra SOG u AIS poruci.

Nakon detekcije točaka, obavlja se njihovo grupiranje u stacionarne objekte koristeći klasteriranje. Kao što je prije navedeno, za grupiranje putnih točaka, pa tako i stacionarnih, koristi se inkrementalni DBSCAN algoritam. DBSCAN oblikuje klasterne objekata zasnovano na gustoći objekata u njihovoj okolini.

U kontekstu metode TREAD putne točke koje nisu povezane, s obzirom na definiciju 2.4.8, s drugim putnim točkama ne pripadaju niti jednom klasteru te se smatraju šumom.

DBSCAN algoritam ne zahtijeva prethodno poznavanje broja klastera.

Karakteristična klasa putnih točaka unutar odabranog okvira su ulazne točke, ENs i izlazne točke, EXs . Kad god plovilo uđe/izađe iz prostornog okvira koji se promatra, generiraju se ili ažuriraju ulazne/izlazne točke respektivno. Za razliku od stacionarnih točaka, ulazne i izlazne točke se mogu mijenjati ovisno o okviru kojeg se promatra. Slično kao i kod stacionarnih točaka, ulazne i izlazne točke identificiraju se inkrementalnim DBSCAN algoritmom.

U nastavku je prikazan algoritam klasteriranja putnih točaka (Algoritam 2).

Algoritam 2 Klasteriranje putnih točaka (*Online_WPs_Clustering*) [24]

Require: Vs, v ․ lista plovila Vs i plovilo v koje pokreće klasteriranje
Require: WPs, Rs ․ lista putnih točaka za klasteriranje i ruta za modifikaciju
Require: Eps, N ․ parametri za inkrementalni DBSCAN; N je minimalan broj točaka za generiranje klastera $wp_n \in WPs$ u Eps okolini $v.track(end)$

```

1: [ $WPs, op$ ] ← Incremental_DBSCAN( $WPs, v.track(end), N, Eps$ )
2: if  $op = 'none'$  then ․ šum
3:    $v.wps(end + 1) \leftarrow ('Unclassified', v.track(end))$ 
4: else ․ kreiranje, apsorpcija ili spajanje
5:   if  $op = 'new\ cluster'$  then ․ kreirao se novi klaster  $wp_n$ 
6:      $WPs \leftarrow add('WP_n')$ 
7:      $v.wps(end + 1) \leftarrow ('WP_n')$  ․ ažurira se lista putnih točaka plovila
8:      $v.timestamp_{wp}(end + 1) \leftarrow (v.track(end))$ 
       ․ MMSI plovila i njegov zadnji položaj je zabilježen u  $wp_n$ 
9:     [ $wp_n.List\_MMSIs(end + 1), wp_n.tracks(end + 1)$ ] ← ( $v.MMSI, v.track(end)$ )
10:  end if
11:  if  $op = 'cluster\ expanded'$  then ․ apsorpcija u klaster  $wp_n$ 
12:     $v.wps(end + 1) \leftarrow ('WP_n')$ 
13:     $v.timestamp_{wp}(end + 1) \leftarrow (v.track(end))$ 
14:    [ $wp_n.List\_MMSIs(end + 1), wp_n.tracks(end + 1)$ ] ← ( $v.MMSI, v.track(end)$ )
15:  end if

```

```

16:   if  $op = \text{'clusters merged'}$  then           ▶ spajanje dva klastera  $wp_m$  i  $wp_n$  u  $wp_n$ 
17:        $v.wps(end + 1) \leftarrow (\text{'WP}_n\text{'})$ 
18:        $v.timestamp_{wp}(end + 1) \leftarrow (v.track(end))$ 
19:        $wp_n \leftarrow (v.MMSI, v.track(end))$ 
20:        $wp_n \leftarrow merge(wp_n, wp_m)$ 
21:       for all  $\hat{v} \in Vs\{MMSI = wp_m.List\_MMSIs\}$  do
22:            $\hat{v}.wps(\hat{v}.wps = \text{'WP}_m\text{'}) \leftarrow (\text{'WP}_n\text{'})$ 
23:       end for
      ▶ spoji sve prepoznate rute i ažuriraj
24:       for all  $\hat{R} \in (R_s.wps(1) = \text{'WP}_m\text{'} \mid R_s.wps(2) = \text{'WP}_m\text{'})$  do
25:            $\tilde{R} \leftarrow (R_s.wps(\hat{R}.wps = \text{'WP}_m\text{'}) = \text{'WP}_n\text{'})$ 
26:            $\tilde{R} \leftarrow merge(\tilde{R}, \hat{R})$ 
27:            $delete(\hat{R})$ 
28:       end for
29:        $delete(\text{'WP}_m\text{'})$ 
30:   end if
31: end if
32: return  $WPs, v, Rs$ 

```

Napomena 3.3.1. Radi jednostavnosti pisanja, oznaka N , u Algoritmu 2, označava minimalan broj točaka klastera, odnosno znači isto što i oznaka $MinPts$ iz odjeljka opisa DBSCAN algoritma (2.4.1).

Parametri za klasteriranje Eps i N podešeni su na osnovi vrste putnih točkaka (POs , ENs , EXs) i obilježja promatranog područja.

Prikazani algoritam filtrira šumove koristeći posebnosti DBSCAN algoritma. Oni se ne smatraju valjanim uzorcima za daljnje iskorištavanje spoznaja te se u Algoritmu 2 uklanjaju.

Upravitelj objekata ruta

Nakon identifikacije putnih točaka, rute Rs , mogu se izgraditi klasteriranjem putova plovila, gdje putovi spajaju dvije dobivene putne točke. Postoje različite alternativne metode spajanja: spajanje ulazne i izlazne točke, spajanje ulazne i stacionarne točke, spajanje izlazne i stacionarne točke ili spajanje dviju stacionarnih točaka.

Također, svaka ruta je opisana statičkim i dinamičkim obilježjima plovila na temelju kojeg je kreirana i ažurirana ruta.

U nastavku je prikazan algoritam izgradnje ruta (Algoritam 3).

Algoritam 3 Upravitelj objekata ruta (*Route_Objects_Manager*) [24]

Require: v, WPs, Rs ▷ plovilo v za kojeg se oblikuju putne točke i rute

```

1: if  $length(v.wps) > 1$  then ▷ plovilo je prošlo kroz bar dvije putne točke
2:    $[wp_a, wp_b] \leftarrow v.wps(end - 1 : end)$ 
3:   if not  $(Rs\{wp_a\_to\_wp_b\})$  then ▷ ruta od  $wp_a$  do  $wp_b$  ne postoji; dodaje se u listu  $Rs$ 
4:      $Rs \leftarrow add(Rs\{wp_a\_to\_wp_b\})$ 
5:   end if
6:    $timestamp_{wp_a} = v.timestamp_{wp}(v.wps = wp_a)$ 
7:    $timestamp_{wp_b} = v.timestamp_{wp}(v.wps = wp_b)$ 
8:    $Rs\{wp_a\_to\_wp_b\}.params(end + 1)$ 
    $\leftarrow (v.track(timestamp \in [timestamp_{wp_a}, timestamp_{wp_b}]))$ 
9:    $v.routes \leftarrow add('Rs\{wp_a\_to\_wp_b\}')$  ▷ ažuriraj listu ruta plovila
10: end if
11: return  $v, Rs$ 

```

Jednom kada brod uđe u prostor koji se nadgleda, obilježja njegovog kretanja uspoređuju se s obilježjima postojećeg skupa ruta. Ako već postoji ruta čija su obilježja položaja kompatibilna sa značajkama plovila, to plovilo se dodaje listi ruta plovila, a ruta se dodaje listi putnih točaka kojima je brod prošao.

Opisani postupak predočen je algoritmom nenadgledanog izdvajanja ruta (eng. Unsupervised Route Extraction) (Algoritam 4) koji predstavlja algoritam za otkrivanje prometnih spoznaja metode TREAD.

Algoritam 4 Nenadgledano izdvajanje ruta (*Unsupervised_Route_Extraction*) [24]

Require: $messages$ ▷ AIS poruke koje sadrže statičke i dinamičke podatke
Require: τ ▷ potrebno vrijeme prije nego se označi da je plovilo "izgubljeno"
Require: Vs, ENs, POs, EXs, Rs ▷ lista plovila, putnih točaka i ruta
Require: $N_{ENs}, N_{POs}, N_{EXs}, Eps_{ENs}, Eps_{POs}, Eps_{EXs}$ ▷ parametri za klasteriranje u algoritmu 2

```

1: for all  $message \in messages$  do
2:   if not  $(Vs\{MMSI\})$  then ▷ plovilo s tim MMSI ne postoji
3:      $Vs \leftarrow add(Vs\{MMSI\})$  ▷ to plovilo se dodaje listi plovila  $Vs$ 
4:      $Vs\{MMSI\}.status \leftarrow ('sailing')$  ▷ status mu se postavlja na "plovi"
5:      $Vs\{MMSI\}.track \leftarrow (x, y, COG, SOG, timestamp, etc.)$ 
   ▷ vremenski označena povijest i informacije vektora stanja

```

```

6:      [Rs, ENs, Vs{MMSI}] ← Online_WPs_Clustering                                ▶ algoritam 2
      (ENs, Vs{MMSI}, EpsENs, NENs)
7:      [Rs, Vs{MMSI}] ← Route_Objects_Manager(Rs, Vs{MMSI}) ▶ algoritam 3
8:      else                                ▶ plovilo s tim MMSI postoji i njegovi se parametri ažuriraju
9:      Vs{MMSI}.track(end + 1) ← (x, y, COG, SOG, timestamp, etc.)
10:     Vs{MMSI}.avg_speed = Δpos/Δt                                ▶ prosječna brzina plovila
11:     if Vs{MMSI}.avg_speed < min_speed and Vs{MMSI}.status = 'sailing' then
      ▶ plovilo je stalo i započinje klasteriranje stacionarnih točaka
12:     Vs{MMSI}.status ← ('stationary')
13:     [Rs, POs, Vs{MMSI}] ← Online_WPs_Clustering
      (POs, Vs{MMSI}, EpsPOs, NPOs)
14:     [Rs, Vs{MMSI}] ← Route_Objects_Manager(Rs, Vs{MMSI})
15:     end if
16:     if Vs{MMSI}.status = 'lost' then
      ▶ plovilo koje je bilo "izgubljeno" se opet pojavilo
17:     Vs{MMSI}.status ← ('sailing')
18:     [Rs, ENs, Vs{MMSI}] ← Online_WPs_Clustering
      (ENs, Vs{MMSI}, EpsENs, NENs)
19:     [Rs, Vs{MMSI}] ← Route_Objects_Manager(Rs, Vs{MMSI})
20:     end if
21:     end if
22:     if mod(timestamp, Δdays) = 0 then
      ▶ svakih Δdays traži plovila koja nisu ažurirana u zadnjem τ intervalu; ažuriraj listu
      izlaznih točaka
23:     for all v ∈ Vs do
24:     if v.last_update > τ and v.status ≠ ('lost') then
25:     v.status ← ('lost')
26:     [Rs, EXs, v] ← Online_WPs_Clustering(EXs, v, EpsEXs, NEXs)
27:     [Rs, v] ← Route_Objects_Manager(Rs, v)
28:     end if
29:     end for
30:     end if
31: end for
32: return Vs, EXs, ENs, POs, Rs

```

3.4 Primjena spoznaja

Jednom uočene prometne spoznaje, dobivene postupkom iz prethodnog dijela, mogu se iskoristiti u svrhu klasifikacije ruta, predviđanja budućih ruta kojima će se plovilo kretati i detektiranja anomalijskog ponašanja.

3.4.1 Klasifikacija ruta

Definicija 3.4.1 (σ -algebra).

Neka je Ω^4 neprazan skup. Familija podskupova \mathcal{F} od Ω zove se σ -algebra (ili σ -algebra događaja), ako vrijede sljedeća tri svojstva:

- (i) $\Omega \in \mathcal{F}$,
- (ii) Ako je $A \in \mathcal{F}$, onda je i $A^c \in \mathcal{F}$ (zatvorenost na komplement),
- (iii) Ako su $A_j \in \mathcal{F}$, $j \in \mathbb{N}$, onda je i $\bigcup_{j=1}^{\infty} A_j \in \mathcal{F}$ (zatvorenost na prebrojive unije).

Uređen par (Ω, \mathcal{F}) zove se izmjeriv prostor.

Definicija 3.4.2 (vjerojatnost).

Neka je Ω neprazan skup i \mathcal{F} σ -algebra događaja. Vjerojatnost na izmjerivom prostoru (Ω, \mathcal{F}) je funkcija $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ koja zadovoljava sljedeća tri aksioma:

- (A1) (nenegativnost) Za sve $A \in \mathcal{F}$, $\mathbb{P}(A) \geq 0$,
- (A2) (normiranost) $\mathbb{P}(\Omega) = 1$,
- (A3) (σ -aditivnost) Za svaki niz $(A_j)_{j \in \mathbb{N}}$ po parovima disjunktnih događaja $A_j \in \mathcal{F}$ ($A_i \cap A_j = \emptyset$ za $i \neq j$) vrijedi

$$\mathbb{P}\left(\bigcup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty} \mathbb{P}(A_j).$$

Uređena trojka $(\Omega, \mathcal{F}, \mathbb{P})$ zove se vjerojatnosni prostor.

Klasifikacija ruta dodjeljuje vjerojatnost svakoj ruti koja odgovara položaju plovila. To je reprezentirano aposteriornom vjerojatnošću⁵ da plovilo pripada toj određenoj ruti, promatrajući dijelove traga rute plovila.

⁴Pokus je svaka dobro definirana procedura. Rezultati (ili pojave) pokusa nazivaju se ishodi (ili elementarni događaji). Skup svih ishoda pokusa zove se prostor elementarnih događaja (engl. sample space) i tradicionalno se označava s Ω . Događaj (u Ω) je podskup prostora elementarnih događaja Ω [28].

⁵Aposteriorna vjerojatnost je vjerojatnost da će se neki događaj dogoditi nakon što se uzmu u obzir svi dokazi ili pozadinske informacije [25].

Općenito, trag rute plovila \mathbf{V} je vremenski niz T vektora stanja \mathbf{v}_i (povijest vektora stanja):

$$\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_T\}, \quad (3.2)$$

gdje je svaki \mathbf{v}_i izravno dobiven iz AIS poruke. U metodi TREAD \mathbf{v}_i je izveden iz $v.track$, uvedenom u prethodnom odjeljku, koji predstavlja povijest vremenskih opaženih informacija vektora stanja (parametara položaja i brzine) plovila.

Vrijedi

$$\mathbf{v}_i = [x_i, y_i, \dot{x}_i, \dot{y}_i]^T, \quad (3.3)$$

gdje su x_i i y_i koordinate plovila (predstavljaju parametar položaja vektora stanja), a \dot{x}_i i \dot{y}_i komponente brzine (predstavljaju parametar brzine vektora stanja). \dot{x}_i i \dot{y}_i mogu se izvesti iz kursa i brzine plovila, tj. iz SOG i COG informacije na sljedeći način:

$$SOG_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2},$$

$$COG_i = \text{tg}^{-1} \left(\frac{\dot{y}_i}{\dot{x}_i} \right).$$

Trag rute plovila \mathbf{V} može se povezati s vremenskim nizom područja $\bar{\mathbf{S}} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_T\}$, koja su prostorno identificirana kružnicama radijusa d i s centrom u promatranim koordinatama $[x_i, y_i]$, koje predstavljaju vremenski slijed stanja položaja između slijednih zapažanja. Između tih susjednih zapažanja uzeto je u obzir i vremensko kašnjenje između dva zapažanja, u oznaci Δ_i .

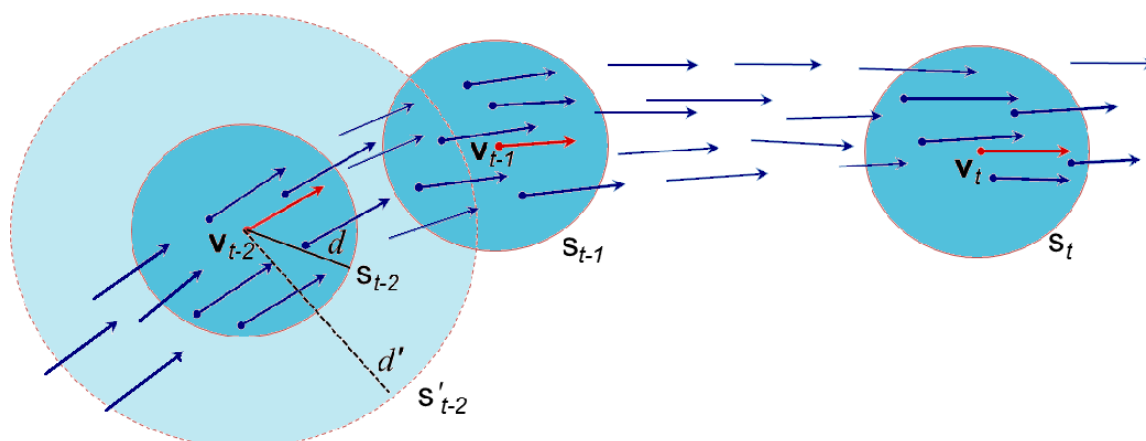
Prostorno područje \mathbf{s}_i , u nastavku samo područje, može se koristiti kako bi se uhvatili elementi rute u okolini promatranog vektora stanja \mathbf{v}_i te okarakterizirati lokalno ponašanje ruta. Bitan je dobar izbor radijusa, odnosno udaljenosti d čime se ujedno određuje i veličina područja \mathbf{s}_i .

Vrijedi:

- ako je udaljenost d premala: karakterizacija lokalnog ponašanja rute zasnovana je na manjoj okolini, pa samim time i smanjenom broju susjednih ruta što vodi do slabe mogućnosti generalizacije,
- ako je udaljenost d prevelika: postoji mogućnost nepristranosti, tj. mogućnost miješanja različitog ponašanja rute.

Izbor udaljenosti d prikazan je na slici 3.2. Promatraju se trag rute $\mathbf{V} = \{\mathbf{v}_{i-2}, \mathbf{v}_{i-1}, \mathbf{v}_i\}$ (vektori naznačeni crvenom bojom), pridruženi niz područja $\bar{\mathbf{S}} = \{\mathbf{s}_{i-2}, \mathbf{s}_{i-1}, \mathbf{s}_i\}$ (kružnice naznačene plavom bojom) i rute (vektori naznačeni tamno plavom bojom) dobivene postupkom otkrivanja prometnih spoznaja opisanog u prethodnom odjeljku. Može se vidjeti kako odabir prevelikog radijusa područja ($d' > d$) utječe na to da se različita lokalna ponašanja

rute uključuju u isto područje, mijenjajući vrijednosti karakterizacije kretanja relevantne promatrane okoline, pa samim time i postupak klasifikacije rute [24].



Slika 3.2: Odabir radijusa područja d [24]

Utvrđeno je da je odabir veličine d od nekoliko nautičkih milja⁶ daje prihvatljive rezultate klasifikacije [24].

Definicija 3.4.3 (uvjetna vjerojatnost).

Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor te $B \in \mathcal{F}$ događaj takav da je $\mathbb{P}(B) > 0$. Uvjetna vjerojatnost događaja A uz dano B definira se formulom

$$\mathbb{P}(A | B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Definicija 3.4.3 daje prikaz vjerojatnosti događaja A ukoliko se dogodio događaj B .

Teorem 3.4.4 (Bayesov teorem (Bayesova formula)).

Neka je $(H_i)_{i \in I}$ potpun sustav događaja⁷ na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Tada za svaki $A \in \mathcal{F}$ takav da je $\mathbb{P}(A) > 0$ vrijedi

$$\mathbb{P}(H_j | A) = \frac{\mathbb{P}(H_j) \mathbb{P}(A | H_j)}{\sum_{i \in I} \mathbb{P}(H_i) \mathbb{P}(A | H_i)}.$$

Događaji H_i nazivaju se hipotezama, originalne vjerojatnosti hipoteza $\mathbb{P}(H_i)$ apriornim vjerojatnostima, a vjerojatnost $\mathbb{P}(H_j | A)$ aposteriornom vjerojatnošću [28].

⁶Nautička milja ili morska milja je jedinica za mjerenje duljine korištena u pomorstvu i zrakoplovstvu.

⁷Potpun sustav događaja je konačna ili prebrojiva familija događaja $(H_i)_{i \in I}$ iz \mathcal{F} takva da je $\mathbb{P}(H_i) > 0$ za sve $i \in I$, $H_i \cap H_j = \emptyset$ za $i \neq j$ te $\bigcup_{i \in I} H_i = \Omega$ [28].

Propozicija 3.4.5 (formula potpune vjerojatnosti).

Neka je $(H_i)_{i \in I}$ potpun sustav događaja. Tada za svaki $A \in \mathcal{F}$ vrijedi

$$\mathbb{P}(A) = \sum_{i \in I} \mathbb{P}(H_i) \mathbb{P}(A | H_i).$$

Koristeći propoziciju 3.4.5, Bayesovu formulu možemo zapisati u obliku:

$$\mathbb{P}(H_j | A) = \frac{\mathbb{P}(H_j) \mathbb{P}(A | H_j)}{\mathbb{P}(A)}.$$

Iz svake AIS poruke može se izvesti tip plovila (npr: putničko plovilo, tanker i dr.), u oznaci c . Problem klasifikacije ruta tada postaje:

pronađi rutu $R_c^{k^*}$, koja maksimizira aposteriornu vjerojatnost $\mathbb{P}(R_c^k | \mathbf{V}, \bar{\mathbf{S}})$, iz $k = 1, 2, \dots, K$ kompatibilnih ruta $R_c^k \in Rs$, tj.

$$R_c^{k^*} = \arg \max_k \mathbb{P}(R_c^k | \mathbf{V}, \bar{\mathbf{S}}). \quad (3.4)$$

Koristeći Bayesov teorem (3.4.4), gornja vjerojatnost $\mathbb{P}(R_c^k | \mathbf{V}, \bar{\mathbf{S}})$ može se zapisati kao

$$\mathbb{P}(R_c^k | \mathbf{V}, \bar{\mathbf{S}}) \propto \mathbb{P}(\mathbf{V}, \bar{\mathbf{S}} | R_c^k) \mathbb{P}(R_c^k), \quad (3.5)$$

gdje \propto označava relaciju proporcionalnosti⁸.

Apriorna vjerojatnost $\mathbb{P}(R_c^k)$ može se empirijski procijeniti (empirijska razdioba) kao omjer broja plovila koji su prošli duž rute R_c^k i ukupnog broja identificiranih plovila u razmatranom području. Vjerojatnost $\mathbb{P}(\mathbf{V}, \bar{\mathbf{S}} | R_c^k)$ označava zajedničku vjerojatnost traga rute \mathbf{V} plovila, koji je vremenski niz opažanja, i niza područja $\bar{\mathbf{S}}$ uz danu rutu R_c^k .

Prema [24], vjerojatnost $\mathbb{P}(\mathbf{V}, \bar{\mathbf{S}} | R_c^k)$, iz (3.5), može se zapisati kao

$$\mathbb{P}(\mathbf{V}, \bar{\mathbf{S}} | R_c^k) = \mathbb{P}(\mathbf{V} | \bar{\mathbf{S}}, R_c^k) \mathbb{P}(\bar{\mathbf{S}} | R_c^k), \quad (3.6)$$

gdje je $\bar{\mathbf{S}}$ fiksiran nakon što je opažen trag rute plovila \mathbf{V} .

⁸Relacija proporcionalnosti između dvije varijable znači da promjena jedne varijable utječe na promjenu druge s konstantnim faktorom.

Vjerojatnost $\mathbb{P}(\mathbf{V} \mid \bar{\mathbf{S}}, R_c^k)$ niza opažanja \mathbf{V} za niz područja $\bar{\mathbf{S}}$ uz rutu R_c^k , iz izraza (3.6), može se zapisati kao

$$\mathbb{P}(\mathbf{V} \mid \bar{\mathbf{S}}, R_c^k) = \prod_{t=1}^T \mathbb{P}(\mathbf{v}_t \mid \mathbf{s}_t, R_c^k), \quad (3.7)$$

te se pretpostavlja da je vjerojatnost opažanja vektora stanja \mathbf{v}_t u jednom području \mathbf{s}_t nezavisna⁹ od opažanja istog vektora stanja u drugim područjima. Radi se o aproksimaciji jer su vektori stanja traga rute plovila \mathbf{V} vezani za isto plovilo, pa samim time i zavisni.

$\mathbb{P}(\mathbf{v}_t \mid \mathbf{s}_t, R_c^k)$ je vjerojatnost opažanja vektora stanja \mathbf{v}_t , s obzirom na elemente rute R_c^k , $\{R_c^k(l).[x, y, \dot{x}, \dot{y}]\}$, unutar područja \mathbf{s}_t , definirane s

$$\{R_c^k(l).[x, y, \dot{x}, \dot{y}]\}, \text{ gdje je } \|R_c^k(i).[x, y] - [x_t, y_t]\| \leq d, \forall i \in l. \quad (3.8)$$

Vjerojatnost $\mathbb{P}(\mathbf{v}_t \mid \mathbf{s}_t, R_c^k)$ izračunava se kao

$$\mathbb{P}(\mathbf{v}_t \mid \mathbf{s}_t, R_c^k) = \mathbb{P}(x_t, y_t, \dot{x}, \dot{y} \mid \mathbf{s}_t, R_c^k) = \mathbb{P}(\dot{x}, \dot{y} \mid x_t, y_t, \mathbf{s}_t, R_c^k) \mathbb{P}(x_t, y_t \mid \mathbf{s}_t, R_c^k), \quad (3.9)$$

gdje je $\mathbb{P}(\dot{x}, \dot{y} \mid x_t, y_t, \mathbf{s}_t, R_c^k)$ vjerojatnost opažanja/promatranja komponenta brzine \dot{x} , \dot{y} , unutar područja \mathbf{s}_t , identificiranih od strane susjeda trenutne pozicije $[x_t, y_t]$, unutar radijusa, tj. udaljenosti d . Drugim riječima, ta komponenta izraza pokazuje u kojoj mjeri je brzina plovila u liniji s prijašnjim brzinama i razdiobom lokalne frekvencije smjera, s obzirom na rutu R_c^k . Pošto su područja \mathbf{s}_t identificirana preko $[x_t, y_t]$ (centar kružnica radijusa d koje predstavljaju \mathbf{s}_t), vjerojatnost $\mathbb{P}(\dot{x}, \dot{y} \mid x_t, y_t, \mathbf{s}_t, R_c^k)$ može se pojednostaviti s $\mathbb{P}(\dot{x}, \dot{y} \mid \mathbf{s}_t, R_c^k)$ [24]. Dakle, izraz (3.9) se može zapisati kao

$$\begin{aligned} \mathbb{P}(\mathbf{v}_t \mid \mathbf{s}_t, R_c^k) &= \mathbb{P}(x_t, y_t, \dot{x}, \dot{y} \mid \mathbf{s}_t, R_c^k) = \mathbb{P}(\dot{x}, \dot{y} \mid x_t, y_t, \mathbf{s}_t, R_c^k) \mathbb{P}(x_t, y_t \mid \mathbf{s}_t, R_c^k) \\ &= \mathbb{P}(\dot{x}, \dot{y} \mid \mathbf{s}_t, R_c^k) \mathbb{P}(x_t, y_t \mid \mathbf{s}_t, R_c^k). \end{aligned} \quad (3.10)$$

Obje vjerojatnosti, $\mathbb{P}(\dot{x}, \dot{y} \mid \mathbf{s}_t, R_c^k)$ i $\mathbb{P}(x_t, y_t \mid \mathbf{s}_t, R_c^k)$, mogu se procijeniti koristeći neparametarske metode, kao što je na primjer KDE metoda (eng. Kernel Density Estimator) [24]. Prije opisa metode uvodi se nekoliko definicija [22].

Definicija 3.4.6 (slučajna varijabla i funkcija razdiobe).

Neka je $(\Omega, \mathcal{F}, \mathbb{P})$ vjerojatnosni prostor. Funkcija $X : \Omega \rightarrow \mathbb{R}$ zove se slučajna varijabla ako vrijedi $\{a \leq X \leq b\} \in \mathcal{F}$ za sve $a, b \in \mathbb{R}$, $a < b$. Slučajna varijabla koja poprima prebrojivo mnogo vrijednosti je diskretna slučajna varijabla.

Funkcija $F = F_X : \mathbb{R} \rightarrow [0, 1]$ definirana sa $F(a) = \mathbb{P}(X \leq a)$, $a \in \mathbb{R}$ naziva se funkcija razdiobe slučajne varijable X .

⁹Slučajne varijable X, Y su nezavisne ako vjerojatnost da Y poprimi neku vrijednost, ne ovisi o tome koju je vrijednost poprimila X (i obratno).

Definicija 3.4.7 (neprekidna slučajna varijabla i (vjerojatnosna) funkcija gustoće).
Slučajna varijabla X je neprekidna slučajna varijabla ako postoji nenegativna (izmjeriva)¹⁰ funkcija $f : \mathbb{R} \rightarrow \mathbb{R}$ takva da je

$$\mathbb{P}(a \leq X \leq b) = \int_a^b f(x) dx, \text{ za sve } a, b \in \mathbb{R}, a < b.$$

Funkcija f naziva se (vjerojatnosna) funkcija gustoće slučajne varijable X .

Definicija 3.4.8 (očekivanje).

Neka je X neprekidna slučajna varijabla s funkcijom gustoće f . Ako vrijedi da $\int_{-\infty}^{\infty} |x|f(x) dx$ konvergira, onda X ima matematičko očekivanje koje se definira kao

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} xf(x) dx.$$

Također vrijedi da ako je $g : \mathbb{R} \rightarrow \mathbb{R}$ (izmjeriva) funkcija, onda je i $g(X)$ slučajna varijabla s očekivanjem

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)f(x) dx.$$

Definicija 3.4.9 (varijanca, standardna devijacija).

Neka je X neprekidna slučajna varijabla s funkcijom gustoće f i očekivanjem $\mathbb{E}(X)$. Varijanca od X definira se kao

$$\text{Var}(X) = \mathbb{E}[(X - \mathbb{E}(X))^2] = \mathbb{E}[X^2] - (\mathbb{E}(X))^2.$$

Standardna devijacija od X je definirana kao $\sigma(X) = \sqrt{\text{Var}(X)} \geq 0$.

Definicija 3.4.10 (uzorak, statistika).

Slučajni uzorak duljine n za slučajnu varijablu X je niz od n nezavisnih i jednako distribuiranih slučajnih varijabli X_1, X_2, \dots, X_n koje imaju istu razdiobu kao i X .

Za $\omega \in \Omega$ je $x_1 = X_1(\omega), x_2 = X_2(\omega), \dots, x_n = X_n(\omega)$ jedna realizacija slučajnog uzorka i naziva se uzorak.

Statistika je funkcija slučajnog uzorka.

¹⁰Neka su (X, \mathcal{F}) i (Y, \mathcal{G}) dva izmjeriva prostora. Funkcija $f : X \rightarrow Y$ je izmjeriva u paru σ -algebri \mathcal{F} i \mathcal{G} ako vrijedi $f^{-1}(\mathcal{G}) \subseteq \mathcal{F}$, tj. $(\forall A \in \mathcal{G})(f^{-1}(A)) \in \mathcal{F}$ [17].

U statistici, KDE (procjenitelj gustoće jezgre) metoda neparametarski je način za procjenu vjerojatnosne funkcije gustoće slučajne varijable, koja je u ovom slučaju položaj i parametri brzine plovila. Procjena vjerojatnosne funkcije gustoće f u KDE metodi je funkcija \hat{f} koja se naziva procjenitelj gustoće jezgre. Funkcija \hat{f} definirana je sa

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - X_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right),$$

gdje je K funkcija jezgre (eng. kernel) koja zadovoljava uvjet $\int_{-\infty}^{\infty} K(x) dx = 1$, a $h > 0$ parametar zaglađivanja (eng. bandwidth) [20]. U ovom slučaju uzima se normalna funkcija jezgre definirana s

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-(1/2)t^2}$$

i optimizirani odabir parametra zaglađivanja h dobivenog minimizacijom funkcije troška (eng. cost/loss/risk function). Optimalan izbor za h u tom slučaju je minimizacija funkcije troška, koja se još naziva i integrirana srednjekvadratna greška, definirana na sljedeći način

$$MISE(h) = \mathbb{E} \left[\int (\hat{f}_h(x) - f(x))^2 dx \right].$$

Time se h može izračunati kao

$$h = \left(\frac{4\hat{\sigma}^5}{3n} \right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}n^{-1/5},$$

gdje je n veličina uzorka, a $\hat{\sigma}$ standardna devijacija uzoraka.

Druga vjerojatnost $\mathbb{P}(\bar{\mathbf{S}} | R_c^k)$, iz izraza (3.6), predstavlja vjerojatnost niza područja $\bar{\mathbf{S}}$ uz danu rutu R_c^k , te se može rastaviti na sljedeći način

$$\mathbb{P}(\bar{\mathbf{S}} | R_c^k) \propto \mathbb{P}(\mathbf{s}_2 | \mathbf{s}_1, R_c^k) \mathbb{P}(\mathbf{s}_3 | \mathbf{s}_2, R_c^k) \dots \mathbb{P}(\mathbf{s}_T | \mathbf{s}_{T-1}, R_c^k), \quad (3.11)$$

pri čemu proporcionalnost slijedi iz pretpostavke da je vjerojatnost početnog područja $\mathbb{P}(\mathbf{s}_1 | R_c^k)$ jednaka za sve moguće sekvence područja u R_c^k [24]. Drugim riječima, niz je podjednako vjerojatan da započne u bilo kojoj točki rute. U izrazu (3.11) uzima se u obzir kompatibilnost niza područja $\{\mathbf{s}_{t-1}, \mathbf{s}_t\}$ s rutom R_c^k , uzimajući u obzir promjenjivost brzine osvježavanja AIS podataka. Estimacija se vrši kao funkcija udaljenosti Δ_p , između promatranog položaja $[x_t, y_t]$ (središte kružnice koja predstavlja područje \mathbf{s}_t) i predviđenog položaja $[\hat{x}_t, \hat{y}_t]$ izračunatog propagacijom položaja $[x_{t-1}, y_{t-1}]$ do trenutnog vremena t , s obzirom na razdiobu brzina na ruti R_c^k [24].

Prikazani postupak iskazan je Algoritmom 5 na sljedećoj stranici.

Algoritam 5 Predviđanje traga (*Track Predictor*)[24]

Require: $R_c^k, [x_{t-1}, y_{t-1}], timestamp_t, timestamp_{t-1}, step_t, Eps$

```

1:  $\Delta_t \leftarrow (timestamp_t - timestamp_{t-1})$ 
2:  $\delta_\tau \leftarrow (\Delta_t) / \lceil (\Delta_t) / step_t \rceil$ 
3: for  $\tau = timestamp_{t-1}$  to  $timestamp_t - \delta_\tau$  step  $\delta_\tau$  do
4:   find  $l$  so that  $\forall i \in l : \|R_c^k(i).[x, y] - [x_\tau, y_\tau]\| \leq Eps$ 
5:    $s_\tau \leftarrow \{R_c^k(l).[x, y, \dot{x}, \dot{y}]\}$ 
6:    $[\dot{x}_{s_\tau}, \dot{y}_{s_\tau}] \leftarrow median(s_\tau.[\dot{x}, \dot{y}])$ 
7:    $[x_{\tau+1}, y_{\tau+1}] \leftarrow [x_\tau, y_\tau] + [\dot{x}_{s_\tau}, \dot{y}_{s_\tau}]\delta_\tau$ 
8: end for
9: return  $[\hat{x}_t, \hat{y}_t]$ 

```

U Algoritmu 5, $\lceil \cdot \rceil$ predstavlja funkciju najmanjeg cijela¹¹ ("strop"), a δ_τ približno određeno povećanje vremena između dva položaja, koje se odabire proizvoljno ovisno o kompleksnosti rute.

Udaljenost Δ_p može se iskoristiti za procjenu vjerojatnosti opažanja područja s_t , s obzirom na prethodno područje s_{t-1} i rutu R_c^k . Δ_p se može se posmatrati kao slučajna varijabla koja opisuje pogrešku predviđanja, odnosno pomak trenutno promatranog položaja u odnosu na očekivani, uzimajući u obzir vremensko kašnjenje Δ_t . Udaljenost Δ_p izračunava se kao euklidska udaljenost¹², $\Delta_p = \|[x_t, y_t] - [\hat{x}_t, \hat{y}_t]\|$, jer je većina promatranih udaljenosti uglavnom ispod osam nautičkih milja, sa smanjenom zakrivljenosti [24]. Za istraživanje varijabilnosti Δ_p izabrana je parametarska Weibull-ova razdioba [24], gdje je vjerojatnosna funkcija gustoće slučajne varijable x definirana s

$$f(x; \alpha, \beta) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} e^{-(x/\alpha)^\beta} & x \geq 0 \\ 0 & x < 0 \end{cases},$$

gdje se $\beta > 0$ zove parametar oblika, $\alpha > 0$ parametar skaliranja, a $e^{-(x/\alpha)^\beta}$ funkcija pouzdanosti¹³ [7, 8].

Primjenom Weibull-ovog modela, vjerojatnost $\mathbb{P}(s_t | s_{t-1}, R_c^k)$ se izjednačava s funkcijom pouzdanosti [24], te se zapisuje kao

$$\mathbb{P}(s_t | s_{t-1}, R_c^k) = \exp \left[- \left(\frac{\Delta_p}{\alpha_k} \right)^{\beta_k} \right]. \quad (3.12)$$

¹¹ $\lceil x \rceil$ - najmanji cijeli broj koji nije manji od x , tj. $\lceil x \rceil = \min \{n \in \mathbb{Z} | n \geq x\}$

¹²Za vektore definirana kao $d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, $x, y \in \mathbb{R}^n$.

¹³Funkcija pouzdanosti određuje vjerojatnost opstanka slučajne varijable x .

Parametar oblika β_k se u principu ne mijenja s vremenom, dok parametar skaliranja α_k ovisi o vremenskom kašnjenju Δ_t , koje se još naziva i vremenski prozor, između dva susjedna opažanja [24]. Vrijedi:

$$\alpha_k = m_k \Delta_t, \text{ za } \Delta_t > 0. \quad (3.13)$$

Očekivana vrijednost (sredina) slučajne varijable Δ_p , u Weibull-ovoj razdiobi, je

$$\mathbb{E}(\Delta_p) = \alpha_k \cdot \Gamma\left(1 + \frac{1}{\beta_k}\right), \quad (3.14)$$

a varijanca

$$\text{Var}(\Delta_p) = \alpha_k^2 \cdot \Gamma\left(1 + \frac{2}{\beta_k}\right) - (\mathbb{E}(\Delta_p))^2, \quad (3.15)$$

gdje je $\Gamma : \langle 0, \infty \rangle \rightarrow \mathbb{R}$ gama funkcija definirana s $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$.

Zbog jednadžbe (3.13), varijanca se povećava za Δ_t^2 , računajući rast nesigurnosti povezane s modelom propagacije u dugoročnom predviđanju.

Definicija 3.4.11 (nepistrani procjenitelj).

Procjenitelj $T_n = f_n(X_1, X_2, \dots, X_n)$ je nepistrani procjenitelj za parametar τ ako vrijedi $\mathbb{E}(T_n) = \tau$.

Procjene $\hat{\alpha}_k$ i $\hat{\beta}_k$ dobivaju se pomoću uzoraka udaljenosti između predviđenih položaja $[\hat{x}_t, \hat{y}_t]$ i stvarnih promatranih položaja $[x_t, y_t]$ u navedenoj ruti R_c^k , za svako vremensko kašnjenje Δ_t . Za procjenu parametara koristi se metoda maksimalne vjerodostojnosti (eng. Maximum Likelihood Method). Neka je (x_1, x_2, \dots, x_n) opaženi uzorak za slučajnu varijablu X s funkcijom gustoće $f(x; \theta)$, gdje je $\theta = (\theta_1, \theta_2, \dots, \theta_k) \in \Theta \subseteq \mathbb{R}^k$ nepoznati parametar. Funkcija vjerodostojnosti $L : \Theta \rightarrow \mathbb{R}$ definirana je s

$$L(\theta) = \prod_i^n f(x_i; \theta), \quad \theta \in \Theta.$$

Vrijednost $\hat{\theta} = \hat{\theta}(x_1, x_2, \dots, x_n) \in \Theta$ za koju je

$$L(\hat{\theta}) = \max_{\theta \in \Theta} L(\theta)$$

naziva se procjena metodom maksimalne vjerodostojnosti, gdje je statistika $\hat{\theta}(x_1, x_2, \dots, x_n)$ procjenitelj metodom maksimalne vjerodostojnosti (kraće MLE) [10, 26].

Praktična procjena \hat{m}_k parametra m_k iz izraza (3.13) može se dobiti izravno pomoću linearne regresije za svaku rutu R_c^k . Time jednadžba (3.12) postaje

$$\mathbb{P}(\mathbf{s}_t \mid \mathbf{s}_{t-1}, R_c^k) = \exp\left[-\left(\frac{\Delta_p}{\hat{m}_k \cdot \Delta_t}\right)^{\hat{\beta}_k}\right], \text{ za } \Delta_t > 0. \quad (3.16)$$

Dakle, za dano vremensko kašnjenje Δ_t , vjerojatnost $\mathbb{P}(\mathbf{s}_t | \mathbf{s}_{t-1}, R_c^k)$ se smanjuje kako se udaljenost Δ_p povećava. Također, za danu udaljenost Δ_p , vjerojatnost $\mathbb{P}(\mathbf{s}_t | \mathbf{s}_{t-1}, R_c^k)$ se povećava kako se vremensko kašnjenje Δ_t povećava [24].

3.4.2 Predviđanje ruta

Za promatrani niz vektora stanja plovila određenog tipa c , klasifikacija ruta svakoj kompatibilnoj ruti dodjeljuje vjerojatnost temeljenu na aposteriornoj vjerojatnosti (izraz 3.5) da to plovilo pripada toj ruti. Drugim riječima, uzimajući u obzir zadnji niz vektora stanja plovila i vremensko kašnjenje (vremenski prozor) Δ_t , može se predvidjeti budući položaj plovila koristeći Algoritam 5.

Uz pretpostavku nepostojanja anomalija u promatranom području, prosječna brzina zajedno s nizom ruta dobivenih od prijašnjih plovila predstavljaju skup ograničenja koji se može iskoristiti za efikasno predviđanje budućih položaja plovila, temeljeno na statičkim informacijama, kao što je tip plovila. Zaključak se donosi na temelju ruta dobivenih učenjem opisanom u procesu otkrivanja prometnih spoznaja i najvjerojatnijih ruta dobivenih pomoću jednadžbe (3.7) [24].

3.4.3 Detekcija anomalija

Detekcija anomalije H_1 , u vremenu t , može se smatrati odstupanjem od normalnog ponašanja H_0 , koje dobivamo iz procesa otkrivanja prometnih spoznaja.

Detekciji se može pristupiti na način da se postavi minimalni prag Th za aposteriornu vjerojatnost (3.4):

$$R_c^{k*} = \arg \max_k \mathbb{P}(R_c^k | \mathbf{V}, \bar{\mathbf{S}}) \stackrel{H_1}{\underset{H_0}{\gtrless}} Th \quad (3.17)$$

Detekcija se obavlja u stvarnom vremenu, dakle s nedavnim podacima, koristeći klizni vremenski prozor koji hvata nedavna stanja niza traga rute plovila [24].

3.5 Razmatranje obilježja TREAD metode

Pristup koji je ovdje predstavljen je praktični kompromis kako bi se dobila pouzdana reprezentacija pomorskog prometa bez povećanja kompleksnosti modela. Praktična prednost je u tome što se TREAD može lako nositi s nejednakim duljinama ruta ili rutama s prazninama, što zna biti često zbog raznoraznih slučajeva kao što su: namjerno isključenje AIS uređaja, različitost intervala javljanja različitih tipova AIS uređaja i slično.

Rute dobivene iz procesa otkrivanja prometnih spoznaja, čiji se broj ne zna unaprijed,

opisane su informacijama o ulaznom/izlaznom vremenu plovila koje je ušlo/izašlo iz područja koje se promatra, zajedno s tipom tog plovila, što pruža više informacija (tip plovila, razdioba ruta, prosječno vrijeme putovanja, ...).

Identificirane rute ponekad imaju iste ulazne i izlazne točke, ali njihov put odstupa od ostalih putova unutar iste rute. Takvi uzorci nisu valjani te se za daljnju primjenu spoznaja moraju ukloniti. U tu svrhu pogodan je odabrani DBSCAN algoritma, koji takve uzorke smatra šumovima te se automatski filtriraju u samom algoritmu.

Učinak učenja TREAD metodologije analiziran je s obzirom na omjer između broja AIS poruka preslikanih u izdvojeni sustav ruta i broja obrađenih AIS poruka. Dobivene su spoznaje da što je promet više ograničen ili reguliran, to će točniji biti rezultati nenadgledanog učenja. Također izbor inkrementalnog učenja omogućuje aplikaciju metode u stvarnom vremenu. Nadalje se raspravljaju odabiri metoda u procesu primjene spoznaja. Za neparametarsku metodu, za procjenu vjerojatnosti položaja i brzine plovila, izabrana je KDE metoda jer ona za razliku od GMM metode (eng. Gaussian Mixture Models), koja je popularna u radovima prepoznavanja uzoraka, nema pretpostavke o vjerojatnosnoj funkciji gustoće (GMM pretpostavlja normalnu aka. Gauss-ovu razdiobu) i ne zahtjeva unaprijed specificiran broj komponenti, što je glavna mana GMM modela, čime je pokazala vrhunsku sposobnost preciznog modeliranja.

Za parametarsku metodu, za istraživanje varijabilnosti Δ_p , izabran je Weibull-ov model razdiobe jer eksponencijalni modeli pokazuju dobru prilagodbu, a ujedno se podudaraju s povezanim literaturama. Weibull-ov model pokazuje dobru korelaciju¹⁴ s empirijskom razdiobom opaženih udaljenosti temeljenim na stvarnim AIS podacima u različitim područjima [24].

¹⁴Korelacija predstavlja suodnos ili međusobnu povezanost između različitih pojava predstavljenih vrijednostima dviju varijabli.

Poglavlje 4

Programska izvedba metode TREAD u programskom jeziku R

4.1 Uvod u R

R je okruženje za statističke izračune i vizualizaciju sa zasebnim programskim jezikom. R je slobodno okruženje što znači da se može slobodno koristiti i distribuirati. R pruža širok izbor statističkih metoda za linearno i nelinearno modeliranje, klasične statističke testove, analize vremenskih serija, klasteriranje. Lako je proširiv s velikim izborom grafičkih tehnika. Dostupan je putem Interneta i radi na raznim platformama UNIX i sličnim sustavima (uključujući FreeBSD i Linux), Windows i MacOS [29].

Osnovne funkcionalnosti R-a mogu se proširiti dodavanjem knjižica. Knjižice se instaliraju preko R konzole koristeći sljedeću naredbu:

```
> install.packages("package_name")
```

Da bi funkcije instalirane knjižice bile dostupne potrebno ih je svaki puta prilikom početka rada u sustavu učitati u radni prostor naredbom:

```
> library(package_name)
```

Za više informacija o svakoj knjižici koriste se naredbe:

```
> help(package_name)
```

```
> ?package_name
```

4.2 Ulazni podaci

AIS podaci za rad dobiveni su sa stanica pomorskih vlasti Kraljevine Danske i dostupni su na internetskoj stranici [4].

Dostupni su samo povijesni podaci, te su za potrebe ovog rada uzeti podaci za jedan dan i to točnije 29.09.2019.

4.2.1 Struktura podataka

Podaci su preuzeti u obliku CSV (eng. Comma Separated Values) datoteke koja predstavlja tekstualnu datoteku koja sadrži popis podataka odvojenih zarezom.

Stupci reprezentiraju sljedeće:

- Timestamp - vremenska oznaka AIS bazne stanice, format: 31/12/2015 00:00:00,
- Type of mobile - tip AIS uređaja,
- MMSI - MMSI broj plovila,
- Latitude, Longitude - geografska širina i dužina, format: 57,8794,
- Navigational status - navigacijski status iz AIS poruke ako je dostupan (npr. bavljenje ribolovom),
- ROT, SOG, COG - brzina okreta, brzina plovljenja i kurs plovila ako su dostupni,
- Heading - smjer kretanja,
- IMO - IMO broj plovila,
- Callsign, Name - pozivni znak i ime plovila,
- Ship type, Cargo type - tip plovila i tip tereta,
- Width, Length - dimenzije plovila,
- Type of position fixing device - vrsta uređaja za pričvršćivanje položaja,
- Draught - okomiti razmak između vodene linije i dna trupa,
- Destination, ETA - destinacija i procijenjeno vrijeme dolaska,
- Data source type - izvor podataka, npr. AIS,
- A, B, C, D - duljina od GPS-a do pramca, krme, daljnje bočne strane, bliže bočne strane.

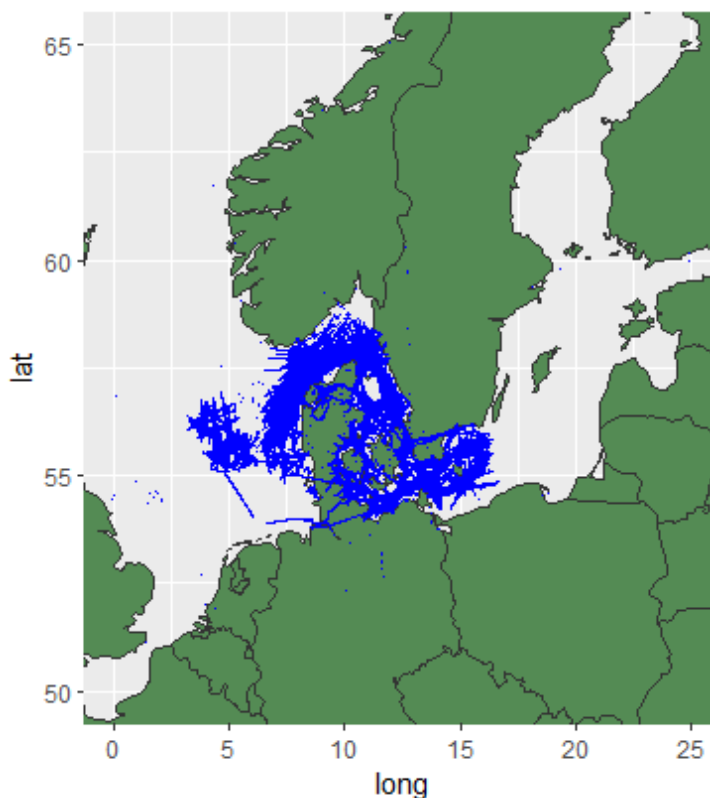
Timestamp, Type of mobile, Navigational status, IMO, Callsign, Name, Ship type, Cargo type, Type of position fixing device, Destination, ETA, Data Source type su podaci tipa text (eng. character), dok su ostali tipa realnog broja (eng. double).

4.2.2 Pregled podataka

Podaci se učitavaju u R korištenjem `readr` knjižice, dok se za grafički prikaz koristi `ggplot2` knjižica. Kod izvedbe za sve grafičke prikaze može se naći u dodatku C.

Da bi se steklo malo više uvida u podatke koristi se naredba (čime se dobiva prikaz 4.1):

```
> summary()
```



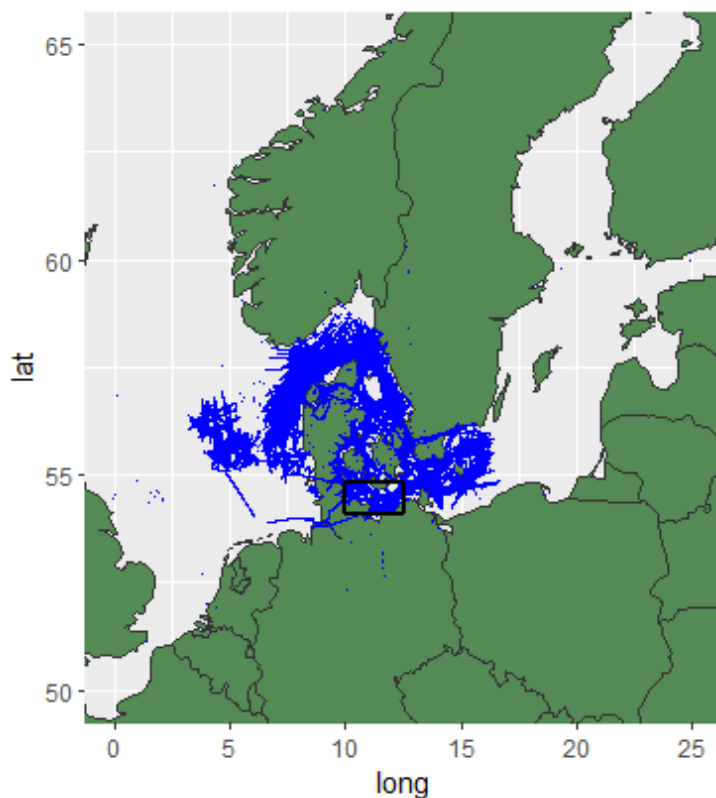
Slika 4.1: Grafički prikaz svih podataka za dan 29.09.2019.

1	Timestamp	Type of mobile	MMSI	Latitude
2				
3	Length :10181117	Length:10181117	Min. : 0	Min. : -110.96
4	Class :character	Class :character	1st Qu.:219002857	1st Qu.: 55.32
5	Mode :character	Mode :character	Median :219076000	Median : 56.04
6			Mean :243837272	Mean : 56.65
7			3rd Qu.:255801670	3rd Qu.: 57.16
8			Max. :992651016	Max. : 91.00
9				
10	Longitude	Navigational status	ROT	SOG
11				
12	Min. : -223.575	Length :10181117	Min. : -709	Min. : 0.0
13	1st Qu.: 9.165	Class :character	1st Qu.: 0	1st Qu.: 0.0
14	Median : 10.599	Mode :character	Median : 0	Median : 0.7
15	Mean : 12.942		Mean : 0	Mean : 5.4
16	3rd Qu.: 12.072		3rd Qu.: 0	3rd Qu.: 10.4
17	Max. : 181.000		Max. : 709	Max. : 211.0
18			NA :3330027	NA :579028
19				
20	COG	Heading	IMO	Callsign
21				
22	Min. : 0.0	Min. : 0.0	Length:10181117	Length:10181117
23	1st Qu.: 76.2	1st Qu.: 86.0	Class :character	Class :character
24	Median :202.2	Median :196.0	Mode :character	Mode :character
25	Mean :181.9	Mean :181.9		
26	3rd Qu.:272.7	3rd Qu.:260.0		
27	Max. :359.9	Max. :510.0		
28	NA :969849	NA :2412937		
29				
30	Name	Ship type	Cargo type	Width
31				
32	Length:10181117	Length:10181117	Length:10181117	Min. : 1.0
33	Class :character	Class :character	Class :character	1st Qu.: 6.0
34	Mode :character	Mode :character	Mode :character	Median : 10.0
35				Mean : 13.2
36				3rd Qu.: 18.0
37				Max. :100.0
38				NA :902857
39				
40	Length	Type of position fixing device	Draught	Destination
41				
42				
43	Min. : 1.0	Length:10181117	Min. : 0.2	Length:10181117
44	1st Qu.: 17.0	Class :character	1st Qu.: 2.7	Class :character
45	Median : 38.0	Mode :character	Median : 4.2	Mode :character
46	Mean : 70.4		Mean : 4.8	
47	3rd Qu.:101.0		3rd Qu.: 6.4	
48	Max. :400.0		Max. :25.5	
49	NA :896298		NA :2645401	
50				
51	ETA	Data source type	A	B
52				
53	Length:10181117	Length:10181117	Min. : 1.0	Min. : 1.0
54	Class :character	Class :character	1st Qu.: 10.0	1st Qu.: 7.0
55	Mode :character	Mode :character	Median : 19.0	Median : 13.0
56			Mean : 43.7	Mean : 27.2
57			3rd Qu.: 70.0	3rd Qu.: 26.0
58			Max. :288.0	Max. :283.0
59			NA :924452	NA :1000129
60				
61	C	D		
62				
63	Min. : 1.0	Min. : 1.0		
64	1st Qu.: 3.0	1st Qu.: 3.0		
65	Median : 5.0	Median : 5.0		
66	Mean : 6.7	Mean : 6.6		
67	3rd Qu.: 9.0	3rd Qu.: 9.0		
68	Max. :45.0	Max. :63.0		
69	NA :981374	NA :1059647		

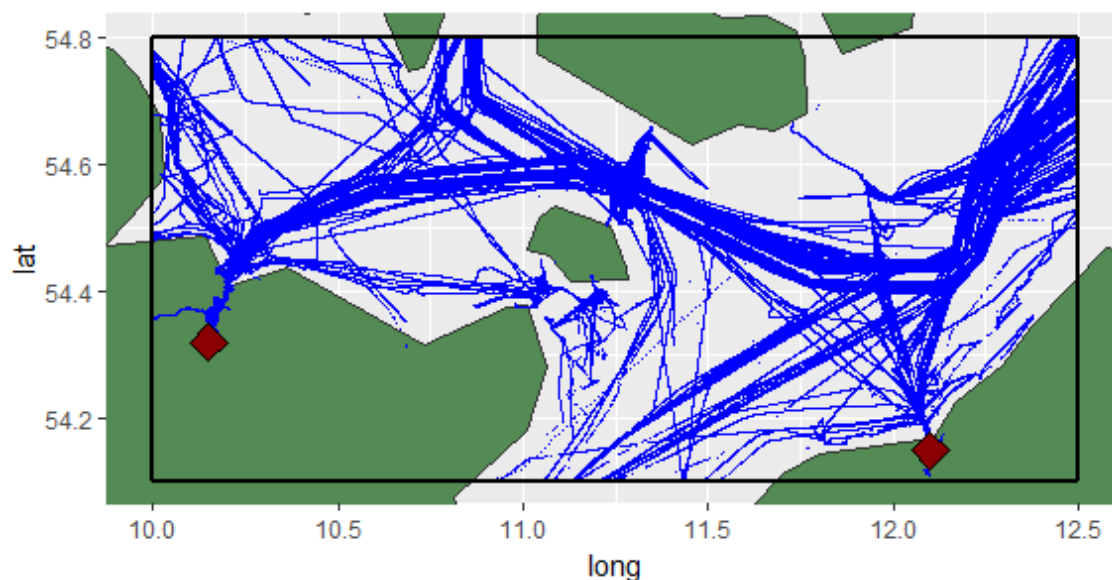
Prikaz 4.1: Sažetak AIS podataka

Napomena 4.2.1. Uočava se vrlo velik broj *NA* vrijednosti što znači lošiju kvalitetu podataka. Zbog važnosti *COG* i *SOG* parametra u klasifikaciji ruta, iz podataka se brišu svi oni kojima su ti parametri jednaki *NA*.

Odabire se promatrano područje/okvir. Uzima se okvir za promatranje koji obuhvaća luke Kiel i Rostok, geografske širine između 54.1 i 54.8, te geografske dužine između 10 i 12.5.



Slika 4.2: Odabir okvira



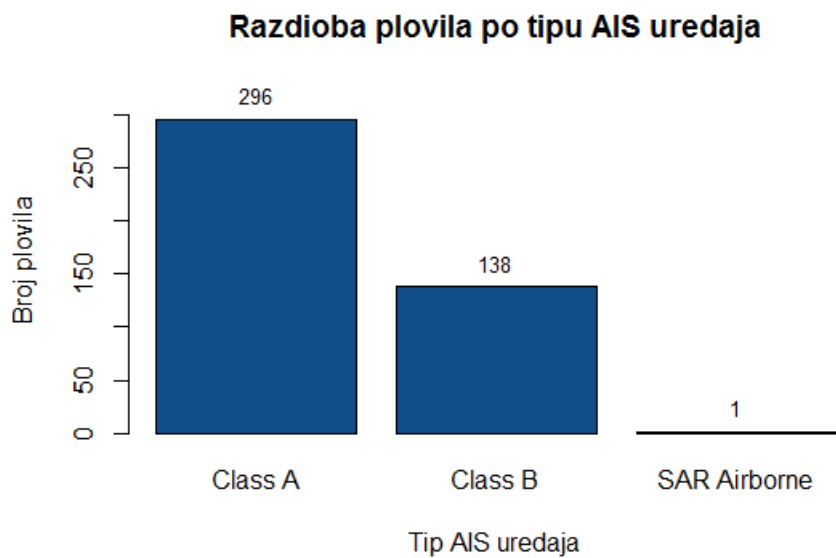
Slika 4.3: Prikaz promatranog područja sa svim podacima za dan 29.09.2019.

Zbog uzimanja podataka samo jednog dana, točnije 29.09.2019., varijabla Timestamp se mijenja iz DateTime formata u Time format, jer je datum za sve podatke isti.

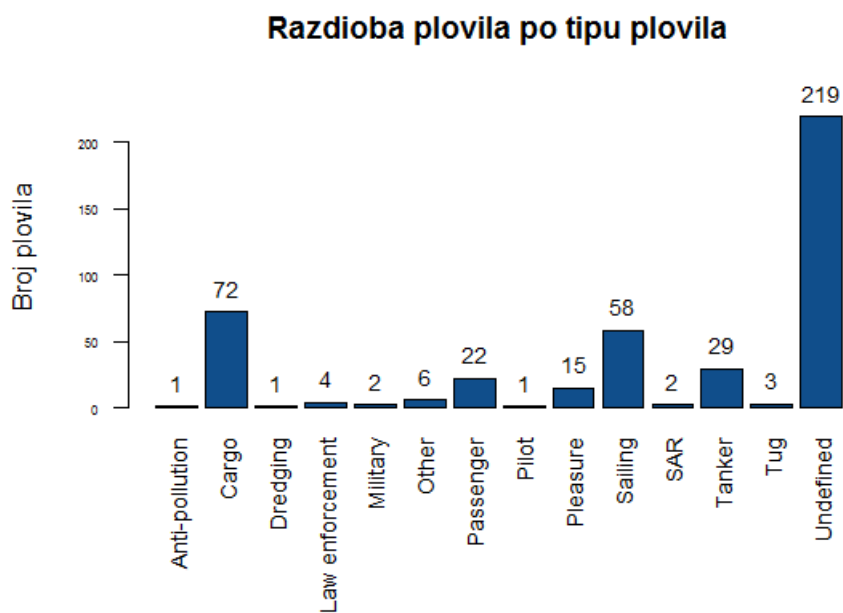
Daljnijim promatranjem podataka uočeno je kako se neki podaci ne podudaraju. Naime događa se da postoje podaci takvi da se određeno plovilo u isto vrijeme (do na sekundu) nalazi u dva različita položaja (različite vrijednosti varijabli Longitude i Latitude), što je nemoguće, pa se takvi podaci otklanjaju iz početnog skupa podataka.

Radi jednostavnosti, uzeti su samo najpotrebniji podaci (Timestamp, MMSI, Longitude, Latitude, SOG, COG, Name, Ship type) za nastavak rada.

U nastavku slijede grafički prikazi broja plovila, unutar promatranog područja, po vrsti AIS uređaja i po tipu plovila.



Slika 4.4: Broj plovila po tipu AIS uređaja



Slika 4.5: Broj plovila po tipu plovila

4.3 Rezultati izvedbe

Parametar τ , koji označava vremenski interval veći od maksimalne stope ažuriranja AIS poruka, postavljen je na vrijednost od 6 minuta (00:06:00 u hh:mm:ss formatu). Dakle, ukoliko plovilo nije promatrano 6 ili više minuta, njegov status se ažurira na "izgubljen".

Po algoritmu nenadgledanog izdvajanja ruta (Algoritam 4) svakih Δ_{days} se traže plovila koja nisu ažurirana u zadnjem τ intervalu. Pošto je za potrebe ovog rada uzet samo jedan dan (29.09.2019.), parametar Δ_{days} postavljen je u skladu s promatranjem samo jednog dana i iznosi 12 sati (12:00:00 u hh:mm:ss formatu).

Minimalna brzina za koju se detektira stacionaran objekt postavljena je na 4 m/s, što iznosi otprilike 7.8 čvora.

Navedeni parametri postavljeni su kao globalne varijable.

Nenadgledano izdvajanje ruta poziva se naredbom:

```
> UnsupervisedRouteExtraction(N_ENs, N_POs, N_EXs, Eps_ENs,  
                               Eps_POs, Eps_EXs)
```

Argumenti, odnosno parametri N_ENs, N_POs, N_EXs, Eps_ENs, Eps_POs i Eps_EXs, iz algoritma nenadgledanog izdvajanja ruta (Algoritam 4), za potrebe DBSCAN algoritma, određeni su eksperimentalno unutar promatranog područja.

Uzete su sljedeće vrijednosti tih parametara:

$$N_ENs = 4,$$

$$N_POs = 25,$$

$$N_EXs = 3,$$

$$Eps_ENs = 0.035,$$

$$Eps_POs = 0.2,$$

$$Eps_EXs = 0.2.$$

Funkcija također koristi globalne varijable *VS*, *RS* u obliku *data.frame*¹ objekta opisanih u nastavku i *data.frame* objekt *messages* koji predstavlja AIS poruke u promatranom području.

Funkcija vraća *data.frame* objekt *VS* koji sadrži podatke o plovilima u promatranom području, njihove statuse, podatke o položajima plovila, brzini, kursu, putnim točkama kroz koje je prošao ili u kojima je stacioniran u skladu s Algoritmom 4. Nadalje, funkcija vraća *data.frame* objekt *RS* koji sadrži izdvojene podatke o rutama, naziv rute (oblika "početna točka završna točka"), početne i završne točke i sve podatke o plovilima koja su zabilježena da plove tom rutom. Uz to globalno se ažuriraju i tri *data.frame* objekta *ENS*, *POS*, *EXS* koji predstavljaju ulazne, stacionarne i izlazne putne točke te sadrže položaje (Longitude, Latitude), MMSI plovila koji je zabilježen u toj putnoj točki i stupac *cluster* koji označava kojoj putnoj točki, unutar ulaznih, stacionarnih i izlaznih točaka, položaj pripada (npr. u tablici *ENS*, ako jedan jedan redak ima vrijednost stupca *cluster* jednak 3, znači da položaj iz tog retka pripada ulaznoj točki 3 (*EN 3*)).

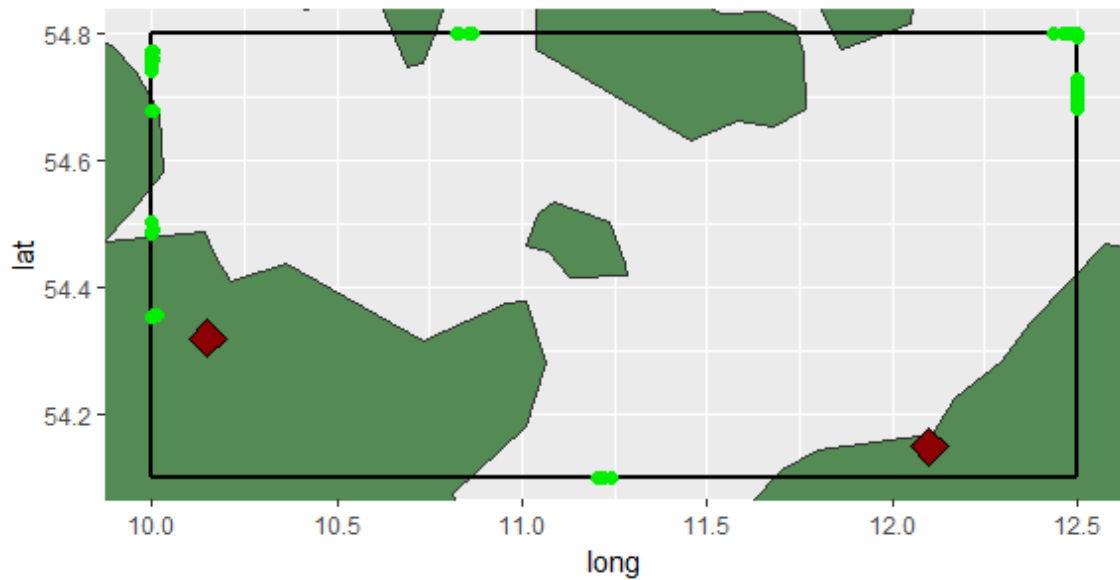
Provođenjem nenadgledanog izdvajanja ruta, uz parametre navedene u ovom potpoglavlju, dobivaju se ulazne, stacionarne i izlazne točke, te rute između njih.

Broj otkrivenih putnih točaka prikazan je sljedećom tablicom.

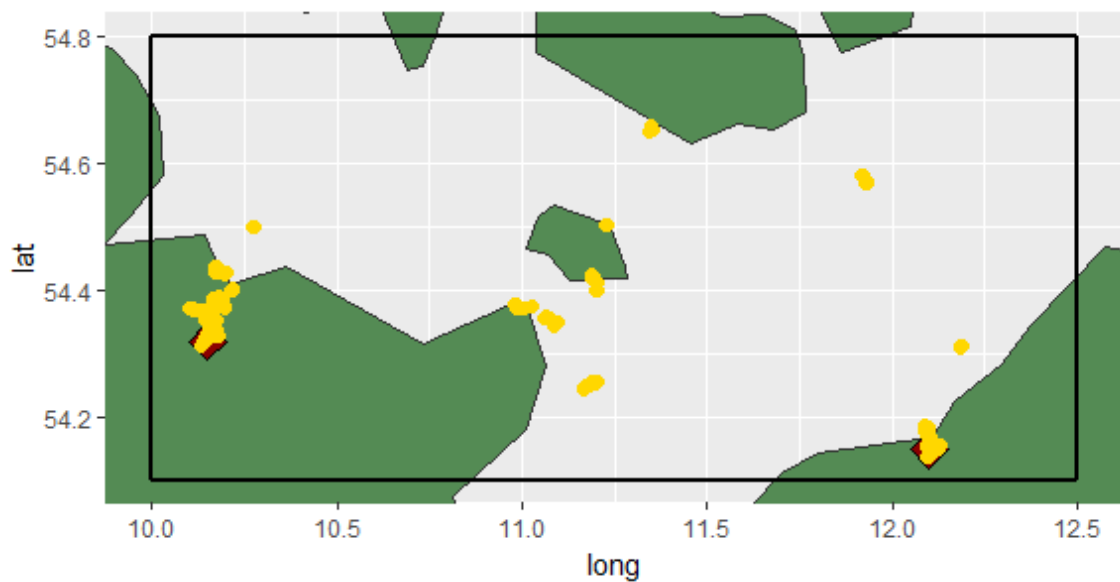
Objekt	Ukupno
Ulazne točke	8
Stacionarne točke	13
Izlazne točke	8

Sljedeći grafički prikazi prikazuju otkrivene ulazne točke (zelena boja), stacionarne točke (žuta boja) i izlazne točke (crvena boja). Zadnji grafički prikaz prikazuje sve putne točke (gdje iste boje označavaju vrstu putne točke kao i u prethodnim prikazima) i rute između njih (plava boja).

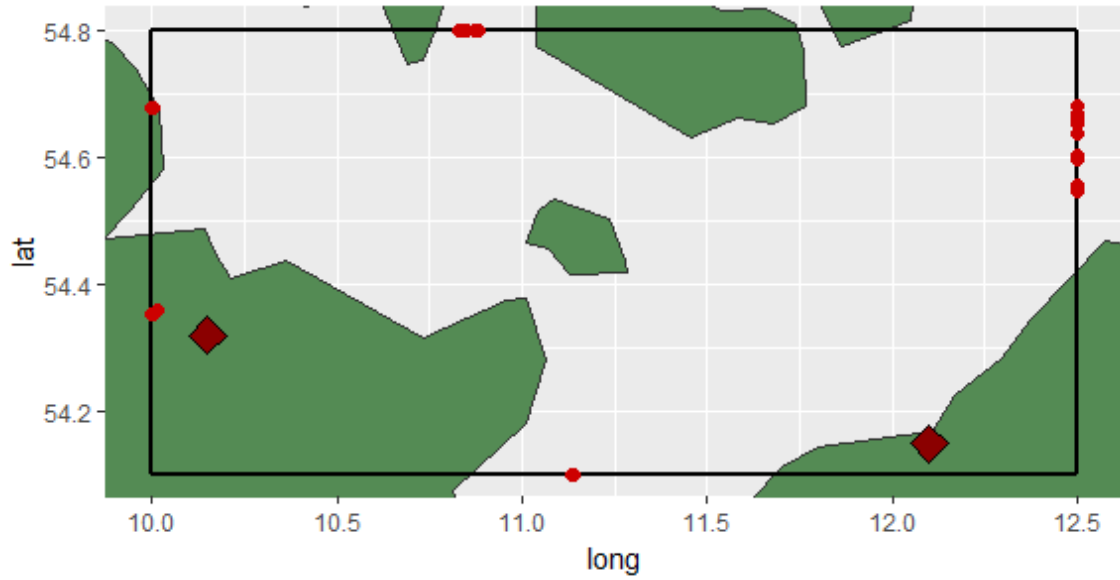
¹Tablica ili dvodimenzionalna struktura slična nizu u kojoj svaki stupac sadrži vrijednosti jedne varijable, a svaki redak sadrži po jedan skup vrijednosti iz svakog stupca.



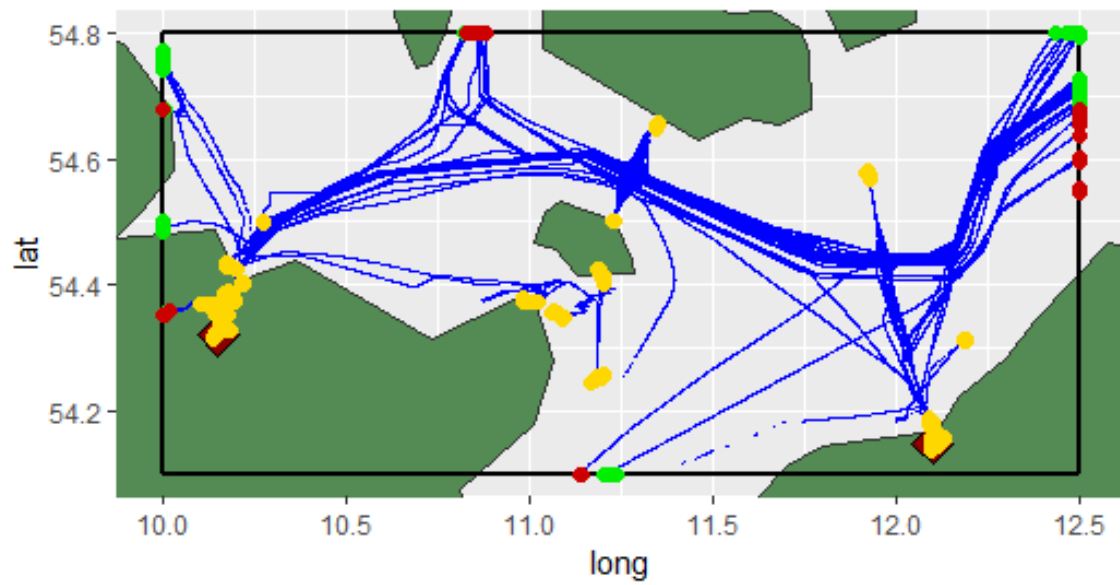
Slika 4.6: Otkrivene ulazne točke



Slika 4.7: Otkrivene stacionarne točke



Slika 4.8: Otkrivene izlazne točke



Slika 4.9: Otkrivene putne točke i rute

Za potrebe primjene, odabire se plovilo od interesa (VOI) za kojeg se provodi primjena znanja dobivenog prethodnim nenadgledanim izdvajanjem ruta. Odabire se plovilo s MMSI brojem 209214000. Radi se o tankeru imena VERDI.

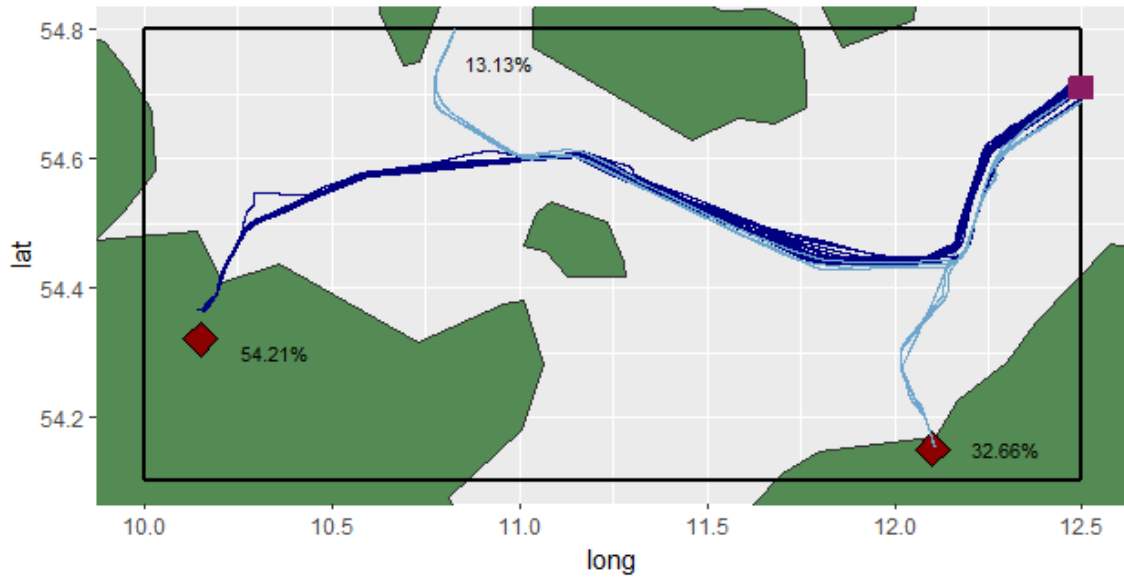
Uz pretpostavku da nema anomalija, daje se vjerojatnost dolaska u kompatibilnu putnu točku, koja pripada kompatibilnim rutama plovila VERDI s obzirom na odabrane položaje, korištenjem jednadžbe (3.7) i to s 2 različita početna položaja.

Slike 4.10 i 4.11 prikazuju vjerojatnost dolazaka plovila VERDI u kompatibilne točke uz odabrani početni položaj označen ljubičastim kvadratom. Najizglednija ruta naznačena je tamno plavom bojom, dok su ostale, manje izgledne rute, naznačene svijetlo plavom bojom (vjerojatnosti (u postocima) prikazane su uz samu destinaciju). Za radijus područja d , opisan u potpoglavlju 3.4, odabrani su:

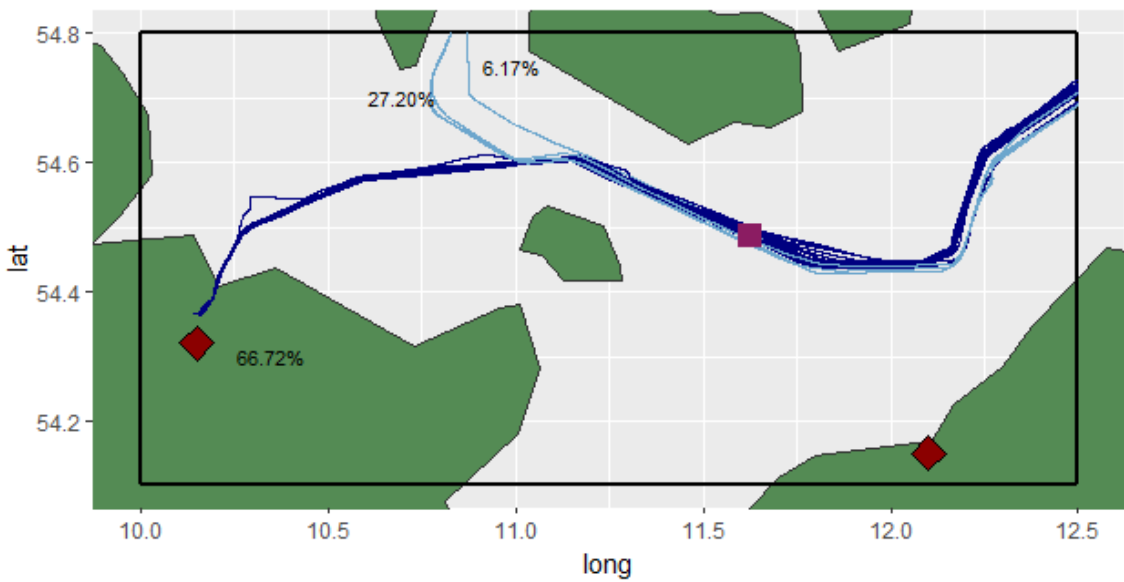
- slika 4.10: $d = 6173$ metara ≈ 3.33 nautičke milje,
- slika 4.11: $d = 4417$ metara ≈ 2.38 nautičke milje.

Eksperimentalno odabrani radijusi razlikuju se zbog činjenice da kad se plovilo nalazi u dijelu gdje je gušći promet, radijus d treba biti odabran manji kako bi se spriječilo hvatanje previše različitih ruta. Za razliku od toga, u manje prometnim dijelovima, parametar d se uzima veći.

Uočeno je da ruta prema putnoj točki za koju je dobivena najveća vjerojatnost jest upravo ona ruta kojom plovi odabrano plovilo.



Slika 4.10: Predviđanje destinacije (1)

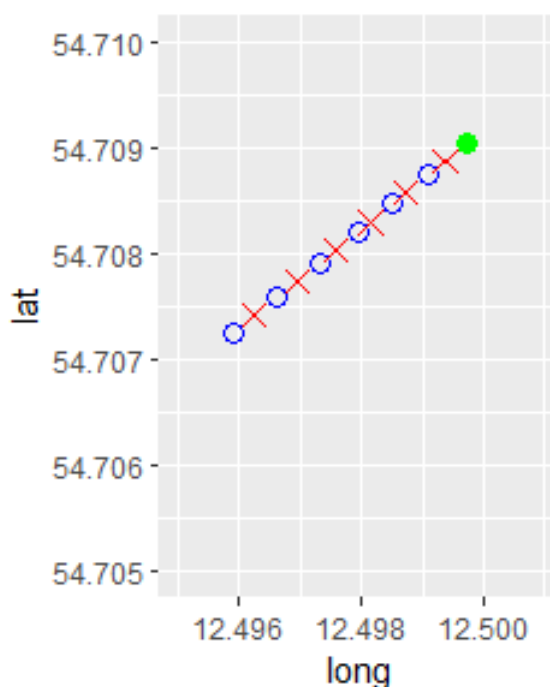


Slika 4.11: Predviđanje destinacije (2)

Nadalje, slika 4.12 prikazuje rezultate algoritma predviđanja traga (Algoritam 5). Kao primjer uzet je prvi položaj plovila VERDI kao na slici 4.10. Pokrenut je Algoritam 5 uz:

- Eps jednak veličini radijusa za taj položaj (≈ 3.33 nautičke milje),
- step = 6,
- rutu u luku Kiel (za koju je gornja analiza pokazala da je najizglednija).

Na slici zeleni krug označava početak predviđanja, plavi krugovi stvarne položaje, a crveni križevi predviđene položaje u skladu s Algoritmom 5.



Slika 4.12: Predviđanje položaja

Vrijednosti stvarnih i predviđenih položaja dani su u sljedećoj tablici:

Stvarni položaj	Predviđeni položaj
12.49910, 54.70874	12.49936, 54.70887
12.49852, 54.70846	12.49873, 54.70857
12.49795, 54.70820	12.49815, 54.70830
12.49732, 54.70790	12.49758, 54.70803
12.49662, 54.70758	12.49694, 54.70774
12.49592, 54.70725	12.49625, 54.70741

Za proučavanje anomalija izabrano je plovilo sa MMSI brojem 212491000 imena Dalsland, nepoznatog tipa. Postavljena je granica za aposteriornu vjerojatnost najizglednije rute na 0.03 (u postocima). Primjenom analize dobiveno je da je njegova najizglednija ruta ona u luku Kiel te aposteriorna vjerojatnost za tu rutu, u vremenskom rasponu od 11:13:18 do 11:13:49, iznosi 0.02407 posto, što je manje od dane granice. Proučavanjem plovila Dalsland u tom periodu uočeno je naglo stajanje u 3 navrata, iako po parametru SOG on i dalje plovi konstantnom brzinom. Promjene položaja u danom vremenskom intervalu vidljive su u sljedećoj tablici:

Vrijeme	Položaj
11:13:18	12.49997, 54.71540
11:13:23	12.49997, 54.71540
11:13:28	12.49939, 54.71516
11:13:31	12.49939, 54.71516
11:13:39	12.49875, 54.71489
11:13:48	12.49875, 54.71489
11:13:49	12.49816, 54.71465

Može se zaključiti da je plovilo u datom vremenskom intervalu naglo stajalo radi izbjegavanja sudara ili sličnih događaja.

Kod izvedbe nenadgledanog izdvajanja ruta i primjene može se naći u dodatku A i B, dok se kod za sve grafičke prikaze iz ovog poglavlja može naći u dodatku C.

Poglavlje 5

Zaključak

Potpuno iskorištavanje redovito generiranih masivnih skupova AIS podataka za potrebe prepoznavanja uzoraka pomorskog prometa i kretanja brodova traži primjenu postupaka strojnog učenja.

Metodologija TREAD automatski dobiva znanje o pomorskom prometu (luke, obalne platforme, ulazne i izlazne točke i rute) iz AIS podataka bez prethodnih informacija o domeni tog područja.

Proces učenja robustan je s obzirom na različit broj senzora, njihovu pokrivenost i stopu ažuriranja te razmjer promatranog područja. Otkrivene skupine karakteriziraju prostorne i vremenske informacije, koje omogućuju otkrivanje ponašanja s malim vjerojatnostima i predviđanje budućih položaja i mogućih odredišta. Postupak izdvajanja ruta temelji se na nenadgledanom inkrementalnom učenju i može se primijeniti u stvarnom vremenu ili na povijesnim podacima, kao što je slučaj u ovom radu.

U ovom radu plovila su analizirana kao kolektivni entitet koji konstruira i oblikuje prometne obrasce u promatranom području. Znanje iz velikog skupa AIS podataka, dobiveno nenadgledanim inkrementalnim učenjem, iskorištava se dalje za klasificiranje ruta, predviđanje odredišta plovila, predviđanje samih budućih položaja s obzirom na trenutni položaj te detekciju anomalija, temeljem aposteriorne vjerojatnosti, neovisno o broju uređaja iz kojih se dobivaju podaci, njihovim obilježjima i platformi.

Rezultirajuća detekcija anomalija često može biti u potpunosti objašnjena kroz interakciju između objekata, odnosno plovila (npr. iznenadna promjena smjera ili brzine može biti posljedica manevara izbjegavanja sudara u odnosu na drugo plovilo).

Takva razina interakcije može pomoći u poboljšanju tumačenja i razumijevanja ponašanja i namjera plovila.

Bibliografija

- [1] IALA AISM, *An overview of AIS 1082*, 2011, dostupno na: <https://www.iala-aism.org/product/an-overview-of-ais-1082/>, (5. veljače 2020.).
- [2] *All about AIS*, dostupno na: <http://www.allaboutais.com/index.php/en/>, (5. veljače 2020.).
- [3] E. Alpaydin, *Introduction to machine learning*, The MIT Press, 2009.
- [4] Danish Maritime Authority, *AIS data*, dostupno na: <https://www.dma.dk/SikkerhedTilSoes/Sejladsinformation/AIS/Sider/default.aspx>, (5. veljače 2020.).
- [5] H. H. Bock, *Proximity Measures*, Encyclopedia of Statistics in Behavioral Science, Wiley, New York (2005), 1621–1628.
- [6] J. Brownlee, *14 different types of learning in machine learning*, 2019, dostupno na: <https://machinelearningmastery.com/types-of-learning-in-machine-learning/>, (5. veljače 2020.).
- [7] A. Bulić, *Procjena parametra Weibullove distribucije*, Magistarska radnja, Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku, Financijska matematika i statistika, 2016.
- [8] *Characteristics of the Weibull Distribution*, dostupno na: <https://www.weibull.com/hotwire/issue14/relbasics14.htm>, (5. veljače 2020.).
- [9] B. Dalbelo Bašić i J. Šnajder, *Uvod u strojno učenje - Fer - UniZG*, 2015.
- [10] M. Erdeg, *Metode procjene parametara*, Magistarska radnja, Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku, 2016.
- [11] M. Ester, H.P. Kriegel, J. Sander, M. Wimmer i X. Xu, *Incremental clustering for mining in a data warehousing environment*, In Proceedings of the 24th International Conference on Very Large Data Bases, New York, NY, USA (1998), 323–333.

- [12] M. Ester, H.P. Kriegel, J. Sander i X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, In Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA (1996), 226–231.
- [13] B. Gajić, *Sustavi nadzora i upravljanja u pomorskoj navigaciji*, Magistarska radnja, Sveučilište u Rijeci, Pomorski fakultet, 2014, str. 46–74.
- [14] J. George, J. Crassidis, T. Singh i A.M. Fosbury, *Anomaly detection using context-aided target tracking*, Adv. Inf. Fusion (2011), 39–56.
- [15] M. Gregorić, *Strojno učenje kao alat za zaključivanje*, Magistarska radnja, Sveučilište u Zagrebu, Filozofski fakultet, Odsjek za informacijske i komunikacijske znanosti, 2019.
- [16] *Hot topic for project, thesis, and research – Machine Learning*, 2013, dostupno na: <https://www.techsparks.co.in/hot-topic-for-project-and-thesis-machine-learning/>, (5. veljače 2020.).
- [17] *Izmjerive funkcije*, dostupno na: https://web.math.pmf.unizg.hr/nastava/mi/files/Izmjerive_funkcije_07.pdf, (5. veljače 2020.).
- [18] V. Losing, B. Hammer i H. Wersing, *Incremental on-line learning: A review and comparison of state of the art algorithms*, Elsevier (2017).
- [19] M. McCloskey i N. J.Cohen, *Catastrophic interference in connectionist networks: The sequential learning problem*, Elsevier (1989), 109–165.
- [20] J. Milanović, *Procjena funkcije gustoće*, Magistarska radnja, Sveučilište J. J. Strossmayera u Osijeku, Odjel za matematiku, 2012.
- [21] T. Mitchell, *Machine learning*, McGraw-Hill Education, ožujak 1997.
- [22] *Neprekidne slučajne varijable*, 2019, dostupno na: https://web.math.pmf.unizg.hr/nastava/stat/files/chap2_novo.pdf, (5. veljače 2020.).
- [23] D. Nevell, *Anomaly detection in white shipping*, In Proceedings of 2nd IMA Conference on Mathematics in Defence, Farnborough, UK (2009), 1–7.
- [24] G. Pallotta, M. Vespe i K. Bryan, *Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction*, Entropy (2013), 2218–2245.

- [25] *Posterior probability and the posterior distribution - Statistics how to*, dostupno na: <https://www.statisticshowto.datasciencecentral.com/posterior-distribution-probability/>, (5. veljače 2020.).
- [26] *Procjena parametara*, 2019, dostupno na: https://web.math.pmf.unizg.hr/nastava/stat/files/chap4_novo.pdf, (5. veljače 2020.).
- [27] M. Riveiro, *Visual Analytics for Maritime Anomaly Detection*, Disertacija, Örebro University, Örebro, Sweden, 2011.
- [28] N. Sandrić i Z. Vondraček, *Vjerojatnost predavanja*, 2019, dostupno na: https://web.math.pmf.unizg.hr/nastava/vjer/files/vjer_predavanja.pdf, (5. veljače 2020.).
- [29] Srce, *Upoznavanje sa sintaksom jezika R i njegova primjena u osnovnoj statističkoj i grafičkoj analizi podataka*, 2015, dostupno na: https://www.srce.unizg.hr/files/srce/docs/edu/R/s720_polaznik.pdf, (5. veljače 2020.).
- [30] J. Swarndeeep Saket i Dr. Sharnil Pandya, *An overview of partitioning algorithms in clustering techniques*, International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) (2016), 1943–1946.
- [31] *Tehnike višestrukog prijenosa signala*, 2019, dostupno na: http://mirza-sa.tripod.com/telekom/tehnike_visestrukog_prijenosa.htm, (5. veljače 2020.).
- [32] S. Ullman, T. Poggio, D. Harari, D. Zysman i D. Seibert, *Unsupervised learning: Clustering*, dostupno na: <http://www.mit.edu/~9.54/fall14/slides/Class13.pdf>, (5. veljače 2020.).
- [33] *Understanding AIS (Automatic Identification System)*, 2019, dostupno na: <https://maritimeintelligence.informa.com/resources/product-content/understanding-the-automatic-identification-system>, (5. veljače 2020.).
- [34] M. Vespe, K. Bryan, P. Braca i I. Visentini, *Unsupervised learning of maritime traffic patterns for anomaly detection*, In Proceedings of 9th IET Data Fusion and Target Tracking Conference, London, UK (2012).
- [35] *VesselFinder: Free AIS Ship Tracking of Marine Traffic*, 2019, dostupno na: <https://www.vesselfinder.com/>, (5. veljače 2020.).
- [36] J. Will, C. Claxton i L. Peel, *Fast maritime anomaly detection using KD-tree Gaussian processes*, In Proceedings 2nd IMA Conference on Maths in Defence, Shrivenham, UK (2011).

Sažetak

Razumijevanje uzoraka ponašanja plovila u operativnom radu ključno je za razvijanje svijesti o situacijama u pomorskom prometu, posebice za klasificiranje i predviđanje aktivnosti. Razvojem automatskog identifikacijskog sustava (eng. Automatic Identification System, AIS), omogućeno je praćenje putanja plovila, kao ulaza za postupke identifikacije uobičajenih obrazaca ponašanja, definicije koridora, detekcije neuobičajenih manevara, optimizaciju upravljanja plovidbenim tokovima, strategijsko planiranje, te povećanje učinkovitosti navigacije i smanjenje učinaka na okoliš. Navedeni elementi svijesti o situaciji mogu se izvesti iz AIS poruka korištenjem statističke analize i strojnog učenja na opaženim podacima (AIS poruke).

Metoda ovog rada, izdvajanje prometnih ruta i detekcija anomalija, nazvana TREAD (eng. Traffic Route Extraction and Anomaly Detection), ima za cilj pretvoriti velike količine AIS podataka u znanje koje se može iskoristiti za klasificiranje i predviđanje ruta te detekciju anomalija, neovisno o broju uređaja iz kojih se dobivaju podaci, njihovim obilježjima i platformi. Postupak izdvajanja ruta temelji se na nenadgledanom inkrementalnom učenju, što omogućuje karakterizaciju pomorskog prometa bez *a priori* znanja, dok se predviđanje ruta i detekcija anomalija temelji na aposteriornoj vjerojatnosti.

Summary

Understanding vessel behavior patterns in operational work is crucial for developing awareness of maritime traffic situations, in particular for classifying and predicting activities. With the development of the Automatic Identification System (AIS), it is possible to monitor vessel trajectories as inputs for procedures for identifying common behavioral patterns, defining corridors, detecting unusual maneuvers, optimizing navigational flow management, strategic planning, and increasing navigation efficiency and reducing effects on the environment. The aforementioned situation awareness elements can be derived from AIS messages using statistical analysis and machine learning on observed data (AIS messages).

The method of this work, Traffic Route Extraction and Anomaly Detection (TREAD), aims to convert large amounts of AIS data into knowledge that can be used to classify and predict routes and detect anomalies, regardless of number the devices from which the data is obtained, their performance and the platform. The route extraction process is based on unsupervised incremental learning, which allows characterization of maritime traffic without *a priori* knowledge, while route prediction and anomaly detection are based on posterior probability.

Životopis

Mihaela Gamulin rođena je 29.09.1995. godine u Zagrebu. Pohađala je osnovnu školu "Marin Držić" i paralelno osnovnu glazbenu školu "Pavla Markovca". Nakon završetka osnovne škole, 2010. godine, upisuje opću 2. gimnaziju. Srednjoškolsko obrazovanje završava 2014. godine i iste godine upisuje preddiplomski sveučilišni studij Matematika na matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu. 2017. godine stiče titulu sveučilišne prvostupnice matematike, te iste godine upisuje diplomski studij Računarstvo i matematika na istom sveučilištu.

Dodatak A

Kod izvedbe nenadgledanog izdvajanja putnih točaka i ruta

```
1 library(readr)
2 library(tidyverse)
3 library(dplyr)
4 library(mapproj)
5 library(sp)
6 library(measurements)
7 library(sf)
8 library(dbscan)
9 library(geosphere)
10 library(lubridate)
11 library(rgdal)
12
13 ##### Ucitavanje podataka/ciscenje/graficki prikaz #####
14
15 # Ucitavanje podataka
16 rawData <- read_csv("C:/Users/mihaela/Downloads/diplomski/aisdk_20190929.csv")
17
18 # Brisanje podataka kojima je SOG ili COG parametar jednak NA
19 rawData <- rawData %>% drop_na(SOG)
20 rawData <- rawData %>% drop_na(COG)
21 colnames(rawData) <- c("Timestamp", "Type of mobile", "MMSI", "Latitude", "Longitude",
22                       "Navigational status", "ROT", "SOG", "COG", "Heading", "IMO",
23                       "Callsign", "Name", "Ship type", "Cargo type", "Width", "Length",
24                       "Type of position fixing device", "Draught", "Destination",
25                       "ETA", "Data source type", "A", "B", "C", "D")
26 keep <- c("Timestamp", "Type of mobile", "MMSI", "Latitude", "Longitude",
27          "Navigational status", "SOG", "COG", "Name", "Ship type")
28 rawData <- rawData[keep]
29
30 # Podaci u promatranom podrucju
31 selectedData <- filter(rawData, Longitude >= 10 & Longitude <= 12.5
32                       & Latitude >= 54.1 & Latitude <= 54.8)
33
34 # Promjena vremena iz DateTime u Time
35 Time <- format(as.POSIXct(strptime(selectedData$Timestamp, "%d/%m/%Y %H:%M:%S", tz="")),
36              format="%H:%M:%S")
37 selectedData$Timestamp <- Time
```



```

38
39 messages <- selectedData
40
41 # Promjena Longitude, Latitude u Universal Transverse Mercator (UTM)
42 cord.dec <- SpatialPoints(cbind(selectedData$Longitude, -selectedData$Latitude),
43                             proj4string=CRS("+proj=longlat"))
44 cord.UTM <- spTransform(cord.dec, CRS("+init=epsg:32632"))
45 utm <- selectedData
46 utm$Longitude <- cord.UTM$coords.x1
47 utm$Latitude <- cord.UTM$coords.x2
48 utm <- utm %>% rename(Longitude_UTM=Longitude, Latitude_UTM=Latitude)
49 messages <- cbind(messages, utm[,c(4,5)])
50
51 # Brisanje duplih javljanja (kad je plovilo na dva mjesta u isto vrijeme -> nemoguće)
52 messages <- messages[!duplicated(messages[,c("MMSI","Timestamp")], fromLast=T),]
53
54 ##### Postavljanje parametara #####
55
56 ENs <- data.frame(Longitude=11.92874, Latitude=54.57182, MMSI=219024178)
57 POs <- data.frame(Longitude=11.92874, Latitude=54.57182, MMSI=219024178)
58 EXs <- data.frame(Longitude=10.98745, Latitude=54.68632, MMSI=258829000)
59
60 tau <- "00:06:00"
61 str1 <- "Vs_"
62 Delta_days <- "12:00:00"
63 Vs <- data.frame(MMSI=double())
64 track <- setNames(data.frame(matrix(ncol=7, nrow=0)), c("Longitude_UTM", "Latitude_UTM",
65                                                         "COG", "SOG", "timestamp", "Longitude", "Latitude"))
66 min_speed <- 4
67 Rs <- data.frame(name=character(), wp_a=character(), wp_b=character(), MMSI=double(),
68                 track=track, stringsAsFactors=FALSE)
69
70 N_ENs <- 4
71 Eps_ENs <- 0.035
72 N_POs <- 25
73 Eps_POs <- 0.02
74 N_EXs <- 3
75 Eps_EXs <- 0.02
76 clustered_ENs <- dbscan(ENs[,c(1,2)], Eps_ENs, N_ENs)
77 ENs <- ENs %>% rename(track.Longitude=Longitude, track.Latitude=Latitude)
78 ENs$cluster <- clustered_ENs$cluster
79 clustered_POs <- dbscan(POs[,c(1,2)], Eps_POs, N_POs)
80 POs <- POs %>% rename(track.Longitude=Longitude, track.Latitude=Latitude)
81 POs$cluster <- clustered_POs$cluster
82 clustered_EXs <- dbscan(EXs[,c(1,2)], Eps_EXs, N_EXs)
83 EXs <- EXs %>% rename(track.Longitude=Longitude, track.Latitude=Latitude)
84 EXs$cluster <- clustered_EXs$cluster
85 EN <- data.frame(ENs=character(), stringsAsFactors=FALSE)
86 PO <- data.frame(POs=character(), stringsAsFactors=FALSE)
87 EX <- data.frame(EXs=character(), stringsAsFactors=FALSE)
88
89 ##### Nenadgledano izdvajanje ruta #####
90
91 Unsupervised_Route_Extraction <- function(N_ENs, N_POs, N_EXs, Eps_ENs, Eps_POs, Eps_EXs)
92 {
93   for(row in 1:nrow(messages))
94   {
95     message <- messages[row,]
96
97     if(!(message$MMSI %in% Vs$MMSI))

```

```

98 {
99   rowCounter <- nrow(Vs)+1
100   Vs[rowCounter,1] <- message$MMSI
101   Vs_MMSI <- data.frame(MMSI=double(), status=character(), track=track,
102                        avg_speed=double(), wps=character(), timestamp_wps=character(),
103                        routes=character(), stringsAsFactors=FALSE)
104   rowCounter1 <- nrow(Vs_MMSI)+1
105   Vs_MMSI[rowCounter1,1] <- message$MMSI
106   Vs_MMSI[rowCounter1,2] <- "sailing"
107   Vs_MMSI[rowCounter1,3] <- message$Longitude_UTM
108   Vs_MMSI[rowCounter1,4] <- message$Latitude_UTM
109   Vs_MMSI[rowCounter1,5] <- message$COG
110   Vs_MMSI[rowCounter1,6] <- message$SOG
111   Vs_MMSI[rowCounter1,7] <- message$Timestamp
112   Vs_MMSI[rowCounter1,8] <- message$Longitude
113   Vs_MMSI[rowCounter1,9] <- message$Latitude
114   assign(paste(str1,toString(message$MMSI)), Vs_MMSI,envir=.GlobalEnv)
115
116   # Online_WPs_clustering + Route_Objects_Manager
117   if(Vs_MMSI[rowCounter1,8]<=10.02 | Vs_MMSI[rowCounter1,8]>=12.48 |
118      Vs_MMSI[rowCounter1,9]>=54.78 | Vs_MMSI[rowCounter1,9]<=54.11)
119   {
120     list <- Online_WPs_Clustering(EN, Vs_MMSI, Eps_ENs, N_ENs, "EN", Rs)
121     EN <- list[[1]]
122     Rs <- list[[3]]
123     Vs_MMSI <- list[[2]]
124     assign(paste(str1,toString(message$MMSI)),Vs_MMSI,envir=.GlobalEnv)
125     list1 <- Route_Objects_Manager(Rs, Vs_MMSI)
126     Rs <- list1[[1]]
127     Vs_MMSI <- list1[[2]]
128     assign(paste(str1,toString(message$MMSI)),Vs_MMSI,envir=.GlobalEnv)
129   }
130 }
131 else
132 {
133   Vs_MMSI <- data.frame(MMSI = double(), status=character(), track=track,
134                        avg_speed=double(), wps=character(), timestamp_wps=character(),
135                        routes=character(), stringsAsFactors=FALSE)
136   end <- nrow(Vs_MMSI)
137   Vs_MMSI[end+1,1] <- message$MMSI
138   Vs_MMSI[end+1,3] <- message$Longitude_UTM
139   Vs_MMSI[end+1,4] <- message$Latitude_UTM
140   Vs_MMSI[end+1,5] <- message$COG
141   Vs_MMSI[end+1,6] <- message$SOG
142   Vs_MMSI[end+1,7] <- message$Timestamp
143   Vs_MMSI[end+1,8] <- message$Longitude
144   Vs_MMSI[end+1,9] <- message$Latitude
145
146   temp <- data.frame(MMSI = double(), status=character(), track=track,
147                    avg_speed=double(), wps=character(), timestamp_wps=character(),
148                    routes=character(), stringsAsFactors=FALSE)
149   temp <- bind_rows(eval(as.name(paste(str1,toString(message$MMSI))))), Vs_MMSI)
150   rowCounter2 <- nrow(temp)
151
152   Delta_long <- temp[rowCounter2,3]-temp[rowCounter2-1,3]
153   Delta_lat <- temp[rowCounter2,4]-temp[rowCounter2-1,4]
154   Delta_pos <- sqrt((Delta_long)^2+(Delta_lat)^2)
155   Delta_t <- period_to_seconds(hms(temp[rowCounter2,7]))
156   -period_to_seconds(hms(temp[rowCounter2-1,7]))
157

```

```

158 temp[rowCounter2,10] <- Delta_pos/Delta_t
159 if(is.nan(temp[rowCounter2,10])){ temp[rowCounter2,10] <- 0 }
160
161 if(temp[rowCounter2,10]<min_speed & temp[rowCounter2-1,2]=="sailing"
162 & temp[rowCounter2,6]<1.6)
163 {
164   temp[rowCounter2,2] <- "stationary"
165   assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
166
167   # Online_WPs_clustering + Route_Objects_Manager
168   list <- Online_WPs_Clustering(PO, temp, Eps_POs, N_POs, "PO", Rs)
169   PO <- list[[1]]
170   Rs <- list[[3]]
171   temp <- list[[2]]
172   assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
173   list1 <- Route_Objects_Manager(Rs, temp)
174   Rs <- list1[[1]]
175   temp <- list1[[2]]
176   assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
177 }
178 else if(temp[rowCounter2-1,2]=="lost")
179 {
180   temp[rowCounter2,2] <- "sailing"
181   assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
182
183   # Online_WPs_clustering + Route_Objects_Manager
184   if(temp[rowCounter2,8]<=10.02 | temp[rowCounter2,8]>=12.48 |
185       temp[rowCounter2,9]>=54.78 | temp[rowCounter2,9]<=54.11)
186   {
187     list <- Online_WPs_Clustering(EN, temp, Eps_ENs, N_ENs, "EN", Rs)
188     EN <- list[[1]]
189     Rs <- list[[3]]
190     temp <- list[[2]]
191     assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
192     list1 <- Route_Objects_Manager(Rs, temp)
193     Rs <- list1[[1]]
194     temp <- list1[[2]]
195     assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
196   }
197 }
198 else
199 {
200   if(temp[rowCounter2,10]<min_speed & temp[rowCounter2,6]<1.6)
201   {
202     temp[rowCounter2,2] <- "stationary"
203   }
204   else
205   {
206     temp[rowCounter2,2] <- "sailing"
207   }
208   assign(paste(str1,toString(message$MMSI)), temp,envir=.GlobalEnv)
209 }
210 }
211
212 if(period_to_seconds(hms(Delta_days))-period_to_seconds(hms(message$Timestamp))==0)
213 {
214   for(i in 1:nrow(Vs))
215   {
216     v <- Vs[i,]
217     vv <- eval(as.name(paste(str1,toString(v))))

```

```

218     rowCounter3 <- nrow(vv)
219
220     last_update <- period_to_seconds(hms(vv$track.timestamp[nrow(vv)]))
221     time_interval <- period_to_seconds(hms(Delta_days)-hms(tau))
222
223     if(time_interval>last_update & vv[rowCounter3,2]!="lost")
224     {
225         vv[rowCounter3,2] <- "lost"
226         assign(paste(str1,toString(v)), vv,envir=.GlobalEnv)
227
228         # Online_WPs_clustering + Route_Objects_Manager
229         if(vv[rowCounter3,8]<=10.02 | vv[rowCounter3,8]>=12.48 |
230            vv[rowCounter3,9]>=54.78 | vv[rowCounter3,9]<=54.11)
231         {
232             list <- Online_WPs_Clustering(EX, vv, Eps_EXs, N_EXs, "EX", Rs)
233             EX <- list[[1]]
234             Rs <- list[[3]]
235             vv <- list[[2]]
236             assign(paste(str1,toString(v)), vv,envir=.GlobalEnv)
237             list1 <- Route_Objects_Manager(Rs, vv)
238             Rs <- list1[[1]]
239             vv <- list1[[2]]
240             assign(paste(str1,toString(v)), vv,envir=.GlobalEnv)
241         }
242     }
243 }
244 }
245 print(row)
246 }
247 Rs <- Check(ENs, Rs, "EN")
248 Rs <- Check(POs, Rs, "PO")
249 Rs <- Check(EXs, Rs, "EX")
250 VS <- data.frame()
251 for(j in 1:nrow(Vs))
252 {
253     VS <- bind_rows(VS, eval(as.name(paste(str1,toString(Vs[j,1])))))
254 }
255 lst <- list(VS, Rs, Vs)
256 return(lst)
257 }
258
259 Online_WPs_Clustering <- function(WPs, Vs_MMSI, Eps, N, con, Rs)
260 {
261     e <- nrow(Vs_MMSI)
262     v <- Vs_MMSI[e,c(8,9)]
263     MMSI <- Vs_MMSI[1,1]
264     v <- cbind(v, MMSI)
265
266     l <- Incremental_DBSCAN(WPs, v, N, Eps, con)
267     op <- l[[2]]
268     WPs <- l[[1]]
269
270     if(op=="none")
271     {
272         Vs_MMSI[e,11] <- "Unclassified Waypoint"
273         Vs_MMSI[e,12] <- Vs_MMSI[e,7]
274     }
275     else
276     {
277         if(op=="new cluster")

```

```

278 {
279   if(con=="EN")
280   {
281     WPs[nrow(WPs)+1,1] <- paste(as.name(con), toString(nrow(WPs)+1))
282     Vs_MMSI[e,11] <- paste(as.name(con), toString(ENs[nrow(ENs),4]))
283     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
284     l <- Update(ENs, Vs_MMSI, con, Rs, MMSI)
285     Vs_MMSI <- l[[1]]
286     Rs <- l[[2]]
287   }
288   if(con=="PO")
289   {
290     WPs[nrow(WPs)+1,1] <- paste(as.name(con), toString(nrow(WPs)+1))
291     Vs_MMSI[e,11] <- paste(as.name(con), toString(POs[nrow(POs),4]))
292     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
293     l <- Update(POs, Vs_MMSI, con, Rs, MMSI)
294     Vs_MMSI <- l[[1]]
295     Rs <- l[[2]]
296   }
297   if(con=="EX")
298   {
299     WPs[nrow(WPs)+1,1] <- paste(as.name(con), toString(nrow(WPs)+1))
300     Vs_MMSI[e,11] <- paste(as.name(con), toString(EXs[nrow(EXs),4]))
301     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
302     l <- Update(EXs, Vs_MMSI, con, Rs, MMSI)
303     Vs_MMSI <- l[[1]]
304     Rs <- l[[2]]
305   }
306 }
307 if(op=="cluster expanded")
308 {
309   if(con=="EN")
310   {
311     Vs_MMSI[e,11] <- paste(as.name(con), toString(ENs[nrow(ENs),4]))
312     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
313     l <- Update(ENs, Vs_MMSI, con, Rs, MMSI)
314     Vs_MMSI <- l[[1]]
315     Rs <- l[[2]]
316   }
317   if(con=="PO")
318   {
319     Vs_MMSI[e,11] <- paste(as.name(con), toString(POs[nrow(POs),4]))
320     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
321     l <- Update(POs, Vs_MMSI, con, Rs, MMSI)
322     Vs_MMSI <- l[[1]]
323     Rs <- l[[2]]
324   }
325   if(con=="EX")
326   {
327     Vs_MMSI[e,11] <- paste(as.name(con), toString(EXs[nrow(EXs),4]))
328     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
329     l <- Update(EXs, Vs_MMSI, con, Rs, MMSI)
330     Vs_MMSI <- l[[1]]
331     Rs <- l[[2]]
332   }
333 }
334 if(op=="clusters merged")
335 {
336   if(con=="EN")
337   {

```

```

338     Vs_MMSI[e,11] <- paste(as.name(con), toString(ENs[nrow(ENs),4]))
339     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
340     l <- Update(ENs, Vs_MMSI, con, Rs, MMSI)
341     Vs_MMSI <- l[[1]]
342     Rs <- l[[2]]
343     if(nrow(WPs)>0){ WPs <- head(WPs,-1) }
344   }
345   if(con=="P0")
346   {
347     Vs_MMSI[e,11] <- paste(as.name(con), toString(POs[nrow(POs),4]))
348     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
349     l <- Update(POs, Vs_MMSI, con, Rs, MMSI)
350     Vs_MMSI <- l[[1]]
351     Rs <- l[[2]]
352     if(nrow(WPs)>0){ WPs <- head(WPs,-1) }
353   }
354   if(con=="EX")
355   {
356     Vs_MMSI[e,11] <- paste(as.name(con), toString(EXs[nrow(EXs),4]))
357     Vs_MMSI[e,12] <- Vs_MMSI[e,7]
358     l <- Update(EXs, Vs_MMSI, con, Rs, MMSI)
359     Vs_MMSI <- l[[1]]
360     Rs <- l[[2]]
361     if(nrow(WPs)>0){ WPs <- head(WPs,-1) }
362   }
363 }
364 }
365 lst <- list(WPs, Vs_MMSI, Rs)
366 return(lst)
367 }
368
369 Update <- function(sampl, Vs_MMSI, con, Rs, MMSI)
370 {
371   filterCluster <- filter(sampl, sampl$cluster==sampl[nrow(sampl),4])
372   for(i in 1:nrow(filterCluster))
373   {
374     findMMSI <- filterCluster[i,3]
375     if(MMSI==findMMSI)
376     {
377       Id <- which(Vs_MMSI$track.Longitude==filterCluster[i,1] &
378                 Vs_MMSI$track.Latitude==filterCluster[i,2] &
379                 !is.na(Vs_MMSI$wps))
380       Vs_MMSI[Id,11] <- paste(as.name(con), toString(sampl[nrow(sampl),4]))
381     }
382     else
383     {
384       findData <- eval(as.name(paste(toString("Vs_"), toString(findMMSI))))
385       Id <- which(findData$track.Longitude==filterCluster[i,1] &
386                 findData$track.Latitude==filterCluster[i,2] &
387                 !is.na(findData$wps))
388       findData[Id,11] <- paste(as.name(con), toString(sampl[nrow(sampl),4]))
389       assign(paste(toString("Vs_"), toString(findMMSI)), findData, envir=.GlobalEnv)
390     }
391   }
392   filterCluster1 <- distinct(filterCluster, MMSI, .keep_all=TRUE)
393   for(j in 1:nrow(filterCluster1))
394   {
395     findMMSI <- filterCluster1[j,3]
396     if(findMMSI==MMSI)
397     {

```

```

398     l <- Route_Objects_Manager(Rs, Vs_MMSI)
399     Rs <- l[[1]]
400     Vs_MMSI <- l[[2]]
401   }
402   else
403   {
404     findData <- eval(as.name(paste(toString("Vs_"), toString(findMMSI))))
405     l <- Route_Objects_Manager(Rs, findData)
406     Rs <- l[[1]]
407     findData <- l[[2]]
408     assign(paste(toString("Vs_"), toString(findMMSI)), findData, envir=.GlobalEnv)
409   }
410 }
411 lst <- list(Vs_MMSI, Rs)
412 return(lst)
413 }
414
415 Incremental_DBSCAN <- function(WPs, v, N, Eps, con)
416 {
417   if(con=="EN")
418   {
419     v <- cbind(v, cluster=NA)
420     a <- bind_rows(ENS, v)
421     res1 <- dbSCAN(a[,c(1,2)], Eps, N)
422     a$cluster <- res1$cluster
423     v$cluster <- a[nrow(a),4]
424
425     # Broj klastera se povecao -> stvorio se novi klaster
426     if(max(res1$cluster)>max(clustered_ENs$cluster))
427     {
428       op <- "new cluster"
429       ll <- list(WPs=WPs, op=op)
430       assign("ENS", a, envir=.GlobalEnv)
431       assign("clustered_ENs", res1, envir=.GlobalEnv)
432       return(ll)
433     }
434     # Broj klastera se smanjio -> clusters merged
435     if(max(res1$cluster)<max(clustered_ENs$cluster))
436     {
437       op <- "clusters merged"
438       ll <- list(WPs=WPs, op=op)
439       assign("ENS", a, envir=.GlobalEnv)
440       assign("clustered_ENs", res1, envir=.GlobalEnv)
441       return(ll)
442     }
443   }
444   # Broj klastera je isti
445   if(max(res1$cluster)==max(clustered_ENs$cluster) & v$cluster!=0)
446   {
447     op = "cluster expanded"
448     ll <- list(WPs=WPs, op=op)
449     assign("ENS", a, envir=.GlobalEnv)
450     assign("clustered_ENs", res1, envir=.GlobalEnv)
451     return(ll)
452   }
453   # Nova tocka ne pripada ni jednom klasteru
454   if(v$cluster==0)
455   {
456     op <- "none"
457     ll <- list(WPs=WPs, op=op)

```

```

458     assign("ENs", a, envir=.GlobalEnv)
459     assign("clustered_ENs", res1, envir=.GlobalEnv)
460     return(ll)
461   }
462 }
463 if(con=="PO")
464 {
465   v <- cbind(v, cluster=NA)
466   a <- bind_rows(POs,v)
467   res1 <- dbscan(a[,c(1,2)], Eps, N)
468   a$cluster <- res1$cluster
469   v$cluster <- a[nrow(a),4]
470
471   if(max(res1$cluster)>max(clustered_POs$cluster))
472   {
473     op <- "new cluster"
474     ll <- list(WPs=WPs, op=op)
475     assign("POs", a, envir=.GlobalEnv)
476     assign("clustered_POs", res1, envir=.GlobalEnv)
477     return(ll)
478   }
479   if(max(res1$cluster)<max(clustered_POs$cluster))
480   {
481     op <- "clusters merged"
482     ll <- list(WPs=WPs, op=op)
483     assign("POs", a, envir=.GlobalEnv)
484     assign("clustered_POs", res1, envir=.GlobalEnv)
485     return(ll)
486   }
487   if(max(res1$cluster)==max(clustered_POs$cluster) & v$cluster!=0)
488   {
489     op = "cluster expanded"
490     ll <- list(WPs=WPs, op=op)
491     assign("POs", a, envir=.GlobalEnv)
492     assign("clustered_POs", res1, envir=.GlobalEnv)
493     return(ll)
494   }
495   if(v$cluster==0)
496   {
497     op <- "none"
498     ll <- list(WPs=WPs, op=op)
499     assign("POs", a, envir=.GlobalEnv)
500     assign("clustered_POs", res1, envir=.GlobalEnv)
501     return(ll)
502   }
503 }
504 if(con=="EX")
505 {
506   v <- cbind(v, cluster=NA)
507   a <- bind_rows(EXs,v)
508   res1 <- dbscan(a[,c(1,2)], Eps, N)
509   a$cluster <- res1$cluster
510   v$cluster <- a[nrow(a),4]
511
512   if(max(res1$cluster)>max(clustered_EXs$cluster))
513   {
514     op <- "new cluster"
515     ll <- list(WPs=WPs, op=op)
516     assign("EXs", a, envir=.GlobalEnv)
517     assign("clustered_EXs", res1, envir=.GlobalEnv)

```



```

518     return(ll)
519   }
520   if(max(res1$cluster)<max(clustered_EXs$cluster))
521   {
522     op <- "clusters merged"
523     ll <- list(WPs=WPs, op=op)
524     assign("EXs", a, envir=.GlobalEnv)
525     assign("clustered_EXs", res1, envir=.GlobalEnv)
526     return(ll)
527   }
528   if(max(res1$cluster)==max(clustered_EXs$cluster) & v$cluster!=0)
529   {
530     op <- "cluster expanded"
531     ll <- list(WPs=WPs, op=op)
532     assign("EXs", a, envir=.GlobalEnv)
533     assign("clustered_EXs", res1, envir=.GlobalEnv)
534     return(ll)
535   }
536   if(v$cluster==0)
537   {
538     op <- "none"
539     ll <- list(WPs=WPs, op=op)
540     assign("EXs", a, envir=.GlobalEnv)
541     assign("clustered_EXs", res1, envir=.GlobalEnv)
542     return(ll)
543   }
544 }
545 }
546
547 Route_Objects_Manager <- function(Rs, Vs_MMSI)
548 {
549   waypoints <- filter(Vs_MMSI,!is.na(Vs_MMSI$wps) & Vs_MMSI$wps!="Unclassified Waypoint")
550   nmrOfWaypoints <- nrow(waypoints)
551   len <- nmrOfWaypoints - 1
552   a <- 1
553   b <- 0
554
555   if(nmrOfWaypoints>1)
556   {
557     for(i in 1:len)
558     {
559       wp_a <- waypoints[nrow(waypoints)-a,]
560       wp_b <- waypoints[nrow(waypoints)-b,]
561
562       if((substr(wp_a$wps,1,2)=="P0" & substr(wp_b$wps,1,2)=="P0"
563           & substr(wp_a$wps,4,4)!=substr(wp_b$wps,4,4))
564          | (substr(wp_a$wps,1,2)=="EN" & substr(wp_b$wps,1,2)=="P0")
565          | (substr(wp_a$wps,1,2)=="P0" & substr(wp_b$wps,1,2)=="EX")
566          | (substr(wp_a$wps,1,2)=="EN" & substr(wp_b$wps,1,2)=="EX"))
567       {
568         name <- paste(toString(wp_a$wps), toString(wp_b$wps))
569         timestamp_wp_a <- wp_a$track.timestamp
570         timestamp_wp_b <- wp_b$track.timestamp
571         new <- filter(Vs_MMSI[,-c(2,10,11,12,13)], Vs_MMSI$track.timestamp>=timestamp_wp_a
572                     & Vs_MMSI$track.timestamp<=timestamp_wp_b)
573         add <- setNames(data.frame(matrix(ncol=3, nrow=nrow(new))), c("name", "wp_a", "wp_b"))
574         add[,1] <- name
575         add[,2] <- wp_a$wps
576         add[,3] <- wp_b$wps
577         new <- cbind(add, new)

```

```

578     Rs <- bind_rows(Rs, new)
579     id <- which(Vs_MMSI$timestamp_wps==wp_b$timestamp_wps)
580     Vs_MMSI[id,13] <- name
581   }
582   a <- a + 1
583   b <- b + 1
584 }
585 }
586 Rs <- Rs[!duplicated(Rs[,c("MMSI", "track.timestamp")], fromLast=T),]
587 lst <- list(Rs, Vs_MMSI)
588 return(lst)
589 }
590
591 # Run checking
592 Check <- function(sampl, Rs, con)
593 {
594   for(j in 1:max(sampl$cluster))
595   {
596     filt <- filter(sampl, sampl$cluster==j)
597     for(i in 1:nrow(filt))
598     {
599       findMMSI <- filt[i,3]
600       findData <- eval(as.name(paste(toString("Vs_"), toString(findMMSI))))
601       Id <- which(findData$track.Longitude==filt[i,1] &
602                 findData$track.Latitude==filt[i,2] &
603                 !is.na(findData$wps))
604       findData[Id,11] <- paste(as.name(con), toString(j))
605       assign(paste(toString("Vs_"), toString(findMMSI)), findData, envir=.GlobalEnv)
606     }
607     filt1 <- distinct(filt, MMSI, .keep_all=TRUE)
608     for(k in 1:nrow(filt1))
609     {
610       findMMSI <- filt1[k,3]
611       findData <- eval(as.name(paste(toString("Vs_"), toString(findMMSI))))
612       l <- Route_Objects_Manager(Rs, findData)
613       Rs <- l[[1]]
614       findData <- l[[2]]
615       assign(paste(toString("Vs_"), toString(findMMSI)), findData, envir=.GlobalEnv)
616     }
617   }
618   return(Rs)
619 }
620
621 # Pozivanje nenadgledanog izdvajanja ruta
622 lista <- Unsupervised_Route_Extraction(N_ENs, N_POs, N_EXs, Eps_ENs, Eps_POs, Eps_EXs)
623 VS <- lista[[1]]
624 Rs <- lista[[2]]
625 Vs <- lista[[3]]

```

Dodatak B

Kod izvedbe primjene

```
1 library(ks)
2 library(EnvStats)
3
4 ##### Track Predictor #####
5
6 Track_Predictor <- function(route, vessel, timestamp1, timestamp2, step, d)
7 {
8   time1 <- period_to_seconds(hms(timestamp1))
9   time2 <- period_to_seconds(hms(timestamp2))
10  delta_t <- time2-time1
11  delta_tau <- delta_t/ceiling(delta_t/step)
12  predicted <- setNames(data.frame(matrix(nrow=0,ncol=3)),c("xp","yp"))
13
14  for(j in seq(time1,time2,delta_tau))
15  {
16    VOII <- filter(vessel,period_to_seconds(hms(vessel[,7]))
17                  %in% (ceiling(j-6):ceiling(j+6)))
18    x_u <- VOII[1,3]
19    y_u <- VOII[1,4]
20    tem <- filter(Rs, ((Rs$track.Longitude_UTM-x_u)^2+(Rs$track.Latitude_UTM-y_u)^2)<d^2)
21    ss <- inner_join(route,tem,by = c("name","wp_a","wp_b","MMSI","track.Longitude_UTM",
22                                   "track.Latitude_UTM","track.COG","track.SOG",
23                                   "track.timestamp","track.Longitude","track.Latitude"
24                                   ))
25    speed <- conv_unit(ss$track.SOG, "knot", "m_per_sec")
26    ss$SOG_m <- speed
27    xx <- sqrt((ss$SOG_m)^2/(1+(tan(ss$track.COG))^2))
28    ss$xx <- xx
29    yy <- sqrt((ss$SOG_m)^2-(ss$xx)^2)
30    ss$yy <- yy
31    ss <- ss[,13:14]
32    sx <- median(ss$xx)
33    sy <- median(ss$yy)
34    xp <- VOII[1,3]-sx*delta_tau
35    yp <- VOII[1,4]+sy*delta_tau
36
37    addPredicted <- data.frame(xp = xp, yp = yp)
38    predicted <- rbind(predicted, addPredicted)
39  }
40  predicted <- predicted[-nrow(predicted),]
```

```

40 | sputm <- SpatialPoints(cbind(predicted$xp, -predicted$yp),
41 |                       proj4string=CRS("+proj=utm +zone=32U +datum=WGS84"))
42 | spgeo <- spTransform(sputm, CRS("+proj=longlat +datum=WGS84"))
43 | predicted$xp <- spgeo$coords.x1
44 | predicted$yp <- spgeo$coords.x2
45 | return(predicted)
46 | }
47 |
48 |
49 | ##### Primjena #####
50 |
51 | # 1) Predviđanje destinacije za odabrano vozilo u 2 različita trenutka
52 | VOI <- 'Vs_ 209214000'
53 |
54 | # prvo promatranje -> d = 3.33 nautičke milje
55 | VOI1 <- VOI[1:7,]
56 | speed <- conv_unit(VOI1$track.SOG, "knot", "m_per_sec")
57 | VOI1$SOG_m <- speed
58 | V <- setNames(data.frame(matrix(ncol=6, nrow=0)), c("x", "y", "xx", "yy", "x_u", "y_u"))
59 | S <- NULL
60 | d <- 6173
61 |
62 | for(i in 1:nrow(VOI1))
63 | {
64 |   xx <- sqrt((VOI1[i,14])^2/(1+(tan(VOI1[i,5]))^2))
65 |   yy <- sqrt((VOI1[i,14])^2-xx^2)
66 |   x <- VOI1[i,8]
67 |   y <- VOI1[i,9]
68 |   x_u <- VOI1[i,3]
69 |   y_u <- VOI1[i,4]
70 |   V[i,c(1:6)] <- c(x, y, xx, yy, x_u, y_u)
71 |   tem <- filter(Rs, ((Rs$track.Longitude_UTM-x_u)^2+(Rs$track.Latitude_UTM-y_u)^2)<d^2)
72 |   tem <- cbind(tem, i)
73 |   S <- rbind(S, tem)
74 | }
75 | k <- nrow(S %>% distinct(wp_b))
76 |
77 | P1 <- setNames(data.frame(matrix(1, nrow=k, ncol=2)), c("prob1", "destination"))
78 | for(i in 1:k)
79 | {
80 |   routesNames <- S %>% distinct(wp_b)
81 |   route <- filter(Rs, wp_b==routesNames[i,1])
82 |   for(t in 1:nrow(VOI1))
83 |   {
84 |     st <- filter(S, S[,12]==t)
85 |     stt <- st[1:11]
86 |
87 |     # KDE za položaj
88 |     forKDE1 <- inner_join(route, stt, by=c("name", "wp_a", "wp_b", "MMSI",
89 |                                           "track.Longitude_UTM", "track.Latitude_UTM", "track.COG",
90 |                                           "track.SOG", "track.timestamp", "track.Longitude", "track.Latitude"))
91 |     forKDE1 <- forKDE1[,10:11]
92 |     px <- density(forKDE1$track.Longitude, bw="nrd0")
93 |     py <- density(forKDE1$track.Latitude, bw="nrd0")
94 |
95 |     if(!is.na(approx(px$x, px$y, xout=V[t,1])$y))
96 |     {
97 |       p_x <- integrate(approxfun(px), lower=min(forKDE1$track.Longitude),
98 |                       upper=max(forKDE1$track.Longitude))
99 |     }

```

```

100   if(is.na(approx(px$x, px$y, xout=V[t,1])$y))
101     p_x[1] <- 0
102   if(!is.na(approx(py$x, py$y, xout=V[t,2])$y))
103     {
104       p_y <- integrate(approxfun(py), lower=min(forKDE1$track.Latitude),
105                       upper=max(forKDE1$track.Latitude))
106     }
107   if(is.na(approx(py$x, py$y, xout=V[t,2])$y))
108     p_y[1] <- 0
109   p1 <- as.numeric(p_x[1])*as.numeric(p_y[1])
110
111   # KDE za brzinu i kurs
112   forKDE2 <- inner_join(route, stt, by = c("name", "wp_a", "wp_b", "MMSI",
113     "track.Longitude_UTM", "track.Latitude_UTM", "track.COG", "track.SOG",
114     "track.timestamp", "track.Longitude", "track.Latitude"))
115   speed <- conv_unit(forKDE2$track.SOG, "knot", "m_per_sec")
116   forKDE2$SOG_m <- speed
117   xx <- sqrt((forKDE2$SOG_m)^2/(1+(tan(forKDE2$track.COG))^2))
118   forKDE2$xx <- xx
119   yy <- sqrt((forKDE2$SOG_m)^2-(forKDE2$xx)^2)
120   forKDE2$yy <- yy
121   forKDE2 <- forKDE2[,13:14]
122   pxx <- density(forKDE2$xx, bw="nrd0")
123   pyy <- density(forKDE2$yy, bw="nrd0")
124
125   if(!is.na(approx(pxx$x, pxx$y, xout=V[t,3])$y))
126     p_xx <- integrate(approxfun(pxx), lower=min(forKDE2$xx), upper=max(forKDE2$xx))
127   if(is.na(approx(pxx$x, pxx$y, xout=V[t,3])$y))
128     p_xx[1] <- 0
129   if(!is.na(approx(pyy$x, pyy$y, xout=V[t,4])$y))
130     p_yy <- integrate(approxfun(pyy), lower=min(forKDE2$yy), upper=max(forKDE2$yy))
131   if(is.na(approx(pyy$x, pyy$y, xout=V[t,4])$y))
132     p_yy[1] <- 0
133   p2 <- as.numeric(p_xx[1])*as.numeric(p_yy[1])
134
135   P1[i,1] <- P1[i,1]*p1*p2
136   P1[i,2] <- route[1,3]
137 }
138 }
139
140 P2 <- setNames(data.frame(matrix(1, nrow=k, ncol=2)), c("prob2", "destination"))
141 for(i in 1:k)
142 {
143   routesNames <- S %>% distinct(wp_b)
144   route <- filter(Rs, wp_b==routesNames[i,1])
145   step <- 6
146   predicted <- Track_Predictor(route, VOI, VOI1[1,7], VOI1[7,7], step, d)
147   actual <- VOI1[2:7,8:9]
148   distances <- c()
149   for(j in 1:nrow(predicted))
150     {
151       distance <- as.numeric(distm(c(actual[j,1], actual[j,2]),
152                                   c(predicted[j,1], predicted[j,2]), fun = distHaversine))
153       distances <- c(distances, distance)
154     }
155   mle <- eweibull(distances, method = "mle")
156   alpha <- mle[["parameters"]][["scale"]]
157   beta <- mle[["parameters"]][["shape"]]
158
159   for(t in 2:nrow(VOI1))

```

```

160 {
161   delta_t <- period_to_seconds(hms(VOI1[t,7]))-period_to_seconds(hms(VOI[t-1,7]))
162   Delta_p <- distances[t-1]
163   m_k <- alpha/delta_t
164   p <- exp(-(Delta_p/(m_k*delta_t))^beta)
165   P2[i,1] <- P2[i,1]*p
166   P2[i,2] <- route[1,3]
167 }
168 }
169
170 prior <- setNames(data.frame(matrix(nrow=k,ncol=2)),c("prior","route"))
171 for(i in 1:k)
172 {
173   routesNames <- S %>% distinct(wp_b)
174   route <- filter(Rs, wp_b==routesNames[i,1])
175   nmbrVessel <- nrow(route %>% distinct(MMSI))
176   whenDetected <- which(Vs$MMSI == VOI[1,1])
177   prior[i,1] <- nmbrVessel/whenDetected
178   prior[i,2] <- route[1,3]
179 }
180
181 P <- cbind(P1,P2,prior)
182 for(a in 1:nrow(P))
183 {
184   P[a,7] <- prod(P[a,1],P[a,3],P[a,5],100)
185 }
186
187 #####
188 # drugo promatranje -> d = 2.38 nauticke milje
189 VOI1 <- VOI[1501:1507,]
190 speed <- conv_unit(VOI1$track.SOG, "knot", "m_per_sec")
191 VOI1$SOG_m <- speed
192 V <- setNames(data.frame(matrix(ncol=6, nrow=0)), c("x","y","xx","yy","x_u","y_u"))
193 S <- NULL
194 d <- 4417
195
196 for(i in 1:nrow(VOI1))
197 {
198   xx <- sqrt((VOI1[i,14])^2/(1+(tan(VOI1[i,5]))^2))
199   yy <- sqrt((VOI1[i,14])^2-xx^2)
200   x <- VOI1[i,8]
201   y <- VOI1[i,9]
202   x_u <- VOI1[i,3]
203   y_u <- VOI1[i,4]
204   V[i,c(1:6)] <- c(x, y, xx, yy, x_u, y_u)
205   tem <- filter(Rs, ((Rs$track.Longitude_UTM-x_u)^2+(Rs$track.Latitude_UTM-y_u)^2)<d^2)
206   tem <- cbind(tem, i)
207   S <- rbind(S, tem)
208 }
209 k <- nrow(S %>% distinct(wp_b))
210
211 P1 <- setNames(data.frame(matrix(1,nrow=k,ncol=2)),c("prob1","destination"))
212 for(i in 1:k)
213 {
214   routesNames <- S %>% distinct(wp_b)
215   route <- filter(Rs, wp_b==routesNames[i,1])
216   for(t in 1:nrow(VOI1))
217   {
218     st <- filter(S, S[,12]==t)
219     stt <- st[1:11]

```

```

220
221 # KDE za položaj
222 forKDE1 <- inner_join(route, stt, by=c("name", "wp_a", "wp_b", "MMSI",
223   "track.Longitude_UTM", "track.Latitude_UTM", "track.COG",
224   "track.SOG", "track.timestamp", "track.Longitude", "track.Latitude"))
225 forKDE1 <- forKDE1[, 10:11]
226 px <- density(forKDE1$track.Longitude, bw="nrd0")
227 py <- density(forKDE1$track.Latitude, bw="nrd0")
228
229 if(!is.na(approx(px$x, px$y, xout=V[t, 1])$y))
230 {
231   p_x <- integrate(approxfun(px), lower=min(forKDE1$track.Longitude),
232     upper=max(forKDE1$track.Longitude))
233 }
234 if(is.na(approx(px$x, px$y, xout=V[t, 1])$y))
235   p_x[1] <- 0
236 if(!is.na(approx(py$x, py$y, xout=V[t, 2])$y))
237 {
238   p_y <- integrate(approxfun(py), lower=min(forKDE1$track.Latitude),
239     upper=max(forKDE1$track.Latitude))
240 }
241 if(is.na(approx(py$x, py$y, xout=V[t, 2])$y))
242   p_y[1] <- 0
243 p1 <- as.numeric(p_x[1])*as.numeric(p_y[1])
244
245 # KDE za brzinu i kurs
246 forKDE2 <- inner_join(route, stt, by = c("name", "wp_a", "wp_b", "MMSI",
247   "track.Longitude_UTM", "track.Latitude_UTM", "track.COG", "track.SOG",
248   "track.timestamp", "track.Longitude", "track.Latitude"))
249 speed <- conv_unit(forKDE2$track.SOG, "knot", "m_per_sec")
250 forKDE2$SOG_m <- speed
251 xx <- sqrt((forKDE2$SOG_m)^2/(1+(tan(forKDE2$track.COG))^2))
252 forKDE2$xx <- xx
253 yy <- sqrt((forKDE2$SOG_m)^2-(forKDE2$xx)^2)
254 forKDE2$yy <- yy
255 forKDE2 <- forKDE2[, 13:14]
256 pxx <- density(forKDE2$xx, bw="nrd0")
257 pyy <- density(forKDE2$yy, bw="nrd0")
258
259 if(!is.na(approx(pxx$x, pxx$y, xout=V[t, 3])$y))
260   p_xx <- integrate(approxfun(pxx), lower=min(forKDE2$xx), upper=max(forKDE2$xx))
261 if(is.na(approx(pxx$x, pxx$y, xout=V[t, 3])$y))
262   p_xx[1] <- 0
263 if(!is.na(approx(pyy$x, pyy$y, xout=V[t, 4])$y))
264   p_yy <- integrate(approxfun(pyy), lower=min(forKDE2$yy), upper=max(forKDE2$yy))
265 if(is.na(approx(pyy$x, pyy$y, xout=V[t, 4])$y))
266   p_yy[1] <- 0
267 p2 <- as.numeric(p_xx[1])*as.numeric(p_yy[1])
268
269 P1[i, 1] <- P1[i, 1]*p1*p2
270 P1[i, 2] <- route[1, 3]
271 }
272 }
273
274 P2 <- setNames(data.frame(matrix(1, nrow=k, ncol=2)), c("prob2", "destination"))
275 for(i in 1:k)
276 {
277   routesNames <- S %>% distinct(wp_b)
278   route <- filter(Rs, wp_b==routesNames[i, 1])
279   step <- 6

```

```

280 predicted <- Track_Predictor(route, VOI, VOI1[1,7], VOI1[7,7], step, d)
281 actual <- VOI1[2:7,8:9]
282 distances <- c()
283 for(j in 1:nrow(predicted))
284 {
285   distance <- as.numeric(distm(c(actual[j,1],actual[j,2]),
286                               c(predicted[j,1],predicted[j,2]), fun = distHaversine))
287   distances <- c(distances, distance)
288 }
289 mle <- eweibull(distances, method = "mle")
290 alpha <- mle[["parameters"]][["scale"]]
291 beta <- mle[["parameters"]][["shape"]]
292
293 for(t in 2:nrow(VOI1))
294 {
295   delta_t <- period_to_seconds(hms(VOI1[t,7]))-period_to_seconds(hms(VOI[t-1,7]))
296   Delta_p <- distances[t-1]
297   m_k <- alpha/delta_t
298   p <- exp(-(Delta_p/(m_k*delta_t))^beta)
299   P2[i,1] <- P2[i,1]*p
300   P2[i,2] <- route[1,3]
301 }
302 }
303
304 prior <- setNames(data.frame(matrix(nrow=k,ncol=2)),c("prior","route"))
305 for(i in 1:k)
306 {
307   routesNames <- S %>% distinct(wp_b)
308   route <- filter(Rs, wp_b==routesNames[i,1])
309   nmbrVessel <- nrow(route %>% distinct(MMSI))
310   whenDetected <- which(Vs$MMSI == VOI[1,1])
311   prior[i,1] <- nmbrVessel/whenDetected
312   prior[i,2] <- route[1,3]
313 }
314
315 P <- cbind(P1,P2,prior)
316 for(a in 1:nrow(P))
317 {
318   P[a,7] <- prod(P[a,1],P[a,3],P[a,5],100)
319 }
320
321 #####
322 # 2) Predvidanje polozaja
323 VOI1 <- VOI[1:7,]
324 speed <- conv_unit(VOI1$track.SOG, "knot", "m_per_sec")
325 VOI1$SOG_m <- speed
326 V <- setNames(data.frame(matrix(ncol=6, nrow=0)), c("x","y","xx","yy","x_u","y_u"))
327 S <- NULL
328 d <- 6173
329 step <- 6
330
331 for(i in 1:nrow(VOI1))
332 {
333   xx <- sqrt((VOI1[i,14])^2/(1+(tan(VOI1[i,5]))^2))
334   yy <- sqrt((VOI1[i,14])^2-xx^2)
335   x <- VOI1[i,8]
336   y <- VOI1[i,9]
337   x_u <- VOI1[i,3]
338   y_u <- VOI1[i,4]
339   V[i,c(1:6)] <- c(x, y, xx, yy, x_u, y_u)

```



```

340 tem <- filter(Rs, ((Rs$track.Longitude_UTM-x_u)^2+(Rs$track.Latitude_UTM-y_u)^2)<d^2)
341 tem <- cbind(tem, i)
342 S <- rbind(S, tem)
343 }
344
345 routesNames <- S %>% distinct(wp_b)
346 route <- filter(Rs, wp_b=="PO 2")
347 predicted <- Track_Predictor(route, VOI, VOI1[1,7], VOI1[7,7], step, d)
348 predicted <- predicted %>% distinct(xp,yp,.keep_all = TRUE)
349 actual <- VOI1[2:7,8:9]
350 compare <- cbind(actual, predicted)
351
352 #####
353 # 3) Detekcija anomalija
354 VOI <- 'Vs_ 212491000'
355 VOI1 <- VOI[1:7,]
356 speed <- conv_unit(VOI1$track.SOG, "knot", "m_per_sec")
357 VOI1$SOG_m <- speed
358 V <- setNames(data.frame(matrix(ncol=6, nrow=0)), c("x", "y", "xx", "yy", "x_u", "y_u"))
359 S <- NULL
360 d <- 7050
361
362 for(i in 1:nrow(VOI1))
363 {
364   xx <- sqrt((VOI1[i,14])^2/(1+(tan(VOI1[i,5]))^2))
365   yy <- sqrt((VOI1[i,14])^2-xx^2)
366   x <- VOI1[i,8]
367   y <- VOI1[i,9]
368   x_u <- VOI1[i,3]
369   y_u <- VOI1[i,4]
370   V[i,c(1:6)] <- c(x, y, xx, yy, x_u, y_u)
371   tem <- filter(Rs, ((Rs$track.Longitude_UTM-x_u)^2+(Rs$track.Latitude_UTM-y_u)^2)<d^2)
372   tem <- cbind(tem, i)
373   S <- rbind(S, tem)
374 }
375 k <- nrow(S %>% distinct(wp_b))
376
377 P1 <- setNames(data.frame(matrix(1, nrow=k, ncol=2)), c("prob1", "destination"))
378 for(i in 1:k)
379 {
380   routesNames <- S %>% distinct(wp_b)
381   route <- filter(Rs, wp_b==routesNames[i,1])
382   for(t in 1:nrow(VOI1))
383   {
384     st <- filter(S, S[,12]==t)
385     stt <- st[1:11]
386
387     # KDE za polozaj
388     forKDE1 <- inner_join(route, stt, by=c("name", "wp_a", "wp_b", "MMSI",
389       "track.Longitude_UTM", "track.Latitude_UTM", "track.COG",
390       "track.SOG", "track.timestamp", "track.Longitude", "track.Latitude"))
391     forKDE1 <- forKDE1[,10:11]
392     px <- density(forKDE1$track.Longitude, bw="nrd0")
393     py <- density(forKDE1$track.Latitude, bw="nrd0")
394
395     if(!is.na(approx(px$x, px$y, xout=V[t,1])$y))
396     {
397       p_x <- integrate(approxfun(px), lower=min(forKDE1$track.Longitude),
398         upper=max(forKDE1$track.Longitude))
399     }

```

```

400   if(is.na(approx(px$x, px$y, xout=V[t,1])$y))
401     p_x[1] <- 0
402   if(!is.na(approx(py$x, py$y, xout=V[t,2])$y))
403     {
404       p_y <- integrate(approxfun(py), lower=min(forKDE1$track.Latitude),
405                        upper=max(forKDE1$track.Latitude))
406     }
407   if(is.na(approx(py$x, py$y, xout=V[t,2])$y))
408     p_y[1] <- 0
409   p1 <- as.numeric(p_x[1])*as.numeric(p_y[1])
410
411   # KDE za brzinu i kurs
412   forKDE2 <- inner_join(route, stt, by = c("name", "wp_a", "wp_b", "MMSI",
413     "track.Longitude_UTM", "track.Latitude_UTM", "track.COG", "track.SOG",
414     "track.timestamp", "track.Longitude", "track.Latitude"))
415   speed <- conv_unit(forKDE2$track.SOG, "knot", "m_per_sec")
416   forKDE2$SOG_m <- speed
417   xx <- sqrt((forKDE2$SOG_m)^2/(1+(tan(forKDE2$track.COG))^2))
418   forKDE2$xx <- xx
419   yy <- sqrt((forKDE2$SOG_m)^2-(forKDE2$xx)^2)
420   forKDE2$yy <- yy
421   forKDE2 <- forKDE2[,13:14]
422   pxx <- density(forKDE2$xx, bw="nrd0")
423   pyy <- density(forKDE2$yy, bw="nrd0")
424
425   if(!is.na(approx(pxx$x, pxx$y, xout=V[t,3])$y))
426     p_xx <- integrate(approxfun(pxx), lower=min(forKDE2$xx), upper=max(forKDE2$xx))
427   if(is.na(approx(pxx$x, pxx$y, xout=V[t,3])$y))
428     p_xx[1] <- 0
429   if(!is.na(approx(pyy$x, pyy$y, xout=V[t,4])$y))
430     p_yy <- integrate(approxfun(pyy), lower=min(forKDE2$yy), upper=max(forKDE2$yy))
431   if(is.na(approx(pyy$x, pyy$y, xout=V[t,4])$y))
432     p_yy[1] <- 0
433   p2 <- as.numeric(p_xx[1])*as.numeric(p_yy[1])
434
435   P1[i,1] <- P1[i,1]*p1*p2
436   P1[i,2] <- route[1,3]
437 }
438 }
439
440 P2 <- setNames(data.frame(matrix(1, nrow=k, ncol=2)), c("prob2", "destination"))
441 for(i in 1:k)
442 {
443   routesNames <- S %>% distinct(wp_b)
444   route <- filter(Rs, wp_b==routesNames[i,1])
445   step <- 6
446   predicted <- Track_Predictor(route, VOI, VOI1[1,7], VOI1[7,7], step, d)
447   actual <- VOI1[2:7,8:9]
448   distances <- c()
449   for(j in 1:nrow(predicted))
450     {
451       distance <- as.numeric(distm(c(actual[j,1], actual[j,2]),
452                                   c(predicted[j,1], predicted[j,2]), fun = distHaversine))
453       distances <- c(distances, distance)
454     }
455   mle <- eweibull(distances, method = "mle")
456   alpha <- mle[["parameters"]][["scale"]]
457   beta <- mle[["parameters"]][["shape"]]
458
459   for(t in 2:nrow(VOI1))

```

```

460 {
461   delta_t <- period_to_seconds(hms(VOI1[t,7]))-period_to_seconds(hms(VOI[t-1,7]))
462   Delta_p <- distances[t-1]
463   m_k <- alpha/delta_t
464   p <- exp(-(Delta_p/(m_k*delta_t))^beta)
465   P2[i,1] <- P2[i,1]*p
466   P2[i,2] <- route[1,3]
467 }
468 }
469
470 prior <- setNames(data.frame(matrix(nrow=k,ncol=2)),c("prior","route"))
471 for(i in 1:k)
472 {
473   routesNames <- S %>% distinct(wp_b)
474   route <- filter(Rs, wp_b==routesNames[i,1])
475   nmbrVessel <- nrow(route %>% distinct(MMSI))
476   whenDetected <- which(Vs$MMSI == VOI[1,1])
477   prior[i,1] <- nmbrVessel/whenDetected
478   prior[i,2] <- route[1,3]
479 }
480
481 P <- cbind(P1,P2,prior)
482 for(a in 1:nrow(P))
483 {
484   P[a,7] <- prod(P[a,1],P[a,3],P[a,5],100)
485 }

```

Dodatak C

Kod izvedbe grafičkih prikaza

```
1 library(readr)
2 library(maps)
3 library(ggplot2)
4 library(tidyverse)
5 library(dplyr)
6 library(mapproj)
7 library(maptools)
8 library(geosphere)
9
10 summary(rawData)
11 data <- rawData
12
13 # Prikaz podataka na mapi
14 mp <- NULL
15 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
16 mp <- ggplot() + mapWorld
17 mp <- mp + geom_point(data=data, aes(x=Longitude, y=Latitude), color="blue", shape=".",
18   size=2) + scale_fill_grey() + coord_quickmap(ylim=c(50,65), xlim=c(0,25))
19 mp
20
21 # Odabrano područje
22 line1 <- data.frame(xx=c(10,10), yy=c(54.1,54.8))
23 line2 <- data.frame(xx=c(10,12.5), yy=c(54.1,54.1))
24 line3 <- data.frame(xx=c(12.5,12.5), yy=c(54.1,54.8))
25 line4 <- data.frame(xx=c(10,12.5), yy=c(54.8,54.8))
26
27 # Prikaz promatranog područja
28 mp1 <- NULL
29 mapWorld <- borders("world", colour = "gray20", fill = "palegreen4")
30 mp1 <- ggplot() + mapWorld
31 mp1 <- mp1 + geom_point(data=data, aes(x=Longitude, y=Latitude), color="blue", shape=".",
32   size=2) + scale_fill_grey() + coord_quickmap(ylim=c(50,65), xlim=c(0,25)) +
33   geom_line(data=line1, aes(x=xx, y=yy), color="black", size=1)+
34   geom_line(data=line2, aes(x=xx, y=yy), color="black", size=1)+
35   geom_line(data=line3, aes(x=xx, y=yy), color="black", size=1)+
36   geom_line(data=line4, aes(x=xx, y=yy), color="black", size=1)
37 mp1
38
39 selectedData <- filter(rawData, Longitude>=10 & Longitude<=12.5
40   & Latitude>=54.1 & Latitude<=54.8)
```

```

41
42 # Priblizeni prikaz odabranog podrucja
43 ports <- data.frame(longitude=c(10.15,12.10), latitude=c(54.32,54.15))
44 mp2 <- NULL
45 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
46 mp2 <- ggplot() + mapWorld
47 mp2 <- mp2 + geom_point(data=selectedData, aes(x=Longitude, y=Latitude), color="blue",
48   shape=".", size=2) + scale_fill_grey() +
49   coord_quickmap(ylim=c(54.1,54.8), xlim=c(10,12.5))+
50   geom_point(data=ports, aes(x=longitude, y=latitude), fill="darkred",
51     shape=23, size=5)
52 mp2 <- mp2 +geom_line(data=line1, aes(x=xx, y=yy), color="black", size=1)+
53   geom_line(data=line2, aes(x=xx, y=yy), color="black", size=1)+
54   geom_line(data=line3, aes(x=xx, y=yy), color="black", size=1)+
55   geom_line(data=line4, aes(x=xx, y=yy), color="black", size=1)
56 mp2
57
58 messages <- selectedData
59
60 # Razdioba plovila po tipu AIS uredaja
61 mes <- messages %>% distinct(MMSI, .keep_all=TRUE)
62 barplot1data <- table(mes$'Type of mobile')
63 ylim <- c(0, 1.1*max(barplot1data))
64 barplot1 <- barplot(barplot1data, ylim=ylim, main="Razdioba plovila po tipu AIS uredaja"
65   ,ylab="Broj plovila", xlab="Tip AIS uredaja",col="#104E8B")
66 text(x=barplot1, y=barplot1data, label=barplot1data, pos=3, cex=0.8, col="black")
67
68 # Razdioba plovila po tipu plovila
69 op <- par(mar=c(7,4,4,2))
70 mess <- messages %>% distinct(MMSI, .keep_all=TRUE)
71 barplot2data <- table(mess$'Ship type')
72 ylim <- c(0, 1.1*max(barplot2data))
73 barplot2 <- barplot(barplot2data, ylim=ylim, main="Razdioba plovila po tipu plovila"
74   ,ylab="Broj plovila",col="#104E8B",las=2,cex.axis=0.5,cex.names=0.9,
75   width=5)
76 text(x=barplot2, y=barplot2data, label=barplot2data, pos=3, cex=0.9, col="black")
77
78 # Prikaz rezultata nenadgledanog izdvajanja ruta
79 ENN <- filter(ENs, cluster!=0)
80 POO <- filter(POs, cluster!=0)
81 EXX <- filter(EXs, cluster!=0)
82 ports <- data.frame(longitude=c(10.15,12.10), latitude=c(54.32,54.15))
83 mp6 <- NULL
84 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
85 mp6 <- ggplot() + mapWorld
86 mp6 <- mp6 + geom_point(data=Rs, aes(x=track.Longitude, y=track.Latitude),
87   color="blue", shape=".", size=2)+scale_fill_grey() +
88   coord_quickmap(ylim=c(54.1,54.8),xlim=c(10,12.5)) +
89   geom_point(data=ports, aes(x=longitude, y=latitude),
90     fill="darkred", shape=23, size=5)
91 mp6 <- mp6 + geom_line(data=line1, aes(x=xx, y=yy), color="black", size=1)+
92   geom_line(data=line2, aes(x=xx, y=yy), color="black", size=1)+
93   geom_line(data=line3, aes(x=xx, y=yy), color="black", size=1)+
94   geom_line(data=line4, aes(x=xx, y=yy), color="black", size=1)
95 mp6 <- mp6 + geom_point(aes(x=ENN[,1], y=ENN[,2]), colour="green2", size=2,
96   data=ENN)
97 mp6 <- mp6 + geom_point(aes(x=POO[,1], y=POO[,2]), colour="gold1", size=2,
98   data=POO)
99 mp6 <- mp6 + geom_point(aes(x=EXX[,1], y=EXX[,2]), colour="red3", size=2,
100   data=EXX)

```

```

101 mp6
102
103 # Prikaz kompatibilnih ruta i najizglednija
104 # 1.1
105 compRoute <- S %>% distinct(wp_b, wp_a)
106 route1 <- filter(Rs, Rs$wp_b==compRoute[1,1] & Rs$wp_a==compRoute[1,2])
107 route2 <- filter(Rs, Rs$wp_b==compRoute[3,1] & Rs$wp_a==compRoute[3,2])
108 route3 <- filter(Rs, Rs$wp_b==compRoute[6,1] & Rs$wp_a==compRoute[6,2])
109 df <- data.frame(x=c(10.35, 10.95, 12.3),y = c(54.3, 54.75, 54.15),
110   text = c("54.22%", "13.13%", "32.66%"))
111
112 ports <- data.frame(longitude=c(10.15,12.10), latitude=c(54.32,54.15))
113 mp6 <- NULL
114 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
115 mp6 <- ggplot() + mapWorld
116 mp6 <- mp6 + scale_fill_grey() + coord_quickmap(ylim=c(54.1,54.8),xlim=c(10,12.5)) +
117   geom_point(data=ports, aes(x=longitude, y=latitude),
118     fill="darkred", shape=23, size=5) +
119   geom_line(data=line1, aes(x=xx, y=yy), color="black", size=1)+
120   geom_line(data=line2, aes(x=xx, y=yy), color="black", size=1)+
121   geom_line(data=line3, aes(x=xx, y=yy), color="black", size=1)+
122   geom_line(data=line4, aes(x=xx, y=yy), color="black", size=1)
123 mp6 <- mp6+ geom_point(data=route2, aes(x=track.Longitude, y=track.Latitude),
124   color="skyblue3", shape=".", size=2)
125 mp6 <- mp6+ geom_point(data=route1, aes(x=track.Longitude, y=track.Latitude),
126   color="navyblue", shape=".", size=2)
127 mp6 <- mp6+ geom_point(data=route3, aes(x=track.Longitude, y=track.Latitude),
128   color="skyblue3", shape=".", size=2)
129 mp6 <- mp6+ geom_point(data=VOI1, aes(x=track.Longitude, y=track.Latitude),
130   color="maroon4", shape=15, size=4)
131 mp6 <- mp6 + geom_text(aes(x=df$x,y=df$y,label=df$text), size=3)
132 mp6
133
134 # 1.2
135 compRoute <- S %>% distinct(wp_b, wp_a)
136 route1 <- filter(Rs, Rs$wp_b==compRoute[1,1] & Rs$wp_a==compRoute[1,2])
137 route2 <- filter(Rs, Rs$wp_b==compRoute[7,1] & Rs$wp_a==compRoute[7,2])
138 route3 <- filter(Rs, Rs$wp_b==compRoute[8,1] & Rs$wp_a==compRoute[8,2])
139 df <- data.frame(x=c(10.35, 10.7, 10.99),y = c(54.3, 54.70, 54.75),
140   text = c("66.72%", "27.20%", "6.17%"))
141
142 ports <- data.frame(longitude=c(10.15,12.10), latitude=c(54.32,54.15))
143 mp6 <- NULL
144 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
145 mp6 <- ggplot() + mapWorld
146 mp6 <- mp6 + scale_fill_grey() + coord_quickmap(ylim=c(54.1,54.8),xlim=c(10,12.5)) +
147   geom_point(data=ports, aes(x=longitude, y=latitude),
148     fill="darkred", shape=23, size=5) +
149   geom_line(data=line1, aes(x=xx, y=yy), color="black", size=1)+
150   geom_line(data=line2, aes(x=xx, y=yy), color="black", size=1)+
151   geom_line(data=line3, aes(x=xx, y=yy), color="black", size=1)+
152   geom_line(data=line4, aes(x=xx, y=yy), color="black", size=1)
153 mp6 <- mp6+ geom_point(data=route3, aes(x=track.Longitude, y=track.Latitude),
154   color="skyblue3", shape=".", size=2)
155 mp6 <- mp6+ geom_point(data=route1, aes(x=track.Longitude, y=track.Latitude),
156   color="navyblue", shape=".", size=2)
157 mp6 <- mp6+ geom_point(data=route2, aes(x=track.Longitude, y=track.Latitude),
158   color="skyblue3", shape=".", size=2)
159 mp6 <- mp6+ geom_point(data=VOI1, aes(x=track.Longitude, y=track.Latitude),
160   color="maroon4", shape=15, size=4)

```

```
161 mp6 <- mp6 + geom_text(aes(x=df$x,y=df$y,label=df$text), size=3)
162 mp6
163
164 # Predvidanje položaja
165 mp6 <- NULL
166 mapWorld <- borders("world", colour="gray20", fill="palegreen4")
167 mp6 <- ggplot() + mapWorld
168 mp6 <- mp6 + geom_point(data=actual, aes(x=track.Longitude, y=track.Latitude),
169                          color="blue", shape=1, size=3) + scale_fill_grey() +
170   coord_quickmap(ylim=c(54.705,54.71),xlim=c(12.495,12.501)) +
171   geom_point(data=predicted, aes(x=xp, y=yp),
172             color="red", shape=4, size=3)+
173   geom_point(data=VOI1[1,], aes(x=track.Longitude, y=track.Latitude),
174             color="green", shape=16, size=3)
175 mp6
```