Analiza repova čekanja i razine uslužnosti urbanih prometnica korištenjem algoritama strojnog učenja i NoSQL baza podataka

Tišljarić, Leo

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:716383

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Leo Tišljarić

ANALIZA REPOVA ČEKANJA I RAZINE USLUŽNOSTI URBANIH PROMETNICA KORIŠTENJEM ALGORITAMA STROJNOG UČENJA I NOSQL BAZA PODATAKA

DIPLOMSKI RAD

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 3. travnja 2018.

Zavod: **Zavod za inteligentne transportne sustave**
Predmet: **Napredne baze podataka**

DIPLOMSKI ZADATAK br. 4681

Pristupnik: **Leo Tišljarić (0135235019)**
Studij: Inteligentni transportni sustavi i logistika
Smjer: Inteligentni transportni sustavi

Zadatak: **Analiza repova čekanja i razine uslužnosti urbanih prometnica korištenjem algoritama strojnog učenja i NoSQL baza podataka**

Opis zadatka:

Prikupljene GPS podatke o gibanju vozila potrebno je pohraniti u bazu podataka ili u neku drugu podatkovnu strukturu koja će uz odabir prikladnog načina indeksiranja dohvaćati podatke aplikaciji u Python programskom jeziku koja primjenjuje tehnike strojnog učenja. Zadatak aplikacije je odrediti razinu uslužnosti raskrižja i prikazati metodu računanja duljine repova čekanja na raskrižju.

Mentor:

prof. dr. sc. Tonči Carić

Predsjednik povjerenstva za diplomski ispit:

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

DIPLOMSKI RAD

ANALIZA REPOVA ČEKANJA I RAZINE USLUŽNOSTI URBANIH PROMETNICA KORIŠTENJEM ALGORITAMA STROJNOG UČENJA I NOSQL BAZA PODATAKA

ANALYSIS OF QUEUES AND LEVEL OF SERVICE ON URBAN ROADS USING MACHINE LEARNING ALGORITHMS AND NOSQL DATABASE

Mentor:
dr. sc. Tonči Carić

Student: Leo Tišljarić
JMBAG: 0135235019

Zagreb, srpanj 2018.

Zahvaljujem se prof. dr. sc. Tončiju Cariću na mentorstvu tijekom studiranja i prilikom izrade ovog rada. Zahvaljujem se dr. sc. Juraju Fosinu na savjetima i Tomislavu Erdeliću, mag. ing. el. techn. inf. na iznimnom strpljenju i pomoći prilikom tehničke izvedbe ovog rada. Zahvaljujem se i svojoj obitelji koja me podržala tijekom cijelog trajanja studija.

ANALIZA REPOVA ČEKANJA I RAZINE USLUŽNOSTI URBANIH PROMETNICA KORIŠTENJEM ALGORITAMA STROJNOG UČENJA I NOSQL BAZA PODATAKA

Sažetak:

Kvalitetni prometni podaci osnovna su polazišna točka prilikom procjene stanja prometnog sustava ili prijedloga mogućih poboljšanja. Brzina vozila i vrijeme provedeno čekajući na raskrižju osnovni su parametri za procjenu razine uslužnosti neke prometnice ili raskrižja. U ovom radu opisana je metoda prikupljanja i obrade GPS podataka u svrhu procjene osnovnih parametara za opisivanje stanja prometnog toka i razine uslužnosti za prometnice i raskrižja. Prikazana je metoda procjene duljine repova čekanja na semaforiziranim raskrižjima te je primijenjen algoritam strojnog učenja kako bi se grupirala raskrižja na temelju mjerenih mjera učinkovitosti. Za potrebu izračuna svih navedenih parametara izrađena je aplikacija korištenjem Python programskog jezika i MongoDB NoSQL baze podataka. Aplikacija sadrži tri glavne funkcionalnosti: predobrada podataka i unos u bazu podataka, izračun karakterističnih parametara raskrižja i algoritam strojnog učenja za grupiranje podataka.

Ključne riječi: strojno učenje, rep čekanja, razina uslužnosti, grupiranje, GPS, Python, NoSQL, K-means

ANALYSIS OF QUEUES AND LEVELS OF SERVICE ON URBAN ROADS USING MACHINE LEARNING ALGORITHMS AND NOSQL DATABASE

Summary:

Traffic data is main starting point for estimating performance or proposing improvements for traffic system. Vehicle speed and control delay are main parameters in Level of service asesment for roads and intersections. In this thesis methods for collecting, storing and processing GPS data in order to estimate traffic parameters is proposed. Method for estimating queue length on intersections approaches is presented and machine learning algorithm is used for clustering intersections based on their values of measured performance indicators. For calculations and results, GUI based Python application is developed using MongoDB NoSQL database for storing data and results. Application contains three main features: processing and storing data, calculations of performance measures for intersections and implementation of machine learning algorithm for clustering data.

Key words: Machine learning, Queue length, Level of service, clustering, GPS, Python, NoSQL, K-means

Sadržaj

1 Uvod **1**

1.1 Ciljevi i svrha rada . . . 2
1.2 Dosadašnja istraživanja . . . 3
1.3 Struktura rada . . . 3

2 Urbane prometnice **5**

2.1 Prometni parametri . . . 6
2.1.1 Brzina . . . 6
2.1.2 Protok . . . 8
2.1.3 Gustoća . . . 9
2.1.4 Odnosi između prometnih parametara . . . 10
2.2 Prikupljanje podataka . . . 11
2.2.1 Testna vozila . . . 12
2.2.2 Detekcija registarskih tablica . . . 12
2.2.3 Ispitno vozilo u ITS okruženju . . . 13

3 Obrada podataka **14**

3.1 Struktura ulaznih podataka . . . 15
3.2 Predobrada podataka . . . 17
3.2.1 Sezonalnost . . . 17
3.2.2 Podjela podataka . . . 18
3.3 NoSQL baze podataka . . . 19
3.3.1 Vrste NoSQL baza podataka . . . 20
3.3.2 Razlike u odnosu na SQL baze podataka . . . 20
3.4 MongoDB . . . 22
3.4.1 Pohrana podataka . . . 22
3.4.2 Usporedba s Microsoft SQL Server . . . 23

4 Razine uslužnosti i repovi čekanja **28**

4.1 Razina uslužnosti urbane prometnice 28
4.2 Razina uslužnosti semaforiziranog raskrižja 30
4.3 Određivanje razine uslužnosti pomoću GPS zapisa 31
4.3.1 Izračun pada brzine FFSP i razine uslužnosti 31
4.3.2 Izračun upravljačkog vremena kašnjenja CD i razine uslužnosti 32
4.4 Određivanje repova čekanja pomoću GPS zapisa 33

5 Programsko rješenje **36**
5.1 Baza podataka 36
5.2 Programski jezik 37
5.3 Strojno učenje 38
5.3.1 Podjela algoritama strojnog učenja 38
5.3.2 Problemi 40
5.3.3 K-means 42
5.4 Grafičko sučelje 46

6 Rezultati istraživanja **49**
6.1 Repovi čekanja na prilazima raskrižjima 49
6.2 Razina uslužnosti 52
6.3 Detekcija sličnih raskrižja 53

7 Zaključak **57**

Bibliografija **59**

Popis kratica **63**

Popis ilustracija **64**

Popis tablica **65**

Popis kôdova **66**

Prilog A: Rezultati procjene repa čekanja **68**

Prilog B: Rezultati procjene LOS **72**

Poglavlje 1

Uvod

Za procjenu stanja prometnog sustava kao i prijedloga mogućih poboljšanja potrebni su kvalitetni prometni podaci. Parametri prometnice ili raskrižja koji se najčešće prikupljaju su brzina i gustoća prometnog toka, broj vozila, duljina repa čekanja, vrijeme čekanja i dr. Svaki od navedenih parametara koristi se zasebno ili u kombinaciji kako bi se procijenila razina uslužnosti ili propusna moć prometnice ili raskrižja, [1]. *Highway Capacity Manual* (HCM) definira osnovne mjere učinkovitosti prometnica i raskrižja te opisuje načine definiranja razine usluge te generalnu procjenu kvalitete prometovanja prometnom infrastrukturom. Autori navode kako je brzina vozila osnovna mjera za procjenu razine uslužnosti svih prometnica, a vrijeme čekanja uzrokovano prometnom signalizacijom osnovna mjera za procjenu razine uslužnosti raskrižja. Razina uslužnosti (*engl. Level of service* - LOS) predstavlja kvalitativnu mjeru koja se koristi za određivanje performansi prometnog infrastrukturnog objekta koji se razmatra, a može biti prometnica ili raskrižje. Određuje se za za različite modove prijevoza: automobilski, pješački, biciklistički i dr.

Detekcija smanjene učinkovitosti prvi je korak prilikom rješavanja prometnih problema stoga postoji opravdana potreba za sustavima koji će upravo to omogućiti. Kontinuiranim praćenjem prethodno navedenih prometnih parametara moguće je na vrijeme detektirati i poduzeti pravovremene akcije kako bi se spriječila prometna zagušenja ili umanjile posljedice istih. Navedenim se mogu smanjiti ekonomski i ekološki troškovi uzrokovani čekanjem u koloni: vrijeme čekanja (radno vrijeme), potrošnja goriva, emisija stakleničkih plinova i dr.

Danas se algoritmi strojnog učenja koriste u mnogim aplikacijama i područjima (sektorima) od detekcije neželjene pošte, prijedloga marketinškog sadržaja, upravljanja prometom i transportom do upravljanja autonomnim vozilima. Zajednička karakteristika ovih aplikacije je ta da se odluke donose na temelju velikog broja podataka koje čovjek u realnom vremenu ne može obraditi niti donijeti odluku. U ovom radu svako analizirano raskrižje prikazano je s vektorom karakterističnih vrijednosti prometnih parametara raskrižja duljine n. Ako se svi vektori ka-

rakterističnih vrijednosti smjeste u n-dimenzionalni prostor moguće je problem definirati kao problem grupiranja (*engl. Clustering*) koji spada u probleme nenadziranog učenja. Cilj nenadziranog učenja je podijeliti ulazni neoznačeni skup podataka u grupe, odnosni prepoznati obrazac na temelju zajedničkih karakteristika. Algoritam strojnog učenja koji se primjenjuje treba neoznačene podatke koji opisuju raskrižja grupirati u grupe (klastere) prema danih prometnih parametara. Na taj način raskrižja će biti grupirana na temelju karakterističnih parametara te će se prema koeficijentima učinkovitosti "slična" raskrižja grupirati u iste grupe. Grupe će tada biti rangirane od "najbolje" do"najgore".

1.1 Ciljevi i svrha rada

U ovom radu obrađeni su povijesni podaci u svrhu izračuna prometnih parametara i detekcije raskrižja s niskom razinom učinkovitosti. Povijesni podaci sastavljeni su od *Global Positioning System* (GPS) zapisa praćenih vozila u periodu od pet godina od kolovoza 2009. do listopada 2014. Svrha rada je obrada GPS zapisa kako bi se izračunali prometni parametri koji opisuju prometni tok na raskrižju i izrada algoritma strojnog učenja koji će grupirati raskrižja te identificirati raskrižja s niskom razinom učinkovitosti raskrižja.

Ciljevi rada su slijedeći:

- definiranje prometnih parametara kojima se opisuje prometni tok na raskrižjima,
- prikazati postojeće metode izračuna razine uslužnosti raskrižja,
- definirati vektor vrijednosti koji opisuje prometni tok na raskrižju,
- grupirati raskrižja na temelju karakterističnog vektora vrijednosti,
- izraditi programsko rješenje koje će omogućiti ispunjenje navedenih ciljeva.

1.2 Dosadašnja istraživanja

Većina istraživanja u ovom području uspoređuje se s HCM metodologijom koja opisuje načine računanja i procjene parametara prometnog toka, [1]. Autori u [2] prikazuju metodu za definiranja klasa za procjenu razine uslužnosti urbanih prometnica drugačijih u odnosu na HCM prijedlog. Prikazuju način prikupljanja i obrade GPS podataka kako bi procijenili klase za određivanje razine uslužnosti u Indijskim gradovima. U radu [3] autori prikazuju metodu prikupljanja i obrade podataka u svrhu grupiranja vrijednosti brzina kako bi izračunali vrijednosti za određivanje razine uslužnosti urbanih prometnica. Svaka grupa predstavlja vrijednosti brzina koje karakteriziraju razinu uslužnosti koja se opisuje grupom. U radu [4] koriste se GPS podaci za izračun brzine i vremena čekanja. Autori ističu primjenu sustava koji koriste podatke s GPS praćenih vozila u ITS okruženju (*engl. Intelligent Transport Systems* - ITS). Radovi [5], [6] i [7] razrađuju tematiku procjene duljine repa čekanja na raskrižju u stvarnom vremenu korištenjem praćenih mobilnih senzora (postavljenih u vozila) i procesiranjem video materijala.

1.3 Struktura rada

U prvom dijelu, UVODU, definiran je problem i predmet istraživanja, navedeni su ciljevi i svrha istraživanja. Dan je osvrt na dosadašnja istraživanju u ovom području te je obrazložena struktura diplomskog rada.

U drugom dijelu pod naslovom URBANE PROMETNICE, prikazani su osnovni prometni parametri poput brzine, gustoće i protoka. Prikazani su fundamentalni dijagrami koji opisuju povezanost osnovnih prometnih parametara. Također, opisuju se poznate metode prikupljanja podataka sa urbanih prometnica.

U trećem dijelu pod naslovom OBRADA PODATAKA, prikazani su ulazni podaci. Opisane su metode podjele podataka s obzirom na mjesece i dane u tjednu. Također, istaknuta je sezonalnost kao karakteristika prometnog toka u Republici Hrvatskoj. Opisane su NoSQL baze podataka te su prikazane razlike u odnosu na relacijske baze podataka. Opisana je baza podataka MongoDB koja se koristi za realizaciju programskog rješenja.

U četvrtom dijelu pod naslovom RAZINE USLUŽNOSTI I REPOVI ČEKANJA, opisane su poznate metode koje se koriste za izračun razine uslužnosti i repova čekanja te je prikazano kako su ti parametri izračunati korištenjem GPS zapisa.

U petom dijelu, PROGRAMSKO RJEŠENJE opisano je rješenje izrađeno korištenjem programskog jezika Python i baze podataka MongoDB. Opisane su metode strojnog učenja koje su korištene i implementirane u programskom rješenju.

Šesti dio rada prikazuje dobivene rezultate na analiziranim i obrađenim raskrižjima.

U posljednjem dijelu rada, ZAKLJUČAK dan je osvrt na rad i komentar na rezultate istraživanja. Prikazani su nedostaci istraživanja i potencijalne nadogradnje te budući razvoj.

Poglavlje 2

Urbane prometnice

Urbane prometnice jedinstveni su primjer prometnica u odnosu na ostale tipove prometnica kao što su autoceste, brze ceste i prometnice u ruralnim sredinama. Urbana prometnica često poslužuje simultano više modova prometa koji su u međusobnoj interakciji. Modovi prometa uključuju cestovni, pješački, biciklistički, javni gradski prijevoz i tranzitni promet. Poboljšanje učinkovitosti jednog prometnog moda na urbanoj prometnici ili raskrižju može negativno utjecati na neki drugi prometni mod. Primjer takvog scenarija je dodjela prioriteta na raskrižjima vozilima javnog gradskog prijevoza (JGP). Iako su vozilima JGP-a poboljšane performanse i smanjeno vrijeme putovanja, ostalim cestovnim vozilima može se povećati vrijeme putovanja i smanjiti razina usluge koju pruža urbana prometnica, [1].

2.1 Prometni parametri

Osnovni prometni parametri su pokazatelji koji prema teoriji prometnih tokova opisuju prometni tok, zakonitosti kretanja na prometnici i interakcije između vozila. Prometni tok se definira kao kretanje više entiteta (vozila) po prometnoj infrastrukturi (cesti) prema određenim zakonitostima, [8].

Glavni prometni parametri i njihove oznake su:

- protok vozila - q [voz/s],
- gustoća prometnog toka - g [voz/km],
- brzina prometnog toka - v [m/s],
- vremenski interval slijeđenja u prometnom toku - t [s],
- razmak slijeđenja vozila u prometnom toku - s [m].

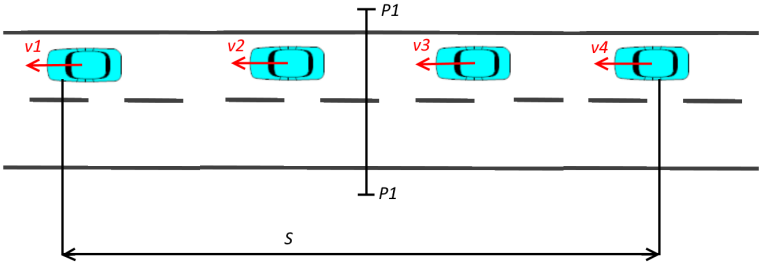
2.1.1 Brzina

Brzina prometnog toka prikazuje srednju brzinu svih vozila koja prometuju po promatranoj infrastrukturi. Ovisno o načinu mjerenja razlikuju se: (i) srednja vremenska brzina prometnog toka i (ii) srednja prostorna brzina prometnog toka.

Srednja vremenska brzina

Srednja vremenska brzina prostorno je vezana za mjesto mjerenja (poprečni presjek prometnice), a vremenski za period promatranja. Predstavlja aritmetičku sredinu svih izmjerenih brzina u prometnom toku vozila kojih prolaze kroz promatrani presjek na prometnici u definiranom vremenskom periodu. Slika 2.1 prikazuje način izračuna srednje vremenske brzine $\overline{v}_t$, [9]. Srednja vremenska brzina definirana je izrazom (2.1):

$$\overline{v}_t = \frac{1}{N} \sum_{i=1}^{N} v_i \tag{2.1}$$

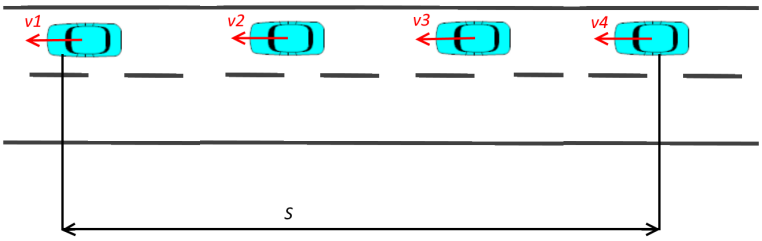


Slika 2.1: Izračun srednje vremenske brzine

Srednja prostorna brzina

Srednja prostorna brzina prostorno je vezana za odsjek puta (dionica prometnice), a vremenski za trenutak u vremenu. Predstavlja aritmetičku sredinu brzina svih vozila na određenom odsjeku puta (prometnice) u određenom vremenskom trenutku. Slika 2.2 prikazuje način izračuna srednje prostorne brzine $\overline{v}_s$. U literaturi se također naziva i srednjom trenutnom brzinom, [9]. Srednja prostorna brzina definirana je izrazom (2.2):

$$v_s = \frac{N}{\sum_{i=1}^{N} \frac{1}{v_i}} \tag{2.2}$$



Slika 2.2: Izračun srednje prostorne brzine

Podjela brzina s obzirom na stanje na prometnici

U ovisnosti o uvjetima na prometnici brzine $\overline{v}_t$ i $\overline{v}_s$ prema teoriji prometnog toka poprimaju sljedeće nazive, [9]:

– **brzina slobodnog toka** (*engl. Free Flow Speed* - FFS) - opisuje brzinu kretanja vozila u idealnim uvjetima bez interakcije s drugim vozilima. Najčešće predstavlja maksimalnu brzinu kretanja po prometnici,

– **brzina normalnog toka** - dijeli se na brzinu stabilnog, polustabilnog i nestabilnog prometnog toka. Ovisi o trenutnim uvjetima na prometnici,

– **brzina zasićenog toka** - brzina koju vozila ostvaruju kada je prometni tok pri maksimalnom kapacitetu prometnice. Opisuje prometovanje u uvjetima kada vozila interaktivno utječu na brzinu vozila u svojoj okolini. U takvim uvjetima vozila postižu približno jednaku brzinu zasićenog toka $\overline{v}_{zt}$. Stoga vrijedi: $\overline{v}_{zt} \cong \overline{v}_t \cong \overline{v}_s$,

– **brzina forsiranog toka** - opisuje brzinu vozila prilikom stani-kreni vožnje. Brzina svih vozila je približno jednaka i oscilira u intervalu $\langle 0, \overline{v}_{zt}]$.

2.1.2 Protok

Protok vozila se izražava kao broj vozila koji prođe određenim presjekom prometnice u jedinici vremena. Posebno se računa svaki smjer prometovanja, [9], [8]. Razlikuju se dvije vrste mjerenja prometnog toka s obzirom na prostor: (i) protok vozila na presjeku i (ii) protok vozila na dionici prometnice.

Protok vozila na presjeku

Opisuje protok vozila s obzirom na presjek prometnice u jedinici vremena i prikazuje se izrazom (2.3):

$$q = gV \tag{2.3}$$

gdje je:

– q [voz/h] - protok vozila,

– g [voz/km] - gustoća vozila,

– V [km/h] - brzina vozila.

Protok vozila na dionici prometnice

Predstavlja aritmetičku sredinu svih zabilježenih protoka na promatranom dijelu prometnice.

Navedene relacije za izračun protoka odnose se na dijelu prometnice u jednom smjeru. Osnovna jedinica za protok je broj vozila po satu (voz/h), ali se po potrebi protok može izražavati drugim jedinicama voz/dan ili voz/s.

2.1.3 Gustoća

Gustoća prometnog toka izražava se kao broj vozila koji se nalazi na promatranom dijelu prometnice. Posebno se izražava za jedan smjer, po prometnoj traci, odnosno za oba mjera za dvosmjerne prometnice, [8].

S obzirom na vremenski period promatranja, gustoća se može izraziti kao:

1. broj vozila u na prometnici u trenutku promatranja koja se računa prema izrazu (2.4):

$$g = \frac{N}{s} \tag{2.4}$$

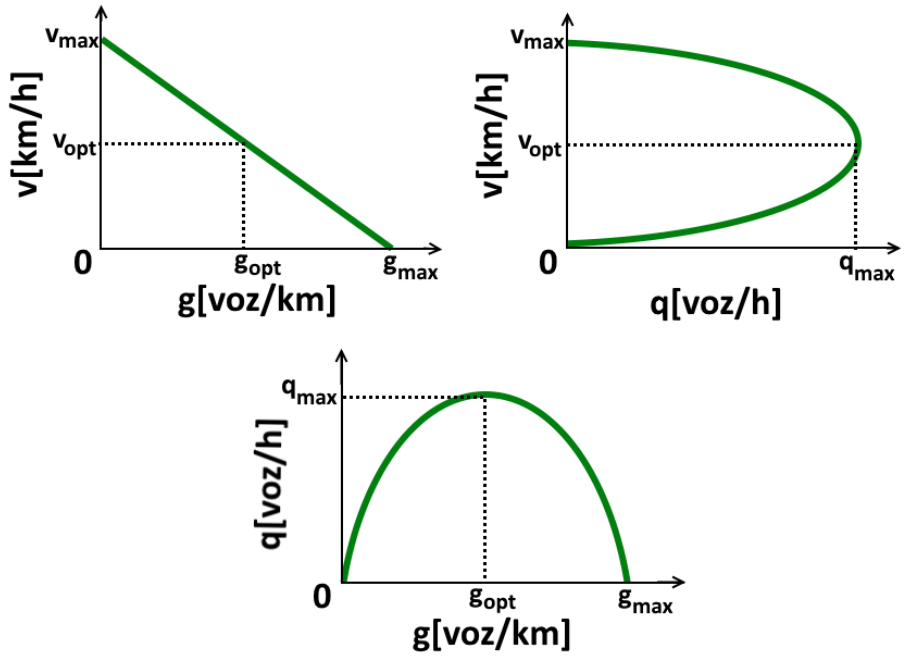
   gdje je:

   - g [voz/km] - gustoća vozila,
   - N [voz] - broj vozila koji se nalazi na prometnici u trenutku promatranja,
   - s [km] - duljina dionice prometnice koja se promatra.

2. Aritmetička sredina više trenutnih mjerenja gustoće na promatranoj prometnici

2.1.4 Odnosi između prometnih parametara

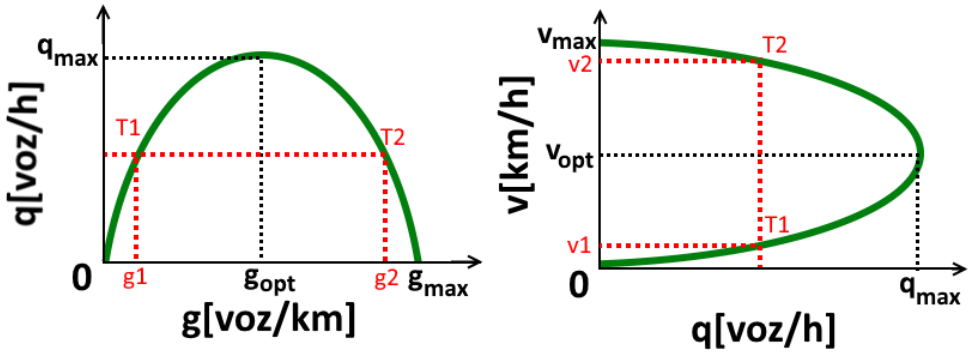
Osnovne relacije između osnovnih prometnih parametara prikazuje se takozvanim *fundamentalnim dijagramima prometnog toka*. Prikazuju odnos protoka q, gustoće g i brzine v u idealnim uvjetima jednosmjernog homogenog prometnog toka, [8].



Slika 2.3: Fundamentalni dijagrami prometnog toka

Slika 2.3 prikazuje ovisnost triju osnovnih prometnih parametara. Protok q raste od ishodišta sve do točke g_{opt} u kojoj poprima maksimalnu vrijednost. Maksimalna vrijednost protoka q_{max} prikazuje projektirani maksimalni kapacitet promatrane prometnice. U toj točki postiže se optimalna gustoća g_{opt} pri kojoj je iskorištenje prometnice maksimalno. Nakon točke g_{opt} gustoća nastavlja rasti, a vrijednost protoka pada. Vrijednost protoka pada zbog sve većeg broja vozila na prometnici i nastaju zagušenja. Protok pada sve do točke g_{max} u kojoj poprima vrijednost 0 i na prometnici nastaje potpuni zastoj.

Iz fundamentalnih dijagrama također se može procijeniti pouzdanost vrijednosti parametara koji se koriste za upravljanje prometnim tokovima. Slika 2.4 prikazuje kako ista vrijednost protoka može prikazivati dvije vrlo različite situacije. Protok je jednak u točkama T_1 i T_2, no gustoća g_1 i g_2 te brzina v_1 i v_2 nisu jednake. Naime gustoća g_2 i brzina v_2 ukazuju na prometno zagušenje. Iz navedenoga proizlazi kako su brzina i gustoća bolji parametri prilikom procjene trenutnog stanja na prometnici.



Slika 2.4: Različita stanja prometnog sustava za iste vrijednosti protoka

2.2 Prikupljanje podataka

Prikupljanje podataka vrlo je važna komponenta pri izradi sustava za upravljanje prometnim tokovima. Podaci se najčešće prikupljaju u centraliziranom prometnom centru korištenjem strateški smještenih senzora na prometnicama. Za brojanje prometa i procjenu brzine vozila najčešće se koriste senzori poput induktivnih petlji, prometnih kamera, radara, ultrazvučnih i magnetskih senzora. Svaki od navedenih senzora statično (fiksno) je smješten na prometnoj infrastrukturi i bežičnom mrežom šalje podatke u prometni centar. Prednost ovakvih senzora je u tome što detektiraju sva vozila koja se kreću u njihovoj blizini, a mane se iskazuju u statičnoj poziciji odnosno pomicanje senzora na drugu lokaciju predstavlja određeni vremenski i financijski trošak. Osim fiksnih senzora postoje i tehnike prikupljanja podataka orijentirane na praćenje vozila. Neke od metoda su: (i) testna vozila, (ii) detekcija registarskih tablica i (iii) testno vozilo u ITS okruženju.

Navedenim tehnikama najčešće se prikupljaju podaci o trenutnoj brzini vozila, poziciji vozila i vremenskom trenutku uzorka. Iz prikupljenih podataka potrebno je ekstrahirati korisne informacije kao što su vrijednosti prometnog toka, gustoće, repa čekanja na raskrižju ili razina uslužnosti prometnice.

2.2.1 Testna vozila

Prikupljanje podataka pomoću testnih vozila najčešće se vrši pomoću korištenja GPS prijemnika. GPS prijemnik prikuplja podatke o trenutnoj geografskoj lokaciji vozila te uz svaki zapis pridružuje vremenski trenutak uzorka signala. Uz lokaciju i vrijeme GPS prijemnik često računa i trenutnu brzinu vozila, [10].

Prednosti:

- GPS prijemnik daje veliku količinu podataka,
- niska cijena (u odnosu na ugradnju senzora na prometnicu).

Nedostaci:

- mogućnost prostorne pogreške zbog slabe vidljivosti satelita ili nepreciznih senzora,
- poteškoće pri spremanju podataka.

2.2.2 Detekcija registarskih tablica

Za detekciju registarskih tablica koriste se prometne kamere koje su opremljene algoritmima za obradu slika. Svrha implementacije ovakvog sustava je praćenje vozila kroz prometnu mrežu i mjerenje vremena protekloga od jedne točke u mreži do druge, [10].

Prednosti:

- mogućnost identifikacije vozila.

Nedostaci:

- fiksna pozicija kamere,
- točnost algoritama za procesiranje slika,
- poteškoće pri spremanju podataka.

2.2.3 Ispitno vozilo u ITS okruženju

Ispitna vozila u ITS okruženju opremljena su posebnom opremom za prikupljanje podataka koja komunicira s ITS infrastrukturom ugrađenom na prometnice. Neke od tehnologija koje se primjenjuju u ovom području su odašiljači na prometnoj infrastrukturi, automatska identifikacija vozila (*engl. Automatic Vehicle Identification* - AVI), korištenje radiovalova za komunikaciju, mobilne ćelijske mreže i GPS prijemnici. Vozila mogu biti opremljena i s dinamičnim rutnim vodičima te prikupljati podatke bez uređaja na prometnoj infrastrukturi. Takva vozila nazivaju se testna vozila, [10]. Testna vozila podatke šalju u prometni centar pomoću komunikacijske opreme koja je najčešće mobilna mreža. Najčešći način prikupljanja podataka je ugradnja komunikacijske opreme u prometnu infrastrukturu (prometne znakove ili posebne spremnike pored prometnice). Kada vozilo opremljeno za prikupljanje podataka prilazi mjestu prikupljanja podataka, vozilo se identificira i predaje podatke u definiranom formatu. Podaci se tada šalju u prometni centar na daljnju obradu, [11].

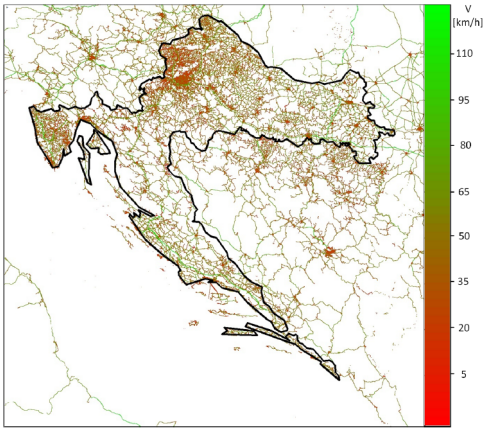
Poglavlje 3

Obrada podataka

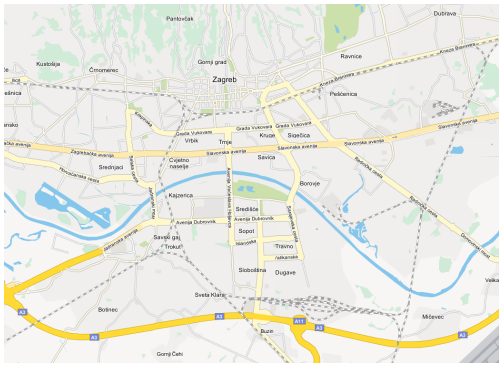
Prikupljanje podataka s prometne mreže osnovna je pretpostavka za istraživanje i izučavanje prometnih procesa. Dvije osnovne značajke prikupljenih podataka su rezolucija uzorkovanja i pouzdanost podataka. U ovom radu korišteni su podaci prikupljeni kroz projekt SORDITO [1]. Prikupljeno je približno 7 milijardi GPS zapisa vozila diljem Republike Hrvatske u razdoblju od pet godina od kolovoza 2009. do listopada 2014. GPS zapise prikupljalo je približno 4200 vozila. Zapisi sadržavaju geografske koordinate, smjer kretanja i brzinu vozila. Podaci su prikupljeni korištenjem navigacijskih uređaja ugrađenih u praćena vozila, te u konačnici čine povijesne podatke prikupljene tehnikom pokretnih osjetila (*engl. Floating car data* - FCD) korištenjem navigacijskog uređaja u vozilima. Rezolucija prikupljenih podataka je približno svakih 100 m kada se vozilo kreće i približno svakih 5 min kada vozilo stoji. Slika 3.1a prikazuje prikupljene podatke na teritoriju Republike Hrvatske. Crvenom bojom označene su prometnice na kojima je izmjerena brzina manja ili jednaka od 5 km/h, a zelenom bojom brzine manje od 100 km/h, [12], [13].

Podaci su raspoređeni po digitalnoj karti Republike Hrvatske ustupljenoj od tvrtke Mireo d.d. Na digitalnoj karti prometnice su prikazane linkovima (njih 450 000), u prosjeku duljine 176 m. Jedan link je dio prometnice omeđen s dva raskrižja. Svaki link definiran je s dvije geografske koordinate (početna i završna točka), jedinstvenim identifikacijskim brojem (ID), ograničenjem brzine, duljinom i kategorijom. ID je različit za svaki smjer ako se radi o dvosmjernoj prometnici, [12], [14]. Slike 3.1b i 3.1c prikazuje korištenu digitalnu kartu i linkove.

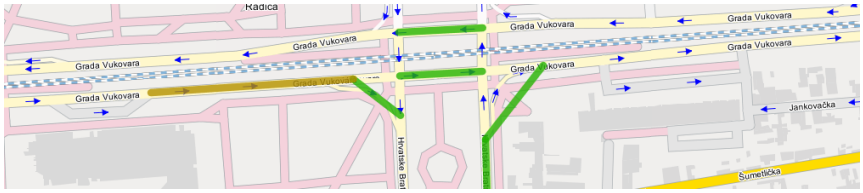
[1] Projekt SORDITO - Sustav za optimizaciju ruta u dinamičkom transportnom okruženju RC.2.2.08-0022, koji je financiran sredstvima Europske unije iz Europskog fonda za regionalni razvoj



(a) Prikupljeni GPS zapisi



(b) Digitalna karta



(c) Linkovi na digitalnoj karti

Slika 3.1: Prikaz prikupljenih podataka na teritoriju Republike Hrvatske

3.1 Struktura ulaznih podataka

Prikupljeni GPS podaci obrađeni su i strukturirani u CSV (*engl. Coma Separated Values* - CSV) formatu zbog daljnje obrade unutar aplikacije. Svako praćeno vozilo u CSV datoteci reprezintirano je kao nova ruta i označeno oznakom *"NEW_ROUTE"*. Nakon oznake slijede svi zapisi prikupljeni za razmatranu rutu. Slika 3.2 i tablica 3.1 prikazuju podatke u CSV strukturi i opis svakog atributa. Svaka ruta sadrži više GPS zapisa definiranih sa slijedećim atributima:

LINK;MATCH_X;MATCH_Y;ORG_X;ORG_Y;MATCH_D;MATCH_H;ORG_H;ORG_S

```
NEW_ROUTE
-214697;1352969386;15.952941030263901;45.779258712251504;15.952963829040527;45.779261518209182;104;352;77;353;77
-214696;1352969391;15.952770709991455;45.780174382278325;15.952802896499634;45.780177188189924;102;354;75;354;75
-214696;1352969397;15.952580273151398;45.781350046872596;15.952609777450562;45.781352852725036;131;352;80;353;80
-214695;1352969400;15.952487736940384;45.781942078607585;15.952498465776443;45.781943013881801;66;352;78;353;78
-214695;1352969405;15.952364355325699;45.782888568100674;15.952332168817520;45.782886697583947;105;354;76;354;76
-214695;1352969410;15.952239632606506;45.783845329191777;15.952204763889313;45.783843458707139;106;354;77;355;77
-214695;1352969415;15.952124297618866;45.784728190939994;15.952126979827881;45.784729126167470;98;358;64;358;64
-214694;1352969421;15.952069312334061;45.785613844344681;15.952090770006180;45.785614779557328;98;358;59;358;59
-214717;1352969428;15.952071994543076;45.786376972634919;15.952041149139404;45.786370426235671;84;350;33;350;33
NEW_ROUTE
-214697;1353330220;15.952922254800797;45.779357856003834;15.952931642532349;45.779359726638972;104;352;70;353;70
-214697;1353330220;15.952922254800797;45.779357856003834;15.952931642532349;45.779359726638972;0;352;70;353;70
-214697;1353330220;15.952922254800797;45.779357856003834;15.952931642532349;45.779359726638972;0;352;70;353;70
-214696;1353330226;15.952737182378769;45.780382019355244;15.952742546796799;45.780382954655636;114;352;68;353;68
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;77;352;71;352;71
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;0;352;71;352;71
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;0;352;71;352;71
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;0;352;71;352;71
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;0;352;71;352;71
-214696;1353330230;15.952624529600143;45.781078813804136;15.952612459659576;45.781078813804136;0;352;71;352;71
```

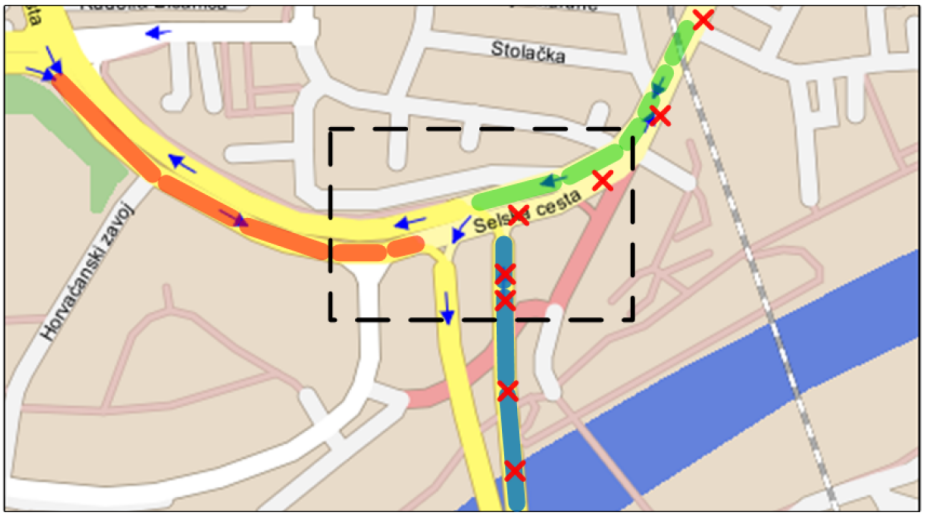
Slika 3.2: CSV format podataka

Tablica 3.1: Opis atributa

Atribut	Primjer	Opis
LINK	-214696	ID pridruženog linka i digitalnoj karti
UTC	1350032645	UTC vrijeme u sekundama (od 1.1.1970)
MATCH_X	15.952734500169754	Pridružena geografska dužina [°]
MATCH_Y	45.780397919459887	Pridružena geografska širina [°]
ORG_X	15.952774733304977	Originalna geografska dužina [°]
ORG_Y	45.7804016606603244	Originalna geografska širina [°]
MATCH_D	110	Pridružena prijeđena cestovna udaljenost od prethodne zabilježene točke [m]
MATCH_H	348	Pridruženi geografski smjer kretanja u odnosu na sjever [°]
MATCH_S	50	Pridružena brzina kretanja [km/h]
ORG_H	348	Originalni geografski smjer kretanja u odnosu na sjever [°] ($360° = 0°$)
ORG_S	50	Originalna brzina kretanja [km/h]

3.2 Predobrada podataka

Za izračun performansi raskrižja izdvojeni su linkovi za svaki prilaz raskrižju. Za svaki link agregirani su svi dostupni GPS zapisi praćenih vozila. Slika 3.3 prikazuje jedno analizirano raskrižje između Jadranskog mosta, Selske ceste i Savske ceste označeno s crnom iscrtanom linijom. Raskrižje se sastoji od tri geografski odvojena prilaza: (i) Sjeverni (S) - Savska cesta (zelena boja); (ii) Zapadni (Z) - Selska cesta (narančasta boja); (iii) Južni (J) - Jadranski most (plava boja). Svaki prilaz sastoji se od više linkova na digitalnoj karti. Prilaz je definiran sa svim linkovima koji se nalaze između semafora na prilazu i prvog prethodnog raskrižja. Na taj način prate se vozila između dva raskrižja. Jedna ruta vozila koje prolazi kroz raskrižje prikazana uzorcima GPS signala označenih crvenim znakom x.

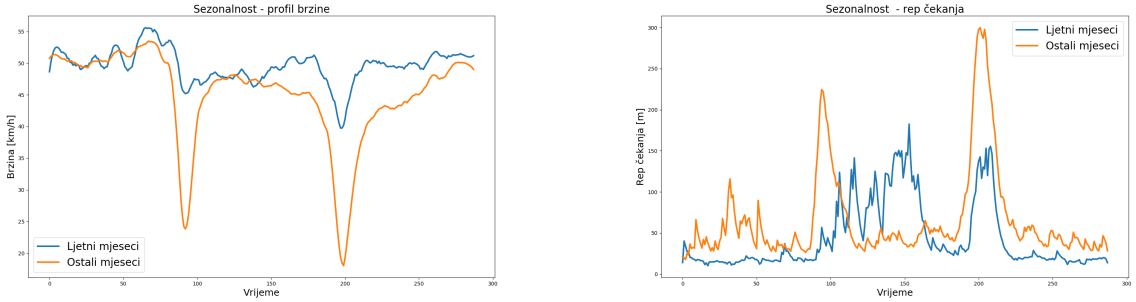


Slika 3.3: Promatrano raskrižje opisano linkovima i rutama

3.2.1 Sezonalnost

U Republici Hrvatskoj sezonalnost značajno utječe na prometne tokove. Tijekom ljetnih mjeseci zbog velikog priljeva turista i odlaska mnogo ljudi na odmor, može se zamijetiti značajna razlika u prometnom toku između ljetnih mjeseci i ostalih mjeseci u godini. U ovom radu za ljetne mjesece odabrani su srpanj i kolovoz u kojima su primijećene najznačajnije promjene. Slika 3.4a prikazuje profile brzine i slika 3.4b profile repova čekanja (profili brzine i repova čekanja opisani su u poglavljima 4.3 i 4.4) koji prikazuju razliku između pada brzine i duljine repova čekanja u ljetnim mjesecima i ostalim mjesecima na lokaciji Jadranskog mosta u gradu Zagrebu.

Vrijednosti brzine zimskog profila manje su u odnosu na vrijednosti za ljetni profil brzine što dovodi do zaključka da je brzina manja zbog većeg broja vozila na prometnici. Tvrdnju potkrepljuje i prikaz profila repa čekanja na slici 3.4b. Zbog povećanog broja vozila na prometnici rep čekanja, pogotovo u vršnim satima raste.



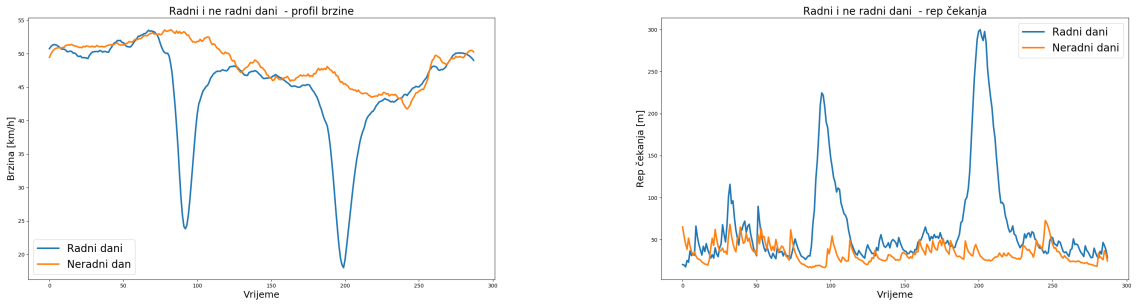
(a) Profili brzina

(b) Profili repova čekanja

Slika 3.4: Sezonalnost prometnog toka - profili brzina i repova čekanja

3.2.2 Podjela podataka

Daljnja podjela podataka odnosi se na podjelu po danima u tjednu. Primijećena je značajna razlika između različitih dana u tjednu pogotovo promatrajući radne dane (ponedjeljak do petak) u odnosu na neradne dane (subota i nedjelja). Slika 3.5a prikazuje različite profile brzina i repova čekanja (3.5b) za radne i neradne dane. Primjećuje se značajna razlika, stoga se u ovom radu odvojeno analiziraju radni i neradni dani.



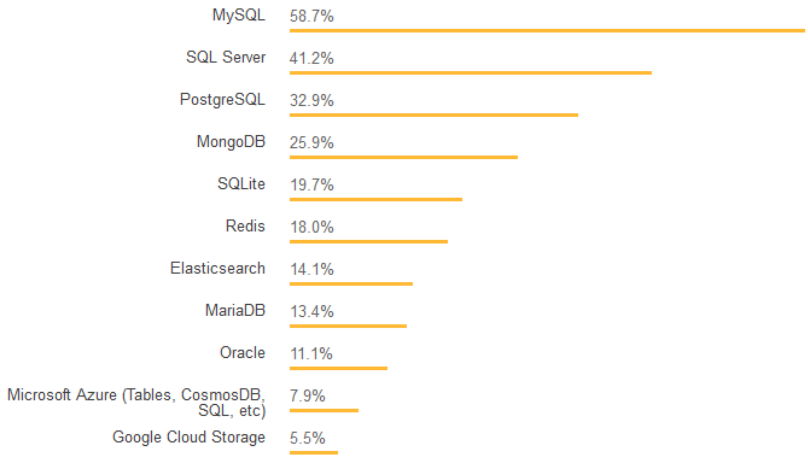
(a) Profili brzina

(b) Profili repova čekanja

Slika 3.5: Vrijednosti profila brzina za različite dane u tjednu

3.3 NoSQL baze podataka

Za pohranu podataka u ovom radu odabrana je NoSQL baza podataka. NoSQL baze pohranjuju i prezentiraju podatke koji nisu striktno podijeljeni u tablice i ograničeni relacijama kao u relacijskim bazama podataka. Korištenje takvih baza podataka intenzivno počinje s pojavom velikih kompanija koje svoje poslovanje temelje na prikupljanju velikih količina raznovrsnih podataka kao Google, Facebook ili Amazon. Struktura i vrsta prikupljenih podataka brzo se mijenja što ne odgovara relacijskim bazama podataka. Relacijske baze podataka nisu dizajnirane na način da mogu podržati velike količine podataka koji mijenjaju strukturu u relativno malom vremenu što je karakteristika mnogih modernih aplikacija.



Slika 3.6: Rezultati istraživanja o korištenju baza podataka, [15]

Jedna od važnih karakteristika koje omogućavaju $NoSQL$ bazama podataka prihvaćanje velikog broja ne strukturiranih podataka je dinamička shema. Kod relacijskih baza podataka shema mora biti unaprijed definirana, a tek onda se spremaju podaci. U modernom razvoju softvera često korišteni razvojni pristup je Agilni razvoj softvera. Prilikom razvoja softvera na takav način događa se više malih dnevnih promjena ili većih tjednih promjena. Takav razvoj zahtjeva bazu podataka koja koristi dinamičku shemu zbog velikog broja promjena. Relacijske baze podataka takve promjene ne mogu podržati bez promjene sheme. Takav postupak iziskuje mnogo vremena pogotovo kod velikih baza podataka. Istraživanja [15], [16] pokazuju kako su SQL baze podataka još uvijek prvi izbor prilikom odabira baze podataka, no zbog svojih prednosti NoSQL baze sve više dobivaju na značaju. Rezultati istraživanja provedenog od strane StackOverflow zajednice prikazano je slikom 3.6.

3.3.1 Vrste NoSQL baza podataka

NoSQL ne predstavlja jednu tehnologiju ili proizvod stoga se razvilo više vrsta baza podataka svaka s prednostima i manama. Četiri osnovne vrste u koje se dijele NoSQL baze podataka su, [17]:

- ***Document database*** - spremanje podataka u obliku dokumenta koji prati određenu strukturu, najčešće JSON (poglavlje 3.4) strukturu. Unutar svakog dokumenta nalaze se *key-value* parovi koji sadrže podatke.
- ***Spremanje pomoću grafa*** - način spremanja koji omogućava pohranu informacija najčešće o mrežama korisnika te njihovim vezama i atributima.
- ***Key-value spremanje*** - najjednostavnija struktura koja se koristi kod spremanja podataka. Svaki podatak definiran je dvjema vrijednostima: *key* koji je identifikacijski ključ podatka i *value* koja predstavlja vrijednost.
- ***Wide-column spremanje*** - baze podataka optimirane za rad s upitima na velikim količinama podataka. Umjesto redaka koriste se stupci s podacima koji se zajedno spremaju.

3.3.2 Razlike u odnosu na SQL baze podataka

Relacijske baze podataka razlikuju se u odnosu na NoSQL baze podataka. Tablica 3.2 prikazuje obje vrste baza podataka i naglašava razlike među njima. Glavna prednost relacijskih baza podataka je ACID (*engl. Atomicity, Consistency, Isolation, Durability*) podrška. U računalnoj znanosti termin ACID predstavlja skup svojstava koja jamče pouzdanost transakcije baze podataka [18].

Tablica 3.2: Usporedba relacijski i NoSQL baza podataka, [19]

	Relacijske baze	NoSQL
Tipovi	SQL baza podataka.	Više vrsta: key-value, dokument database, wide-column store i graph store.
Povijesni razvoj	Razvijene 1970-ih kao podrška aplikacijama.	Razvijene u kasnim 2000-ima kao odgovor na ograničenja SQL baza podataka posebice vezano za skalabilnost, različite strukture podataka i agilni način razvoja softvera.
Primjeri	MySQL, PostgresSQL, Microsoft SQL, Server, Oracle database	MongoDB, Redis, Cassandra, HBase, Neo4j
Način spremanja podataka	Zasebni zapisi koji se spremaju kao redovi u tablice gdje stupci izričito definiraju tip podatka. Podaci u različitim tablicama kombiniraju se pomoću strukturiranih upita korištenjem upitnog jezika (SQL). Tablice se spajaju pomoću primarnih i stranih ključeva.	Ovisi o odabranom tipu baze podataka.
Sheme	Struktura i tipovi podataka unaprijed su određeni. Novi podaci moraju pratiti unaprijed zadanu strukturu.	U većini slučajeva implementirana je dinamička shema. Podaci u mogu biti različito strukturirani.
Skaliranje (povećavanje obujma)	Obično baza radi na jednom serveru. Distribuiranje relacijske baze podataka kompleksan je proces.	Prilagođene za distribuirane i *cloud* sustave.
Licence	Kombinacija *open-source* licenci i licenci koje se plaćaju.	*Open-source* licence
Manipulacija podacima	Korištenjem SQL upitnog jezika.	Korištenjem objektno orjentiranih programskih biblioteka.
ACID podrška	Da.	Većinom ne.

3.4 MongoDB

MongoDB je NoSQL baza podataka koja za pohranu podataka koristi dokumente (*engl. Document database*). MongoDB sprema podatke koristeći binarnu reprezentaciju naziva BSON (*engl. Binary JSON*). BSON format podataka predstavlja nadogradnju na klasični JSON (*engl. JavaScript Object Notation*) format. Omogućava pohranu formata *int, long, date, floating point i decimal* koji nisu podržani u JSON formatu. Format sadrži jedno ili više polja koje može primiti jednu vrijednost, listu vrijednosti, podatke u binarnom obliku ili neki drugi dokument. Kôd 3.1 prikazuje jedan dokument u BSON formatu.

```
{
    ime : "Leo",
    prezime : "Tisljaric",
    godine : 24,
    fakultet : "Fakultet prometnih znanosti",
    prosjek_ocjena : 4.232
    studij : {
        naziv : "ITS i logistika",
        godina : 2
    }
}
```

Kôd 3.1: Prikaz dokumenta u BSON formatu

3.4.1 Pohrana podataka

Dokumenti se pohranjuju u takozvane kolekcije (*engl. Collections*). Svaki dokument mora pripadati jednoj kolekciji dok jedna kolekcija sadrži jedan ili više dokumenata. Kolekcije su skupina dokumenata koji imaju neka zajednička svojstva i funkcionalno odgovara tablicama u relacijskim bazama podataka.

Spremanje podataka u ovakvom obliku donosi sljedeće prednosti, [20]:

- Jednostavnije spremanje ne strukturiranih podataka - zbog dinamičke sheme baza podataka ne mora posjedovati unaprijed definiranu shemu.
- Dokument je nezavisna jedinica - omogućava lakšu distribuciju podataka na više servera.
- Jednostavnija logika prilikom programiranja - definirani objekt vrlo je lako pretvoriti u dokument i spremiti u bazu.
- Podrška za Agilni način razvoja softvera - omogućeno je vrlo lako mijenjanje sheme zbog već spomenute dinamične sheme.



Slika 3.7: Grafički prikaz jedne kolekcije u MongoDB bazi podataka

3.4.2 Usporedba s Microsoft SQL Server

Svrha NoSQL baza podataka je unaprijediti i zaobići mane relacijski baza podataka. Mane relacijskih baza javljaju se zbog današnjeg ”novog” Agilnog načina razvoja softvera koji zahtjeva brze promjene čak i na dnevnoj bazi. MongoDB zadržava sličnu strukturu relacijskim bazama kako bi migracija s relacijskih na MongoDB bazu bio što jednostavniji. Tablica 3.3 prikazuje razliku u terminologiji i strukturi dviju baza podataka Microsoft SQL Server i MongoDB.

U nastavku će biti prikazane razlike s obzirom na CRUD operacije. Skraćenica CRUD dolazi od engleskih naziva za osnovne operacije nad bazom podataka *create, read, update* i *delete*.

Tablica 3.3: Analogija između Microsoft SQL Server i MongoDB baze podataka

Microsoft SQL Server	MongoDB
Tablica	Kolekcija
Redak	Dokument
Indeks	Indeks
Funkcija *JOIN*	Ugrađeni dokument ili referenca

Unos novih podataka

Relacijske baze podataka prije unosa podataka zahtijevaju strogo definiranu strukturu podataka koji se unose. Svaki stupac u tablici mora biti definiran s nazivom i tipom podataka uz mogućnost dodavanja dodatnih opcija. Tablica se izrađuje naredbom prikazanom kôdom 3.2. Nakon izrade tablice slijedi unos podataka korištenjem naredbe *INSERT INTO* u kojoj se definiraju stupci i vrijednosti za retke (Kôd 3.3).

```
CREATE TABLE studenti (
    studentID int PRIMARY KEY IDENTITY(1,1),
    ime varchar(100),
    prezime varchar(200),
    godine int,
    IDfakultet int,
    FOREIGN KEY (IDfakultet)
    REFERENCES fakulteti(fakultetID)
);
```

Kôd 3.2: Izrada tablice u relacijskoj bazi podataka

```
INSERT INTO studenti (ime, prezime, godine)
VALUES ("Leo","Tisljaric",24)
```

Kôd 3.3: Unos podataka u tablicu korištenjem SQL jezika

MongoDB kao NoSQL baza podataka podržava unos nestrukturiranih podataka. Što znači da korisnik baze ne mora unaprijed definirati strukturu podataka koji se unose. Ako kolekcija u koji se podaci unose ne postoji, automatski je kreirana i dodaje se dokument koji se unosi u bazu. Za unos u bazu koristi je jedna od naredbi *db.collection.insertOne()* za unos jednog dokumenta ili *db.collection.insertMany()* za unos više dokumenata (Kôd 3.4).

```
db.studenti.insertOne(
    {
        ime : "Leo",
        prezime : "Tisljaric",
        godine : 24
    }
)
```

Kôd 3.4: Unos dokumenta u kolekciju u MongoDB baze podataka

Čitanje iz baze podataka

Upit se sastoji od *SELECT* naredbe s definiranim stupcima, tablice iz koje se dohvaćaju podaci, uvjeta pretrage i modifikatora pretrage. Jednostavni SQL upit prikazan je kôdom 3.5. U prikazanom slučaju, rezultat SQL upita biti će vrijednosti svih redaka za tri stupca: *id, ime, prezime* iz tablice *student* za koje vrijedi da je vrijednost stupca *prosjek* veća od 4.7. Naredba *LIMIT* ograničiti će prikaz rezultata na 100 vrijednosti.

```
SELECT  id, ime, prezime
FROM    studenti
WHERE   prosjek > 4.7
LIMIT   100
```

Kôd 3.5: SQL upit

MongoDB upit se sastoji od imenovane kolekcije, funkcije *find()*, uvjeta pretrage u JSON formatu i modifikatora pretrage (naredba *limit*). MongoDB upit prikazan je kôdom 3.6, prikazuje istu funkcionalnost kao u SQL primjeru.

```
db.studenti.find{
    {prosjek : {&gt : 18}},
    {ime : 1, prezime : 1}
}.limit(100)
```

Kôd 3.6: MongoDB upit

Ažuriranje podataka

Za ažuriranje podataka u relacijskim bazama podataka koristi se naredba *UPDATE* prikazana kôdom 3.7. Naredbom se vrši ažuriranje svih zapisa koji odgovaraju definiranom uvjetu.

```
UPDATE studenti
SET godine = 25
WHERE ime = "Leo" AND prezime = "Tisljaric"
```

Kôd 3.7: Ažuriranje podataka u relacijskoj bazi podataka

MongoDB za ažuriranje koristi jednu od tri naredbi *db.collection.updateOne()* za ažuriranje jednog dokumenta, *db.collection.updateMany()* za ažuriranje svih dokumenata koji udovoljavaju uvjetu ili *db.collection.replaceOne()* za zamjenu jednog dokumenta s novim dokumentom. Kôd 3.8 prikazuje korištenje naredbe za ažuriranje jednog dokumenta.

```
db.studenti.updateOne(
    {ime : "Leo", prezime : "Tisljaric"},
    {$set : {godine : 25}}
)
```

Kôd 3.8: Ažuriranje podataka u MongoDB bazi podataka

Brisanje podataka

Za brisanje podataka relacijske baze koriste naredbu *DELETE*. Naredba će izbrisati sve zapise koji zadovoljavaju određeni uvjet. Sintaksa naredbe prikazana je kôdom 3.9.

```
DELETE FROM studenti
WHERE ime = "Leo"
```

Kôd 3.9: Brisanje zapisa korištenjem SQL jezika

MongoDB za brisanje podataka pruža na raspolaganje dvije naredbe *db.collection.deleteOne()* za brisanje jednog dokumenta iz baze i *db.collection.deleteMany()* za brisanje svih dokumenata koji odgovaraju definiranom uvjetu. Kôd 3.10 prikazuje korištenje naredbe za brisanje jednog dokumenta. Ako postoji više dokumenata koji odgovaraju definiranom uvjetu biti će izbrisan samo jedan i to prvi koji se pojavljuje unutar definirane kolekcije.

```
db.studenti.deleteOne(
    {ime : "Leo"}
)
```

Kôd 3.10: Brisanje zapisa u MongoDB bazi podataka

Poglavlje 4

Razine uslužnosti i repovi čekanja

Razina uslužnosti je kvalitativna mjera koja se koristi za određivanje performansi prometnog infrastrukturnog objekta koji se razmatra, a može biti prometnica ili raskrižje. Određuje se za automobilski, pješački i biciklistički promet. Procjena vrijednosti razine uslužnosti vrši se na temelju neke od mjera učinkovitosti promatranog infrastrukturnog objekta: duljina repa čekanja, vrijeme putovanja, brzina vozila, vremensko kašnjenje na raskrižju i dr. Razina uslužnositi može se procijeniti za cijelo raskrižje, posebno za svaki prilaz raskrižju ili za svaku grupu trakova. Iskazuje se povećanjem vremena putovanja, smanjenjem subjektivne procjene vozača o kvaliteti vožnje i povećanom potrošnjom goriva, [1].

HCM 2010 [1], dijeli razine uslužnosti u šest kategorija. Kategorije su označene slovima A do F gdje je od A najbolja dok je F najgora mjera. Za urbane prometnice vrijedi dodatna podjela na četiri kategorije označene rimskim slovima od I do IV. U nastavku je detaljnije prikazana podjela kategorija za određivanje razine uslužnosti za urbane prometnice i semaforizirana raskrižja.

4.1 Razina uslužnosti urbane prometnice

Razina uslužnosti za urbane prometnice procjenjuje na temelju jednog parametra, prosječna brzina prometovanja na razmatranoj prometnici. Određuje se za prometnicu bez smetnji odnosno za dio prometnice omeđen raskrižjima. Postotni pad brzine u odnosu na slobodni tok (*engl. Free Flow Speed Percentage* - FFSP) i odnos prometnog volumena i kapaciteta prometnice (*engl. Volume-to-capacity ratio* - VCR) glavni su parametri koji služe za procjenu razine uslužnosti. VCR prikazuje razinu do koje se koristi kapacitet prometnice. Ako VCR prijeđe vrijednost 1.0 to znači da je trenutni prometni volumen veći od kapaciteta prometnice i razina uslužnosti se procjenjuje kao najgora odnosno poprima vrijednost F.

Razina uslužnosti A opisuje stanje na raskrižju u uvjetima kada je FFSP veća ili jednaka 85 %, a VCR je manji od 1.0. Podrazumijeva se da se vozila kreću približno brzinom slobodnog toka odnosno maksimalnom brzinom. Vozila u tom slučaju nemaju gotovo nikakvu međusobnu interakciju.

Razina uslužnosti B opisuje stanje na raskrižju u uvjetima kada je FFSP između 67 % i 85 %, a VCR je manji od 1.0. Prikazuje gotovo neometano prometovanje uz vrlo malo čekanje na granicama prometnice (raskrižjima).

Razina uslužnosti C opisuje stanje na raskrižju u uvjetima kada je FFSP između 50 % i 67 %, a VCR je manji od 1.0. Opisuje stabilni prometni tok. Pad brzine uzrokovan je većim čekanjima na raskrižjima i smanjenom mogućnosti manevra.

Razina uslužnosti D opisuje stanje na raskrižju u uvjetima kada je FFSP između 40 % i 50 %, a VCR je manji od 1.0. Opisuje polu-stabilan prometni tok odnosno prometni tok je malo iznad kapaciteta prometnice. Zbog toga su moguća dulja čekanja i smanjena brzina vozila.

Razina uslužnosti E opisuje stanje na raskrižju u uvjetima kada je FFSP između 30 % i 40 %, a VCR je manji od 1.0. Opisuje nestabilni prometni tok najčešće uzrokovan neefikasnim upravljanjem raskrižjem, odnosno potrebno je poboljštati signalni plan raskrižja..

Razina uslužnosti F opisuje stanje na raskrižju u uvjetima kada je FFSP manji od 30 %, a VCR je manji od 1.0. Opisuje prometni tok koji je puno veći od optimalnog kapaciteta prometnice. U većini slučajeva dolazi do stani - kreni vožnje i vrlo malih brzina.

Tablica 4.1: Vrijednosti postotnog pada brzine i razine uslužnosti, [1]

Pad brzine u odnosu na FFS [%]	LOS(VCR ¡= 1.0)	LOS(VCR ¿ 1.0)
> 85	A	F
$\langle 67, 85]$	B	F
$\langle 50, 67]$	C	F
$\langle 40, 50]$	D	F
$\langle 30, 40]$	E	F
≤ 30	F	F

4.2 Razina uslužnosti semaforiziranog raskrižja

Razina uslužnosti koristi se kao kvalitativna mjera za opisivanje performansi raskrižja odnosno efektivnosti implementiranog signalnog plana. Parametri koji služe za procjenu razine uslužnosti su upravljačko vrijeme kašnjenja (*engl. Control Delay* - CD) i odnos prometnog volumena i kapaciteta prometnice. CD iskazuje vrijeme kašnjenja svakog vozila na prilazu raskrižju uzrokovano upravljačkim uređajem ili znakom. VCR prikazuje razinu do koje se koristi kapacitet prilaza raskrižju. Ako VCR prijeđe vrijednost 1.0 to znači da je trenutni prometni volumen veći od kapaciteta prilaza i razina uslužnosti se procjenjuje kao najgora odnosno F.

Razina uslužnosti A opisuje stanje na raskrižju u uvjetima kada je CD do 10 s/voz, a VCR je manji od 1.0. Opisuje prometni tok koji je značajno ispod kapaciteta prometnice. Većina vozila prolazi za vrijeme zelene faze signalnog plana bez zaustavljanja na raskrižju.

Razina uslužnosti B opisuje stanje na raskrižju u uvjetima kada je CD između 10 s/voz i 20 s/voz, a VCR je manji od 1.0. Prometni tok je još uvijek značajno ispod kapaciteta prometnice, ali na raskrižju se zaustavlja više vozila nego u slučaju A.

Razina uslužnosti C opisuje stanje na raskrižju u uvjetima kada je CD između 20 s/voz i 35 s/voz, a VCR je manji od 1.0. Na raskrižju se zaustavlja značajan broj vozila. Može biti uzrokovano i duljinom ciklusa.

Razina uslužnosti D opisuje stanje na raskrižju u uvjetima kada je CD između 35 s/voz i 55 s/voz, a VCR je manji od 1.0. Mnogo vozila se zaustavlja na raskrižju te dolazi do stvaranja repova čekanja. Može biti uzrokovano pre dugim ciklusom ili zbog velikog trenutnog volumena prometa.

Razina uslužnosti E opisuje stanje na raskrižju u uvjetima kada je CD između 55 s/voz i 80 s/voz, a VCR je manji od 1.0. Prometni volumen je vrlo blizu kapacitetu prometnice, ciklusi su dugi i nastaju repovi čekanja.

Razina uslužnosti F opisuje stanje na raskrižju u uvjetima kada je CD veći od 80 s/voz, a VCR je manji od 1.0. Opisuje prometni tok pri kapacitetu prometnice uz velike repove čekanja.

Tablica 4.2: Vrijednosti upravljačkog vremena kašnjenja i razine uslužnosti, [1]

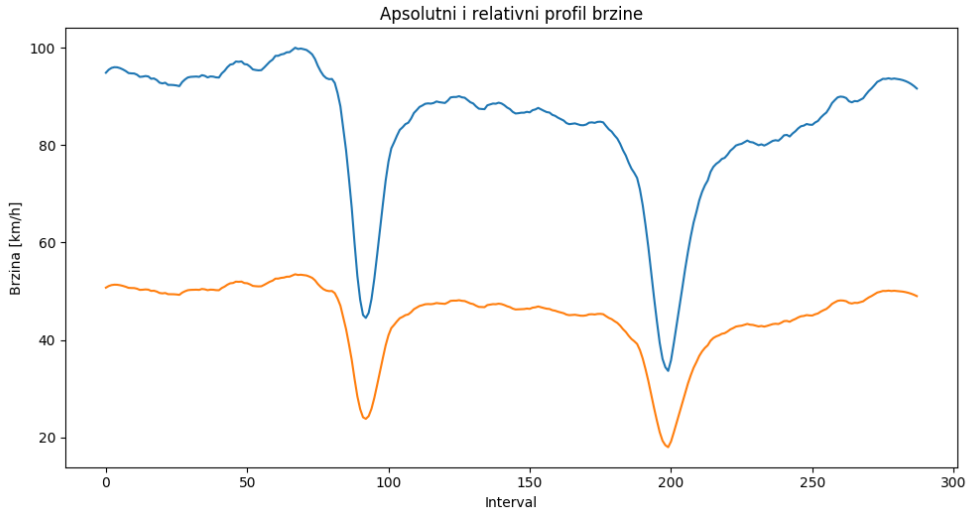
CD [s/veh]	LOS($VCR <= 1.0$)	LOS($VCR > 1.0$)
≤ 10	A	F
$\langle 10, 20]$	B	F
$\langle 20, 35]$	C	F
$\langle 35, 55]$	D	F
$\langle 55, 80]$	E	F
> 80	F	F

4.3 Određivanje razine uslužnosti pomoću GPS zapisa

Kako bi se procijenila razina uslužnosti za urbane prometnice i raskrižja potrebno je prikupiti podatke i izračunati spomenute parametre CD i FFSP. HCM metodologija predlaže izračun tih vrijednosti korištenjem podataka prikupljenih na brojačkim mjestima uz prometnicu. U ovom radu prikazana je metoda izračuna spomenutih parametara korištenjem GPS zapisa.

4.3.1 Izračun pada brzine FFSP i razine uslužnosti

Razina uslužnosti urbane prometnice prema HCM-u procjenjuje se u odnosu na postotni pad brzine u odnosu na brzinu slobodnog toka. Za izračun postotnog pada brzine korišteni su profili brzine, [12]. Profil brzine je funkcija koja prikazuje prosječnu brzinu prometovanja na nekoj prometnici kroz cijeli dan. Ako se prikaže relativni profil brzine u odnosu na maksimalnu prosječnu brzinu postignutu na prometnici može se dobiti profil postotnog pada brzine koji se koristi za procjenu razine uslužnosti prometnice. Slika 4.1 prikazuje apsolutni profil brzine (narančasta) s izračunatim vrijednostima i relativni profil brzine (plava) na lokaciji Jadranskog mosta u Zagrebu.



Slika 4.1: Apsolutni i relativni profil brzine

4.3.2 Izračun upravljačkog vremena kašnjenja CD i razine uslužnosti

CD predstavlja vremensko kašnjenje uzrokovano utjecajem vertikalne prometne signalizacije (semafor ili prometni znak). Približno odgovara vremenu koje vozilo provede u repu čekanja uvećano za: (i) vrijeme koje je utrošeno za smanjivanje brzine prilikom ulaska u rep čekanja i (ii) povećanjem brzine prilikom izlaska iz raskrižja. CD se može odrediti za jedno vozilo ili kao prosječna vrijednost za sva promatrana vozila u nekom vremenskom periodu. Također može se odrediti za cijelo raskrižje, prilaz raskrižju ili jednu signalnu grupu.

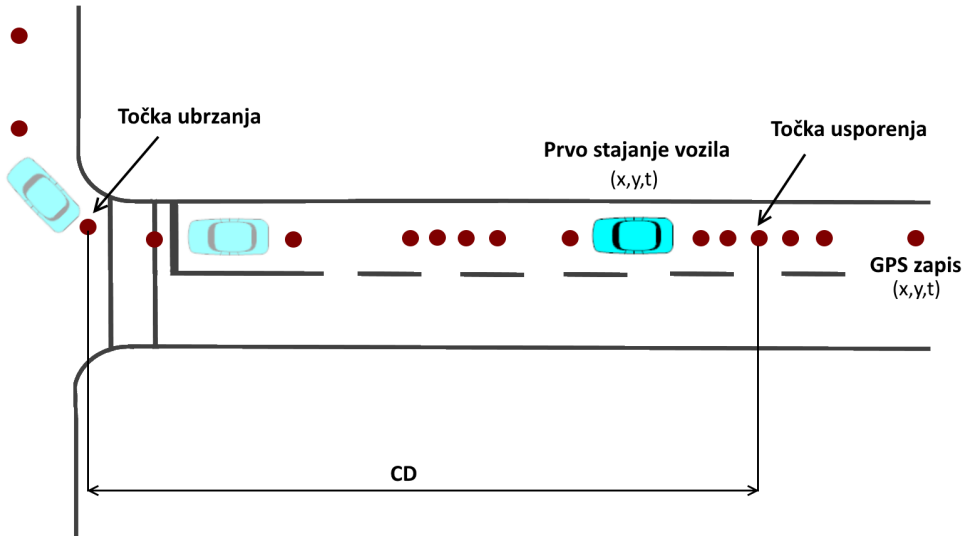
U ovom radu CD se određuje kao vrijeme potrebno kako bi vozilo prošlo između dva GPS zapisa koji označavaju točke usporenja i ubrzanja. Točku usporenja definiramo kao točku u kojoj je vozilo počelo smanjivati brzinu zbog ulaska u rep čekanja. Točku ubrzanja definiramo kao točku u kojoj je vozilo napustilo raskrižje i postiglo dovoljnu brzinu za nastavak normalnog prometovanja. Slika 4.2 prikazuje način izračuna obje točke.

Točka usporenja definirana je na slijedeći način:

1. Odrediti prvu točku na ruti u kojoj je vozilo stalo na prilazu raskrižju.
2. Pronaći prvu točku koja je udaljena minimalno 50 m od prvog stajanja vozila.

Točka ubrzanja definirana je na slijedeći način:

1. Odrediti najbližu točku stop liniji.
2. Pronaći točku na ruti zabilježenu najmanje 50 m nakon najbliže točke stop liniji.

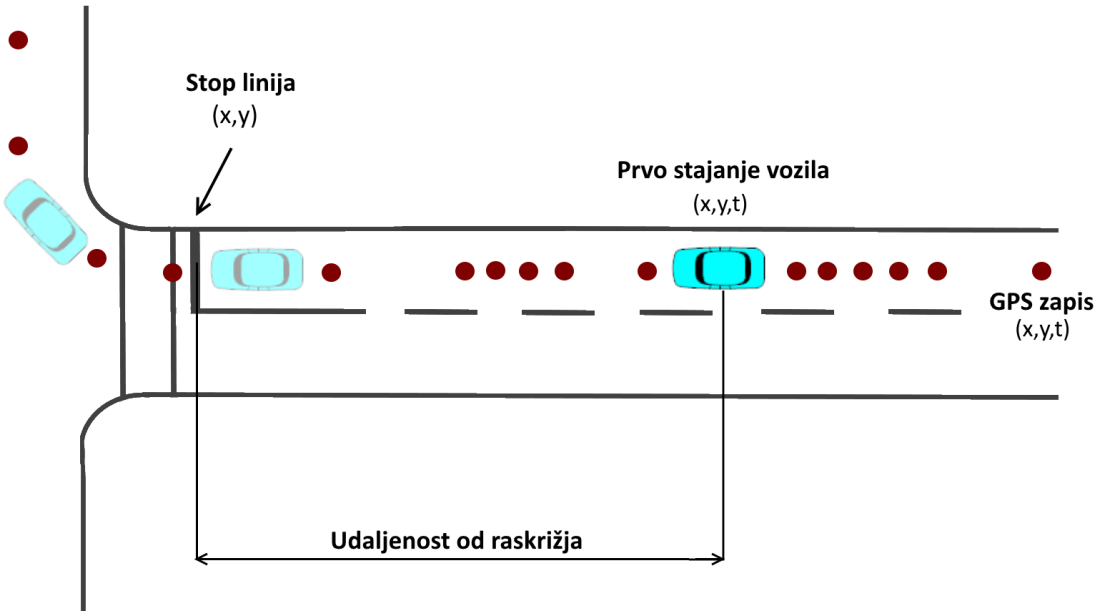


Slika 4.2: Procjena CD parametra

Repovi čekanja na prilazima raskrižju predstavljaju jednu od mjera kvalitete usluge raskrižja odnosno implementiranog signalnog plana. Detekcijom raskrižja s velikim repovima čekanja mogu se pronaći raskrižja s niskom razinom učinkovitosti na kojima je potrebno napraviti određene intervencije u signalnom planu radi povećanja protočnosti. Repovi čekanja predstavljaju problem ne samo zbog povećanja vremena putovanja već i zbog zagađenja okoliša uzrokovanog stani-kreni vožnjom.

4.4 Određivanje repova čekanja pomoću GPS zapisa

U ovom radu prikazati će se metodologija procjene repova čekanja korištenjem GPS zapisa praćenih vozila. Vozila se u ovom slučaju ponašaju kao senzori koji u diskretnim vremenskim trenucima šalju podatke o svojim geografskim koordinatama, brzini i trenutnom vremenu.



Slika 4.3: Procjena udaljenosti od raskrižja

Korištenjem tih podataka moguće je procijeniti duljinu repa čekanja prema sljedećim koracima:

1. Detektirati prvu točku na ruti u kojoj je vozilo stalo.
2. Izračun udaljenosti od prvog stajanja do stop linije.

Kada vozilo prilazi raskrižju i ulazi u rep čekanja postoji mogućnost da će više puta krenuti i stati. Stoga je vrlo važno detektirati prvo stajanje vozila koje prikazuje trenutni rep čekanja na koje vozilo nailazi. Stajanje je detektirano u točki u kojoj vozilo postiže brzinu manju od 5 km/h, [21]. Navedena brzina dovoljno je mala u usporedbi s prosječnom brzinom koju vozila ostvaruju na urbanim prometnicama.

Za izračun udaljenosti između dvije GPS koordinate korištena je *haversine* formula, [22]. *Haversine* formula nije osjetljiva na grešku uzrokovanu Zemljinom zakrivljenosti prilikom mjerenja udaljenosti između dvije bliske točke. Svrha formule je pronalazak najkraće putanje između dvije točke na sferi. Euklidska udaljenost predstavlja udaljenost u dvodimenzionalnom prostoru zbog čega nije primjenjiva za računanje udaljenosti na sferi pogotovo kada se mjeri udaljenost između dvije vrlo udaljene točke. Prilikom korištenja *haversine* formule mora se uzeti u obzir da vrijednosti koeficijenta h ne prelazi vrijednost 1 odnosno da poprima vrijednosti u intervalu $\langle 0, 1]$, [23].

Koeficijent h je odnos između udaljenosti dvije točke d na Zemlji i polumjera Zemlje r, a iznosi $\frac{d}{r}$. Vrijednost se približava 1 kada se mjeri udaljenost između dvije točke koje se nalaze na različitim stranama Zemlje, odnosno $d \cong r$. Za vrijednosti $h > 1$, ako se koristi *haversine* formula, numerička greška raste, [23].

Haversine formula glasi (4.1)[22]:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{4.1}$$

gdje je:

- d [m] - udaljenost između dvije koordinate,
- r [m] - polumjer Zemlje,
- φ_1 - zemljopisna širina prve koordinate,
- φ_2 - zemljopisna širina druge koordinate,
- λ_1 - zemljopisna dužina prve koordinate,
- λ_2 - zemljopisna širina druge koordinate.

Poglavlje 5

Programsko rješenje

U svrhu izračuna i prikaza rezultata izrađena je aplikacija korištenjem *Python* programskog jezika. U aplikaciju su ugrađeni svi potrebni algoritmi i postupci korišteni pri izradi rada (i) učitavanje i predobrada podataka, (ii) spremanje podataka u NoSQL bazu podataka, (iii) izračun profila brzina, (iv) izračun profila repova čekanja, (v) izračun razine uslužnosti za urbane prometnice i raskrižja, (vi) algoritam strojnog učenja za pronalazak sličnih raskrižja.

5.1 Baza podataka

Za pohranu podataka odabrana je NoSQL baza podataka MongoDB. MongoDB je datotečna baza podataka koja za pohranu podataka koristi BSON datoteke. Navedena baza podataka opisana je u poglavlju 3. Podaci za svako raskrižje podijeljeni su u datotečne strukture (kolekcije) na način da svako raskrižje sadrži pet kolekcija:

1. imeRaskrižja_routes - učitani podaci iz $.csv$ datoteka. Sadrži sve rute vozila koje su zabilježene na raskrižju;
2. imeRaskrižja_stops - sadrži svako detektirano stajanje vozila u repu čekanja na prilazu raskrižju;
3. imeRaskrižja_cross_profiles - izračunati profili stajanja koji prikazuju duljinu repa čekanja na svim prilazima raskrižju kroz cijeli dan;
4. imeRaskrižja_speed_profiles - izračunati profili brzina koji prikazuju promjenu brzine na svim prilazima raskrizju kroz cijeli dan;
5. imeRaskrižja_control_delay - izračunato upravljačko vrijeme kašnjenja. Sadrži vrijednosti upravljačkog vremena kašnjenja kroz cijeli dan i za vršne sate.

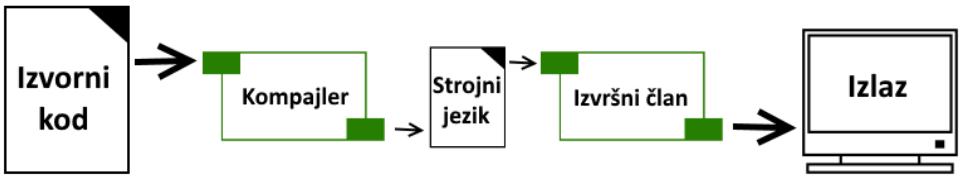
Svaka kolekcija sadrži dokumente odnosno podatke strukturirane u JSON formatu. Ovakav način spremanja podataka omogućava brz pristup podacima zbog spremanja u posebne kolekcije i jednostavniju obradu podataka zbog čitkog formata.

5.2 Programski jezik

Python je viši programski jezik (*engl. High-level*). Objektno je orijentiran no može se koristiti i kao funkcijski ili skriptni programski jezik. Python posjeduje jezični prevoditelj (*engl. Interpreter*) koji ga razlikuje od programskih jezika koji koriste *compiler* za prevođenje i izvršavanje koda, [24]. Slika 5.1 prikazuje razliku između *interpreterskih* i *compilerskih* jezika. Pri izradi ovog rada korišten je *interpreter* Python 3.6, [25].



(a) Interpreterski programski jezik



(b) Compilerski programski jezik

Slika 5.1: Razlika između interpreterkog i compilerskog jezika

Python je programski jezik otvorenog koda (*engl. Open-source*) uz koji dolazi s mnogo ugrađenih programskih biblioteka (*engl. Libraries*) koje sadrže mnoge korisne funkcionalnosti. Jezik je proširiv s mnogim javno dostupnim bibliotekama koje ubrzavaju proces razvoja aplikacija.

Biblioteke korištene u ovom radu su:

1. *NumPy* - biblioteka se koristi za strukturiranje podataka u matrični oblik i omogućava osnovne operacije nad matricama, [26];
2. *matplotlib* - biblioteka omogućava grafički prikaz izračunatih vrijednosti u raznim oblicima grafova, [27];
3. *sklearn* - biblioteka koja sadrži funkcije korištene za implementaciju algoritama strojnog učenja, [28];
4. *scipy* - biblioteka koja sadrži funkcije koje se koriste za statističku obradu podataka, [29].

5.3 Strojno učenje

Strojno učenje se može definirati kao skup algoritama koji omogućuju automatsko detektiranje korisnih pravilnosti i obrazaca u podacima, [30]. Prvu definiciju strojnog učenja dao je Arthur Samuel 1959. godine koja glasi da je strojno učenje područje istraživanja koje računalima omogućava da uče i donose odluke bez izričitog programiranja, [31]. Danas se algoritmi strojnog učenja koriste u mnogim aplikacijama od detekcije neželjene pošte, prijedloga marketinškog sadržaja, upravljanja prometom i transportom do upravljanja autonomnih vozila. Zajednička karakteristika ovih aplikacije je ta da se odluke donose na temelju velikog broja podataka koje čovjek u realnom vremenu ne može obraditi niti donijeti odluku.

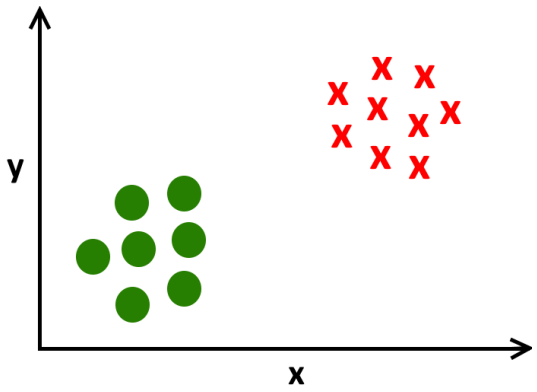
5.3.1 Podjela algoritama strojnog učenja

Algoritam strojnog učenja je računalni proces koji koristi ulazne podatke kako bi izvršio određeni zadatak, s vremenom učio i postajao sve kvalitetniji u rješavanju sličnih zadataka. Proces učenja algoritma se naziva trening (*engl. Training*).

Nadzirano učenje

Korištenjem tehnike nadziranog učenja algoritmu su dani označeni (*engl. Labeled*) podaci. Što znači da je za svaki ulaz u algoritam dobro definiran i poznat izlaz. Algoritam ”uči” relacije i povezanost između ulaznih i izlaznih podataka na skupu podataka za trening (oko 70 % svih podataka). Kada mu se preda nepoznat (testni) ulazni skup podataka, na temelju naučenog može predvidjeti izlaz odnosno rješenje problema. Slika 5.2 prikazuje grafičko rješenje korištenjem nadziranog učenja na problemu klasifikacije. Crvenom i zelenom bojom označene su klase

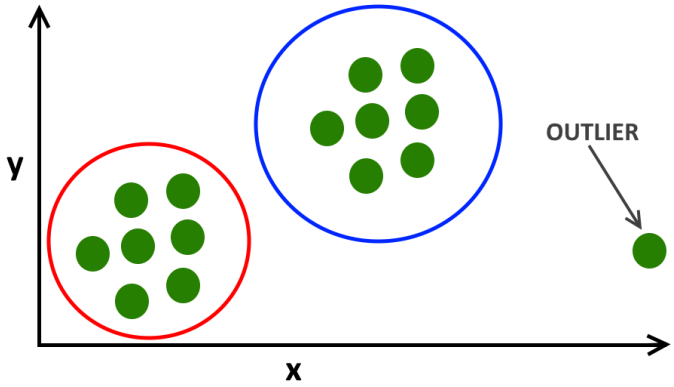
određene na temelju označenog skupa podataka. Novi podatak biti će klasificiran u jednu od klasa temeljem svojih ulaznih vrijednosti.



Slika 5.2: Nadzirano učenje

Nenadzirano učenje

Tehnike nenadziranog učenja koriste se kada se raspolaže neoznačenim podacima. Nenadziranim učenjem rješavaju se problemi poput: grupiranja, određivanja vrijednosti koje odskaču (*engl. Outlier*) i smanjenje dimenzionalnosti (*engl. Dimensionality reduction*). Slika 5.3 grafički prikazuje navedene probleme. Ulaz u algoritam je neoznačeni skup podataka. Kada se primjenjuje tehnika nenadziranog učenja nije moguće dobiti „točan“ odgovor, nego samo pripadnost podatka nekom skupu. U ovom radu koristiti će se tehnike nenadziranog učenja za grupiranje raskrižja. Cilj je detektirati raskrižja s niskom razinom učinkovitosti i svrstati ih u jednu grupu (klasu).



Slika 5.3: Nenadzirano učenje

Ojačano učenje

Tehnika ojačanog učenja se temelji na „nagrađivanju“ i „kažnjavanju“ algoritma. Slično kao i kod nenadziranog učenja, algoritmu se predaju samo ulazni podaci, bez izlaznih podataka. Ako algoritam dobro riješi problem dobiva nagradu, a inače bude kažnjen.

5.3.2 Problemi

Primjena algoritama strojnog učenja može se opisati s problemima koje rješavaju. Problemi se dijele u nekoliko skupina: (i) klasifikacija (*engl. Classification*), (ii) grupiranje (*engl. Clustering*) (iii) i predikcija (*engl. Forecasting*).

Klasifikacija

Problem klasifikacije se svodi na određivanje klase (vrste) podatka na temelju ulaznih karakteristika. Klase su unaprijed definirane, što je temeljna razlika u odnosu na problem grupiranja koji će se obraditi u nastavku. Nakon definiranja klasa prikupljaju se atributi svakog podatka. Na temelju atributa algoritam dodjeljuje podatak u klasu kojoj pripada. Za rješavanje problema klasifikacije često se koriste tehnike nadziranog učenja.

Binarna klasifikacija najčešći je oblik problema strojnog učenja. U jednostavnoj formi odgovara na pitanje: ako je dan x iz domene X, koja će biti vrijednost binarne varijable $y \in \{\pm 1\}$. Jednostavni primjer binarne klasifikacije u prometu je podjela prometnica na urbane prometnice i ne urbane prometnice. Višeklasna klasifikacija je proširenje binarne klasifikacije. Glavna razlika je u tome što je $y \in \{1, ..., n\}$ odnosno postoje više od dvije klase, [32].

Grupiranje

Grupiranje podrazumijeva podjelu podataka u grupe temeljem zajedničkih karakteristika. Tehnika se odnosi na neoznačene podatke, odnosno na podatke za koje nisu unaprijed poznate i definirane klase kojima pripadaju. Svrha je pronaći i grupirati slične podatke. Sličnost podataka definira algoritam na temelju ulaznih podataka. Algoritmi grupiranja koriste se za istraživanje podataka, smanjenje broja značajki, kompresiju podataka i dr., [33].

Postoji više vrsta algoritama grupiranja, a dijele se na [34]: (i) patricijsko grupiranje (*engl. Partitional clustering*) čija je svrha stvaranje međusobno nepovezanih grupa koje pripadaju točki koja reprezentira grupu (centroid) (ii) hijerarhijsko grupiranje (engl. Hierarchical clustering) koje kao rezultat grupiranja daje stablo (dendrogram).

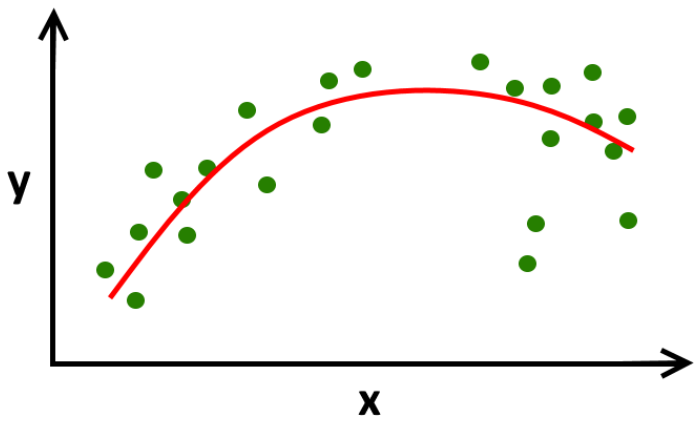
Razlika između metoda je u tome što algoritmi patricijskog grupiranja kao ulaz primaju podatke i broj grupa, a hijerarhijski algoritmi grupiranja zahtijevaju uvjet prekida. Također, složenost particijskih algoritama je linearna dok je za hijerarhijske većinom kvadratna, [35].

Algoritmi grupiranja se mogu podijeliti i prema "čvrstoći" granica između grupa:

- Čvrsto grupiranje (*engl. Hard Clustering*) - svaki podatak pripada isključivo jednoj grupi (npr. K-means algoritam).
- Meko grupiranje (*engl. Soft Clustering*) - podatak može pripadati više grupa (npr. *Fuzzy K-means* i Probabilističko grupiranje).

Predikcija

Rezultat rješavanja problema predikcije je predviđanje budućeg kretanja podataka na temelju prikupljanog znanja i iskustva. Cilj regresijske analize je pronaći regresijsku krivulju koja najbolje opisuje podatke, odnosno posjeduje minimalnu udaljenost između svake točke (podatka) odnosno minimizira grešku (odstupanje) regresijskog vektora od podatka. Slika 5.4 prikazuje regresijsku krivulju i podatke označene zelenim točkama. Za zadani skup podataka X potrebno je pronaći funkciju $f(x)$ koja vrijednostima iz skupa X pridružuje vrijednosti iz skupa R na način da $f(x)$ poprima minimalnu udaljenosti od x, [32].



Slika 5.4: Regresija

5.3.3 K-means

K-means je algoritam grupiranja koji se može svrstati u algoritme nenadziranog učenja. Koristi se u slučajevima kada se obrađuje ne označeni skup podataka kojega je potrebno podijeliti u fiksni broj grupa u kojima podaci posjeduju slične karakteristike. Glavni cilj je definirati k centroida za svaku grupu. Prvi korak algoritma je odabir broja grupa i inicijalno pozicioniranje centroida. Nakon pozicioniranja centroida računa se udaljenost između određenih centroida i svakog podatka. Centroidi se pomiču s obzirom na udaljenost od podataka. Ovaj postupak se iterativno ponavlja sve dok nema promjena u poziciji centroida, [36], [37].

Algoritam 1 K-means algoritam, [36]

Data: k-broj centroida

Result: odabranih k centroida

odabir inicijalnih pozicija k centroida **while** *(nije došlo do promjene pozicije centroida)* **do**

izračun udaljenosti točaka od centroida pomicanje centroida na način da udaljenost između podataka i centroida bude najmanja

end

Algoritam pronalazi najmanju udaljenost između centroida i podataka, stoga se ciljna funkcija može izraziti kao (5.1):

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \|x_i^{(j)} - c_j\|^2 \tag{5.1}$$

gdje je:

- $\|x_i^{(j)} - c_j\|^2$ - udaljenost između $x_i^{(j)}$ i c_j,
- $x_i^{(j)}$ - podatak,
- c_j - centroid grupe.

Koraci algoritma prikazani su algoritmaom 1. Inicijalna (početna) pozicija centroida može mnogo utjecati na rezultat pa se preporuča postavljanje centroida što je moguće dalje jedan od drugog. Problem inicijalnog postavljanja centroida David Arthur i Sergei Vassilvitskii poboljšali su prijedlogom K-means++ algoritma koji odabire početne položaje k centroida. Algoritam 2 prikazuje dio K-means++ algoritma koji odabire početne položaje k centroida gdje je $D(x)$ minimalna udaljenost podatka od najbližeg centra koji već odabran, [38].

Algoritam 2 K-means++ odabir inicijalnih pozicija centroida, [38]

Data: k-broj centroida; X-podaci

Result: Inicijalne pozicije centroida za K-means algoritam

odaberi centroid k_1 (odabran uniformnom razdiobom iz skupa X) **while** *(nije odabrano k centroida)* **do**

  odaberi novi centroid k_i, s izabranim $x \in X$ sa vjerojatnošću $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$

end

nastavi s K-means algoritmom

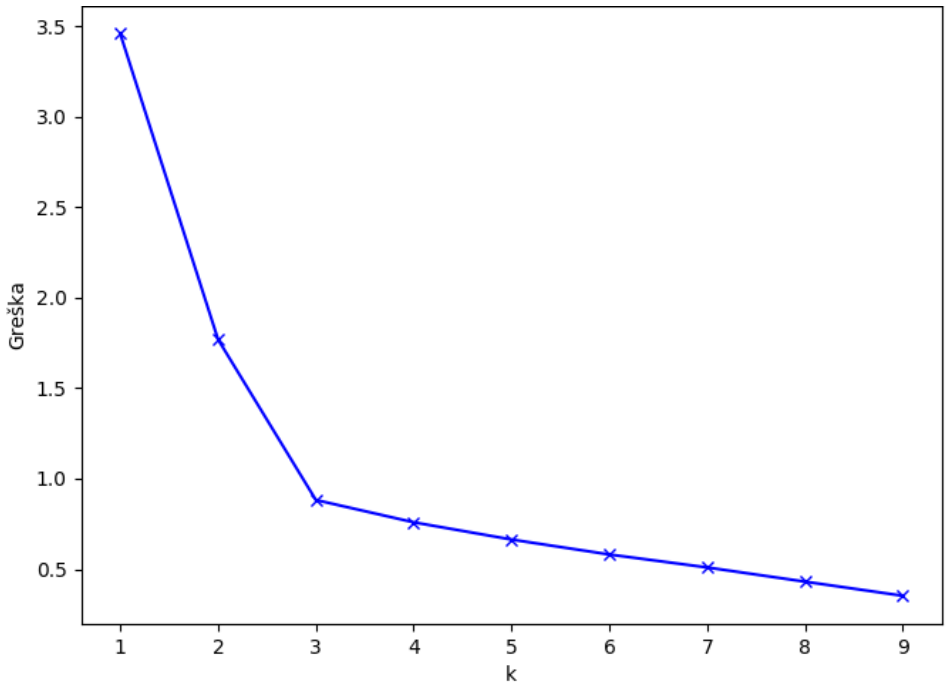
U ovom radu K-means++ algoritam koristi se za grupiranje raskrižja. Temeljem karakterističnih parametara raskrižja će biti podijeljena u grupe koje će prikazivati podjelu prema koeficijentima učinkovitosti raskrižja. Na taj način se raskrižja s niskom razinom učinkovitosti trebaju grupirati u jednu grupu. Svako raskrižje određeno je s jednim vektorom vrijednosti koji sadrži parametre prema kojima se vrši grupiranje. Svaki vektor može biti prikazan u n-dimeznionalnom prostoru gdje je n jednak broju parametara koji se nalaze u vektoru. U ovome radu vektor sadrži slijedeće parametre:

- upravljačko kašnjenje (LOS),
- brzina (LOS),
- duljina repa čekanja.

Jedan od problema koji se javlja prilikom implementacije K-means algoritma je nepoznati inicijalni broj centroida k. Ako je na početku implementacije nepoznat broj k, kao što je slučaj u ovom radu, može se koristiti metoda koja pomaže pri odabiru optimalnog broja centroida. Neke od metoda koje se mogu koristiti za odabir optimalnog broja centroida su: Metoda koljena (*engl. Elbow method*) i Analiza siluete (*engl. Silhouette analysis*).

Metoda koljena

Metoda koljena prikazuje optimalan broj centroida k tako što pokreće K-means algoritam više puta za vrijednosti $k = \{1, 2, ...n\}$ i prikazuje kvadratno odstupanje (grešku) za svaki broj k. U "koljenu" dijagrama treba se nalaziti optimalan broj centroida za K-means algoritam, [39]. Ako se promatra centroid $k = m$, koljeno dijagrama se pojavljuje kada je greška za $k = m - 1$ puno veća nego za $k = m$. Na slici 5.5 optimalan broj k za imaginaran primjer je $k = 3$.



Slika 5.5: Primjer Metode Koljena

Analiza siluete

Analiza siluete je metoda koja se koristi za određivanje optimalnog broja centroida k. Metodom se izračunava udaljenost svake točke od dodijeljenog cetroida i izračunava udaljenosti svake točke od susjednih centroida. Analiza siluete provodi se korištenjem izraza (5.2):

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{5.2}$$

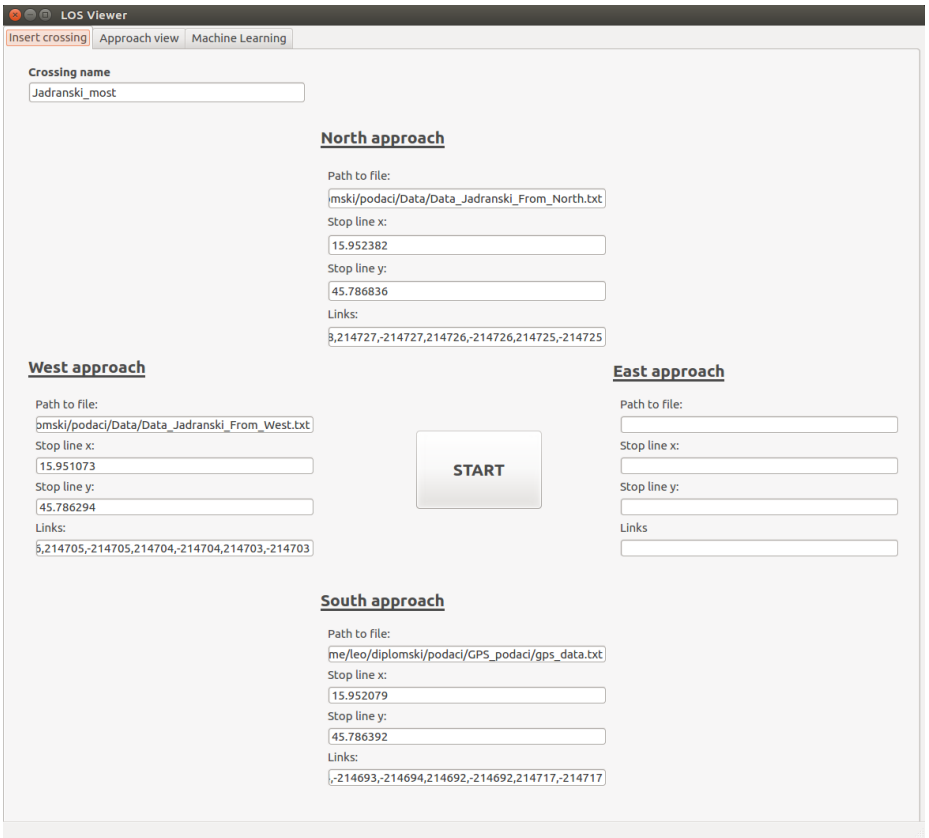
gdje je:

- $s(i)$ - vrijednost siluete,
- $a(i)$ - prosjek udaljenosti točke od centroida kojem pripada,
- $b(i)$ - prosjek udaljenosti točke od susjednih centroida.

Vrijednosti siluete mogu se kretati u intervalu $[-1, 1]$. Vrijednost -1 znači da je točka vrlo udaljena od centroida kojemu je dodijeljena i zaključuje se da je loše klasificirana. Ako je vrijednost siluete 0 to znači kako se točka nalazi na granici između susjednih centroida. Vrijednost 1 znači da je točka vrlo blizu dodijeljenom centroidu i to predstavlja idealan slučaj klasifikacije.

5.4 Grafičko sučelje

Grafičko sučelje sastoji se od tri glavna prozora: *Insert crossing*, *Approach view* i *Machine learning*. *Insert crossing* (Slika 5.6) je prozor koji omogućava korisniku unos podataka kroz grafičko sučelje u bazu podataka. Kako bi korisnik unio podatke potrebno je definirati putanju do datoteke koja sadrži podatke za odabrani prilaz (*Path to file*). Polja *Stop line x* i *Stop line y* služe za unos koordinata stop linije na odabranom prilazu. U polju *Links* nalaze se linkovi koji opisuju raskrižje s digitalne karte. Podaci se spremaju u bazu podataka na način opisan u poglavlju 5.1. U polje *Crossing name* upisuje se ime raskrižja koje se želi obraditi. Prozor sadrži četiri dijela s definiranim nazivima *North, East, South* i *West approach* kako bi korisnik bio u mogućnosti unijeti podatke za svaki za svaki prilaz u ovisnosti o geografskom položaju prilaza. Ako ne postoje podaci za neki prilaz, polja se ostavljaju prazna.



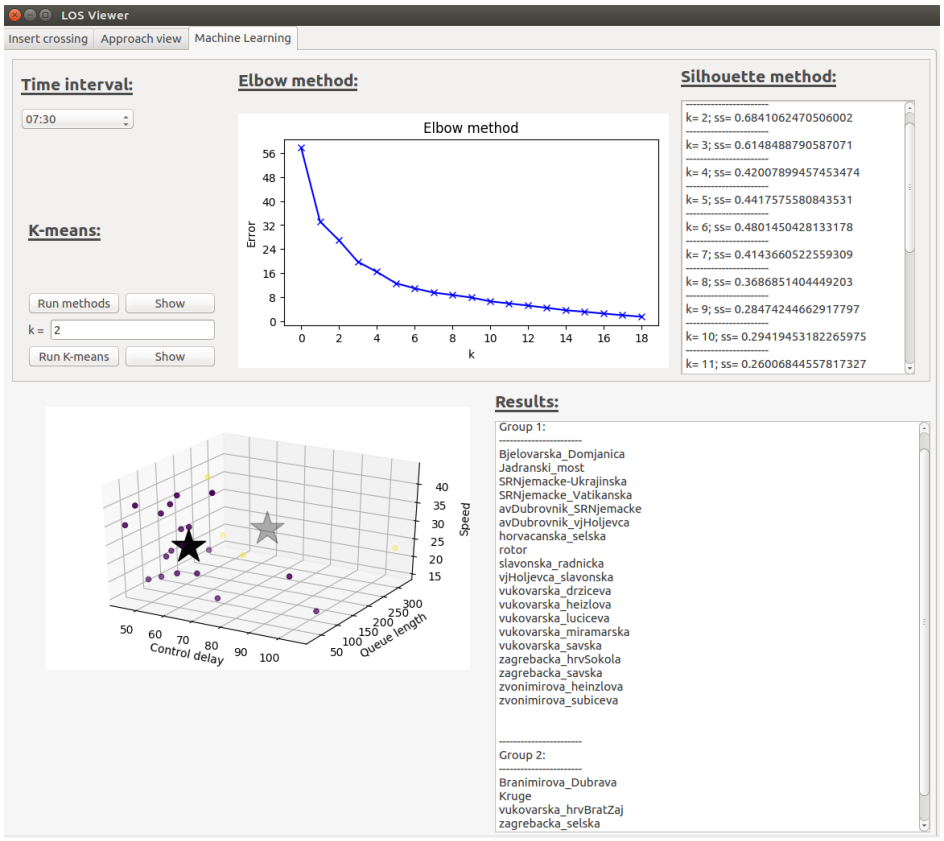
Slika 5.6: GUI - Unos podataka za obradu

Drugi prozor *Approach view* (Slika 5.7) omogućava korisniku pregled profila stajanja i profila brzina za odabrani prilaz raskrižju. U polje *Crossing name* upisuje se ime raskrižja koje se želi obraditi. U padajućem izborniku *Month* moguće je odabrati željeni prikaz za ljetne (*engl. summer*) i ne ljetne mjesece (*engl. other*). Padajući izbornik *Day* omogućava odabir dana u tjednu, radnih dana ili vikenda. U izborniku *Bound* može se odabrati prilaz koji se promatra u ovisnosti o geografskom položaju prilaza. Gumb *Submit* služi za dohvat podataka iz paze podataka i pripremu podataka za prikaz. Nakon pritiska na gumb *Show diagrams* prikazuju se tri grafička prikaza. Prvi prikazuje profil brzine i profil stajanja na jednom grafu. Ovaj prikaz služi kako bi se grafički prikazala ovisnost dvaju profila. Drugi prikazuje profil repa čekanja kroz dan. Plavom bojom označen je zaglađeni profil dok su crvenom iscrtanom linijom prikazani rezultati nakon izračuna. Treći prikazuje profil brzine kroz dan.



Slika 5.7: GUI - prikaz repova čekanja i profila brzina

Treći prozor *Machine learning* omogućava korištenje K-means algoritma za grupiranje raskrižja. U padajućem izborniku *Time interval* odabire se jedan petominutni interval za koji će se provesti grupiranje. Gumbi *Run methods* i *Show* služe za izračun optimalnog broja centroida k za raskrižja u definiranom vremenskom intervalu. Na temelju grafičkog prikaza *Elbow* metode ili tekstualnom prikazu *Silhuouette method* korisnik može odabrati broj k koji će se koristiti u nastavku izračuna i upisati ga u odgovarajuće polje. Gumbima *Run K-means* i *Show* pokreće se K-means algoritam i prikazuju se rezultati. Rezultati su prikazani grafički i tekstualno. U polju *Results* prikazane su grupe i raskrižja koja pripadaju svakoj grupi. Grupe su prikazane počevši od one koja sadrži najmanje rezultate (niske vrijednosti parametara t_{cd}, repa čekanja i brzine).



Slika 5.8: GUI - strojno učenje

Poglavlje 6

Rezultati istraživanja

Rezultati istraživanja repova čekanja i razine uslužnosti prikazani su za šest velikih raskrižja za koje je prikupljeno najveći broj podataka. Rezultati za sva ostala raskrižja dani su prilozima A i B. Rezultati za grupiranje i detekciju sličnih raskrižja prikazani su za sva analizirana raskrižja. Ukupno je analizirano 23 raskrižja. Svi prikazani rezultati izračunati su pomoću razvijene aplikacije. Aplikacija omogućava analizu svakog prilaza zasebno za svaki dan u tjednu, posebno za radne i neradne dane, za ljetne i ne-ljetne mjesece. U ovom poglavlju istaknuti su samo rezultati vezani za radne dane u ne-ljetnim mjesecima jer takvi rezultati daju pravu sliku raskrižja u svakodnevnom prometovanju.

6.1 Repovi čekanja na prilazima raskrižjima

Rezultati duljine repova čekanja prikazani su grafovima koji prikazuju vrijednosti duljine repova čekanja za sve prilaze raskrižju kroz cijeli dan. Rezultati su grupirani na način da prikazuju vrijednosti za radne dane koji su generalno važniji za prometne analize. Slika 6.1 prikazuje dobivene rezultate za analizirana raskrižja i prilaze. Svaki profil duljine repa čekanja izračunat je agregiranjem vrijednosti repova čekanja za sve rute koje prolaze razmatranim prilazom raskrižju.

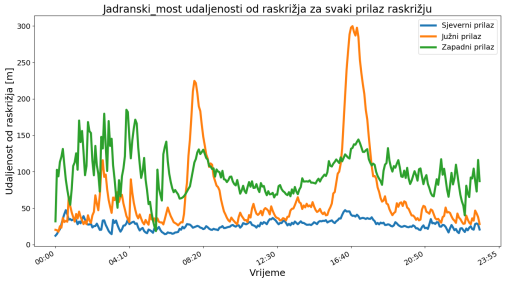
Agregirane vrijednosti za radne dane podijeljeni su u peto-minutne intervale te su vrijednosti uprosječene čime je dobiven profil repa čekanja. Svi prikazani rezultati uključuju vrijednosti za ne ljetne mjesece, odnosno sve mjesece osim srpnja i kolovoza. Tablica 6.1 prikazuje broj podataka na temelju kojih su dobiveni rezultati. Stupac *Prilaz* imenuje svaki prilaz promatranom raskrižju u ovisnosti o geografskom položaju prilaza (sjever - S, istok - I, jug - J, zapad - Z). U stupcu *GPS zapisi* nalazi se broj prikupljenih GPS zapisa praćenih svih vozila koja su prometovala prilazom. *Broj ruta* prikazuje ukupni broj ruta odnosno vozila koja si prolazila prilazom.

Tablica 6.1: Broj prikupljenih podataka

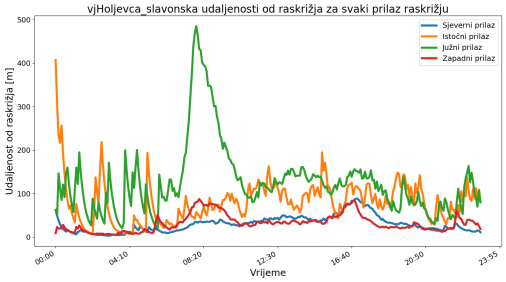
Raskrižje	$Prilaz$	GPS zapisi	Broj ruta
Jadranski most - Savska	J	1005436	109307
	S	295373	54261
	Z	262580	41845
Kruge - Zagrebačka	I	842609	102686
	Z	1497241	119862
V. Holjevca - Slavonska	I	297804	26755
	S	412528	74406
	Z	278103	30432
	J	1027991	68280

Slika 6.1b prikazuje profil repa čekanja na raskrižju *V. Holjevca - Slavonska avenija*. Može se primijetiti kako su vrijednosti za zapadni i sjeverni prilaz vrlo slične i ne prelaze 100 m. Na južnom prilazu vrijednosti su nešto veće kroz cijeli dan. Takvi rezultati prikazuju dobro optimizirane signalne planove koji omogućavaju dovoljnu protočnost na prilazu raskrižju. Rezultati za južni prilaz jasno pokazuju na kojem prilazu dolazi do smanjene učinkovitosti. Vrijednosti duljine repa čekanja su gotovo 500 m u jutarnjem vršnom satu i jasno ukazuju na stvaranje zagušenja na prilazu. Raskrižje *Kruge - Zagrebačka cesta* (Slika 6.1f) primjer je loše optimiziranog signalnog plana na cijelom raskrižju. Na glavnom prometnom pravcu istok-zapad dolazi do stvaranja repa čekanja i do 500 m u vršnim satima s tim da kroz dan vrijednosti duljine repa čekanja ne padaju ispod 200 m. S druge strane prilazi na raskrižju *Jadranski most - Savska cesta* (Slika 6.1a) dobro su optimizirani uz iznimku južnog prilaza koji u vršnim satima poprima vrijednosti duljine repa čekanja i do 300 m. Slični zaključi mogu se izvesti i iz ostalih grafičkih prikaza (Slike 6.1c, 6.1d i 6.1e).

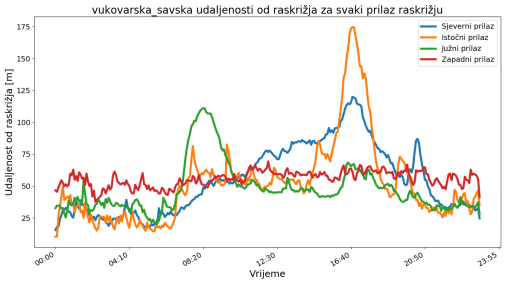
Provedena analiza pruža korisne podatke za planiranje i optimizaciju signalnih planova na raskrižjima. Također, pruža uvid u stanje na raskrižju na temelju kojeg se mogu detektirati raskrižja s niskom razinom učinkovitosti i visokim vrijednostima duljine repova čekanja.



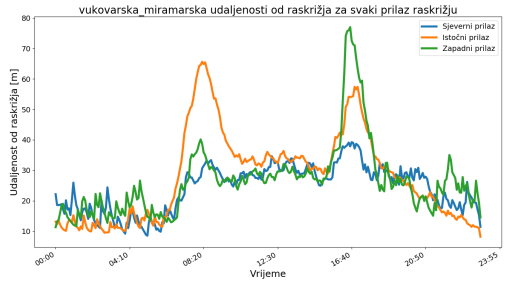
(a) Jadranski most - Savska cesta



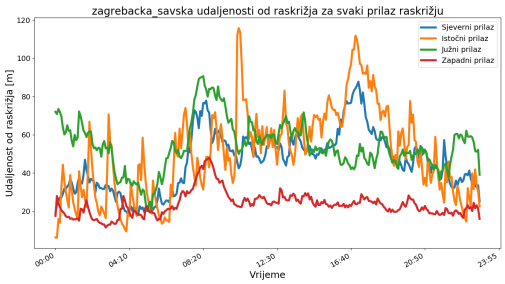
(b) Vjeceslava Holjevca - Slavonska avenija



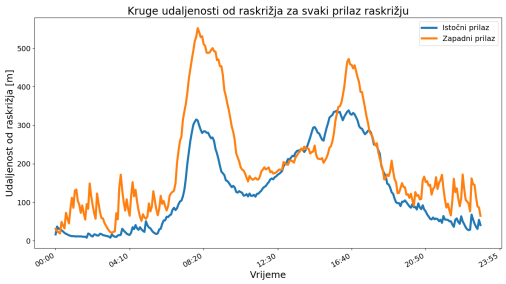
(c) Vukovarska - Savska



(d) Vukovarska - Miramarska



(e) Zagrebačka - Savska



(f) Zagrebačka - Kruge

Slika 6.1: Repovi čekanja na analiziranim raskrižjima

6.2 Razina uslužnosti

Rezultati procjene razine uslužnosti prikazani su tablicom 6.2. Tablica prikazuje vrijednosti upravljačkog kašnjenja t_{cd} i odgovarajuću razinu usluge LOS. Rezultati su prikazani za svaki prilaz te su grupirani u tri grupe koje prikazuju vrijednosti kao aritmetičke sredine kroz cijeli dan, za vršne sate i za ne vršne sate.

Tablica 6.2: Rezultati procjene t_{cd} i LOS

Raskrižje	$Prilaz$	Cijeli dan		Vršni sati		Ne-vršni sati	
		t_{cd}	LoS	t_{cd}	LoS	t_{cd}	LoS
Vukovarska - Savska	S	$92,56$	F	$106,21$	F	$76,28$	E
	I	$68,94$	E	$93,34$	F	$48,08$	D
	J	$70,31$	E	$83,87$	F	$46,36$	D
	Z	$59,07$	E	$62,54$	E	$45,34$	D
Zagrebačka - Savska	S	$57,61$	E	$63,46$	E	$41,20$	D
	I	$73,81$	E	$100,28$	F	$34,84$	C
	J	$64,60$	E	$81,93$	F	$41,73$	D
	Z	$71,42$	E	$79,184$	E	$54,12$	D
Vukovarska - Miramarska	S	$58,99$	E	$63,42$	E	$40,16$	D
	I	$46,10$	D	$53,68$	D	$29,52$	C
	Z	$79,41$	E	$92,39$	F	$50,94$	D
Jadranski most - Savska	S	$44,82$	D	$49,40$	D	$34,30$	C
	J	$50,89$	D	$70,07$	E	$34,27$	C
	Z	$67,93$	E	$78,77$	E	$48,36$	D
Kruge - Zagrebačka	I	$99,65$	F	$107,18$	F	$89,63$	F
	Z	$110,30$	F	$135,12$	F	$86,84$	F
V. Holjevca - Slavonska	S	$41,35$	D	$51,49$	D	$26,23$	C
	I	$52,62$	D	$58,30$	E	$29,59$	C
	J	$61,18$	E	$73,91$	E	$41,30$	D
	Z	$65,16$	E	$79,84$	E	$36,25$	D

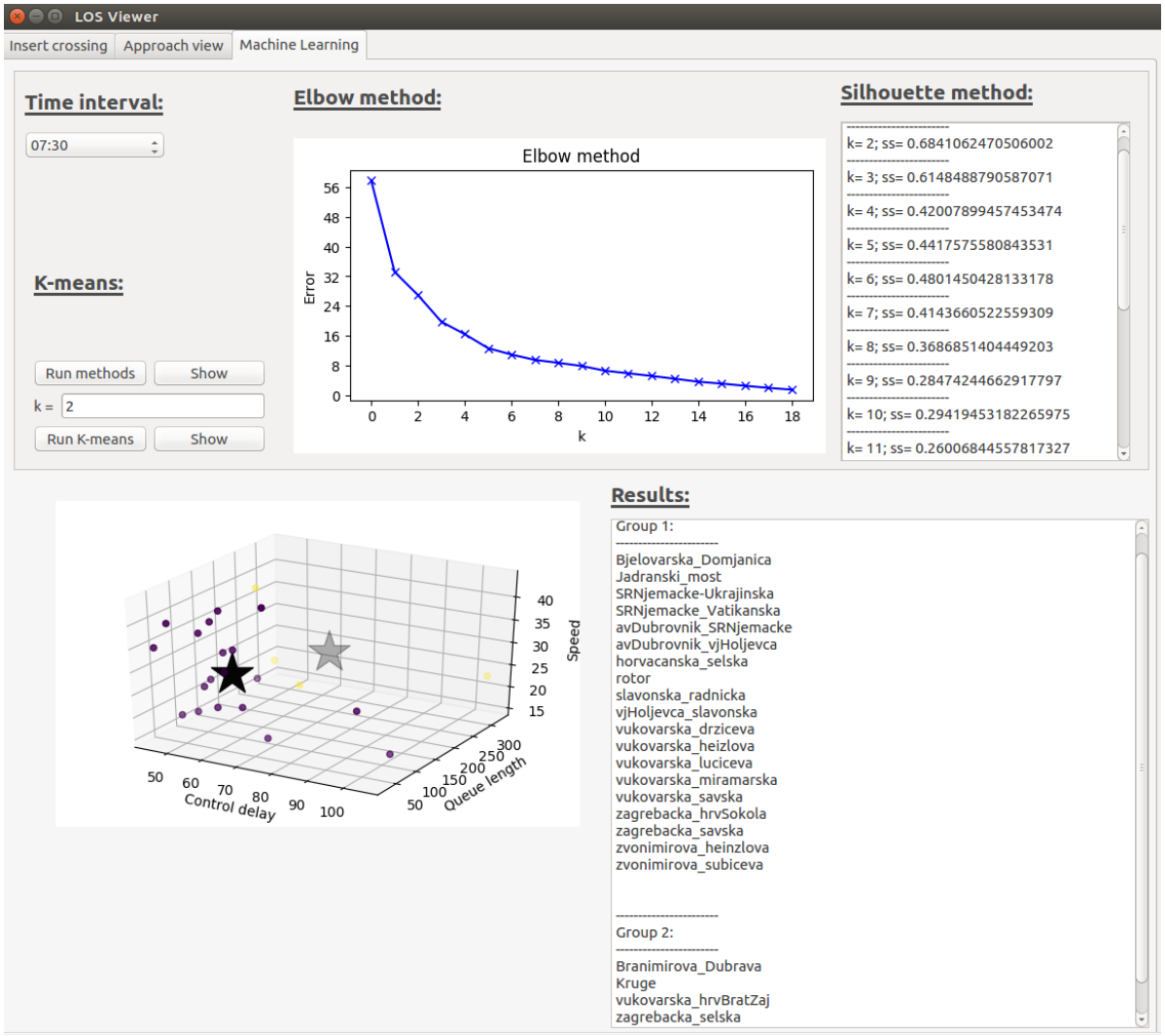
6.3 Detekcija sličnih raskrižja

Vrijednosti svih mjerenih parametara učinkovitosti ovise o dobu dana u kojemu se promatraju. Stoga se može pretpostaviti da će raskrižja biti grupirana na drugačiji način u ovisnosti o dobu dana u kojemu se promatraju. U nastavku su prikazani rezultati za tri karakteristična slučaja: (i) jutarnji vršni sat (*07:30* h) (ii) razdoblje između jutarnjeg i popodnevnog vršnog sata (*11:30* h) (iii) popodnevni vršni sat (*16:30* h). Prikazani su rezultati za radne dane u ne ljetnim mjesecima. Takvim grupiranjem dolazi se do najvećeg opterećenja prometnica.

Zbog manjka podataka i stohastične prirode procesa, nisu prikazani rezultati za noćne intervale budući da se javljaju velike oscilacije u razmatranim parametrima učinkovitosti. Iz tog razloga, noćni interval ne bi se trebalo razmatrati u peto-minutnim intervalima već kao zasebni vremenski period npr. od *05:30* do *22:00* kao što su autori koristili u [12].

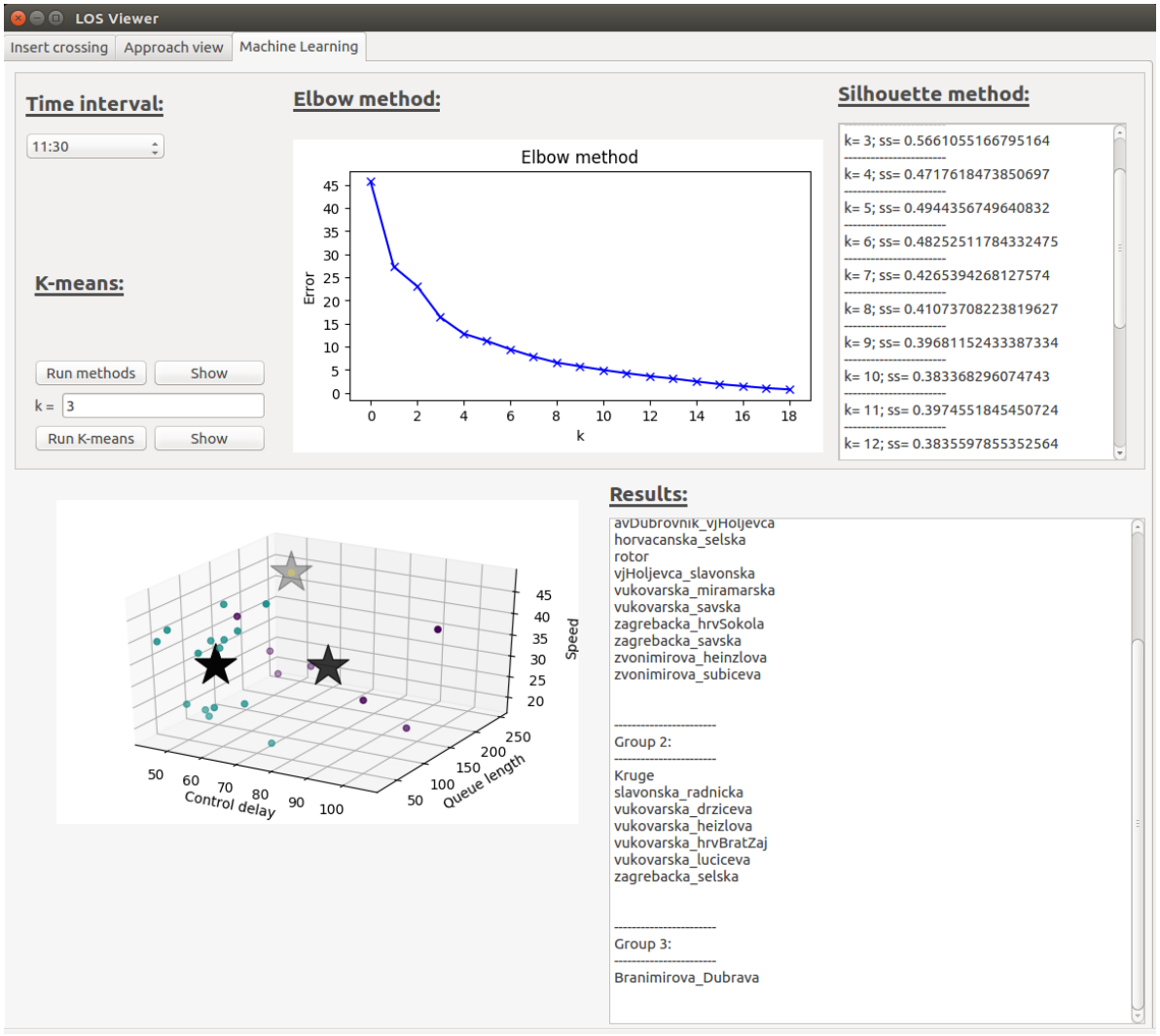
Slika 6.2 prikazuje rezultate grupiranja raskrižja u jutarnjem vršnom satu, odnosno u *07:30* h. U tom razdoblju velik je broj vozila na prometnicama. Brzine prometovanja se smanjuju na svim prilazima raskrižjima, a repovi čekanja rastu. Navedeno vrijedi za raskrižja koja su pod velikim opterećenjem najčešće zbog komutacije ljudi između doma i radnog mjesta.

Vrijednost parametra k prikazana je pomoću Metoda koljena i Analize siluete. Grafičko metodom koljena u ovom slučaju ne prikazuje jasno optimalnu vrijednost centroida k. Stoga se koristila Analiza siluete. Korišten je $k = 2$ odnosno podaci su grupirani u dvije grupe. Grupa 2 predstavlja lošije optimizirana raskrižja nego grupa 1.



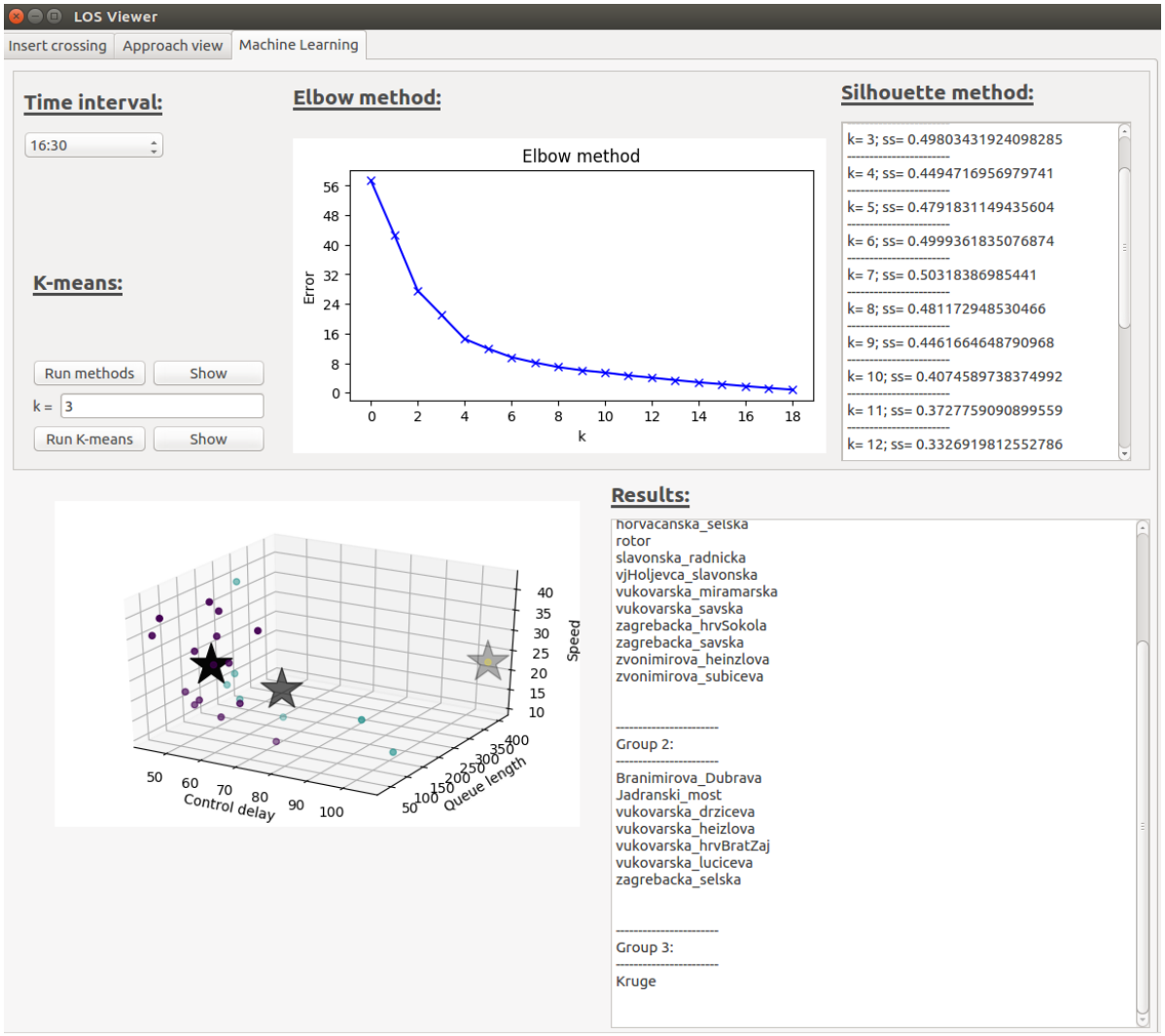
Slika 6.2: Grupiranje raskrižja u *07:30* h

Sredinom radnog dana, u *11:30* h volumen prometa manji je nego u vršnim satima, no na nekim raskrižjima zbog velike potražnje dolazi do stvaranja repova čekanja. Na slici 6.3 vidljivo je kako su raskrižja drugačije grupirana s obzirom na trenutno prometno stanje te je korišten $k = 3$.



Slika 6.3: Grupiranje raskrižja u *11:30* h

Parametri u popodnevnom vršnom satu na većini raskrižja vrlo su slični kao i u jutarnjem vršnom satu. Zbog velikog broja vozila na prometnicama, brzine se smanjuju, a repovi čekanja rastu. Slika 6.4 prikazuje da se neka raskrižja nalaze u istim grupama kao i u jutarnjem vršnom satu (Slika 6.2). Iako metoda analize siluete prikazuje vrijednost $k = 7$ kao optimalan broj centroida, odabran je $k = 3$ zbog metode koljena i malog broja podataka koji se grupiraju. Također, vrijednosti siluete za $k = 3$ i $k = 7$ vrlo su bliske.



Slika 6.4: Grupiranje raskrižja u *16:30* h

Poglavlje 7

Zaključak

U ovom radu predložena je metoda obrade podataka u svrhu procjene parametara učinkovitosti raskrižja korištenjem GPS zapisa praćenih vozila. Parametri korišteni za procjenu su: upravljačko vrijeme kašnjenja, duljina repa čekanja i brzina vozila. Prikazana je metoda procjene duljine repa čekanja korištenjem GPS zapisa. Korištenjem predložene metode, izračunati su profili duljine repova čekanja koji prikazuju promjenu duljine repa čekanja kroz dan za sve prilaze analiziranih raskrižja. Prikazana je i opisana metoda procjene upravljačkog vremena kašnjenja korištenjem GPS zapisa. Upravljačko vrijeme kašnjenja koristi se za procjenu razine uslužnosti prilaza raskrižju. Izračunati profili brzine korišteni su za procjenu razine uslužnosti prometnica. Profili brzine prikazani su relativno u odnosu na brzinu slobodnog toka i na temelju HCM metodologije procijenjena je razina uslužnosti.

Nakon procijene parametara korištena je metoda strojnog učenja kako bi se raskrižja grupirala na temelju karakterističnih parametara te su prema koeficijentima učinkovitosti "slična" raskrižja grupirana u iste grupe. Svako raskrižje opisano je pomoću vektora karakterističnih parametara. Vektor sadrži tri prethodno analizirana parametra: brzina, upravljačko vrijeme kašnjenja i duljina repa čekanja. Na taj način rezultati su se prikazali u trodimenzionalnom prostoru i nad njima je pokrenut K-means algoritam. Za procjenu optimalne vrijednosti broja grupa k korištene su Metoda koljena i Analiza siluete. Rezultat K-means algoritma su grupe vektora u trodimenzionalnom prostoru.

Za potrebu izračuna svih navedenih parametara izrađena je aplikacija korištenjem Python programskog jezika i MongoDB NoSQL baze podataka. Aplikacija sadrži tri glavne funkcionalnosti: predobrada podataka i unos u bazu podataka, izračun karakterističnih parametara raskrižja i algoritam strojnog učenja za grupiranje podataka. Aplikacija omogućava korisniku da na jednostavan način analizira i prikaže osnovne parametre raskrižja te donese zaključke o učinkovitosti raskrižja.

Istraživanje je provedeno na ukupno 23 raskrižja u gradu Zagrebu. Za svako raskrižje prikazani su profili repova čekanja za svaki prilaz raskrižju i rezultati procjene razine uslužnosti. Rezultati grupiranja raskrižja prikazani su u tri karakteristična vremenska perioda: jutarnji vršni sat (*07:30* h), razdoblje između jutarnjeg i popodnevnog vršnog sata (*11:30* h) i popodnevni vršni sat (*16:30* h). Za svaki od perioda prikazan je rezultat grupiranja grafički i tablično.

Prikazana metoda sadrži određene nedostatke koje je potrebno uzeti u obzir:

- frekvencija uzorkovanja GPS podataka je niska zbog ograničenja u vidu prostora za pohranu podataka,
- geografske koordinate stop linija na prilazima raskrižju određena su približno korištenjem *Google maps* aplikacije,
- postoje značajne devijacije kod izračunatih vrijednosti,
- rezultati nisu uspoređeni s drugim istraživanjima jer takva gotovo ne postoje za promatrano područje,
- podaci za svako raskrižje se u aplikaciju unose ručno što značajno povećava ukupno vrijeme obrade podataka,
- zbog malog broja podataka u ranim jutarnjim i kasnim večernjim satima vrijednosti parametara na nekim raskrižjima ne odgovaraju stvarnom stanju na raskrižjima,
- vrijednosti parametara su grupirane za cijeli prilaz raskrižju, nisu razdvojene prema grupama trakova.

Bibliografija

[1] HCM2010, "Highway capacity manual", Transportation Research Board, National Research Council, teh. izv., prosinac 2010.

[2] B. K. Prasanta i K. V. Rao, "Defining level of service criteria of urban streets in Indian context", *European Transport*, br. 49, 2011, ISSN: 1825-3997.

[3] B. K. Prasanta i K. V. Rao, "FCM clustering using GPS data for defining level of service criteria of urban streets in Indian context", sv. 5, siječanj 2010.

[4] G. M. D'Este, R. Zito i M. Taylor, "Using GPS to Measure Traffic System Performance", sv. 14, str. 255–265, prosinac 2002.

[5] X. Zhan, R. Li i S. V. Ukkusuri, "Lane-based real-time queue length estimation using license plate recognition data", *Transportation Research Part C: Emerging Technologies*, sv. 57, str. 85–102, 2015, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2015.06.001. adresa: http://www.sciencedirect.com/science/article/pii/S0968090X15002168.

[6] X. J. Ban, P. Hao i Z. Sun, "Real time queue length estimation for signalized intersections using travel times from mobile sensors", *Transportation Research Part C: Emerging Technologies*, sv. 19, br. 6, str. 1133–1156, 2011, ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2011.01.002. adresa: http://www.sciencedirect.com/science/article/pii/S0968090X11000143.

[7] P. Hao i X. Ban, "Long queue estimation for signalized intersections using mobile data", *Transportation Research Part B: Methodological*, sv. 82, str. 54–73, 2015, ISSN: 0191-2615. DOI: https://doi.org/10.1016/j.trb.2015.10.002. adresa: http://www.sciencedirect.com/science/article/pii/S0191261515002118.

[8] I. Bošnjak i D. Badanjak, *Osnove prometnog inženjerstva*. Zagreb, Hrvatska: Fakultet prometnih znanosti, 2005.

[9] I. Dadić, G. Kos i M. Ševrović, *Terorija prometnog toka*. Zagreb, Hrvatska: Fakultet prometnih znanosti, 2014.

[10] T. V. Mathew, "Signalized Intersection Delay Models", *Transportation Systems Engineering*, E. Cascetta, ur., Springer, 2001, pogl. 35, str. 35.1–35.17.

[11] US Department of Transportation, "Traffic Detector Handbook, Third Edition Volume II", Federal Highway Administration, Research, Development, i Technology Turner-Fairbank Highway Research Center, teh. izv. FHWA-HRT-06-139, listopad 2006.

[12] T. Erdelić, S. Vrbančić i L. Rožić, "A model of speed profiles for urban road networks using G-means clustering", *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, svibanj 2015, str. 1081–1086. DOI: 10.1109/MIPRO.2015.7160436.

[13] T. Erdelić, M. Ravlić i T. Carić, "Travel time prediction using speed profiles for road network of Croatia", *2016 International Symposium ELMAR*, rujan 2016, str. 97–100. DOI: 10.1109/ELMAR.2016.7731763.

[14] T. Erdelić, T. Carić i M. Ravlić, "Predicting waiting times at intersections", *2017 International Symposium ELMAR*, rujan 2017, str. 31–34. DOI: 10.23919/ELMAR.2017.8124428.

[15] StackOverflow. (2018). StackOverflow Developer Survey 2018, adresa: https://insights.stackoverflow.com/survey/2018 (pogledano 20. 6. 2010).

[16] The Statistics Portal. (2018). Ranking of the most popular database management systems worldwide, as of February 2018, adresa: https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/ (pogledano 20. 6. 2010).

[17] A. Stojanović, "OSVRT NA NOSQL BAZE PODATAKA – ČETIRI OSNOVNE TEHNOLOGIJE", *POLYTECHNIC DESIGN*, sv. 4, br. 1, str. 44–53, 2016. DOI: 10.19279/TVZ.PD.2016-4-1-06. adresa: https://hrcak.srce.hr/192140.

[18] A. Jorgensen, J. Segarra, P. LeBlanc, J. Chinchilla i A. Nelson, *Microsoft SQL Server 2012 Bible*, 1st. Wiley Publishing, 2012, ISBN: 1118106873, 9781118106877.

[19] MongoDB. (2018). NoSql explaned, adresa: https://www.mongodb.com/nosql-explained (pogledano 2. 6. 2010).

[20] ——, (2018). Document database, adresa: https://www.mongodb.com/document-databases (pogledano 9. 5. 2010).

[21] J. Ko, M. Hunter i R. Guensler, "MEASURING CONTROL DELAY USING SECOND-BY-SECOND GPS SPEED DATA", *TRB 2007 Annual Meeting*, 2006. adresa: https://pdfs.semanticscholar.org/c56c/5971b3f7f4ffd544b3a513bcc90f035ba507.pdf.

[22] R. W. Sinnott, "Virtues of the haversine", *Sky and Telescope*, sv. 68, br. 2, str. 158–159, 1984.

[23] F. Ivis, "Calculating Geographic Distance: Concepts and Methods", *NESUG*, 2006. adresa: https://www.lexjansen.com/nesug/nesug06/dm/da15.pdf.

[24] M. Lutz, *Learning Python*. Gravenstein Highway North, Sebastopol, CA 95472: O'REILLY, 2013, ISBN: 978-1-449-35573-9.

[25] Python. (2018). Python download page, adresa: https://www.python.org/downloads/ (pogledano 9. 5. 2010).

[26] T. E. Oliphant, *Guide to NumPy*, 2nd. USA: CreateSpace Independent Publishing Platform, 2015, ISBN: 151730007X, 9781517300074.

[27] J. D. Hunter, "Matplotlib: A 2D graphics environment", *Computing In Science & Engineering*, sv. 9, br. 3, str. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot i E. Duchesnay, "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, sv. 12, str. 2825–2830, 2011.

[29] E. Jones, T. Oliphant, P. Peterson i dr., *SciPy: Open source scientific tools for Python*, [Online; accessed ¡today¿], 2001–. adresa: http://www.scipy.org/.

[30] S. Shalev-Shwartz i S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. The Edinburgh Building, Cambridge CB2 2RU, UK: Cambridge University Press, 2014, ISBN: 978-1-107-05713-5.

[31] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", *IBM Journal*, sv. 3, br. 3, str. 535–554, 1959. adresa: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.2254&rep=rep1&type=pdf.

[32] A. Smola i S. Vishwanathan, *INTRODUCTION TO MACHINE LEARNING*. The Edinburgh Building, Cambridge CB2 2RU, UK: Cambridge University Press, 2008, ISBN: 0 521 82583 0.

[33] C. C. Aggarwal i C. K. Reddy, *Data Clustering: Algorithms and Applications*, 1st. Chapman & Hall/CRC, 2013, ISBN: 1466558210, 9781466558212.

[34] K. Saurav. (2015). An Introduction to Clustering and different methods of clustering, adresa: `https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/` (pogledano 1. 7. 2010).

[35] D. Bošnjak, *Usporedba metoda za grupiranje tekstualnih dokumenata*. Završni rad, Fakultet elektrotehnike i računarstva, 2017.

[36] J. MacQueen, "Some methods for classification and analysis of multivariate observations", *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Berkeley, Calif.: University of California Press, 1967, str. 281–297. adresa: `https://projecteuclid.org/euclid.bsmsp/1200512992`.

[37] J. Wu, *Advances in K-means Clustering: A Data Mining Thinking*. Springer Publishing Company, Incorporated, 2012, ISBN: 3642298060, 9783642298066.

[38] D. Arthur i S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding", *Proc. of the Annu. ACM-SIAM Symp. on Discrete Algorithms*, sv. 8, siječanj 2007, str. 1027–1035.

[39] A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA, 2017, sv. 1.

Popis kratica

AVI	*Automatic Vehicle Identification* - sustav identifikacije vozila
BSON	*Binary JavaScript Object Notation* - tip zapisa
CD	*Control delay* - upravljačko vrijeme kašnjenja
CSV	*Comma Separated Values* - tip zapisa
FCD	*Floating Car Data* - podaci s pokretnih vozila
FFS	*Free Flow Speed* - brzina slobodnog toka
FFSP	*Free Flow Speed Percentage* - relativna brzina u odnosu na brzinu slobodnog toka
GPS	*Global Positioning System* - navigacijski sustav
HCM	*Highway Capacity Manual* - priručnik
LOS	*Level of service* - razina uslužnosti
JSON	*JavaScript Object Notation* - tip zapisa
NoSQL	ne relacijska baza podataka
SQL	*Structured Query Language* - upitni jezika (nekada označuje relacijske baze podataka)
VCR	*Volume to Capacity Ratio* - odnos volumena i kapaciteta

Popis slika

2.1 Izračun srednje vremenske brzine 7
2.2 Izračun srednje prostorne brzine 7
2.3 Fundamentalni dijagrami prometnog toka 10
2.4 Različita stanja prometnog sustava za iste vrijednosti protoka 11

3.1 Prikaz prikupljenih podataka na teritoriju Republike Hrvatske 15
3.2 CSV format podataka 16
3.3 Promatrano raskrižje opisano linkovima i rutama 17
3.4 Sezonalnost prometnog toka - profili brzina i repova čekanja 18
3.5 Vrijednosti profila brzina za različite dane u tjednu 18
3.6 Rezultati istraživanja o korištenju baza podataka, [15] 19
3.7 Grafički prikaz jedne kolekcije u MongoDB bazi podataka 23

4.1 Apsolutni i relativni profil brzine 32
4.2 Procjena CD parametra 33
4.3 Procjena udaljenosti od raskrižja 34

5.1 Razlika između interpreterkog i compilerskog jezika 37
5.2 Nadzirano učenje 39
5.3 Nenadzirano učenje 39
5.4 Regresija 41
5.5 Primjer Metode Koljena 44
5.6 GUI - Unos podataka za obradu 46

5.7 GUI - prikaz repova čekanja i profila brzina 47
5.8 GUI - strojno učenje 48

6.1 Repovi čekanja na analiziranim raskrižjima 51
6.2 Grupiranje raskrižja u *07:30* h 54
6.3 Grupiranje raskrižja u *11:30* h 55
6.4 Grupiranje raskrižja u *16:30* h 56

Popis tablica

3.1 Opis atributa 16
3.2 Usporedba relacijski i NoSQL baza podataka, [19] 21
3.3 Analogija između Microsoft SQL Server i MongoDB baze podataka 24

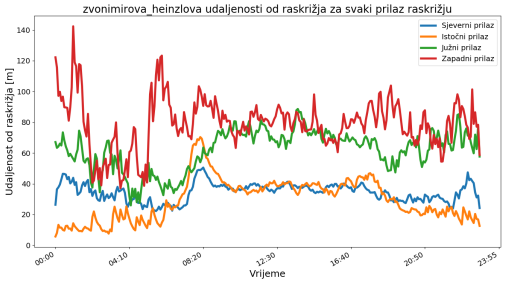
4.1 Vrijednosti postotnog pada brzine i razine uslužnosti, [1] 29
4.2 Vrijednosti upravljačkog vremena kašnjenja i razine uslužnosti, [1] 31

6.1 Broj prikupljenih podataka 50
6.2 Rezultati procjene t_{cd} i LOS 52

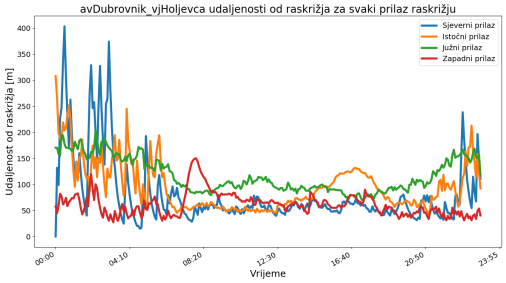
Popis kôdova

3.1 Prikaz dokumenta u BSON formatu 22
3.2 Izrada tablice u relacijskoj bazi podataka 24
3.3 Unos podataka u tablicu korištenjem SQL jezika 24
3.4 Unos dokumenta u kolekciju u MongoDB baze podataka 25
3.5 SQL upit 25
3.6 MongoDB upit 26
3.7 Ažuriranje podataka u relacijskoj bazi podataka 26
3.8 Ažuriranje podataka u MongoDB bazi podataka 26
3.9 Brisanje zapisa korištenjem SQL jezika 27
3.10 Brisanje zapisa u MongoDB bazi podataka 27

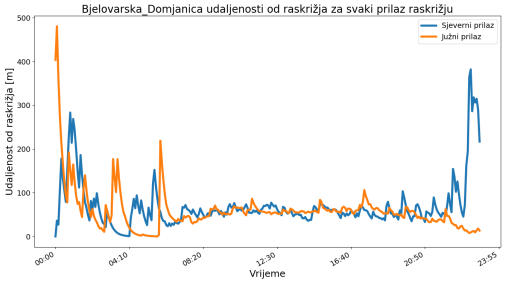
Prilog A: Rezultati procjene repa čekanja



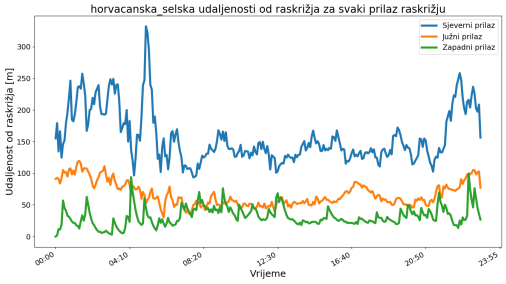
(a) Zvonimirova - Heinzlova



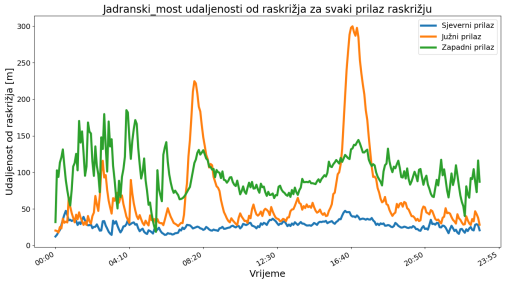
(b) Av. Dubrovnik - V. Holjevca



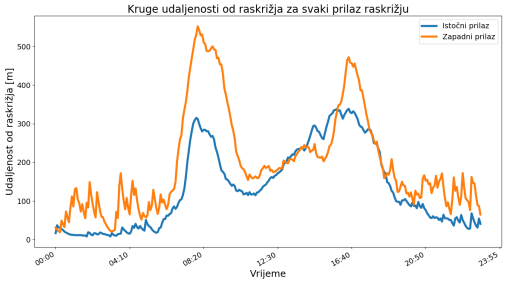
(c) Bjelovarska - Domjanića



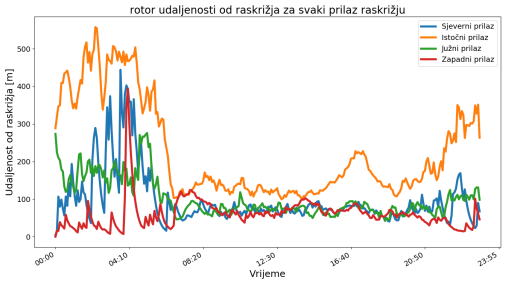
(d) Horvačanska - Selska



(e) Jadranski most - Savska



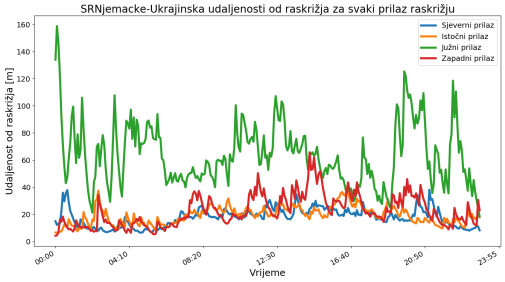
(f) Zagrebačka - Kruge



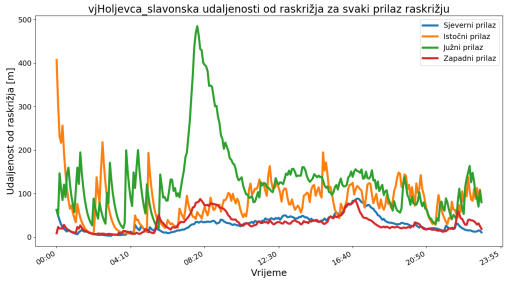
(g) Rotor



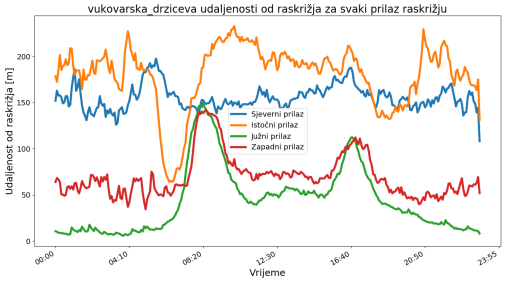
(h) Slavonska - Radnička



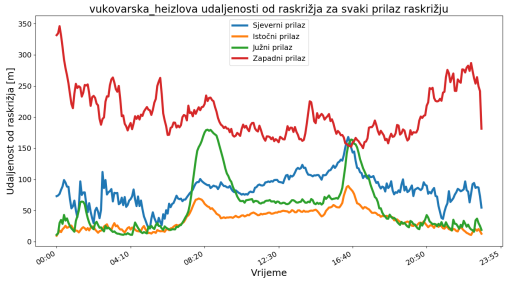
(i) SR Njemačke - Ukrajinska



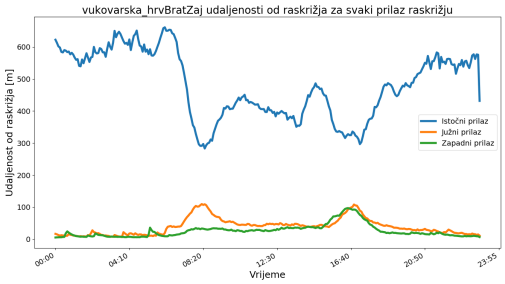
(j) V. Holjevca - Slavonska



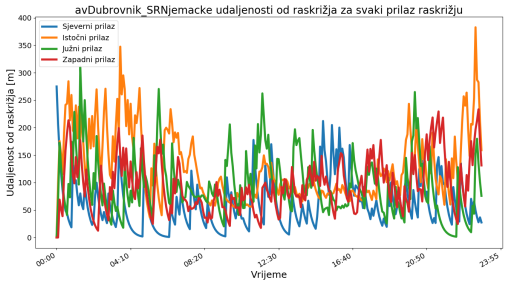
(k) Vukovarska - Držićeva



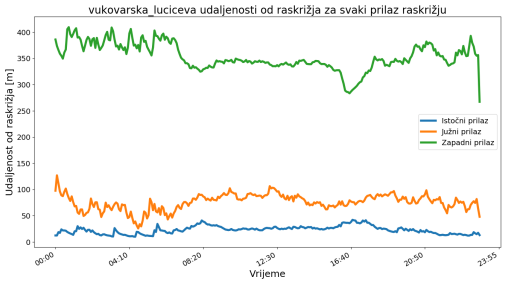
(l) Vukovarska - Heinzlova



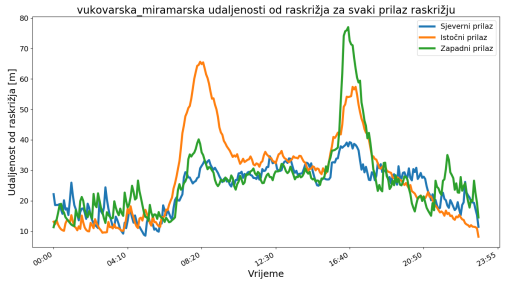
(m) Vukovarska - Hr.Brat. Zajednice



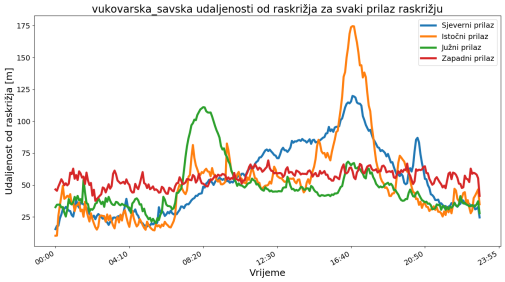
(n) Av. Dubrovnik - SR Njemačke



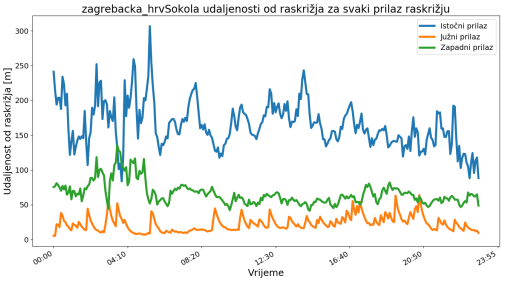
(o) Vukovarska - Lučićeva



(p) Vukovarska - Miramarska



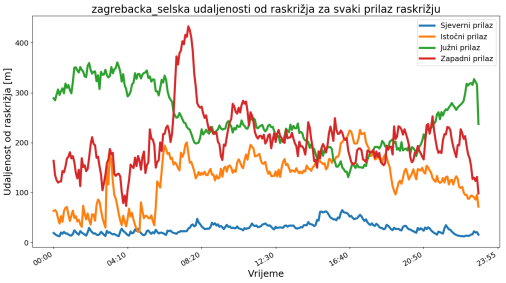
(q) Vukovarska - Savska



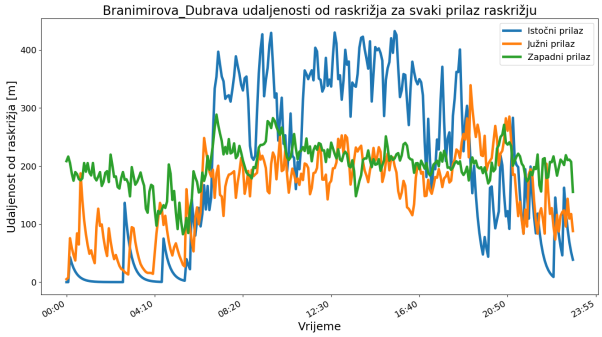
(r) Zagrebačka - HR Sokola



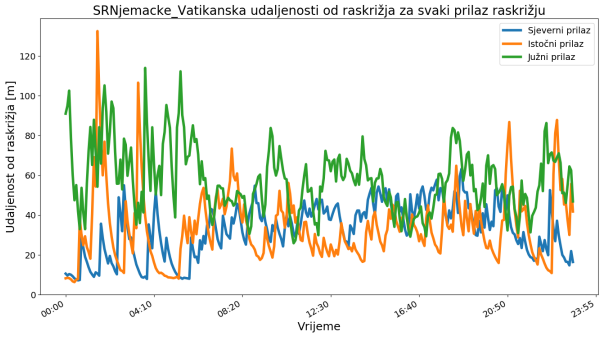
(s) Zagrebačka - Savska



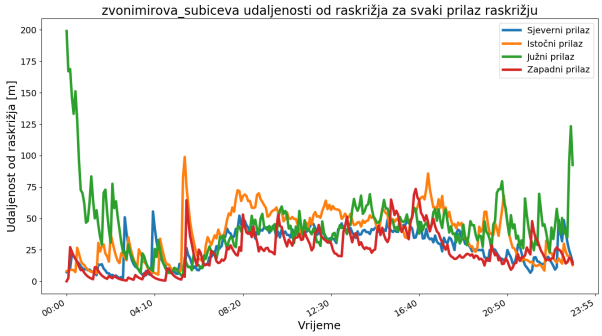
(t) Zagrebačka - Selska



(u) Branimirova - Dubrava



(v) SR Njemačke - Vatikanska



(w) Zvonimirova - Šubićeva

Prilog B: Rezultati procjene LOS

Raskrižje	$Prilaz$	Cijeli dan		Vršni sati		Ne-vršni sati	
		t_{cd}	LoS	t_{cd}	LoS	t_{cd}	LoS
SR Njemačke - Av. Dubrovnik	S	$64,85$	E	$39,82$	D	$35,00$	D
	I	$41,87$	D	$33,27$	C	$29,36$	C
	J	$41,08$	D	$38,58$	D	$27,07$	C
	Z	$69,87$	E	$55,67$	E	$40,13$	D
Vukovarska - Lučićeva	I	$51,99$	D	$59,66$	E	$39,48$	D
	J	$47,35$	D	$47,03$	D	$28,23$	C
	Z	$78,73$	E	$153,96$	F	$36,06$	D
SR Njemačke - Ukrajinska	S	$46,62$	D	$49,36$	D	$29,16$	C
	I	$45,72$	D	$48,35$	D	$33,87$	C
	J	$40,09$	D	$36,36$	D	$21,59$	C
	Z	$47,29$	D	$49,14$	D	$29,64$	C
Domjanića - Bjelovarska	S	$76,88$	E	$82,76$	F	$61,38$	E
	J	$34,02$	C	$33,71$	C	$18,76$	B
Horvačanska - Selska	S	$60,09$	E	$76,37$	E	$25,57$	C
	J	$52,80$	D	$62,76$	E	$22,60$	C
	Z	$46,92$	D	$55,81$	E	$7,54$	A
Rotor	S	$46,21$	D	$52,53$	D	$27,91$	C
	I	$89,46$	F	$112,20$	F	$61,08$	E
	J	$73,52$	E	$74,60$	E	$46,87$	D
	Z	$60,90$	E	$61,63$	E	$45,03$	D
Av. Dubrovnik - V. Holjevca	S	$46,33$	D	$48,33$	D	$35,30$	D
	I	$79,02$	E	$92,31$	F	$49,15$	D
	J	$38,67$	D	$40,44$	D	$27,78$	C
	Z	$81,10$	F	$86,41$	F	$49,94$	D

Raskrižje	$Prilaz$	Cijeli dan		Vršni sati		Ne-vršni sati	
		t_{cd}	LoS	t_{cd}	LoS	t_{cd}	LoS
Vukovarska - Heinzlova	S	$108,75$	F	$110,69$	F	$94,20$	F
	I	$76,00$	E	$79,04$	E	$57,27$	E
	J	$111,65$	F	$131,78$	F	$82,49$	F
	Z	$75,67$	E	$84,92$	F	$54,94$	D
Zagrebačka - Selska	S	$38,54$	D	$42,55$	D	$28,48$	C
	I	$65,99$	E	$86,15$	F	$38,19$	D
	J	$51,78$	D	$62,14$	E	$38,27$	D
	Z	$66,54$	E	$74,74$	E	$50,33$	D
Zagrebačka - Hrv. Sokola	I	$61,72$	E	$67,06$	E	$39,64$	D
	J	$56,59$	E	$59,75$	E	$37,48$	D
	Z	$53,13$	D	$56,10$	E	$32,46$	C
Slavonska - Radnička	S	$49,77$	D	$51,17$	D	$35,78$	D
	J	$62,04$	E	$64,04$	E	$38,26$	D
Vukovarska - Hrv. Brat. Zajednice	I	$96,57$	F	$104,66$	F	$75,11$	E
	J	$62,07$	E	$80,46$	F	$40,73$	D
	Z	$44,54$	D	$54,12$	D	$28,04$	C
Zvonimirova - Heinzlova	S	$62,20$	E	$67,43$	E	$40,81$	D
	I	$53,96$	D	$60,53$	E	$42,03$	D
	J	$45,66$	D	$49,23$	D	$25,62$	C
	Z	53.90	D	$58,40$	E	$32,63$	C
Vukovarska - Držićeva	S	98.31	F	106.05	F	58.06	E
	I	109.07	F	133.07	F	73.04	E
	J	90.05	F	119.81	F	51.51	D
	Z	106.41	F	143.95	F	59.28	E
Vukovarska - Savska	S	$92,56$	F	$106,21$	F	$76,28$	E
	I	$68,94$	E	$93,34$	F	$48,08$	D
	J	$70,31$	E	$83,87$	F	$46,36$	D
	Z	$59,07$	E	$62,54$	E	$45,34$	D
Zagrebačka - Savska	S	$57,61$	E	$63,46$	E	$41,20$	D
	I	$73,81$	E	$100,28$	F	$34,84$	C
	J	$64,60$	E	$81,93$	F	$41,73$	D
	Z	$71,42$	E	$79,184$	E	$54,12$	D

Raskrižje	$Prilaz$	Cijeli dan		Vršni sati		Ne-vršni sati	
		t_{cd}	LoS	t_{cd}	LoS	t_{cd}	LoS
Vukovarska - Miramarska	S	$58,99$	E	$63,42$	E	$40,16$	D
	I	$46,10$	D	$53,68$	D	$29,52$	C
	Z	$79,41$	E	$92,39$	F	$50,94$	D
Jadranski most - Savska	S	$44,82$	D	$49,40$	D	$34,30$	C
	J	$50,89$	D	$70,07$	E	$34,27$	C
	Z	$67,93$	E	$78,77$	E	$48,36$	D
Kruge - Zagrebačka	I	$99,65$	F	$107,18$	F	$89,63$	F
	Z	$110,30$	F	$135,12$	F	$86,84$	F
V.Holjevca - Slavonska	S	$41,35$	D	$51,49$	D	$26,23$	C
	I	$52,62$	D	$58,30$	E	$29,59$	C
	J	$61,18$	E	$73,91$	E	$41,30$	D
	Z	$65,16$	E	$79,84$	E	$36,25$	D
Branimirova - Dubrava	I	$49,05$	D	$52,97$	D	$28,54$	C
	J	$42,92$	D	$45,63$	D	$22,31$	C
	Z	$51,85$	D	$67,88$	E	$31,68$	C
SR Njemačke - Vatikanska	S	$43,79$	D	$47,29$	D	$28,96$	C
	I	$51,44$	D	$55,19$	E	$35,81$	D
	J	$45,99$	D	$51,26$	D	$29,71$	C
Zvonimirova - Šubićeva	S	$50,15$	D	$56,96$	E	$32,96$	C
	I	$66,96$	E	$82,72$	F	$49,93$	D
	J	$40,21$	D	$41,87$	D	$23,19$	C
	Z	$49,05$	D	$57,39$	E	$32,84$	C

Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu diplomskog rada pod naslovom **ANALIZA REPOVA ČEKANJA I RAZINE USLUŽNOSTI URBANIH PROMETNICA KORIŠTENJEM ALGORITAMA STROJNOG UČENJA I NOSQL BAZA PODATAKA** na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 09-07-18

(potpis)