

# Mogućnosti nadopune mjernih podataka prometnih parametara na osnovu povijesnih vrijednosti

---

**Petrač, Pavao**

**Undergraduate thesis / Završni rad**

**2015**

*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:119:816959>

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

*Download date / Datum preuzimanja:* **2024-07-23**



*Repository / Repozitorij:*

[Faculty of Transport and Traffic Sciences -  
Institutional Repository](#)



**SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI**

**Pavao Petrač**

**MOGUĆNOSTI NADOPUNE MJERNIH PODATAKA  
PROMETNIH PARAMETARA NA OSNOVU POVIJESNIH  
VRIJEDNOSTI**

**ZAVRŠNI RAD**

**Zagreb, 2015.**

Zagreb, 22. svibnja 2015.

Zavod: **Zavod za inteligentne transportne sustave**  
Predmet: **Automatsko upravljanje u prometu i transportu**

## **ZAVRŠNI ZADATAK br. 2272**

Pristupnik: **Pavao Petrač (0135230631)**  
Studij: **Inteligentni transportni sustavi i logistika**  
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Mogućnosti nadopune mjernih podataka prometnih parametara na osnovu povijesnih vrijednosti**

### Opis zadatka:

Nova upravljačka rješenja iz domene ITS-a za upravljanje cestovnom mrežom zahtijevaju kvalitetne mjerne podatke o stanju prometnice. Potrebni podaci se dobivaju korištenjem različitih osjetila za mjerenje prometnih parametara, no postoji problem da u slučaju kvara osjetila mjerni podaci postanu nedostupni. Takva situacija se rješava nadopunom podataka primjenom povijesnih mjernih podataka sa istog mjernog mjesta. U radu je potrebno opisati prometne parametre, napraviti pregled postojećih osjetila za mjerenje prometnih parametara cestovne prometnice, prikazati način arhiviranja mjernih podataka, implementirati osnovnu statističku analizu mjernih podataka te na osnovi statističke analize provjeriti točnost nadopune podataka.

Zadatak uručen pristupniku: 24. ožujka 2015.

Mentor:



doc. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za  
završni ispit:



Sveučilište u Zagrebu  
Fakultet prometnih znanosti

## **ZAVRŠNI RAD**

**MOGUĆNOSTI NADOPUNE MJERNIH PODATAKA  
PROMETNIH PARAMETARA NA OSNOVU  
POVIJESNIH VRIJEDNOSTI**

**POSSIBILITIES FOR COMPLEMENTATION OF  
MEASURED TRAFFIC PARAMETERS DATA BASED  
ON HISTORIC VALUES**

Mentor: doc. dr. sc. Edouard Ivanjko  
Student: Pavao Petrač, 0135230631

Zagreb, 2015.

# Predgovor

Zahvaljujem Mariju Krešiću, zamjeniku pučke pravobraniteljice s čijim uredom sam u suradnji pri izradi aplikacije za unos i sistematizaciju svih pismena koje ured zaprima i otprema u skladu s nadležnostima. Hvala gospodine Mario što Ste mi dali dovoljno slobodnog vremena za dovršiti ovaj završni rad.

Hvala gospodi iz tvrtke Led-elektronika d.o.o., Josipu Kovačeviću i Hrvoju Kaluđeru. Bez vas naslov, tema i realizacija mog rada ne bi bila moguća. Hvala vam na pruženim materijalima, opisu sustava i ponuđenom radnom odnosu. Posebno hvala Josipe na ispravljanju programskog kôda u jednom navratu. Također mi je ta vremenska ušteda puno značila.

Hvala profesoru Ognjenu Kuljači. Vaše međunarodno iskustvo, praktično iskustvo statističkih rješenja, poznavanje alata Matlaba i šalice razgovora neće biti zaboravljena.

Za kraj veliko hvala Edouardu Ivanjku, cijenjenom mentoru i profesoru, jer Ste trudom pokazali da Vam je stalo do Vaših studenta. Nadam se da ćemo jednoga dana ponovno surađivati.

## Sažetak

Nova upravljačka rješenja iz domene ITS-a za upravljanje cestovnom mrežom zahtijevaju kvalitetne mjerne podatke o stanju prometnice. Potrebni podaci se dobivaju korištenjem različitih osjetila za mjerenje prometnih parametara, no postoji problem da u slučaju kvara osjetila mjerni podaci postanu nedostupni. Takva situacija se rješava nadopunom podataka primjenom povijesnih mjernih podataka sa istog mjernog mjesta. U radu su opisani prometni parametri, napravljen je pregled postojećih osjetila za mjerenje prometnih parametara cestovne prometnice, prikazan je način arhiviranja mjernih podataka, implementirana je osnovna statistička analiza mjernih podataka te je na temelju statističke analize proračunata točnost nadopune podataka.

**Ključne riječi:** prometni parametri, senzorni uređaji, obrada podataka, modeliranje, statistička analiza, prikaz podataka.

## **Abstract**

New control solutions for traffic network management from the ITS domain demands high quality traffic data, collected directly from the road, or by using another type of approach. Data are collected by sensors created for vehicle counting. In a case of malfunction of such systems, there is a problem that data won't be collected, i.e. data can get lost. Such a situation is resolved by updating the lost data with systems that are using historical data to adopt the system performance. In this thesis traffic parameters are described, a short overview of sensors for traffic counters is made and there is an implementation of the basic statistical analysis which is used for replacement of lost measurement data.

**Keywords:** traffic parameters, sensors, data processing, modelling, statistical analysis, plotting the data.

# Sadržaj

1. Uvod	1
2. Prometni parametri cestovne prometnice	3
2.1. Protok i volumen	3
2.2. Brzina	4
2.3. Gustoća prometa	5
2.4. Prostorni i vremenski razmak između vozila	6
2.5. Relacija između osnovnih parametara	6
2.6. Prometni parametri kao pokazatelji kvalitete usluge	7
3. Osjetila za mjerenje prometnih podataka cestovne prometnice	9
4. Baza za arhiviranje mjerenja	13
4.1. Razumijevanje podataka	14
4.2. Priprema podataka	16
5. Statistička analiza mjernih podataka	18
5.1. Istraživanje	18
5.2. Projektiranje	22
6. Rezultat nadopune podataka	25
7. Zaključak	28
Literatura	29
Popis kratica	30
Popis tablica	31
Popis slika	32
Prilog A	33



# 1. Uvod

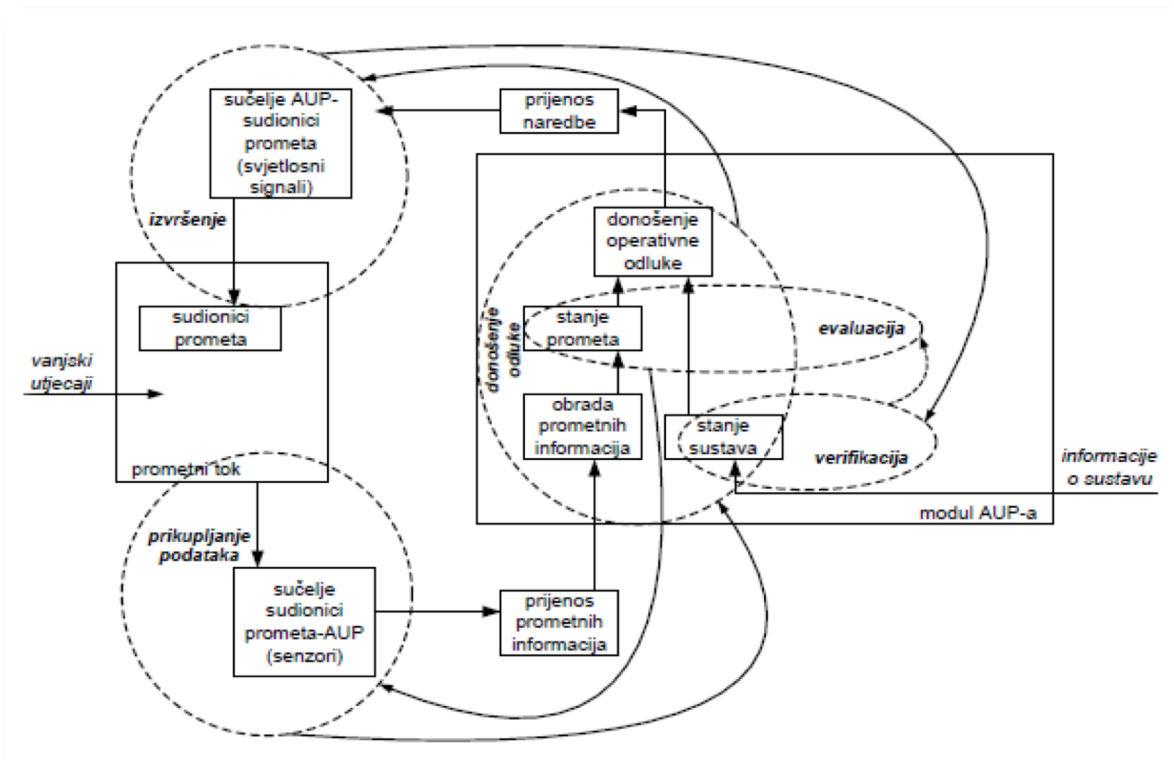
Za opis prometa na prometnicama svih vrsta, koriste se tri osnovne varijable: volumen (ili protok), brzina i gustoća. Na prometne parametre gleda se različito u neprekinutom i prekinutom prometnom toku. Pošto se u radu promatraju podaci prikupljeni na autocesti, gdje se uglavnom najviše vodi neprekinuti prometni tok, upravo se tu postavlja temelj ovog završnog rada.

Kako se površina raspoloživa za izgradnju novih prometnica u gradskim sredinama sukobljava s prostornim ograničenjima, a ujedno se i godišnje povećava broj osobnih vozila, kvaliteta prometnog toka može se zadržati, ili čak i povećati, prometnim upravljanjem. Iz tog pogleda počinje razvoj inteligentnih transportnih sustava (ITS) koji zahtjeva kvalitetne prometne podatke u stvarnom vremenu. Na te potrebe javljaju se razne organizacije i visoko-obrazovni programi koji nude svoje usluge na spomenutom području, te su obavezne prema klijentima izaći s dovoljno dobrim rješenjima kako bi relativno nova trgovina usluga bila prihvaćena.

U ovom radu modeliranjem se nastoje ukloniti pogreške do kojih je došlo tijekom rada senzora za mjerenje prometnih parametara cestovne prometnice. Mjerni uređaji su u više navrata gubili funkcionalnost. To ima za posljedicu da u određenim kraćim ili dužim vremenskim razdobljima nedostaju potrebna mjerenja prometnih parametara. Time prometni upravljački sustav gubi mogućnost reaktivnog upravljanja i može koristiti samo fiksne postavke. Kao primjer korišteni su prometni podaci dobiveni od hrvatske tvrtke Led elektronika d.o.o., a prikupljeni su na autocesti Ibrahim-Rugova u Republici Makedoniji.

Modeliranjem prometnog sustava vrijednosti prometnih parametara ulazne su veličine za automatsko upravljanje prometom, na osnovi kojih se izvedenim rješenjima upravljanja donosi odluka o djelovanju na sustav. Glavne funkcije AUP-a su: prikupljanje stvarno-vremenskih mjernih podataka, donošenje odluke, izvršenje upravljačke naredbe, verifikacija, evaluacija [8].

Ovaj rad pokriva samo dio ciklusa (slika 1.), automatskog upravljanja. Počevši od prijenosa prometnih informacija, zatim obrade prometnih informacija i prepoznavanja stanja prometa, na temelju čega se definira stanje sustava. Donošenje operativne odluke nije u domeni, kao niti utjecaj na sudionike prometa.



Slika 1. - Ciklus osnovnih funkcija AUP-a i tijek informacija [4]

Prema poglavljima koncept rada počinje od poglavlja „Prometni parametri cestovne prometnice“, drugog poglavlja, u kojem se opisuju utjecaji na stanje prometa. Popisani su neki od najvažnijih parametara i objašnjava se njihovo značenje za promet. Kako bi se prometni parametri uopće mogli mjeriti, potrebno je implementirati „Osjetila za mjerenje prometnih podataka cestovne prometnice“, što ujedno i definira treće poglavlje. U tom poglavlju govori se općenito o sensorima, sensorima usmjerenim na promet, te su navedeni najčešći senzori koji se dotiču mjerenja prometnih podataka cestovne prometnice.

Dobivene prometne podatke potrebno je sigurno pohraniti, odnosno arhivirati u bazu podataka. Iz tog razloga otvara se četvrto poglavlje, nazvano „Baza za arhiviranje mjerenja“. Naime, riječ je o relacijskoj bazi podataka, te modelu prema kojem su inženjeri Led-elektronike arhivirali podatke.

Podacima iz baze pristupa se računski, te je u petom poglavlju – „Statistička analiza mjernih podataka“ učinjeno isto nad arhiviranim mjerenjima. Temeljem statističke analize stvoren je dnevni model količine vozila koji se koristi za nadopunu izgubljenih podataka u jednom danu. Stvaranje tog modela opisano je u šestom poglavlju pod naslovom „Rezultat nadopune podataka“, gdje je prikazana preciznost statističke analize. Rad završava zaključkom i prijedlozima za nastavak rada.

## 2. Prometni parametri cestovne prometnice

Prometnim parametrima opisuju se prometni tokovi i zakonitosti kretanja motornih vozila u prometnim tokovima na cestovnim prometnicama. Ti parametri se još zovu osnovni parametri prometnog toka ili osnovne veličine prometnog toka. U prometnom toku djeluje međusobna interakcija vozila, što daje sustavu nelinearan oblik, a predstavlja osnovnu razliku u uvjetima kretanja vozila u prometnim tokovima u odnosu na uvjete kretanja pojedinačnih vozila.

### 2.1 Protok i volumen

Protok i volumen dva su temeljna parametra kojima se kvantificira količina prometa koji protječe kroz jednu točku ili ogranak na cesti u zadanom vremenskom intervalu. Ti termini su definirani kao [6]:

- Volumen – ukupni broj vozila koji prođu neku određenu dužinu prometnice u nekom vremenskom intervalu. Može biti izražen na godišnjoj i dnevnoj razini, te po satima ili minutama;
- Protok – ekvivalentno vremenskoj stopi u kojoj vozila prolaze točku ili dužinu prometnice u intervalu manjem od jednoga sata, najčešće 15min.

Navedeni parametri kvantificiraju prometnu potražnju, koja predstavlja broj vozila čija je namjera koristiti prometnu infrastrukturu u nekom vremenskom periodu. Naravno, postoje limiti kod ostvarivanja namjera, kao na primjer potrošnja goriva, zakrčenost prometnice, zakonske regulative, itd.

Razlikovanje volumena i protoka je bitno. Znači, volumen je broj promatranih ili predviđenih vozila koji prolaze neku točku u nekom vremenskom intervalu, dok je protok broj vozila promatranih u nekom periodu, ali taj broj vozila se dijeli s vremenom promatranja. Kao primjer, volumen od 100 vozila promatranih u 15 minutnom intervalu podrazumijeva protok od  $100 \text{ vozila} / 0,25 \text{ h}$  ili jednostavnije 400 vozila po satu.

Volumen i protok mogu biti ilustrirani preko četiri uzastopnih promatranja 15-min intervala. Primjerice, u prvom intervalu prošlo je 1000 vozila, drugom 1200, 1100 i 1000. Ukupan volumen jednoga sata iznosi 4300 vozila, dok protok varira u svakom od

promatranih intervala. Tako je maksimalni protok vozila 1200 voz / 0,25 h, ili 4800 voz/h. Iz toga je jasno vidljiva razlika navedenih parametara.

Razmatranje vršnih vrijednosti protoka je iznimno bitna kod analiza kapaciteta prometnice. Ukoliko je izmjeren kapacitet segmenta prometnice po prilici 4500 voz/h, mjera će biti prekoračena iako je volumen prometnice, odnosno broj vozila koji su prošli u sat vremena 4300. To postaje velik problem, jer u slučaju zastoja zagušenje može potrajati i do nekoliko sati.

Maksimalni protok i volumen u satu kreiraju faktor vršnog sata (engl. peak-hour factor - PHF), što je postotak ukupnog volumena u satu i vršne vrijednosti protoka. Izražava se formulom [7]:

$$PHF = \frac{\text{volumen u satu}}{\text{vršni protok (u satu)}} \quad (1)$$

Kada je PHF poznat može se jednostavno prebaciti iz volumen vršnog sata u maksimalni protok.

## 2.2 Brzina

Dok prometni volumen predstavlja metodu kvantificiranja kapacitetnih vrijednosti, brzina (ili recipročna vrijednost – vrijeme putovanja) je važna mjera kvalitete prometne usluge. Mjera efektivnosti definirana stupnjem usluge za puno tipova prometne infrastrukture, kao što su ruralne dvotračne autoceste, brze ceste, gradske ceste, itd.

Brzina je definirana kao stupanj pomaka izražen preko prijedenoj udaljenost u jedinici vremena. Generalno se izražava u kilometrima po satu (km/h). U karakterizaciji brzine prometnog toka, mora se koristiti reprezentativna vrijednost, zbog široke distribucije individualnih brzina u promatranom prometnom toku. U dobivenim podacima postoji brzina kojom je vozilo prošlo preko senzora, ali se koristi srednja vrijednost brzina prikupljenih na svim sensorima jedne rute, jer je to statistički najpreciznija mjera u relaciji s ostalim parametrima. Srednja brzina putovanja se računa na način da se podijeli ukupna duljina prometnice, ili dijela prometnice na kojoj se mjeri srednje vrijeme putovanja vozila. Ukoliko su vremena putovanja  $t_1, t_2, t_3, \dots, t_n$  (u satima) izračuni za  $n$  vozila koja protječu segmentom prometnice duljine  $L$  prema formuli:

$$v = \frac{nL}{\sum_{i=1}^n t_i} = \frac{L}{\frac{1}{n} \sum_{i=1}^n t_i}, \quad (2)$$

gdje su:

$v$  = prosječna brzina putovanja (km / h);

$L$  = duljina segmenta prometnice (km);

$t_i$  = vrijeme putovanja  $i$ -tog vozila (h);

$n$  = broj izmjerenih vremena putovanja;

$t_a = \frac{1}{n} \sum_{i=1}^n t_i$  = prosječno vrijeme putovanja po prometnici duljine  $L$  (h).

Izračun vremena putovanja sadrži i vrijeme izgubljeno u prometnim zastojkima, prekidima, itd. Za izračun razine usluge (engl. level of service - LoS), uzima se navedena prosječna brzina putovanja.

## 2.3 Gustoća prometa

Gustoća je broj vozila koji se nalaze na nekoj duljini prometnice u trenutnom vremenu. Gustoća je srednja vrijednost broja vozila u zadanom vremenu. Često se izražava kao broj vozila po kilometru (voz / km).

Direktno mjerenje gustoće na nekom području je zahtjevno, jer u biti zahtijeva snimanje, brojanje, ili promatranje duž cijele interesne površine. No gustoća, dakako, može biti izračunata preko prosječne brzine putovanja i protoka, koji mogu biti izmjereni puno lakše. Formula koja se koristi pri nezasićenim prometnim uvjetima:

$$D = \frac{q}{v}, \quad (3)$$

gdje je:

$q$  = protok vozila (voz / h);

$v$  = prosječna brzina putovanja (km / h);

$D$  = gustoća (voz / km).

Segment prometnice s protokom od 1000 voz/h i prosječnom brzinom od 50 km/h će imati gustoću prometa 20 voz/km. Zaključujemo da je gustoća kritični parametar

infrastrukture koja nudi neprekinuti prometni tok, jer karakterizira kvalitetu prometnih operacija. Opisuje međusobnu blizinu vozila što utječe na slobodu upravljanja vozila unutar nekog prometnog toka.

## 2.4 Prostorni i vremenski razmak između vozila

Prostorni razmak između dva slijedna vozila u prometnom toku je geometrijska udaljenost iste točke dva promatrana vozila (primjerice prednji branik, stražnja osovinica), dok je vremenska udaljenost vrijeme koje prođe dok dva vozila u ne prođu točku na prometnici. Pri vremenskoj udaljenosti, također je važno da točka koju se gleda na oba vozila bude identična.

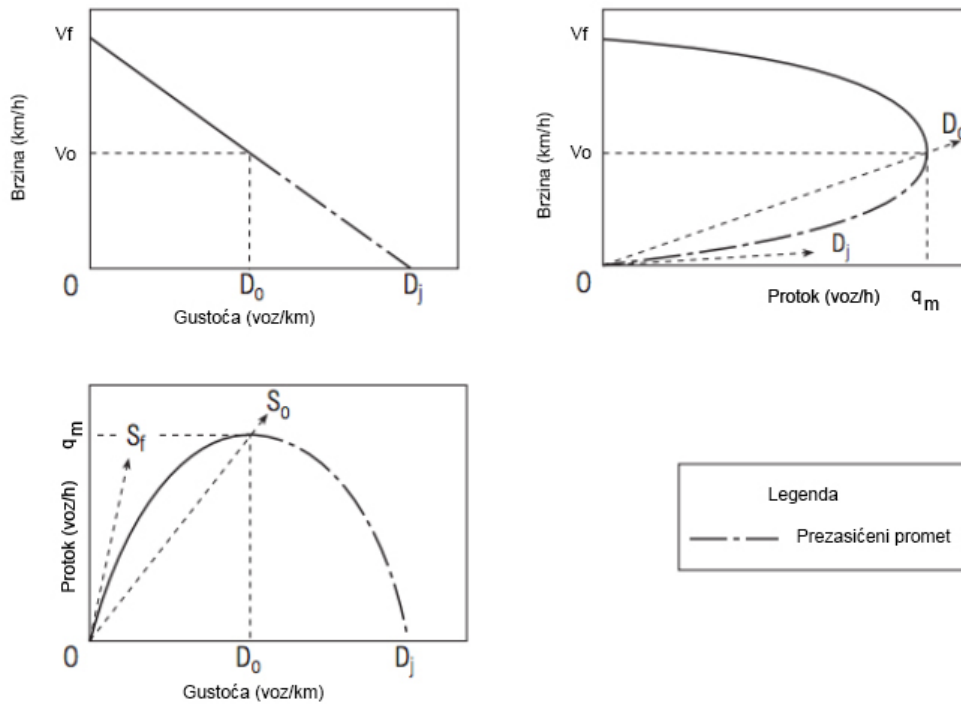
Budući da su to parametri koje dobivamo gledajući individualna vozila, može se reći da su to mikroskopski parametri. Prostorni i vremenski razmak između dva slijedna vozila uglavnom su unutar određenog spektra vrijednosti koje mogu poprimiti i generalno su povezani s brzinom prometnog toka i prevladavajućih uvjeta. Uglavnom, ti mikroskopski parametri vezani su s makroskopskim parametrima, odnosno gustoćom i protokom.

Prostorni razmak mjeri se u metrima. Može biti izmjereno direktno mjerenjem udaljenosti između dvije zajedničke točke u nekom vremenskom trenutku. Generalno, to zahtjeva kompleksne tehnike zračnih fotografiranja, tako da se prostorni razmak najčešće računa preko drugih mjerenja. Vremenski razmak, s druge strane, vrlo je lako izračunati štopericom, promatrajući vozila kako prolaze neku točku na prometnici.

## 2.5 Relacija između osnovnih parametara

Formula 3 definira osnovnu relaciju između tri parametra, opisujući neprekinuti prometni tok. Iako jednačina  $q = D \cdot v$  matematički dozvoljava da se tok vozila događa u beskonačnom broju kombinacija brzine i gustoće, postoje ograničenja koja variraju ovisno o uvjetima lokacije.

Slika 1 prikazuje generaliziranu reprezentaciju tih veza, koje su osnova za analize kapaciteta prometnica. Funkcija protoka i gustoće je postavljena točno ispod funkcije brzine i gustoće, zbog zajedničkog horizontalnog obilježja, dok je funkcija brzine i protoka postavljena desno od funkcije brzine i gustoće, ističući njihovu vertikalnu relaciju.



Slika 2. - Generalizirana veza između brzine, gustoće i protoka

Oblici ovih funkcija ovise o stanju ceste i pretežitom prometu koji se njome odvija. Iako su prikazane funkcije kontinuirane, gotovo je nevjerojatno da bi se puni spektar rješenja mogao primijeniti na svakoj dostupnoj lokaciji. Prikupljeni podaci obično pokazuju prekinuća, koja ne mogu biti vidljiva na navedenim funkcijama.

Funkcije pokazuju nekoliko kritičnih točaka. Prva je kada je protok jednak nuli. Što se događa kada nema vozila ili kada je gustoća dosegla takav stupanj da nema prometa. Druga je točka na sredini, gdje dinamičnost prometnog toka proizvodi maksimalan efekt.

## 2.6 Prometni parametri kao pokazatelji kvalitete usluge

Promjenu vrijednosti određene skupine prometnih parametara sudionici u prometu izravno doživljavaju kao promjenu kvalitete usluge. Tim parametrima može se izraziti kvaliteta usluge koju nudi prometni sustav. U pokazatelje kvalitete usluge mogu se ubrojiti: trenutna brzina, srednja vremenska brzina, srednja prostorna brzina, gustoća, zauzeće, kašnjenja i zaustavljanja.

Tablica 1. - Kvaliteta usluge za autoceste s više prometnih traka [4]

Stupanj usluge	Vrsta toka	Gustoća (vozila/trakumilji)	Opis stupnja usluge
A	slobodan tok	0-11	Individualni vozači neometani su od ostalih sudionika prometa. Velika sloboda izbora brzine i manevriranja u prometnom toku.
B	stabilan tok	12-18	Sloboda izbora brzine još uvijek je velika, ali postoji malo ograničenje slobode manevriranja u prometnom toku. Prisutnost ostalih sudionika počinje utjecati na ponašanje pojedinih vozača.
C	stabilan tok	19-26	Ponašanje vozača znatno je uvjetovano interakcijom s ostalim sudionicima prometnog toka. Izbor brzine ograničen je prisutnošću ostalih vozila. Manevriranje zahtijeva povećanu budnost i oprez vozača.
D	stabilan tok visoke gustoće	26-35	Brzina i sloboda manevriranja su znatno ograničeni. Vozači doživljavaju taj stupanj kao neugodnu i zamornu vožnju.
E	nestabilan zasićeni tok	35-45	Sloboda manevriranja je ekstremno ograničena. Kako je protok u blizini kapaciteta, brzina je mala, ali relativno jednolična.
F	zagušenje	45-	Formiraju se repovi.

Bitan pokazatelj kvalitete odvijanja prometa su vremenski parametri tj. kašnjenja, koja pokazuju trajanje pojedinih situacija koje se događaju u realnom prometnom toku. Mogu se izražavati apsolutno u vremenskim jedinicama ili relativno kao odnos potrebnog vremena u realnim uvjetima i teorijskog vremena koje bi bilo potrebno u uvjetima slobodnog toka.

Podaci o kašnjenju koriste se za procjenu koštanja u smislu potrošenoga goriva i izgubljenog vremena. Nadalje, kašnjenje je važan pokazatelj prometne zagušenosti, pogotovo kašnjenje zaustavljenog vozila. Može se iskoristiti kao osnova za procjenu utjecaja na okoliš u smislu ispušnih plinova i buke. Nezaobilazan je pokazatelj efikasnosti, prilikom evaluacije strategije upravljanja raskrižjem.



### 3. Osjetila za mjerenje prometnih podataka cestovne prometnice

Senzori su fundamentalne komponente kojima se svaka okolina može učiniti pametnom. S obzirom na aplikaciju, različiti senzori su potrebni kako bi se zadovoljili specifični ciljevi, odnosno potrebe. Senzori su važni u nadgledanju, prikupljanju podataka, njihovoj obradi brzim računalnim sustavima i različite aktivnosti mogu biti provedene s obzirom na prikupljene podatke.

Pri odabiru senzora postoji više kriterija koje je potrebno razmatrati pri izradi aplikacija. Preciznost je jedan od najvažnijih parametara za sve senzore i trebao bi biti što bliže 100% stvarne vrijednosti. Naravno, ukoliko je moguće da bude upravo prezentacija realnog sustava kojemu je namijenjen. Realno, tvrtke koje dizajniraju arhitekturu sustava, te navode performanse uzimaju u obzir slijedeće varijable:

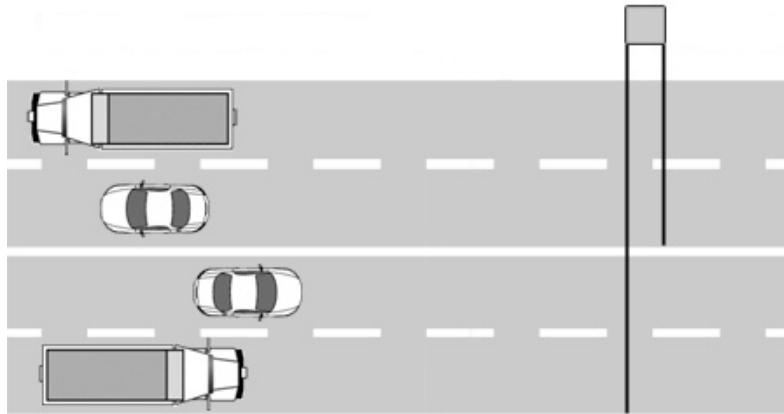
- a. Cijenu senzora, njegove implementacije i održavanja (cijena senzora je općenito mala sama po sebi, ali cijena instrumentacije i elektroničkih krugova za definiciju i procesuiranje signala je velika);
- b. preciznost interpretacije realnog sustava u računalni;
- c. korisničke potrebe, odnosno zahtjeve na koje aplikacija mora odgovarati;
- d. osjetljivost kao parametar koji pokazuje kako se senzor ponaša s obzirom na promjene u vremenu i vremenskim uvjetima.

Bolje razumijevanje djelovanja na prometnu mrežu je ključni faktor za poboljšanje putovanja i prometnog modela. Iz tog razloga nastaje potreba za primjenu osjetila za mjerenje parametara cestovne prometnice. Osim pouzdane i efektivne detekcije vozila i sustava praćenja, zadatak je omogućiti jednostavnu instalaciju i održavanje sa sigurnim i vremenski što kraćim prekidima prometa.

U nastavku su neki od popularnijih osjetila za mjerenje prometnih podataka cestovne prometnice:

1. **Pneumatske cijevi** – Gumene cijevi se postavljaju na cesti, te prolaskom vozila stvara se mehanička sila koja pobuđuje električku aktivnost. Ta aktivnost bilježi se na brojaču koji je smješten uz cestu.

Prednosti pneumatskih cijevi su jednostavnost instalacije, niska cijena, te lagana prenosivost. Glavni nedostatak ove tehnologije je ograničena pokrivenost ceste, smanjena učinkovitost pri vremenskim i temperaturnim promjenama uvjeta, te ima kratki vijek trajanja.

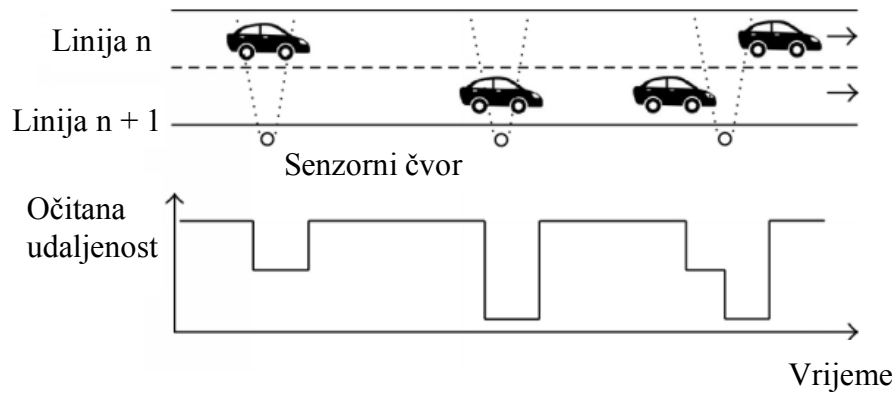


Slika 3. - Postavljanje pneumatskih cijevi na prometnicu

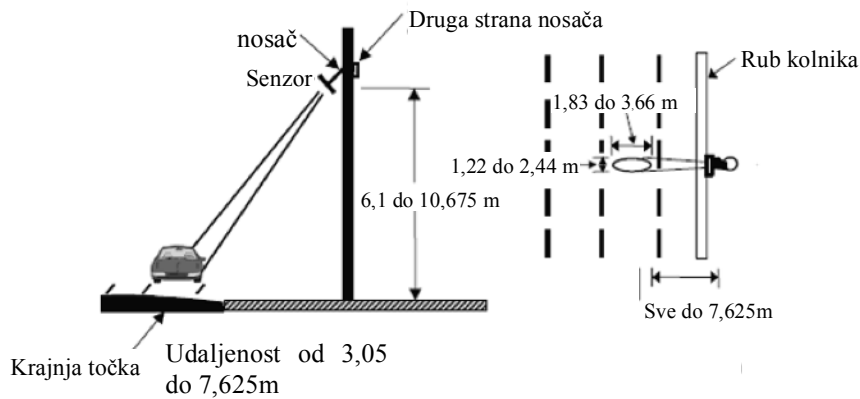
2. **Akustični / ultrazvučni senzori** – Primopredajni uređaj stvara zvučni val koji putuje do prepreke, od koje se odbija i vraća nazad, princip akustičnih / ultrazvučnih senzora je da se mehanička energija pretvara u električnu energiju. Sustav je jeftin i zahtjeva jednostavno sklopovlje. Ovaj sustav se može koristiti za mjerenje težine i brzine. Empirijski se dolazi do tipa vozila.

Glavni nedostaci su velika nepreciznost zbog oblika vozila, odnosno načina na koji se zvuk odbija od vozila i kratak domet zvučnog signala koji prepoznaje vozila. Aerodinamika karoserije vozila također ovisi na putanju odbijenih valova, te vremenski uvjeti.

S obzirom na udaljenost senzora od vozila, odnosno na prometnu traku na kojoj se vozilo nalazi, senzor jače ili slabije očitava promjenu u sustavu. Tu promjenu u stanju sustava računalo bilježi i određuje da se radi o vozilu na n traci. Prikaz takvog djelovanja vidljiv je na Slici 2.

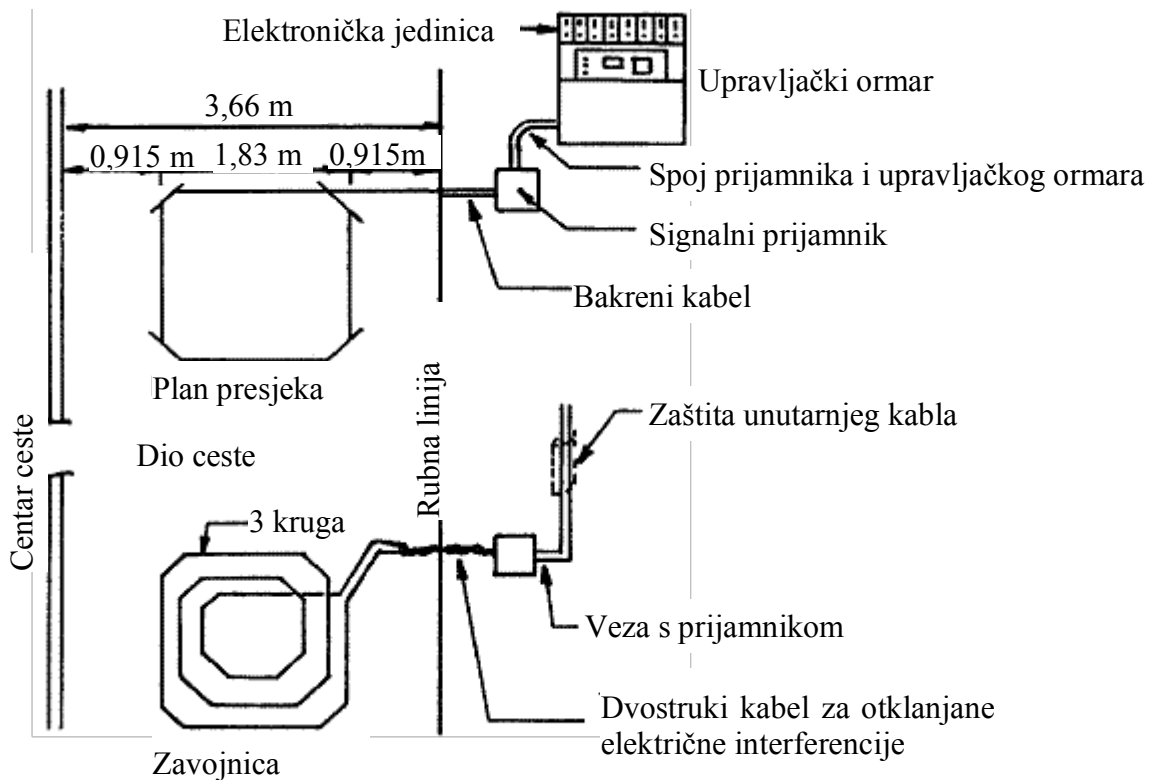


Slika 4. - Detekcija vozila koristeći ultrazvučne senzore [13]



Slika 5. – Prikaz postavljanja ultrazvučnog senzora [2]

3. **Induktivna petlja** – Radi tako da je zavojnica postavljena ispod površine ceste u kojoj se inducira napon prilikom prolaska vozila. Radi gotovo u svim uvjetima, ima kratki vijek trajanja, može biti oštećena od strane teških vozila, ali ne utječe na lošim vremenskim uvjetima. Skupa je za instalaciju i ne radi ako ima snijega ili leda na cesti.

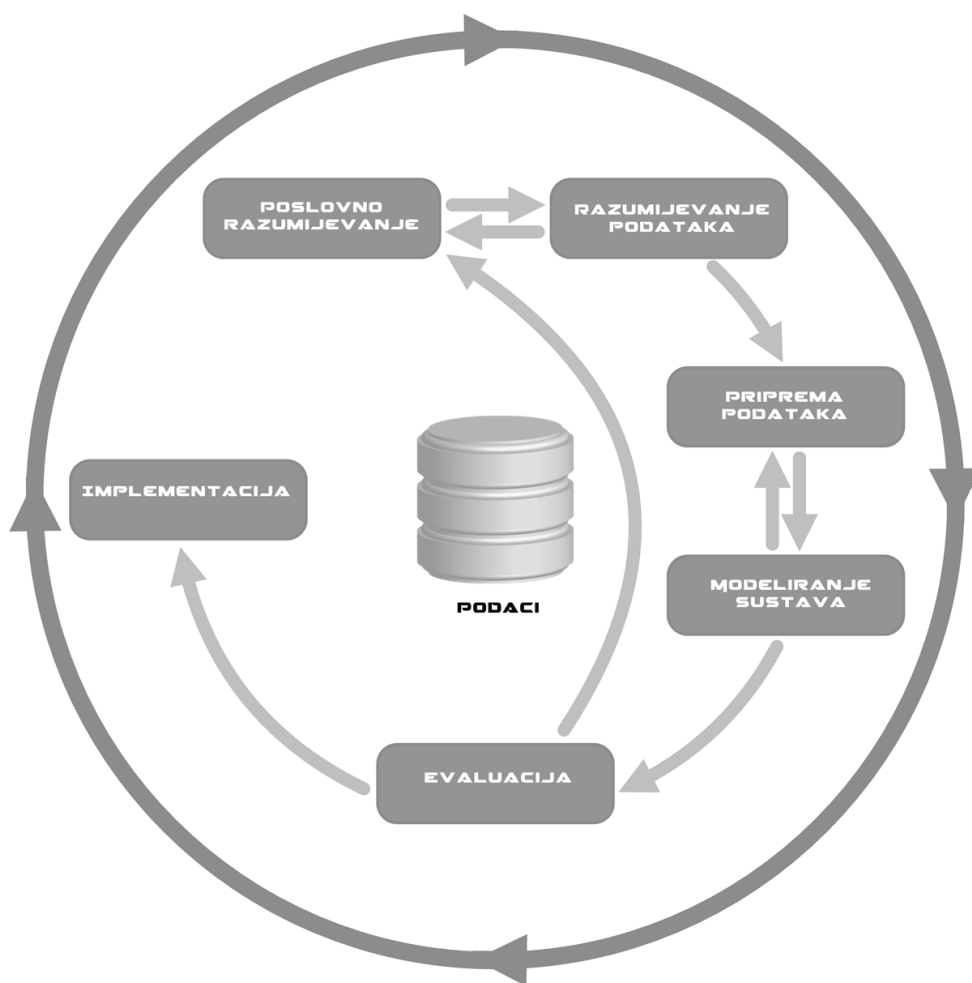


Slika 6. – Model slaganja induktivnih petlji na cesti [2]

4. **Pasivni i aktivni infracrveni senzor** - Prisutnost, brzina i tip vozila su otkrivene na temelju infracrvene energije koja zrači iz područja detekcije. Glavni nedostaci su performanse tijekom nevremena, i ograničena pokrivenost prometnih trakova.
5. **Mikrovalni radari** - Mjerni sustav sastoji se od odašiljača (*transmitter*), antene, puta kroz koji val putuje do reflektora (*transmission path*) te prijemnika (*receiver*). Sustav mjeri vrijeme od trenutka emitiranja signala sa odašiljača do trenutka povratka signala u prijemnik. Udaljenost antene od objekta od kojeg se signal reflektira (*transmission path*) proračunati će se na osnovu izmjerenog vremena proleta i poznate brzine prostiranja mikrovalnog snopa.
6. **Ultrazvučni i pasivni akustični senzori** - Ovi uređaji emitiraju zvuk valova za otkrivanje vozila mjerenjem vremena za signal za povratak na uređaj. Ultrazvučni senzori su postavljeni preko traka, a može biti pod utjecajem temperature ili lošeg vremena. Pasivni akustični uređaji postavljeni su uz cestu i može prikupiti broj vozila, brzinu i klasifikacija podataka. Oni također mogu biti pogođeni lošim vremenskim uvjetima (npr. niske temperature, snijeg).
7. **Obrada video snimke (video nadzor) prometa** - Video kamere broje rekord vozila, tipa i brzine pomoću različitih video tehnika, npr. izlet linije i praćenje sustava može biti osjetljiv na meteorološkim uvjetima.

## 4. Baza za arhiviranje mjerenja

Obrada podataka obuhvaća ne samo brožčana izračunavanja već i postupke kao što su klasifikacija podataka ili njihovo premještanje s jednog mjesta na drugo općenito se pretpostavlja da takve operacije obavlja neka vrsta stroja, bez obzira na to što neki postupci mogu biti obavljani ručno. Na slici 5. prikazan je model pristupa obradi podataka na kružni način.



Slika 7. - Prikaz procesa obrade (rudarenja) podataka [12]

Poslovno razumijevanje može se usporediti s korisničkim potrebama, odnosno zahtjevima. Vjeruje se da je ovaj tip obrade nastao iz umjetne inteligencije i područja algoritma pretraživanja.

## 4.1 Razumijevanje podataka

Podloga na koju se potrebno vezati su skladišta podataka. Takva arhitektura naziva se bazom podataka. Baza podataka je skup povezanih podataka, pohranjenih zajedno bez štetne ili nepotrebne (nekontrolirane) zalihosti (redundancije), s ciljem da ih koriste različite aplikacije. Podaci su pohranjeni u obliku neovisnom od programa koji ih koristi. Unos, izmjena i dohvat podataka obavlja se isključivo kroz zajedničko i kontrolirano sučelje.

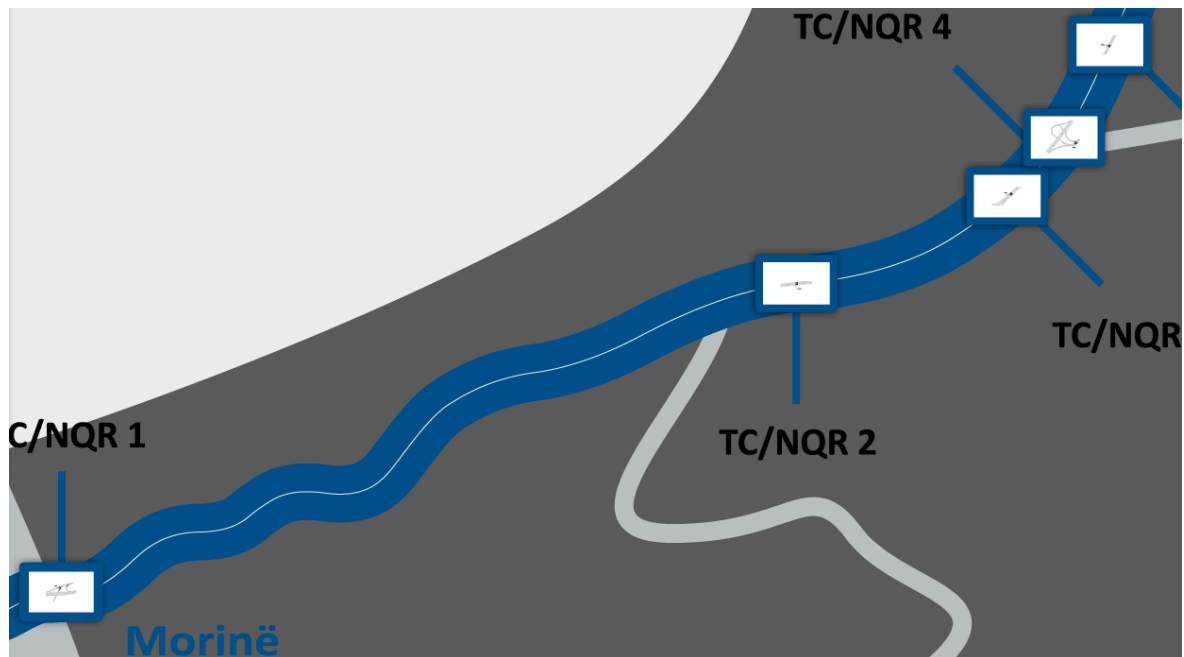
Programski sustav koji omogućava upravljanje bazom podataka je **sustav za upravljanje bazama podataka**, SUBP (**DBMS** – *Database Management System*). Podatke sa senzora potrebno je preuzeti i pohraniti. Za tu vrstu aktivnosti koristi se navedeni DBMS na serveru koji podatke pohranjuje na tvrdim diskovima. S obzirom na količinu podataka koji se zapisuju u dugogodišnjim vremenskim razmacima potrebno je racionalno dimenzionirati sustav, te ih podatke grupirati prema zajedničkim značajkama. Tako bi te operacije bile poželjne i prema modelu prikazanom u tablici:

Tablica 2 – Model zapisa baze podataka dobiven od tvrtke Led-elektronika

DataSource Name	DataKeyNu	Time	MS	Value	Type
TC/NQR 2	1	15:34:43.000	4	111	Car with trailer
TC/NQR 5	3	15:34:42.000	3	121	Car
TC/NQR 1	4	15:34:42.000	3	125	Car
TC/NQR 1	2	15:34:40.000	14	95	Bus

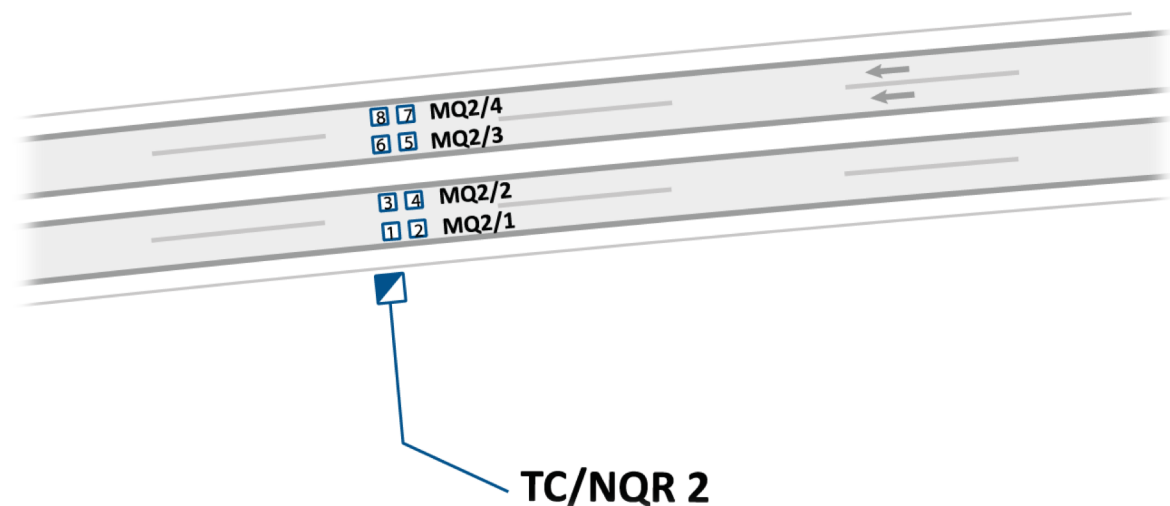
Vidljivo je da prema modelu Led-elektronike imamo veliku redundanciju podataka, što obradu čini nešto procesorski zahtjevnijom. Podaci su bili izvađeni iz „*Crystal report*“ alata za kreiranje izvješća. Zbog velike količine podataka, pristup podacima bio isključivo programerskim putem. Ukupno je postavljeno 26 automatskih brojala prometa sa središnjim sustavom na Ibrahim Rugova Highway.

Radi razumijevanja podataka potrebno je znati što predstavlja koja oznaka. Tako prvi stupac (*DataSource*) predstavlja o kojem djelu prometnice se radi, a to se može predočiti po sljedećoj slici:



Slika 8. - Vektorska slika prometnice u Kosovu koju je prema projektu za Bechtel Enka GP odradila tvrtka Led-elektronika d.o.o.

Sljedeća slika prikazuje detaljnije jedan segment brojila. Iz slike se dobiva ideja o drugom stupcu (*DateKeyNumber*) koji vidljivo označuje kolničku traku.



Slika 9. - Vektorska slika jednog segmenta uređaja

## 4.2 Priprema podataka

Prvi korak u pripremi je modeliranje sustava u svrhu čitanja podataka. *BufferedReader* je prespor za čitanje, zbog njegove korelacije s operativnim sustavom. Prema istraživanju optimalnog rješenja za obradu velikih tekstualnih zapisa odabrana je MMIO (Memory Mapped IO) metoda. Sustav je osmišljen da cijeli niz binarnog zapisa s tvrdog diska učita u memoriju. Iz razloga da se izbjegne *outOfMemory exception*, postavljeno je ograničenje učitavanja podatka većih od 2Gb kapaciteta.

U tu svrhu potrebno je podatke podijeliti u više segmenta. Jedan od načina je *split* naredba iz OSX-ovog terminala. S obzirom da metoda *split*-a memoriju, rubove podataka potrebno je urediti tako da linija ostane cjelovita.

Prikaz metode čitanja podataka:

```
public static void load(File f) throws IOException {
    FileInputStream fis = new FileInputStream(f);
    FileChannel fc = fis.getChannel();

    MappedByteBuffer mmb =
        fc.map(FileChannel.MapMode.READ_ONLY, 0, fc.size());
    byte[] buffer = new byte[(int)fc.size()];
    mmb.get(buffer);

    BufferedReader in = new BufferedReader(new
InputStreamReader(new ByteArrayInputStream(buffer)));

    for (String line = in.readLine(); line != null; line =
in.readLine()) {
        // unos podataka u bazu...
    }
}
```

Iz nekog nepoznatog razloga, dobije se manji broj podatka koristeći osnovne metode *java.lang.String* biblioteke, tako da ispada preciznije koristiti *java.util.StringTokenizer* biblioteku:

```
StringTokenizer st = new
StringTokenizer(line);
while (st.hasMoreTokens()) {
    System.out.println(st.nextToken());
}
```



}

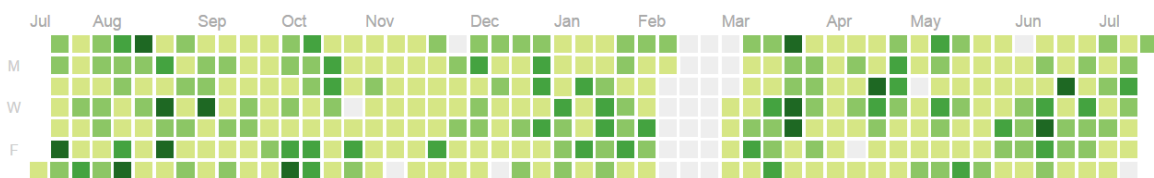
StringTokenizer klasa omogućuje jednostavno razbijanje tekstualnog zapisa u tokene.

Nakon što su svi podaci pohranjeni, prvi korak u procesu revizije je provjera svega što je prikupljeno. Važno je provjeriti postojanje podataka. Takav pristup najjednostavnije se izvodi SQL naredbom, točnije:

```
SELECT „datumProlaska“, COUNT(1)
FROM brojanjeVozila
GROUP BY „datumProlaska“
WHERE „senzor“ = ***
```

Navedena naredba vraća sve datume u kojima postoji barem jedno zabilježeno vozilo na senzoru \*\*\* i broj prolazaka vozila u tom danu.

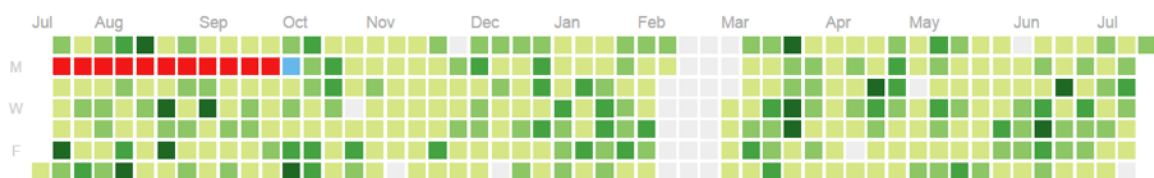
Tim upitom zapravo dobivamo prikaz rupa podacima koji dolaze sa senzora.



Slika 10. - Intenzitet podataka prema količini dnevnih prolazaka

Ova informacija je vrlo korisna u razumijevanju onoga što se dogodilo tijekom prikupljanja podataka na terenu, a također informacije su vrlo korisne pri izračunu težinskih vektora za poboljšanje mjerenja Kalmanovim filtrom.

Prema poljima postojanja podataka, jednostavno je napraviti odabir datuma koji će se uzeti u obzir pri statističkoj obradi.



Slika 11. - Prikaz odabranih ponedjeljka

Crveno označeni dani predstavljaju ponedjeljke koji se uzimaju u obzir pri izračunu srednje vrijednosti i standardne devijacije. Plavo označen ponedjeljak je onaj slijedeći, odnosno ponedjeljak čije parametre će se uspoređivati sa srednjim odstupanjima.

## 5. Statistička analiza mjernih podataka

Modeliranje je stvaranje (oblikovanje) modela. Prema tome bitno je razumijevanje što je model. Za model se kaže da se radi o pojednostavljenom prikazu realnog sustava napravljenog zato da bi poslužio boljem razumijevanju i/ili daljnjem proučavanju tog sustava i njegovom eksperimentiranju. Modelom se naziva kopija, prikaz, slika ili prezentacija određenog ili neodređenog realnog sustava.

Kako bi skup informacija mogli prikazati u obliku sustava potrebno ih je iskoristiti za modeliranje. Modeliranje se koristi u svrhu objašnjenja podataka i njihove povezanosti. Prema građi prometni model kretanja vozila kroz prometnicu je simbolički, odnosno apstraktni. Pošto su podaci jednom preuzeti, te nije u redu bez dozvole pristupa uzimati podatke za slaganje dinamičkog modela.

Statističkom obradom podataka rezultate modela iz stohastičke razdiobe pretvaramo u determinističku. Modeli ne smiju biti odveć složeni ni detaljni, nego treba modelirati samo relevantne elemente sustava. Odviše složene i detaljne modele teško je ili čak nemoguće razumjeti i vrjednovati, što znači da su i njihov razvoj u uporaba teški i neizvjesne kvalitete.

Model ne smije niti odveć pojednostavniti problem npr. Izbacivanjem varijabli nužnih za adekvatni opis sustava ili odveć velikim stupnjem agregiranja komponenti sustava. Razumno ga je rastaviti na više dobro definiranih i jednostavnih modula s točno određenom funkcijom koju je lakše izgraditi i provjeriti.

### 5.1 Istraživanje

Realni sustav koji istražujemo često nam je nedovoljno nepoznat te ćemo na temelju onoga što je poznato ili samo na temelju onoga što se pretpostavlja o tom sustavu napraviti njegov model koji pak treba poslužiti daljnjem proučavanju tog sustava i eksperimentiranju s njim. Ideja je da model posluži za reprodukciju ponašanja sustava kako bi se na temelju analize rezultata verificirala ili falsificirala hipoteza o njegovoj strukturi.

Počevši od dnevnog praćenja podatke uzimamo na sljedeći način:

```
if(calendar2.get(Calendar.DAY_OF_MONTH) == calendar.get(Calendar.DAY_OF_MONTH) &&  
calendar2.get(Calendar.MONTH) == calendar.get(Calendar.MONTH) &&  
calendar2.get(Calendar.YEAR) == calendar.get(Calendar.YEAR)){  
  
    //int sati = Integer.parseInt(vrijeme.split(":")[0]);
```

```

        double min = Double.parseDouble(vrijeme.split(":")[0]) * 60 +
Double.parseDouble(vrijeme.split(":")[1]) + Double.parseDouble(vrijeme.split(":")[2]) /
60;

        int x = (int) Math.ceil(min / 5);

        // Cluster na poziciji x povecati za 1, odnosno ako je u vremenskom periodu od
5 min zabilježeno
        // neko vozilo, tom vremenskom periodu dodajemo jednu znamenku

        polje[x-1]++;
    }

```

U uvjetu se provjerava da li je datum prolaska vozila jednak datumu ponedjeljka koji se uzeo u obzir prilikom istraživanja, odnosno stvaranja hipoteze. Ukoliko da, od svakog podatka, sati i sekunde se pretvaraju i zbrajaju u minute, te dijeljenjem s brojem 5 određuje se varijabla 'x' koja predstavlja spektar od 0 do 287. Naime radi se o broju 288 što predstavlja 288 5-minutnih dijelova dana. Uzimanjem najvećeg cijelog, odnosno *Math.ceil(Double ...)* metodom postavlja se ulazak u 5 minutni vremenski segment.

Za grafički prikaz brojanja prometa s obzirom na 5 minutne intervale, podatke na ekran računala ispisuje JavaFX biblioteka s ponešto izvedenih metoda za ugodniji i funkcionalniji prikaz:

```

final CategoryAxis xAxis = new CategoryAxis();
final NumberAxis yAxis = new NumberAxis();
xAxis.setLabel("Intervali od 5 min");
lineChart = new BarChart<String,Number>(xAxis,yAxis);
lineChart.setTitle("Praćenje prometa TC/NQR 10, "+calendar.getTime());

Task task = new Task<int[]>() {
    @Override
    protected int[] call()throws Exception {
        int[] novoP = getData();
        return novoP;
    }
};

task.setOnSucceeded(new EventHandler<Event>() {
    @Override
    public void handle(Event event) {
        polje = (int[])task.getValue();
        XYChart.Series serija = new XYChart.Series();
        series.setName("Broj vozila");

        for (int i = 0; i < polje.length; i++) {
            String naziv = ((i*5)/60) + " " + (i*5)%60;
            serija.getData().add(new XYChart.Data(naziv, polje[i]));
        }
        lineChart.getData().add(serija);
    }
});

```

Kreira se novi objekt klase *Task* koji se poziva u novom *Thread*-u kako bi se dva procesa paralelno izvodila. Jedan proces je pikselizacija grafičkih elementa na ekran, a drugi

kopanje, odnosno obrada podataka. Stvara se objekt *BarChart()* klase u koju je potrebno dodati objekte. Po rješenju ovog rada dodaju se objekti tipa *String* i *Int*. Objekti se dodaju u objekt 'Serija', a 'Serija' u objekt *BarChart()-a*.

```
VBox layout = new VBox();
layout.getChildren().setAll(createMenuBar(stage, lineChart));

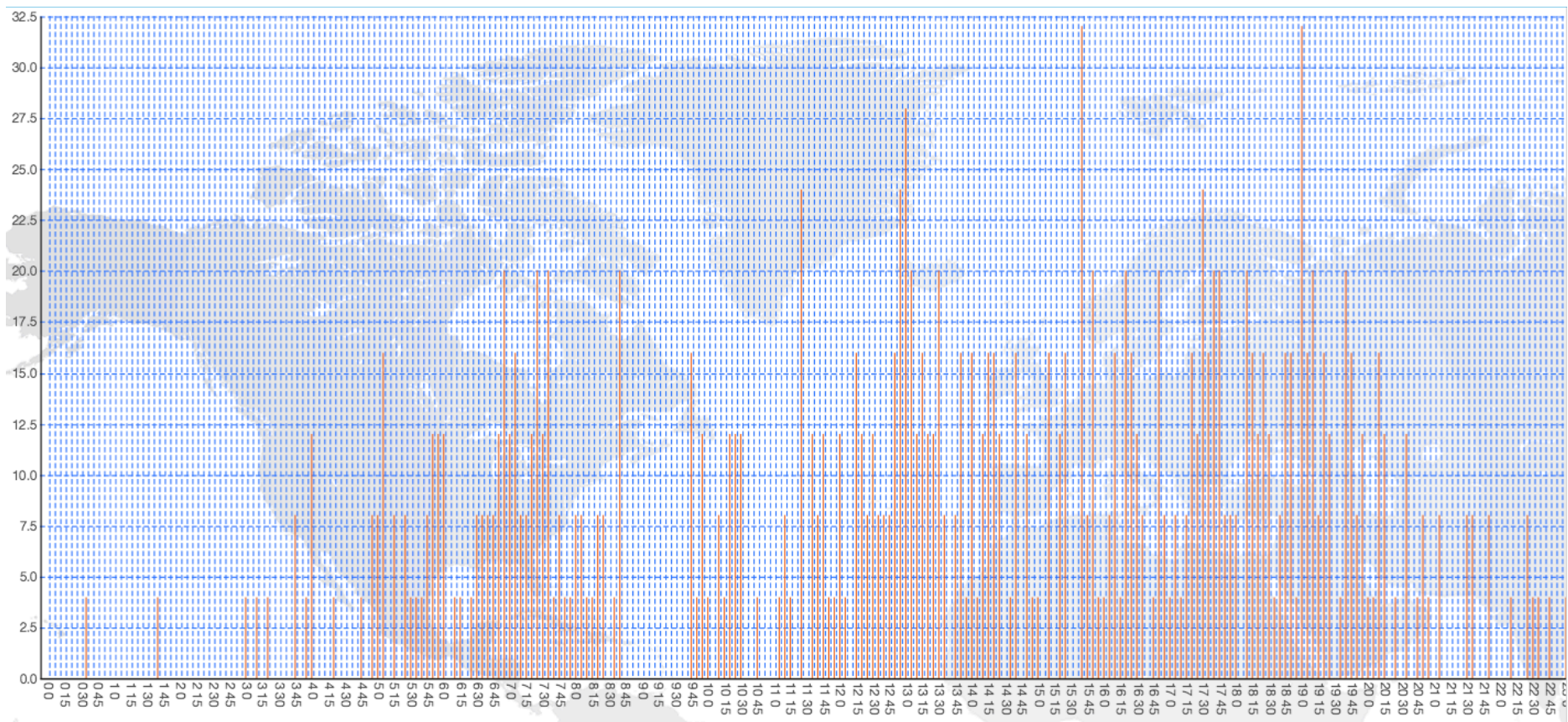
VBox.setVgrow(zoomPane, Priority.ALWAYS);
scene = new Scene(layout, 1600, 1200);

scene.getStylesheets().add("style.css");
```

U ovom segmentu koda, stvara se objekt klase *VBox* koja služi za postavljanje elemenata u prostor. Svaki element koji se doda u objekt dobije svoje mjesto ispod prethodno dodanog elementa. U kreiranju scene,

```
stage.setScene(scene);
stage.show();
newPopup.show(stage);
new Thread(task).start();
```

Na *stage.show()* otvara se vizualni prikaz scene koja je prethodno naredbom *stage.setScene()* bila dodana u argument. Te se podaci učitavaju odvojeno od *Application threada* (glavnog *threada* za grafički prikaz).



Slika 12. - Prikaz dnevnog brojanja u pet minutnim intervalima

Za bolji user experience, pri učitavanju podataka, postavlja se indikator procesa:

```
final ProgressIndicator pin = new ProgressIndicator();
pin.setProgress(-1.0f);
Popup newPopup = new Popup();
newPopup.getContent().add(pin);
```

Rotirajuće kuglice označuju radnju. Zbog količine podataka, radnja punjenja grafa podacima zahtjeva određeno vrijeme, ovisno o kvalitetama komponenta. Rotirajuće kuglice će biti prikazane za vrijeme tog obavljanja procesa.

## 5.2 Projektiranje

Projektiranje predstavlja projekciju ponašanja sustava u situacijama s kojima nema iskustva. Tada se ponašanje prometnog sustava zapravo simulira pomoću određenih alata kako bi se temeljito analizirao ispitivanjem različitih, potencijalno mogućih uvjeta u kojima bi se mogao naći te odabiralo ono rješenje koje se smatra optimalnim. Najčešće se to radi kako bi se sam realni sustav promijenio, pa rezultati simulacije čine temelj za donošenje odluka o promjeni realnog sustava. Za razliku od toga, potrebno je projektirati sustav kako bi nadopunio izgubljene podatke.

Za modeliranje ovakvog tipa sustava potrebno je stupčasti graf pretvoriti u funkciju krivulje, kako bi se u svakoj vremenskoj derivaciji otprilike pretpostavljao broj vozila. Određuju se točke (x-os je vrijeme, a y-os broj vozila), na temelju kojih interpolacijom sustav definira kontinuiranu funkciju. Radi velikih promjena u pet-minutnim intervalima, intervale je potrebno promijeniti u jedno-satne, na način da se zbraja dvanaest 5-minutnih intervala.

Projektiranje je izvedeno statističkim alatom standardnog odstupanja i varijance. Varijanca u teoriji vjerojatnosti je mjera statističke disperzije pojedinih varijabli. Pokazuje kako je stvarna vrijednost raspoređena duž linije očekivanih vrijednosti. Manje apstraktna mjera je korijen varijance, koji se naziva standardna devijacija [14].

$$\sigma^2 = \frac{\sum_{i=0}^N (a_i - \bar{a})^2}{N}, \quad (4)$$

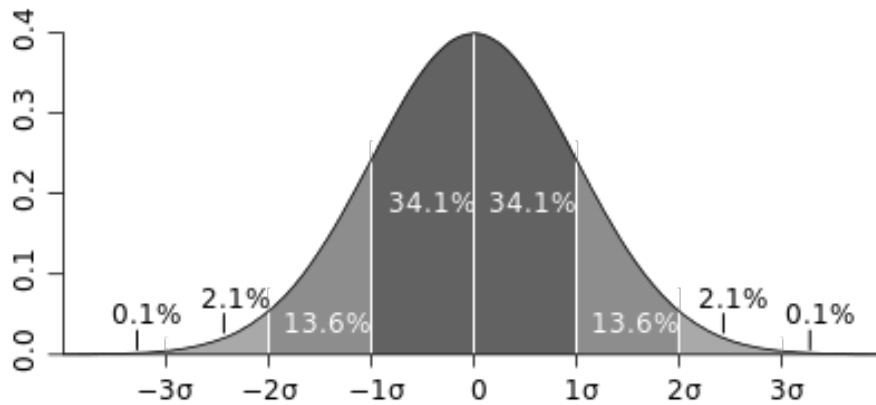
gdje je:

N – broj podataka

$\bar{a}$  – aritmetička sredina skupa

$a_i$  – i-ti član skupa ( $i = 1, 2, 3, \dots, N$ )

U praksi se često pretpostavlja da su podaci iz približno normalno raspodijeljene populacije. Ako je ta pretpostavka statistički gledano opravdana, onda se oko 68% vrijednosti nalazi u intervalu jedne standardne devijacije od sredine. Oko 95% vrijednosti se nalazi u intervalu od dvije standardne devijacije, a 99% tri. U statistici to pravilo naziva se empirijsko pravilo.



Slika 13. - Grafički prikaz empirijskog pravila raspodjele

Varijanca je definira formulom (4), čiji se izvod morao pretočiti u programski jezik. Radilo se s izvađenim podacima u 'txt' formatu, kako bi se prilikom svake interpretacije izbjeglo ponovno rudarenje podacima. Kreirana je nova klasa sa statičkim varijablama i metodama kako bi se izbjeglo korištenje konstruktora, jer za njih nema potrebe.

Prvi korak je pročitati podatke. Tako da kada se u kodu pokrene naredba:

```
reader(poljeSati, selectedFile, 0-8);
```

Aktivira se metoda *reader()*, te joj se šalju argumenti koji su joj potrebni. Njome se pokušavaju samo učitati podaci već izbrojanog prometa kategoriziranog po 5-minutnim segmentima..

```
private static void reader(int[][] poljeSati, File file, int k) {
    try (BufferedReader br = new BufferedReader(new FileReader(file))) {
        String sCurLine;
        int j = 0, i = 0;
        while ((sCurLine = br.readLine()) != null) {
            poljeSati[i][k] = poljeSati[i][k]+Integer.parseInt(sCurLine);
            if (j==12) {
                j=0;
                i++;
            }
            j++;
        }
    }
    catch (IOException e) {e.printStackTrace();}
}
```

U ovih nekoliko koraka podaci se grupiraju u satne intervale. Podaci su se morali pregrupirati radi velike disperzije između različitih 5-minutnih intervala. Ta disperzija uvelike utječe na crtanje krivulje, jer kod crtanja krivulja izvodi se postupak spline interpolacije. Moglo se još pristupiti problemu korištenjem polinomijalne interpolacije, ali rješenje koje daje spline je preciznije i interpolant je glađi.

Slijedeći korak je te vrijednosti posložiti u novo polje nazvano 'poljeSati' iz kojega se uzimaju vrijednosti pri računanju aritmetičke sredine i standardne devijacije. Rezultati se zapisuju u novo polje 'rez', čiju vrijednost vraća kako bi vrijednosti bile dostupne za crtanje funkcija. Naime radi se o dvostrukom polju, matrici kojoj stupac definira radi li se o srednjoj vrijednosti, gornjoj granici standardne devijacije ili donjoj, a redak o kojem satu je riječ.

```
double[][] rez = new double[24][3];
```

Slijedećih par naredbi, iz pročitanih vrijednosti izračunava aritmetičku sredinu:

```
int[] prosjek = new int[24];
for (int i = 0; i < 24; i++) {
    for (int j = 0; j < 9; j++) {
        prosjek[i] = prosjek[i]+poljeSati[i][j];
    }
    prosjek[i] = prosjek[i]/9;
    rez[i][1] = prosjek[i];
}
```

U sekciji idućih linija koda vidljivo je računanje varijance:

```
int [] o2 = new int[24];
int suma = 0;
for (int i = 0; i < o2.length; i++) {
    for (int j = 0; j < 9; j++) {
        suma = (int) Math.pow(poljeSati[i][j]-prosjek[i], 2);
    }
    o2[i] = suma / 9;
}
```

Te iz izračunate varijance, vadi se korijen, kako bi se dobilo standardno odstupanje

```
double[] o = new double[24];
for (int i = 0; i < o.length; i++) {
    o[i] = Math.sqrt(o2[i]);
    rez[i][0] = prosjek[i] - o[i];
    rez[i][2] = prosjek[i] + o[i];
    System.out.println(i+" - "+ o[i]);
}

return rez;
}
```

Metoda *standardnoOdstupanje()* vraća *double* vrijednosti, kako bi se zadržala preciznost dobivena iz operacije korjenovanja.



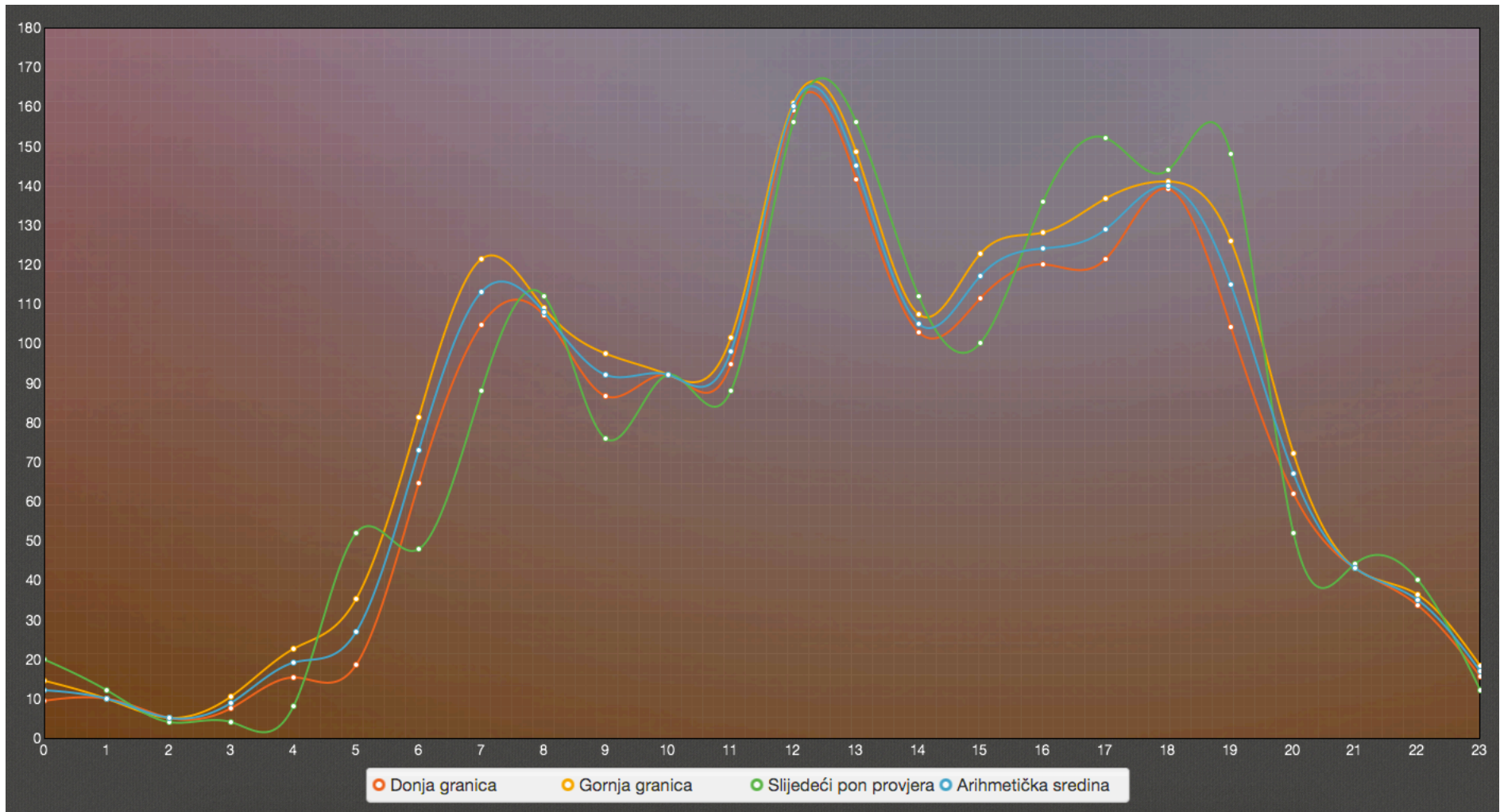
## 6. Rezultat nadopune podataka

Radi predočavanja brojki, podatke je potrebno ispisati skupa sa sljedećim ponedjeljkom kao grafičkim prikazom odstupanja od srednjih vrijednosti. Na gotovo isti način kao i zadnji prikaz, stvaramo XYChart, no vrijednosti dodane u objekte *series* na drugačiji su način raspoređene na grafički prikaz.

```
BezierovaKrivulja chart2 = new BezierovaKrivulja(  
    new NumberAxis(0,23,1), new NumberAxis(0,180,10));  
  
final XYChart.Series<Number, Number> series1 = new XYChart.Series<Number, Number>();  
final XYChart.Series<Number, Number> series2 = new XYChart.Series<Number, Number>();  
final XYChart.Series<Number, Number> series3 = new XYChart.Series<Number, Number>();  
final XYChart.Series<Number, Number> series4 = new XYChart.Series<Number, Number>();  
  
series1.setName("Donja granica");  
series2.setName("Gornja granica");  
series3.setName("Slijedeći pon provjera");  
series4.setName("Aritmetička sredina");  
  
for (int i = 0; i < 24; i++) {  
    series1.getData().add(new XYChart.Data<Number,Number>(i,rez[i][0]));  
    series2.getData().add(new XYChart.Data<Number,Number>(i,rez[i][2]));  
    series3.getData().add(new XYChart.Data<Number,Number>(i,poljeSatiPon[i]));  
    series4.getData().add(new XYChart.Data<Number,Number>(i,rez[i][1]));  
}  
  
chart2.getData().addAll(series1, series2, series3, series4);
```

S time da je klasa *BezierovaKrivulja()* izvedena prema uzoru na Bézierovu krivulju. Bézierova krivulja je parametrična krivulja važna u području matematičke numeričke analize, a osobito se primjenjuje u računalnoj grafici i srodnim djelatnostima. Implementacija proširuje *AreaChart()* i *override*-a *layoutPlotChildren()* metodu. Metoda stvara sve potrebne čvorove (*Node*) za crtanje grafa te generira sva svojstva serije i podatka. Zatim se poziva super.*layoutPlotChildren()* kako bi standardni kod odradio sve što treba odraditi, a napravljena je zamjena *LineTo* klase u *CubicCurveTo* s dvije nove kontrolne točke izračunate na način na koji je proračunato u članku [9]. Jedini nepotrební trošak je u tome što se kreira objekt klase *LineTo*, pa odmah briše kako bi se zamijenio s *CubicCurveTo* objektom. Kod koji to prikazuje nalazi se u prilogu A.

To bi bilo to. Još je potrebno prema volji, odnosno oku u *.css file*-u urediti grafičke elemente. Pokretanjem aplikacije dobivamo sljedeći prikaz



Slika 14. - Prikaz srednje vrijednosti, standardne devijacije i slijedećeg ponedjeljka

Pristup kompletnom kodu dostupan je globalno svima na adresi poveznice [15]. Naime, radi se o skladištu (*repository*) postavljenom na bitbucket portalu, koji koristi 'git' *version control system*. Git sustav osmišljen je za radove u timovima programera gdje više ljudi radi na istom projektu, te na skladištenju starijih verzija aplikacija ukoliko postoji potreba da se korisnicima omogući podrška kroz više verzija aplikacije.

Zelena linija predstavlja sljedeći ponedjeljak kojega se pokušalo aproksimirati aritmetičkom sredinom. Prema grafu, jasno je vidljivo njegovo odstupanje i zapravo neprihvatljiva preciznost. Iz tog razloga nužno je proći neke druge metode modeliranja, kako bi rezultat bio dovoljno dobar da postane upotrebljiv i primjenjiv.

## 7. Zaključak

U ovom radu obrađen je problem nadopune nedostajućih mjernih podataka za prometni tok. Korišteni su stvarni podaci kako bi se izradila aplikacija koja pristupa podacima iz baze te statističkom obradom podataka odredio profil toka za ponedjeljak. Korištena je prosječna vrijednost, a pomoću standardne devijacije ocjenjena je točnost dobivene karakteristike.

Gledajući uzastopne ponedjeljke, teško je po njihovim obilježjima precizno predvidjeti obilježje sljedećeg. Iz tog razloga potrebno je isprobavati mnoge druge metode dok se rezultat ne prikaže dovoljno dobrim da postane uporabljiv. Idući pokušaj mogao bi se temeljiti na algoritmima učenja, primjerice neuronskim mrežama. Dobrim poznavanjem sustava moguće je uvrstiti još informacija u razmatranje. Na primjer, odnose između različitih senzora na istoj prometnici ili grupa senzora koji predstavljaju ulaz ili izlaz na autocestu. Možda promatrajući segmente prometnice, odnosno kako promet na jednom dijelu utječe na druge dijelove. Tu se radi o vremenu proučavanja specifičnog sustava, te je prema skupljenom znanju i informacijama upitna njegova primjena na druge sustave.

Tradicionalna uporaba senzora na cesti za prikupljanje podataka (kao npr. Induktivne petlje) prema dosadašnjim modelima je nužna, ali ne i dovoljna zbog ograničene pokrivenosti i velikih troškova implementacije te održavanja. U posljednjih nekoliko godina svjedoci smo razvoja alternativnih izvora prometnih podataka, kao što su društvene mreže te sama vozila.

Računanje prometnih parametara alternativnim metodama, primjerice metodama koje se zasnivaju na lokaciji vozila (npr. GPS podaci o pokretnom vozilu) djeluje kao obećavajuće rješenje u odnosu na fiksne detektore. Čak i ako ideja prikupljanja podataka iz vozila uređajima putem mobitela ili GPS-a nije posve nova, primjerice to tržište tek sada raste sa širokim rasponom aplikacija i naknada. To ne samo da poboljšava upravljanje prometom, nego također i pomaže zadovoljiti rastuću potražnju za vozače koji su spremni platiti uslugu sve dok imaju pristup relevantnim informacijama na koje se mogu osloniti.

Sama aplikacija je napravljena u programskom jeziku Java, koji omogućava rješavanje i složenije problematike, te implementaciju u bilo koji sustav. Objektno je orijentiran i konstantno evoluirao. Iz tog razloga je implementirana aplikacija pogodna za daljnji razvoj odnosno testiranje naprednijih algoritama nadopune nedostajućih mjernih podataka.

## Literatura

- [1] LAWRENCE A. KLEIN AND MICHAEL R. KELLEY. DETECTION TECHNOLOGY, FOR IVHS – VOLUME 1, FINAL REPORT ADDENDUM, 1995.
- [2] LAWRENCE A. KLEIN, MILTON K. MILLS AND DAVID R. P. GIBSON. TRAFFIC DETECTOR HANDBOOK. THIRD EDITION – VOLUME 1, 2006.
- [3] HENRY LEUNG, SUBHAS CHANDRA MUKHOPADHYAY. INTELLIGENT ENVIRONMENTAL SENSING – (Smart sensors, Measurement and Instrumentation), 2013.
- [4] NIKO JELUŠIĆ: EVALUACIJA SENZORSKIH TEHNOLOGIJA U SUSTAVU AUTOMATSKOG UPRAVLJANJA CESTOVNIM PROMETOM, DOKTORSKA DISERTACIJA, FAKULTET PROMETNIH ZNANOSTI SVEUČILIŠTE U ZAGREBU, 2007.
- [5] SANJA PALAJSA: ANALIZA DISTRIBUCIJE ZEMLJE PORIJEKLA VOZILA NA ZAGREBAČKOJ OBILAZNICI, ZAVRŠNI RAD 2015.
- [6] JOHN D. ZEGER: HIGHWAY CAPACITY MANUAL 2000, CHAPTER 7 – TRAFFIC FLOW PARAMETERS INTRO
- [7] PLANKO ROŽIĆ: UTJECAJ PROMETNOG OPTEREĆENJA NA GOSPODARENJE CESTOVNIM SUSTAVOM. GRAĐEVINAR 57 10, PP. 769-777, IZVORNI ZNANSTVENI RAD 2005 GOD.
- [8] GORDON, R. L. ET AL. TRAFFIC CONTROL SYSTEMS HANDBOOK, FEDERAL HIGHWAY ADMINISTRATION, U.S. DOT, FINAL REPORT, REPORT NUMBER: FHWA-SA 95-032, WASHINGTON DC, 1996.
- [9] OLEG V. POLIKARPOTCHKIN, PETER LEE: DRAW A SMOOTH CURVE THROUGH A SET OF 2D POINTS WITH BEZIER PRIMITIVES, 2009. PRISTUPLJENO 03.09.2015. (<http://www.codeproject.com/Articles/31859/Draw-a-Smooth-Curve-through-a-Set-of-2D-Points-wit>)
- [10] <http://www.hindawi.com/journals/ijdsn/2014/961073/fig2/>, Pristupljeno 03.09.2015.
- [11] U.S. DEPARTMENT OF TRANSPORT. TRAFFIC DETECTOR HANDBOOK: THIRD EDITION – VOLUME II, PRISTUPLJENO 03.09.2015. (<http://www.fhwa.dot.gov/publications/research/operations/its/06139/chapt5d.cfm>)
- [12] IBM Inc.: IBM SPSS Modeler CRISP-DM Guide, 2011
- [13] YOUNGTAE JO, JINSUP CHOI AND INBUM JUNG: TRAFFIC INFORMATION ACQUISITION SYSTEM WITH ULTRASONIC SENSORS IN WIRELESS SENSOR NETWORKS, 2014.
- [14] E. KOVAČ-STRIKO, N. KAPETANOVIĆ I B. IVANKOVIĆ. VJEROJATNOST I STATISTIKA, 2005.
- [15] PAVAO PETRAČ: ZAVRŠNI RAD – PROGRAMSKI KOD, DOSTUPNO ONLINE 03.09.2015. ([HTTPS://BITBUCKET.ORG/NAJDUZI\\_NIZ/ZAVR-NI-RAD.GIT](HTTPS://BITBUCKET.ORG/NAJDUZI_NIZ/ZAVR-NI-RAD.GIT))

## Popis kratica

PHF	<i>Peak-hour factor</i>	Faktor vršnog sata
LOS	<i>Level of service</i>	Razina usluge
AUP	<i>Automatic traffic management</i>	Automatsko upravljanje prometom
DBMS	<i>Database management system</i>	Sustav upravljanja bazom podataka
DM	<i>Data mining</i>	Rudarenje podacima

# Popis tablica

<i>Tablica 1. - Kvaliteta usluge za autoceste s više prometnih traka [4] .....</i>	<i>8</i>
<i>Tablica 2. – Model zapisa baze podatka dobiven od tvrtke Led-elektronika .....</i>	<i>14</i>

# Popis slika

<i>Slika 1. - Ciklus osnovnih funkcija AUP-a i tijek informacija [4] .....</i>	<i>2</i>
<i>Slika 2. - Generalizirana veza između brzine, gustoće i protoka .....</i>	<i>7</i>
<i>Slika 3. - Postavljanje pneumatskih cijevi na prometnicu.....</i>	<i>10</i>
<i>Slika 4. - Detekcija vozila koristeći ultrazvučne senzore [13].....</i>	<i>11</i>
<i>Slika 5. - Prikaz postavljanja ultrazvučnog senzora [2].....</i>	<i>11</i>
<i>Slika 6. – Model slaganja induktivnih petlji na cesti [2].....</i>	<i>12</i>
<i>Slika 7. - Prikaz procesa rudarenja podataka .....</i>	<i>13</i>
<i>Slika 8. - Vektorska slika prometnice u Kosovu koju je prema projektu za Bechtel Enka GP odradila tvrtka Led-elektronika d.o.o.....</i>	<i>15</i>
<i>Slika 9. - Vektorska slika jednog segmenta uređaja.....</i>	<i>17</i>
<i>Slika 10. - Intenzitet podataka prema količini dnevnih prolazaka.....</i>	<i>17</i>
<i>Slika 11. - Prikaz odabranih ponedjeljka.....</i>	<i>17</i>
<i>Slika 12. - Prikaz dnevnog brojanja u pet minutnim intervalima.....</i>	<i>21</i>
<i>Slika 13. - Grafički prikaz empirijskog pravila raspodjele .....</i>	<i>21</i>
<i>Slika 14. - Prikaz srednje vrijednosti, standardne devijacije i slijedećeg ponedjeljka .....</i>	<i>26</i>



# Prilog A

```
@Override protected void layoutPlotChildren() {
    super.layoutPlotChildren();
    for (int seriesIndex=0; seriesIndex < getDataSize(); seriesIndex++) {
        final XYChart.Series<Number, Number> series = getData().get(seriesIndex);
        final Path seriesLine = (Path)((Group)series.getNode()).getChildren().get(1);
        final Path fillPath = (Path)((Group)series.getNode()).getChildren().get(0);
        smooth(seriesLine.getElements(), fillPath.getElements());
    }
}

private int getDataSize() {
    final ObservableList<XYChart.Series<Number, Number>> data = getData();
    return (data!=null) ? data.size() : 0;
}

private static void smooth(ObservableList<PathElement> strokeElements,
ObservableList<PathElement> fillElements) {
    // Pošto nema direktnog pristupa podacima, potrebno je napraviti listu svih x i
    // y koordinata (točaka)
    final Point2D[] dataPoints = new Point2D[strokeElements.size()];
    for (int i = 0; i < strokeElements.size(); i++) {
        final PathElement element = strokeElements.get(i);
        if (element instanceof MoveTo) {
            final MoveTo move = (MoveTo)element;
            dataPoints[i] = new Point2D(move.getX(), move.getY());
        } else if (element instanceof LineTo) {
            final LineTo line = (LineTo)element;
            final double x = line.getX(), y = line.getY();
            dataPoints[i] = new Point2D(x, y);
        }
    }
    // preuzimanje Y koordinate
    final double zeroY = ((MoveTo) fillElements.get(0)).getY();

    // čišćenje i rebuild elementa
    strokeElements.clear();
    fillElements.clear();
    Pair<Point2D[], Point2D[]> result = calcCurveControlPoints(dataPoints);
    Point2D[] firstControlPoints = result.getKey();
    Point2D[] secondControlPoints = result.getValue();
    // početak putanja
    strokeElements.add(new MoveTo(dataPoints[0].getX(),dataPoints[0].getY()));
    fillElements.add(new MoveTo(dataPoints[0].getX(),zeroY));
    fillElements.add(new LineTo(dataPoints[0].getX(),dataPoints[0].getY()));
    // dodavanje krivulja
    for (int i = 1; i < dataPoints.length; i++) {
        final int ci = i-1;
        strokeElements.add(new CubicCurveTo(
            firstControlPoints[ci].getX(),firstControlPoints[ci].getY(),
            secondControlPoints[ci].getX(),secondControlPoints[ci].getY(),
            dataPoints[i].getX(),dataPoints[i].getY()));
        fillElements.add(new CubicCurveTo(
            firstControlPoints[ci].getX(),firstControlPoints[ci].getY(),
            secondControlPoints[ci].getX(),secondControlPoints[ci].getY(),
            dataPoints[i].getX(),dataPoints[i].getY()));
    }
    // kraj putanja
    fillElements.add(new LineTo(dataPoints[dataPoints.length-1].getX(),zeroY));
    fillElements.add(new ClosePath());
}
}
```



Sveučilište u Zagrebu  
Fakultet prometnih znanosti  
10000 Zagreb  
Vukelićeva 4

## METAPODACI

**Naslov rada:** Mogućnosti Nadopune mjernih podataka prometnih parametara na osnovu povijesnih vrijednosti

**Autor:** Pavao Petrač, 0135230631

**Mentor:** doc. dr. sc. Edouard Ivanjko

**Naslov na drugom jeziku (engleski):**

Possibilities for Complementation of Measured Traffic Parameters Data Based on Historic Values

**Povjerenstvo za obranu:**

- prof. dr. sc. Sadko Mandžuka , predsjednik
- doc. dr. sc. Edouard Ivanjko , mentor
- mr. sc. Goran Jurković , član
- prof. dr. sc. Hrvoje Gold , zamjena

**Ustanova koja je dodjelila akademski stupanj:** Fakultet prometnih znanosti Sveučilišta u Zagrebu

**Zavod:** Zavod za inteligentne transportne sustave

**Vrsta studija:** sveučilišni

**Naziv studijskog programa:** Inteligentni transportni sustavi i logistika

**Stupanj:** preddiplomski

**Akademski naziv:** univ. bacc. ing. traff.

**Datum obrane završnog rada:** Rujan 2015.



Fakultet prometnih znanosti  
10000 Zagreb  
Vukelićeva 4

## IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj završni 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 završnog rada

pod naslovom **Mogućnost nadopune mjernih podataka prometnih**

**parametara na osnovu povijesnih vrijednosti**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom

repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 03/09/15

(potpis)