Izrada informacijsko komunikacijskog sustava za detekciju anomalija prometnog toka na urbanim prometnicama

Majstorović, Željko

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Željko Majstorović

IZRADA INFORMACIJSKO - KOMUNIKACIJSKOG SUSTAVA ZA DETEKCIJU ANOMALIJA PROMETNOG TOKA NA URBANIM PROMETNICAMA

DIPLOMSKI RAD

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
POVJERENSTVO ZA DIPLOMSKI ISPIT

Zagreb, 1. travnja 2020.

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

DIPLOMSKI ZADATAK br. 5886

Pristupnik: **Željko Majstorović (0135232496)**
Studij: Promet
Smjer: Informacijsko-komunikacijski promet

Zadatak: **Izrada informacijsko komunikacijskog sustava za detekciju anomalija prometnog toka na urbanim prometnicama**

Opis zadatka:

Predobrađene GPS podatke u obliku prijelaznih matrica brzina potrebno je pohraniti u MongoDB bazu podataka.Potrebno je odabrati prikladni način indeksiranja i omogućiti dohvat podataka aplikaciji za prikaz i obradu podataka. Aplikaciju je potrebno izraditi u Python programskom jeziku s tri osnovne funkcionalnosti i to obrada slike u svrhu određivanja težišta objekta, detekcija anomalija i vizualizacija rezultata na karti. Metodu za procjenu udaljenosti između dvije prijelazne matrice brzina u svrhu detekcije anomalije je potrebno usporediti s ostalim mjerama udaljenosti koje se koriste u ovom području.

Mentor:

prof. dr. sc. Tonči Carić

Predsjednik povjerenstva za diplomski ispit:

Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

IZRADA INFORMACIJSKO - KOMUNIKACIJSKOG SUSTAVA ZA DETEKCIJU ANOMALIJA PROMETNOG TOKA NA URBANIM PROMETNICAMA

DEVELOPMENT OF AN INFORMATION AND COMMUNICATION SYSTEM FOR THE DETECTION OF TRAFFIC FLOW ANOMALIES ON URBAN ROADS

Mentor: prof. dr.sc. Tonči Carić

Student: Željko Majstorović
JMABG: 0135232496

Zagreb, srpanj 2020.

Zahvaljujem se prof. dr. sc. Tončiju Cariću na mentorstvu tijekom izrade ovog rada. Hvala Leu Tišljariću, mag. ing. traff., na iznimnom strpljenju i nesebičnoj pomoći tijekom tehničke izvedbe ovog rada.

Posebne zahvale mojoj obitelji, djevojci i prijateljima koji su mi pružali bezuvjetnu podršku tijekom studija.

IZRADA INFORMACIJSKO KOMUNIKACIJSKOG SUSTAVA ZA DETEKCIJU ANOMALIJA PROMETNOG TOKA NA URBANIM PROMETNICAMA

SAŽETAK:

Zagušenja na prometnicama su česta pojava s kojom se svakodnevno susreće veliki broj ljudi. Ona nepovoljno utječu ne samo na sudionike u prometu nego i na stanovnike u gradu i okoliš. U ovom radu opisana je metoda za detekciju anomalija korištenjem prikupljenih GPS podataka, prikazanih koristeći koncept prijelaznih matrica brzina na području grada Zagreba. Opisane su metode pristupa detekciji anomalija te izrada sustava za pohranu podataka, kao i metode korištene za obradu prometnih podataka. U radu je opisan postupak određivanja težišta i mjere udaljenosti između težišta promatrane i referentne prijelazne matrice brzina. Cilj rada je izrada informacijsko - komunikacijskog sustava za detekciju anomalija na urbanim prometnicama te usporedba predložene mjere za udaljenost s ostalim mjerama udaljenosti. U tu svrhu je izrađena aplikacija za detekciju anomalija pomoću programskog jezika Python i MongoDB, NoSQL baze podataka, a detektirane anomalije su prikazane na interaktivnoj karti grada Zagreba.

KLJUČNE RIJEČI: detekcija anomalija, obrada slika, težišta, vizualizacija podataka, GPS, IQR, NoSQL, MongoDB, Python

DEVELOPMENT OF AN INFORMATION AND COMMUNICATION SYSTEM FOR THE DETECTION OF TRAFFIC FLOW ANOMALIES ON URBAN ROADS

SUMMARY:

This thesis describes the method for anomaly detection using the collected GPS data in form of the speed transition matrices in the city of Zagreb. Methods for anomaly detection and development of data storage system are described, as well as methods used for image processing. The thesis describes the procedure for determining the center of mass and measures the distance between the center of mass of the observed and the reference speed transition matrix. The aim of this thesis is to develop an information and communication system for the detection of anomalies on urban roads and to compare the proposed distance measure with other distance measures. For this purpose, an application for the detection of anomalies using the Python programming language and the MongoDB, NoSQL database was developed. Detected anomalies are shown on an interactive map of the city of Zagreb.

KEY WORDS: anomaly detection, image processing, center of mass, data visualization, GPS, IQR, NoSQL, MongoDB, Python

Sadržaj

1. Uvod

Pojam anomalije se razlikuje ovisno o okviru područja u kojem se promatra. U bankarskim sustavima anomalijom se može smatrati sumnjiva transakcija, u komunikacijskim sustavima primjer anomalije može biti neovlašteni pristup mreži ili sustavu, dok u prometnom sustavu anomalija se može manifestirati kao ekstremno prometno zagušenje. Osim što se pojam anomalije razlikuje u ovisnosti o području promatranja isto tako se razlikuje i interpretacija anomalija. Primjerice, oscilacije tjelesne temperature se u medicini smatraju anomalijom dok se oscilacije na tržištu nekretnina ne smatraju anomalijama [1].

Detekcija anomalija u prometnom toku na urbanim prometnicama je važno područje istraživanja jer omogućava precizniju i učinkovitiju analizu urbanih prometnica i bolje upravljanje prometom. Postoji više definicija anomalije, prema [2] anomalija je podatak koji toliko odstupa od ostatka podataka da postoji sumnja da taj podatak potječe iz drugog izvora.

U ovom radu korišteni su prikupljeni GPS (*engl. Global Positioning System*) podaci koje je zabilježilo oko 4200 vozila tijekom petogodišnjeg perioda. Svrha ovog rada je obrada prikupljenih podataka te izrada informacijsko - komunikacijskog sustava za obradu slika u svrhu detekcije anomalija na urbanim prometnicama.

Ciljevi rada su:

- Izraditi podsustav za pohranjivanje podataka koristeći NoSQL (*engl. Not only SQL*) bazu podataka
- Opisati razlike između SQL (*engl. Structured Query Language*) i NoSQL baza podataka
- Izraditi aplikaciju za obradu slika koja koristi podatke spremljene u NoSQL bazu podataka
- Prikazati predloženu metodu za detekciju anomalija
- Usporediti predloženu metodu s drugim metodama za mjerenje udaljenosti između dvije matrice
- Izraditi grafičko korisničko sučelje za aplikaciju

1.1. Struktura rada

U prvom poglavlju opisan je predmet istraživanja te su definirani svrha i ciljevi rada. Isto tako iznesen je pregled dosadašnjih istraživanja na temu detekcije anomalija te korišteni podaci za izradu rada.

Drugo poglavlje opisuje detekciju anomalija na urbanim prometnicama. Dan je pregled metoda pristupa detekciji anomalija te mogućih izvora anomalija.

U trećem poglavlju je opisana usporedba relacijskih baza podataka i NoSQL baza podataka, istaknute su bitne značajke te razlike između spomenutih baza podataka. Također opisana je pohrana i dohvat podataka u NoSQL bazu podataka.

Četvrto poglavlje je opisuje izradu aplikacijskog rješenja. Opisane su značajke programskog jezika i korištenih alata za izradu aplikacijskog rješenja. Također, opisane su i korištene metode za obradu slika i detekciju anomalija te popratni korišteni alati u procesu detekcije anomalija.

U petom poglavlju dan je pregled dobivenih rezultata dobivenih korištenjem relativne Euklidske udaljenosti kao mjere za udaljenost, te usporedba dobivenih rezultata s mjerama udaljenosti Manhattan, Kosinusna udaljenost i Jaccard udaljenost.

U šestom poglavlju dan je komentar i zaključak na dobivene rezultate istraživanja i cjelokupni rad. Istaknute su prednosti i nedostatci istraživanja te moguća poboljšanja i budući razvoj.

1.2. Pregled dosadašnjih istraživanja

Autori rada [3] su razvili metodu za detekciju različitih javnih događaja koji su se razlikovali od uobičajenih uzoraka kretanja stanovništva. Metoda je temeljena na faktorizaciji tenzora gdje su podaci o kretanju mase stanovništva prikupljeni iz sustava dijeljenja bicikala (*engl. Bike-sharing system*), upotpunjeni s podacima o aktivnosti na društvenim mrežama. U radu [4], autori su svoje istraživanje temeljili na višestrukim izvorima podataka kako bi detektirali anomalije s nepoznatim vanjskim utjecajima te kako bi anomalije bile detektirane prije nego dosegnu vrhunac. Autori rada [5] i [6] su koristili metode dekompozicije tenzora [7] za modeliranje prometnih podataka koji sadrže prostorno vremenske podatke te lokalni indeks

anomalije (e*ngl. Local Outlier Factor* - LOF), kako bi uspješno detektirali anomalije i odredili vremenske intervale pojave anomalija u određenom području.

U radu [8] autori su istraživanje temeljili na podacima prikupljenim iz prometnih senzora koristeći metodu analize glavnih komponenata (*engl. Principal Component Analysis* - PCA) za analizu prometnih podataka kako bi se identificirale značajke prostornih i vremenskih uzoraka. Predložena metoda je dala jasne geografski raspoređene značajke prostornog i vremenskog uzorka prometnog toka u obliku toplinske karte (*engl. Heatmap*) gdje su anomalije bile lako uočljive. Taksi vozila su vrijedan izvor GPS podataka ako uzmemo u obzir prirodu njihove djelatnosti. Autori rada [9] su koristili GPS podatke prikupljene od taxi vozila kako bi stvorili matricu prometnog toka i razvili metodu temeljenu na PCA metodi za detekciju anomalija u gradskim četvrtima. Predložena metoda se pokazala učinkovitom u otkrivanju anomalija u susjednim područjima kao i identifikaciji prometnih tokova koji uzrokuju prometne anomalije.

Autori rada [10] su također koristili GPS podatke prikupljene od taksija i razvili metodu detekcije anomalija koja se fokusira na cestovne segmente umjesto na putanje što je rezultiralo detekcijom velikog broja događaja (anomalija), a svaki događaj je povezan s cestovnim segmentom i vremenskim periodom kada se događaj odvijao.

Jednostavne značajke poput brzine i udaljenosti također mogu poslužiti kao pokazatelj za detekciju anomalija. Temeljem fizikalnog obilježja udaljenosti između polazišta i odredišta, autori rada [11] su razvili metodu pomoću koje se određuje anomalija, odnosno odabir optimalne rute za vožnju taksijem. Autori rada [12] su na temelju brzine vozila koja je očitana senzorima mobilnih terminalnih uređaja, uspješno detektirali oštećenja kolnika i uspornike prometa te su tako detektirali anomalije na gradskim prometnicama koje se odnose na fizičke zapreke na prometnicama. U radu [13], autori su detektirali prometna zagušenja na temelju procjene brzine prometnog toka u normalnim uvjetima i stvarne brzine zabilježene na pojedinom cestovnom segmentu.

2. Detekcija anomalija na urbanim prometnicama

Detekcija anomalija označava problematiku identifikacije uzorka podataka koji se značajnije razlikuje od ostatka podataka. U različitim područjima istraživanja anomalije poprimaju i različite nazive pa se tako mogu naći nazivi poput: šum, iznimka, nedostatak, greška itd. Anomalije su aktualna tema u gotovo svim područjima istraživanja neovisno o tome radi li se o bankarskim sustavima, komunikacijskim mrežama ili primjerice, prometnim mrežama, svako od spomenutih područja je podložno anomalijama [14–16].

Tijekom procesa prikupljanja, obrade i analize podataka anomalije mogu nastati iz različitih izvora i očitovati se na različite načine, a kvaliteta ulaznih podataka je važna jednako kao i metoda detekcije anomalija. Najčešći uzroci anomalija su [17]:

- Greška prilikom unosa podataka - ljudska pogreška
- Greška mjerenja - pogreška uređaja
- Eksperimentalna pogreška - pogreška ekstrakcije ili pogreška planiranja odnosno provođenja eksperimenta
- Namjerna pogreška - testni podaci za testiranje metoda za detekciju
- Pogreška obrade podataka - nenamjerne izmjene podataka prilikom manipulacije podacima
- Pogreška uzorkovanja - ekstrakcija ili miješanje podataka iz pogrešnih ili različitih izvora
- Prirodni - novosti u podacima, nisu greška

Postoje različite metode detekcije anomalija, a odabir metode ovisi o podacima koji se obrađuju i anomalijama koje se žele detektirati. Podaci koji se analiziraju često sadrže i šum što detekciju anomalija čini složenim procesom i ne postoji univerzalno pravilo za detekciju anomalija. Osim toga, rijetko su dostupni i testni podaci na kojima se može provesti istraživanje na kojima bi se mogla testirati uspješnost metode za detekciju anomalija.

Najveći izazov u detekciji anomalija je odrediti što je „normalno“, s tim da granica između normalnog i anomaličnog stanja nije precizna pa svakom slučaju primjene se mora pristupiti na odgovarajući način, sukladno cilju koji se želi ostvariti i sukladno podacima s kojima se raspolaže.

Općenito pristup detekciji anomalija se može podijeliti na dvije kategorije, a to je nadzirani (*engl. Supervised)* i nenadzirani (*engl. Unsupervised*) pristup. Svaki od spomenutih pristupa objedinjuje određene metode detekcije anomalija koje se mogu podijeliti u tri kategorije [18]:

1. **Metode temeljene na udaljenosti** (*engl. Distance – based methods*) – metode koje podatke promatraju kao objekte, te se detekcija anomalija temelji na mjeri udaljenosti između objekata

2. **Metode temeljene na gustoći** (*engl. Density – based methods*) – ove metode detektiraju anomalije na temelju određene distribucije gustoće ulaznih podataka, anomalijom se smatraju podaci koji se nalaze u području niske gustoće podataka

3. **Metode temeljene na strojnom učenju** (*engl. Machine learning methods*) – anomalije se detektiraju primjenom teorije neizrazitih skupova (*engl. Fuzzy sets*) i teorije grubih skupova (*engl. Rough sets*)

3. Izrada NoSQL podsustava za pohranu i dohvat podataka

Baza podataka je skup međusobno povezanih podataka, pohranjenih zajedno bez štetne ili nepotrebne zalihosti, koje koriste različite aplikacije [19]. Za upravljanje podacima u bazi podataka zadužen je DBMS (*engl. Database Management System*), programski sustav za upravljanje bazama podataka. Sustav za upravljanje bazama podataka omogućava korisnicima kreiranje baza podataka po vlastitoj želji, brine o smještaju podataka, pruža usluge administriranja i zaštite podataka od neovlaštenog pristupa te prima upite prema bazi podataka i isporučuje zatražene podatke.

Baze podataka imaju ključnu ulogu u informacijskim sustavima. Bez baza podataka i sustava za upravljanje bazama podataka efektivno upravljanje podacima ne bi bilo moguće, svi zadaci koje obavljaju baze podataka bi se morali ručno izvoditi. Pojavom računala nastala je i potreba za bazama podataka te su relacijske baze podataka temeljene na SQL jeziku postigle veliki značaj zbog svojih karakteristika poput brzine izvođenja zadataka, pouzdanosti, skalabilnosti i intuitivnosti jezika. Međutim, popularizacijom internetskih društvenih mreža i primjenom agilnog razvoja softvera kada se često događaju manje izmjene, NoSQL baze podataka su došle do izražaja. jer se pojavila značajnija potreba za pohranom nestrukturiranih podataka i prilagodljivom shemom podataka [20].

NoSQL baze su kraće vrijeme u primjeni od već ustaljenih SQL baza podataka i unatoč tome, zbog svojih prednosti postale su prihvaćene u mnogim sustavima. Prema DB-Engines statistici prikazanoj u tablici 1, brzo su se svrstale u 10 najpopularnijih sustava baza podataka.

Tablica 1: Ljestvica najpopularnijih baza podataka

RANG LJESTVICA LIPANJ 2020.	***DBMS***	***TIP BAZE PODATAKA***
1.	Oracle	Relacijska
2.	MySQL	Relacijska
3.	Microsoft SQL Server	Relacijska
4.	PostgreSQL	Relacijska
5.	MongoDB	Dokument
6.	IBM Db2	Relacijska
7.	Elasticsearch	*Search engine*
8.	Redis	*Key-value*
9.	SQLite	Relacijska
10.	Cassandra	*Wide column*

Izvor: [21]

3.1. SQL baze podataka

SQL je upitni jezik koji se koristi za upravljanje relacijskim bazama podataka. Standardiziran je od strane ANSI (*engl. American National Standards Institute*) i ISO (*engl. International Organization for Standardization*) standardizacijskih organizacija. SQL jezik se koristi za dodavanje, ažuriranje ili brisanje podataka pohranjenih u bazi, kao i za dohvat pohranjenih podataka. Sustavi poput *Oracle Database, PostgreSQL, MySQL* i *Microsoft SQL Server* su poznati sustavi za upravljanje bazama podataka koji koriste SQL jezik kao osnovu za rad s relacijskim bazama podataka. Svaki od spomenutih sustava ima svoju varijantu SQL jezika koje se koriste samo u njihovom sustavu baza podataka. Te razlike u varijantama variraju od razlika u sintaksi do razlika u namjeni i funkcionalnosti baza podataka.

Relacijske baze podataka za pohranu podataka koriste tablice (relacije). Podaci su organizirani po tablicama, a svaka tablica se sastoji od određenog broja redaka i stupaca (atributa). Atributi su određeni tipom podataka i rasponom vrijednosti koje mogu primati. Redci u tablici se nazivaju n-torke i pohranjuju sve vrijednosti atributa. Struktura tablica je strogo definirana i podaci koji se unose u tablicu moraju slijediti definiranu strukturu.

Svaki redak tablice se označava sa jedinstvenim identifikatorom koji se naziva primarnim ključem (*engl. Primary key*). Veze između tablica se ostvaruju pomoću stranog ključa (*engl. Foreign key*). Strani ključ predstavlja primarni ključ jedne tablice, koji se kao veza prema svojoj originalnoj tablici javlja u drugoj tablici [19]. Takav pristup organizaciji i označavanju podataka omogućava da se na jednostavan način povezivanjem tablica dobiju složeni podaci bez reorganizacije tablica i same baze podataka.

Podaci u relacijskoj bazi podataka imaju ACID svojstva koja pružaju pouzdanost transakcije. Akronim ACID dolazi od riječi ***A**tomicity, **C**onsistency, **I**solation* i ***D**urability* koje imaju sljedeća značenja [22]:

- ***Atomicity*** – nedjeljivost transakcije, transakcija se mora izvršiti ili se uopće ne smije izvršiti
- ***Consistency*** – konzistentnost, transakcija ne može ostaviti podatke u nekonzistentnom stanju
- ***Isolation*** – izolacija, transakcije se ne mogu međusobno ometati
- ***Durability*** – izdržljivost, ako se transakcija izvršila njezin učinak se ne smije izgubiti ako se dogodi kvar servera ili ponovno pokretanje servera

3.2. NoSQL baze podataka

NoSQL baze su nastale kao odgovor na popularizaciju internetskih usluga i potrebu pohranjivanja i obrade velike količine podataka gdje su tradicionalne tehnologije baza podataka počele pokazivati svoje nedostatke. Naziv NoSQL dolazi od riječi *Not only SQL* i odnosi se na baze podataka koje nisu relacijske baze podataka. Budući da naziv NoSQL obuhvaća sve što nisu relacijske baze podataka razlikuju se četiri osnovne kategorije NoSQL baza podataka [23]:

- ***Key-values pohrana*** – najjednostavniji oblik pohrane u NoSQL bazama podataka. Za pohranu podataka se koriste *key – value* parovi, gdje *key* predstavlja identifikacijski ključ, a *value* predstavlja pohranjenu vrijednost koja može biti znakovni niz (*string*), broj, ali i čitav set novih *key-value* parova

- ***Column-oriented database*** – za pohranu se koriste tablice, ali su podaci podijeljeni u više manjih tablica. Takva metoda pohrane podataka daje dobre performanse bazi podataka kad su u pitanju velike količine podataka
- ***Document database*** – podaci se spremaju u obliku dokumenta koji prati određenu strukturu, najčešće JSON format
- ***Graph database*** – najsloženiji oblik pohrane u NoSQL bazama podataka. Koristi se u situacijama kada podaci međusobno imaju veliki broj veza

NoSQL baze podataka su uglavnom dizajnirane da osiguraju dostupnost podataka zbog čega pati konzistentnost podataka. To znači da nanovo upisan podatak u bazu podataka možda neće biti dostupan istoga trenutka, nego će biti vidljiva zadnja inačica podatka što je u suprotnosti s bazama koje održavaju konzistentnost podataka. Baze podataka koje održavaju konzistentnost, prilikom upisa novog podatka u bazu taj podatak čine dostupnim na svim čvorovima ili vraćaju grešku.

Podaci u NoSQL bazi imaju BASE svojstva. BASE akronim dolazi od riječi ***B****asically available*, ***S****oft state* i ***E****ventual consistency*, koje imaju sljedeća svojstva [22]:

- ***Basically available*** – sustav će funkcionirati u većini slučajeva, nastoji se osigurati dostupnost podataka čak i u slučaju poteškoća u radu sustava
- ***Soft state*** – pohranjeni podaci se možda s vremenom izmjene zato što je zanemarena konzistentnost podataka
- ***Eventual consistency*** – pohranjeni podaci ne postižu konzistentnost odmah, nego u nekom kasnijem trenutku, ali ni to nije garantirano

Još jedna značajka NoSQL baza podataka je pohrana nestrukturiranih podataka. Ne postoji definirana standardna shema za pohranu podataka kao kod SQL baza gdje podaci koji se unose moraju slijediti definiranu strukturu, nego administrator baze sam određuje strukturu podataka na način koji najbolje odgovara slučaju primjene [24].

3.3. Usporedba SQL i NoSQL baza podataka

Prilikom odabira između SQL ili NoSQL baze podataka jedan od glavnih faktora odabira je struktura podataka koji se pohranjuju, odnosno jesu li podaci strukturirani ili ne. Međutim, struktura podataka nije i jedini uvjet kojeg treba razmotriti, potrebno je razmotriti i pitanje konzistentnosti podataka, mogućnosti skaliranja baze, mogućnosti postavljanja upita prema bazi, isto tako važno je i pitanje podrške proizvođača sustava. Pregled značajki SQL i NoSQL baza podataka je prikazan u tablici 2.

Tablica 2: Pregled značajki SQL i NoSQL baza podataka

	SQL	***NoSQL***
Struktura podataka	Prikladna za strukturirane podatke	Prikladna za nestrukturirane podatke
Shema	Strogo definirana shema podataka	Shema podataka nije definirana (Dinamička shema podataka)
Skalabilnost	Vertikalna skalabilnost	Horizontalna skalabilnost
Složenost upita	Pogodna za složene upite	Nije pogodna za složene upite
Konzistentnost podataka	ACID svojstva podataka su standard za SQL baze	NoSQL baze najčešće imaju BASE svojstva
Licenca	Otvorenog koda (*engl. Open Source*) i komercijalna rješenja	Uglavnom otvorenog koda
Primjeri baza	MySQL, Microsoft SQL Server, Oracle Database, PostgreSQL	Cassandra, MongoDB, Redis, Hbase

Izvor: [25]

3.4. Pohrana i dohvat podataka

Za izradu podsustava za pohranu i dohvat podataka u ovom radu korištena je NoSQL MongoDB baza podataka koja za pohranu koristi dokumente (*engl. Document type*). Dokumenti se pohranjuju u kolekcije, odnosno skupine dokumenata koji imaju neka zajednička svojstva što u relacijskim bazama podataka odgovara tablicama. Podaci se spremaju pomoću BSON (*engl. Binary JSON*) formata koji predstavlja nadogradnju standardnog JSON (*engl. JavaScript Object Notation*) formata. BSON format pruža mogućnost pohrane formata poput *int, float point, decimal, long* i *date* koji nisu podržani u JSON formatu. Na slici 1 prikazan je primjer dokumenta u BSON formatu.

```
{
    ime: "Željko",
    prezime: "Majstorović",
    godina_rodjenja: 1994,
    jmbag: "0135232496",
    fakultet: "Fakultet prometnih znanosti",
    grad: "Zagreb"
    studij:{
        razina: "diplomski studij",
        smjer: "Informacijsko - komunikacijski promet",
        status: "redoviti student" }
}
```

Slika 1: Primjer dokumenta u BSON formatu

Spremanje podataka u BSON formatu ima sljedeće prednosti [26]:

- **Jasna struktura podataka** – Nema potrebe za *JOIN* naredbama niti pohranjivanja podataka u tablice što rezultira kraćim i jednostavnijim kodom i boljim performansama baze podataka
- **Prilagodljiva shema -** podaci koji se pohranjuju ne moraju biti strukturirani, strukturu dokumenta je moguće promijeniti u bilo kojem trenutku
- **Distribuiranost –** dokumenti pohrane su neovisni što olakšava distribuciju podataka na više servera
- **Univerzalnost JSON formata –** zbog jednostavnosti i lake čitljivosti ljudima, JOSN format je postao standard za razmjenu podataka i pohranu

Indeksiranje podataka

Podaci pohranjeni u bazu se indeksiraju što omogućava učinkovitu pretragu baze podataka. Kada podaci u bazi nisu indeksirani, MongoDB mora izvršiti skeniranje kolekcije dokumenata (*engl. Collection scan*), odnosno pretražiti svaki dokument u bazi podataka i provjeriti postoji li zapis koji odgovara postavljenom upitu prema bazi podataka. Primjenom indeksa se ograničava broj dokumenata koje treba pretražiti i ubrzava pretraga podataka.

Indeksiranje u MongoDB bazi se može izvoditi automatski gdje sustav sam određuje jedinstveni indeks zapisa prilikom unosa u bazu podataka. Indeks također može biti korisnički definiran na određeni atribut (slika 2) ili više atributa (slika 3) ovisno o slučaju primjene. Kreiranje indeksa na određeni atribut izvodi se pomoću naredbe *createIndex()* i sortiraju se uzlazno ako je vrijednost koja se predaje 1 ili silazno ako je vrijednost -1 [27].

```
db.studenti.createIndex( {fakultet:1} )
```

Slika 2: Kreiranje indeksa na jedan atribut

```
db.studenti.createIndex(
{ fakultet: 1, ime: -1 }
)
```

Slika 3: Kreiranje indeksa na više atributa

Jednom kreiran indeks više nije moguće mijenjati. Da bi se izmijenio potrebno ga je obrisati i ponovo kreirati naredbom *dropIndex()* kako je prikazano na slici 4 [27].

```
db.studenti.dropIndex( {fakultet: 1} )
```

Slika 4: Brisanje indeksa iz baze podataka

U bazi podataka informacijskog sustava postavljeni su indeksi na atribute *_id, origin_id* i *destination_id*. Indeksima su automatski dodijeljena imena *_id_* i *origin_id_1_destination_id_1* što omogućava lakši dohvat podataka kako je prikazano na slici 5. Indeks imena *origin_id_1_destination_id_1* dodan je iz razloga što se podaci u aplikaciji dohvaćaju korištenjem jedinstvenog identifikatora prometnice. Za potrebe izrade prijelaznih matrica brzina, identifikatori izvorišnih linkova su spremljeni u atribut naziva *origin_id*, dok su identifikatori odredišnih linkova spremljeni u atribut naziva *destination_id*.

```
db.spatialMatrix5080.createIndex(
    {
        "key" : { "_id" : 1},
        "name" : "_id_"
    },
    {   "key" : { "origin_id" : 1,
                  "destination_id": 1
                },
        "name" : "origin_id_1_destination_id_1"
    }
)
```

Slika 5: Indeksiranje u bazi podataka informacijskog sustava

Unos novog zapisa u bazu podataka

Za unos podataka u MongoDB bazu koristi se naredba *insertOne()* ili *insertMany()*. Kao što se može zaključiti iz naziva naredbi, *insertOne()* se koristi za unos jednog dokumenta u bazu, dok se *insertMany()* koristi za unos više dokumenata u bazu [27]. Kao što je prethodno rečeno, kod NoSQL ne postoji strogo definirana struktura podataka kao kod SQL baza podataka, nego korisnik sam određuje strukturu podataka u bazi. Ako ne postoji kolekcija u koju se unose podaci ona se automatski kreira prilikom unosa dokumenta u bazu. Primjer koda za unos podataka u bazu je prikazan na slici 6.

```
db.studenti.insertOne(
    {
        ime: "Željko",
        prezime: "Majstorović",
        godina_rodjenja: 1994
    }
)
```

Slika 6: Primjer koda za unos novog zapisa u bazu podataka

Dohvat postojećeg zapisa

Za dohvat postojećeg zapisa koristi se naredba *find()*, uz *find* naredbu mogu se postavljati uvjeti i pretrage u JSON formatu [27]. Primjer *find* upita koji pronalazi sve studente s godinom rođenja 1994. prikazan je na slici 7.

```
db.studenti.find(
{
    godina_rodjenja: "1994"
}
)
```

Slika 7: Primjer koda za dohvat postojećeg zapisa u bazi podataka

Ažuriranje zapisa

Ažuriranje zapisa u bazi podataka se izvršava pomoću naredbe *updateOne()* ili *updateMany()*. Kao i kod unosa podataka, *updateOne()* se koristi za ažuriranje jednog dokumenta u bazi podataka dok se *updateMany()* koristi za ažuriranje više dokumenata u bazi podataka [27]. Na slici 8 prikazan je primjer koda za ažuriranje unosa u bazi podataka. Primjer koda pronalazi studenta s odgovarajućim JMBAG-om i mijenja naziv grada u „Samobor“.

```
db.studenti.updateOne(
    {   jmbag: "0135232496" },
    {   $set: {grad: Samobor}}
)
```

Slika 8: Primjer koda za ažuriranje zapisa u bazi podataka

Brisanje zapisa iz baze podataka

Brisanje zapisa iz baze podataka izvodi se pomoću naredbi *deleteOne()* ili *deleteMany()*. Opet, kao i kod ažuriranja, *deleteOne()* briše jedan dokument iz baze dok *deleteMany()* briše više dokumenata iz baze podataka prema zadanim uvjetima [27]. Primjer koda s *deleteOne* naredbom prikazan je na slici 9, prikazani kod briše studenta s odgovarajućim JMBAG-om.

```
db.studenti.deleteOne(
    { jmbag: "0135232496" }
)
```

Slika 9: Primjer koda za brisanje zapisa iz baze podataka

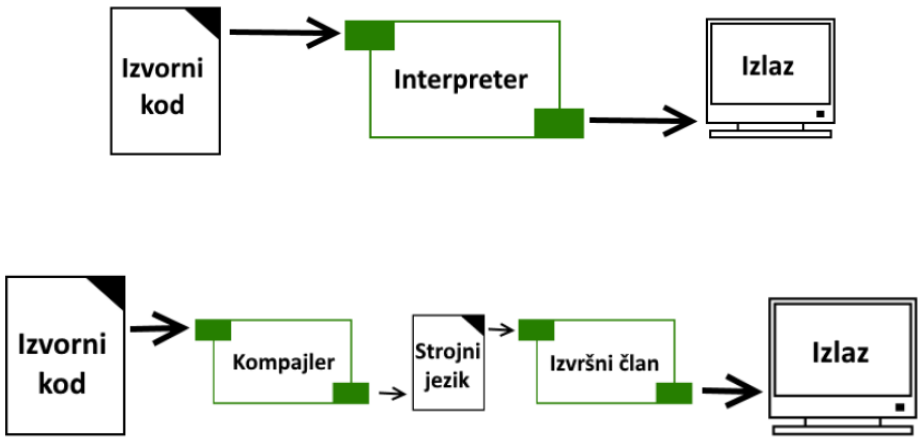
4. Aplikacija za detekciju anomalija na urbanim prometnicama

Za izradu aplikacije za detekciju anomalija na urbanim prometnicama korišten je programski jezik Python te su u aplikaciju ugrađeni svi potrebni algoritmi za uspješnu detekciju i vizualizaciju anomalija.

4.1. Programski jezik i alati

Programski jezik Python je viši programski jezik (*engl. High-level*) otvorenog koda *(engl. Open Source)*, odlikuje ga jednostavnost korištenja i interoperabilnost s drugim programskim jezicima. Python je svestran programski jezik, primjenjiv u gotovo svim područjima od izrade web stranica, izrade aplikacija i video igrica do primjena u robotici i automatizaciji. Jedna od značajki Python programskog jezika je i portabilnost što znači da kôd napisan u Python programskom jeziku se jednako izvršava na različitim platformama s minimalnim ili čak nikakvim izmjenama izvornog koda.

Za razliku od programskog jezika C i sličnih programskih jezika koji koriste *compiler*, Python je programski jezik koji koristi *interpreter*. Kod kompajlerskih jezika izvorni kôd se prevodi na jezik razumljiv računalu (strojni jezik) i jednom kompajliran kôd nije potrebno ponovo kompajlirati. Kompajlerske programske jezike odlikuje velika brzina izvođenja nakon što su jednom prevedeni na strojni jezik. Interpreterski jezik poput Python-a se prevodi u oblik razumljiv računalu prilikom svakog pokretanja programa i takav način izvođenja programskog koda može biti vremenski zahtjevan [28]. Razlike između kompajlerskog i interpreterskog jezika grafički su prikazane na slici 10.



Slika 10: Grafička usporedba compilerskog i interpreterskog programskog jezika [29]

Python programski jezik posjeduje i standardne biblioteke koje omogućavaju funkcionalnost poput analize teksta. Osim standardnih biblioteka ugrađenih u Python, funkcionalnost programskog jezika moguće je proširiti s dodatnim bibliotekama ili softverima trećih strana.

Za obradu podataka u ovom diplomskom radu su korištene sljedeće biblioteke:

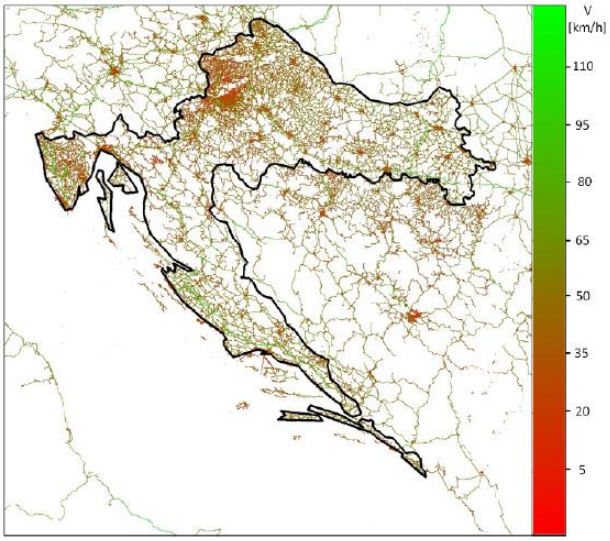
- ***Numpy*** – biblioteka koja omogućava rad s matricama i izvođenje osnovnih operacija nad matricama [30]
- ***Matplotlib*** – biblioteka koja omogućava grafičku vizualizaciju podataka [31]
- ***Xlrd*** – biblioteka koja omogućava čitanje podataka iz datoteka s ekstenzijom .xls i .xlsx [32]
- ***Xlswriter*** – biblioteka koja omogućava kreiranje i zapisivanje podataka u .xls ili .xlsx datoteku [33]
- ***OpenCV*** – biblioteka koja omogućava obradu slika i videozapisa te prepoznavanje objekata na slikama [34]
- ***PIL*** – biblioteka koja pruža funkcionalnost obrade slika i rada sa slikama [35]

4.2. Korišteni podaci

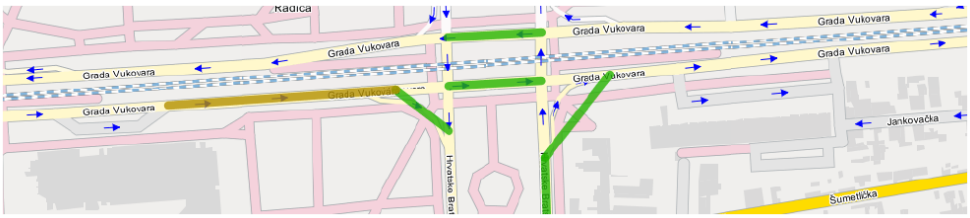
Ključan preduvjet za detekciju anomalija i analizu prometa na cestovnim prometnicama su prikupljeni i kvalitetni podaci. Za izradu ovog rada korišteni su GPS podaci prikupljeni kroz SORDITO projekt. Tijekom trajanja projekta prikupljeno je približno 7 milijardi GPS zapisa od približno 4200 vozila na teritoriju Republike Hrvatske u razdoblju od pet godina. Podaci su bilježeni približno svakih 100 m kada je vozilo u pokretu te približno svakih 5 min kada vozilo stoji. Prikupljeni podaci sadržavaju geografske koordinate, smjer kretanja i brzinu vozila, a prikupljali su se pomoću navigacijskih uređaja ugrađenih u vozila.

Prometnice su podijeljene na manje segmente (linkove), svaki link je omeđen s dva raskrižja, a u slučaju da je prometnica jako duga i nema raskrižja, moguće je da takva prometnica ima više linkova. Svaki link je definiran jedinstvenim identifikatorom (ID-em) koji se razlikuje za svaki smjer kretanja, te dvije koordinate koje označavaju početnu i završnu točku

svakog linka [36]. Slika 11 prikazuje prikupljene podatke na karti, a primjer linkova prikazan je na slici 12.



Slika 11: Prikaz prikupljenih GPS zapisa na karti [29]

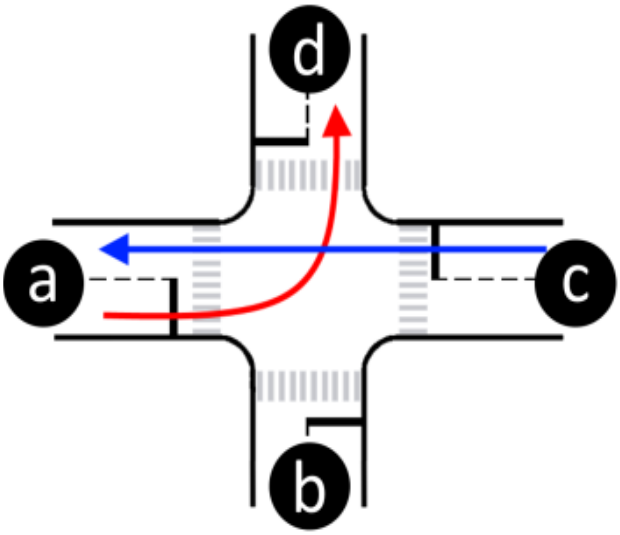


Slika 12: Primjer linkova na prometnicama [29]

Ulazni podaci su prikazani prijelaznim matricama brzina (*engl. Speed Transition Matrix* – STM) [37]. Prijelazna matrica brzina opisuje vjerojatnost da će promatrana vozila prilikom prijelaza s jednog cestovnog segmenta na drugi ostvariti brzinu koja je prikazana matricom.

Prijelaz se može definirati kao promjena putanje vozila tijekom putovanja između dva uzastopna segmenta ceste u vremenskom intervalu t. Primjer prijelaza između segmenata a i d (označen crvenom bojom) i između segmenata c i a (označen plavom bojom) prikazan je na

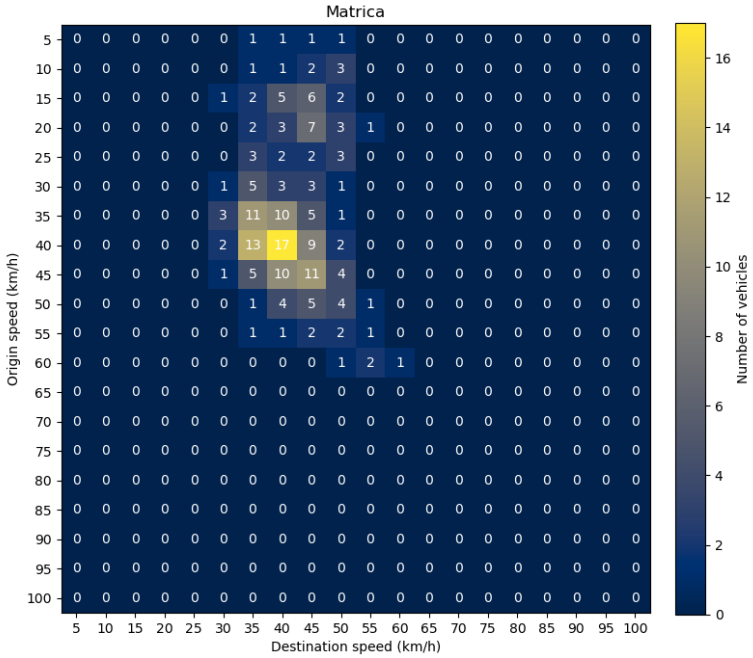
slici 13. Prosječna brzina na ulaznom segmentu ceste označena je kao ulazna brzina v_0, a prosječna brzina na izlaznom segmentu označena je kao izlazna brzina v_d.



Slika 13: Primjer prijelaza između cestovnih segmenata

Ulazne brzine v_0 su prosječne brzine na segmentima a i c, a izlazne brzine v_d su prosječne brzine na segmentima a i d. Pojedini iznosi brzine v_0 i v_d se broje za određeni period vremena i zapisuju u matricu, zatim se izbrojani iznosi brzina pretvaraju u distribuciju vjerojatnosti prijelaznih brzina kako bi se dobila vjerojatnost za svaki prijelaz.

Na x-osi matrice su izlazne brzine, a na y-osi matrice su ulazne brzine. Matrica je veličine 20×20, a veličina matrice ovisi o rezoluciji brzine i maksimalnoj brzini koja može biti zabilježena. U ovom slučaju odabrana je rezolucija od 5 km/h, dok je maksimalna brzina 100 km/h jer su podaci zabilježeni na prometnicama s ograničenjima brzine u rasponu od 50 do 80 km/h. Primjer matrice prijelaznih brzina sa prikazan je na slici 14.



Slika 14: Primjer matrice prijelaznih brzina

4.3. Obrada slika

Digitalna obrada slika može se definirati kao primjena računalnih algoritama kako bi se poboljšala kvaliteta slike ili kako bi se iz tih slika dobile korisne informacije. U digitalnom smislu slika je definirana kao dvodimenzionalna funkcija $f(x, y)$ od konačnog broja elemenata, gdje x i y predstavljaju prostorne koordinate, a vrijednost funkcije $f(x, y)$ predstavlja intenzitet, odnosno vrijednost svakog piksela na određenim koordinatama x i y [38]. U matričnom zapisu slika ima sljedeći zapis (1):

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,n-1) \\ \vdots & \vdots & \vdots & & \vdots \\ f(m-1,0) & f(m-1,1) & f(m-1,2) & \dots & f(m-1,n-1) \end{bmatrix} \quad (1)$$

gdje oznake imaju sljedeća značenja:

- $f(x, y)$ – dvodimenzionalna funkcija
- m – broj redaka u matrici
- n – broj stupaca u matrici

Slike mogu biti zapisane u različitim formatima pa tako razlikujemo sljedeće formate [39][38]:

- **Binarna slika** – kod binarne slike pikseli poprimaju samo dvije vrijednosti – 0 ili 1. Pikseli koji imaju vrijednost 1 na slici su predstavljeni bijelom bojom, a pikseli koji imaju vrijednost 0 poprimaju crnu boju

- **Crno-bijela slika** – slika koja se sastoji samo od crne i bijele boje

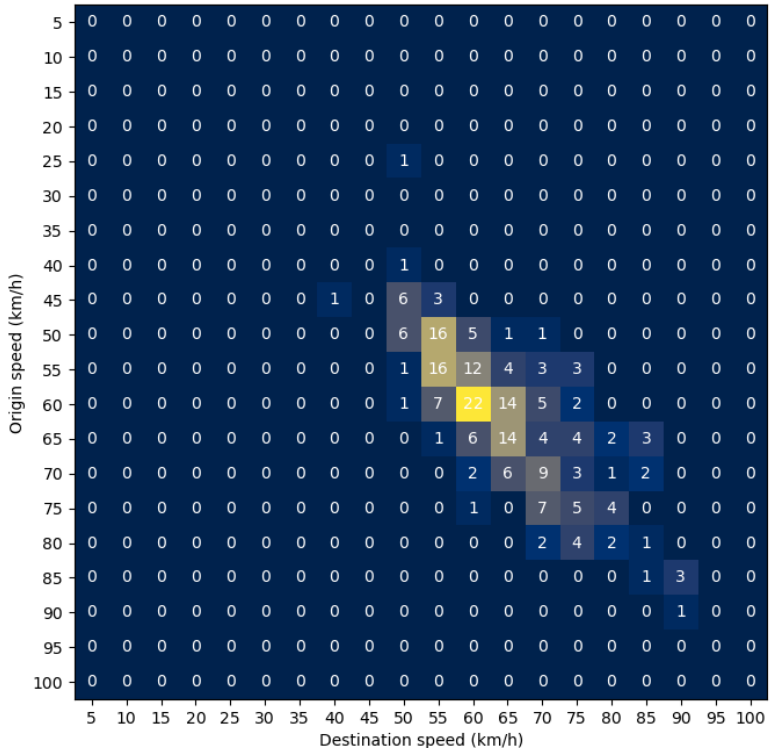
- **8bit-ni format boja** – često korišten format slike kod kojeg se boje prezentiraju u 256 različitih nijansi, gdje 0 predstavlja crnu boju, 255 predstavlja bijelu boju, a 127 predstavlja sivu boju. Također, za ovaj format se koristi i naziv *Grayscale Image*

- **16bit-ni format boja** – ovaj format je poznat kao *High Color format,* razlikuje 65536 boja i koristi se kod zapisa slike u RGB formatu

- **24bit-ni format boja** – slično kao prethodni, ovaj format se također koristi kod zapisa slike u RGB formatu, gdje svaku boju opisuje 256 različitih nijansi. Ovaj format je poznat i pod nazivom *True color format*

- **32bit-ni format boja –** ovaj format zapisa koristi uz tri osnovne komponente RGB koje zapisuje u 24bit-a koristi još i 8bit-ni alfa kanal (*engl. Alpha channel*), poznat je pod nazivom RGBA

Slike koje će se obrađivati imaju rezoluciju 20x20 piksela i boje su zapisane 32bit-nom formatu, a primjer slike je prikazan na slici 15.



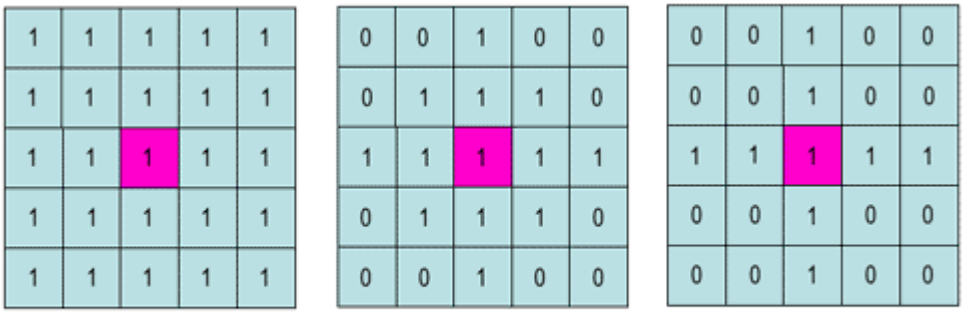
Slika 15: Primjer slike za obradu

4.3.1. Morfološko otvaranje i zatvaranje

Morfološko otvaranje i zatvaranje su operacije koje spadaju u kategoriju morfoloških operacija za obradu slika. Morfološke operacije obrađuju slike na temelju oblika prikazanih na slici i strukturnog elementa (*kernel-a*) koji određuje koliki broj piksela se promatra. Kod morfoloških operacija vrijednost svakog piksela na izlaznoj slici je dobivena usporedbom piksela ulazne slike sa susjednim pikselima [40].

Strukturni element

Strukturni element ili *kernel* je matrica, odnosno binarna slika koja se sastoji od nula ili jedinica. Dimenzije matrice određuju veličinu *kernel-a* i najčešće su veličine 3x3 ili 5x5. Uzorak nula i jedinica određuje oblik *kernela* pa tako *kernel* može imati kvadratni oblik, oblik romba ili oblik plusa [40], kako je prikazano na slici 16.



Strukturni element kvadratnog oblika | Strukturni element u obliku dijamanta | Strukturni element u obliku plusa

Slika 16: Mogući oblici kernela u morfološkim metodama [41]

Morfološka erozija

Kod morfološke erozije svi pikseli blizu granice objekta i pozadine će poprimiti vrijednost 0, ovisno o veličini kernela te se na taj način smanjuje „debljina“ objekta kako to prikazuje slika 17. Logika kernela funkcionira na sljedeći način: promatrani piksel na slici će imati vrijednost 1 samo ako svi pikseli koje pokriva kernel imaju vrijednost 1, u suprotnom će poprimiti vrijednost 0. Ova metoda je korisna kada se želi razdvojiti dva objekta ili kad se želi ukloniti šum sa slike [40].

Slika 17: Rezultat primjene postupka erozije [40]

Morfološka dilatacija

Morfološka dilatacija je suprotna eroziji i primjenom morfološke dilatacije površina objekta se poveća (slika 18). Logika kernela funkcionira na način da promatrani piksel poprima vrijednost 1 ako barem jedan piksel koje pokriva kernel ima vrijednost 1. Budući da se veličina objekta smanji nakon primjene erozije, morfološka dilatacija se koristi da bi se veličina objekta vratila u stanje što sličnije početnom stanju [42].

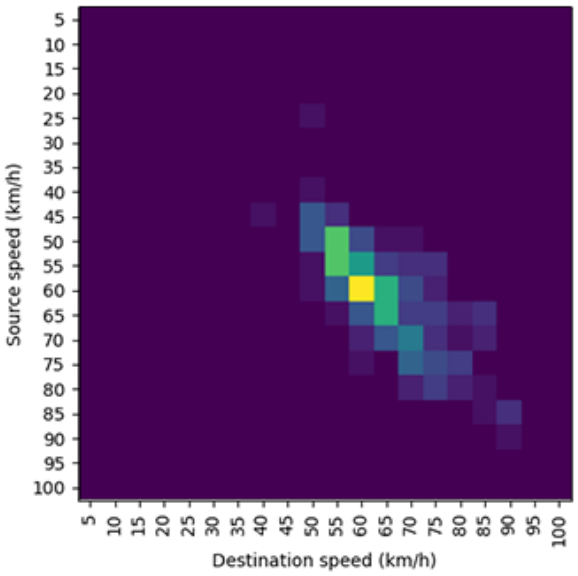
Slika 18: Rezultat primjene postupka dilatacije [40]

Morfološko otvaranje se sastoji od dva koraka, erozije i dilatacije, koji se uzastopno primjenjuju. Morfološko otvaranje može se koristiti za uklanjanje šuma na slici. Morfološko zatvaranje je metoda koja je suprotna otvaranju pa se i koraci primjenjuju obrnutim redoslijedom – dilatacija pa erozija. Morfološko zatvaranje se može koristiti za uklanjanje šuma na objektu. Za primjenu metoda na slikama korišten je kôd prikazan na slici 19, prikazani kôd je dio modula OpenCV za programski jezik Python.

```
img = cv2.imread('j.png',0) #učitavanje slike
kernel = np.ones((5,5),np.uint8) #definiranje postavki kernel-a
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel) #primjena morfološkog otvaranja
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel) #primjena mofrološkog zatvaranja
```

Slika 19: Korišteni kôd za morfološko otvaranje i zatvaranje

Rezultat primjene metoda morfološkog otvaranja i zatvaranja je prikazan na slici 20. Ove metode nisu dale dobar rezultat, a razlog tome je što ove metode ovise o veličini kernela. Morfološko otvaranje i zatvaranje je pogodnije za slike veće rezolucije, dok su slike obrađivane u ovom diplomskom radu rezolucije 20×20 piksela.



Slika 20: Rezultat primjene metoda morfološkog otvaranja i zatvaranja

4.3.2. Metoda praga (*Thresholding*)

Metoda praga (*engl. Thresholding*) se često koristi kada se želi izolirati objekt od pozadine. Primjena metode praga podrazumijeva usporedbu vrijednosti svakog piksela sa određenom razinom praga što u konačnici dijeli piksele slike u dvije grupe [43]:

1. Pikseli s intenzitetom nižim od postavljene razine praga
2. Pikseli s intenzitetom višim od postavljene razine praga

Pikseli s intenzitetom nižim od postavljene razine praga poprimaju vrijednost 0 i na slici se vide kao ljubičasto obojani pikseli, dok pikseli s intenzitetom višim od postavljenog praga poprimaju vrijednost 1 i vidljivi su kao žuto obojani pikseli.

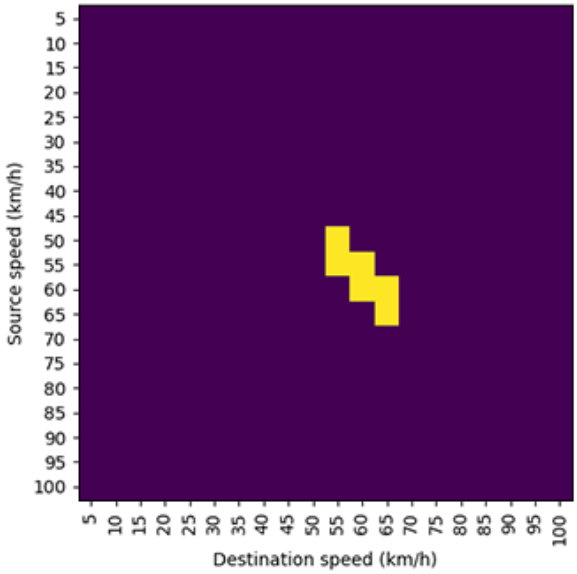
```
def get threshold(Image):          #Funkcija prima vrijednost kao sliku

    thresh = 2                     #Vrijednost praga 2
    im = Image.convert('L')        #Filtriranje slike u crno-bijele nijanse

    imgdata = np.asarray(im)       #Slika se pretvara u brojevni niz
    threshdata = (imgdata>thresh) #Filtriranje vrijednosti
```

Slika 21: Kôd korišten za metodu praga

Primjer koda prikazan na slici 21 daje izlazni rezultat prikazan na slici 22. Metoda praga daje dobre rezultate kada je u pitanju određivanje težišta (opisano u poglavlju 4.4.), ali nedostatak metode je što pikseli poprimaju samo dvije vrijednosti zbog čega se gubi osjećaj o „težini“ filtriranih piksela.



Slika 22: Rezultat primjene metode praga

4.3.3. Metoda sivih tonova

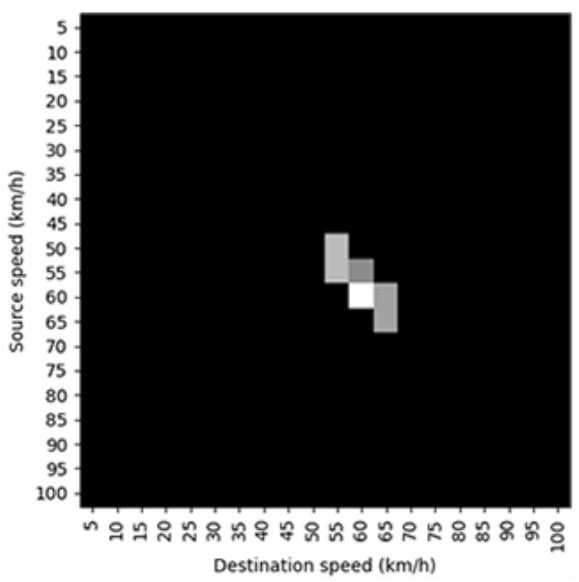
Metoda sivih tonova (*engl. Grayscaling*) je proces pretvaranja slike iz formata u boji poput RGB ili CMYK, u nijanse sive koje variraju od potpuno crne od bijele boje. Većina računalnih algoritama za obradu slika je prilagođena upravo za rad sa slikama pretvorenim u sive tonove.

Pretvaranje u sive tonove ima određene prednosti kada je u pitanju obrada slika, a glavna prednost je pojednostavljenje zapisa jer se smanjuje broj komponenti u zapisu. Standardna RGB slika koristi tri boje što znači da je svaki piksel definiran sa tri komponente, dok se kod slike pretvorene u sive tonove koristi samo jedna komponenta [44].

```
def get_gray(matrix):   #Funkcija prima vrijednost kao matricu
    pil_image=pil.Image.fromarray(np.uint8(matrix))     #Pretvaranje matrice u sliku
    gray = pil_image.convert('L')   #Filtriranje slike u crno-bijele nijanse
    bw = gray.point(lambda x : 0 if x < int(0.2*matrix.max()) else x)   #Filtriranje vrijednosti
return bw   #Funkcija vraća filtrirane vrijednosti
```

Slika 23: Kôd korišten za metodu sivih tonova

Primijenjeni kôd prikazan i objašnjen na slici 23 daje dobre rezultate (slika 24) koji su jako slični metodi praga. Prednost ove metode je u tome što pikseli poprimaju vrijednosti u rasponu 0 – 255, što rezultira gradijentnim prikazom boja, za razliku od metode praga gdje pikseli mogu imati vrijednost 0 ili 1. Takav prikaz daje osjećaj o težini pojedinog piksela što je prednost u odnosu na druge metode.



Slika 24: Rezultat primjene metode sivih tonova

4.4. Određivanje težišta

Određivanje težišta matrice je ključan korak u procesu detekcije anomalija. Temeljem uspješno detektiranog težišta matrice moguće je odrediti Euklidsku udaljenost između dvije matrice. Svaka STM matrica sadrži određene vrijednosti koje mogu biti grupirane ili raspršene

po matrici. Neovisno o tome jesu li podaci grupirani ili raspršeni, težište matrice može biti određeno korištenjem marginalne distribucije tako da se svaka pojedina vrijednost u matrici podijeli s ukupnim zbrojem svih vrijednosti u matrici. Zatim se dobivene vrijednosti zbroje po x i y osi, svaki redak odnosno stupac zasebno i te dobivene vrijednosti predstavljaju marginalnu distribuciju kako je prikazano u tablici 3.

Tablica 3: Primjer marginalnih distribucija

x/y	y_1	y_2	y_3	y_i	$\sum x_i$
x_1	$p(x_1, y_1)$	$p(x_1, y_2)$	$p(x_1, y_3)$	$p(x_1, y_i)$	$p(x_1)$
x_2	$p(x_2, y_1)$	$p(x_2, y_2)$	$p(x_2, y_3)$	$p(x_2, y_i)$	$p(x_2)$
x_3	$p(x_3, y_1)$	$p(x_3, y_2)$	$p(x_3, y_3)$	$p(x_3, y_i)$	$p(x_3)$
x_i	$p(x_i, y_1)$	$p(x_i, y_2)$	$p(x_i, y_3)$	$p(x_i, y_i)$	$p(x_i)$
$\sum y_i$	$p(y_1)$	$p(y_2)$	$p(y_3)$	$p(y_i)$	1

Koordinata težišta po x osi je očekivana vrijednost $E(p(x_i))$ gdje je $p(x_i)$ marginalna distribucija x osi, dok je koordinata težišta po y osi očekivana vrijednost $E(p(y_i))$ gdje je $p(y_i)$ marginalna distribucija y osi koje se računaju prema formuli (2) i (3) [45]:

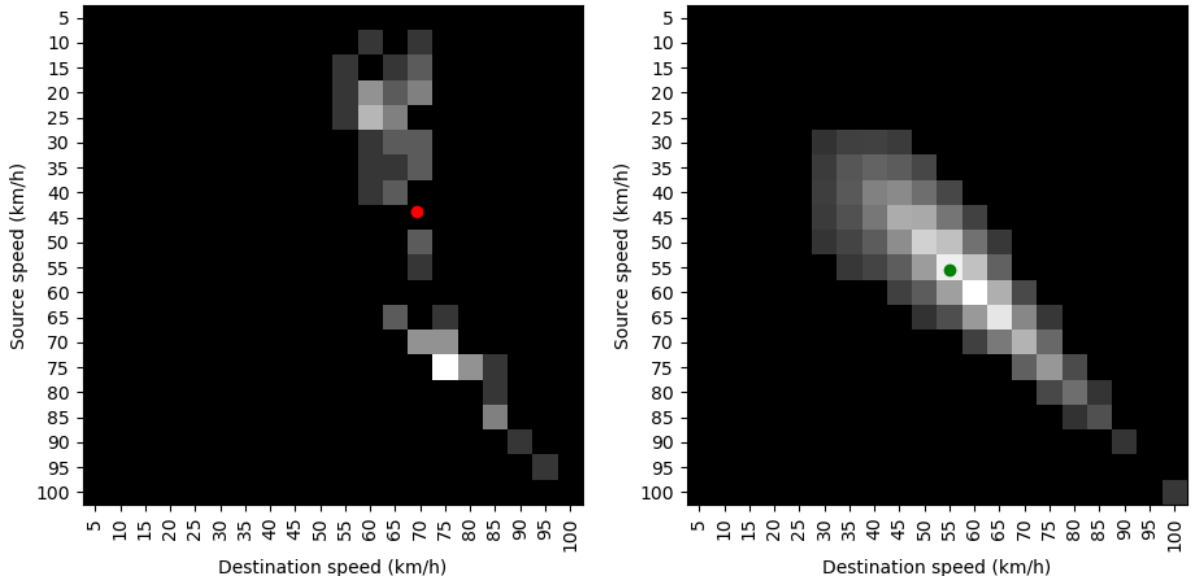
$$C_x = \sum_{i=1}^{20} p_x(x_i) \cdot i \tag{2}$$

$$C_y = \sum_{j=1}^{20} p_y(y_j) \cdot j \tag{3}$$

gdje oznake imaju sljedeće značenje:

- i – indeks x koordinate
- j – indeks y koordinate
- $p_x(x_i)$ – vrijednost marginalne distribucije x_i koordinate
- $p_y(y_j)$ – vrijednost marginalne distribucije y_j koordinate

Primjenom formula (2) i (3) na x odnosno y os matrice, kao rezultat dobiju se koordinate težišta matrice označene kao C_x i C_y, kao što to prikazuje primjer matrice sa određenim težištem na slici 25 a).



a) STM s određenim težištem (crvena točka) *b) Median STM kao referentna matrica*

Slika 25:Primjeri STM s određenim težištima

Slika 25 b) prikazuje median matricu s određenim težištem matrice (zelena točka). Median matrica će biti korištena kao referentna matrica koja predstavlja stanje normalnog odvijanja prometa. Median matrica je izračunata kao median vrijednost svih matrica prijelaznih brzina, karakteristika median matrice su podaci grupirani oko sredine matrice i prema donjem desnom kutu, što znači da su brzine prometnog toka blizu dopuštenog ograničenja brzine.

4.5. Izračun relativne Euklidske udaljenosti

Na temelju određene pozicije težišta moguće je izračunati euklidsku udaljenost između koordinata težišta promatrane STM i koordinata referente median STM. Euklidska udaljenost je standardna metoda mjerenja udaljenosti koja se koristi u geometriji, a definira se kao najkraća udaljenost između dvije točke u prostoru koja je jedinstvena i računa se po formuli (4) :

$$d\left(c_{STM(i)}, c_M\right) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

gdje oznake imaju sljedeće značenje:

- $d\left(c_{STM(i)}, c_M\right)$ – Euklidska udaljenost
- $c_{STM(i)}$ – promatrana matrica
- c_M – referentna median matrica
- (x_1, y_1) – koordinate težišta promatrane matrice označene kao $c_{STM(i)}$
- (x_2, y_2) – koordinate težišta referentne median matrice označene kao c_M

Kako bi se dobio rezultat koji je lakše interpretirati, Euklidska udaljenost je izražena relativno u odnosu na najveću udaljenost u matrici (dijagonalu) prema formuli (5):

$$d_{rel} = \frac{d\left(c_{STM(i)}, c_M\right)}{20\sqrt{2}} \quad (5)$$

gdje oznake imaju sljedeća značenja:

- d_{rel} – relativna Euklidska udaljenost
- $d\left(c_{STM(i)}, c_M\right)$ – Euklidska udaljenost
- $20\sqrt{2}$ – dijagonala kvadrata

Euklidska udaljenost je podijeljena s $20\sqrt{2}$ iz razloga što matrica ima dimenzije 20×20, a dijagonala matrice se računa kao dijagonala kvadrata po formuli $a\sqrt{2}$, gdje a predstavlja dužinu jedne stranice kvadrata.

4.6. Detekcija anomalija statističkom metodom

Često korištena statistička metoda za detekciju anomalija je metoda interkvartilnog raspona (*engl. Interquartile Range* - IQR) kod koje se podaci dijele na kvartile, a interkvartilni raspon predstavlja razliku između prvog i trećeg kvartila. Drugim riječima, ova metoda mjeri raspršenost podataka oko srednje vrijednosti (mediana) svih podataka, a vrijednosti koje značajnije odstupaju od srednje vrijednosti se smatraju anomalijom.

Postupak primjene statističke metode interkvartilnog raspona se izvodi po koracima kako slijedi [46]:

1. Sortirati podatke po veličini
2. Odrediti prvi i treći kvartil ($Q1$ *i* $Q3$)
3. Odrediti interkvaritlni raspon (IQR)
4. Odrediti granicu za detekciju anomalija

Q1 i Q3 određuju se kao 25. odnosno 75. percentil svih podataka, a IQR se računa prema formuli (6):

$$IQR = Q3 - Q1 \tag{6}$$

gdje oznake imaju sljedeća značenja:

- IQR – interkvartilni raspon
- $Q1$ – prvi kvartil
- $Q3$ – treći kvartil

Gornja granica za detekciju anomalija se računa prema formuli (7):

$$Q3 + 1.5 \cdot IQR \tag{7}$$

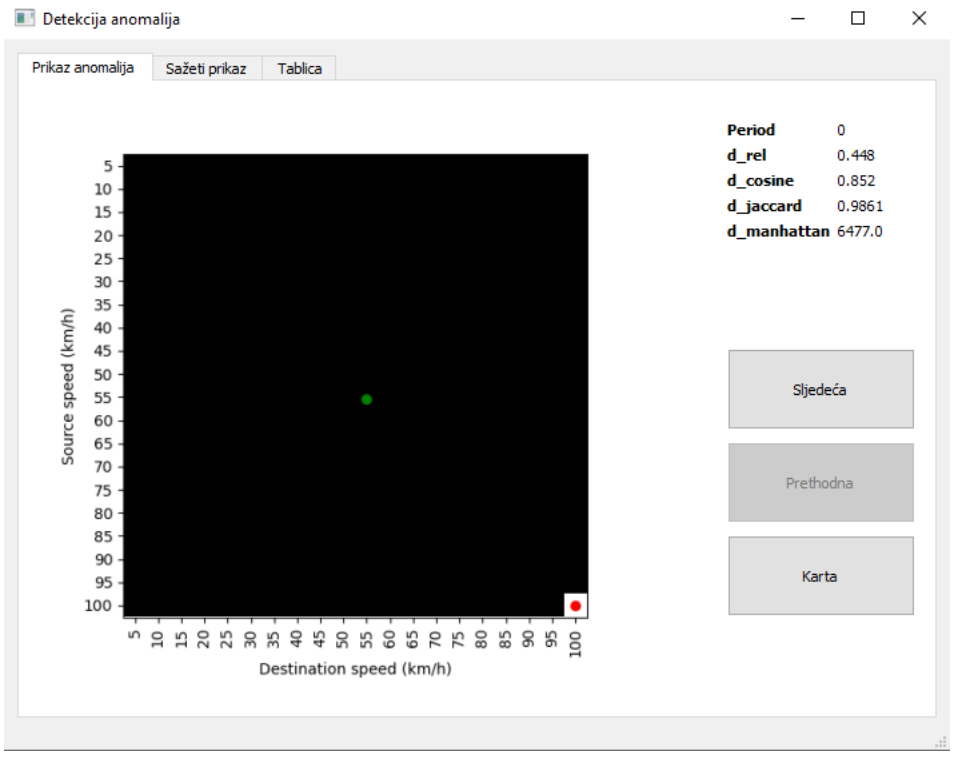
Svaka izračunata udaljenost koja ima vrijednost veću od vrijednosti dobivene izrazom (7) predstavlja anomaliju u podacima.

4.7. Grafičko sučelje informacijsko – komunikacijskog sustava

Grafičko sučelje informacijsko – komunikacijskog sustava izrađeno je pomoću Qt Designer alata i programskog jezika Python. Qt Designer je dio Qt *framework-a* za razvoj i izradu računalnih i mobilnih aplikacija. Qt *framework* je podržan na većini operacijskih sustava poput Linux OS-a, Windows OS-a, Android, iOS-a i ostalih operacijskih sustava, što ga čini odličnim alatom za izradu programskih sučelja.

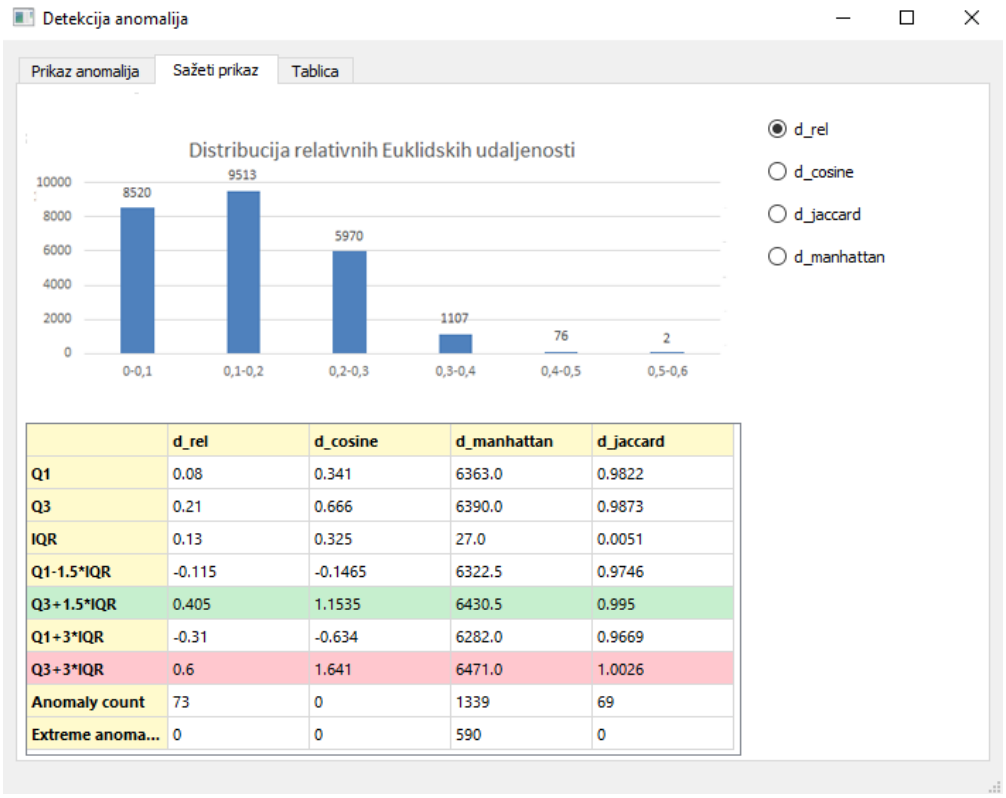
Grafičko sučelje informacijsko – komunikacijskog sustava sastoji se od tri kartice, a to su: Prikaz anomalija, Sažeti prikaz i Tablica. Kartica grafičkog sučelja „Prikaz anomalija“ (slika 26) prikazuje detektirane anomalije s označenim težištima referentne matrice (zelena točka) i promatrane matrice (crvena točka). Uz grafički prikaz detektiranih anomalija prikazuje se i period u kojem je anomalija detektirana te izračunate mjere udaljenosti.

Gumbovi „Sljedeća“ i „Prethodna“ služe za prikazivanje detektiranih anomalija, s tim da se gumbovi uključe odnosno isključe u ovisnosti o tome može li se prikazati sljedeća ili prethodna anomalija. Gumb „Karta“ u web pregledniku učitava interaktivnu kartu s vizualizacijom anomalija na karti grada Zagreba.



Slika 26: Grafičko sučelje - prikaz anomalija

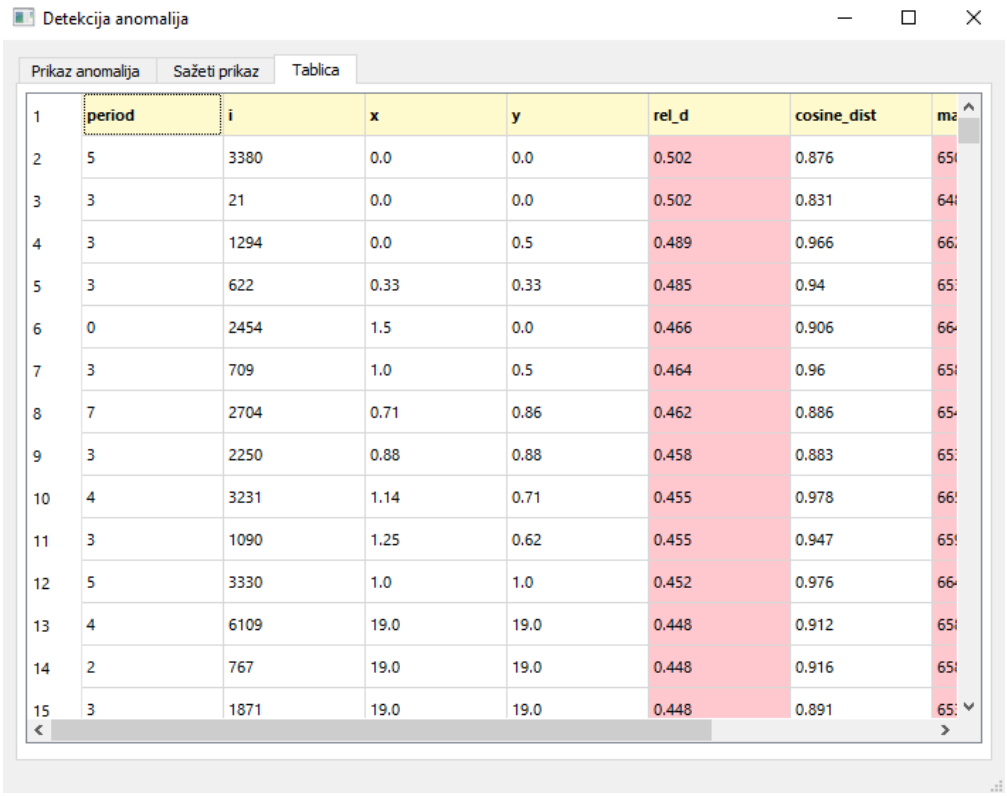
Iduća kartica je „Sažeti prikaz“ (slika 27), u ovoj kartici se prikazuje sažetak obrađenih rezultata. Rezultati statističke obrade svih mjera udaljenosti se prikazuju u obliku tablice s pripadajućim oznakama i bojom istaknutim poljima. U ovoj kartici se također prikazuju i grafovi distribucije za svaku mjeru udaljenosti koje je moguće mijenjati pomoću *radio button* izbornika s desne strane.



	d_rel	d_cosine	d_manhattan	d_jaccard
Q1	0.08	0.341	6363.0	0.9822
Q3	0.21	0.666	6390.0	0.9873
IQR	0.13	0.325	27.0	0.0051
Q1-1.5*IQR	-0.115	-0.1465	6322.5	0.9746
Q3+1.5*IQR	0.405	1.1535	6430.5	0.995
Q1+3*IQR	-0.31	-0.634	6282.0	0.9669
Q3+3*IQR	0.6	1.641	6471.0	1.0026
Anomaly count	73	0	1339	69
Extreme anoma...	0	0	590	0

Slika 27: Grafičko sučelje - sažeti prikaz

Zadnja kartica, „Tablica“ prikazuje sve podatke o obrađenim matricama (slika 28). U ovoj tablici se mogu pronaći podaci poput perioda, rednog broja matrice „i“, x i y koordinata težišta te rezultati mjera udaljenosti za svaku pojedinu matricu. Također, anomalične matrice su istaknute crvenom bojom kako bi se lakše razlikovale za daljnju analizu.



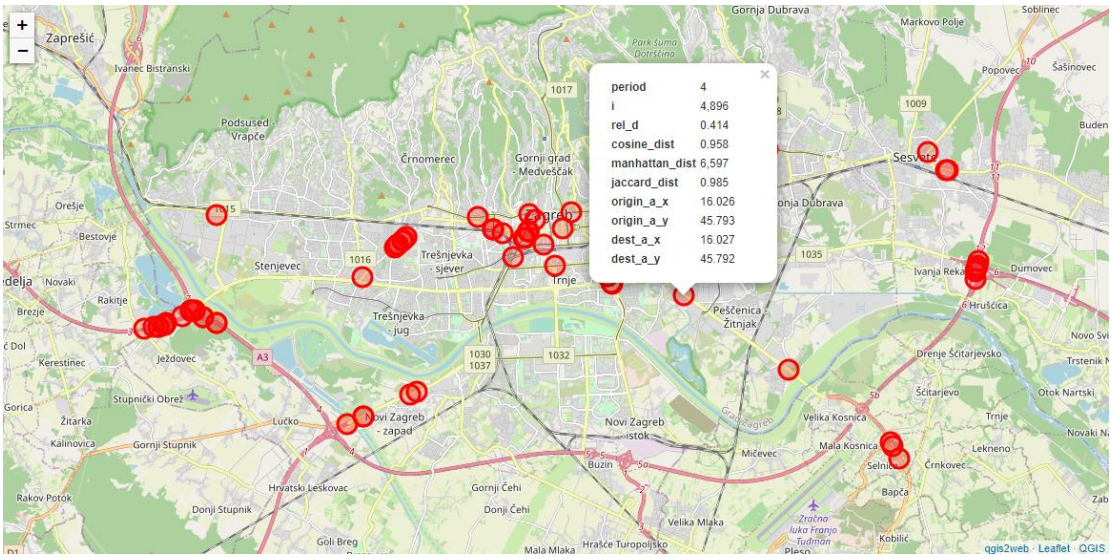
	period	i	x	y	rel_d	cosine_dist	ma
2	5	3380	0.0	0.0	0.502	0.876	65
3	3	21	0.0	0.0	0.502	0.831	64
4	3	1294	0.0	0.5	0.489	0.966	66
5	3	622	0.33	0.33	0.485	0.94	65
6	0	2454	1.5	0.0	0.466	0.906	66
7	3	709	1.0	0.5	0.464	0.96	65
8	7	2704	0.71	0.86	0.462	0.886	65
9	3	2250	0.88	0.88	0.458	0.883	65
10	4	3231	1.14	0.71	0.455	0.978	66
11	3	1090	1.25	0.62	0.455	0.947	65
12	5	3330	1.0	1.0	0.452	0.976	66
13	4	6109	19.0	19.0	0.448	0.912	65
14	2	767	19.0	19.0	0.448	0.916	65
15	3	1871	19.0	19.0	0.448	0.891	65

Slika 28: Grafičko sučelje – tablica

Klikom na gumb „Karta“ u web pregledniku se otvara karta (slika 29) s crveno označenim anomalijama. Prikazana anomalija sadrži i određene metapodatke kojima se može pristupiti klikom na anomaliju.

Vizualizacijom detektiranih anomalija na karti grada Zagreba vidljivo je da su anomalije detektirane u različitim dijelovima grada. Detektirane anomalije se mogu podijeliti u dvije skupine: prva skupina su anomalije detektirane na cestama poput pristupnih cesta gradu koje su smještene na rubnim dijelovima grada i druga skupina su anomalije detektirane na gradskim prometnicama bliže centru grada.

Anomalije detektirane na gradskim pristupnim cestama se pojavljuju tijekom jutarnjeg i popodnevnog vršnog sata, a rezultat su dnevnih migracija stanovnika iz okolnih gradova. Drugi tip anomalija je detektiran na gradskim prometnicama i one su rezultat kretanja stanovništva unutar grada. Te anomalije se pojavljuju tijekom jutarnjeg i popodnevnog vršnog sata, ali i u drugim dijelovima dana. Takav ishod je rezultat velikog broja ljudi koji rade u gradu, velikog broja turističkih posjeta i dostupnosti raznim vrstama zabavnog sadržaja.



Slika 29: Vizualizacija detektiranih anomalija na karti grada Zagreba

5. Evaluacija predložene mjere za udaljenost

Za evaluaciju predložene mjere udaljenosti svi rezultati dobiveni obradom prikupljenih podataka su pohranjeni i obrađeni u Microsoft Excel programskom alatu. Rezultati su analizirani na način da su vrijednosti dobivene Euklidskom metodom mjerenja udaljenosti uspoređeni s mjerama udaljenosti: Manhattan udaljenost, Kosinusna udaljenost i Jaccard udaljenost.

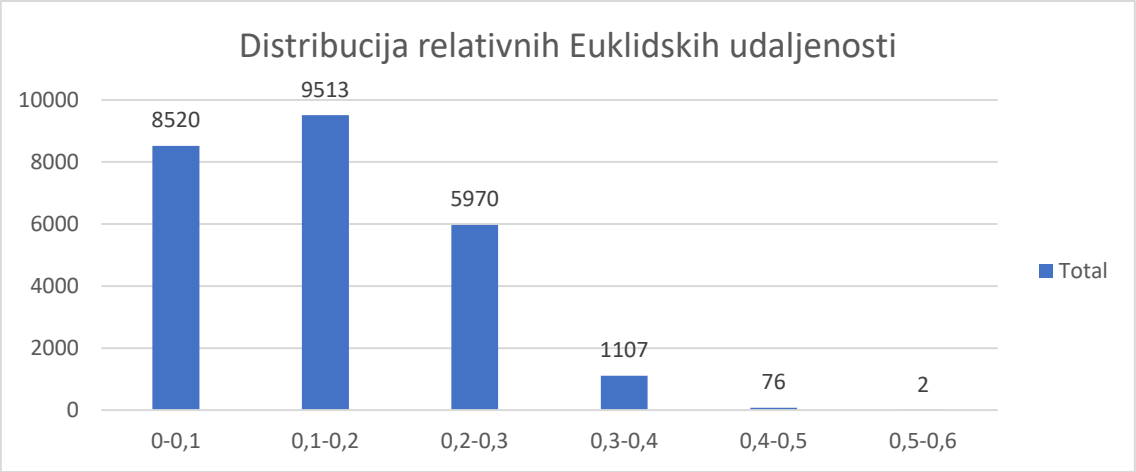
5.1. Relativna Euklidska udaljenost

Izračun euklidske udaljenosti je detaljno opisan u poglavlju 4.4., u ovom dijelu rada su prikazani i analizirani rezultati dobiveni s mjerom Euklidske udaljenosti (tablica 4).

Tablica 4: Rezultati obrade podataka za relativnu Euklidsku udaljenost

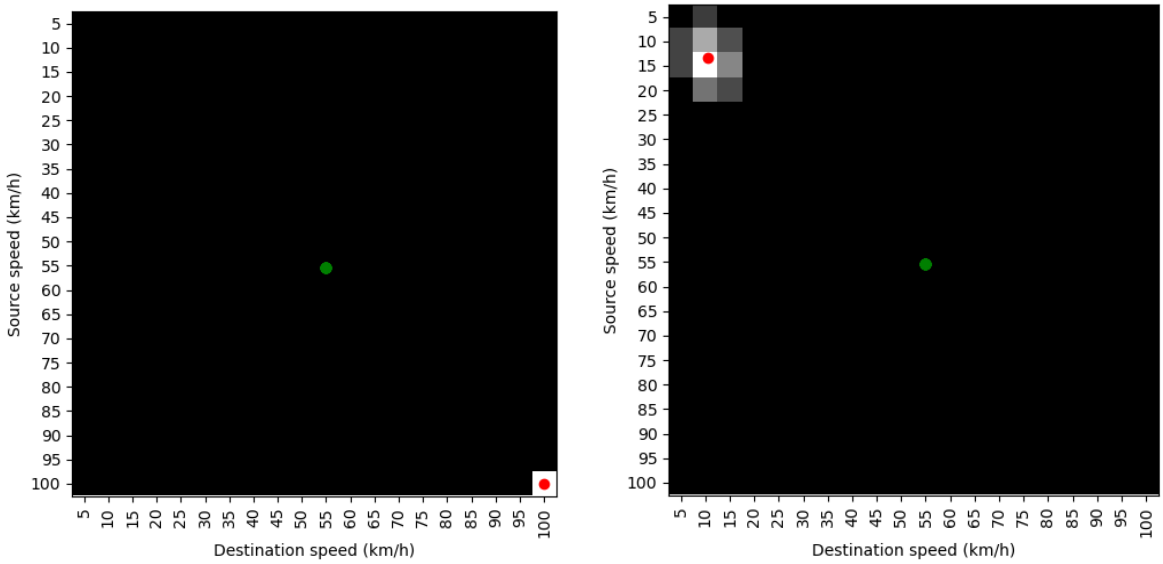
	d_rel	d_manhattan	d_cosine	d_jaccard
Q1	0,08	6363	0,341	0,9822
Q3	0,21	6390	0,666	0,9873
IQR	0,13	27	0,325	0,0051
Q1-1.5*IQR	-0,115	6322,5	-0,1465	0,97455
Q3+1.5*IQR	0,405	6430,5	1,1535	0,99495
Q1-3*IQR	-0,31	6282	-0,634	0,9669
Q3+3*IQR	0,6	6471	1,641	1,0026
Anomaly count	73	1339	0	69
Extreme anomaly count	0	590	0	0

Pomoću relativne Euklidske udaljenosti i primjenom standardne statističke metode za detekciju anomalija uspješno je detektirano 73 anomalije i niti jedna ekstremna anomalija od ukupnih 25188 zapisa s rezultatima u intervalu [0, 0.6]. Anomalijama se smatraju sve matrice koje imaju relativnu Euklidsku udaljenost $d_{rel} > 0{,}405$.



Slika 30: Graf distribucije relativnih Euklidskih udaljenosti

Iz grafa distribucije relativnih Euklidskih udaljenosti prikazanog na slici 30, vidljivo je da je većina udaljenosti grupirano do polovice spektra vrijednosti što neće biti slučaj i kod drugih metoda čije udaljenosti su previše raspršene ili previše stisnute u određenom spektru vrijednosti. Primjeri matrica s detektiranim anomalijama su prikazani na slici 31.



Slika 31: Primjeri matrica s detektiranim anomalijama

5.2. Manhattan udaljenost

Manhattan udaljenost, poznata još kao i *Taxicab* ili *Cityblock distance* dobila je naziv po tome što predstavlja udaljenost koju bi prešao automobil u gradu poput Manhattana, grada s ravnim ulicama koje su međusobno okomite i paralelne, te zgradama raspoređenim u pravilne blokove [47] .

Primjerice, ako su dvije točke A i B smještene u istoj ulici tada se udaljenost između njih računa kao pravocrtna Euklidska udaljenost, a u slučaju da se dvije točke A i B nalaze u različitim ulicama tada se udaljenost računa kao broj blokova koje vozilo mora proći da bi došao od točke A do točke B.

Za razliku od Euklidske udaljenosti koja računa pravocrtnu udaljenost između dvije točke, Manhattan udaljenost računa duljinu puta s obzirom da je dozvoljeno kretanje samo uzduž linija koordinatne mreže koje su paralelne i okomite na x-os. U takvoj geometriji između točaka može postojati više putova koji su istih duljina. Za određivanje Manhattan udaljenosti koriste se vektorizirane matrice, drugim riječima stupci matrice se slažu u stupce jedan ispod drugog te se dvodimenzionalna matrica pretvara u jednodimenzionalni niz. Manhattan udaljenost se računa po formuli (8) [47]:

$$d_M(STM, M) = \sum_{i=1}^{n} |STM(i) - M(i)| \tag{8}$$

gdje oznake imaju sljedeća značenja:

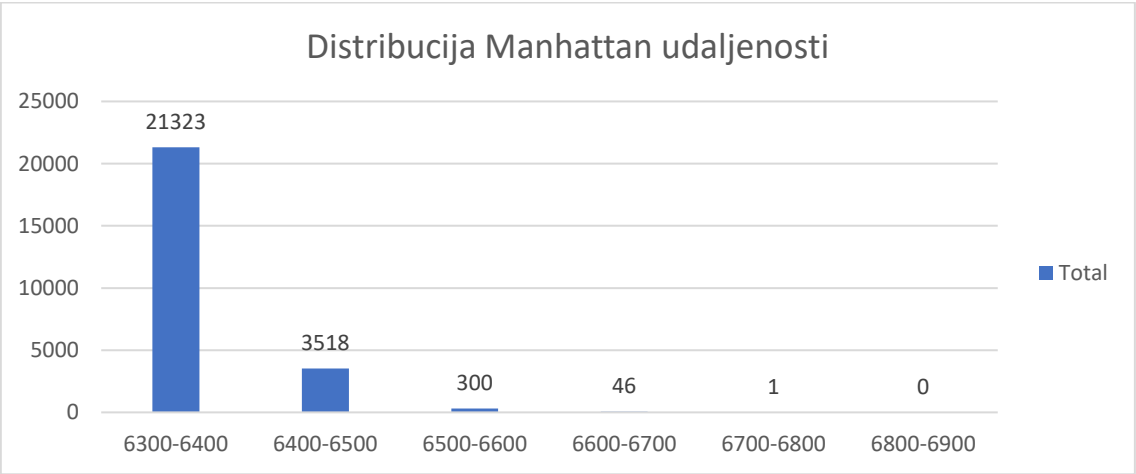
- d_M – Manhattan udaljenost
- $STM(i)$ – i-ta vrijednost vektorizirane STM
- $M(i)$ – i-ta vrijednost vektorizirane median STM

Rezultati obrade podataka za Manhattan udaljenost prikazani su u tablici 5.

Tablica 5: Rezultati obrade za Manhattan udaljenost

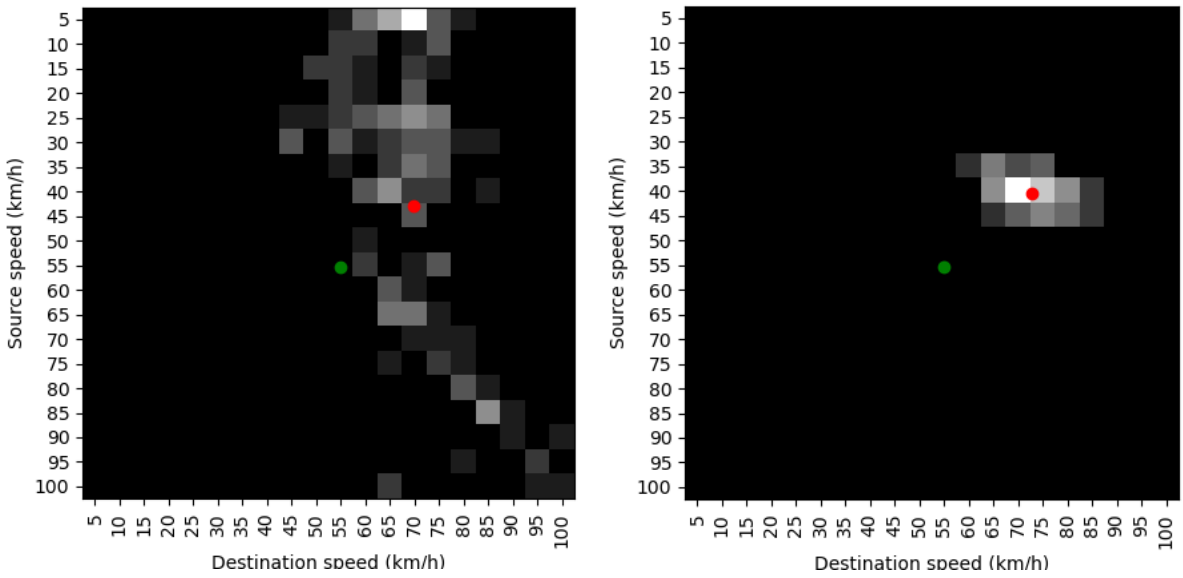
	d_rel	d_manhattan	d_cosine	d_jaccard
Q1	0,08	6363	0,341	0,9822
Q3	0,21	6390	0,666	0,9873
IQR	0,13	27	0,325	0,0051
Q1-1.5*IQR	-0,115	6322,5	-0,1465	0,97455
Q3+1.5*IQR	0,405	6430,5	1,1535	0,99495
Q1-3*IQR	-0,31	6282	-0,634	0,9669
Q3+3*IQR	0,6	6471	1,641	1,0026
Anomaly count	73	1339	0	69
Extreme anomaly count	0	590	0	0

Kod metode Manhattan udaljenosti anomalijom su se smatrale matrice koje su imale udaljenost $d_M > 6430{,}5$ i kao takvih je detektirano 1339 anomalija te 590 ekstremnih anomalija od ukupno 25188 zapisa. Rezultati su u intervalu $[6300, 6800]$ i s obzirom na vrijednosti Q1 i Q3 može se zaključiti da su podaci stisnuti u određenom dijelu spektra vrijednosti.



Slika 32: Graf distribucije Manhattan udaljenosti

Pogledom na graf distribucije Manhattan udaljenosti (slika 32) vidljivo je da su vrijednosti stisnute na početku spektra udaljenosti što sugerira da ova mjera nije prikladna za detekciju anomalija korištenjem STM-a, a tu tvrdnju objašnjava slika 33.



Slika 33: Primjeri matrica s niskim d_{rel} i visokom d_M vrijednošću

Slika 33 prikazuje primjere matrica koje su Manhattan metodom detektirane kao anomalije, ali vidljivo je da su vrijednosti relativne Euklidske udaljenosti niske. Konkretno, relativne Euklidske udaljenosti imaju vrijednost $d_{rel} < 0{,}2$, dok vrijednosti Manhattan udaljenosti iznose $d_M > 6460$.

5.3. Kosinusna udaljenost

Kosinusna udaljenost je metoda koja setove podataka tretira kao vektore. Ako dva seta podataka predstavimo kao vektore, tada sličnost između dva seta podataka odgovara odnosu između dva vektora, odnosno mjerenjem kuta između dva vektora se može odrediti sličnost podataka. Prema [48], mjera Kosinusne udaljenosti je metoda koja zapravo računa kut između dva vektora i računa se po formuli (9):

$$d_C(STM, M) = \frac{STM \cdot M}{\|STM\| \times \|M\|} \tag{9}$$

gdje oznake imaju sljedeće značenje:

- d_C – Kosinusna udaljenost
- STM – vektorizirani skup podataka promatrane matrice
- M – vektorizirani skup podataka referentne median matrice

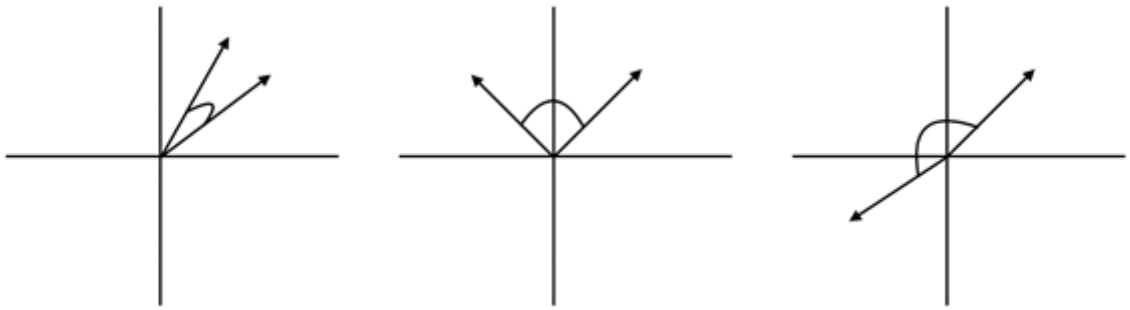
$\|STM\|$ i $\|M\|$ su Euklidske norme vektora STM i M, odnosno dužine ta dva vektora koje se računaju po formuli (10) i (11) [49]:

$$\|STM\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_i^2} \tag{10}$$

$$\|M\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_i^2} \tag{11}$$

gdje x_i predstavlja vrijednosti vektoriziranih matrica.

Rezultat mjere Kosinusne udaljenosti može varirati i intervalu [-1, 1] gdje vrijednost -1 označava da su vektori podataka isti ali suprotno orijentirani, a vrijednost 1 označava potpunu sličnost. Vrijednost 0 znači da su vektori ortogonalni, odnosno da uspoređeni podaci nemaju zajedničkih elemenata.



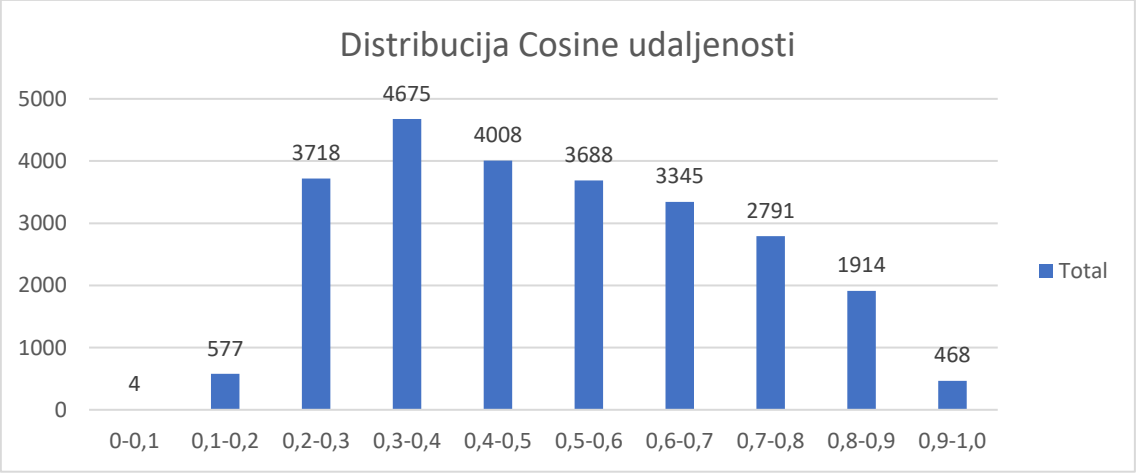
Slika 34: Grafički prikaz usporedbe smjerova vektora [50]

Mjera Kosinusne udaljenosti nije osjetljiva na duljinu podataka, ova metoda samo određuje koliko su slični smjerovi vektoriziranih podataka što je grafički prikazano na slici 34. Kosinusna udaljenost se često koristi kao mjera za usporedbu dokumenata. Rezultati obrade podataka za Kosinusnu udaljenost su prikazani u tablici 6.

Tablica 6: Rezultati obrade za Kosinusnu udaljenost

	d_rel	d_manhattan	d_cosine	d_jaccard
Q1	0,08	6363	0,341	0,9822
Q3	0,21	6390	0,666	0,9873
IQR	0,13	27	0,325	0,0051
Q1-1.5*IQR	-0,115	6322,5	-0,1465	0,97455
Q3+1.5*IQR	0,405	6430,5	1,1535	0,99495
Q1-3*IQR	-0,31	6282	-0,634	0,9669
Q3+3*IQR	0,6	6471	1,641	1,0026
Anomaly count	73	1339	0	69
Extreme anomaly count	0	590	0	0

Za razliku od Euklidske udaljenost, metoda Kosinusne udaljenosti nije detektirala niti jednu anomaliju, a razlog tome su značajno veće vrijednosti za Q1, Q3 i IQR nego što je to slučaj kod Euklidske udaljenosti. Anomaličnima se smatraju matrice koje imaju $d_C > 1{,}1535$, a takvih matrica nema jer vrijednost kosinusne udaljenosti se kreće u intervalu [-1, 1].



Slika 35: Graf distribucije Kosinusnih udaljenosti

Iz grafa distribucije Kosinusnih udaljenosti (slika 35) vidljivo je da su vrijednosti poprilično raspršene po čitavom spektru [0, 1], a to nam govori i mjera interkvartilnog razmaka (IQR) koja ima vrijednost 0,325. Očigledno da ova metoda nije pogodna za detekciju anomalija.

5.4. Jaccard udaljenost

Jaccard mjera sličnosti ili Jaccard indeks podatke promatra kao skupove, a računa se kao omjer presjeka dva skupa podataka i unije ta dva skupa podataka. Prema [51], Jaccard indeks se izračunava po formuli (12):

$$d_J(STM, M) = \frac{|STM \cap M|}{|STM \cup M|} \tag{12}$$

gdje oznake imaju sljedeće značenje:

- d_J – Jaccard indeks
- STM – skup podataka promatrane matrice
- M – skup podataka referentne median matrice

Drugim riječima, ova mjera izračunava sličnost tako što dijeli broj zajedničkih elemenata dvaju skupova s brojem ukupnih elemenata dvaju skupova. Rezultat Jaccard mjere je u intervalu [0, 1] gdje vrijednost 0 znači da skupovi podataka nemaju zajedničkih elemenata, a vrijednost 1 znači da su uspoređena dva identična skupa podataka.

Jaccard udaljenost je mjera različitosti podataka i izračunava se po formuli (13) [51]:

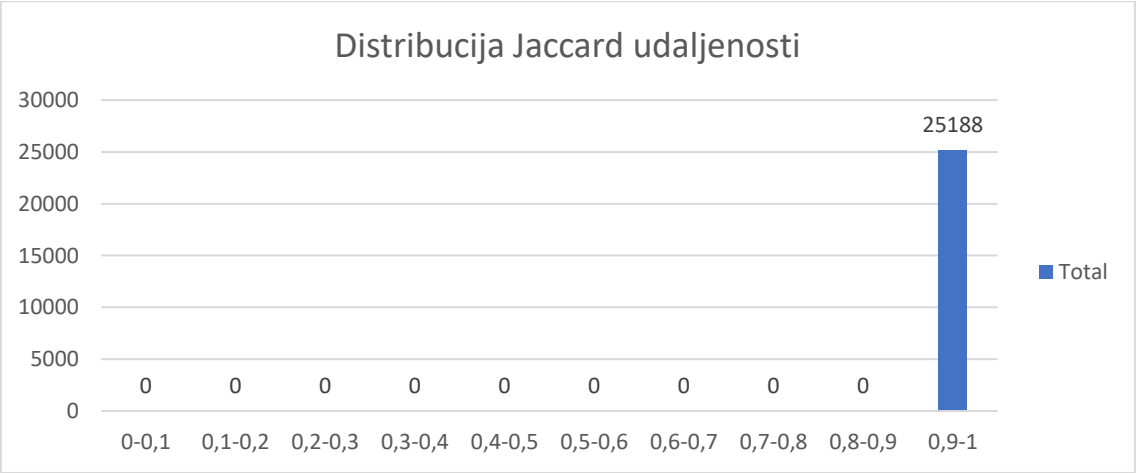
$$Jaccard\ distance = 1 - d_J(STM, M) \tag{13}$$

Vrijednost Jaccard udaljenosti se kreće u intervalu [0, 1], u ovom slučaju vrijednost bliža 1 predstavlja veću različitost podataka. Rezultati obrade podataka za Jaccard udaljenost prikazani su u tablici 7.

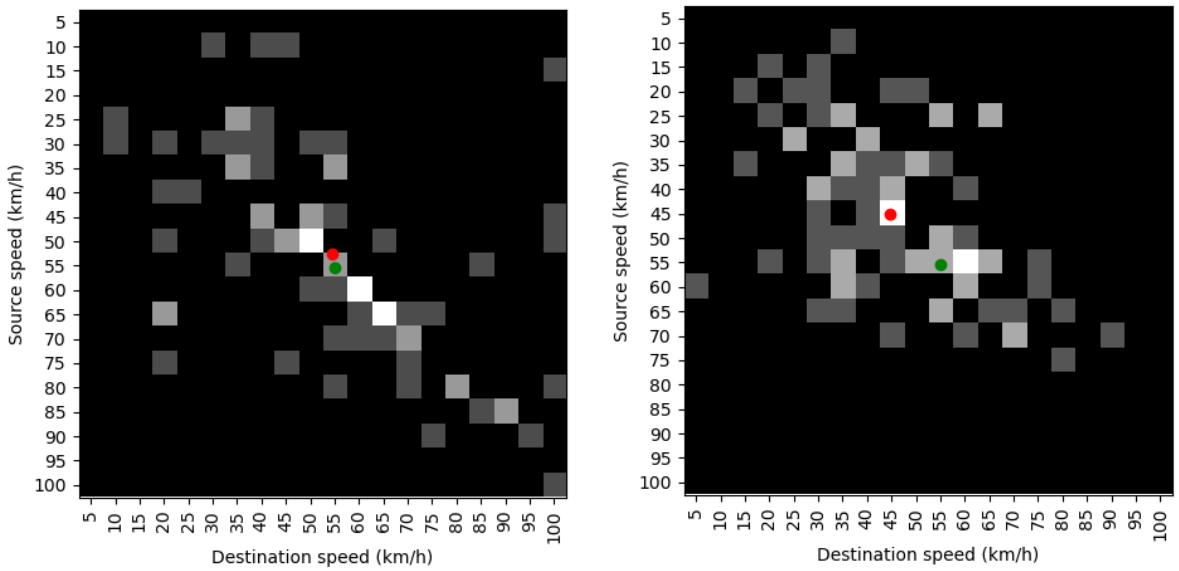
Tablica 7: Rezultati obrade za Jaccard udaljenost

	d_rel	d_manhattan	d_cosine	d_jaccard
Q1	0,08	6363	0,341	0,9822
Q3	0,21	6390	0,666	0,9873
IQR	0,13	27	0,325	0,0051
Q1-1.5*IQR	-0,115	6322,5	-0,1465	0,97455
Q3+1.5*IQR	0,405	6430,5	1,1535	0,99495
Q1-3*IQR	-0,31	6282	-0,634	0,9669
Q3+3*IQR	0,6	6471	1,641	1,0026
Anomaly count	73	1339	0	69
Extreme anomaly count	0	590	0	0

Jaccard udaljenost je dala rezultat najsličniji metodi Euklidskih udaljenosti, anomalijom se smatraju matrice koje imaju udaljenost $d_J > 0{,}99495$ i takvih je detektirano 69 od ukupno 25188 zapisa. Rezultati izmjerenih udaljenosti se kreću u intervalu $[0.9,\ 1]$ i s obzirom na vrijednosti Q1 i Q3 koje su jako blizu 1 i niskom IQR vrijednosti može se zaključiti da su podaci grupirani u određenom dijelu spektra. U ovom slučaju je to u intervalu [0.9, 1] što je vidljivo i na grafu distribucije Jaccard udaljenosti prikazanom na slici 36. Takav rezultat sugerira da ova metoda nije prikladna za detekciju anomalija korištenjem STM, a to dokazuju i primjeri matrica prikazani na slici 37.



Slika 36: Graf distribucije Jaccard udaljenosti



Slika 37: Primjeri matrica koje imaju visoku d_J udaljenost, a nisku d_{rel} udaljenost

Slike prikazuju primjere matrica koje je Jaccard metoda za mjeru udaljenosti detektirala kao anomalične. Karakteristično je da ove matrice imaju jako visoku d_J vrijednost, ali nisku vrijednost relativne Euklidske udaljenosti (d_{rel}). Lijeva slika prikazuje matricu gdje se težišta referentne i prikazane matrice skoro preklapaju, Euklidska udaljenost ima iznos svega 0,021, dok Jaccard udaljenost iznosi 0,995. Slično je i kod matrice na desnoj slici, Euklidska udaljenost je nešto veća (0,102) dok Jaccard udaljenost ima vrijednost 0,995.

Zanimljivo je također da ova metoda detektira matrice koje imaju raspršenije podatke što je i očekivano obzirom da Jaccard udaljenost mjeri koliko su slični odnosno različiti podaci u odnosu na referentnu matricu.

6. Zaključak

U ovom radu opisana je izrada informacijsko – komunikacijskog sustava za detekciju anomalija prometnog toka na urbanim prometnicama. U svrhu detekcije anomalija prometnog toka izrađena je aplikacija za obradu podataka s tri glavne funkcionalnosti: (1) dohvat podataka iz baze podataka, (2) obrada podataka u svrhu detekcije anomalija i (3) vizualizacija podataka na digitalnoj karti.

Za pohranu i dohvat podataka korištena je NoSQL baza podataka u kojoj su pohranjeni povijesni GPS podaci u formi prijelaznih matrica brzina. Nakon dohvata podataka aplikacija obrađuje podatke, generira sliku matrice prijelaznih brzina te pomoću metode za obradu slika odredi težište matrice i iznose udaljenosti za relativnu Euklidsku udaljenost. Rezultati detekcije su uspoređeni s Manhattan, Jaccard i Kosinusnim mjerama za izračunavanje udaljenosti. Izračunate udaljenosti se pohranjuju i obrađuju statističkom metodom za detekciju anomalija. Obrađeni podaci su prikazani putem grafičkog sučelja aplikacije u tabličnom obliku, u obliku grafova i u obliku karte grada Zagreba s anomalijama označenim na prometnicama.

Nakon obrade podataka provedena je evaluacija rezultata, uspoređena je relativna Euklidska udaljenost s ostalim mjerama udaljenosti. Korištenjem relativne Euklidske udaljenosti detektirane su 73 anomalije, dok ostale mjere za udaljenost nisu rezultirale zadovoljavajućim rezultatima. Korištenjem Manhattan mjere udaljenosti detektirano je 1339 anomalija, dok primjenom Kosinusne udaljenosti nije detektirana niti jedna anomalija. Po broju detektiranih anomalija, najsličniji rezultat relativnoj Euklidskoj udaljenosti je imala Jaccard mjera udaljenosti koja je detektirala 79 anomalija. Međutim, detaljnijom analizom rezultata pokazalo se da Jaccard mjera udaljenosti nije prikladna za detekciju anomalija.

Prednost relativne Euklidske udaljenosti u odnosu na druge metode je ta što zbog načina obrade podataka raspolaže koordinatama težišta pojedince matrice. Metoda ne uzima u obzir jako niske vrijednosti i raspolaže sa koordinatama težišta matrice jer su podaci obrađivani kao dvodimenzionalni niz. Manhattan, Jaccard i Kosinusna udaljenost podatke primaju kao jednodimenzionalni niz i iz tog razloga ne mogu odrediti poziciju anomalije u matrici. To ih ne čini lošim metodama za detekciju anomalija, ali nisu pogodne za obradu ovakve vrste podataka.

Poboljšanja su moguća u vidu optimizacije aplikacije kako bi se skratilo vrijeme obrade podataka te korištenje dodatnih izvora podataka što bi pozitivno utjecalo na pouzdanost metode.

Literatura

[1] Chandola V., Banerjee A., Kumar V.: Anomaly detection: A survey, *ACM Computing Surveys*, vol. 41, no. 3. Jul. 01, 2009, doi: 10.1145/1541880.1541882.

[2] Hawkins D.: *Identification of outliers*. Chapman and Hall, 1980.

[3] Chen L., Jakubowicz J., Yang D., Zhang D., Pan G.: Fine-Grained Urban Event Detection and Characterization Based on Tensor Cofactorization, *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, 2017, doi: 10.1109/THMS.2016.2596103.

[4] Zhang H., Zheng Y., Yu Y.: Detecting Urban Anomalies Using Multiple Spatio-Temporal Data Sources, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, 2018, doi: 10.1145/3191786.

[5] Lin C., Zhu Q., Guo S., Jin Z., Lin Y. R., Cao N.: Anomaly detection in spatiotemporal data via regularized non-negative tensor analysis, *Data Mining and Knowledge Discovery*, vol. 32, no. 4, pp. 1056–1073, Jul. 2018, doi: 10.1007/s10618-018-0560-3.

[6] Zhang M., Li T., Shi H., Li Y., Hui P.: A decomposition approach for urban anomaly detection across spatiotemporal data, *IJCAI International Joint Conference on Artificial Intelligence*, 2019, vol. 2019-August.

[7] Fanaee-T H., Fernandes S., Gama J.: Evolving Social Networks Analysisvia Tensor Decompositions: From Global Event Detection Towards Local Pattern, *Discovery Science*, 2019.

[8] Zhang Z., He Q., Tong H., Gou J., Li X.: Spatial-temporal traffic flow pattern identification and anomaly detection with dictionary-based compression theory in a large-scale urban network, *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 284–302, Oct. 2016, doi: 10.1016/j.trc.2016.08.006.

[9] Kuang W., An S., Jiang H.: Detecting Traffic Anomalies in Urban Areas Using Taxi GPS Data, *Mathematical Problems in Engineering*, vol. 2015, 2015, doi: 10.1155/2015/809582.

[10] Huang C., Wu X.: Discovering road segment-based outliers in urban traffic network, *2013 IEEE Globecom Workshops, GC Wkshps 2013*, 2013, doi: 10.1109/GLOCOMW.2013.6825182.

[11] Ge Y., Xiong H., Liu C., Zhou Z. H.: A taxi driving fraud detection system, *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2011, doi: 10.1109/ICDM.2011.18.

[12] Silva N., Shah V., Soares J., Rodrigues H.: Road anomalies detection system evaluation, *Sensors (Switzerland)*, vol. 18, no. 7, Jul. 2018, doi: 10.3390/s18071984.

[13] Wang Z., Lu M., Yuan X., Zhang J., H. van de Wetering,: Visual traffic jam analysis based on trajectory data, *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, 2013, doi: 10.1109/TVCG.2013.228.

[14] Marella S. T., Karthikeya K., Myla S., Sai M., Allam V.: Detecting fraudulent credit card transactions using outlier detection, *International Journal of Scientific and Technology Research*, vol. 8, no. 10, pp. 630–637, Oct. 2019.

[15] Fernandes G., Rodrigues J., Carvalho L., Al-Muhtadi J., Proença M.: A comprehensive survey on network anomaly detection, *Telecommunication Systems*, vol. 70, no. 3. Springer New York LLC, pp. 447–489, Mar. 15, 2019, doi: 10.1007/s11235-018-0475-8.

[16] Kuang W., An S., Jiang H.: Detecting Traffic Anomalies in Urban Areas Using Taxi GPS Data, *Mathematical Problems in Engineering*, vol. 2015, 2015, doi: 10.1155/2015/809582.

[17] Santoyo S:: A Brief Overview of Outlier Detection Techniques, *Towards data sciemce*, 2017, Preuzeto sa: https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561, [Pristupljeno: travanj 2020.]

[18] Gogoi P, Bhattacharyya D. K., Borah B., Kalita J. K.: A survey of outlier detection methods in network anomaly identification, *Computer Journal*, vol. 54, no. 4, 2011, doi: 10.1093/comjnl/bxr026.

[19] Carić T., Erdelić T.: Baze podataka, nastavni materijali, Zavod za Inteligentne transportne sustave, Fakultet prometnih znanosti, Zagreb .

[20] Foote K. D.: A Brief History of Non-Relational Databases, *DATAVERSITY Education*, 2019, Preuzeto sa: https://www.dataversity.net/a-brief-history-of-non-relational-databases/#, [Pristupljeno: svibanj 2020.]

[21] DB-Engines Ranking, 2020., Preuzeto sa: https://db-engines.com/en/ranking, [Pristupljeno: lipanj 2020.]

[22] Merkl Sasaki B: Graph Databases for Beginners: ACID vs. BASE Explained, *neo4j* , 2018, Preuzeto sa: https://neo4j.com/blog/acid-vs-base-consistency-models-explained/, [Pristupljeno: svibanj 2020.]

[23] Meysman A.: NoSQL Database Types, *Database Zone*, 2016, Preuzeto sa: https://dzone.com/articles/nosql-database-types-1, [Pristupljeno: svibanj 2020.]

[24] Drake M: A Comparison of NoSQL Database Management Systems and Models," *DigitalOcean, LLC*, 2019, Preuzeto sa: https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models, [Pristupljeno: svibanj 2020.]

[25] SQL vs NoSQL: What's the difference?, *Guru99,* Preuzeto sa: https://www.guru99.com/sql-vs-nosql.html, [Pristupljeno: svibanj 2020.]

[26] What is a Document Database?, *MongoDB, Inc*., Preuzeto sa: https://www.mongodb.com/document-databases, [Pristupljeno: svibanj 2020.]

[27] MongoDB CRUD Operations, *MongoDB, Inc.*, Preuzeto sa: https://docs.mongodb.com/manual/crud/#create-operations, [Pristupljeno: svibanj 2020.]

[28] Miller B.: The Python Programming Language, *Runestone Interactive*, Preuzeto sa: https://runestone.academy/runestone/books/published/thinkcspy/GeneralIntro/ThePythonProgrammingLanguage.html, [Pristupljeno: svibanj 2020.]

[29] Tišljarić L.: Analiza repova čekanja i razine uslužnosti urbanih prometnica korištenjem algoritama strojnog učenja i NoSQL baza podataka, diplomski rad, Fakultet prometnih znanosti, Zagreb, 2018.

[30] What is NumPy?, *The SciPy community*, Preuzeto sa: https://numpy.org/doc/stable/user/whatisnumpy.html, [Pristupljeno: svibanj 2020.]

[31] The Matplotlib development team: Matplotlib history, *Matplotlib*, Preuzeto sa: https://matplotlib.org/users/history.html, [Pristupljeno: svibanj 2020.]

[32] Machin S. J.: Xlrd documentation, *Lingfo Pty Ltd.*, Preuzeto sa: https://xlrd.readthedocs.io/en/latest/, [Pristupljeno: svibanj 2020.]

[33] McNamara J.: Creating Excel files with Python and XlsxWriter, *XlsxWriter*, Preuzeto sa: https://xlsxwriter.readthedocs.io., [Pristupljeno: svibanj 2020.]

[34] OpenCV Introduction, *Open Source Computer Vision*, Preuzeto sa: https://docs.opencv.org/master/d1/dfb/intro.html, [Pristupljeno: svibanj 2020.]

[35] Clark A: Pillow Overview, *Alex Clark*, Preuzeto sa: https://pillow.readthedocs.io/en/stable/handbook/overview.html, [Pristupljeno: svibanj 2020.]

[36] Erdelić T. Vrbančić S., Rožić L: 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 2015 - Proceedings*, 2015, doi: 10.1109/MIPRO.2015.7160436.

[37] Tišljarić L., Carić T.: Clustering of the Anomalous Spatiotemporal Traffic Patterns Using Tensor Decomposition Method, *Proceedings of the 3rd Symposium on Management of Future Motorway and Urban Traffic Systems (MFTS)*, pp. 1–4, 2020.

[38] Kumar N: Digital Image Processing Basics, *Geeks for Geeks*. Preuzeto sa: https://www.geeksforgeeks.org/digital-image-processing-basics/, [Pristupljeno: svibanj 2020.]

[39] Types of Images, *Tutorials Point*, Preuzeto sa: https://www.tutorialspoint.com/dip/Types_of_Images.htm, [Pristupljeno: svibanj 2020.].

[40] Mordvintsev A., Abid K.: Morphological Transformations, *OpenCV*. Preuzeto sa: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html; [Pristupljeno: svibanj 2020.].

[41] Efford N.: *Digital Image Processing: A Practical Introduction Using Java*. Addison-Wesley Longman Publishing Co., Inc.,75 Arlington Street, Suite 300 Boston, MA, United States, 2000.

[42] Types of Morphological Operations, The Math Works Inc., Preuzeto sa: https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html, [Pristupljeno: svibanj 2020.]

[43] Dwivedi P.: Segmentation using Thresholding, *Geeks for Geeks*, Preuzeto sa: https://www.geeksforgeeks.org/opencv-segmentation-using-thresholding/, [Pristupljeno: svibanj 2020.]

[44] Sinha S.: Grayscaling of Images using OpenCV, *Geeks for Geeks*, Preuzeto sa: https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/, [Pristupljeno: svibanj 2020.]

[45] Jordaan I.:, *Decisions under uncertainty : probabilistic analysis for engineering decisions*. Cambridge University Press, 2005.

[46] Glen S.: Interquartile Range (IQR): What it is and How to Find it, *Statistics How To*, Preuzeto sa: https://www.statisticshowto.com/probability-and-statistics/interquartile-range/, [Pristupljeno: svibanj 2020.]

[47] Craw S.: Manhattan Distance, *Encyclopedia of Machine Learning and Data Mining*, Springer US, 2017, pp. 790–791.

[48] Huang A.: Similarity measures for text document clustering, *New Zealand Computer Science Research Student Conference, NZCSRSC 2008 - Proceedings*, 2008.

[49] Rapp B. E.: Vector Calculus, *Microfluidics: Modelling, Mechanics and Mathematics*, 2017.

[50] What is Cosine Similarity?, *Deep AI, Inc.*, Preuzeto sa: https://deepai.org/machine-learning-glossary-and-terms/cosine-similarity, [Pristupljeno: svibanj 2020.]

[51] Glen S: What is the Jaccard Index?, *Statistics How To*, Preuzeto sa: https://www.statisticshowto.com/jaccard-index, [Pristupljeno: svibanj 2020.]

Popis kratica

GPS	*Global Positioning System* – globalni pozicijski sustav
SQL	*Structured Query Language* – upitni jezik za upravljanje bazama podataka
NoSQL	*Not only SQL* – naziv za baze podataka
LOF	*Local Outlier Factor* – lokalni indeks anomalije
PCA	*Principal Component Analisys* – Analiza glavnih komponenti
DBMS	*Database Management System* – sustav za upravljanje bazama podataka
ANSI	*American National Standards Institute* – Američki institut za standardizaciju
ISO	*International Standardisation Organisation* – Međunarodna organizacija za standardizaciju
JSON	*JavaScript Object Notation* – tip zapisa
BSON	*Binary* JSON – tip zapisa
RGB	*Red-Green-Blue* – Model miješanja boja
RGBA	*Red-Green-Blue-Alpha* – model miješanja boja
CMYK	*Cyan-Magenta-Yellow-Key* – Model miješanja boja
STM	*Speed Transition Matrix* – Matrica prijelaznih stanja
IQR	*InterQuartile Range* – Interkvartilni razmak

Popis slika

Popis tablica

Popis kodova

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 **Izrada informacijsko - komunikacijskog sustava za detekciju anomalija prometnog toka na urbanim prometnicama** na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 3.7.2020.

Student/ica:
Željko Majstorović
(potpis)