

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1375

**PREPOZNAVANJE OPASNIH SITUACIJA U PROMETU  
PRIMJENOM SUSTAVA GEOPANDAS**

Mirna Knez

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1375

**PREPOZNAVANJE OPASNIH SITUACIJA U PROMETU  
PRIMJENOM SUSTAVA GEOPANDAS**

Mirna Knez

Zagreb, lipanj 2024.

## **ZAVRŠNI ZADATAK br. 1375**

Pristupnica: **Mirna Knez (0036539653)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: prof. dr. sc. Dejan Škvorc

Zadatak: **Prepoznavanje opasnih situacija u prometu primjenom sustava GeoPandas**

### Opis zadatka:

GeoPandas je programski sustav za obradu geoprostornih podataka. U ovom radu potrebno je proučiti i opisati primjenu sustava GeoPandas. Posebnu pažnju posvetiti učitavanju i pretraživanju geoprostornih podataka te geoprostornim operatorima za njihovu obradu. Oblikovati postupak za prepoznavanje opasnih situacija u prometu, kao što su vožnja u zabranjenom smjeru ili približavanje vozila iz suprotnog smjera, primjenjujući dostupne geoprostorne operatore u sustavu GeoPandas. Programski ostvariti sustav za prepoznavanje opasnih situacija u prometu i upozoravanje vozača primjenom geoprostornih operatora u sustavu GeoPandas zasnovan na zemljopisnoj karti OpenStreetMap. Ulazni podaci su logički model prometnica preuzet iz zemljopisne karte OpenStreetMap i tragovi kretanja skupine vozila u obliku niza zemljopisnih koordinata prikupljenih iz GPS prijemnika. Izlazni podaci su vrsta prepoznate opasne situacije i popis vozila u dometu kojima je potrebno poslati upozorenje. Pokusnim ispitivanjem odrediti točnost i brzinu rada ostvarenog sustava. Točnost prepoznavanja opasnih situacija u prometu provjeriti za različite konfiguracije prometnica, kao što su gusta gradska mreža, prorijeđena ruralna mreža te raskršća. Opisati arhitekturu i programsko ostvarenje sustava te rezultate ispitivanja.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem svom mentoru, obitelji i prijateljima na podršci,  
savjetima i ohrabrenju tijekom izrade završnog rada.

## Sadržaj

1.	Uvod .....	1
2.	Slični sustavi.....	3
2.1.	Inteligentni Transportni Sustavi (ITS).....	3
2.2.	Aplikacije za detekciju sudara .....	4
2.3.	Tehnologije za detekciju sudara .....	7
3.	Korištene tehnologije.....	9
3.1.	GeoPandas .....	9
3.1.1.	Osnovne karakteristike .....	9
3.1.2.	Struktura sustava GeoPandas.....	10
3.1.3.	Primjena sustava GeoPandas .....	10
3.2.	Python.....	11
3.3.	OpenStreetMap (OSM) .....	12
3.4.	QGIS.....	15
4.	Analiza problema i kreiranje rješenja.....	17
4.1.	Analiza problema.....	17
4.2.	Kreiranje rješenja.....	18
4.2.1.	Prikupljanje i učitavanje podataka.....	19
4.3.	Detekcija opasnih situacija .....	20
4.3.1.	Funkcija check_proximity .....	20
4.3.2.	Funkcija find_street .....	22
4.3.3.	Funkcija check_oneway_and_direction .....	22
4.3.4.	Funkcija is_right_of_lane .....	24
4.3.5.	Funkcija calculate_direction.....	25
4.3.6.	Funkcija calculate_normal.....	26
4.4.	Prikaz rezultata .....	26

5.	Analiza rezultata .....	29
5.1.	Ispitani podaci za gradsku mrežu .....	29
5.2.	Vremenska analiza za gradsku mrežu .....	30
5.2.1.	Vrijeme učitavanja podataka .....	30
5.2.2.	Vrijeme obrade točaka .....	30
5.2.3.	Ukupno vrijeme ispitivanja .....	30
5.3.	Ispitani podaci za ruralnu mrežu .....	31
5.4.	Vremenska analiza za ruralnu mrežu .....	31
5.4.1.	Vrijeme učitavanja podataka .....	31
5.4.2.	Vrijeme obrade točaka .....	32
5.4.3.	Ukupno vrijeme ispitivanja .....	32
5.5.	Komparacija rezultata između gradske i ruralne mreže .....	32
5.5.1.	Vrijeme učitavanja podataka .....	33
5.5.2.	Vrijeme obrade točaka .....	33
5.5.3.	Ukupno vrijeme ispitivanja .....	34
5.6.	GeoPandas i PostGIS usporedba sustava .....	34
6.	Zaključak .....	36
7.	Literatura .....	37
8.	Sažetak .....	39
9.	Summary .....	40

# 1. Uvod

U suvremenom svijetu upravljanje prometom i osiguranje sigurnosti na cestama postaje sve veći izazov zbog stalnog porasta broja vozila i povećane gustoće prometa u urbanim i ruralnim područjima. Sigurnost u prometu ključni je aspekt urbanog planiranja i upravljanja, s obzirom na to da prometne nesreće predstavljaju jedan od glavnih uzroka ozljeda i smrtnosti u mnogim zemljama. Prema podacima Svjetske zdravstvene organizacije (WHO), svake godine u prometnim nesrećama na cestama širom svijeta pogine oko 1,35 milijuna ljudi, odnosno dnevno prosječno oko 3700 ljudi izgubi život u automobilskim nesrećama. Između 20 i 50 milijuna ljudi godišnje zadobije ozljede zbog automobilskih nesreća, koje rezultiraju invaliditetom ili dugotrajnim posljedicama. Automobilske nesreće vodeći su uzrok smrti među mladima u dobi od 5 do 29 godina. Prepoznavanje i prevencija opasnih situacija u prometu, poput vožnje u zabranjenom smjeru, vožnje iznad dozvoljene granice brzine te vožnje pod utjecajem opojnih sredstava, od temeljne su važnosti za smanjenje broja prometnih nesreća i poboljšanje opće sigurnosti na cestama.

Praćenje prometa i prepoznavanje opasnih situacija nužni su za stvaranje sigurnijeg prometnog okruženja. Razvoj naprednih tehnologija omogućio je korištenje sofisticiranih sustava za nadzor i analizu prometa koji mogu bitno pridonijeti smanjenju broja nesreća. Ključni alat korišten u ovom radu je sustav GeoPandas, Python biblioteka za obradu i analizu geoprostornih podataka. Sustav GeoPandas omogućuje manipulaciju geoprostornim podacima na način sličan obradi tabličnih podataka u sustavu Pandas, ali s dodatkom prostornih operacija koje su ključne za ovu vrstu analize. Korištenjem sustava GeoPandas moguće je učitati, pretražiti i modificirati geoprostorne podatke s lakoćom, što programerima omogućuje fokusiranje na implementaciju logike specifične za detekciju opasnih prometnih situacija, a ne na složenost samih podataka.

Osnova sustava temelji se na geoprostornim podacima preuzetim iz zemljopisnih karata OpenStreetMap i tragovima kretanja vozila prikupljenim putem GPS prijemnika, dok u samom radu tragove kretanja vozila simuliramo QGIS alatom. Implementacija sustava uključuje korištenje različitih geoprostornih operatora kako bi se obradili ovi podaci i identificirale potencijalne opasnosti na cestama.

Praktična implementacija sustava uključuje obradu i analizu geoprostornih podataka kako bi se identificirale opasne situacije na cestama. Pritom je poseban naglasak stavljen na točnost i brzinu detekcije kako bi se osiguralo da vozači dobiju pravodobna upozorenja.

Cilj ovog rada nije samo predstaviti tehničko ostvarenje sustava za prepoznavanje opasnih situacija u prometu, koji ulazne podatke koordinata vozila te informacija o geografskim mapama transformira u poruke upozorenja, već i demonstrirati kako moderna tehnologija može pridonijeti sigurnijem prometnom okruženju. Rezultati i uvidi dobiveni kroz ovaj rad pridonijet će boljem razumijevanju potencijalnih primjena tehnoloških rješenja u stvarnim prometnim situacijama, čime se otvaraju nove mogućnosti za unaprjeđenje prometne sigurnosti.



## 2. Slični sustavi

### 2.1. Inteligentni Transportni Sustavi (ITS)

Inteligentni Transportni Sustavi (ITS) predstavljaju napredne tehnološke sustave osmišljene za poboljšanje učinkovitosti, sigurnosti i mobilnosti u prometu. Korištenjem raznih tehnoloških inovacija ITS omogućuje bolju kontrolu i upravljanje prometom, smanjujući zastoje, povećavajući sigurnost na cestama te optimizirajući ukupnu prometnu infrastrukturu.

Komponente inteligentnih transportnih sustava su:

- **kamere i senzori** – ključne komponente sustava za nadzor prometa uključuju kamere postavljene na strateškim mjestima poput raskrižja, autocesta i prometno opterećenih područja, dok senzori ugrađeni na cestama ili uz cestu prikupljaju podatke o protoku vozila, brzini i gustoći prometa
- **podaci u stvarnom vremenu** – kamere i senzori omogućuju prikupljanje podataka u stvarnom vremenu, što pomaže operaterima u kontrolnim centrima da brzo reagiraju na prometne incidente, nesreće ili druge nepredviđene događaje
- **pametni semafori** – ITS koristi prilagodljive semafore koji mogu mijenjati trajanje svjetlosnih signala na temelju trenutnih uvjeta prometa, stoga ovi sustavi koriste algoritme za optimizaciju protoka prometa, smanjujući zastoje i čekanja na raskrižjima
- **koordinacija signala** – integrirani sustavi upravljanja signalima omogućuju sinkronizaciju semafora duž glavnih prometnih pravaca, što poboljšava protok vozila i smanjuje vrijeme putovanja
- **detekcija opasnih situacija** – ITS može identificirati i upozoriti na potencijalno opasne situacije, poput vozila u suprotnom smjeru, vozila koja voze prebrzo ili vozila koja naglo koče
- **komunikacija s vozačima** – kroz dinamičke prometne znakove, mobilne aplikacije ili sustave unutar vozila, ITS može obavijestiti vozače o opasnostima na cesti, savjetovati alternativne rute ili upozoriti na uvjete na cesti, kao što su led, magla ili radovi na cesti

- **GPS i zemljopisne karte** – ITS integrira podatke iz GPS sustava i digitalnih zemljopisnih karata kako bi pružio precizne informacije o lokaciji vozila, navigaciji i vremenu dolaska
- **planiranje ruta** – napredni navigacijski sustavi koriste podatke o trenutačnom prometu kako bi vozačima ponudili najbrže ili najkraće rute do odredišta, uzimajući u obzir zastoje, radove na cestama i druge prepreke.

Prikupljanje podataka u inteligentnim transportnim sustavima odvija se putem raznih izvora:

- **kamera i senzora** – prikupljaju vizualne i senzorske podatke koji se koriste za analizu prometa, prepoznavanje incidenata i optimizaciju protoka prometa
- **GPS podataka** – omogućuju praćenje kretanja vozila u stvarnom vremenu, što je korisno za navigaciju, upravljanje flotama i planiranje ruta
- **zemljopisnih karata** – digitalne karte pružaju kontekstualne informacije koje pomažu u razumijevanju prometne infrastrukture, identifikaciji prepreka i planiranju optimalnih ruta.

## 2.2. Aplikacije za detekciju sudara

### Tesla Autopilot i Tesla Vehicle Safety Features

Tesla Autopilot napredni je sustav za pomoć vozaču koji kombinira sofisticirane senzore, kamere, radare i napredni softver kako bi omogućio poluautomatsku vožnju. Ovaj sustav može preuzeti mnoge zadatke vozača, uključujući držanje vozila u traci, regulaciju brzine i održavanje sigurne udaljenosti od drugih vozila. Glavne komponente Tesla Autopilot sustava uključuju:

- **kamere** – Tesla vozila opremljena su s osam kamera koje pružaju 360-stupanjski pregled oko vozila. Kamere visoke rezolucije omogućuju sustavu da prepozna druge automobile, pješake, bicikliste i različite objekte na cesti
- **radare** – prednji radar ima doomet od oko 160 metara i može detektirati vozila i prepreke čak i u lošim vremenskim uvjetima poput magle, kiše ili snijega, koriste radiovalove za mjerenje udaljenosti i brzine objekata ispred vozila

- **ultrazvučne senzore** – Tesla vozila imaju dvanaest ultrazvučnih senzora smještenih oko vozila koji omogućuju preciznu detekciju objekata u blizini vozila, što je posebno korisno pri parkiranju i vožnji na uskim prostorima
- **softver** – Tesla koristi napredne algoritme strojne inteligencije kako bi obradio podatke prikupljene senzorima i kamerama, ove podatke softver koristi za donošenje odluka u stvarnom vremenu, poput održavanja vozila u traci, prilagođavanja brzine i reakcije na prometne uvjete.

### **Sigurnosne značajke Tesla vozila**

Tesla vozila su dizajnirana s naglaskom na sigurnost, a njihov napredni sustav za detekciju sudara ključna je komponenta u postizanju visokih sigurnosnih standarda. Glavne sigurnosne značajke uključuju:

- **automatsko kočenje u hitnim situacijama (AEB)** – AEB sustav koristi kamere, radare i ultrazvučne senzore za detekciju potencijalnih sudara, što sustavu omogućuje bržu reakciju od vozača
- **upozorenje na napuštanje trake** – ovaj sustav koristi kamere za praćenje oznaka traka na cesti kako bi u slučaju napuštanja trake bez uključivanja pokazivača smjera zvučno ili vizualno upozorio vozača
- **detekciju pješaka i biciklista** – kamere i senzori omogućuju Teslinim vozilima da prepoznaju pješake i bicikliste u blizini vozila te u slučaju opasnosti sustav može automatski kočiti
- **sustav za izbjegavanje sudara** – Tesla vozila opremljena su tehnologijom koja može prepoznati potencijalne sudare s vozilima ispred, iza ili sa strane, koristeći kombinaciju kamera, radara i ultrazvučnih senzora za kontinuirano praćenje okoline vozila
- **automatsku obavijest hitnim službama** – kada sustav detektira sudar, automatski šalje obavijest Teslinom centru za podršku, koji potom kontaktira lokalne hitne službe i pruža im točne podatke o lokaciji nesreće i stanju vozila.

### **General Motors OnStar**

General Motors (GM) poznat je po svojoj inovativnosti i naprednoj tehnologiji, a jedan od njihovih najznačajnijih doprinosa sigurnosti u automobilskom sektoru je OnStar,

telematski sustav koji pruža razne sigurnosne i komunikacijske usluge vozačima. OnStar je predstavljen 1996. godine i od tada se razvio u jedan od najsofisticiranijih sustava te vrste, nudeći širok spektar funkcionalnosti koje znatno poboljšavaju sigurnost i udobnost vožnje.

OnStar telematski je sustav koji kombinira GPS tehnologiju, mobilne komunikacije i razne senzore unutar vozila kako bi pružio usluge koje obuhvaćaju sigurnost, navigaciju, dijagnostiku vozila i pomoć na cesti. Sustav je dizajniran kako bi pružao neposrednu pomoć vozačima u slučaju nesreće, kao i niz drugih korisnih usluga koje poboljšavaju cjelokupno iskustvo vožnje:

- **detekcija sudara i hitna pomoć** – kada senzori u vozilu detektiraju sudar, OnStar automatski aktivira alarm, koji odmah šalje signal središnjem operativnom centru
- **pomoć na cesti** – vozači mogu kontaktirati OnStar operatere za pomoć na cesti u slučaju kvara, prazne gume ili drugih problema
- **ukradena vozila** – ako je vozilo ukradeno, OnStar može surađivati s policijom kako bi ga locirali i zaustavili te daljinski usporili vozilo i onemogućili ga za vožnju.

## **Automatic Pro**

Automatic Pro je napredni OBD-II (On-Board Diagnostics) uređaj koji se povezuje s mobilnom aplikacijom i pruža vozačima detaljne informacije o njihovim vozilima u stvarnom vremenu.

Automatic Pro jednostavno se priključuje u OBD-II port vozila koji se obično nalazi ispod upravljačke ploče. Nakon povezivanja uređaj komunicira s automobilskim računalom i prikuplja podatke o različitim aspektima performansi vozila. Ti se podaci zatim sinkroniziraju s mobilnom aplikacijom putem Bluetootha ili mobilne mreže, omogućujući vozačima da u stvarnom vremenu prate ključne informacije o svom vozilu.

Sustav ima mogućnosti praćenja raznih komponenti za omogućavanje sigurne vožnje, a ovo su samo neke:

- **brzina i ubrzanje**
- **naglo kočenje**
- **potrošnja goriva**
- **dijagnostika vozila**

- **detekcija sudara** – u slučaju sudara uređaj automatski detektira udar i šalje obavijest hitnim kontaktima putem mobilne aplikacije, koja uključuje precizne informacije o lokaciji nesreće
- **SOS upozorenja**
- **Geofencing** – funkcija geofencinga omogućuje vozačima da postave virtualne granice na karti, ako vozilo napusti ili uđe u određeno područje, aplikacija automatski šalje obavijest.

## **Life360**

Life360 sveobuhvatna je mobilna aplikacija dizajnirana za praćenje lokacije članova obitelji u stvarnom vremenu, koristeći GPS podatke. Ova aplikacija, popularna među roditeljima i obiteljima, omogućuje korisnicima da ostanu povezani i sigurni, pružajući brojne funkcionalnosti koje olakšavaju praćenje i koordinaciju unutar obitelji. Uz osnovnu funkcionalnost praćenja lokacije, Life360 uključuje i napredne sigurnosne značajke, kao što je detekcija sudara koja koristi podatke iz senzora u pametnom telefonu kako bi brzo reagirala u hitnim situacijama.

Osnovne karakteristike ove aplikacije su:

- **praćenje lokacije**
- **Geofencing**
- **povijest lokacija**
- **detekcija sudara**
- **SOS upozorenja**
- **sigurnosna izvješća.**

## **2.3. Tehnologije za detekciju sudara**

### **GPS**

GPS uređaji pružaju podatke o lokaciji vozila u stvarnom vremenu, što je korisno za brzo pronalaženje mjesta nesreće.

## **Akcelerometar**

Akcelerometri detektiraju promjene u ubrzanju vozila. Naglo usporavanje ili ubrzavanje može ukazivati na sudar.

## **Žiroskop**

Žiroskopi mjere orijentaciju vozila i mogu otkriti rotacijske pokrete koji su često prisutni tijekom sudara.

## **Kamere i Lidar**

Kamere i lidar senzori koriste se za prepoznavanje prepreka na cesti i mogu pomoći u izbjegavanju sudara ili automatskom kočenju.

## **Telematika**

Telematski sustavi prikupljaju i analiziraju podatke iz različitih senzora u vozilu, pružajući sveobuhvatnu sliku o stanju vozila i okolnostima nesreće.

## 3. Korištene tehnologije

### 3.1. GeoPandas

GeoPandas moćna je biblioteka za programski jezik Python, razvijena s ciljem jednostavne obrade prostornih podataka. Nastala je kao proširenje popularne Pandas biblioteke, omogućavajući integraciju prostornih podataka i geometrijskih operacija u okruženje za znanstvenu obradu podataka. Koristeći sustav GeoPandas korisnici mogu lako manipulirati, analizirati i vizualizirati geografske podatke, što je ključno za mnoge znanstvene i istraživačke projekte.

#### 3.1.1. Osnovne karakteristike

- **Geometrijski objekti** – sustav GeoPandas koristi Shapely biblioteku za rad s geometrijskim objektima, kao što su točke, linije i poligoni. Ovi objekti predstavljaju osnovne građevinske elemente za bilo koju prostornu analizu.
- **Geometrijske operacije** – sustav GeoPandas omogućava izvođenje geometrijskih operacija, kao što su presjek (intersection), unija (union) i razlikovanje (difference). Ove operacije omogućujući identificiranje i kvantificiranje prostornih odnosa između različitih geometrijskih objekata.
- **Prostorne relacije** – biblioteka omogućava jednostavnu provjeru prostornih relacija, kao što su udaljenost, dodirivanje (touches), preklapanje (overlaps) i unutar (within), što olakšava shvaćanje odnosa geometrijskih objekata međusobno u prostornom kontekstu.
- **Transformacija koordinatnih sustava** – GeoPandas podržava transformaciju koordinatnih sustava, što omogućava da se svi geometrijski objekti prikazuju u istom referentnom sustavu.

### 3.1.2. Struktura sustava GeoPandas

Sustav GeoPandas oslanja se na nekoliko ključnih klasa koje omogućavaju rad s prostornim podacima:

- **GeoSeries** – proširenje sustava Pandas Series koje može sadržavati geometrijske objekte. GeoSeries omogućava manipulaciju i analizu pojedinačnih geometrijskih objekata, kao što su točke, linije i poligoni. Svaki element u sustavu GeoSeries može imati povezane atribute koji pružaju dodatne informacije o objektu
- **GeoDataFrame** – proširenje sustava Pandas DataFrame koje može sadržavati GeoSeries kao stupac. GeoDataFrame omogućava rad s tabličnim podacima gdje jedan od stupaca sadrži geometrijske informacije, dok ostali stupci mogu sadržavati atribute vezane uz te geometrije.

	geometry	area	centroid	boundary
0	POLYGON ((-88.06695 34.91611, -88.06695 34.916...	2.340600e-08	POINT (-88.06683 34.91607)	LINESTRING (-88.06695 34.91611, -88.06695 34.9...
1	POLYGON ((-87.92466 34.99480, -87.92468 34.994...	9.460500e-09	POINT (-87.92474 34.99481)	LINESTRING (-87.92466 34.99480, -87.92468 34.9...
2	POLYGON ((-87.93682 34.98534, -87.93673 34.985...	2.457200e-08	POINT (-87.93675 34.98546)	LINESTRING (-87.93682 34.98534, -87.93673 34.9...
3	POLYGON ((-87.94345 34.98608, -87.94344 34.986...	1.247000e-08	POINT (-87.94351 34.98613)	LINESTRING (-87.94345 34.98608, -87.94344 34.9...
4	POLYGON ((-87.92005 34.95808, -87.92005 34.958...	1.181500e-08	POINT (-87.92009 34.95815)	LINESTRING (-87.92005 34.95808, -87.92005 34.9...
...	...	...	...	...
2460399	POLYGON ((-85.55245 34.36618, -85.55247 34.366...	1.357250e-08	POINT (-85.55251 34.36625)	LINESTRING (-85.55245 34.36618, -85.55247 34.3...
2460400	POLYGON ((-85.53906 34.34948, -85.53897 34.349...	3.011100e-08	POINT (-85.53908 34.34960)	LINESTRING (-85.53906 34.34948, -85.53897 34.3...
2460401	POLYGON ((-85.57303 34.33163, -85.57296 34.331...	1.107200e-08	POINT (-85.57300 34.33172)	LINESTRING (-85.57303 34.33163, -85.57296 34.3...
2460402	POLYGON ((-85.55382 34.33916, -85.55388 34.339...	2.137800e-08	POINT (-85.55390 34.33921)	LINESTRING (-85.55382 34.33916, -85.55388 34.3...
2460403	POLYGON ((-85.51582 34.36896, -85.51581 34.368...	5.413500e-09	POINT (-85.51575 34.36894)	LINESTRING (-85.51582 34.36896, -85.51581 34.3...

Slika 1 Prikaz oblika sustava Geopandas GeoDataFrame.

### 3.1.3. Primjena sustava GeoPandas

#### Učitavanje i rad s prostornim podacima

GeoPandas podržava različite formate prostornih podataka, kao što su Shapefile, GeoJSON, KML i drugi. Ova fleksibilnost omogućava korisnicima da lako učitavaju, analiziraju i vizualiziraju prostorne podatke iz različitih izvora.



## **Manipulacija geometrijama**

Prednost sustava GeoPandas je mogućnost jednostavne manipulacije geometrijskim objektima. Korisnici mogu kreirati i modificirati geometrijske objekte te provoditi složene operacije, kao što su presjek, unija i razlikovanje.

## **Geometrijske operacije**

Sustav GeoPandas omogućava izvođenje geometrijskih operacija koje su ključne za mnoge znanstvene i istraživačke projekte. Na primjer, presjek dvaju poligona može otkriti zajedničke prostore, dok unija može spojiti različite geometrijske objekte u jedan. Ove operacije su nužne za analizu prostornog rasporeda i interakcija između različitih geometrijskih objekata.

## **Prostorne relacije i analiza**

Prostorne relacije bitne su za razumijevanje kako se geometrijski objekti odnose jedan prema drugom u prostoru. Sustav GeoPandas omogućava provođenje ovih analiza, pružajući alate za provjeru udaljenosti, dodirivanja, preklapanja i drugih prostornih odnosa.

## **Transformacija koordinatnih sustava**

Rad s podacima iz različitih izvora često zahtijeva transformaciju koordinatnih sustava. Sustav GeoPandas podržava ovu funkcionalnost, omogućavajući korisnicima da konzistentno analiziraju i vizualiziraju podatke. Transformacija koordinatnih sustava osigurava da se svi geometrijski objekti prikazuju u istom referentnom sustavu.

## **3.2. Python**

Python je programski jezik visokog nivoa, opće namjene, koji se ističe svojom čitljivošću, jednostavnošću i svestranošću. Od svog nastanka 1991. godine Python je postao jedan od najpopularnijih jezika za razvoj softvera, znanstveno računarstvo, strojno učenje, web razvoj i mnoge druge primjene. Jedna od glavnih prednosti programskog jezika Python je

njegova jednostavna i intuitivna sintaksa, koja omogućava brz razvoj i lako održavanje koda.

Python podržava različite programske paradigme, uključujući proceduralno, objektno orijentirano i funkcionalno programiranje. Ova fleksibilnost čini ga pogodnim za različite vrste projekata, od malih skripti do velikih aplikacija. Python također nudi bogatu standardnu biblioteku koja pokriva širok spektar funkcionalnosti, od obrade teksta i rada s datotekama do mrežnog programiranja i baze podataka.

Jedno od najvažnijih područja primjene programskog jezika Python je znanstveno računarstvo i analiza podataka. U ovom kontekstu Python nudi širok spektar biblioteka koje olakšavaju rad s podacima. Neke od najvažnijih su NumPy za numeričke izračune, pandas za manipulaciju i analizu tabličnih podataka, matplotlib za vizualizaciju podataka, te SciPy za znanstvene i tehničke izračune. Ove biblioteke omogućavaju učinkovitu manipulaciju, analizu i vizualizaciju podataka, što je od suštinskog značaja za donošenje informiranih odluka u znanosti i industriji.

Jedna od najznačajnijih primjena programskog jezika Python u području je geografskih informacijskih sustava (GIS). GIS tehnologije omogućavaju prikupljanje, analizu i vizualizaciju geografskih podataka. U ovom kontekstu sustav GeoPandas, proširenje popularne biblioteke Pandas, igra važnu ulogu. Sustav GeoPandas omogućava jednostavno rukovanje geografskim podacima, integrirajući prostorne operacije i manipulaciju geometrijskim objektima.

### **3.3. OpenStreetMap (OSM)**

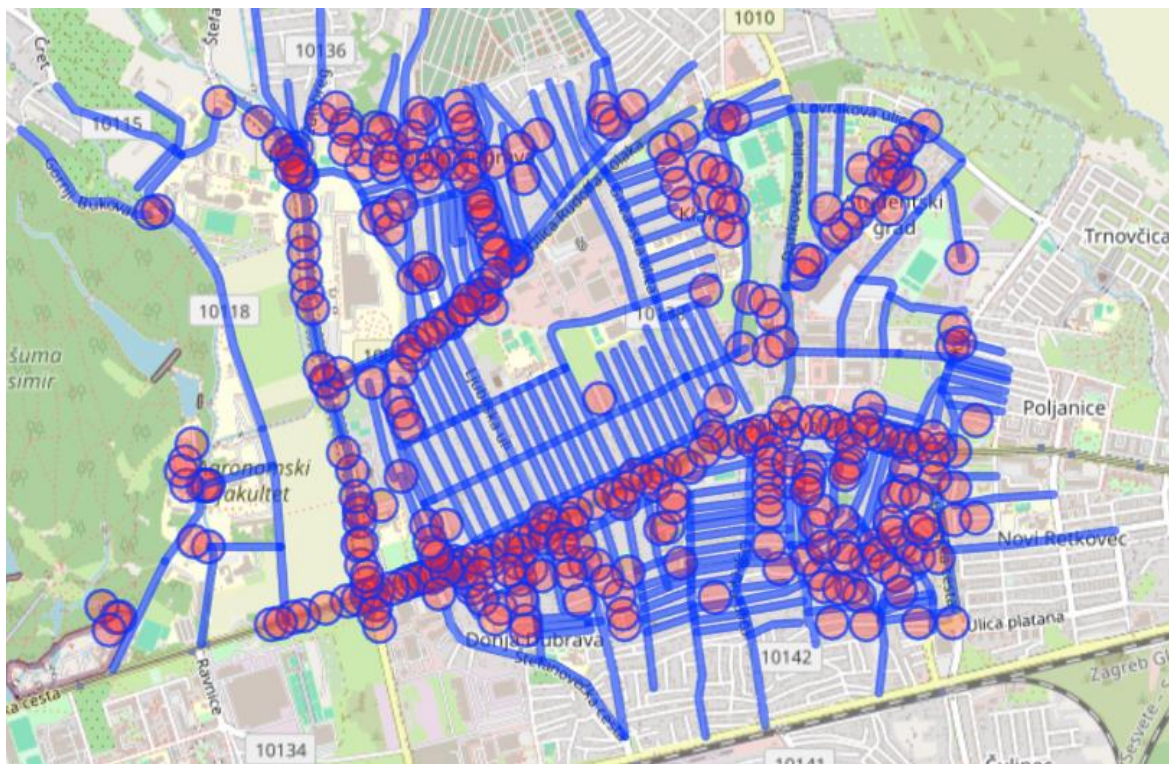
OpenStreetMap (OSM) je otvoreni i slobodni projekt koji ima za cilj stvoriti i pružiti besplatne geografske podatke i karte svijeta.

OSM je zasnovan na modelu crowdsourcinga, gdje volonteri širom svijeta prikupljaju, uređuju i održavaju geografske podatke koristeći različite izvore, kao što su GPS uređaji, zračne i satelitske snimke te lokalno znanje. Ovi podaci uključuju informacije o cestama, zgradama, parkovima, vodnim tijelima, željezničkim linijama i mnogim drugim vrstama geografskih objekata.

**Overpass Turbo** je napredni web alat koji omogućava pristup i pretraživanje OSM podataka putem Overpass API-ja. Kroz jednostavno korisničko sučelje korisnici mogu pisati složene upite za pretraživanje specifičnih geografskih informacija unutar OSM baze podataka i vizualizirati rezultate na interaktivnoj karti. Nakon što se upit izvrši Overpass Turbo omogućava izvoz rezultata u različitim formatima, uključujući GeoJSON, koji je široko prihvaćen standard za razmjenu geografskih podataka.

GeoJSON je format za kodiranje različitih geografskih podataka struktura, uključujući točke, linije i poligone, u obliku JSON (JavaScript Object Notation) objekata. Njegova uporaba olakšava razmjenu geografskih informacija između različitih sustava i aplikacija koje podržavaju rad s prostornih podacima.

Kombinacija sustava OpenStreetMap i Overpass Turbo omogućava preciznu i detaljnu analizu prostornih podataka, što je od velike koristi za urbaniste, istraživače, analitičare i druge profesionalce koji rade s geografskim informacijskim sustavima (GIS). Izvoz podataka u GeoJSON formatu olakšava integraciju i korištenje tih podataka u različitim aplikacijama i analitičkim alatima.



*Slika 2 Prikaz mreže prometnica na mapi.*

```

{
  "type": "Feature",
  "properties": {
    "@id": "way/29689090",
    "highway": "residential",
    "lanes": "1",
    "lit": "yes",
    "name": "Duknovičeva ulica",
    "oneway": "yes",
    "parking:condition:right": "ticket",
    "parking:lane:left": "no_stopping",
    "parking:lane:right": "parallel",
    "parking:lane:right:parallel": "on_street",
    "parking:zone": "2.1",
    "sidewalk": "none",
    "surface": "asphalt"
  },
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [
        16.0116565,
        45.8224491
      ],
      [
        16.0116222,
        45.8226568
      ],
      [
        16.0116083,
        45.8227239
      ]
    ]
  },
  "id": "way/29689090"
},

```

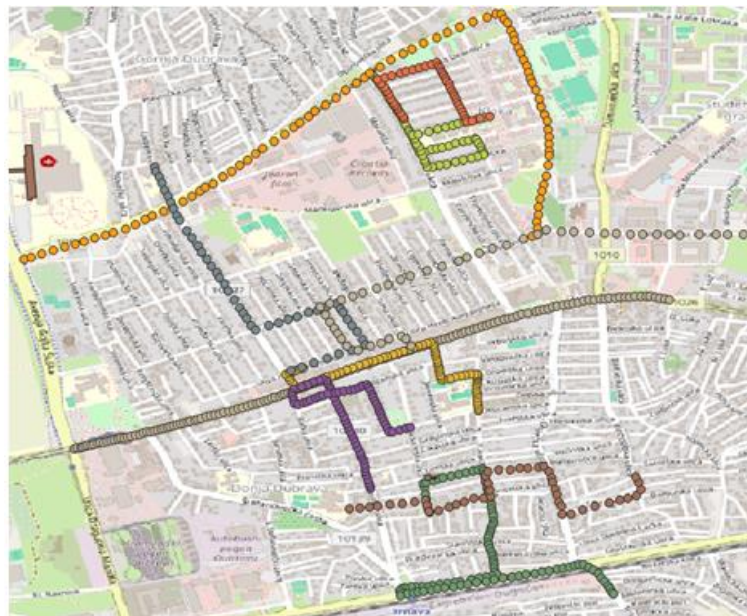
*Slika 3 Primjer GeoJSON zapisa ulice.*

### 3.4. QGIS

QGIS (Quantum GIS) je besplatni i otvoreni sustav za geografske informacijske sustave (GIS). Ovaj alat omogućava korisnicima da prikupljaju, analiziraju i vizualiziraju prostorne podatke. QGIS podržava širok spektar vektorskih i raster formata podataka te baze podataka, omogućujući jednostavnu integraciju s drugim GIS alatima i platformama. QGIS nudi korisnicima mogućnost prilagodbe i proširenja putem dodataka (plugins), što ga čini izuzetno fleksibilnim i prilagodljivim za različite potrebe korisnika.

QGIS se koristi u raznim područjima, uključujući urbanizam, ekologiju, geologiju, arheologiju i mnoge druge discipline koje zahtijevaju rad s prostornim podacima. Korištenjem alata QGIS korisnici mogu obavljati složene analize, modelirati prostorne procese, kreirati karte i razmjenjivati podatke s drugim GIS sustavima.

U kontekstu ovog rada, QGIS je korišten za simuliranje tragova kretanja vozila. Tragovi vozila predstavljaju prostorne zapise koji prate kretanje vozila kroz određeni prostor. Svaka zasebna koordinata u kretanju pojedinog vozila spremljena je u obliku „Multipoint“ tipa geometrije.



*Slika 4 Prikaz simulacije vožnje 10 vozila.*

```
{
  "type": "FeatureCollection",
  "name": "vozilo3",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
  "features": [
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.047408327382289, 45.83796050615134 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.047479841423339, 45.837875095031194 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.047602436922279, 45.837746978105159 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.047719924275437, 45.83759039035052 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.04784762792017, 45.837433802155232 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.047975331564906, 45.837248742810878 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048046845605956, 45.83710638905027 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048143900375955, 45.836974711497469 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048240955145953, 45.836803886099801 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048358442499104, 45.836615265780665 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048455497269103, 45.836465792620871 ] ] } },
    { "type": "Feature", "properties": { "vozilo3": null }, "geometry": { "type": "MultiPoint", "coordinates": [ [ 16.048552552039101, 45.836305642361246 ] ] } },
  ]
}
```

*Slika 5 Prikaz zapisa koordinata pojedinog vozila.*

## **4. Analiza problema i kreiranje rješenja**

### **4.1. Analiza problema**

U ovom radu potrebno je proučiti i opisati primjenu sustava GeoPandas s posebnim naglaskom na učitavanje i pretraživanje geoprostornih podataka te korištenje geoprostornih operatora za njihovu obradu. Zadatak je oblikovanje postupka za prepoznavanje opasnih situacija u prometu, poput vožnje u zabranjenom smjeru, koristeći raspoložive alate u sustavu GeoPandas. Ulazni podaci uključuju logički model prometnica iz sustava OpenStreetMap i tragove kretanja vozila simulirane QGIS alatom. Na temelju tih podataka sustav treba prepoznati opasne situacije i upozoriti vozače u dometu.

Analiza problema uključuje nekoliko ključnih aspekata:

#### **1. proučavanje i opisivanje primjene sustava GeoPandas**

Detaljno proučavanje svih mogućnosti i funkcionalnosti koje sustav GeoPandas nudi za obradu geoprostornih podataka te opis primjene sustava u različitim scenarijima i analizama

#### **2. učitavanje i pretraživanje geoprostornih podataka**

Razvijanje metoda za učinkovito učitavanje geoprostornih podataka iz različitih izvora, u ovom slučaju iz sustava OpenStreetMap te osiguravanje učinkovitog pretraživanja i filtriranja podataka kako bi se identificirali relevantni prostorni objekti

#### **3. korištenje geoprostornih operatora za obradu podataka**

Primjena prostorno-vremenskih operatora za analizu geoprostornih podataka te korištenje operatora za identifikaciju i analizu različitih prostorno-vremenskih relacija između objekata

#### 4. prepoznavanje opasnih situacija u prometu

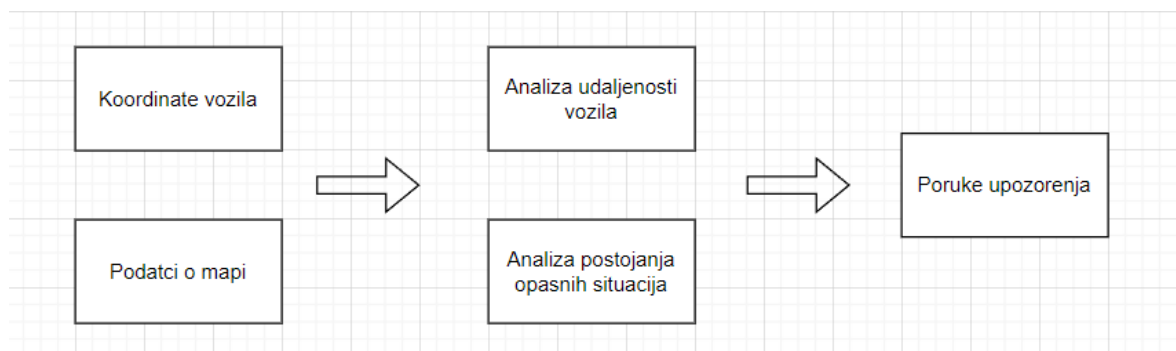
Oblikovanje postupka za prepoznavanje opasnih situacija, kao što su vožnja u zabranjenom smjeru ili približavanje vozila iz suprotnog smjera uz primjenu raspoloživih geoprostornih operatora u sustavu GeoPandas za identifikaciju opasnih situacija

#### 5. upozoravanje vozača

Razvijanje sustava za prepoznavanje opasnih situacija i slanje upozorenja vozačima u realnom vremenu te definiranje izlaznih podataka koji uključuju vrstu prepoznate opasne situacije i popis vozila kojima je potrebno poslati upozorenje

#### 6. ispitivanje sustava

Provođenje pokusnih ispitivanja kako bi se odredila točnost i brzina rada sustava.



*Slika 6 Prikaz ulaza, koraka rješavanja te izlaza problema.*

## 4.2. Kreiranje rješenja

Na temelju analize problema identificirani su ključni izazovi u prepoznavanju opasnih situacija u prometu koristeći sustav GeoPandas. Ovdje će biti detaljno opisan proces kreiranja rješenja, koji obuhvaća prikupljanje i učitavanje podataka, obradu podataka i prostornu analizu, identifikaciju opasnih situacija, razvoj sustava za upozoravanje vozača te ispitivanje i validaciju sustava.



## 4.2.1. Prikupljanje i učitavanje podataka

### 1. Podaci o prometnicama

Podaci su prikupljeni iz sustava OpenStreetMap (OSM), koji pruža detaljne geoprostorne informacije o cestama, raskršćima, prometnim znakovima i drugim relevantnim elementima infrastrukture. Podatci su iz GeoJSON formata učitani koristeći sustav GeoPandas.

```
geojson_file = "city_area.geojson"

edges_gdf = gpd.read_file(giojson_file)
edges_gdf = edges_gdf[edges_gdf['highway'].notna()]

if edges_gdf.crs != "EPSG:3857":
    edges_gdf = edges_gdf.to_crs("EPSG:3857")
```

*Kod 1 Prikaz učitavanja geoprostornih informacija koristeći sustav GeoPandas (gdp) te pretvorba u odgovarajući format.*

### 2. Tragovi vozila

U realnom sustavu tragovi vozila bi bili prikupljeni koristeći GPS, no pošto je ovo simulacija rada, tragovi su generirani koristeći QGIS. Tragovi su u kod učitani koristeći sustav GeoPandas te pretvoreni u odgovarajući referentni sustav radi omogućavanja te ubrzanja izvođenja GeoPandas funkcija.

```
def load_coordinates(files):
    all_coords = []
    total_points = 0
    for file in files:
        gdf = gpd.read_file(file)
        if gdf.crs is None:
            gdf = gdf.set_crs("EPSG:4326")
        if gdf.crs != "EPSG:3857":
            gdf = gdf.to_crs("EPSG:3857")
```

```

coords = []
for geom in gdf.geometry:
    if geom.geom_type == 'MultiPoint':
        coords.extend([point for point in
geom.geoms])
    else:
        coords.append(geom)
all_coords.append(GeoSeries(coords, crs="EPSG:3857"))
total_points += len(coords)
return all_coords, total_points

```

*Kod 2 Prikaz funkcije za učitavanje koordinata tragova vozila.*

## 4.3. Detekcija opasnih situacija

### 4.3.1. Funkcija `check_proximity`

Nakon učitanih podataka analiza sustava obuhvaća analizu međusobnih položaja vozila u svakom trenutku. Iz glavnog koda pokreće se funkcija *check\_proximity* koja se nalazi u modulu. Ta funkcija prvo provjerava jesu li parovi vozila na udaljenosti u radijusu od 100m te ako jesu, pokreće analizu mogućnosti opasne situacije.

```

def check_proximity(point1, point1previous, point2,
point2previous, point_index, vehicle1, vehicle2, edges_gdf,
distance_threshold=10):
    if point1.distance(point2) > 100:
        return None

    street1 = find_street(point1, edges_gdf,
distance_threshold)
    street2 = find_street(point2, edges_gdf,
distance_threshold)
    if street1 and street2 and street1 == street2:
        oneway, direction =
check_oneway_and_direction(street1, edges_gdf)
        if not oneway:
            street_geom = edges_gdf.loc[edges_gdf['name'] ==
street1, 'geometry'].iloc[0]

```

```

        line_start, line_end = street_geom.coords[0],
street_geom.coords[-1]
        line_start = GeoSeries.from_xy([line_start[0]],
[line_start[1]])
        line_end = GeoSeries.from_xy([line_end[0]],
[line_end[1]])
        dir1 = calculate_direction(point1previous,
point1)
        dir2 = calculate_direction(point2previous,
point2)
        if not((is_right_of_line(point1,
line_start.iloc[0], line_end.iloc[0]) and
is_right_of_line(point2, line_end.iloc[0],
line_start.iloc[0])) or
(is_right_of_line(point2, line_start.iloc[0],
line_end.iloc[0]) and is_right_of_line(point1,
line_end.iloc[0], line_start.iloc[0]))):
            if dir1 != dir2:
                return f"Vehicles {vehicle1} and
{vehicle2} at position {point_index + 1} are in danger of
collision on two-way street '{street1}'."
            else:
                dir1 = calculate_direction(point1previous,
point1)
                dir2 = calculate_direction(point2previous,
point2)
                if (dir1 != direction or dir2 != direction) and
dir1 != dir2:
                    return f"Vehicles {vehicle1} and {vehicle2}
at position {point_index + 1} are in danger of collision on
one-way street '{street1}'."
        return None

```

### *Kod 3 Funkcija check\_proximity.*

Početak analize potencijalne opasne situacije, nakon provjere blizine, započinje se provjerom nalaze li se vozila u istoj ulici. Ako se vozila nalaze u istoj ulici, sljedeći korak provjere je razvrstati ulicu na jednosmjernu ili dvosmjernu te odrediti smjer pružanja ulice, što se dobiva funkcijom *find\_street*. Kada se vozila nalaze na dvosmjernoj ulici, daljnja provjera se bazira na tome da se vozila moraju nalaziti s desne strane ceste, što se provjerava funkcijom *is\_right\_of\_lane*, smjer kretanja vozila se provjerava funkcijom

*calculate\_direction*. Ako je pronađeno nedozvoljeno kretanje, funkcija vraća odgovarajuću poruku. U slučaju da se vozila nalaze na jednosmjernoj ulici, provjera se bazira na smjeru pružanja ulice u odnosu na smjer kretanja vozila, ako se vozila kreću u suprotnim smjerovima, odnosno jedno prema drugome, funkcija vraća poruku opasne situacije.

### 4.3.2. Funkcija *find\_street*

Funkcija *find\_street* koristi se za pronalaženje najbliže prometnice za zadanu geografsku točku. Funkcija prima tri parametra: geografsku točku (*point*), GeoDataFrame koji sadrži podatke o prometnicama (*edges\_gdf*) te opcionalni prag udaljenosti (*distance\_threshold*). Ako je najbliža prometnica unutar zadane udaljenosti, funkcija vraća ime te prometnice, u suprotnom, vraća *None*. Prvi je korak analize opasnih situacija, jer nema smisla provoditi analizu ako vozila nisu u istoj ulici te ne idu u susret jedan drugome.

```
def find_street(point, edges_gdf, distance_threshold=100):
    distances = edges_gdf.geometry.apply(lambda geom:
    geom.distance(point))
    nearest_edge_idx = distances.idxmin()
    nearest_edge = edges_gdf.loc[nearest_edge_idx]
    min_distance = distances.min()
    if min_distance <= distance_threshold:
        return nearest_edge['name']
    return None
```

*Kod 4 Funkcija find\_street.*

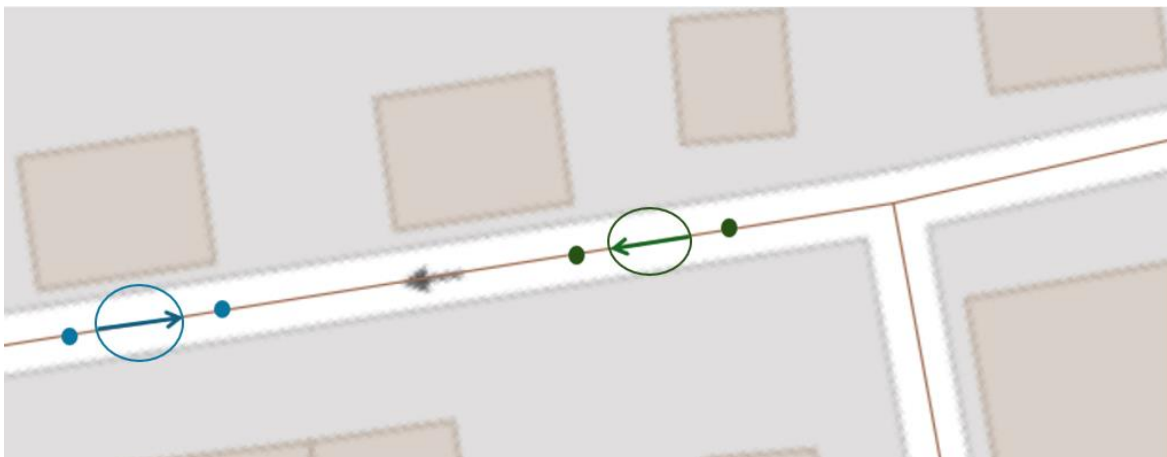
### 4.3.3. Funkcija *check\_oneway\_and\_direction*

Funkcija *check\_oneway\_and\_direction* koristi se za provjeru je li određena prometnica jednosmjerna i ako jest, za određivanje njezina smjera. Funkcija prima dva parametra: naziv prometnice (*street*) i GeoDataFrame koji sadrži podatke o prometnicama (*edges\_gdf*). Vraća bool vrijednost koja označava je li prometnica jednosmjerna i smjer prometnice u stupnjevima. Započinje pronalaženjem retka u GeoDataFrame-u

(edges\_gdf) koji odgovara nazivu zadane prometnice (street). Zatim provjerava sadrži li redak atribut oneway i je li taj atribut postavljen na yes, što označava da je prometnica jednosmjerna. Ako je prometnica jednosmjerna, funkcija dohvaća geometriju prometnice i određuje početnu (line\_start) i krajnju (line\_end) točku linije prometnice. Koristeći funkciju *calculate\_direction* izračunava smjer prometnice od početne do krajnje točke. Funkcija vraća True i smjer prometnice ako je jednosmjerna, inače vraća False i None. Provjerom atributa oneway i izračunom smjera prometnice omogućava sustavu da utvrdi ispravan smjer kretanja vozila u odnosu na prometnicu. Ova informacija je vitalna za prepoznavanje opasnih situacija, kao što je vožnja u suprotnom smjeru.

```
def check_oneway_and_direction(street, edges_gdf):
    row = edges_gdf[edges_gdf['name'] == street].iloc[0]
    if 'oneway' in row and row['oneway'] == 'yes':
        street_geom = row.geometry
        line_start, line_end = street_geom.coords[0],
street_geom.coords[-1]
        line_start = GeoSeries.from_xy([line_start[0]],
[line_start[1]])
        line_end = GeoSeries.from_xy([line_end[0]],
[line_end[1]])
        direction = calculate_direction(line_start.iloc[0],
line_end.iloc[0])
        return True, direction
    return False, None
```

*Kod 5 Funkcija check\_oneway\_and\_direction.*



Slika 7 Prikaz dva vozila na jednosmjernoj cesti koja se kreću ususret jedno drugome s pripadajućim smjerovima azimuta.

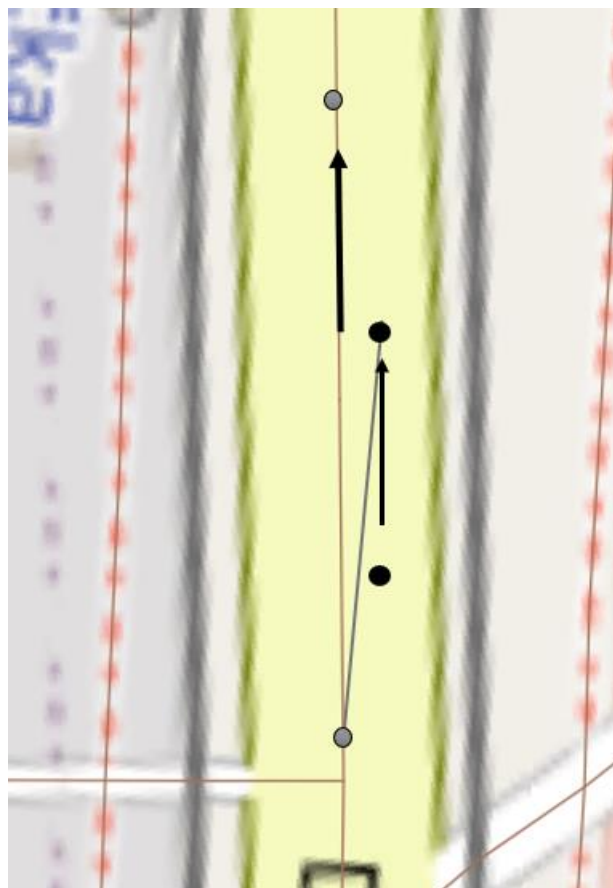
#### 4.3.4. Funkcija `is_right_of_lane`

Funkcija `is_right_of_line` koristi se za provjeru nalazi li se zadana točka (vozilo) desno od linije definirane dvjema točkama (početak i kraj linije). Funkcija prima tri parametra: točku vozila (`vehicle_point`), početnu točku linije (`line_start`) i krajnju točku linije (`line_end`).

```
def is_right_of_line(vehicle_point, line_start, line_end):
    normal_x, normal_y = calculate_normal(line_start,
    line_end)
    vector_x = vehicle_point.x - line_start.x
    vector_y = vehicle_point.y - line_start.y
    return (vector_x * normal_x + vector_y * normal_y) > 0
```

Kod 6 Funkcija `is_right_of_lane`.

Funkcija `is_right_of_line` započinje izračunavanjem normalnog vektora na liniju definiranu početnom i krajnjom točkom koristeći funkciju `calculate_normal`. Zatim izračunava vektor od početne točke linije do točke vozila. Provodi se skalarni produkt između ovog vektora i normalnog vektora. Ako je rezultat pozitivan, to znači da se točka nalazi desno od linije, i funkcija vraća `True`, inače vraća `False`. Provjerom relativne pozicije vozila u odnosu na liniju, funkcija omogućava identifikaciju pravilnog ili nepravilnog kretanja vozila.



*Slika 8 Prikaz dvosmjerne ceste, trenutnog i prethodnog položaja vozila, vektora kretanja vozila, vektora udaljenosti početne točke ulice i trenutnog položaja vozila te vektora normale(točke početka i kraja segmenta ulice).*

#### **4.3.5. Funkcija `calculate_direction`**

Funkcija `calculate_direction` koristi se za izračunavanje smjera kretanja između dvije geografske točke. Funkcija prima dva parametra: početnu točku (`point1`) i krajnju točku (`point2`). Vraća smjer kretanja u stupnjevima, u rasponu od 0 do 360. Započinje izračunavanjem razlike u x koordinatama (`dx`) i y koordinatama (`dy`) između dviju točaka. Zatim koristi arkus-tangens funkciju (`math.atan2`) za izračun kuta između rezultirajućeg vektora i osi x. Ovaj kut pretvara se iz radijana u stupnjeve koristeći funkciju `math.degrees` i normalizira na raspon od 0 do 360 stupnjeva. Izračunom smjera između dvije točke omogućava sustavu da uspoređi smjer kretanja vozila sa smjerom prometnice i tako prepozna nepravilnosti u kretanju.

```
def calculate_direction(point1, point2):
    dx = point2.x - point1.x
    dy = point2.y - point1.y
    return math.degrees(math.atan2(dy, dx)) % 360
```

*Kod 7 Funkcija calculate\_direction.*

#### 4.3.6. Funkcija calculate\_normal

Funkcija `calculate_normal` koristi se za izračunavanje normalnog vektora na liniju definiranu dvjema točkama. Funkcija prima dva parametra: početnu točku (`point1`) i krajnju točku (`point2`). Vraća komponentne normalnog vektora u obliku dvaju brojeva. Započinje izračunavanjem razlike u x koordinatama (`dx`) i y koordinatama (`dy`) između dviju točaka. Zatim izračunava veličinu vektora koristeći Pitagorin teorem. Normalni vektor dobiva se tako da se y komponenta vektora zamijeni s negativnom vrijednošću y komponente, a x komponenta ostavi nepromijenjena, te se oba rezultata podijele s veličinom vektora. Bitna je za određivanje relativnog položaja vozila u odnosu na liniju ceste. Izračunom normalnog vektora omogućava se analiza smjera kretanja vozila i njihova pozicija u odnosu na cestu.

```
def calculate_normal(point1, point2):
    dx = point2.x - point1.x
    dy = point2.y - point1.y
    magnitude = math.sqrt(dx**2 + dy**2)
    return -dy / magnitude, dx / magnitude
```

*Kod 8 Funkcija calculate\_normal.*

#### 4.4. Prikaz rezultata

Prikaz rezultata sustava za prepoznavanje opasnih situacija u prometu omogućuje korisnicima da identificiraju potencijalno opasne situacije na temelju analize podataka prikupljenih iz GPS tragova vozila (u ovom slučaju iz simulacije vožnje, koja je napravljena QGIS alatom) i geoprostornih podataka o prometnicama. Glavni kod sustava organiziran je tako da u svakom vremenskom koraku analizira položaje svih vozila te generira odgovarajuće poruke upozorenja kada se otkrije opasna situacija.



Glavni kod sustava započinje učitavanjem podataka o prometnicama iz GeoJSON datoteke te konfiguriranjem koordinatnog sustava u odgovarajući EPSG:3857 format, kako bi se omogućilo korištenje GeoPandas funkcija te ubrzanje vremena izvođenja koda. Zatim se učitavaju GPS tragovi vozila iz niza datoteka, omogućujući sustavu praćenje kretanja više vozila u realnom vremenu. Nakon što su svi podaci učitani i pripremljeni sustav započinje iteraciju kroz vremenske korake kako bi analizirao položaje vozila u svakom trenutku.

U svakom vremenskom koraku glavni kod uspoređuje položaje svih parova vozila pozivajući funkciju `check_proximity`. Ova funkcija analizira međusobnu udaljenost vozila i određuje nalaze li se vozila u potencijalno opasnoj situaciji. Ako se otkrije opasna situacija, funkcija `check_proximity` vraća poruku koja opisuje prirodu opasnosti, uključujući identifikatore vozila i prometnicu na kojoj se opasnost događa.

Nakon što funkcija `check_proximity` generira poruku upozorenja glavni kod obrađuje ovu poruku i prikazuje rezultate korisnicima. Poruke upozorenja uključuju vrstu prepoznate opasne situacije, identifikatore vozila koja sudjeluju u opasnoj situaciji te prometnicu na kojoj se situacija događa.

Prikaz rezultata osigurava da korisnici sustava mogu brzo i učinkovito reagirati na prepoznate opasne situacije. Sustav također identificira vozila u blizini opasne situacije (u radijusu od 100m od vozila koja se nalaze u potencijalnoj opasnoj situaciji) i generira dodatna upozorenja za ta vozila, čime se povećava sigurnost svih sudionika u prometu.

```
paths, total_points = sd.load_coordinates(files)
end_time = time.time()
max_length = max(len(path) for path in paths)
start_time2 = time.time()
for point_index in range(1, max_length):
    for i in range(len(paths)):
        for j in range(i + 1, len(paths)):
            if point_index < len(paths[i]) and point_index <
len(paths[j]):
                point1 = paths[i][point_index]
                point2 = paths[j][point_index]
                point1previous = paths[i][point_index - 1]
                point2previous = paths[j][point_index - 1]

                message = sd.check_proximity(point1,
point1previous, point2, point2previous, point_index, i + 1, j
+ 1, edges_gdf)
```

```

        if message:
            vehicles_in_radius = set([i + 1, j + 1])
            nearby_vehicles = set()
            for k in range(len(paths)):
                if k != i and k != j and point_index
< len(paths[k]):
                    point = paths[k][point_index]
                    if point1.distance(point) <= 100
or point2.distance(point) <= 100:
                        nearby_vehicles.add(k + 1)

            for vehicle in vehicles_in_radius:
                print(f"Vehicle {vehicle}:
{message}")

            for vehicle in nearby_vehicles:
                print(f"Vehicle {vehicle}:
{message}")

```

*Kod 9 Prikaz dijela glavnog koda zaduženog za pokretanje analize te javljanje poruke odgovarajućim vozilima.*

```

Vehicle 9: Vehicles 9 and 14 at position 33 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 14: Vehicles 9 and 14 at position 33 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 10: Vehicles 9 and 14 at position 33 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 12: Vehicles 12 and 13 at position 33 are in danger of collision on one-way street 'Ulica Rudolfa Kolaka'.
Vehicle 13: Vehicles 12 and 13 at position 33 are in danger of collision on one-way street 'Ulica Rudolfa Kolaka'.
Vehicle 9: Vehicles 9 and 14 at position 34 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 14: Vehicles 9 and 14 at position 34 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 10: Vehicles 9 and 14 at position 34 are in danger of collision on two-way street 'Ulica križnog puta'.
Vehicle 13: Vehicles 5 and 13 at position 40 are in danger of collision on one-way street 'Ulica Rudolfa Kolaka'.
Vehicle 5: Vehicles 5 and 13 at position 40 are in danger of collision on one-way street 'Ulica Rudolfa Kolaka'.

```

*Slika 9 Prikaz obavijesti vozilima koja su u potencijalnoj opasnosti na jednosmjernoj i dvosmjernoj ulici.*

## 5. Analiza rezultata

### 5.1. Ispitani podaci za gradsku mrežu

Analiza rezultata sustava za prepoznavanje opasnih situacija u prometu obuhvaća procjenu performansi sustava na temelju vremena potrebnog za učitavanje podataka i obradu pojedinačnih točaka, kao i ukupnu učinkovitost prepoznavanja opasnih situacija. U ispitivanju su korištena dva skupa podataka različitih veličina, te unutar njih putanje vozila različitih duljina, kako bi se procijenila skalabilnost i pouzdanost sustava. Cilj analize je identificirati ključne faktore koji utječu na performanse sustava.

Podatci za ispitivanje:

	<i>10 vozila</i>	<i>20 vozila</i>
Ukupan broj točaka	728	1349
Ukupno vrijeme potrebno za ispitivanje	15.36s	19.65s
Vrijeme potrebno za učitavanje podataka – mapa	12.88s	12.65s
Vrijeme potrebno za učitavanje podataka – točke	0.23s	0.45s
Vrijeme za učitavanje podataka po točki	0.0003s	0.0003s
Ukupno vrijeme potrebno za obradu točaka	2.25s	6.55s
Vrijeme obrade po točki	0.003s	0.005s

*Tablica 1 Prikaz ispitanih podataka za gradsku mrežu.*

## **5.2. Vremenska analiza za gradsku mrežu**

### **5.2.1. Vrijeme učitavanja podataka**

Vrijeme potrebno za učitavanje podataka predstavlja značajan dio ukupnog vremena obrade. U prvom skupu podataka, za 10 vozila s ukupno 728 točaka, vrijeme učitavanja mape iznosi 12.88 sekundi, a vrijeme učitavanja točaka iznosi 0.23 sekundi, što daje prosječno vrijeme učitavanja po točki od 0.0003 sekundi. U drugom skupu podataka, s 20 vozila i ukupno 1349 točaka, vrijeme učitavanja mape iznosi 12.65 sekundi, a vrijeme učitavanja točaka iznosi 0.45 sekundi, što znači prosječno vrijeme učitavanja po točki od 0.0003 sekundi. Ovaj konzistentni rezultat ukazuje na učinkovitost sustava prilikom učitavanja podataka, bez obzira na veličinu skupa podataka.

### **5.2.2. Vrijeme obrade točaka**

Vrijeme obrade točaka ključan je faktor za procjenu performansi sustava. U prvom skupu podataka ukupno vrijeme potrebno za obradu točaka iznosi 2.25 sekundi, što daje prosječno vrijeme obrade po točki od 0.003 sekundi. U drugom skupu podataka ukupno vrijeme obrade točaka je 6.55 sekundi, što znači prosječno vrijeme obrade po točki od 0.005 sekundi. Povećanje vremena obrade po točki u većem skupu podataka može biti rezultat složenijih odnosa među točkama (veći broj potencijalnih opasnih situacija, koje detekcijom zahtijevaju provođenje većeg broja funkcija) i povećanih zahtjeva za računalne resurse.

### **5.2.3. Ukupno vrijeme ispitivanja**

Ukupno vrijeme potrebno za ispitivanje sustava obuhvaća vrijeme učitavanja i obrade podataka. U prvom skupu podataka ukupno vrijeme ispitivanja iznosi 15.36 sekundi, dok u drugom skupu podataka iznosi 19.65 sekundi. Iako drugi skup podataka ima veći broj točaka, vrijeme ispitivanja proporcionalno raste, što ukazuje na skalabilnost sustava.

### 5.3. Ispitani podaci za ruralnu mrežu

Kao i u prethodnom primjeru mrežu je ispitana na dva skupa podataka.

Podatci za ispitivanje:

	<i>10 vozila</i>	<i>20 vozila</i>
Ukupan broj točaka	1038	2224
Ukupno vrijeme potrebno za ispitivanje	8.79s	13.101s
Vrijeme potrebno za učitavanje podataka – mapa	6.902s	6.97s
Vrijeme potrebno za učitavanje podataka – točke	0.264s	0.74s
Vrijeme za učitavanje podataka po točki	0.00025s	0.00033s
Ukupno vrijeme potrebno za obradu točaka	1.62s	5.38s
Vrijeme obrade po točki	0.0016s	0.0024s

*Tablica 2 Prikaz ispitanih podataka za ruralnu mrežu.*

### 5.4. Vremenska analiza za ruralnu mrežu

#### 5.4.1. Vrijeme učitavanja podataka

Vrijeme potrebno za učitavanje podataka predstavlja značajan dio ukupnog vremena obrade. U prvom skupu podataka, za 10 vozila s ukupno 1038 točaka, vrijeme učitavanja mape iznosi 6.902 sekundi, a vrijeme učitavanja točaka iznosi 0.264 sekundi, što daje prosječno vrijeme učitavanja po točki od 0.00025 sekundi. U drugom skupu podataka, s 20 vozila i ukupno 2224 točaka, vrijeme učitavanja mape iznosi 6.97 sekundi, a vrijeme učitavanja točaka iznosi 0.74 sekundi, što daje prosječno vrijeme učitavanja po točki od

0.00033 sekundi. Vrijeme učitavanja mape pokazuje razinu konzistentnosti, budući da je gotovo identično u oba testna skupa, što je očekivano s obzirom na to da se koristi ista mapa. Vrijeme učitavanja točaka je dulje u skupu s većim brojem točaka, ali prosječno vrijeme učitavanja po točki ostaje u istom rasponu, što dodatno potvrđuje stabilnost i pouzdanost sustava.

#### **5.4.2. Vrijeme obrade točaka**

Vrijeme obrade točaka faktor je za procjenu performansi sustava. U prvom skupu podataka ukupno vrijeme potrebno za obradu točaka iznosi 1.62 sekundi, što daje prosječno vrijeme obrade po točki od 0.0016 sekundi. U drugom skupu podataka ukupno vrijeme obrade točaka je 5.38 sekundi, što znači prosječno vrijeme obrade po točki od 0.0024 sekundi. Ukupno vrijeme obrade je duže u većem skupu podataka, također prosječno vrijeme obrade po točki je povećano u većem skupu podataka u odnosu na manji skup podataka, što može ukazivati na povećanje složenosti obrade s povećanjem broja točaka.

#### **5.4.3. Ukupno vrijeme ispitivanja**

Ukupno vrijeme potrebno za ispitivanje sustava obuhvaća vrijeme učitavanja i obrade podataka. U prvom skupu podataka ukupno vrijeme ispitivanja iznosi 8.79 sekundi, dok u drugom skupu podataka iznosi 13.101 sekundi. Iako drugi skup podataka ima veći broj točaka, vrijeme ispitivanja raste proporcionalno, što ukazuje na skalabilnost sustava u različitim kontekstima, uključujući ruralne mreže.

### **5.5. Komparacija rezultata između gradske i ruralne mreže**

Komparacija rezultata između gradske i ruralne mreže pokazuje nekoliko temeljnih razlika i sličnosti.

### **5.5.1. Vrijeme učitavanja podataka**

Vrijeme učitavanja podataka značajan je faktor u procjeni ukupne učinkovitosti sustava. U urbanim područjima vrijeme učitavanja mape pokazalo se znatno duljim nego u ruralnim područjima. Za prvi skup podataka u gradskoj mreži vrijeme učitavanja mape iznosilo je 12.88 sekundi, dok je za drugi skup iznosilo 12.65 sekundi. Nasuprot tome, vrijeme učitavanja mape u ruralnoj mreži za prvi skup podataka bilo je 6.902 sekundi, a za drugi skup podataka 6.97 sekundi. Ova razlika može se pripisati složenijoj infrastrukturi i većoj količini podataka prisutnih u urbanim područjima.

Vrijeme učitavanja točaka također pokazuje različite trendove između dvije mreže. U gradskoj mreži vrijeme učitavanja točaka za prvi skup podataka iznosilo je 0.23 sekundi, dok je za drugi skup podataka iznosilo 0.45 sekundi. S druge strane, u ruralnoj mreži vrijeme učitavanja točaka bilo je 0.264 sekundi za prvi skup podataka i 0.74 sekundi za drugi skup podataka. Ovi rezultati sugeriraju da učitavanje podataka u ruralnim područjima može biti brže zbog manje složene strukture podataka.

### **5.5.2. Vrijeme obrade točaka**

U gradskoj mreži prosječno vrijeme obrade po točki variralo je od 0.003 sekundi za prvi skup podataka do 0.005 sekundi za drugi skup podataka. U ruralnoj mreži prosječno vrijeme obrade po točki bilo je 0.0016 sekundi za prvi skup podataka, dok je za drugi skup podataka iznosilo 0.0024 sekundi. Ukupno vrijeme obrade točaka u gradskoj mreži bilo je 2.25 sekundi za prvi skup podataka i 6.55 sekundi za drugi skup podataka. U ruralnoj mreži ukupno vrijeme obrade točaka iznosilo je 1.62 sekundi za prvi skup podataka i 5.38 sekundi za drugi skup podataka. Iako je ukupno vrijeme obrade dulje u većem skupu podataka za obje mreže, prosječno vrijeme obrade po točki je kraće u ruralnoj mreži za veći skup podataka. Ovo može ukazivati na manje složene prometne odnose u ruralnim područjima, gdje su vozila manje gusta i opasne situacije rjeđe.

### 5.5.3. Ukupno vrijeme ispitivanja

U gradskoj mreži ukupno vrijeme ispitivanja iznosilo je 15.36 sekundi za prvi skup podataka i 19.65 sekundi za drugi skup podataka. U ruralnoj mreži ukupno vrijeme ispitivanja iznosilo je 8.79 sekundi za prvi skup podataka i 13.101 sekundi za drugi skup podataka. Iako drugi skup podataka ima veći broj točaka, vrijeme ispitivanja proporcionalno raste za obje mreže, što ukazuje na skalabilnost sustava. Ovi rezultati sugeriraju da je sustav učinkovit u obradi podataka u oba konteksta, iako je potrebno više vremena za obradu podataka u urbanim područjima zbog složenijih prometnih uvjeta.

## 5.6. GeoPandas i PostGIS usporedba sustava

Komparacija sustava provedena je u odnosu na rad koji koristi sustav PostGIS za implementaciju sustava za prepoznavanje opasnih situacija u prometu. PostGIS je proširenje za PostgreSQL bazu podataka koje omogućuje pohranu i obradu prostorno-referenciranih podataka.

PostGIS sustav omogućuje obradu geoprostornih podataka unutar relacijske baze podataka. Prednosti sustava PostGIS uključuju visoku skalabilnost i mogućnost rukovanja velikim skupovima podataka. Ispitivanje je provedeno na sličan način kao i u GeoPandas sustavu, uključujući učitavanje podataka, obradu točaka, i identifikaciju opasnih situacija.

Komparacija sustava bazirana je na vremenu obrade točaka kroz niz funkcija.

	Vrijeme obrade točke	Vrijeme učitavanja podataka za jednu točku	Trajanje slanja poruke
PostGIS	0,021s	0,17s	0,0046s
GeoPandas	0.0016s	0.00026s	0.0012s

*Tablica 3 Prikaz tablice usporedbe za ruralna područja.*



	Vrijeme obrade točke	Vrijeme učitavanja podataka za jednu točku	Trajanje slanja poruke
PostGIS	0,0082s	0,17s	0,0032s
GeoPandas	0.003s	0.0003s	0.001s

*Tablica 4 Prikaz tablice usporedbe za gradska područja za 10 vozila.*

	Vrijeme obrade točke	Vrijeme učitavanja podataka za jednu točku	Trajanje slanja poruke
PostGIS	0,0094s	0,19s	0,0053s
GeoPandas	0.005s	0.0003s	0.0013s

*Tablica 5 Prikaz tablice usporedbe za gradska područja za 20 vozila.*

Usporedbom performansi između sustava GeoPandas i PostGIS primijećene su određene razlike. Međutim, važno je napomenuti da su testni skupovi podataka korišteni u ovom istraživanju relativno mali. Iako su razlike u performansama vidljive, zaključci doneseni na temelju ovih testova mogu biti ograničeni zbog veličine skupa podataka. Kako bi se dobila jasnija slika o stvarnim prednostima i nedostacima svakog sustava, potrebno je provesti daljnje ispitivanje na većim skupovima podataka i u realnim uvjetima.

## 6. Zaključak

Razvoj sustava za prepoznavanje opasnih situacija u prometu pomoću sustava GeoPandas pokazuje kako se suvremene tehnologije mogu koristiti za poboljšanje sigurnosti na cestama. Ovaj rad istražio je primjenu GeoPandas sustava za učitavanje, pretraživanje i obradu geoprostornih podataka, simulaciju kretanja vozila te za identifikaciju potencijalno opasnih situacija u prometu. Kroz različite faze razvoja i ispitivanja ocijenjena je učinkovitost sustava u urbanim i ruralnim prometnim mrežama.

Sustav GeoPandas unutar Python programskog jezika pokazao je sposobnost analize geoprostornih podataka te njihov međusobni utjecaj, vrlo je koristan i brz alat za korištenje na malom skupu podataka, velike razmjere podataka tek treba ispitati. Vrijeme potrebno za učitavanje podataka pokazalo je razlike između gradske i ruralne mreže. U urbanim područjima vrijeme učitavanja mape bilo je znatno dulje nego u ruralnim područjima, što se može pripisati složenijoj infrastrukturi i većoj količini podataka. Integracija s Python sustavom omogućuje složenije analize te vizualizacije zbog mogućnosti korištenja drugih biblioteka, no oslanjanje na vanjske datoteke može dovesti do komplikacija prilikom instalacije i kompatibilnosti verzija. Sustav GeoPandas je tehničke zahtjeve ovog rada zadovoljio, no kod naprednijih zadataka mogućnost je dodatne prilagodbe i integracije s drugim alatima i tehnologijama.

Ovaj rad naglašava važnost primjene naprednih alata za geoprostornu analizu u poboljšanju prometne sigurnosti. Razvojem sustava koji mogu brzo i precizno prepoznati opasne situacije moguće je značajno smanjiti broj prometnih nesreća i povećati sigurnost sudionika u prometu. Korištenje geoprostornih podataka za analizu i prepoznavanje obrazaca ponašanja u prometu omogućuje razvoj inteligentnih sustava koji mogu donijeti informirane odluke i pravodobno reagirati na potencijalne opasnosti.

## 7. Literatura

- [1] GeoPandas Development Team, GeoPandas Documentation, GeoPandas Development Team, 2021. Dostupno na: <https://geopandas.org/>
- [2] QGIS Development Team, QGIS User Guide, QGIS Project, 2021. Dostupno na: <https://docs.qgis.org/>
- [3] OpenStreetMap Contributors, OpenStreetMap, OpenStreetMap Foundation, 2021. Dostupno na: <https://www.openstreetmap.org/>
- [4] Overpass API Developers, Overpass Turbo Documentation, 2021. Dostupno na: <https://overpass-turbo.eu/>
- [5] Python Software Foundation, Python Documentation, Python Software Foundation, 2021. Dostupno na: <https://docs.python.org/>
- [6] Wikipedia, "Intelligent Transport Systems (ITS)", 2021. Dostupno na: [https://en.wikipedia.org/wiki/Intelligent\\_transportation\\_system](https://en.wikipedia.org/wiki/Intelligent_transportation_system)
- [7] Tesla, "Autopilot", Tesla, Inc., 2021. Dostupno na: <https://www.tesla.com/autopilot>
- [8] General Motors, "OnStar Services", General Motors, 2021. Dostupno na: <https://www.onstar.com/>
- [9] Automatic Labs, "Automatic Pro", Automatic Labs, 2021. Dostupno na: <https://automatic.com/>
- [10] Life360, "How Life360 Works", Life360, 2021. Dostupno na: <https://www.life360.com/>
- [11] Wikipedia, "GPS", 2021. Dostupno na: [https://en.wikipedia.org/wiki/Global\\_Positioning\\_System](https://en.wikipedia.org/wiki/Global_Positioning_System)
- [12] Wikipedia, "Accelerometer", 2021. Dostupno na: <https://en.wikipedia.org/wiki/Accelerometer>
- [13] Wikipedia, "Gyroscope", 2021. Dostupno na: <https://en.wikipedia.org/wiki/Gyroscope>
- [14] Wikipedia, "Lidar", 2021. Dostupno na: <https://en.wikipedia.org/wiki/Lidar>

- [15] Wikipedia, "Telematics", 2021. Dostupno na: <https://en.wikipedia.org/wiki/Telematics>
- [16] Armis Kantarević, Prepoznavanje opasnih situacija u prometu primjenom sustava PostGIS, 2024.

## 8. Sažetak

Naslov:

Prepoznavanje opasnih situacija u prometu primjenom sustava GeoPandas

Sadržaj:

U današnjem svijetu, gdje su sigurnost i učinkovitost prometa od vitalne važnosti, razvoj sustava za prepoznavanje opasnih situacija predstavlja značajan korak naprijed. Cilj ovog rada implementacija je i ispitivanje sustava za prepoznavanje opasnih situacija u prometu koristeći sustav GeoPandas, alat za geoprostornu analizu unutar Python okruženja. Sustav omogućuje učitavanje, pretraživanje i obradu geoprostornih podataka te prepoznavanje potencijalno opasnih situacija, kao što je vožnja u suprotnom smjeru. Ispitivanje je provedeno na različitim skupovima podataka koji uključuju gradske i ruralne prometne mreže, kako bi se procijenila učinkovitost i skalabilnost sustava.

Ključne riječi:

GeoPandas, Python, geoprostorna analiza, opasna situacija u prometu, gradsko područje, ruralno područje

## 9. Summary

Title:

Traffic incident alerting using GeoPandas Geospatial Framework

Content:

In today's world, where traffic safety and efficiency are of vital importance, the development of systems for recognizing dangerous situations represents a significant step forward. The aim of this paper is to implement and test a system for recognizing dangerous traffic situations using GeoPandas system, a tool for geospatial analysis within the Python environment. The system enables loading, searching, and processing of geospatial data as well as recognizing potentially dangerous situations such as driving in the wrong direction. Testing was conducted on different datasets that include urban and rural traffic networks to evaluate the system's efficiency and scalability.

Keywords:

GeoPandas, Python, geospatial analysis, dangerous traffic situation, urban area, rural area