

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD

**ANALIZA PROMETA JAVNOG GRADSKOG
PRIJEVOZA**

Luka Miličević

Voditelj: izv. prof. dr. sc. Ivana Bosnić

Zagreb, svibanj, 2025.

Analiza prometa javnog gradskog prijevoza

Luka Miličević

Sažetak

Seminar se bavi analizom prometa javnog gradskog prijevoza u stvarnom vremenu, temeljenom na podacima Zagrebačkog električnog tramvaja (ZET). Seminar se usredotočuje na prikupljanje i analizu podataka o vozilima javnog prijevoza, uključujući trenutnu lokaciju vozila, vrijeme dolaska na stanice te uvjete na cesti i vremenske prilike. Podaci se prikupljaju u stvarnom vremenu putem API poziva, a zatim obrađuju kako bi se mogli koristiti za treniranje prediktivnog modela. Na temelju tih podataka, razvija se model za predviđanje vremena dolaska, koji se kontinuirano ažurira tijekom periodične batch obrade podataka. Seminar će istražiti osnovne koncepte i slična rješenja u industriji kroz prototipnu aplikaciju kojoj je cilj povećanje učinkovitosti i točnosti podataka kako bi korisnicima javnog prijevoza omogućio pravovremene i precizne informacije o dolasku vozila.

Ključne riječi: Javni gradski prijevoz; Obrada u stvarnom vremenu; Prediktivni model; GTFS

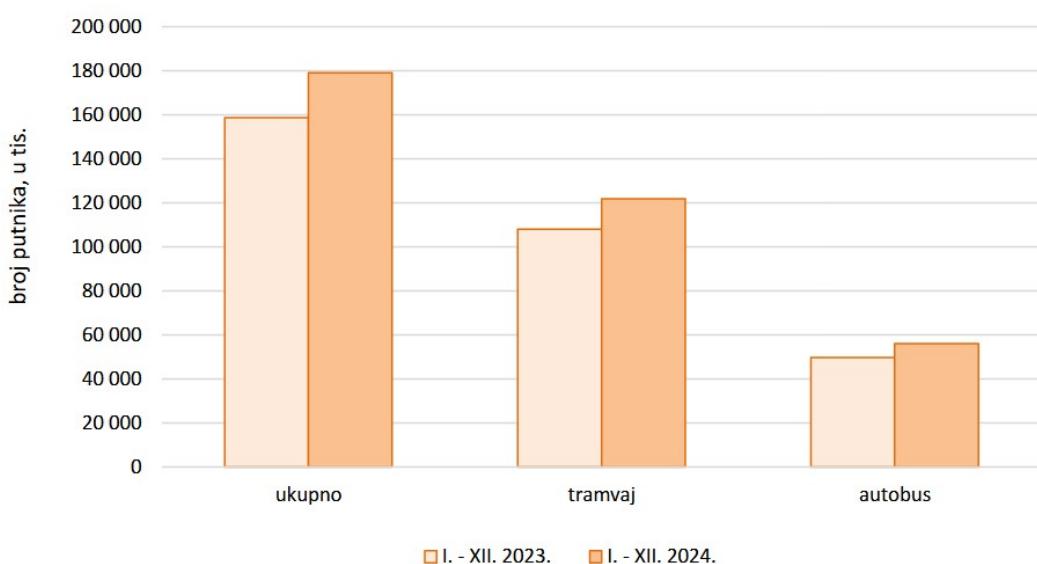
Sadržaj

Sažetak	1
1. Uvod	3
2. Prikupljanje i obrada podataka	5
2.1. Opis podataka	6
2.2. Obrada podataka	8
3. Prediktivni model	12
3.1. Neuronska mreža	13
4. Prototipna aplikacija	17
4.1. Rezultati	18
5. Zaključak	22
Literatura	23

1. Uvod

U urbanim sredinama, javni gradski prijevoz ima ključnu ulogu u svakodnevnom životu građanina. On omogućuje efikasan i ekološki prihvatljiv način transporta velikog broja ljudi, smanjujući potrebu za individualnim vozilima, koje za posljedicu ima smanjenje prometnih gužvi i emisije štetnih plinova. Međutim, izazovi kao što su kašnjenja, nepozdani rasporedi i nedostatak pravovremenih informacija umanjuju njegovu učinkovitost i privlačnost korisnicima.

Grad Zagreb u zadnjem izvješću bilježi značajan porast broja putnika u javnom prijevozu s obzirom na isto razdoblje prethodne godine. Prema podacima za 2024. godinu za razdoblje od srpnja do prosinca, ukupno je prevezeno 97,3 milijuna putnika, što je u usporedbi s istim razdobljem 2023. porast za 26,0%, kada je prevezeno 77,2 milijuna putnika. Promatraljući cijelu godinu javnim gradskim prijevozom prevezen je ukupno 179,1 milijun putnika, što je 12,9% više nego u 2023[1].



Slika 1.1. Broj putnika javnog gradskog prijevoza grada Zagreba, siječanj - prosinac

Porast broja putnika moguće je djelomično pripisati nabavi rabljenih tramvaja te uvođenju novih niskopodnih tramvaja koje proizvodi tvrtka Končar, čime se povećava broj prijevoznih sredstava. Međutim, povećanje broja vozila i putnika ne bi trebao biti jedini strateški cilj ZET-a. Umjesto oslanjanja isključivo na kvantitativne pokazatelje potrebno je ulagati i u poboljšanje kvalitete usluge i iskustvo korisnika. Kako bi to ostvarili, ključno je osigurati pravovremene i točne informacije korisnicima koje će im omogućiti lakše i jednostavnije korištenje javnog prijevoza.

Ovaj seminar istražuje mogućnosti analize prometa u stvarnom vremenu te primjene umjetne inteligencije za razvoj prediktivnog modela kako bi se unaprijedilo pružanje informacija o dolasku vozila, čime bi se omogućilo učinkovitije planiranje i općenito bolja usluga za krajnje korisnike. Poseban naglasak stavljen je na korištenje otvorenih podataka Zagrebačkog električnog tramvaja (ZET) u kombinaciji s metodama strojnog učenja. U radu se također analiziraju relevantni istraživački radovi koji se bave sličnim temama[2][3][4].

2. Prikupljanje i obrada podataka

Danas najčešće korišteni format za razmjenu podataka u industriji javnog prijevoza je GTFS (General Transit Feed Specification). GTFS je otvoreni format razvijen s ciljem omogućavanja standardizirane razmjene informacija o voznim redovima i rutama između pružatelja usluga javnog prijevoza i programera aplikacija. Nastao je kao rezultat suradnje između agencije za javni prijevoz TriMet iz Portlanda (Oregon, SAD) i Googlea. Cilj suradnje bio je razviti standardizirani i lako održiv format za prikaz podataka o javnom prijevozu unutar aplikacije Google Maps. Prvotno nazvan Google Transit Feed Specification, kasnije preimenovan u General Transit Feed Specification, danas predstavlja temelj digitalne infrastrukture mnogih sustava javnog prijevoza te je korišten u preko 10000 agencija i 100 zemalja diljem svijeta[5].

Zagrebački električni tramvaj (ZET) jedna je od tih agencija, no njihova implementacija još se nalazi u ranoj fazi razvoja. ZET trenutačno koristi verziju 1.0 GTFS specifikacije, što znači da sadrži osnovne informacije o voznim redovima i rutama, ali još uvijek nedostaju naprednije funkcionalnosti poput potpune integracije real-time podataka ili potpore za sve dodatne GTFS ekstenzije koje su dostupne u novijim verzijama. Uanatoč toj činjenici i ovako djelomična implementacija GTFS-a omogućuje pristup ključnim informacijama potrebnim za analizu javnog prijevoza, posebno kada se kombiniraju s djelomično dostupnim real-time zapisima.

Pored GTFS podataka koji su temelj ovog seminara, koristit će se i podaci o vremenskim uvjetima, poput oborina i temperature, koji mogu utjecati na promet, kao i podaci o stanju na cestama, uključujući zastoje, nesreće i informaciju o protočnosti prometa. Uključivanjem ovih podataka u prediktivni model omogućuje bolju prilagodbu modela stvarnim uvjetima na terenu. Tradicionalni pristupi koji se oslanjaju isključivo na GTFS podatke pretpostavljaju da se vozila kreću prema voznom redu i u idealnim uvjetima, što

u stvarnosti vrlo često nije slučaj.

2.1. Opis podataka

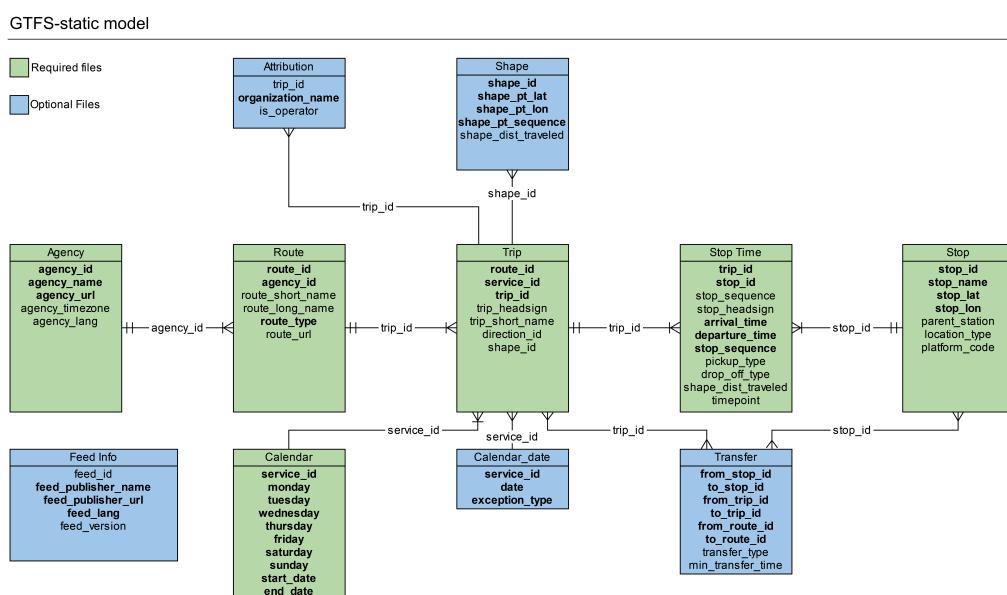
Kao što je već navedeno, koriste se GTFS podaci te podaci o vremenu i prometu. U nastavku su ukratko navedena glavna obilježja tih podataka:

1. GTFS podaci (General Transit Feed Specification)

Sastoje se od dva ključna dijela: **static** i **realtime**.

GTFS static podaci obuhvaćaju minimalno šest CSV datoteka koje sadrže informacije o linijama, voznim redovima, stajalištima, putovanjima i drugim karakteristikama javnog prijevoza. Ove podatke održava i ažurira nadležna agencija u skladu s promjenama u voznim redovima ili zbog drugih operativnih potreba. U praksi se takva ažuriranja događaju tjedno ili mjesечно. Svaka GTFS static datoteka sadrži barem jedan ključni atribut (primarni ili strani ključ), čime se omogućuje međusobno povezivanje datoteka te rekonstrukcija logičkih odnosa između entiteta. Dijagram na slici 2.1. prikazuje jedan mogući raspored i spajanje GTFS datoteka u relacijski model.

Neki od ključnih podataka sadržanih u GTFS static podacima su identifikatori linija, vozila i pojedinih putovanja; nazivi linija i stanica; raspored očekivanih odlažaka i dolazaka; redoslijed postaja na liniji; geometrija puta i tako dalje.



Slika 2.1. Primjer GTFS static relacijskog modela

GTFS realtime podaci omogućuju praćenje javnog prijevoza u stvarnom vremenu. Sadrži informacije o geografskoj lokaciji vozila, statusu pojedinog putovanja poput vremena dolaska na pojedinu stanicu, kašnjenje u odnosu na raspored te obavijesti o mogućim poremećajima u prometu ili izmjenama u voznom redu. Oni također sadrže identifikatore koji omogućuju njihovo povezivanje s GTFS static podacima. Glavna prednost GTFS realtime podataka je korištenje formata Protocol Buffers (Protobuf). Protocol Buffers su jezično i platformski neutralni binarni format za serijalizaciju podataka. Dizajnirani su s ciljem brze obrade i minimalne veličine podataka (minimalan overhead) koji omogućuju učinkovito kodiranje, dekodiranje i prijenos podataka.

2. Vremenski podaci

Seminar će koristiti vremenske podatke poput temperature, tlaka, brzine vjetra te, najvažnije, podatke o vrsti i količini oborina, kako bi se ispitalo imaju li ti čimbenici utjecaj na odstupanja u voznom redu, iako je prethodno istraživanje Rashvand i sur. ([2]) pokazalo da vremenski uvjeti nemaju značajan utjecaj na pogrešku predviđanja dolazaka. Kao izvor podataka koristit će se podaci OpenWeatherMap API-ja, budući da nudi veću pokrivenost i detaljnije vremenske informacije u odnosu na Državni hidrometeorološki zavod (DHMZ).

3. Podaci o stanju u prometu

Podaci o stanju u prometu mogu imati značajan utjecaj na odstupanja u voznom redu, stoga će se razmatrati informacije o trenutačnoj brzini prometa, brzini prometa kada nema zagušenja te podaci o zastoju. Seminar će koristiti prometne podatke dobivene od TomTom-a, koji je odabran zbog svoje jednostavnosti korištenja i količine besplatnih API poziva. Alternativni izvori podataka mogu biti Google Maps Traffic API, koji je jedan od najpreciznijih i najopsežnijih izvora prometnih informacija, no njegova je uporaba ograničena zbog relativno visokih troškova. Ostale opcije uključuju HERE Traffic API, Bing Traffic API te druge slične usluge koje nude detaljne podatke o stanju u prometu.

2.2. Obrada podataka

Kao što je već spomenuto, GTFS static i realtime podaci sadrže brojne ključne atributе (primarne i strane ključeve) koji omogućuju povezivanje podataka i predstavljaju temelj aplikacije. Prije nego što ih je moguće povezati, potrebno je napraviti nekoliko koraka pripreme podataka. U slučaju GTFS static podataka koji su odmah čitljivi bilo kojim uređivačem teksta, nije potrebna posebna pretvorba podataka. Ipak, radi jednostavnijeg korištenja i povezivanja podataka preporučuje se njihovo učitavanje u bazu podataka. U okviru ovog seminara korištena je PostgreSQL baza podataka, koja omogućuje jednostavan uvoz CSV datoteka kao tablica te pomoću SQL upita lako povezivanje i izdvajanje željenih informacija. Da bi se olakšao taj postupak ostvarena je skripta koja preuzima najsvježije podatke od ZET-a, raspakira ih te ih učitava u bazu podataka.

GTFS realtime podaci zahtijevaju dodatni korak pretvorbe prije nego što ih je moguće učinkovito koristiti. Razlog tome, kako je ranije spomenuto, jest format Protocol Buffers koji je binarni format nečitljiv bez prethodnog dekodiranja. Korištenje binarnog formata u odnosu na tekstualne formate poput JSON-a ili XML-a donosi svoje prednosti poput uzimanja znatno manje memorije za prijenos istih podataka (ostvaruje manji overhead) što rezultira bržim prijenosom podataka preko mreže, smanjuje kašnjenje podatka i opterećenje mreže. Jedini značajni nedostatak je potreba za kodiranjem i dekodiranjem podataka za ostvarenje prijenosa, što zahtijeva posebni alat ili biblioteku te odgovarajući opis podataka kako bi taj postupak bio uspješan. U praksi to ne predstavlja veliki problem, budući da je jedan od osnovnih principa pri dizajnu Protocol Buffersa bila neovisnost o korištenoj platformi i programskom jeziku tako da danas postoji široka podrška za najpopularnije programske jezike. Seminar pomoću odgovarajuće biblioteke lako obavlja kodiranje/dekodiranje podataka te ih je tada moguće učinkovito koristiti. Na slikama 2.2. i 2.3. prikazan je rezultat dekodiranja ZET-ovih GTFS realtime podataka.

Nakon što su ti koraci obavljeni, aplikacija će izvući podatke od značaja, poput identifikatora putovanja, broja rute, potvrđenog dolaska na posljednju stanicu (uključujući vrijeme i redni broj stanice) te identifikatora poruke. Potreba za identifikatorom poruke proizlazi iz činjenice da GTFS realtime podaci sadrže dva tipa poruka: poruku o statusu putovanja i poruku o podacima o vozilu. Tako se pomoću identifikatora poruke dohvaćaju podaci o trenutnoj lokaciji vozila za pojedino putovanje.

```

1  {
2    "id": "X20EUPS0EY",
3    "tripUpdate": {
4      "trip": {
5        "tripId": "0_30_406_4_11309",
6        "startDate": "20250110",
7        "scheduleRelationship": "SCHEDULED",
8        "routeId": "4"
9      },
10     "stopTimeUpdate": [
11       {
12         "stopSequence": 28,
13         "arrival": {
14           "delay": -433,
15           "time": "1736523175"
16         },
17         "departure": {
18           "delay": -433,
19           "time": "1736523175"
20         },
21         "stopId": "201_1",
22         "scheduleRelationship": "SCHEDULED"
23       }
24     ],
25     "timestamp": "1736523134"
26   }
27 }

```

```

1  {
2    "id": "X20EUPS0EY_22135",
3    "vehicle": {
4      "trip": {
5        "tripId": "0_30_406_4_11309",
6        "startDate": "20250110",
7        "scheduleRelationship": "SCHEDULED",
8        "routeId": "4"
9      },
10     "position": {
11       "latitude": 45.82920455932617,
12       "longitude": 16.060285568237306
13     },
14     "timestamp": "1736523134",
15     "vehicle": {
16       "id": "22135"
17     }
18   }
20 }

```

Slika 2.3. GTFS realtime podaci o vozilu

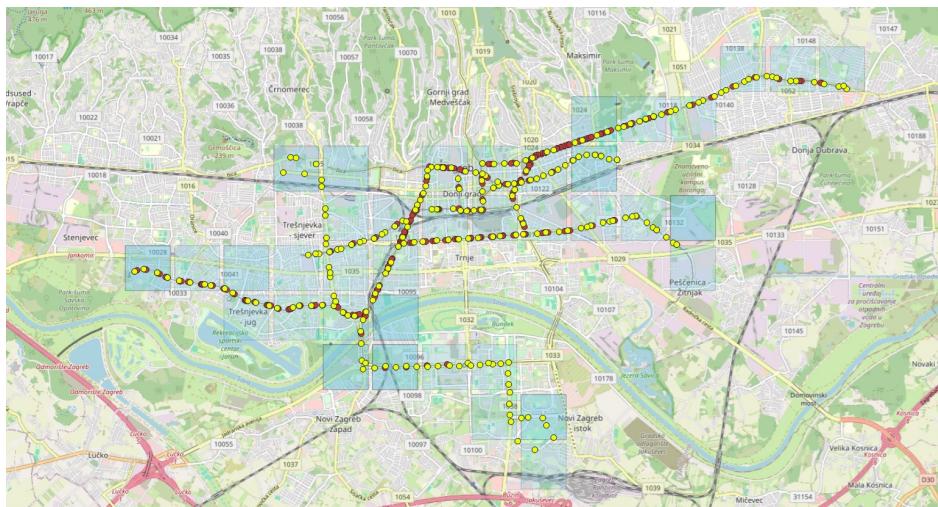
Slika 2.2. GTFS realtime podaci o statusu putovanja

Na prikupljene podatke o putovanju i vozilu nadodaju se podaci o stanju u prometu i vremenski podaci na način da se preuzmu podaci od najbliže točke koja se koristila za mjerjenje. Način odabira te točke bit će ukratko objašnjen u nastavku. Ti privremeni podaci predstavljaju jedno nezavršeno putovanje koje aplikacija sprema u radnu memoriju koristeći map strukturu podataka, gdje je ključ identifikator putovanja, što omogućuje brzo pretraživanje i pristup podacima. Oni se zadržavaju u memoriji onoliko dugo koliko je potrebno da bi pristigla poruka o dolasku vozila na sljedeću stanicu, odnosno kraj promatranog segmenta putovanja. Drugim riječima, najprije se bilježe podaci o dolasku tramvaja na početnu stanicu (vrijeme dolaska, broj stanice) i njegova trenutna lokacija. Na te se podatke nadodaju informacije o prometu i vremenskim uvjetima, kao i izračunata udaljenost do ciljne (sljedeće) stanice. Svi ti podaci se privremeno čuvaju u memoriji dok ne pristigne poruka o dolasku tramvaja na sljedeću stanicu. Taj proces se izvodi u beskonačnoj petlji.

Kada je prikupljen određen broj podataka o završenim putovanjima, aplikacija prebacuje podatke iz radne memoriju u perzistentnu memoriju, odnosno bazu podataka gdje se čuvaju i čekaju obrađivanje unutar neuronske mreže.

Odabir točaka za mjerjenje, odnosno lokacija nad kojima se izvršavaju upiti o vremenskim uvjetima i prometnoj situaciji, temelji se na GTFS static podacima i bit će objašnjen na primjeru sa slika 2.4. i 2.5. Na početku je potrebno definirati promatrane rute (npr. linije 5, 17, 109, 13, 3) te se za njih uz pomoć *trips.txt* i *shapes.txt* pronađu sve moguće geometrije puta (putanje). Putanje se potom uzorkuju svakih 500 metara te nastaju kandidatne točke (crvene točke), koje se zatim zbog preklapanja i blizine s drugim kandidatima grupiraju pomoću algoritma za grupiranje točaka s obzirom na udaljenost. Izlaz algoritma su reprezentativne točke (žute točke) koje će biti korištene za upite o stanju prometa. Razlog grupiranja je vrlo jednostavan, a to je smanjenje potrebnih API poziva, jer je moguće da su neke točke međusobno jako blizu te će stanje u prometu za njih biti gotovo jednako (Slika 2.5.). Tako za primjer u kojem su promatrane linije 5, 17, 109, 13, 3 na početku postoji 640 kandidatnih točaka, provedbom algoritma grupiranja taj broj se smanjuje na 222 točke.

Za točke koje će se koristiti u upitima o vremenskim uvjetima potrebno je napraviti još jedan korak – optimizaciju kojom se dodatno smanjuje broj potrebnih API poziva. S obzirom na to da područje grada Zagreba nije značajno podložno mikroklimatskim razlikama, moguće je grupirati prostorno bliske točke i koristiti jednu zajedničku vremensku prognozu za sve njih. U tu svrhu nad kartom grada Zagreba konstruira se mreža sastavljena od kvadrata dimenzija 1×1 kilometar. Ako se unutar pojedinog kvadrata nalazi barem jedna odabrana točka (žuta točka), središte tog kvadrata odabire se kao reprezentativna točka za vremenski upit. Tako se za isti primjer kao prije, od 222 odabrane točke, broj smanjuje na samo 38 točaka.



Slika 2.4. Odabране тоčke za mjerjenja



Slika 2.5. Uvećana slika kojom je uočljivo grupiranje

3. Prediktivni model

Istraživački radovi koji su poslužili kao nadahnuće za ovaj rad pokazali su da metode strojnog učenja, a posebno neuronske mreže, predstavljaju trenutno najučinkovitiji pristup za predviđanje vremena dolaska vozila javnog prijevoza u stvarnim uvjetima. U usporedbi s tradicionalnim metodama poput regresijskih modela ili modela podrške vektorima (SVM), neuronske mreže bolje se nose s nelinearnostima, vremenskom ovisnošću i višedimenzionalnim značajkama koje su prisutne u ulaznim podacima[2][3].

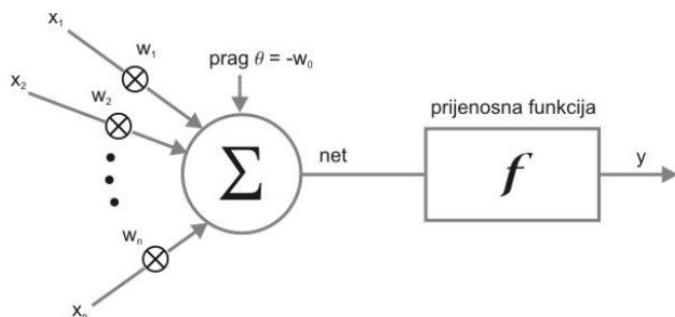
U istraživanju kojeg su proveli Rashvand i sur. ([2]), korištenjem potpuno povezane neuronske mreže (FCNN), postignuta je prosječna pogreška predikcije (RMSE) od 35,74 sekunde, što znači da je predviđeno vrijeme dolaska autobusa u prosjeku imalo odstupanje manje od 36 sekundi na skupu podataka iz New Yorka koji uključuje 200 linija autobusa i 2 milijuna ulaznih točaka. S druge strane, C. Fors Johansson koristio je naprednu metodu neuronske mreže, Long Short-Term Memory (LSTM), koja omogućava dugoročno pamćenje informacija i učenje zavisnosti između podataka tijekom vremena, čime postiže preciznija predviđanja. Primjenom LSTM-a za prepoznavanje zavisnosti između pojedinih stanica na liniji prijevoza, uspio je ostvariti čak 63,7% predviđanja unutar okvira od ± 1 minute[4].

S obzirom na sličnost problema s prethodno navedenim radovima, u ovom seminaru primijenit će se pristup temeljen na neuronskim mrežama s ciljem izgradnje modela koji može uzimati u obzir podatke o prometu i ostale varijabilne značajke te isporučivati točne informacije o očekivanom dolasku vozila. U nastavku će biti detaljno opisane korištene metode, arhitektura neuronske mreže, način treniranja modela te evaluacija njegove točnosti.

3.1. Neuronska mreža

"Neuronska mreža jest skup međusobno povezanih jednostavnih procesnih elemenata, jedinica ili čvorova, čija se funkcionalnost temelji na biološkom neuronu. Pri tome je obradbena moć mreže pohranjena u snazi veza između pojedinih neurona tj. težinama do kojih se dolazi postupkom prilagodbe odnosno učenjem iz skupa podataka za učenje." (Dalbelo Bašić, Čupić i Šnajder [6]).

Drugim riječima neuronske mreže su pojednostavljeni matematički model ljudskog mozga u kojem se podaci obrađuju s pomoću neurona (čvorova) povezanih vezama različite težine (jakost). Svaki neuron prima ulazne podatke, obrađuje ih i primjenjuje aktivacijsku funkciju (prijenosnu funkciju) te izlaz prosljeđuje dalje.

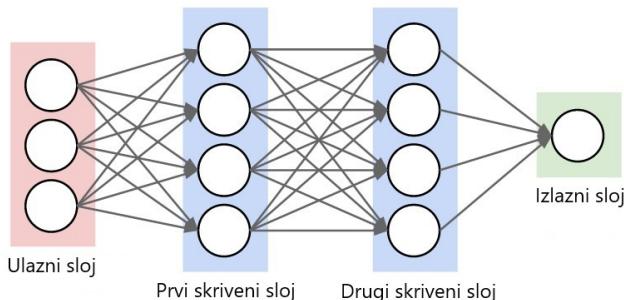


Slika 3.1. Umjetni neuron [6]

Neuronska mreža ima sposobnost učenja, to jest ima sposobnost prilagođavanja svojih težina kako bi smanjila pogrešku između stvarnog i očekivanog izlaza. Zbog te sposobnosti, neuronske mreže odlično rješavaju probleme klasifikacije i predviđanja te imaju jako široku primjenu u rješavanju zadataka poput raspoznavanja uzoraka, aproksimacije funkcija, obrade slika, videozapisa i govora i raznih drugih problema u području umjetne inteligencije.

Neuronska mreža je sastavljena od najčešće velikog broja međusobno povezanih neurona organiziranih u slojeve. Ti slojevi su:

- **Ulazni sloj** - prima podatke iz vanjskog svijeta (nemaju funkcionalnost neurona, samo prosljeđuju podatke)
- **Skriveni sloj** - najčešće postoji više skrivenih slojeva koji zajedno vrše obradu podataka
- **Izlazni sloj** - daje krajnji rezultat neuronske mreže



Slika 3.2. Slojevi neuronske mreže

Broj slojeva i neurona mreže, odnosno njena arhitektura utječu na performanse modela. Mreže s većom arhitekturom mogu se primjeniti na složenije zadatke, bolje prepoznati nelinearne veze te proučiti više značajki (ulazne informacije), ali su sklonije problemima poput prenaučenosti i zahtijevaju više računalnih resursa. Prenaučenost je pojam koji opisuje situaciju kada model tijekom treniranja previše prilagodi svoje parametre na podatke iz skupa za učenje. Posljedica toga, kada se model počne ispitivati na skupu podataka za testiranje koji predstavlja dosad neviđene podatke, točnost modela značajno opada. Manje neuronske mreže primjenjuju se na jednostavnije zadatke te su manje računalno zahtjevne i manje sklone prenaučenosti.

Prije samog učenja mreže, ulazni podaci se moraju normalizirati. Normalizacija sprječava velike razlike u rasponima vrijednosti pojedinih značajki, što može uzrokovati da velike vrijednosti dominiraju nad manjima. Primjenom normalizacije se uravnotežuje utjecaj značajki i smanjuje negativan utjecaj ekstremnih vrijednosti. Osim toga, normalizacija ubrzava proces učenja i doprinosi boljoj generalizaciji modela, smanjujući rizik od prenaučenosti. Dvije od najpopularnijih tehniki normalizacije su standardizacija numeričkih (brojčanih) značajki i "One-Hot Encoding". Standardizacija numeričkih značajki

čajki transformira značajke tako da imaju srednju vrijednost 0 i standardnu devijaciju 1, dok se "One-Hot Encoding" koristi za pretvorbu kategorijskih značajki u binarne vektore koje model može shvatiti.

Nakon što se obavi normalizacija ulaznih podataka, oni se dijele u dva skupa: skup za treniranje i skup za testiranje. Skup za treniranje koristi se tijekom procesa učenja, gdje neuronska mreža prilagođava svoje težine kako bi što bolje predviđala ishode na temelju ulaznih podataka. Moguće je dodatno razdijeliti ga na skup za treniranje i skup za validaciju. S druge strane, skup za testiranje koristi se nakon procesa učenja kako bi se evaluirala točnost modela nad dosad neviđenim podacima. Za evaluaciju točnosti modela se često koristi MAE (Mean absolute error) - srednja apsolutna pogreška. Ona se računa kao prosjek apsolutnih razlika između predviđenih i stvarnih vrijednosti te prikazuje prosječnu pogrešku. Pored MAE koriste se i MSE (Mean Squared Error) te RMSE (Root Mean Squared Error).

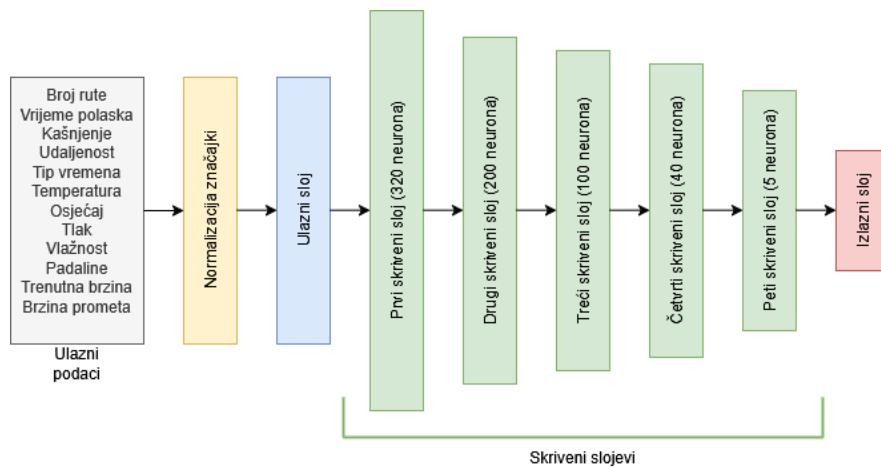
Proces učenja neuronske mreže se provodi algoritmom koji prilagođava težine veza između neurona s ciljem smanjenja pogreške između predviđenih i stvarnih podataka. Taj proces se ponavlja kroz više iteracija (epoha), dok mreža ne postigne zadovoljavajuću točnost. Ukratko, svaki neuron prima izlaze neurona iz prethodnog sloja s kojima je povezan, zatim ih sumira, dodaje pomak (bias) i na dobiveni rezultat primjenjuje aktivacijsku funkciju. Aktivacijska funkcija određuje izlaz neurona i omogućuje da izlaz nije samo linearna kombinacija ulaza. Izlaz neurona potom se šalje sljedećem sloju mreže. Nakon što se podaci propagiraju kroz sve slojeve, izračunava se ukupna pogreška modela. Tada se obavlja korigiranje težina te kod metode gradijentnog spusta korigiranje se provodi unatrag kroz mrežu — od izlaznog prema ulaznom sloju.

Kroz ovaj seminar proučene su četiri različite konfiguracije neuronskih mreža. Odabrane arhitekture razlikuju se po broju skrivenih slojeva i broju neurona po skrivenom sloju te primjenom dodatnih tehnika poput ranog zaustavljanja (early stopping), što omogućuje bolju generalizaciju i smanjenje prenaučenosti modela. U Tablici 3.1. prikazani su rezultati treniranja za svaku od konfiguracija, uključujući metrike pogreške na skupu za učenje, validaciju i testiranje.

Skriveni slojevi	64x32	128x64	320x200x100x40x5	320x200x100x40x5 + early stopping
Training Loss (MSE)	1580.2980	1561.3629	1115.0692	973.2731
Validation Loss (MSE)	1629.9922	1552.5287	1352.4648	1292.6776
Training MAE (sekunde)	29.2435	29.0271	24.1745	22.5490
Validation MAE (sekunde)	29.8322	28.8568	26.1842	25.3439
Test MAE (sekunde)	29.11	28.15	24.94	24.27

Tablica 3.1. Rezultati treniranja neuronskih mreža za različite konfiguracije

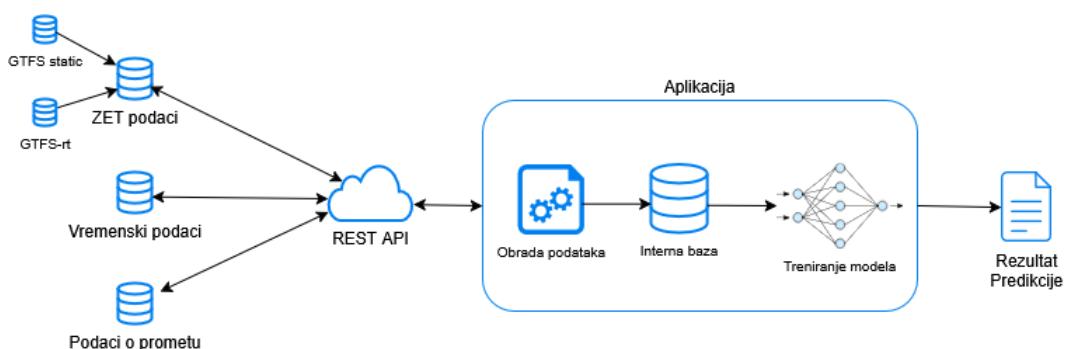
Analizom rezultata prikazanih u tablici vidljivo je da povećanje složenosti arhitekture neuronske mreže dovodi do smanjenja pogreške. Najjednostavnija konfiguracija, sastavljena od dva skrivena sloja (64x32), ostvarila je najveće vrijednosti pogrešaka na svim skupovima podataka. Dodavanjem dodatnih slojeva i povećanjem broja neurona značajno su poboljšane performanse modela, što je vidljivo kroz smanjenje vrijednosti MSE i MAE metrika. Najbolje rezultate postigla je mreža sačinjena od pet skrivenih slojeva (320x200x100x40x5), preuzeta iz rada Rashvanda i suradnika ([2]), nad kojom je dodatno primijenjena tehnika ranog zaustavljanja (early stopping). Ova konfiguracija ostvarila je najniže pogreške nad skupovima za treniranje, validaciju i testiranje te je odabrana kao referentni primjer za koji će biti prikazani grafovi rezultata učenja.



Slika 3.3. Odabrana struktura mreže

4. Prototipna aplikacija

Iako je prototipna aplikacija djelomično predstavljena u ranijim poglavljima, u nastavku slijedi sažeti prikaz njezine funkcionalnosti kako bi se pružio jasan pregled rada aplikacije.



Slika 4.1. Sažeti pregled sustava

Na dijagramu na slici 4.1. prikazan je pogled na sustav s visoke razine. Aplikacija pomoću REST API sučelja i odgovarajućih metoda dohvata podatke iz tri izvora:

- GTFS podaci sa službene stranice ZET-a
- Vremenski podaci preuzeti s platforme OpenWeatherMap
- Podaci o prometu preuzeti s TomTom

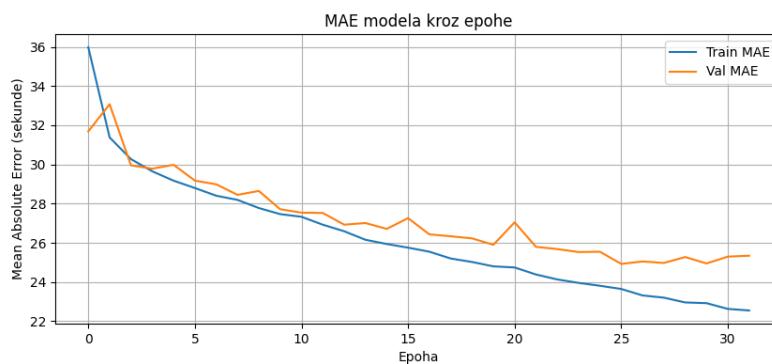
Usluge OpenWeatherMap i TomTom odabrane su zbog jednostavnosti korištenja te dovoljnog broja besplatnih API poziva za izradu ovog seminara. U slučaju daljnog razvoja aplikacije preporučljivo je razmatranje alternativnih opcija za dohvatanje podataka o stanju prometa poput Google Maps API, koji može ponuditi veću preciznost i skup podataka.

Obrada podataka provodi se prema postupku opisanom u poglavlju 2.2. Nakon obrade, podaci se pohranjuju u internu bazu podataka implementiranu u PostgreSQL-u. Baza sadrži tablicu promatranja koja se koristi kao ulaz za treniranje neuronske mreže. Proces treniranja modela moguće je pokrenuti ručno ili automatizirati kada se prikupi dovoljno podataka (batch obrada). U oba slučaja, podaci se najprije dohvataju iz baze podataka, zatim se nad njima provodi normalizacija i podjela na skupove za treniranje, validaciju i testiranje, nakon čega slijedi proces učenja neuronske mreže koji je opisan u poglavlju 3.1. U istom poglavlju prikazana je i odabrana arhitektura mreže. Istrenirani model moguće je spremiti kako bi se u budućnosti mogao koristiti za izvođenje predikcija. Za predikciju vremena dolaska tramvaja potrebno je unijeti trenutačnu lokaciju tramvaja, trenutačno vrijeme te stanje na cesti

4.1. Rezultati

Kao što je već navedeno u poglavlju 3.1., rezultati i grafovi temelje se na odabranoj neuronskoj mreži sa strukturon skrivenih slojeva 320x200x100x40x5 neurona.

Treniranje je provedeno na skupu od $N = 36687$ ulaznih podataka prikupljenih tijekom vremenskog razdoblja od pet dana. Od toga, 75 % podataka koristi se za trening i validacijski skup, dok je preostalih 25 % rezervirano za testni skup. Unutar skupa za treniranje, 10 % podataka izdvojeno je kao validacijski skup. Krivulja prosječne greške



Slika 4.2. Prosječna pogreška treniranja kroz epohe

tijekom treniranja na početku brzo opada, a zatim se pad usporava, što je tipično za neuronske mreže.

U nastavku je prikazan isječak konkretnih ispisa predikcija modela za neke testne primjere uspoređeni sa stvarnim vremenima dolaska.

Tramvaj 5 je krenuo sa stanice 22 u 15:34:43, očekivano vrijeme dolaska na

↪ narednu stanicu 23 je 15:37:14, stvarno vrijeme dolaska 15:36:35

Tramvaj 4 je krenuo sa stanice 5 u 16:21:07, očekivano vrijeme dolaska na

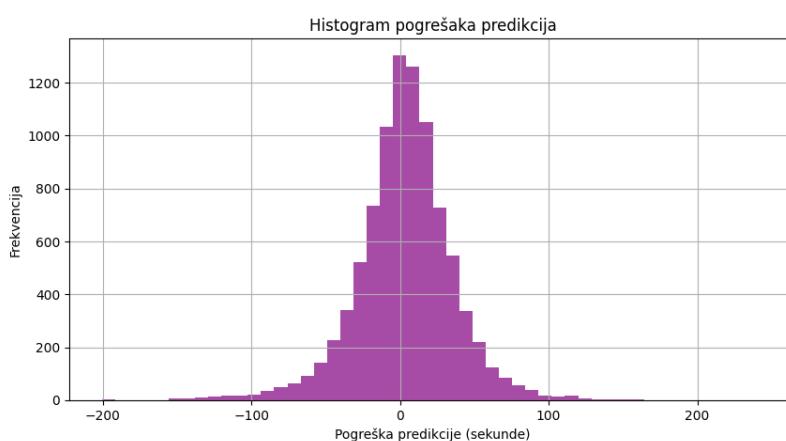
↪ narednu stanicu 6 je 16:22:11, stvarno vrijeme dolaska 16:22:14

Tramvaj 17 je krenuo sa stanice 15 u 17:36:22, očekivano vrijeme dolaska na

↪ narednu stanicu 16 je 17:37:29, stvarno vrijeme dolaska 17:37:27

Ako promotrimo sve testne primjere i njihove pogreške izražene u sekundama, možemo ih vizualizirati pomoću histograma koji prikazuje raspodjelu pogrešaka modela 4.3. Histogram poprima oblik približan normalnoj (Gaussovoj) distribuciji, što ukazuje da većina testnih primjera poprima malu pogrešku. Točnije, ispis modela prikazuje i postotak testnih primjera koji pripadaju određenom rasponu mјerenog od stvarnog vremena dolaska, pri čemu su rezultati:

- Rasponu od ± 10 sekundi od stvarnog dolaska pripada 30.61% testnih primjera
- Rasponu od ± 30 sekundi od stvarnog dolaska pripada 70.91% testnih primjera
- Rasponu od ± 60 sekundi od stvarnog dolaska pripada 92.85% testnih primjera
- Rasponu od ± 120 sekundi od stvarnog dolaska pripada 99.26% testnih primjera

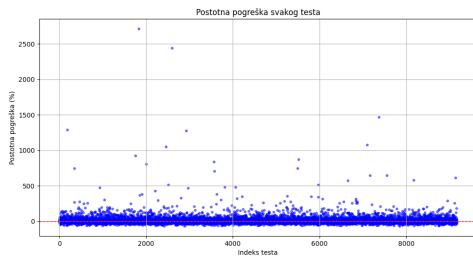


Slika 4.3. Histogram pogrešaka predikcije

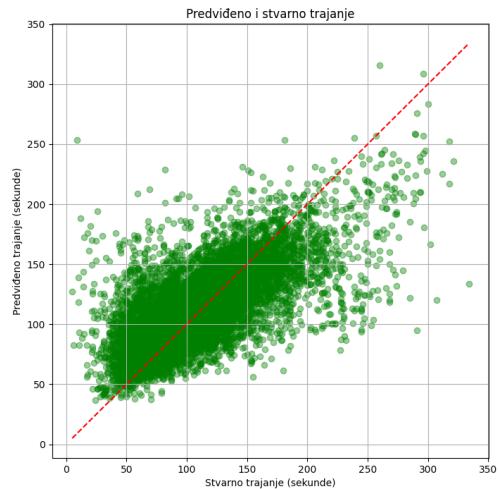
Vrijednosti su vrlo dobre ako ih usporedimo s istraživanjem C. Fors Johanssona, koji

je primjenom LSTM modela za prepoznavanje zavisnosti između stanica na liniji prijevoza ostvario 63,7% predviđanja unutar okvira od ± 1 minute [4]. Međutim, treba imati na umu da se rezultati ne bi trebali shvaćati previše ozbiljno zbog ograničenosti skupa podataka — korišteno je samo 36687 primjera prikupljenih tijekom 5 dana, u kojima nije bilo značajnih odstupanja u vremenu i prometu.

Pored histograma na slikama 4.4. i 4.5. prikazana su dva grafa koji također vizualno prikazuju pogrešku testnih primjera.



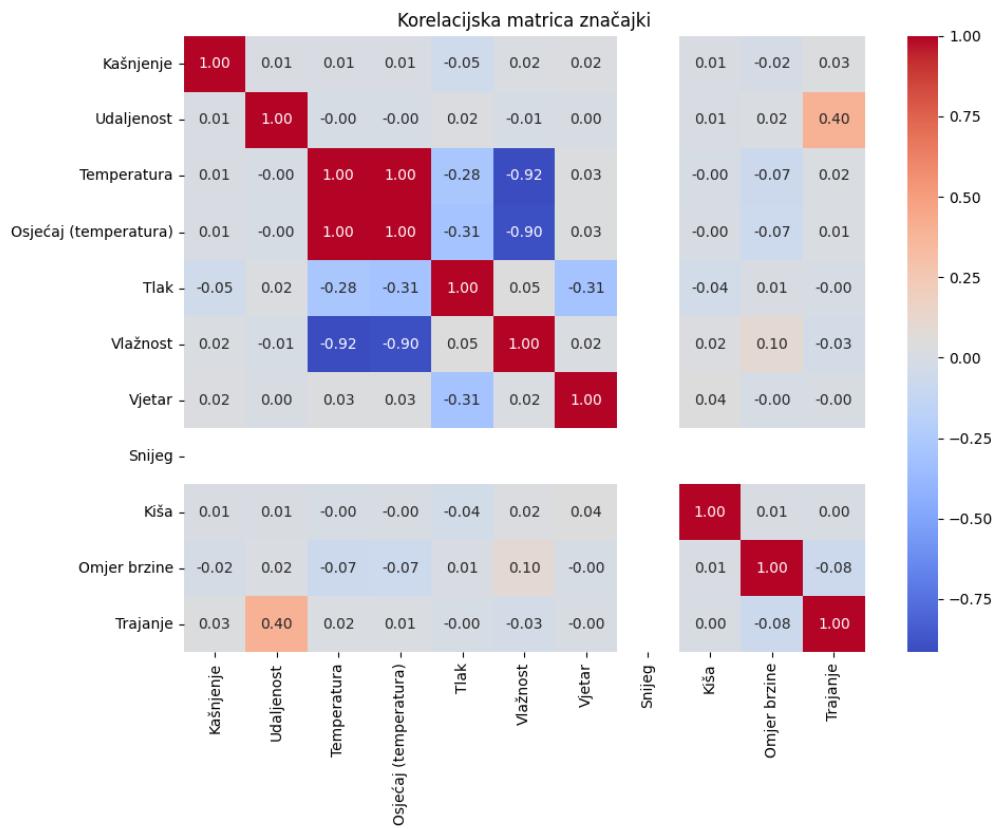
Slika 4.4. Postotna pogreška testnih primjera



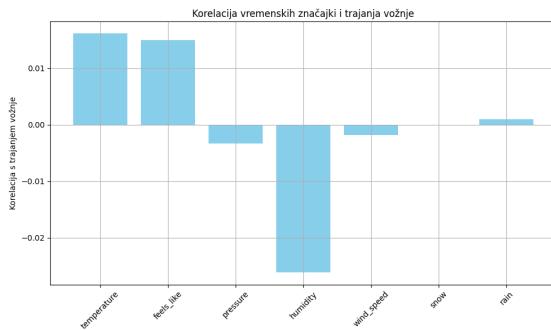
Slika 4.5. Ovisnost predviđenog vremena trajanja puta i stvarnog trajanja

Zanimljivo je proučiti graf na slici 4.6., koji prikazuje korelacijsku matricu svih značajki te omogućuje bolji uvid u međusobne odnose među podacima. Za vremenske značajke temperatura i osjećaj (feels_like temperatura) može se zaključiti da su u gotovo savršenoj korelaciji, što upućuje na to da jedna od njih nije nužna za treniranje modela. Također se može primijetiti da je temperatura u obrnutoj korelaciji s tlakom i vlagom, dok tlak i brzina vjetra pokazuju obrnutu korelaciju, što je u skladu s prirodnim zakonitostima, ali nema značajan utjecaj na model. Općenito vremenske značajke nemaju značajan utjecaj na predikcije modela što je vidljivo na grafu 4.7. te je u skladu s tvrdnjom istraživanja Rashvand i sur. ([2]) da vremenski uvjeti nemaju značajan utjecaj na pogrešku predviđanja dolazaka. Daljnjam proučavanjem matrice korelacija može se primijetiti da najveći utjecaj na pogrešku predviđanja dolazaka ima udaljenost (korelacija od 0.4) te brzina kretanja (korelacija od -0.08), što je u skladu s očekivanjima, budući da

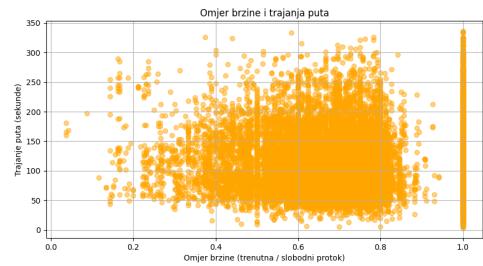
veća udaljenost i niža brzina doprinose dužem trajanju putovanja.



Slika 4.6. Matrica korelacija značajki



Slika 4.7. Korelacija vremenskih značajki i trajanja vožnje



Slika 4.8. Utjecaj brzine na trajanje puta

5. Zaključak

U ovom seminarskom radu prikazan je jedan mogući primjer sustava koji bi se mogao koristiti za praćenje javnog gradskog prijevoza i ostvarivanje predikcija dolazaka prijevoza. Takav sustav mogao bi se dalje implementirati unutar korisničke aplikacije koja bi omogućila pristup ovim informacijama široj javnosti, čime bi se unaprijedilo iskustvo i planiranje putovanja korisnika javnog prijevoza. Primjena neuronske mreže u ovom sustavu pokazala se kao dobar izbor, jer je omogućila relativno precizne i pouzdane predikcije. S obzirom na ograničenja rada, koja se prvenstveno odnose na veličinu i raznolikost skupa podataka – budući da je prikupljanje provedeno u relativno kratkom vremenskom razdoblju i s ograničenim resursima (pristup API-ju) - za budući rad i razvoj takvog sustava potrebno je raditi na proširenju i točnosti dostupnog skupa podataka te dodatna optimizacija neuronske mreže radi povećanja učinkovitosti i preciznosti modela. Zaključno, rad potvrđuje potencijal primjene umjetne inteligencije u unapređenju javnog gradskog prijevoza, što može doprinijeti većem zadovoljstvu korisnika i učinkovitijem prometnom sustavu.

Literatura

- [1] Grad Zagreb, Gradski ured za gospodarstvo, ekološku održivost i strategijsko planiranje, "Priopćenje o prometu za razdoblje vii.-xii. 2024." 2024. [Mrežno]. Adresa: https://www.zagreb.hr/userdocsimages/arhiva/statistika/2024/prijevoz%202024/Prijevoz%20I%20-%20XII.%202024_web.pdf
- [2] N. Rashvand, S. S. Hosseini, M. Azarbajani, i H. Tabkhi, "Real-time bus arrival prediction: A deep learning approach for enhanced urban mobility", 2024. [Mrežno]. Adresa: <https://arxiv.org/abs/2303.15495>
- [3] F. Sun, Y. Pan, J. White, i A. Dubey, "Real-time and predictive analytics for smart public transportation decision support system", 2016. [Mrežno]. Adresa: <https://ieeexplore.ieee.org/document/7501714>
- [4] C. Fors Johansson, "Arrival time predictions for buses using recurrent neural networks", 2019. [Mrežno]. Adresa: <https://liu.diva-portal.org/smash/get/diva2:1424184/FULLTEXT01.pdf>
- [5] Gtfs.org, "General transit feed specification", 2025. [Mrežno]. Adresa: <https://gtfs.org/getting-started/what-is-GTFS/>
- [6] B. D. Bašić, M. Čupić, i J. Šnajder, *Umjetne neuronske mreže*. Zagreb: Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2008.