

# ANALIZA TOČNOSTI RAČUNALNIH MODELA ZA PREDVIĐANJE METEOROLOŠKIH PODATAKA USPOREDBOM S MJERENJIMA

---

**Drokan, Dino**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:190:522691>

*Rights / Prava:* [Attribution 4.0 International](#) / [Imenovanje 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2025-02-03**



*Repository / Repozitorij:*

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

**ANALIZA TOČNOSTI RAČUNALNIH MODELA ZA  
PREDVIĐANJE METEOROLOŠKIH PODATAKA USPOREDBOM  
S MJERENJIMA**

Rijeka, srpanj 2024.

Dino Drokan

0069082791

SVEUČILIŠTE U RIJECI

**TEHNIČKI FAKULTET**

Diplomski sveučilišni studij računarstva

Diplomski rad

**ANALIZA TOČNOSTI RAČUNALNIH MODELA ZA  
PREDVIĐANJE METEOROLOŠKIH PODATAKA USPOREDBOM  
S MJERENJIMA**

Mentor: prof. dr. sc. Zoran Čarija

Komentor: dr. sc. Ivana Lučin

Rijeka, srpanj 2024.

Dino Drokan

0069082791

Rijeka, 19.03.2024.

Zavod: Zavod za mehaniku fluida i računalno inženjerstvo  
Predmet: Programiranje: skriptni jezici

## ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Dino Drokan (0069082791)**  
Studij: Sveučilišni diplomski studij računarstva (1400)  
Modul: Računalni sustavi (1442)

Zadatak: **Analiza točnosti računalnih modela za predviđanje meteoroloških podataka usporedbom s mjerenjima / Analysis of the accuracy of meteorological forecasting models by comparison with measurement data**

### Opis zadatka:

U radu je potrebno izraditi računalni programa za prikupljanje meteoroloških podataka za odabrana područja u Hrvatskoj koji će prikupljati vrijednosti izmjerene na meteorološkim postajama i vrijednosti prognoze vremena. Prikupljeni podaci će se koristiti za analizu i usporedbu točnosti predikcije matematičkih i računalnih modela s glavnim fokusom na temperaturu zraka. Dodatno, istražiti će se mogućnosti predikcije meteoroloških uvjeta pomoću metoda strojnog učenja na temelju povijesnih podataka o meteorološkim uvjetima.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 15.03.2024.

Mentor:  
prof. dr. sc. Zoran Čarija

Komentor:  
dr. sc. Ivana Lučin

Predsjednik povjerenstva za  
diplomski ispit:  
prof. dr. sc. Miroslav Joler

## **IZJAVA**

Sukladno članku 8. pravilnika o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 31. siječnja 2020., izjavljujem da sam samostalno izradio/izradila diplomski rad prema zadatku preuzetom dana 15. ožujka 2024.

---

Dino Drokan

## **ZAHVALA**

Zahvaljujem se kolegama, profesorima i mentorima na podršci i prenesenom znanju tijekom studija.

# SADRŽAJ

1. UVOD.....	1
2. VREMENSKA PROGNOZA.....	3
2.1. Povijest prognoziranja vremena.....	3
2.2. Numerička prognoza vremena .....	4
3. KORIŠTENE TEHNOLOGIJE.....	6
4. OPIS I PRIKUPLJANJE PODATAKA .....	8
4.1. Podaci Državnog hidrometeorološkog zavoda.....	8
4.2. Podaci Norveškog instituta za meteorologiju .....	12
4.3. Prikupljanje i pohrana podataka.....	13
5. ANALIZA TOČNOSTI VREMENSKE PROGNOZE.....	20
5.1. Analiza vremenske prognoze meteorologa .....	20
5.2. Analiza numeričke prognoze Norveškog instituta za meteorologiju za: tlak zraka, brzinu i smjer vjetera te relativnu vlažnost zraka .....	25
5.3. Analiza numeričke prognoze temperature DHMZ-a .....	30
5.4. Analiza numeričke prognoze količine oborina DHMZ-a .....	34
5.5. Analiza prognoze oborina Norveškog instituta za meteorologiju.....	38
5.6. Analiza prognoze temperature Norveškog instituta za meteorologiju.....	40
6. PREDIKCIJA TEMPERATURE POMOĆU STROJNOG UČENJA .....	44
6.1. Povijesni podaci temperature zraka .....	46
6.2. Predikcija temperature zraka gradient boosting algoritmom .....	47
6.2.1. Gradient boosting algoritam .....	47
6.2.2. Priprema podataka i izrada modela.....	48
6.2.3. Povećanje točnosti predikcije smanjenjem vremenskog prozora predikcije .....	56
6.3. Predikcija temperature zraka pomoću neuronske mreže .....	60
6.3.1. Rekurentne neuronske mreže i LSTM mreže .....	62
6.3.2. Predikcija temperature pomoću LSTM mreže.....	64
6.4. Usporedba gradient boosting algoritma i LSTM neuronske mreže .....	70
7. ZAKLJUČAK.....	71
8. LITERATURA .....	73
9. SAŽETAK.....	75
8. POPIS OZNAKA I KRATICA.....	76

# 1. UVOD

Meteorologija ima značajnu ulogu u raznim aspektima ljudskog života, od poljoprivrede i transporta do upravljanja rizicima i urbanog planiranja. Točna meteorološka prognoza ključna je za donošenje dobrih odluka i ublažavanje potencijalnih rizika povezanih s vremenskim nepogodama. Razni sektori ljudske djelatnosti oslanjaju se na točnu i pouzdanu vremensku prognozu. Kod zračne plovidbe zrakoplova, vremenska prognoza i trenutno stanje atmosfere su neophodne informacije za operativni rad. Magla ili jako niski oblaci mogu stvoriti nepovoljne uvjete za slijetanje i uzlijetanje zrakoplova. Tijekom leta postoji mogućnost pojave leda na zrakoplovu što može dovesti do prestanka rada raznih komponenti zrakoplova. Atmosferske turbulencije, odnosno, kaotične promjene u tlaku zraka koje stvaraju vjetrove različitih smjerova i brzina, stvaraju pritisak na strukturu zrakoplova te čine zračnu plovidbu, u tom trenutku, jako neugodnom. Sigurnost i udobnost morske plovidbe također ovisi o vremenu. Brzina i smjer vjetra, valovi, morske mijene i padaline su faktori koji utječu na morsku plovidbu. Radi efikasnosti prijenosa, stvoreni su razni načini kodiranja vremenske prognoze. Vremenska prognoza za morsku plovidbu se, između ostalih kanala prijenosa, šalje pomoću radio valova. Mnogi poljoprivredni procesi su pod utjecajem vremena. Na primjer, sušenje sijena je izvedivo samo po suhom vremenu. Dugotrajna razdoblja suše mogu uništiti usjeve raznih biljaka. Hladnija razdoblja u proljeće također mogu uništiti usjeve. Količina padalina je također jako bitna, ona ne smije biti premala niti prevelika [1].

S obzirom na ključnu ulogu vremenske prognoze u raznim ljudskim djelatnostima, cilj ovog rada je analizirati točnost meteoroloških prognostičkih modela usporedbom s podacima izmjerenim na mjernim postajama. Analizirati će se točnost računalnih modela u predviđanju temperature zraka, relativne vlažnosti zraka, količine oborina, tlaka zraka i smjera i brzine vjetra. Također, analizirati će se točnost predikcije temperature službenih meteorologa Državnog hidrometeorološkog zavoda (DHMZ). Meteorološki podaci su prikupljeni sa mrežnog sjedišta Hrvatskog hidrometeorološkog zavoda i sa mrežnog sjedišta Norveškog instituta za meteorologiju. Za potrebe prikupljanja podataka izrađen je računalni program u programskom jeziku Python koji pohranjuje meteorološke podatke u bazu podatka. Analiza točnosti prognoze napravljena je usporedbom podataka prognoze sa podacima mjerenja te je uspoređena numerička prognoza DHMZ-a i numerička prognoza Norveškog instituta za meteorologiju.



Umjetna inteligencija revolucionira razne industrije automatizacijom i ubrzanjem raznih poslovnih procesa. Obuhvaća širok raspon područja, od obrade prirodnog jezika, programiranja do robotike i računalnog vida. Sustavi umjetne inteligencije mogu analizirati velike količine podataka nevjerojatnom brzinom i preciznošću, omogućujući napredak u područjima kao što su zdravstvo i financije. Zbog naglog razvoja umjetne inteligencije i njezinog velikog potencijala primjene u meteorologiji, ovaj rad, uz analizu točnosti vremenske prognoze, ima za cilj istražiti mogućnosti metoda strojnog učenja u predviđanju temperature zraka na temelju povijesnih podataka o temperaturi.

Izvori, struktura i proces prikupljanja analiziranih meteoroloških podataka opisani su u četvrtom poglavlju, a analiza točnosti prognoza vremena zajedno sa rezultatima je zabilježena u petom poglavlju. Proces izrade i evaluacije dva modela strojnog učenja opisan je u šestom poglavlju rada.

## 2. VREMENSKA PROGNOZA

Prognoziranje vremena je proces predviđanja stanja atmosfere na određenom mjestu i u određenom vremenu u budućnosti. Uključuje analizu različitih meteoroloških podataka, kao što su temperatura, relativna vlažnost zraka, brzina vjetera, atmosferski tlak i oborine, kako bi se napravila prognoza budućih vremenskih uvjeta. Prognoza vremena može biti subjektivna i objektivna. Subjektivne prognoze izrađuje meteorolog prognostičar na osnovu raznih dostupnih izmjerenih podataka, izračuna više prognostičkih modela, ali i vlastitog iskustva. Objektivna ili računalna prognoza je izravan rezultat nekog atmosferskog prognostičkog modela. Prognostičke modele i objektivne prognoze razvijaju i izrađuju meteorolozi stručnjaci za numeričku prognozu vremena [2].

### 2.1. Povijest prognoziranja vremena

Drevne civilizacije, uključujući Babilonce i Kineze od 300. godine pr. Kr., oslanjale su se na uzorke oblaka i astrologiju za predviđanje vremena. Aristotel je dokumentirao vremenske uzorke u svom djelu "Meteorologija" (*latin.* Meteorologica) oko 350. godine prije Krista, dok je Teofrast napisao knjigu o vremenskoj prognozi koja se zove Knjiga Znakova. Istodobno, i drevni indijski astronomi razvili su metode predviđanja vremena. Drevne metode predviđanja vremena uglavnom su se temeljile na promatranim uzorcima događanja, poznatim kao prepoznavanje uzoraka. Na primjer, crveni zalasci sunca često su označavali lijepo vrijeme sljedeći dan. Ova opažanja, koja su se prenosila generacijama, formirala su skup znanja o predviđanju vremena. Međutim, nisu sva predviđanja bila pouzdana, a mnoga, kasnije, nisu izdržala rigorozna statistička ispitivanja. Drevne civilizacije ukazuju na rani ljudski interes i napore u razumijevanju i predviđanju vremena.

Francis Beaufort i Robert FitzRoy poznati su kao pioniri moderne vremenske prognoze. Unatoč početnom kriticizmu, njihov značajan doprinos unutar britanske mornarice i državne sfere postavio je temelje za moderne metode predviđanja vremena [1]. Beaufortov rad uključuje razvoj ljestvice snage vjetera i kodiranja vremenskih zapisa, izradu tablica plime i oseke i uvođenje izrade vremenskih zapisa opažanja u postajama britanske obalne straže. FitzRoy, imenovan 1854. godine da nadgleda novi državni odjel za prikupljanje vremenskih podataka, uspostavio je preteču modernog Meteorološkog ureda. Uz postaje obalne straže, kapetani na brodovima bili su zaduženi za prikupljanje i obradu vremenskih podataka na moru. Inspiriran razornom olujom 1859. godine, FitzRoy je predstavio

inovativne metode za predviđanje vremena u kojima je prvi put upotrebljen izraz "vremenska prognoza". Osnovao je prvu službu za upozoravanje na oluje i pokrenuo objavu dnevne vremenske prognoze u dnevnim novinama "The Times". FitzRoy-evi doprinosi značajno su unaprijedili mogućnosti predviđanja vremena [1]. Širenje električne telegrafске mreže omogućilo je stvaranje nacionalne mreže opažanja, omogućivši analizu podataka i brzo širenje vremenskih upozorenja. Dodatno, standardizirani telegrafski kodovi za podatke o vremenu i klasifikacije oblaka koje je uveo Luke Howard, poboljšali su točnost i učinkovitost izvješćivanja o vremenu. Kako se električna telegrafska mreža dalje širila, standardizirani rječnik za opisivanje oblaka postao je neophodan. Klasifikacije Luke-a Howarda pružile su ključni okvir za točan opis formacija oblaka. Ovaj standardizirani rječnik, u kombinaciji sa stalnim poboljšanjem tehnika promatranja i širenjem vremenskih upozorenja, učvrstio je temelje moderne vremenske prognoze i postavio temelje za sofisticirane metode koje se danas koriste [3].

## **2.2. Numerička prognoza vremena**

Numerička prognoza vremena oslanja se na matematičke modele atmosfere i mora za predviđanje vremena, koristeći trenutne i povijesne podatke mjerenja. Prvi pokušaji izrade numeričkih modela započinju u 1920-im godina, no numerički modeli su počeli generirati pouzdane rezultate tek uvođenjem računalnih simulacija 1950-ih godina. Danas su brojni globalni i regionalni modeli prognoze operativni u raznim zemljama, koristeći vremenska promatranja u stvarnom vremenu prikupljena s radio-sondi, meteoroloških satelita i drugih sustava opservacije. Obrada velikih skupova podataka i izvršavanje složenih izračuna potrebnih za moderno numeričko predviđanje vremena zahtijeva ogromnu računalnu snagu. Čak i uz sve veću snagu superračunala, sposobnost predviđanja numeričkih vremenskih modela proteže se na okvirno šest dana u budućnost. Čimbenici koji utječu na točnost numeričkih predviđanja uključuju gustoću i kvalitetu mjerenja koja se koriste kao ulazni podaci za predviđanja, zajedno s nedostacima u samim numeričkim modelima. Tehnike naknadne obrade kao što je statistika izlaza iz modela razvijene su kako bi se poboljšalo rukovanje pogreškama u numeričkim predviđanjima. Temeljni problem leži u kaotičnoj prirodi parcijalnih diferencijalnih jednadžbi koje opisuju atmosferu. Te jednadžbe je nemoguće točno riješiti, a male pogreške povećavaju se s vremenom (udvostručuju se otprilike svakih pet dana) [3]. Sadašnje razumijevanje je da kaotično ponašanje ograničava točne prognoze na, okvirno, 14 dana čak i uz točne ulazne podatke

i idealno točan model. Osim toga, parcijalne diferencijalne jednačbe korištene u modelu moraju se nadopuniti parametrima za sunčevo zračenje, vlažne procese (oblake i oborine), izmjenu topline, tlo, vegetaciju, površinske vode. U nastojanju da se kvantificira velika količina inherentne neizvjesnosti u numeričkim predviđanjima, uvedene su skupne prognoze kako bi olakšale prognozu. Pristup skupnih prognoza analizira višestruke prognoze stvorene pomoću pojedinačnog modela prognoze ili više modela [3].

Sustav predviđanja klime (*eng.* Climate Forecast System) je primjer numeričkog modela za prognozu vremena koji modelira interakcije između mora, kopna i atmosfere na globalnoj skali te izvršava numeričku predikciju vremena. Model (sustav) je razvijen od strane Američkog nacionalnog centra za predviđanje okoliša (NCEP). Veliki broj znanstvenika je sudjelovao u razvoju modela. Američki nacionalni centar za predviđanje okoliša također je razvio numerički sustav predikcije vremena nazvan Globalni sustav predikcije pomoću ansambla (*eng.* Global Ensemble Forecast System). Sustav generira 21 različitu prognozu, gdje je svaka prognoza takozvani član ansambla. Različite prognoze se koriste kako bi se uzela u obzir neizvjesnosti i nepotpunosti ulaznih podatka. Nakon generiranja svih prognoza, statističkim metodama računa se konačna prognoza.

ALADIN (Aire Limitée Adaptation dynamique Développement InterNational) je numerički model prognoze vremena sa ograničenom površinom primjene, odnosno, ALADIN je regionalni model. Cilj regionalnih modela je smanjiti troškove izvršavanja modela usporedno sa globalnim modelima, ali i dalje zadržati istu rezoluciju rezultata modela. Integrirani sustav prognoze (*eng.* Integrated Forecasting System) je numerički model razvijen od strane Europskog centra za srednjoročne vremenske prognoze (ECMWF) te se zajedno u kombinaciji sa ALADIN modelom koristi u numeričkoj prognozi DHMZ-a [4]. HERS je model numeričke prognoze vremena sa rezolucijom od 9 km i sa prognozom od 9 do 10 dana unaprijed. Model je razvio Europski centar za prognozu vremena srednjeg dometa. Norveški institut za meteorologiju koristi HERS model za numeričku predikciju vremena za područja koja ne pripadaju nordijskom području [5][6]. U ovome radu analizirani i uspoređeni su rezultati ALADIN i HERS numeričkih modela prognoze vremena. Analizirana je pojedinačna točnost svakog modela te su dva modela međusobno uspoređena na temelju točnosti predikcije temperature i točnosti predikcije količine padalina. Točnost ALADIN modela analizirana je na podacima predikcije temperature zraka i količine padalina, a točnost HERS modela analizirana je na podacima predikcije temperature zraka, količine padalina, relativne vlažnosti zraka, tlaka zraka te brzine i smjera vjetra.

### 3. KORIŠTENE TEHNOLOGIJE

Implementacija zadatka ovog rada napravljena je u potpunosti u programskom jeziku Python uz podršku raznih knjižnica funkcija i radnih okvira.

Python je programski jezik visoke razine (eng. high-level) koji je poznat po svojoj jednostavnoj sintaksi i čitljivosti. Jezik je kreirao Guido van Rossum te je prvi put objavljen 1991. godine. Python podržava module i pakete što potiče modularnost programa i ponovnu upotrebu koda. Python-ova standardna knjižnica funkcija, koja je lako dostupna putem raznih modula i paketa, omogućuje programerima da se bave širokim spektrom programskih zadataka bez potrebe za stvaranjem novih alata iz nule. Jezik ima raznoliku primjenu, od web i poslovnih aplikacija do edukacijskih i znanstvenih primjena. Lagano se uči te je zato često prvi izbor kao programski jezik za učenje programiranja. Filozofija dizajna jezika naglašava čitljivost koda, sa sintaksom koja programerima omogućuje izražavanje koncepata u manje redaka nego što je to moguće u jezicima kao što su C++ ili Javi. Python koristi dinamičko tipkanje, što znači da se ne mora deklarirati tip podataka varijable kod definicije varijable, nego Python automatski određuje vrstu podataka na temelju vrijednosti koja je dodijeljena. Štoviše, upravljanjem memorijom u Pythonu upravlja skupljač smeća, a jezik pruža strukture podataka visoke razine poput lista i rječnika koji su prikladni za pisanje složenih programa. Sve te značajke doprinose tome da Python postane jedan od najpopularnijih i najkorištenijih programskih jezika na svijetu [7].

Beautiful Soup je Python knjižnica funkcija dizajnirana za procese dohvaćanja podataka sa interneta (eng. web scraping), posebno za ekstrakciju podataka iz HTML i XML dokumenata. Omogućuje praktično i intuitivno sučelje za navigaciju i upravljanje sadržajem web stranica, što ga čini moćnim alatom za ekstrakciju podataka s web stranica. Beautiful Soup stvara strukturu stabla iz HTML/XML izvora, dopuštajući korisnicima pretraživanje, filtriranje i izdvajanje određenih elemenata ili atributa na temelju različitih kriterija. Fleksibilnost i jednostavnost korištenja čine ovu knjižnicu funkcija popularnom među programerima za zadatke kao što su internetsko rudarenje i ekstrakcija podataka [8].

SQLAlchemy je Python knjižnica funkcija koja pruža alate za rad sa SQL-om i sustav objektno-relacijskog mapiranja (ORM). Programerima omogućuje interakciju s bazama podataka koristeći Python kod, apstrahirajući složenost SQL upita i veza s bazom podataka. SQLAlchemy nudi

kvalitetno sučelje visoke razine za rad s bazom podataka, omogućujući korisnicima lagano stvaranje, čitanje, ažuriranje i brisanje podataka [9].

Pandas je snažna i svestrana biblioteka otvorenog koda za analizu i manipulaciju podataka. Pruža podatkovne strukture kao što su *Series* i *DataFrame*, koje omogućuju učinkovito rukovanje i analizu velikih skupova podataka. Sa svojom intuitivnom sintaksom i bogatom funkcionalnošću, pandas pojednostavljuje zadatke poput čišćenja, transformacije i agregacije podataka. Pandas se lako integrira s drugim bibliotekama kao što su NumPy, matplotlib i scikit-learn [10].

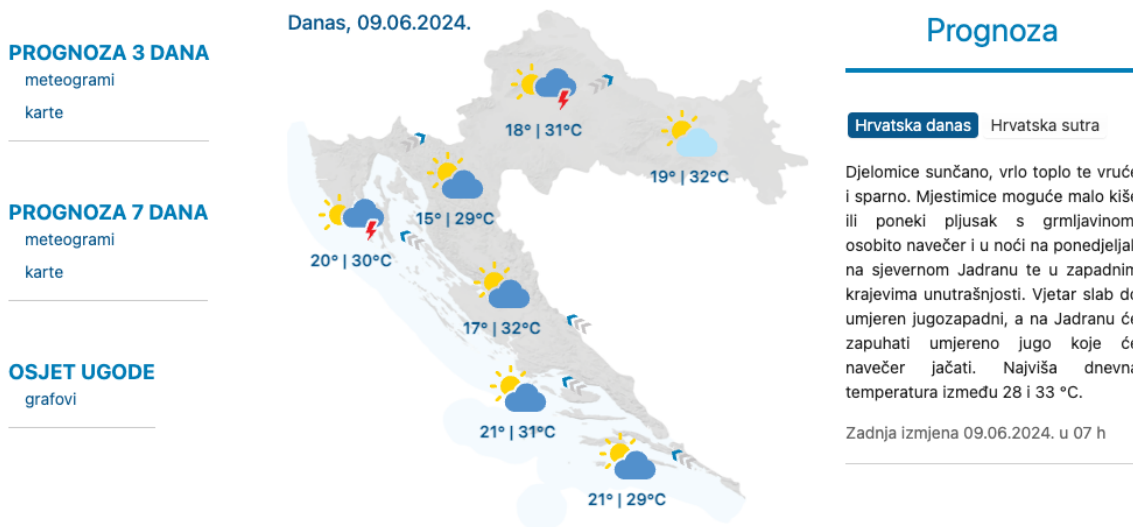
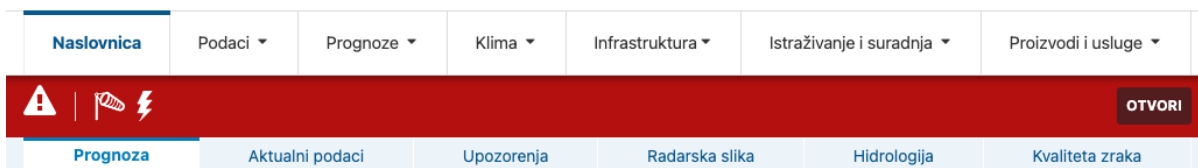
Matplotlib je biblioteka za izradu vizualizacija u Pythonu. Pruža fleksibilan programski okvir za izradu visokokvalitetnih dijagrama i grafikona, što ju čini bitnim alatom za vizualizaciju podataka. Matplotlibov svestrani API omogućuje korisnicima generiranje širokog raspona vizualnih prikaza, od jednostavnih linijskih grafova i trakastih dijagrama do složenih 3D prikaza. S opsežnim opcijama prilagodbe i podrškom za više izlaznih formata, Matplotlib je ključan alat u znanstvenoj zajednici i u zajednici za analizu podataka u Pythonu [11].

## **4. OPIS I PRIKUPLJANJE PODATAKA**

U nastavku, opisani su izvori, struktura, značenje i proces prikupljanja analiziranih podataka. Također, opisane su lokacije za koje su se prikupljali podaci kao i vremenski periodi u kojima su prikupljeni podaci. Korištena su dva veća izvora podataka: podaci Državnog hidrometeorološkog zavoda (DHMZ) i podaci Norveškog instituta za meteorologiju. Internet sjedišta DHMZ-a i Norveškog instituta za meteorologiju dostupni su na sljedećim poveznicama: [www.meteo.hr](http://www.meteo.hr), [www.met.no](http://www.met.no). DHMZ obavlja stručne poslove praćenja hidroloških i meteoroloških procesa, prikupljanja, obrađivanja i objavljivanja meteoroloških i hidroloških podataka. Norveški institut za meteorologiju je službeni državni institut za meteorologiju osnovan 1866. godine koji se također, između ostalog, bavi prikupljanjem, obradom i prognoziranjem meteoroloških podataka. Podaci su prikupljeni u periodu od 2.4.2024. do 10.6.2024., što obuhvaća period malo veći od dva mjeseca.

### **4.1. Podaci Državnog hidrometeorološkog zavoda**

Državni hidrometeorološki zavod (DHMZ), osnovan 27. kolovoza 1947., je državna upravna i znanstvena organizacija koja obavlja stručne poslove praćenja hidroloških i meteoroloških procesa, prikupljanja, obrađivanja i objavljivanja meteoroloških i hidroloških podataka, poslove istraživanja atmosfere i vodnih resursa te različite primjene meteorologije i hidrologije u područjima klimatologije, pomorske meteorologije, agrometeorologije, zrakoplovne meteorologije, prostornog planiranja i projektiranja i ostalih primjena. DHMZ upravlja i mrežom meteoroloških i hidroloških postaja i centara. DHMZ tako svakodnevno objavljuje: aktualne meteorološke i hidrološke podatke, podatke kvalitete zraka, vremenske i hidrološke analize i prognoze, opća i specijalna upozorenja na opasne vremenske i hidrološke pojave, mjereni i prognozirani UV indeks i mnoge druge informacije. Od vremenskih prognoza, DHMZ nudi subjektivne prognoze za Zagreb i za regije Republike Hrvatske (RH). Nudi subjektivnu prognozu za nautičare i pomorce te agrometeorološku i bio meteorološku prognozu. DHMZ nudi i numeričku prognozu koja je numerički rezultat atmosferskih prognostičkih modela ALADIN i modela Europskog centra za srednjoročne vremenske prognoze (ECMWF).



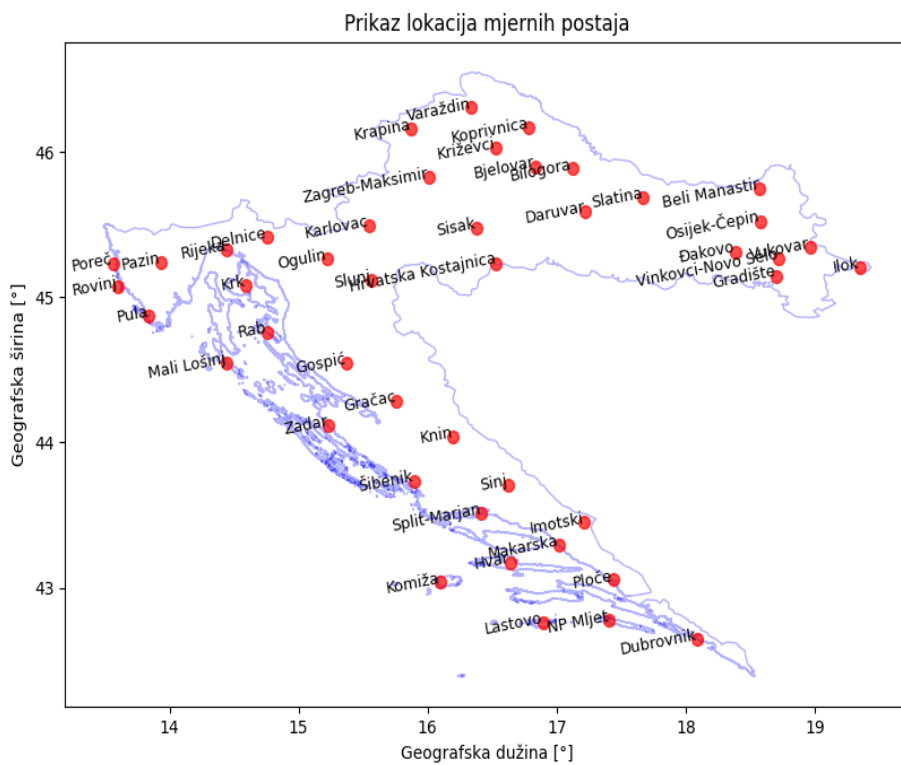
*Slika 4.1. Prikaz početne internetske stranice DHMZ-a*

Aktualni meteorološki podaci su podaci koji se mjere na mjernim postajama. Podaci mjerenja sa mjernih postaja sadrže informacije o temperaturi, relativnoj vlažnosti zraka, tlaku zraka, tendenciji tlaka zraka, brzini i smjeru vjetra te stanju vremena na svakoj mjernoj postaji DHMZ-a u Hrvatskoj. Parametri se mjere svakih sat vremena te se podaci na internetskom sučelju ažuriraju nekoliko minuta nakon punog sata. Podaci mjerenja količine oborina sadrže informacije o količini oborina u proteklih 24 sata za svaku mjernu postaju na kojoj su zabilježene oborine. Podaci se mjere i ažuriraju svakoga dana u 8 sati ujutro. Podaci predikcije vremena pomoću numeričkih modela sadrže sedmodnevnu numeričku prognozu vremena DHMZ-a sa vremenskom rezolucijom od 3 sata (prognoza se izrađuje za svaki treći sat u budućnosti). Numerički model prognozira količinu oborina, smjer i jačinu vjetra, stanje oblaka i temperaturu. Podaci subjektivne prognoze od meteorologa su podaci koje je napravio meteorolog na temelju mjerenja, izračuna, opservacija i vlastitog iskustva. Subjektivne prognoze su raspoređene na sedam regija Hrvatske i na grad Zagreb te su dostupni podaci o minimalnoj i maksimalnoj temperaturi, jačini vjetra i stanju oblaka. Dostupan je i tekst prognoze koji je napisao meteorolog. Podaci subjektivne prognoze ažuriraju se svakoga dana ujutro.





Slika 4.2. Prikaz regija Republike Hrvatske



Slika 4.3. Prikaz lokacija mjernih postaja DHMZ-a koje su uzete u obzir u ovom radu

Sa internetskog sučelja DHMZ-a prikupljena su tri seta podatka: meteorološki podaci mjerenja s mjernih postaja (mjerenja temperature, relativne vlažnosti zraka, tlaka zraka, tendencije tlaka zraka, količine oborina, brzine i smjera vjetra), podaci predikcije vremena pomoću numeričkog modela te prognoza vremena od strane meteorologa. Internet sučelje DHMZ-a omogućuje pristup raznim podacima u XML formatu zapisa. Podacima se pristupa HTTP GET zahtjevom prema određenim mrežnim mjestima. Tablica 4.1. prikazuje mrežna mjesta s kojih su prikupljeni meteorološki podaci DHMZ-a. Aktualni podaci mjerenja i oborina s mjernih postaja prikupljeni su za sve mjerne postaje u Hrvatskoj. Sedmodnevna numerička prognoza prikupljena je za sva dostupna mjesta prognoze, a subjektivna prognoza je prikupljena za Zagreb i za regije Hrvatske. Mjerenja pomoću kojih je napravljena analiza točnosti su uzeta iz podataka mjerenja sa mjernih postaja za lokacije koje su prikazane na slici 4.3.

*Tablica 4.1. Mrežna mjesta određenih meteoroloških podatka DHMZ-a*

<b>Prognoza</b>	<b>Mrežno mjesto podataka prognoze</b>
Aktualni podaci sa mjernih postaja	<a href="https://vrijeme.hr/hrvatska_n.xml">https://vrijeme.hr/hrvatska_n.xml</a>
Aktualni podaci količine oborina sa mjernih postaja	<a href="https://vrijeme.hr/oborina.xml">https://vrijeme.hr/oborina.xml</a>
Sedmodnevna numerička prognoza	<a href="https://prognoza.hr/sedam/hrvatska/7d_meteogrami.xml">https://prognoza.hr/sedam/hrvatska/7d_meteogrami.xml</a>
Subjektivna prognoza za trenutni dan	<a href="https://prognoza.hr/prognoza_danas.xml">https://prognoza.hr/prognoza_danas.xml</a>
Subjektivna prognoza za naredni dan	<a href="https://prognoza.hr/prognoza_sutra.xml">https://prognoza.hr/prognoza_sutra.xml</a>

## 4.2. Podaci Norveškog instituta za meteorologiju

Norveški meteorološki institut prognozira vrijeme, prati klimu i provodi istraživanja. Otkako je institut osnovan 1866. godine, norveški meteorolozi zauzeli su značajnu ulogu u razvoju znanstvene discipline meteorologije. Norveški meteorološki institut je vodeći međunarodni stručni centar u području meteorologije. Norveški meteorološki institut pruža razne numeričke i subjektivne prognoze. Za numeričku prognozu na području Norveške, koristi se MetCoOp EPS (MEPS) model s rezolucijom od 2.5 kilometra između točki izračuna. Za Arktičko područje koristi se Arome-Arctic model s rezolucijom od 2.5 kilometara. A za ostatak svijeta koristi se HERS model s rezolucijom od 9 kilometara. Numerička prognoza, dostupna je za bilo koju lokaciju na planeti. Zbog značajne uloge Norveškog meteorološkog instituta u području meteorologije, ovaj rad, uz ostale analize, analizira i točnost HERS modela numeričke prognoze.

U ovom radu prikupljeni su numerički podaci modela HERS te su uspoređeni sa ALADIN modelom koje koristi DHMZ. Podaci modela pružaju predikciju vremena za narednih 9 dana za odabranu lokaciju definiranu geografskim koordinatama. Podaci sadrže predikciju temperature, brzine i smjera vjetera, relativne vlažnosti zraka, tlaka, stanja oblaka, magle, niskih oblaka, visokih oblaka i padalina. Podaci se odnose na područje Zagreba. Predikcije su dostupne za svaki sat za naredna tri do četiri dana, zatim su dostupne predikcije za svaki šesti sat za sljedećih pet do šest dana. Ažuriranje podataka nije periodično te se događa svakih nekoliko sati. Podaci su dostupni putem internet sučelja na sljedećoj mrežnoj lokaciji: <https://api.met.no/weatherapi/locationforecast/2.0/classic?lat=45.82&lon=16.01&altitude=130> .

Served by the Norwegian Meteorological Institute and NRK

YR Search Menu

My locations

	Sunday	Monday	Tuesday	Wednesday
Zagreb ☆	☀️ 28°	☁️ 25°	☁️ 20°	☀️ 22°
Oslo ☆	☀️ 14°	☀️ 17°	☀️ 20°	☁️ 15°
Trondheim ☆	☁️ 12°	☁️ 13°	☁️ 11°	☀️ 14°
Tromsø ☆	☀️ 18°	☀️ 20°	☀️ 18°	☁️ 9°
Berlin ☆	☀️ 18°	☀️ 18°	☀️ 16°	☀️ 17°

Map Weather warnings Water temperatures Mountain passes

Other services from The Norwegian Meteorological Institute

Subjective forecast chart Text forecast for Norway Satellite images

Yr is served by NRK Meteorologisk institutt

See all help articles Contact us Choose language / velg Help

Slika 4.4. Prikaz početne internet stranice YR-a, internet sjedišta za posluživanje prognoze Norveškog instituta za meteorologiju

### 4.3. Prikupljanje i pohrana podataka

Podatke je potrebno kontinuirano prikupljati kako bi se kreirao skup podataka koji je dovoljno velik i reprezentativan za analizu. Za potrebu prikupljanja podataka napisana je Python skripta naziva *DohvatPodataka.py*. Skripta dohvaća podatke sa izvora podataka, pretražuje XML strukturu podataka, uzima potrebne podatke, radi transformaciju nad podacima te sprema podatke u bazu podataka. Skripta se sastoji od nekoliko funkcija i glavne petlje programa. Glavna petlja programa je beskonačna petlja koja pomoću poziva određenih funkcija periodično prikuplja podatke. Petlja je vizualno prikazana na slici 4.5., a kod je prikazan u isječku koda 4.1. Podaci numeričkih modela DHMZ-a i YR-a nemaju određenu frekvenciju ažuriranja stoga je prilikom svakog dohвата podataka numeričkih modela potrebno provjeriti jesu li se podaci ažurirali.

```

while True:
    trenutnoVrijeme = datetime.now()
    print("Trenutno vrijeme dohvata: ", trenutnoVrijeme)
    if trenutnoVrijeme.hour == 8:
        dohvatiMjerenjaOborina()
    if trenutnoVrijeme.hour == 10:
        dohvatiSubjektivnuPrognozu()

    dohvatiMjerenjaPostaja()
    dohvatiDHMZRacunalnuPrognozu()
    dohvatiYRRacunalnuPrognozu()

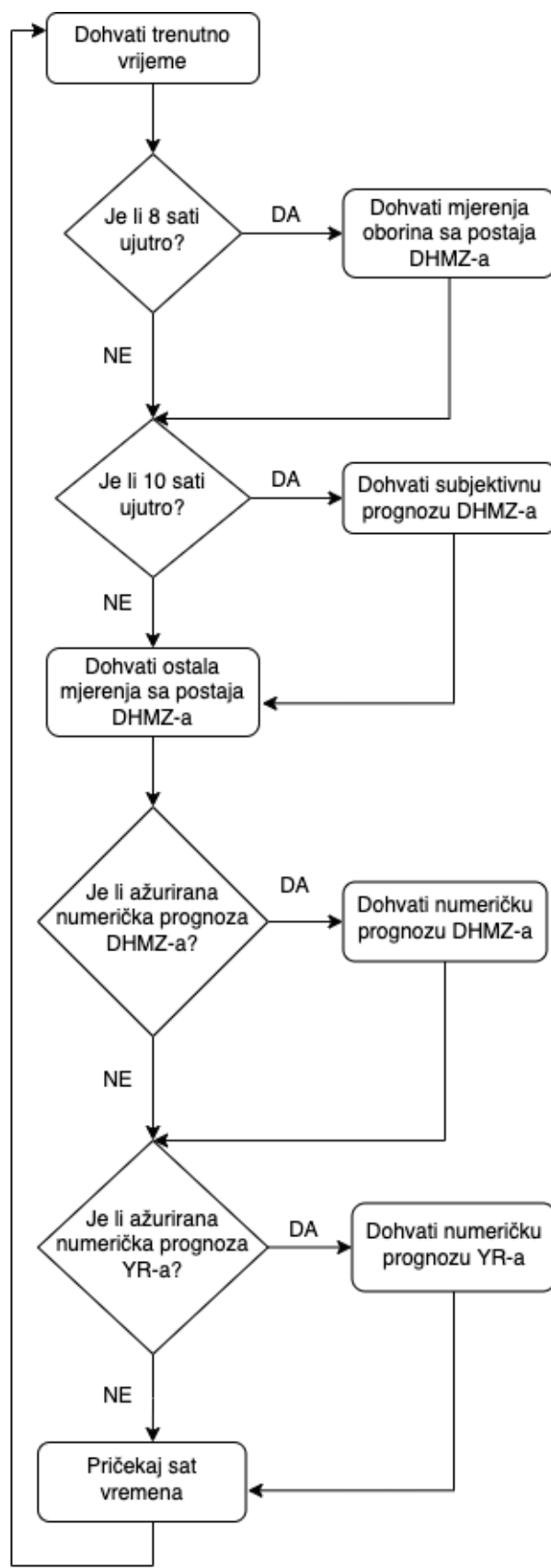
    with open("LOG/DohvatPodatakaLog.log", 'a') as file:
        file.write(str(trenutnoVrijeme) + "\n")

    time.sleep(3595)

```

#### *Isječak koda 4.1. Glavna petlja programa dohvata podataka*

Programski kod petlje se izvršava svakih sat vremena, što je omogućeno pozivom funkcije *time.sleep(3595)* koja će zaustaviti izvođenje programa na 3595 sekundi te će zatim nastaviti izvođenje. Funkcija zaustavlja izvođenje pet sekundi kraće od punog sata zbog toga što je potrebno vrijeme da se funkcije dohvata podataka izvrše. Na taj način je povećana točnost periodičnosti dohvata podataka. Funkcije *dohvatiYRRacunalnuPrognozu()*, *dohvatiDHMZRacunalnuPrognozu()* i *dohvatiMjerenjaPostaja()* se pozivaju svakih sat vremena, dok se funkcija *dohvatiMjerenjaOborina()* poziva svakog dana u 8 sati ujutro, a funkcija *dohvatiSubjektivnuPrognozu()* se poziva svakog dana u 10 sati ujutro.



Slika 4.5. Prikaz dijagrama toka beskonačne petlje za prikupljanje podataka

Funkcija *dohvatiMjerenjaPostaja()* dohvaća XML podatke mjerenja sa mjernih postaja DHMZ-a sa internetskog izvora podataka pomoću funkcije *XMLuBS()* koja vraća podatke mjerenja sa mjernih postaja zajedno sa vremenskim trenutkom. Funkcija *XMLuBS(URL)* je pomoćna funkcija koja će: napraviti HTTP GET zahtjev prema URL resursu, napraviti BeautifulSoup objekt pomoću dohvaćenog XML-a te vratiti objekt sa podacima zajedno sa vremenskim trenutkom dohvata podataka. Dodatno, funkcija radi provjeru statusa rezultata HTTP GET zahtjeva. Zatim, nakon dohvata podataka, funkcija *dohvatiMjerenjaPostaja()* ispisuje u konzolu vremensku oznaku mjerenja radi lakšeg praćenja izvođenja programa. Nakon toga, inicijalizira se prazna lista, pod nazivom *data*, za pohranu podataka za svaku meteorološku mjernu postaju. Kod iterira kroz svaku meteorološku stanicu u XML podacima i izdvaja sljedeće podatke: naziv postaje, temperatura, vlažnost zraka, tlak, tendencija tlaka, brzina vjetra, smjer vjetra i vremenski uvjeti. Ako je podatak predstavljen sa "-" znakom, odnosno, podatak ne postoji, zamjenjuje je s "NULL" ili sa praznim nizom znakova. Za svaku mjernu postaju, ekstrahirani podaci se dodaju kao nova lista u *data* listu podataka. Na kraju, kod iterira kroz sve podatke i konstruira SQL INSERT naredbe za umetanje podataka u tablicu baze podataka pod nazivom *podaci\_postaja*. Svaki unos u listi podataka predstavlja jedan redak u tablici baze podataka.

```

for zapis in data:
    insertString = "INSERT INTO diplomski.podaci_postaja " \
                  "VALUES ('" + zapis[0] + "', " + zapis[1] + \
                  ", " + zapis[2] + ",'" + zapis[3] + \
                  "', '" + zapis[4] + "', '" + zapis[5] + "', '" \
                  + zapis[6] + "', '" + zapis[7] + "', '" + \
                  zapis[8] + "')"
    res = connection.execute(insertString)

```

#### *Isječak koda 4.2. Konstrukcija SQL upita*

Funkcija *dohvatiDHMZRacunalnaPrognozu()* dohvaća numeričku vremensku prognozu DHMZ-a za narednih 7 dana. Dohvaća XML podatke i izdvaja vremenski trenutak posljednjeg ažuriranja prognoze. Ako prognoza nije ažurirana, ispisuje se poruka koja pokazuje da prognoza nije ažurirana i funkcija završava izvođenje. Inače, ako je prognoza ažurirana, funkcija ažurira globalnu varijablu *zadnjaIzmjenaDHMZ* novim vremenskim trenutkom i ispisuje vremenski trenutak posljednjeg ažuriranja. Funkcija zatim formatira vremensku oznaku posljednjeg ažuriranja i pretvara je u objekt datuma i vremena. Inicijalizira praznu listu pod nazivom *data* za pohranu podataka prognoze. Zatim

prolazi kroz sve podatke svakog grada u XML podacima i izvlači podatke prognoze za svaki 3-satni interval za sljedećih 7 dana. Za svaki unos prognoze funkcija izdvaja podatke temperature i količina oborine, formatira vremensku oznaku trenutka prognoze (vremenski trenutak na koji se odnosi prognoza) i dodaje podatke u listu svih podataka. Na kraju, iterira kroz listu podataka i konstruira SQL INSERT naredbe za unos podataka prognoze u tablicu baze podataka pod nazivom *prognoza\_7\_dana*.

```
for dan in prognoza:
    datum = str(dan['datum'])
    sat = str(dan['sat'])
    temperatura = dan.find('t_2m').text
    kolicinaOborina = dan.find('oborina').text

    # izrada timestamp-a prognoziranog trenutka
    datumTS = datetime.strptime(datum, '%d.%m.%Y.')
    timeTS = datetime.strptime(sat, '%H')
    timestampPrognoze = datetime(year=datumTS.year, month=datumTS.month,
                                  day=datumTS.day,
                                  hour=timeTS.hour, minute=0, second=0)
    timestampPrognozeStr = timestampPrognoze.strftime('%Y-%m-%d %H:%M:%S')
    data.append(
        [ime,
         temperatura,
         kolicinaOborina, timestampPrognozeStr, vremenskiTrenutak])
```

#### *Isječak koda 4.3. Ekstrakcija i transformacija podataka prognoze*

Funkcija *dohvatiYRRacunalnuPrognozu()* dohvaća i pohranjuje podatke numeričke vremenske prognoze sa API-ja Norveškog instituta za meteorologiju. Nakon dohvata podataka, funkcija pretvara vremensku oznaku u objekt datuma i vremena i dodaje vremenskoj oznaci dva sata kako bi se vrijeme pretvorilo u centralno europsko vrijeme. Putem globalne varijable naziva *zadnjaIzmjenaYR* prati se jesu li dohvaćeni podaci ažurirani, ako jesu, funkcija nastavlja sa izvršavanjem. Funkcija zatim analizira podatke prognoze za svaki budući sat i izdvaja podatke temperature, vlažnosti zraka, tlaka, smjera vjetra, brzina vjetra, količina padalina. Konstruira i izvršava SQL INSERT naredbe za unos podataka prognoze u odgovarajuće tablice u bazi podataka: tablica *prognoza\_oborine\_yr* pohranjuje podatke o oborinama, tablica *prognoza\_yr* pohranjuje ostale podatke prognoze.

Funkcija *dohvatiMjerenjaOborina()* dohvaća mjerenja količine oborina u proteklih 24 sata. Izvor podataka je DHMZ. Mjerenja su izmjerena na mjernim postajama te se podaci ažuriraju svakog dana u 8 sati ujutro. Funkcija preuzima XML podatke te ih ekstrahira i zapisuje u tablicu naziva *mjerenja\_oborina* u bazi podataka.



Tablica 4.2. Prikaz podskupa podatka iz tablice mjerenja\_oborina

Postaja	Količina oborina [mm]	Vremenski trenutak
Bednja	0.7	2024-04-03 08:54:52
Beli Manastir	8.1	2024-04-03 08:54:52
RC Bilogora	5.6	2024-04-03 08:54:52
Bjelovar	6.8	2024-04-03 08:54:52
Brestovac	3	2024-04-03 08:54:52

Funkcija *dohvatiSubjektivnuPrognozu()* dohvaća i pohranjuje podatke subjektivne DHMZ prognoze temperature za trenutni i naredni dan. Izvršava se svakoga dana u 10 sati ujutro, nakon što se prognoza ažurira na internet DHMZ izvoru. Funkcija dohvaća podatke prognoze temperature za trenutni dan, ekstrahira podatke te ih zapisuje u tablicu baze podatka, tablica se zove *prognoza\_subjektivna*. Zatim funkcija preuzima podatke prognoze za naredni dan te ih ekstrahira i pohranjuje u tablicu naziva *prognoza\_subjektivna*.

Tablica 4.3. prikazuje podskup podataka iz tablice *prognoza\_subjektivna* u bazi podataka. Stupac *Regija* označava regiju RH na koju se odnosi prognoza, stupci *Temperatura<sub>min</sub>* i *Temperatura<sub>max</sub>* označavaju prognoziranu minimalnu i maksimalnu temperaturu u danu. Stupac *Vremenski trenutak prognoze* označava dan na koji se odnosi prognoza, a stupac *Vremenski trenutak izrade prognoze* označava vrijeme kada se je prognoza napravila. Stoga, četvrti redak tablice se interpretira na sljedeći način: „Dana 3.4.2024. u 10 sati ujutro, predviđa se da će dana 4.4.2024. u središnjoj hrvatskoj minimalna i maksimalna temperatura zraka iznositi 7 i 22 stupnja Celzijeva“. DHMZ-ov API ne pruža informaciju o točnom vremenu izrade subjektivne prognoze. Ručnom provjerom utvrđeno je da je vrijeme izrade prognoze, uglavnom, svakog dana u 10 sati ujutro te će se za analizu koristiti to vrijeme.

*Tablica 4.3. Prikaz podskupa podatka iz tablice prognoza\_subjektivna*

<b>Regija</b>	<b>Temperatura<sub>min</sub> [°C]</b>	<b>Temperatura<sub>max</sub> [°C]</b>	<b>Vremenski trenutak prognoze</b>	<b>Vremenski trenutak izrade prognoze</b>
središnja	4	17	2024-04-03	2024-04-03 10:54:59
istočna	6	20	2024-04-03	2024-04-03 10:54:59
gorska	6	15	2024-04-03	2024-04-03 10:54:59
središnja	7	22	2024-04-04	2024-04-03 10:54:59
istočna	8	23	2024-04-04	2024-04-03 10:54:59
gorska	5	18	2024-04-04	2024-04-03 10:54:59

## 5. ANALIZA TOČNOSTI VREMENSKE PROGNOZE

U sljedećim potpoglavljima opisani su postupci analize podatka i prikazani su rezultati analize. Analize podataka su napravljene pomoću Python skripti. Podaci su ekstrahirani iz baze podataka u CSV datoteke te su zatim učitani u Python skripte. Podaci i skripte su priloženi u dodatku rada.

### 5.1. Analiza vremenske prognoze meteorologa

Subjektivna prognoza meteorologa je dnevna prognoza gdje meteorolog predviđa minimalnu temperaturu zraka, maksimalnu temperaturu zraka, tip i smjer vjetra, oborine te stanje oblaka. Zbog prirode podataka prognoze, u ovoj analizi uspoređivati će se točnost prognozirane minimalne i maksimalne temperature zraka. Podaci prognozirane temperature uspoređivati će se sa podacima izmjerene temperature. Za potrebu analize vremenske prognoze meteorologa koriste se samo izmjereni podaci temperature zraka.

Cilj analize je izračunati prosječno apsolutno odstupanje prognozirane maksimalne i minimalne temperature zraka od izmjerene maksimalne i minimalne temperature zraka. Analiza je napravljena po regijama Hrvatske (uključujući i Zagreb) i za cijelu Hrvatsku.

U programskom kodu skripte analize, podaci iz tablice *prognoza\_subjektivna* su učitani u strukturu podataka, podatkovni okvir (*eng.* Dataframe), naziva *prognozaSubjektivna*, a podaci iz tablice *podaci\_postaja* su učitani u podatkovni okvir naziva *podaciPostaja*. Svaka mjerna postaja je pridružena jednoj od sedam regija, a mjerne postaje grada Zagreba su grupirane zajedno u jednu skupinu. Zatim se iterira kroz svaki zapis podatkovnog okvira *prognozaSubjektivna* te se za svaku prognoziranu minimalnu ( $T_{\min}$ ) i maksimalnu temperaturu ( $T_{\max}$ ) određene regije računaju prosječna izmjerena minimalna ( $T_{\min\_avg}$ ) i prosječna izmjerena maksimalna ( $T_{\max\_avg}$ ) temperatura za tu regiju. Prosječna izmjerena minimalna temperatura regije se računa tako što se zbroje sve minimalne temperature izmjerene taj dan u regiji te se suma podijeli sa brojem mjerenja. Na isti način se računa i prosječna maksimalna temperatura regije. Podskup podataka usporedbe je prikazan u tablicama 5.1. i 5.2. gdje tablica 5.1. prikazuje usporedbu prognoze s mjerenjima za trenutni dan a tablica 5.2. usporedbu prognoze s mjerenjima za naredni dan. U tablicama se može primijetiti odstupanje izmjerenih vrijednosti od prognoziranih vrijednosti. Odstupanje se kreće oko 1 °C što se je i vidljivo u rezultatima analize u tablici 5.3. Programski kod izračuna prikazan je u isječku koda 5.1.

*Tablica 5.1. Prikaz podskupa podataka usporedbe subjektivne prognoze s mjerenjima za trenutni dan*

<b>Regija</b>	<b>T<sub>min</sub> [°C ]</b>	<b>T<sub>max</sub> [°C ]</b>	<b>T<sub>min_avg</sub> [°C ]</b>	<b>T<sub>max_avg</sub> [°C ]</b>
Središnja	4	17	4.58	16.18
Istočna	6	20	5.86	18.97
Gorska	6	15	6.62	12.49

*Tablica 5.2. Prikaz podskupa podataka usporedbe subjektivne prognoze s mjerenjima za naredni dan*

<b>Regija</b>	<b>T<sub>min</sub> [°C ]</b>	<b>T<sub>max</sub> [°C ]</b>	<b>T<sub>min_avg</sub> [°C ]</b>	<b>T<sub>max_avg</sub> [°C ]</b>
Središnja	7	22	7.74	20.71
Istočna	8	23	10.64	20.86
Gorska	5	18	7.85	16.76

```

prognozaSubjektivnaUsporedba = prognozaSubjektivna.copy()
for i, predikcija in prognozaSubjektivna.iterrows():

    # mjerenja sa postaja na dan predikcije vremenske prognoze
    mjerenja = podaciPostaja[
        podaciPostaja["Timestamp"].dt.date == predikcija["Timestamp Prognoze"]
    ]

    # lista postaja za trenutnu regiju
    postajeRegije = regijeDict[predikcija["regija"]]

    TminSuma = 0
    TmaxSuma = 0
    j = 0
    for postaja in postajeRegije:
        # podaci mjerenja samo za trenutnu postaju
        podaciPostaje = mjerenja[mjerenja["Postaja"] == postaja]

        # ako postoje mjerenja za postaju, dodaj sumama min i max temp.
        if podaciPostaje.__len__() != 0:
            TminSuma += podaciPostaje["Temperatura zraka"].min()
            TmaxSuma += podaciPostaje["Temperatura zraka"].max()
            j += 1

    # prosjecna min i max temp za regiju u danu prognoze
    TminAVG = TminSuma / j
    TmaxAVG = TmaxSuma / j

    # zapis u podatkovni okvir usporedbe
    prognozaSubjektivnaUsporedba.loc[i, "TminAVGRealna"] = TminAVG
    prognozaSubjektivnaUsporedba.loc[i, "TmaxAVGRealna"] = TmaxAVG

```

### *Isječak koda 5.1. Izračun usporedbe temperatura*

Zatim se računa prosječno apsolutno odstupanje (PAO) prognozirane maksimalne i minimalne temperature od prosječne izmjerene maksimalne i prosječne izmjerene minimalne temperature, odnosno greška predviđanja temperature. Odstupanje se računa za predikciju minimalne i maksimalne temperature tako što se sumiraju sve razlike (za svako predviđanje) između predviđene temperature i izmjerene prosječne temperature. Na primjer, sa podacima iz tablice 5.2., razlika (odstupanje) prognoze minimalne temperature u prvom retku se računa prema izrazu 5.1 te se zatim prema izrazu 5.2 sve apsolutne vrijednosti razlika zbroje i na kraju podjele sa brojem razlika kako bi se dobilo prosječno apsolutno odstupanje prognozirane minimalne temperature od izmjerenih vrijednosti. Isti postupak je primijenjen i za izračun PAO prognozirane maksimalne temperature. Sva prosječna apsolutna odstupanja pokazuju točnost subjektivne prognoze za cijelu Hrvatsku. Napravljene su dvije analize: jedna sa podacima prognoze za trenutni dan i druga sa podacima prognoze za naredni dan.

Zatim je, po istom prethodno opisanom postupku, izračunato prosječno apsolutno odstupanje (PAO) prognozirane maksimalne i minimalne temperature po regijama Hrvatske.

$$\epsilon_{min} = |T_{min} - T_{min\_avg}| \quad (5.1)$$

Gdje je:

$\epsilon_{min}$	apsolutno odstupanje prognozirane minimalne temperature od prosječne izmjerene minimalne temperature za određenu regiju u određenom danu
$T_{min}$	minimalna prognozirana temperatura za određenu regiju u određenom danu
$T_{min\_avg}$	prosječna izmjerena minimalna temperatura u određenoj regiji u određenom danu

$$PAO_{min} = \frac{\sum_{i=1}^n \epsilon_{min\_i}}{n} \quad (5.2)$$

Gdje je:

$PAO_{min}$	prosječno apsolutno odstupanje prognoze minimalne temperature za određenu regiju
$\epsilon_{min\_i}$	jedno od svih apsolutnih odstupanja prognozirane minimalne temperature od prosječne izmjerene minimalne temperature za određenu regiju, izračunato prema izrazu 5.1
n	broj svih apsolutnih odstupanja u određenoj regiji u svim vremenskim trenucima

Rezultati analize prikazani su u tablici 5.3. Intuitivno se može zaključiti da će greška predikcije minimalne i maksimalne temperature za naredni dan biti veća od greške predikcije za trenutni dan. Ali ako se promatra prosječno apsolutno odstupanje za cijelu Hrvatsku može se uočiti da je prosječna greška predikcije minimalne i maksimalne temperature za trenutni i naredni dan vrlo slična, odnosno, razlika je jako mala. Isti uzorak može se uočiti i u prosječnim greškama po regijama. Najveće greške imaju Gorska Hrvatska i Unutrašnjost Dalmacije dok Južni Jadran ima najmanje greške. Iz analize subjektivne prognoze meteorologa, može se zaključiti da je subjektivna prognoza DHMZ-a pouzdana i točna prognoza sa prihvatljivim pogreškama koje se kreću oko 1 °C.

*Tablica 5.3. Točnost prognoze meteorologa za cijelu RH i za regije*

<b>Regija</b>	<b>PAO Temperature<sub>min</sub> za trenutni dan [°C]</b>	<b>PAO Tempearture<sub>max</sub> za trenutni dan [°C]</b>	<b>PAO Temperature<sub>min</sub> za naredni dan [°C]</b>	<b>PAO Temperature<sub>max</sub> za naredni dan [°C]</b>
RH	1.18	1.46	1.27	1.46
Zagreb	1.34	1.08	1.48	1.07
Sjeverni Jadran	0.91	1.21	1.07	1.25
Istočna Hrvatska	1.37	1.62	1.49	1.71
Središnja Hrvatska	0.81	1.92	0.94	2.05
Gorska Hrvatska	1.52	1.96	1.54	1.88
Srednji Jadran	0.99	1.52	0.98	1.43
Južni Jadran	0.69	0.82	0.98	0.81
Unutrašnjost Dalmacije	1.79	1.54	1.71	1.52

## **5.2. Analiza numeričke prognoze Norveškog instituta za meteorologiju za: tlak zraka, brzinu i smjer vjetra te relativnu vlažnost zraka**

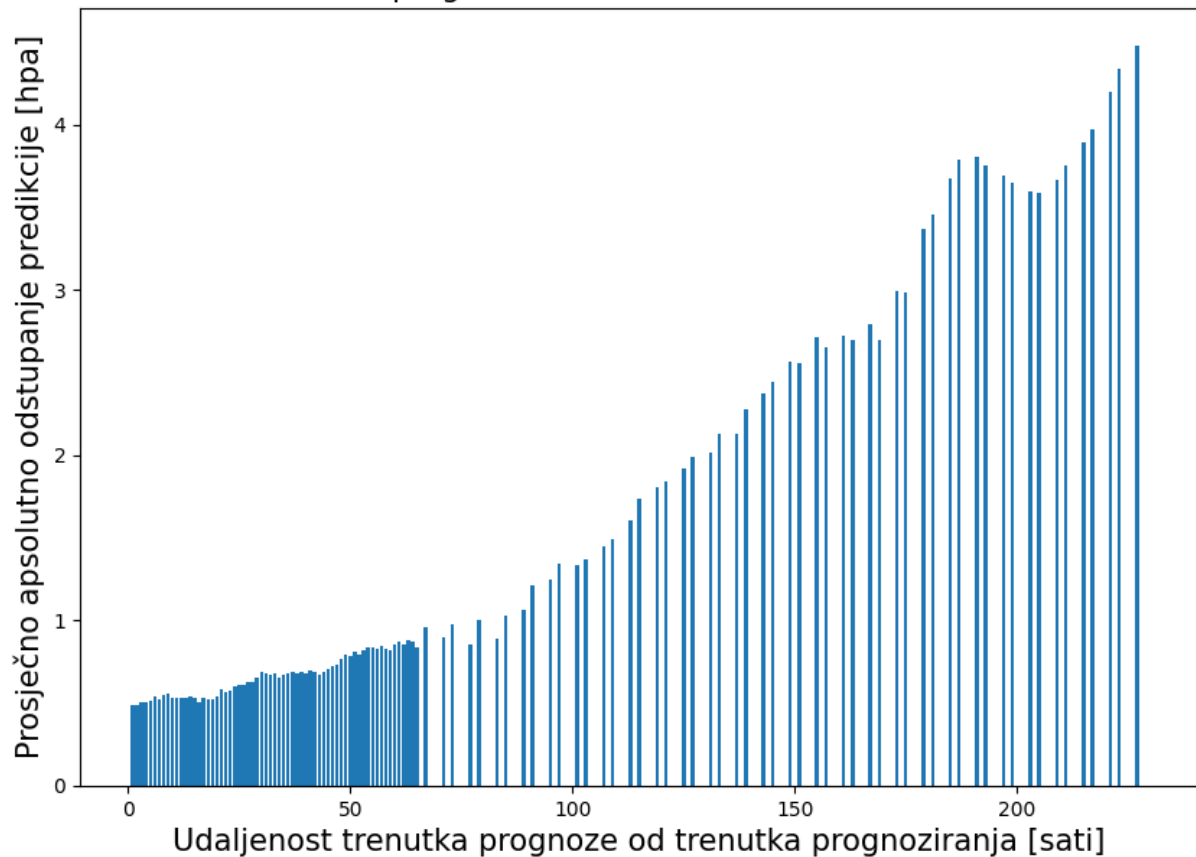
Analiza prognoze Norveškog instituta za meteorologiju, napravljena je usporedbom prognoziranih vrijednosti sa izmjerenim vrijednostima. Uspoređeni podaci se odnose na područje Maksimira u gradu Zagrebu. Točnost prognoze prikazana je kao prosječno apsolutno odstupanje prognoziranih vrijednosti od izmjerenih vrijednosti. Točnost je podijeljena u grupe koje predstavljaju udaljenost trenutka izrade prognoze od budućeg trenutka na koji se odnosi prognoza.

Programski kod učitava podatke mjerenja i podatke prognoze te izvršava transformacije podataka kako bi se mogla provesti analiza. Kod transformacije podataka, pored ostalih transformacija, iz podataka mjerenja izdvojeni su podaci mjerenja samo sa mjerne postaje u Maksimiru te su izdvojeni samo oni podaci za koje postoje meteorološka predviđanja. Nakon obrade podataka, provodi se izračun točnosti prognoze. Iterira se kroz podatke predikcije te se za svaku predikciju računa njezino apsolutno odstupanje od izmjerene vrijednosti. Računa se i udaljenost trenutka izrade prognoze od budućeg trenutka na koji se odnosi prognoza. Nakon izračuna svih odstupanja, računaju se prosječna apsolutna odstupanja po budućim vremenskim udaljenostima prognoze. Analiza točnosti predikcije smjera vjetra u programskom kodu je izračunata iteracijom kroz predikcije smjera vjetra te usporedbom svake predikcije sa izmjerenim smjerom vjetra. Smjerovi vjetra su klasificirani u 8 kategorija: sjever, sjeveroistok, istok, jugoistok, jug, jugozapad, zapad i sjeverozapad. Predikcija smjera vjetra se smatra točnom ako je izmjereni smjer vjetra jednak prognoziranom smjeru vjetra. Predikcija se smatra djelomično točnom ako je izmjereni smjer vjetra susjedan prognoziranom smjeru vjetra. Na primjer, ako je predviđen smjer vjetra bio jug, a izmjeren je jugozapadni ili jugoistočni vjetar, predikcija će se smatrati kao djelomično točna. U nastavku su prikazani rezultati analiza.

Rezultati analize prognoze tlaka prikazani su na slici 5.1. Prikazano je prosječno apsolutno odstupanje za, buduće, vremenske trenutke za koje se prognozira tlak zraka. Bitno je napomenuti da za određene buduće vremenske trenutke ne postoje vrijednosti odstupanja jer za te vremenske trenutke nisu postojala mjerenja ili je količina mjerenja bila premala da bi analiza bila reprezentativna. Očekivano, može se uočiti uzorak rasta pogreške predikcije tlaka zraka zajedno sa rastom vremenskih trenutaka prognoze.



### Točnost numeričke prognoze tlaka zraka, Norveški meteorološki institut

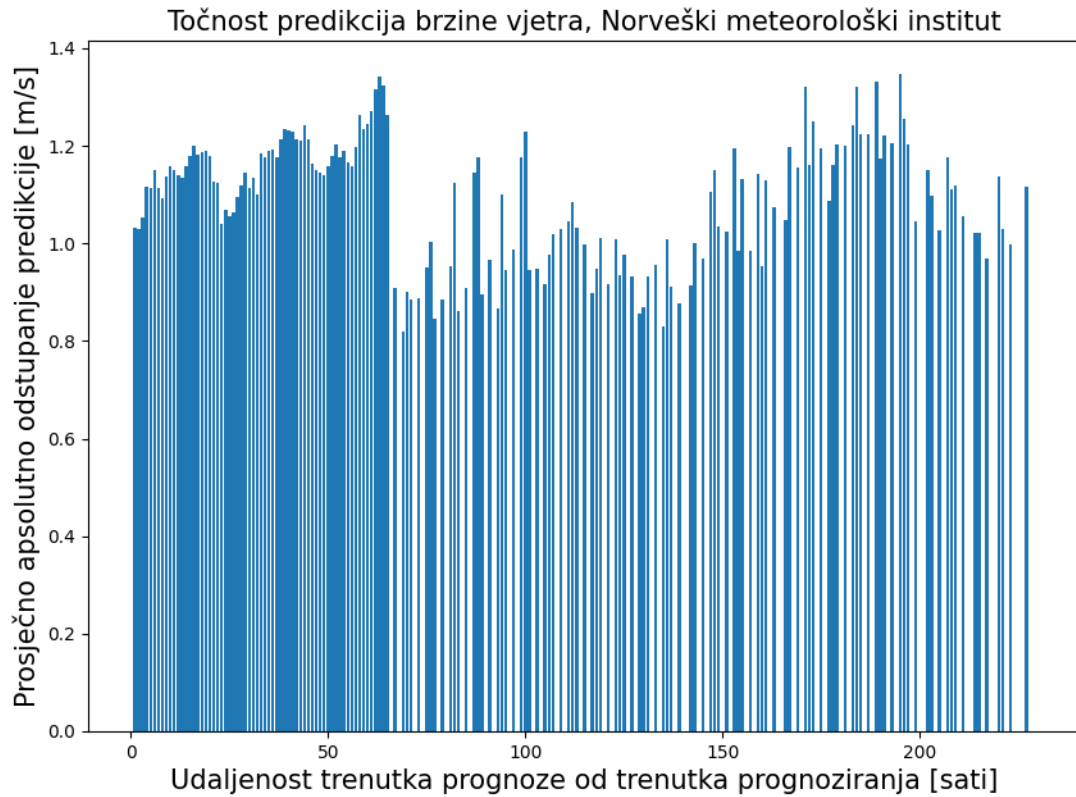


Slika 5.7. Točnost numeričke predikcije tlaka zraka za Maksimir u Zagrebu, YR

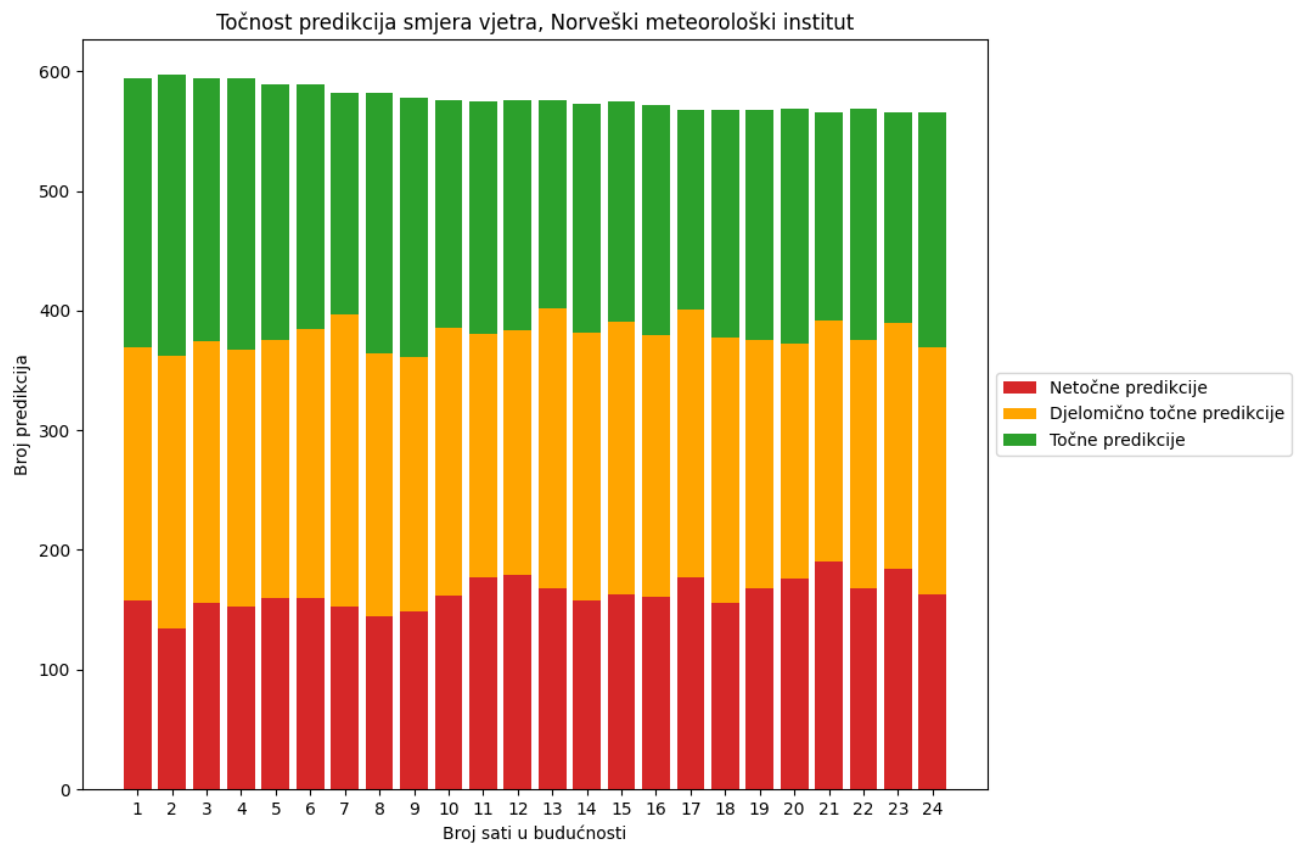
Rezultati analize predikcije brzine vjetra su prikazani na slici 5.2. Prikazano je prosječno apsolutno odstupanje za, buduće, vremenske trenutke za koje se prognozira brzina vjetra. Zanimljivo je uočiti kako nakon 70. vremenskog trenutka greška naglo pada sa 1.26 m/s na 0.90 m/s te nakon toga oscilira oko 0.9 m/s do 150. trenutka u kojemu greška opet raste. Iako je razlika između greški relativno mala, može se zaključiti da predikcija brzine vjetra ima kaotičnu prirodu ako se detaljnije promatra točnost predikcije.

Rezultati analize predikcije smjera vjetra prikazani su na slikama 5.3. i 5.4. Točnost predikcije smjera vjetra je prikazana kao postotak točnih i djelomično točnih predviđenih vrijednosti u odnosu na netočne predikcije. Iz slike 5.3. može se uočiti kako se sa vremenom broj točnih predikcija vrlo malo smanjuje, a broj djelomično točnih predikcija se vrlo malo povećava. Broj netočnih predikcija, uglavnom, ostaje isti. Iz grafikona sa slike 5.4. može se zaključiti da pouzdanost predikcije smjera

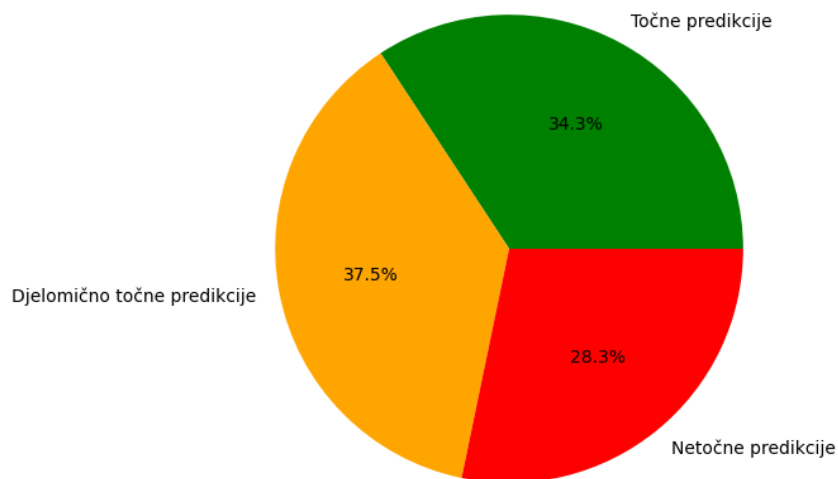
vjetra za sljedeća 24 sata nije velika. Ako se pretpostavi da su djelomično točne predikcije jednake točnim predikcijama, dobiva se podatak da je predikcija smjera vjetra za naredna 24 sata pouzdana u 72 % slučajeva.



*Slika 5.2. Točnost numeričke predikcije brzine vjetra za Maksimir u Zagrebu, YR*

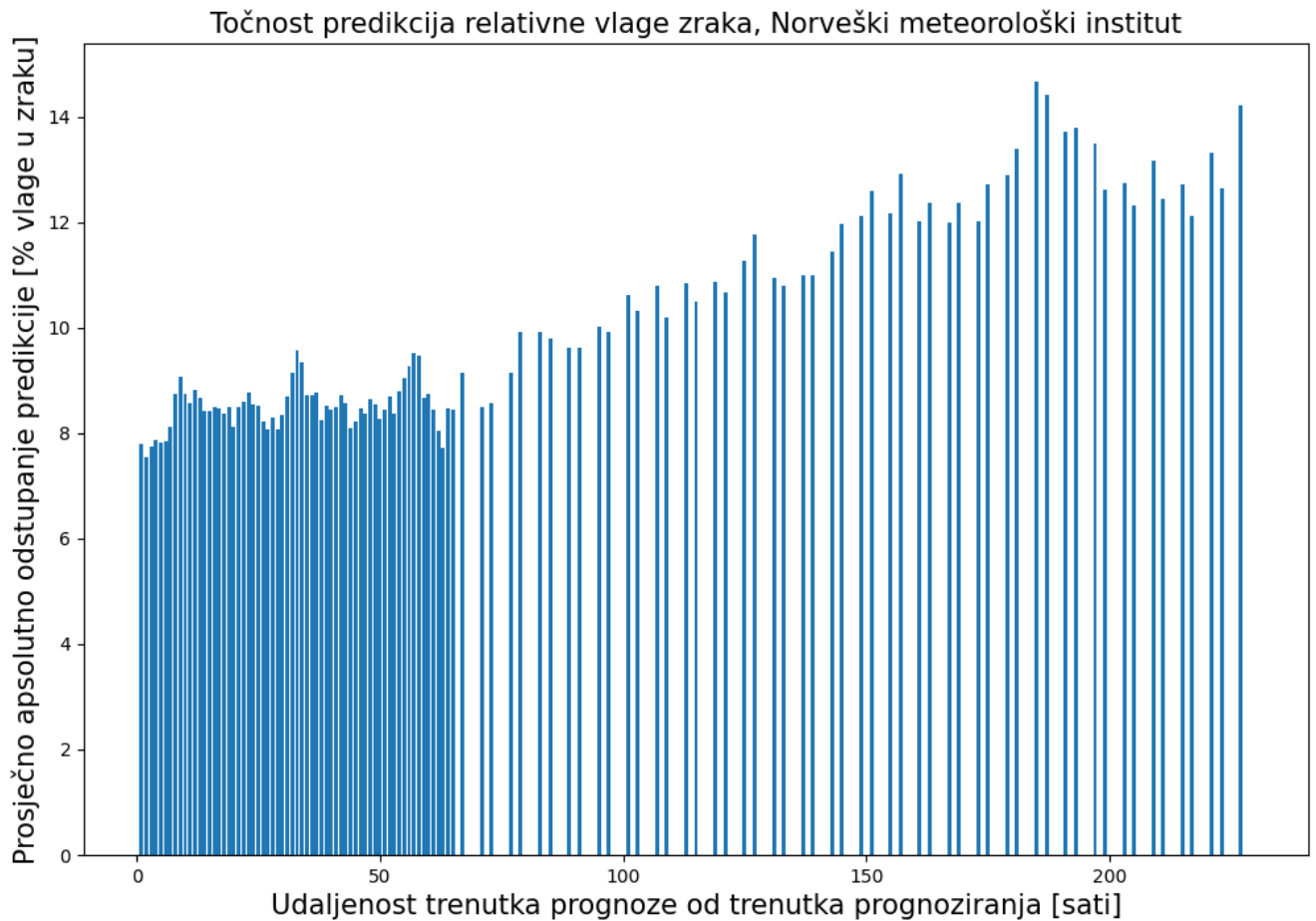


Slika 5.3. Točnost predikcije smjera vjetra za Maksimir u Zagrebu, YR



Slika 5.4. Točnost predikcije smjera vjetra za buduća 24 sata za Maksimir u Zagrebu, YR

Rezultati točnosti predikcije relativne vlage zraka prikazani su na slici 5.5. Prikazano je prosječno apsolutno odstupanje za, buduće, vremenske trenutke za koje se prognozira relativna vlažnost zraka. Očekivano, može se uočiti uzorak rasta pogreške predikcije relativne vlage zraka zajedno sa rastom vremenskih trenutaka prognoze.



Slika 5.5. Točnost numeričke predikcije vlage zraka za Maksimir u Zagrebu, YR

### 5.3. Analiza numeričke prognoze temperature DHMZ-a

U nastavku su opisane analize napravljene nad numeričkom sedmodnevnom prognozom DHMZ-a. Analizirana je točnost numeričke predikcije temperature i količine padalina. Prognoza predviđa temperaturu i količinu padalina u budućim vremenskim trenutcima koji su međusobno udaljeni za 3 sata. Na slici 4.3. prikazane su lokacije mjernih postaja za koje je napravljena analiza numeričke prognoze.

Točnost predikcije temperature prikazana je kao prosječno apsolutno odstupanje prognozirane vrijednosti od izmjerene vrijednosti. Točnosti predikcije grupirane su s obzirom na udaljenost vremenskog trenutka kada je izračunata predikcija od vremenskog trenutka na koji se odnosi predikcija. Skupovi podataka su učitani, u programski kod, iz tablica *prognoza\_7\_dana* i *podaci\_postaja* u podatkovne okvire *prognoza* i *podaciPostaja*. Podskup podataka prognoze, zajedno sa podacima mjerenja, prikazani su u tablici 5.4. Podaci prognoze se interpretiraju na način da stupac *Vremenski trenutak izrade prognoze* predstavlja vremenski trenutak kada je napravljeno prognoziranje (izrađena prognoza), a stupac *Vremenski trenutak prognoze* predstavlja vremenski trenutak na koji se odnosi prognoza (vremenski trenutak u budućnosti). Stupac *Prognozirana temperatura* predstavlja temperaturu u vremenskom trenutku prognoze, a stupac *Prognozirana količina oborina* predstavlja akumulaciju oborina između prethodnog i trenutnog vremenskog trenutka prognoze. Na primjer, prvi redak tablice ukazuje da je drugog travnja 2024. godine u 18 sati predviđeno da će temperatura zraka drugog travnja 2024. godine u 20 sati iznositi 16 °C u gradu Hvaru. Predzadnji redak ukazuje da je drugog travnja 2024. godine u 18 sati predviđeno da će se drugog travnja 2024. godine između 20. i 23. sata pasti 3.7 milimetara oborina u mjestu Ilok.

U programskom kodu, nakon učitavanja podatka napravljena je transformacija podatka. Određeni nazivi mjernih postaja, u podacima s mjerenjima, nisu bili jednaki nazivima gradova u podacima s prognozom. U svrhu povezivanja dvaju podatkovnih okvira, napravljena je struktura podataka tipa rječnik (*eng.* Dictionary) u kojoj je ime svake mjerne postaje mapirano na jedno prognozirano mjesto. Djelomični prikaz mapiranja je vidljiv u isječku koda 5.2.

```
naziviGradovaNovi = {  
    "Daruvar": "Daruvar",  
    "Dubrovnik": "Dubrovnik",  
    "Gospić": "Gospic",  
    "RC Gradište (kod Županje)": "Gradiste",
```

*Isječak koda 5.2. Dio podatka u rječniku mapiranja naziva gradova*

Nakon izrade rječnika mapiranja, imena gradova u podacima s prognozom su promijenjena u imena mjernih postaja te je iz podataka prognoze izdvojen podskup podataka za koje postoje mjerenja sa mjernih postaja. Tipovi podatka stupaca koji bilježe vremenske trenutke su pretvoreni iz tekstualnog tipa podataka u *datetime* tip podataka te se vremenski trenutak, u kojemu je izračunata prognoza, postavlja na sljedeći puni sat. Zatim se iterira kroz podatke prognoze te se uz svaku predviđenu vrijednost temperature dodaje izmjerena vrijednost temperature u određenom vremenskom trenutku na određenom geografskom mjestu. Usporedba prognoziranih temperatura i izmjerenih temperatura je prikazana u tablici 5.4. Zatim program iterira opet kroz podatke, računa apsolutna odstupanja te zbraja sva odstupanja kako bi se na kraju iteracije izračunalo prosječno apsolutno odstupanje prema izrazu 5.3. Prosječna apsolutna odstupanja grupirana su po udaljenosti između trenutka izrade prognoze i trenutka u budućnosti na koji se odnosi prognoza.

$$PAO = \frac{\sum_{i=1}^n |T_i - T_{m,i}|}{n} \quad (5.3)$$

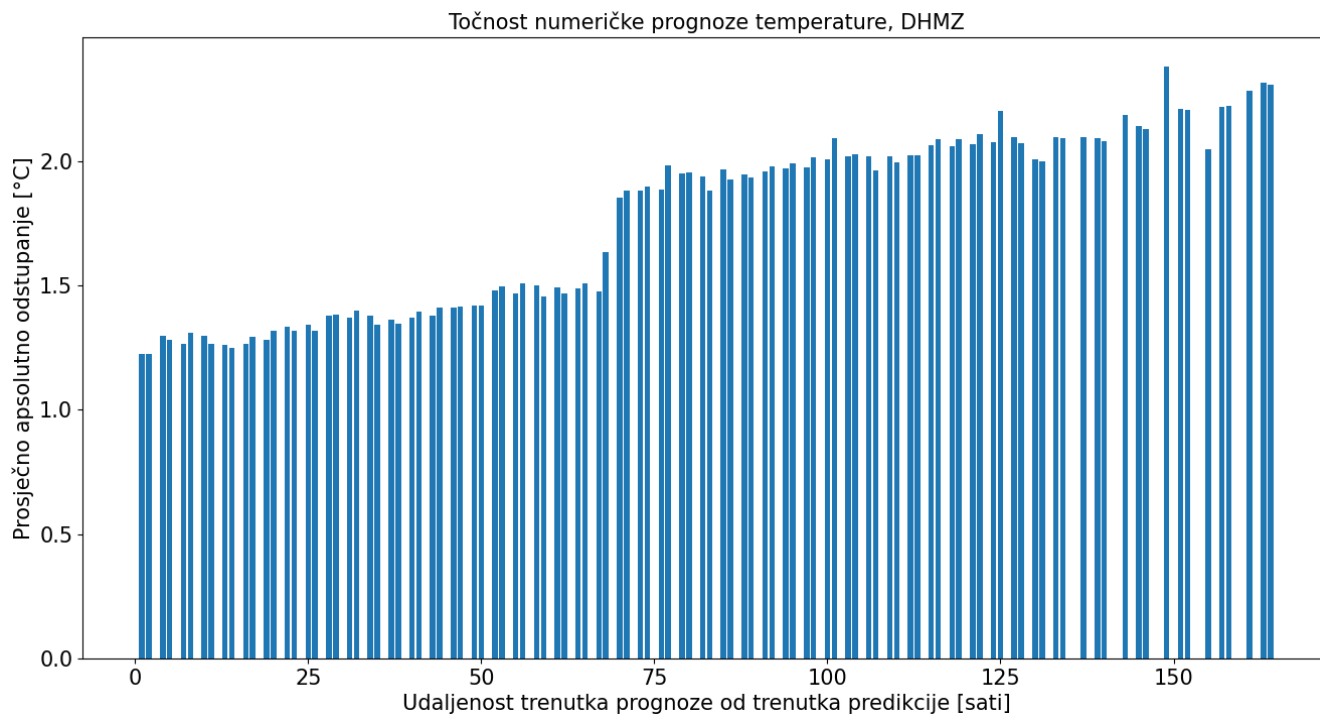
Gdje je:

$PAO$	prosječno apsolutno odstupanje prognoze temperature
$T_i$	prognozirana vrijednost temperature zraka
$T_{m,i}$	izmjerena temperatura
$n$	broj prognoziranih vrijednosti temperature

Na slici 5.6. prikazani su rezultati analize točnosti numeričke prognoze temperature DHMZ-a. Prikazano je prosječno apsolutno odstupanje za, buduće, vremenske trenutke za koje se prognozira temperatura. Bitno je napomenuti da za određene buduće vremenske trenutke ne postoje vrijednosti odstupanja jer za te vremenske trenutke nisu postojala mjerenja ili je količina mjerenja bila premala da bi analiza bila reprezentativna. Prosječno apsolutno odstupanje predikcije za 25. sat u budućnost iznosi 1.34 °C, za 50. sati iznosi 1.42 °C, a za stoti sat iznosi 2.00 °C. Iz prikaza sa slike može se uočiti trend rasta pogreške predikcije zajedno sa rastom vremenskog trenutka prognoze. Zanimljivo je uočiti nešto veći skok greške prije 75. sata vremenskog trenutka prognoze.

Tablica 5.4. Prikaz podskupa podatka iz podatkovnog okvira prognoza usporedno s mjerenjem

Grad	Prognozirana temperatura [°C]	Izmjerena temperatura [°C]	Prognozirana količina oborina [mm]	Vremenski trenutak prognoze	Vremenski trenutak izrade prognoze
Hvar	16	16.2	0.0	2024-04-02 20:00:00	2024-04-02 18:00:00
Hvar	13	13.7	0.0	2024-04-02 23:00:00	2024-04-02 18:00:00
Hvar	12	12.4	2.0	2024-04-03 02:00:00	2024-04-02 18:00:00
Ilok	9	8.5	3.0	2024-04-02 20:00:00	2024-04-02 18:00:00
Ilok	10	10.5	3.7	2024-04-02 23:00:00	2024-04-02 18:00:00
Ilok	9	8.4	0	2024-04-03 02:00:00	2024-04-02 18:00:00



Slika 5.6. Prikaz točnosti numeričke prognoze temperature DHMZ-a



#### 5.4. Analiza numeričke prognoze količine oborina DHMZ-a

Podaci mjerenja količine oborina ažuriraju se svaka 24 sata u 8 sati ujutro te su podaci dostupni samo za mjerne postaje na kojima su zabilježene oborine. Tablica 4.2. prikazuje podatke mjerenja oborina. Za analizu točnosti numeričke predikcije količine padalina korištena je metrika prosječnog apsolutnog odstupanja prognoziranih vrijednosti od izmjerenih vrijednosti padalina. Zbog bolje interpretacije točnosti prognoze dodatno je korištena metrika prosječnog apsolutnog odstupanja prognoze od mjerenja u postocima. Prosječna apsolutna odstupanja su izračunata za 6 vremenskih prozora (intervala). Svaki vremenski prozor je dugačak 24 sata i predstavlja buduće vremenske trenutke na koje se odnosi prognoza udaljene od trenutka izrade prognoze za određeno vrijeme. Na primjer, prvi vremenski prozor predstavlja prosječno apsolutno odstupanje prognoze za trenutke prognoze udaljene 1 do 24 sata u budućnost od trenutka izrade prognoze. Sljedeći vremenski prozor predstavlja prosječno apsolutno odstupanje prognoze za vremenske trenutke prognoze udaljene 24 do 48 sati od trenutka izrade prognoze. Dužina vremenskih prozora (24 sata) izabrana je zbog toga što podaci mjerenja oborina predstavljaju akumulaciju oborina unutar 24 sata. Stoga je kod izračuna točnosti, za svaki vremenski prozor, uzeta u obzir predikcija akumulacije količine oborina u vremenskom prozoru i mjerenje akumulacije količine oborina za promatrani vremenski prozor.

Programski kod analize, nakon učitavanja podataka u podatkovne okvire, radi neophodne transformacije nad podacima prije nego što se podaci analiziraju. Prva transformacija odnosi se na imena mjernih postaja i imena mjesta u vremenskoj prognozi. Za određene gradove postoji gramatička ili semantička razlika u imenu mjerne postaje i samog grada. Stoga je potrebno mapirati imena mjesta sa imenima mjernih postaja. Mapiranje je definirano ručno u rječniku naziva *naziviGradovaNovi* koji je prikazan u isječku koda 5.2., a promjena naziva mjesta (gradova) u podacima prognoze je napravljena programski upotrebom definiranog rječnika i pomoćnih funkcija. Zbog dvadeset i četiri satne periodičnosti ažuriranja mjerenja i početka mjerenja u 8 sati ujutro, u obzir se uzimaju prognoze čije je vrijeme izrade prognoze bilo u 8 sati ujutro. Odnosno, iz podataka prognoze uzimaju se samo prognoze sa vremenskim trenutkom prognoziranja u 8 sati ujutro za koje postoje mjerenja sa mjernih postaja. Nakon pripreme i transformacije podataka provodi se analiza prognoze. Iterira se kroz svaki vremenski trenutak prognoziranja te se za svaki vremenski trenutak prognoziranja iterira kroz gradove. Vremenski trenutci prognoze se dijele u vremenske prozore te se za svaki vremenski prozor računa prosječno apsolutno odstupanje i prosječno apsolutno odstupanje u postocima. Prosječna apsolutna odstupanja vremenskih prozora računaju se prema izrazima 5.4, 5.5, 5.6 i 5.7

$$\epsilon_k = |(\sum_{i=1}^n O_{ik}) - O_{m,k}| \quad (5.4)$$

Gdje je:

$O_{ik}$	prognoza akumulacije oborina u i-tom trenutku vremenskog prozora k
$n$	broj prognoza akumulacije oborina u vremenskom prozoru k
$O_{m,k}$	izmjerena količina oborina na kraju vremenskog prozora k
$\epsilon_k$	apsolutno odstupanje prognozirane količine oborina od izmjerene količine oborina u vremenskom prozoru k

$$PAO_k = \frac{\sum_{j=1}^m \epsilon_{kj}}{m} \quad (5.5)$$

Gdje je:

$PAO_k$	prosječno apsolutno odstupanje prognoze količine oborina za k-ti vremenski prozor
$\epsilon_{kj}$	jedno od svih apsolutnih odstupanja u vremenskom prozoru k
$m$	broj svih apsolutnih odstupanja u vremenskom prozoru k

$$\epsilon_{postoci_k} = \frac{|(\sum_{i=1}^n O_{ik}) - O_{m,k}|}{O_{m,k}} \quad (5.6.)$$

Gdje je:

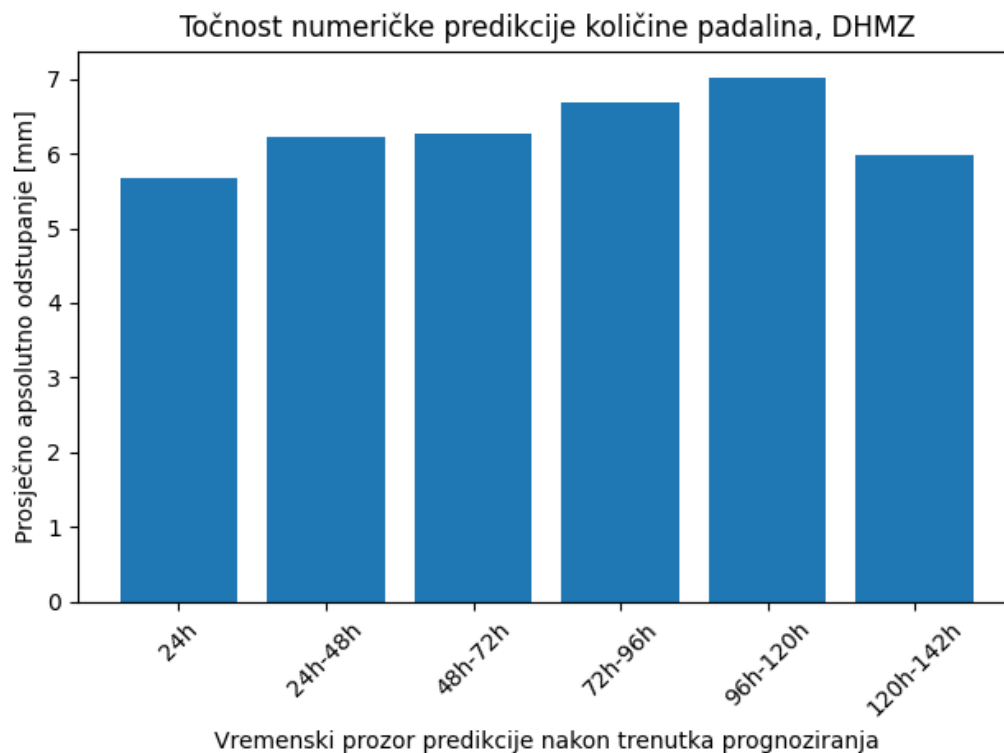
$O_{ik}$	prognoza akumulacije oborina u i-tom trenutku vremenskog prozora k
$n$	broj predikcija u vremenskom prozoru k
$O_{m,k}$	izmjerena količina oborina na kraju vremenskog prozora k
$\epsilon_{postoci_k}$	apsolutno odstupanje prognozirane količine oborina od izmjerene količine oborina u vremenskom prozoru k izraženo u postocima

$$PAO_{postoci_k} = \frac{\sum_{j=1}^m \epsilon_{postoci_{kj}}}{m} \quad (5.7)$$

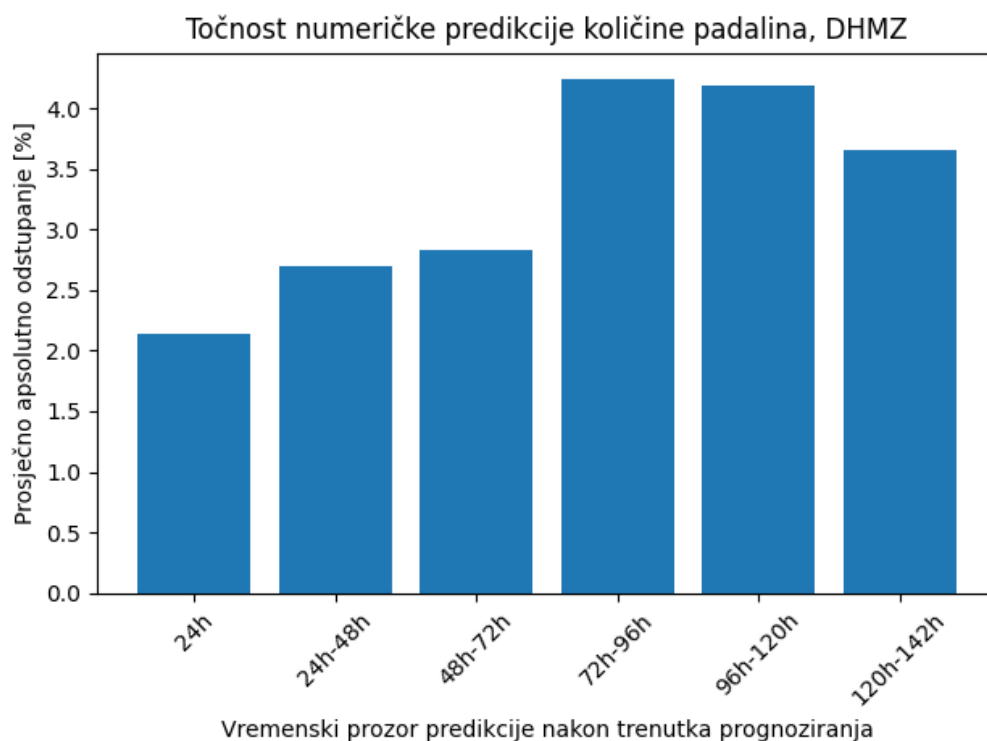
Gdje je:

$PAO_{postoci_k}$	prosječno apsolutno odstupanje prognoze količine oborina za k-ti vremenski prozor izraženo u postocima
$\epsilon_{postoci_{kj}}$	jedno od svih apsolutnih odstupanja u vremenskom prozoru k
$m$	broj svih odstupanja u vremenskom prozoru k

Na slikama 5.7. i 5.8. prikazani su rezultati analize numeričke prognoze količine padalina. Slike prikazuju prosječna apsolutna odstupanja u milimetrima i u postocima za vremenske prozore nakon trenutka izrade prognoze. Stoga, prvi stupac na slikama prikazuje točnost prognoze akumulacije količine padalina u sljedeća 24 sata. Drugi stupac predstavlja točnost predikcije akumulacije količine padalina u vremenskom prozoru koji počinje 24 sata nakon trenutka izrade prognoze i završava 48 sati nakon trenutka izrade prognoze. Iz rezultata je vidljivo da računalni model poprilično dobro prognozira količinu padalina unutar Republike Hrvatske. Prosječno apsolutno odstupanje se kreće od 5.66 mm do maksimalnih 7.02 mm. Zanimljivo je uočiti kako zadnji, najudaljeniji, vremenski prozor ima relativno malu grešku u odnosu na prethodna dva vremenska prozora. Skup analiziranih podataka u zadnjem vremenskom prozoru ima samo 7 % manje podataka od ostalih vremenskih prozora, stoga su statistički uzorci relevantni za međusobnu usporedbu. Manja greška zadnjeg vremenskog prozora dolazi izražaja na vizualizacijama, ali ako uzmemo u obzir da je razlika u redu veličine jednog milimetra, ta razlika u praksi postaje zanemariva. Zaključno, DHMZ-ova numerička predikcija količine padalina je točna i pouzdana.



*Slika 5.7. Točnost numeričke predikcije količine padalina za Hrvatsku, DHMZ*



*Slika 5.8. Točnost numeričke predikcije količine padalina za Hrvatsku, u postocima, DHMZ*

### **5.5. Analiza prognoze oborina Norveškog instituta za meteorologiju**

Analiza računa prosječno apsolutno odstupanje oborina predikcije u milimetrima i prosječno apsolutno odstupanje predikcije oborina u postocima s obzirom na izmjerene vrijednosti količine oborina. Podaci predikcije padalina su prikupljeni samo za područje Maksimira u gradu Zagrebu te su ti podaci uspoređeni sa mjerenjima padalina sa mjerne postaje u Maksimiru u Zagrebu. Analiza odstupanja je grupirana u vremenske prozore, duljine 24 sata, po udaljenosti trenutka prognoze od vremenskog trenutka predikcije (prognoziranja).

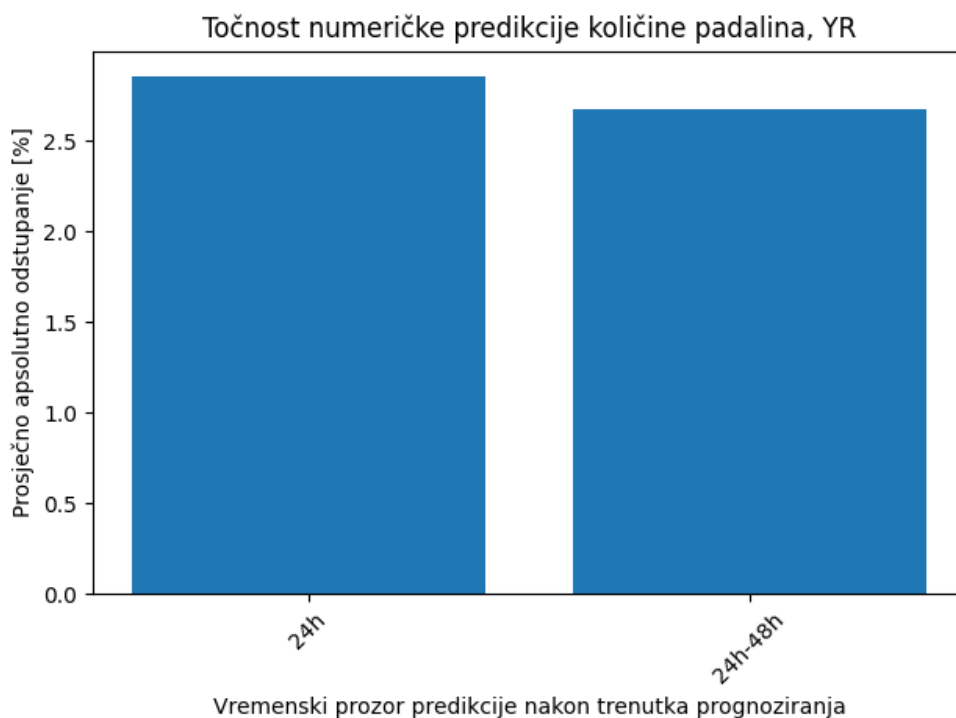
Programski kod učitava podatke mjerenja i podatke prognoze u podatkovne okvire te izvršava transformacije nad podacima. S obzirom da su mjerenja dostupna samo kao akumulacija oborina u dvadeset četiri satnom periodu, analiza uzima u obzir dvadeset četiri satne vremenske prozore nakon trenutka prognoziranja. Odnosno, za svaki trenutak prognoziranja računa se prognozirana akumulacija oborina u budućim vremenskim prozorima te se predviđanja uspoređuju sa mjerenjima. S obzirom da

su podaci predikcije dostupni samo za 48 sati nakon trenutka prognoziranja. Analiza je podijeljena na dva vremenska prozora: vremenski prozor 24 sata nakon trenutka predikcije i vremenski prozor koji počinje u dvadeset i četvrtom satu nakon trenutka predikcije i završava u četrdeset i osmom satu nakon trenutka predikcije.

Na slici 5.9. i u tablici 5.5. prikazani su rezultati analize predikcije količine oborina od strane Norveškog instituta za meteorologiju. U tablici 5.5. prikazana je usporedba točnosti predikcije količine padalina DHMZ prognoze sa YR prognozom, za Maksimir u Zagrebu. U prvom vremenskom prozoru DHMZ-ova predikcija ima prosječno apsolutno odstupanje od 4.22 mm, a YR prognoza ima 5.01 mm. U drugom vremenskom prozoru DHMZ prognoza ima PAO od 4.14 mm, dok YR ima PAO od 6.04 mm. Usporedno sa DHMZ numeričkom prognozom može se zaključiti da Norveški institut za meteorologiju lošije prognozira količinu padalina u prva dva vremenska prozora te da razlike u greškama i same greške nisu velike.

*Tablica 5.5. Usporedba točnosti predikcije količine padalina za Maksimir u Zagrebu, DHMZ usporedno sa YR*

<b>Vremenski prozor predikcije nakon trenutka prognoziranja</b>	<b>PAO DHMZ [mm]</b>	<b>PAO YR [mm]</b>
24 h	4.22	5.01
24 h – 48 h	4.14	6.04



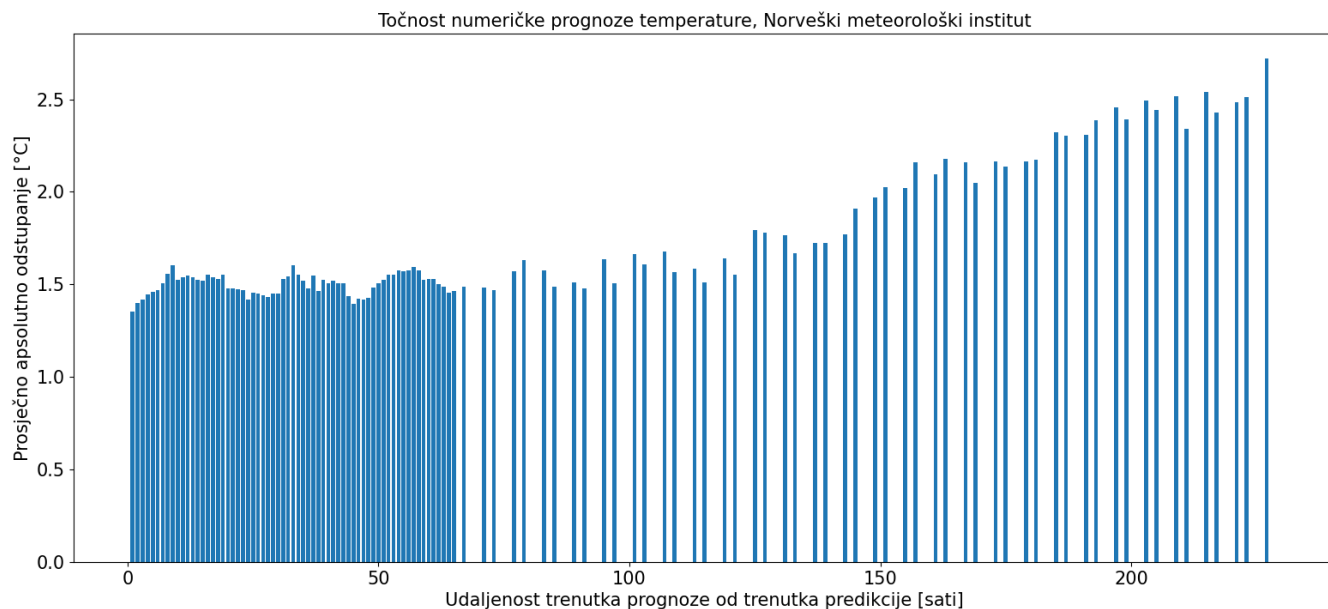
*Slika 5.9. Točnost numeričke predikcije količine padalina, u postocima za Maksimir u Zagrebu, YR*

## 5.6. Analiza prognoze temperature Norveškog instituta za meteorologiju

Analiza prognoze temperature Norveškog meteorološkog instituta napravljena je usporedbom podataka prognoze sa podacima mjerenja sa mjernih postaja. Analiza uspoređuje podatke za područje Maksimira u gradu Zagrebu. Računa se prosječno apsolutno odstupanje prognoziranih vrijednosti temperature u odnosu na izmjerenu temperaturu. Prosječna odstupanja su grupirana u grupe gdje svaka grupa predstavlja udaljenost trenutka prognoze od trenutka prognoziranja.

Programski kod analize započinje učitavanjem podatka prognoze i mjerenja te neophodnim transformacijama podataka prije same analize. Zatim programski kod iterira kroz sve predikcije temperature za koje postoje mjerenja. Te za svaku predikciju temperature pronalazi odgovarajuće mjerenje temperature u promatranom trenutku prognoze. Zatim se računa apsolutno odstupanje prognozirane vrijednosti od izmjerene vrijednosti. Odstupanja se spremaju u grupe ovisno o

udaljenosti trenutka prognoze od trenutka prognoziranja. Te se na kraju programskog koda, za svaku grupu računa prosječno apsolutno odstupanje predikcije.



*Slika 5.10. Točnost numeričke predikcije temperature za Maksimir u Zagrebu, YR*

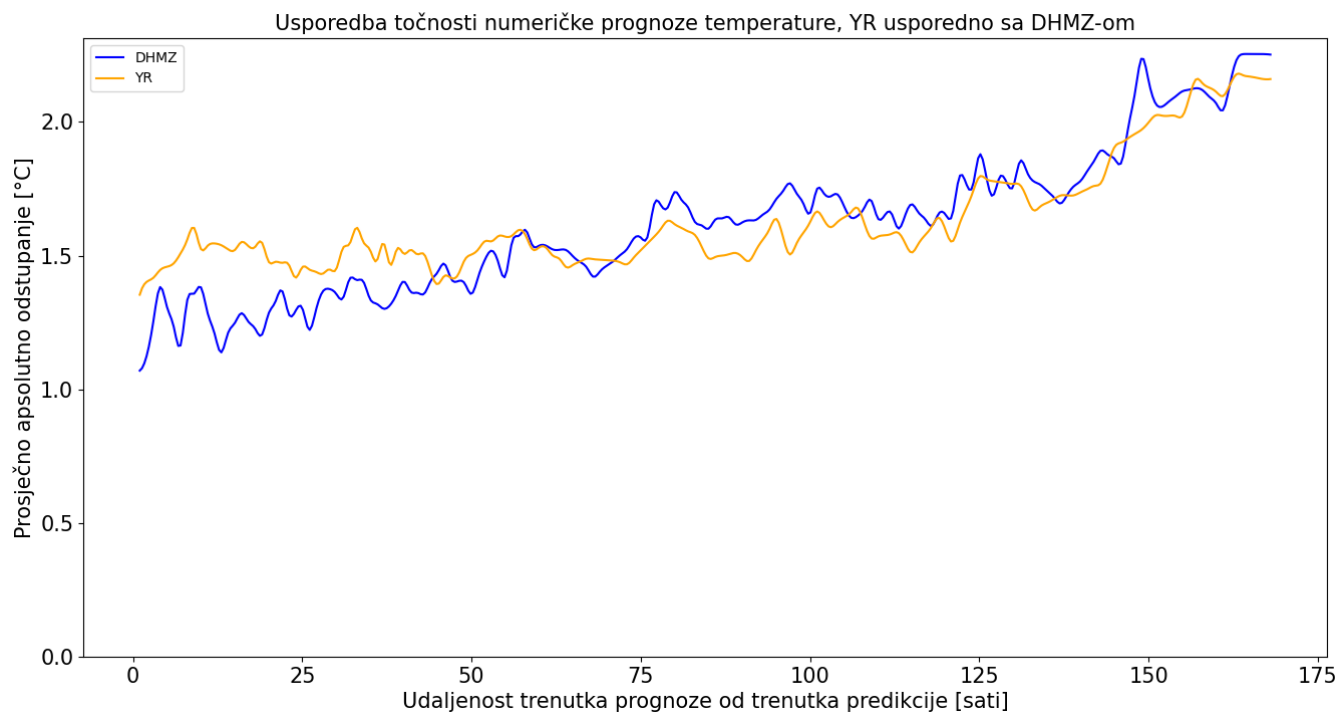
Rezultati analize numeričke prognoze temperature Norveškog instituta za meteorologiju prikazani su na slici 5.10. Prikazano je prosječno apsolutno odstupanje za, buduće, vremenske trenutke za koje se prognozira temperatura. Za određene buduće vremenske trenutke ne postoji analiza jer za te vremenske trenutke nisu postojala mjerenja ili je broj mjerenja bio premali da bi analiza bila reprezentativna. Može se uočiti očekivani rastući uzorak greške predikcije zajedno sa porastom vremenskog trenutka prognoze. Zanimljiv je podatak da prosječno apsolutno odstupanje ima vrijednost oko 1.5 °C sve do 125. vremenskog trenutka prognoze u kojem odstupanje raste i iznosi 1.79 °C. Te nakon 125. vremenskog trenutka odstupanje raste linearno.

U tablici 5.6. i na slici 5.11. uspoređena je točnost predikcije YR prognoze sa DHMZ prognozom za Maksimir u Zagrebu. Može se uočiti kako DHMZ prognoza ima veću točnost u prvih 50 budućih vremenskih trenutaka, a zatim u sljedećim budućim trenucima YR prognoza ima veću točnost.



Tablica 5.6. Usporedba točnosti predikcije temperature, DHMZ usporedno sa YR

Vremenski trenutak prognoze nakon trenutka prognoziranja [sati]	PAO DHMZ [°C]	PAO YR [°C]
25	1.30	1.46
50	1.35	1.50
73	1.53	1.47
97	1.67	1.50
121	1.65	1.55
151	2.07	2.02



Slika 5.11. Usporedba točnosti prognoze temperature, za Maksimir u Zagrebu, između DHMZ-a i YR-a

DHMZ numerička prognoza količine oborina pokazala se točnijom od numeričke prognoze količine oborina Norveškog instituta za meteorologiju. Također DHMZ numerička prognoza nudi predikciju oborina do 162. sata u budućnosti dok YR prognoza nudi predikciju do 65. sata u budućnosti. Iako se DHMZ prognoza, usporedno sa YR prognozom, pokazala malo točnijom. Razlika u točnosti dviju prognoza je praktički zanemariva te su se obje prognoze pokazale kao pouzdane. Bitno je uzeti u obzir da je analizirani skup podataka obuhvatio samo 2 mjeseca prognoza i mjerenja te da se je promatralo samo jedno geografsko mjesto prognoze, Maksimir u Zagrebu. Potencijalna buduća analiza trebala bi uključivati analizu sa većim vremenskim periodom i sa više geografskih lokacija.

## 6. PREDIKCIJA TEMPERATURE POMOĆU STROJNOG UČENJA

Strojno učenje je područje nauke umjetne inteligencije koje omogućuje računalima da uče iz podataka i rezultiraju predviđanjima ili odlukama bez eksplicitnog programiranja. Strojno učenje uključuje stvaranje algoritama i modela koji mogu automatski naučiti uzorke i odnose iz podataka te zatim koristiti naučeno znanje za predviđanje ili klasificiranje podataka. Jedna od ključnih prednosti strojnog učenja leži u sposobnosti učenja iz velikih i složenih skupova podataka koje bi ljudima bilo teško ili nemoguće ručno analizirati. Upotrebom različitih algoritama i tehnika, strojno učenje može otkriti skrivene uzorke, trendove i asocijacije unutar podataka, što se zatim može iskoristiti za rješavanje širokog spektra problema iz stvarnog svijeta u različitim domenama ljudskih djelatnosti. Algoritmi strojnog učenja mogu se općenito kategorizirati u tri glavne vrste: nadzirano učenje (*eng. supervised learning*), nenadzirano učenje (*eng. unsupervised learning*) i strojno podržano učenje (*eng. reinforcement learning*).

Nadzirano učenje bavi se treniranjem modela na označenom skupu podataka, gdje je svaki ulazni podatak povezan sa izlaznom vrijednošću. Cilj je naučiti preslikavanje ulaznih podataka u izlaznu varijablu, omogućujući modelu da predviđa nove, dosad nepoznate izlazne podatke. Uobičajene primjene nadziranog učenja uključuju regresiju (predviđanje kontinuiranih vrijednosti) i klasifikaciju (predviđanje diskretnih oznaka). Učenje bez nadzora bavi se treniranjem modela sa neoznačenim skupom podataka, gdje je cilj otkriti uzorke ili strukture unutar skupa podataka. Glavne primjene nenadziranog strojnog učenja uključuju grupiranje, smanjenje dimenzionalnosti i procjenu gustoće u statistici. Algoritmi grupiranja koriste se u nenadziranom učenju za grupiranje sličnih podataka zajedno u grupe, na temelju karakteristika podataka. Strojno podržano učenje je vrsta učenja u kojoj agent uči donositi odluke u interakcijom s okolinom i primanjem povratnih informacija u obliku nagrada ili kazni. Agent uči kroz pokušaje i pogreške, prilagođavajući svoje postupke na temelju povratnih informacija koje prima, s krajnjim ciljem maksimiziranja kumulativnih nagrada tijekom vremena. Podržano učenje pokazalo je uspjeh u zadacima kao što su igranje igrice, robotika i upravljanje autonomnim vozilom.

Strojno učenje pokreće inovacije i napredak u brojnim industrijama i područjima. U zdravstvu se koristi za poboljšanje dijagnoze bolesti i otkrivanje novih lijekova. U financijama se primjenjuje za otkrivanje prijevara, procjenu rizika i algoritamsko trgovanje. U marketingu, pomaže poduzećima da analiziraju ponašanje kupaca, optimiziraju reklamne kampanje i personaliziraju korisnička iskustva. Strojno učenje je područje koje je u naglom razvoju te sa sobom donosi nove inženjerske probleme.

Na primjer, pristranost podataka, količina podataka i točnost podataka mogu značajno utjecati na izvedbu i pouzdanost modela. Interpretabilnost modela je čest izazov te se u većini slučajeva logika izvršavanja kompleksnih modela ne može interpretirati. Dostupnost resursa, odnosno grafička procesorska snaga predstavlja veliki izazov jer treniranje i izvršavanje modela zahtijevaju veliku količinu resursa. Kako se strojno učenje nastavlja razvijati, znanstvenici i inženjeri istražuju nove algoritme i tehnike za rješavanje prisutnih izazova i problema te se trude postići još veći potencijal strojnog učenja [12].

Sposobnost strojnog učenja u pronalasku različitih uzoraka u velikim količinama podataka omogućuje istraživanje i razvoj novih tehnika za izradu vremenskih prognoza. Područje primjene strojnog učenja u predikciji vremena raste vrlo brzo te će u narednim godinama biti sve relevantnije. Zbog relevantnosti strojnog učenja u području izrade vremenskih prognoza, u sljedećim poglavljima opisana su dva modela strojnog učenja te njihova primjena na povijesnim vremenskim podacima temperature zraka. Cilj modela je predikcija temperature zraka s obzirom na izmjerene povijesne vrijednosti.

## 6.1. Povijesni podaci temperature zraka

Povijesni podaci mjerenja temperature zraka preuzeti su sa web stranice Američkog nacionalnog centra za informacije o okolišu (NOAA). Podaci sadrže povijesne podatke temperature zraka sa mjernih stanica DHMZ-a u Zagrebu. Raspon podataka je od 1.1.2000. godine do 5.4.2024. godine te su mjerenja bilježena svakih sat vremena. Za izradu modela strojnog učenja, koristiti će se povijesni podaci temperature sa mjerne postaje u Maksimiru u Zagreb.

The screenshot shows the NOAA Climate Data Online website. At the top, there is the NOAA logo and the text "NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION" and "NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION". Below this is a navigation bar with links for Home, Climate Information, Data Access, Contact, and About. A search bar is located on the right side of the navigation bar. Below the navigation bar, there is a breadcrumb trail: Home > Climate Data Online > Datasets. A secondary navigation bar contains icons for Datasets, Search Tool, Mapping Tool, Data Tools, and Help. The main content area is titled "Climate Data Online: Dataset Discovery". Below the title, there is a paragraph explaining that clicking on dataset names expands to show more details, including descriptions, links to tools, FTP access, and downloadable samples. A "Help" sidebar is visible on the right, containing links for "Climate Data Online help", "Climate Data Online Web Services", and "NCDC Web Services". Below the introductory text, there is a section titled "Climate Data Online" with a sub-heading "The datasets listed in this section are accessible within the Climate Data Online search interface." followed by a list of dataset categories with expandable icons:

- ☐ Daily Summaries
- ☐ Global Marine Data
- ☐ Global Summary of the Month
- ☐ Global Summary of the Year
- ☐ Local Climatological Data
- ☐ Normals Annual/Seasonal
- ☐ Normals Daily
- ☐ Normals Hourly
- ☐ Normals Monthly
- ☐ Precipitation 15 Minute
- ☐ Precipitation Hourly
- ☐ Weather Radar (Level II)
- ☐ Weather Radar (Level III)

*Slika 6.1. Prikaz početne internet stranice za pronalazak NOAA skupova podataka*

## 6.2. Predikcija temperature zraka gradient boosting algoritmom

U nastavku je opisan gradient boosting algoritam i njegova primjena na povijesnim podacima temperature zraka.

### 6.2.1. Gradient boosting algoritam

Gradient boosting je tehnika strojnog učenja koja je stekla široku popularnost zbog svoje kvalitetne primjene u raznim zadacima predviđanja podataka. Pripada grupi modela koji koriste ansamble (skupine) stabla odlučivanja kao glavne arhitekture modela. Arhitektura kombinira predviđanja više pojedinačnih modela (stabala) kako bi se izračunalo točnije konačno predviđanje. Algoritam trenira model uzastopnim dodavanjem novih modela (stabala) za ispravljanje pogrešaka koje su napravili prethodni modeli. Iterativni proces, dodavanja novih modela, ima za cilj minimizirati funkciju gubitka (*eng.* loss function), koja mjeri odstupanje (*eng.* residual) između predviđenih vrijednosti i stvarnih ciljanih vrijednosti. Najpopularnija implementacija pojačavanja nagiba je predstavljena u radu „Gradient Boosting Machine (GBM)“, koji je napisao Jerome H. Friedman 1999. godine. Od tada je razvijeno nekoliko implementacija algoritma. XGBoost, LightGBM i CatBoost su neke od najčešće korištenih implementacija.

Ključni principi u algoritmu pojačavanja nagiba su: stabla odluke, spuštanje nagiba, stopa učenja i regularizacija. Algoritam koristi stabla odlučivanja kao modele predikcije. Izlaz (predikcija) iz prvog modela je uglavnom prosječna vrijednost oznaka, ciljane vrijednosti predikcije u skupu podataka za treniranje. Zatim svaki sljedeći model nastoji smanjiti grešku (odstupanje) koje je napravio prethodni model. Zadnja izlazna vrijednost predikcije se dobije tako što se sumiraju predikcije svih modela za određeni ulaz. Spuštanje nagiba odnosi se na proces minimiziranja funkcije gubitka. U svakoj iteraciji (procesu dodavanja novih stabala) izračunava se nagib funkcije gubitka u odnosu na predviđene vrijednosti. Novi model (stablo) tada se konstruira da predvidi negativni gradijent, koji će ispraviti pogreške koje su napravili prethodni modeli. Kako bi se spriječilo prekomjerno prilagođavanje (*eng.* overfitting) i poboljšala generalizacija, algoritam koristi tehniku stope učenja. Nakon svake iteracije, predviđanja novog modela skaliraju se malim faktorom prije nego što se dodaju sumi predikcija. Tako se usporava proces učenja i sprječava se prekomjerno prilagođavanje. Pojačavanje nagiba pokazalo je uspjeh u širokom rasponu primjena, uključujući zadatke regresije, klasifikacije i rangiranja [13-15].

## 6.2.2. Priprema podataka i izrada modela

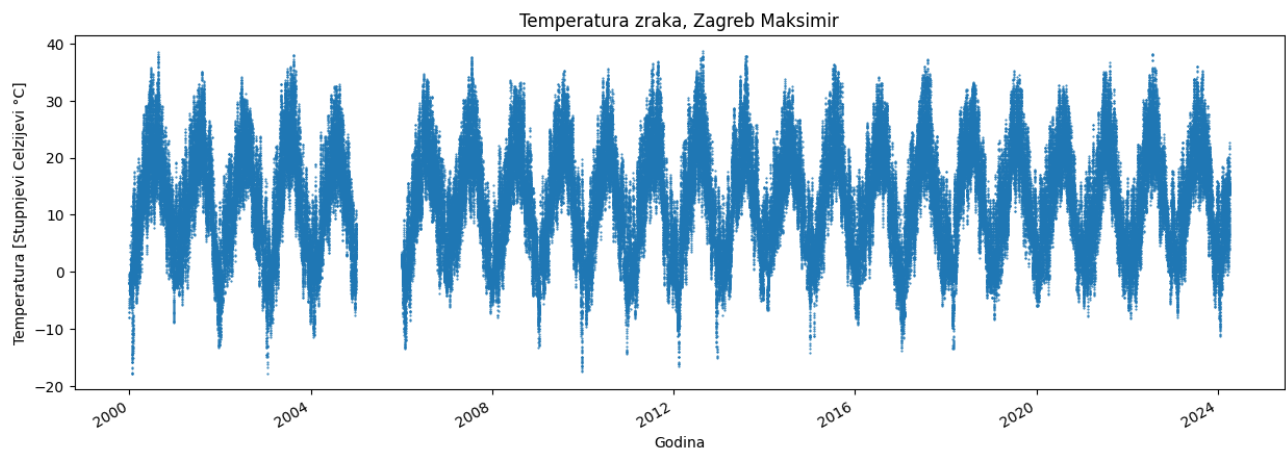
Povijesni podaci temperature se učitavaju u memoriju te se radi analiza i pregled skupa podataka. Skup sadrži podatke za više mjernih postaja, vrijednosti stupca *STATION* su: *'PUNTIJARKA, HR'*, *'ZAGREB, HR'*, *'ZAGREBMAKSIMIR, HR'*, *'ZAGREB GRIC, HR'*. Uz podatke o temperaturi i vremenskom trenutku mjerenja, skup sadrži podatke o geografskoj lokaciji mjerne postaje, nadmorskoj visini mjerne postaje, izvoru podataka i različitim kontrolnim podacima. Zatim se odabiru podaci mjerenja temperature (zajedno sa vremenskim trenutcima) samo za mjernu postaju *ZAGREBMAKSIMIR, HR* te se stupcu temperature dodaje novo ime. Daljnjom analizom, uočeno je da svakog dana ne postoje zapisi u 01, 02, 22 i 23 sata. Zatim, uočeno je da postoji 38 zapisa koji imaju vrijednost temperature 999.9, uvidom u dokumentaciju ti podaci predstavljaju nedostupna mjerenja. Također zapisi temperature za 2005. godinu sadržavaju neispravne vrijednosti temperature. Temperatura ispravnih vrijednosti je zapisana u nizu znakova u formatu: [predznak][temperatura pomnožena sa 10][kod kvalitete]. Stoga temperatura od -6.7 °C je zapisana na sljedeći način -0067,1. Za pretvorbu temperature u decimalni format zapisa, izrađena je funkcija transformacije koja je izvršena na svakom zapisu stupca temperature. Funkcija je prikaza u isječku koda 6.1.

```
def formatTemp(val):
    predznak = val[0]
    if predznak == "-":
        return float(val[1:-2]) / -10
    else:
        return float(val[1:-2]) / 10

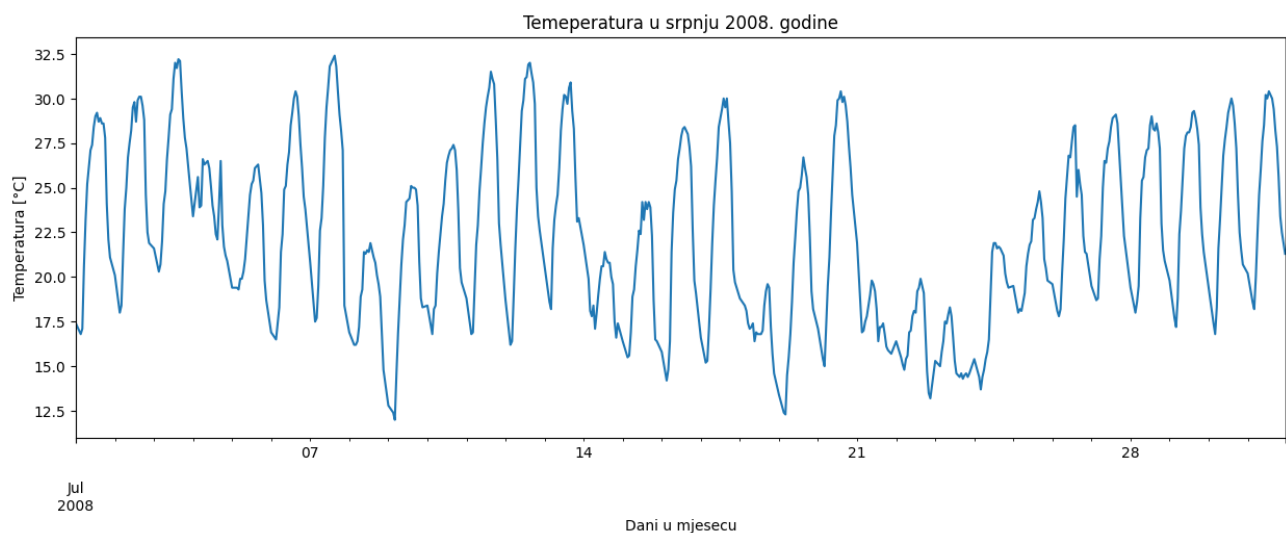
podaci["Temperatura"] = podaci["Temperatura"].apply(formatTemp)
```

### *Isječak koda 6.1. Transformacija stupca temperature*

Podatkovni okvir podatka je ponovno indeksiran na način da indeks bude vremenski trenutak mjerenja. Za nedostupne vremenske trenutke u satima 01, 02, 22 i 23 su napravljeni novi indeksi vremenskih trenutaka sa vrijednostima temperature postavljenima na *NaN*. Zapisi temperature sa vrijednostima 999.9, također se postavljaju u *NaN*, neispravni podaci za 2005. godinu se brišu. Odabrano je da se na kraju sve *NaN* vrijednosti imputiraju linearnom interpolacijom između prethodnog i budućeg vremenskog trenutka. Konačni uređeni skup povijesnih podataka prikazan je na slikama 6.2. i 6.3.



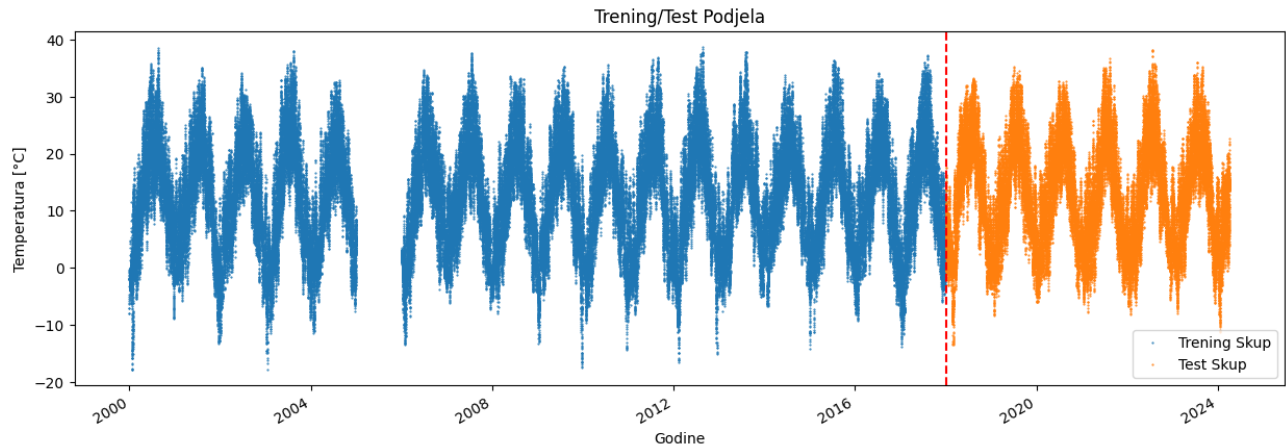
*Slika 6.2. Povijesni podaci temperature u Maksimiru u Zagrebu*



*Slika 6.3. Povijesni podaci temperature u srpnju 2008. godine za Maksimir u Zagrebu, interpolirani prikaz*

Nakon čišćenja podataka, podatke je potrebno podijeliti na podatke za treniranje i na podatke za testiranje modela. Skup podataka za treniranje sadrži 75 % podatka, od 1.1.2000. do 1.1.2018. godine, a skup podatka za testiranje sadrži ostalih 25 % podatka od 1.1.2018. do 5.4.2024. godine. Raspodjela podataka za treniranje i podataka za testiranje prikazana je na slici 6.4.





Slika 6.4. Raspodjela podatka za treniranje i podataka za testiranje

Za bolju predikciju vrijednosti potrebno je izraditi dodatne značajke za svaki zapis. Trenutna značajka za svaki zapis je vremenski trenutak mjerenja, a ciljana vrijednost je temperatura. Stoga za svaki zapis se dodaju nove značajke koje opisuju vremenski trenutak mjerenja. Nove značajke su: sat u danu, mjesec u godini, godina, dan u godini i tjedan u godini. Nove značajke su zapisane u novim stupcima.

```
def znacajke (podaci) :
    podaci = podaci.copy()
    podaci["Sat"] = podaci.index.hour
    podaci["Mjesec"] = podaci.index.month
    podaci["Godina"] = podaci.index.year
    podaci["DanUGodini"] = podaci.index.day_of_year
    podaci["TjedanUGodini"] = podaci.index.isocalendar().week
    return podaci

podaci = znacajke (podaci)
```

Isječak koda 6.2. Izrada značajki s obzirom na vremenski trenutak mjerenja

Nakon izrade značajki, definiraju se podskupovi podatka koji predstavljaju ulazne i izlazne podatke podataka za treniranje i podataka za testiranje te se definira objekt modela koji je tipa *XGBRegressor*. Model kod instanciranja prima različite parametre te su u nastavku opisani osnovni korišteni parametri *XGBRegressor* modela.

```

reg = xgb.XGBRegressor(base_score=trening["Temperatura"].mean(),
                       booster="gbtree",
                       n_estimators=2000,
                       early_stopping_rounds=150,
                       objective='reg:squarederror',
                       max_depth=3,
                       learning_rate=0.1,
                       eval_metric="mae")

reg.fit(x_trening, y_trening,
        eval_set=[(x_trening, y_trening), (x_test, y_test)],
        verbose=50)

```

### Isječak koda 6.3. Instanciranje i treniranje XGBoost modela

Parametar *base\_score* sadrži vrijednost početne predikcije prvog stabla koja je postavljena na srednju vrijednost skupa podataka. Parametar *booster*="gbtree" definira stablo kao tip modela predikcije, *n\_estimators* označava broj iteracija algoritma, odnosno, broj stabala koji će se kreirati. Rano zaustavljanje je postavljeno na 50 iteracija, odnosno, ako se funkcija gubitka na podacima za testiranje ne smanji nakon 50 iteracija, treniranje modela će se zaustaviti kako bi se spriječilo prekomjerno prilagođavanje i uštedili resursi. Parametar *objective*='reg:squarederror' definira vrstu modela i tip funkcije gubitka. U ovom slučaju izabrana je vrsta modela za regresiju i funkcija gubitka koja će računati gubitak tako što će sumirati sva kvadrirana odstupanja te ih podijeliti sa brojem odstupanja, odnosno, računati će prosječno kvadratno odstupanje. Parametar *max\_depth*=3 definira maksimalnu dubinu stabla, *learning\_rate*=0.1 definira stopu učenja, odnosno, koeficijent sa kojim će se množiti svaka nova predikcija. Na kraju, se definira metrika za evaluaciju, izabrana je funkcija koja računa prosječno apsolutno odstupanje. Parametri modela mogu značajno utjecati na performanse modela; Stoga je potrebno pažljivo odabrati vrijednosti parametara modela strojnog učenja kako bi se postigle što bolje performanse. U slučaju kada model ima puno parametara, ručno pretraživanje performansi svih mogućih parametara je otežano ili nemoguće. Zbog toga je razvijeno nekoliko tehnika pretraživanja prostora parametara modela: pretraga mreže (eng. grid search), nasumična pretraga (eng. random search), bayesova optimizacija (eng. Bayesian optimization) [16-18]. U ovom slučaju korištena je pretraga mreže kako bi se postavili prethodno definirani parametri. Prostor pretrage parametara je prikazan u isječku koda 6.4. Pretraga mreže će napraviti treniranje modela za sve definirane parametre (prostor pretrage) te će pronaći najbolju kombinaciju parametara.

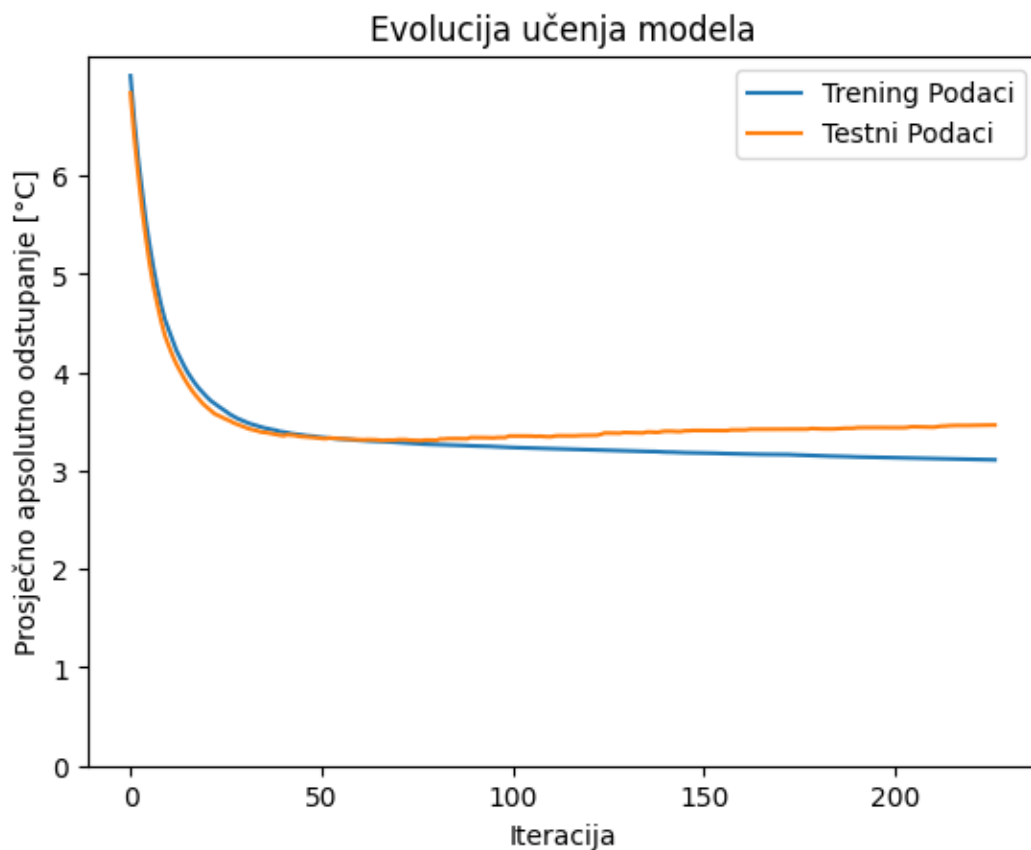
```

parametri = {
    'max_depth': [2, 3, 4],
    'learning_rate': [0.1, 0.01, 0.001, 0.0001],
    'n_estimators': [1000, 2000, 3000, 4000, 5000],
    'early_stopping_rounds': [150]
}

```

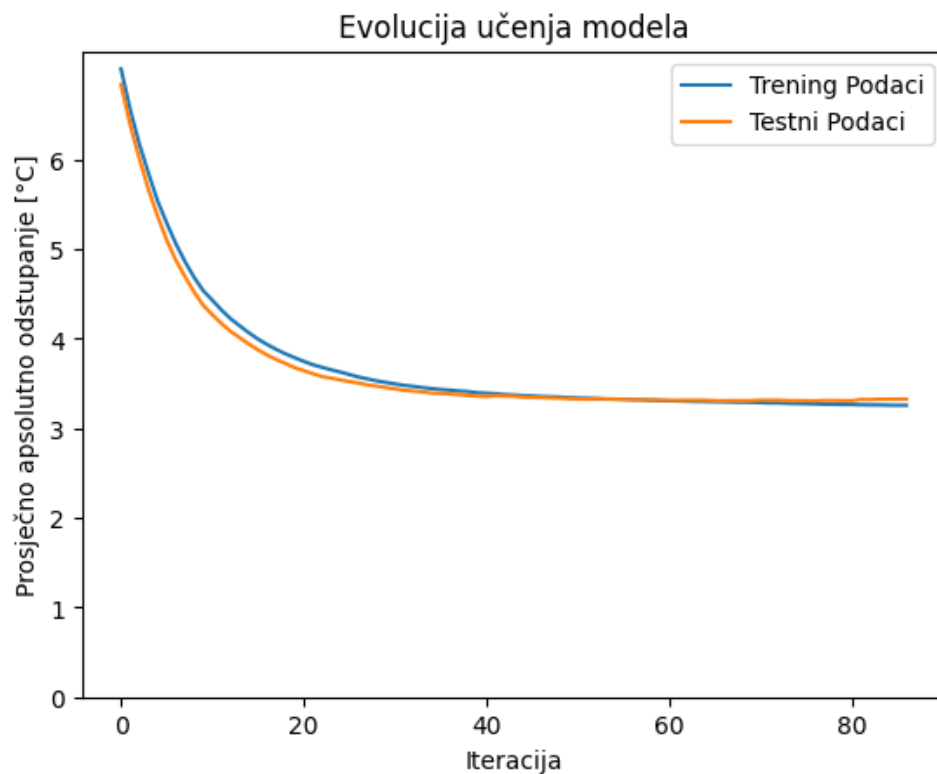
*Isječak koda 6.4. Prostor pretrage parametara modela*

Model se trenira na podacima za treniranje i nakon svake iteracije se ispituje na podacima za treniranje i na podacima za testiranje. Treniranje modela se zaustavlja nakon 724. iteracije zbog parametra ranog zaustavljanja koji je spriječio prekomjerno prilagođavanje podataka trening podacima. Prosječno apsolutno odstupanje iznosi  $3.46361^{\circ}\text{C}$ , a evolucija treniranja modela je prikazana na slici 6.5.

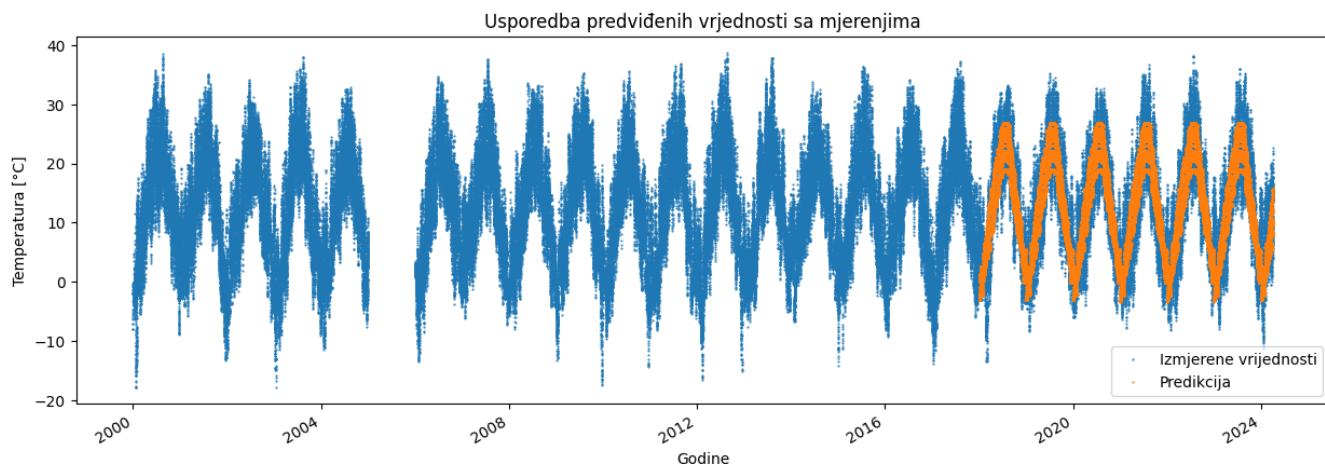


*Slika 6.5. Evolucija učenja modela s prekomjernim prilagođavanjem*

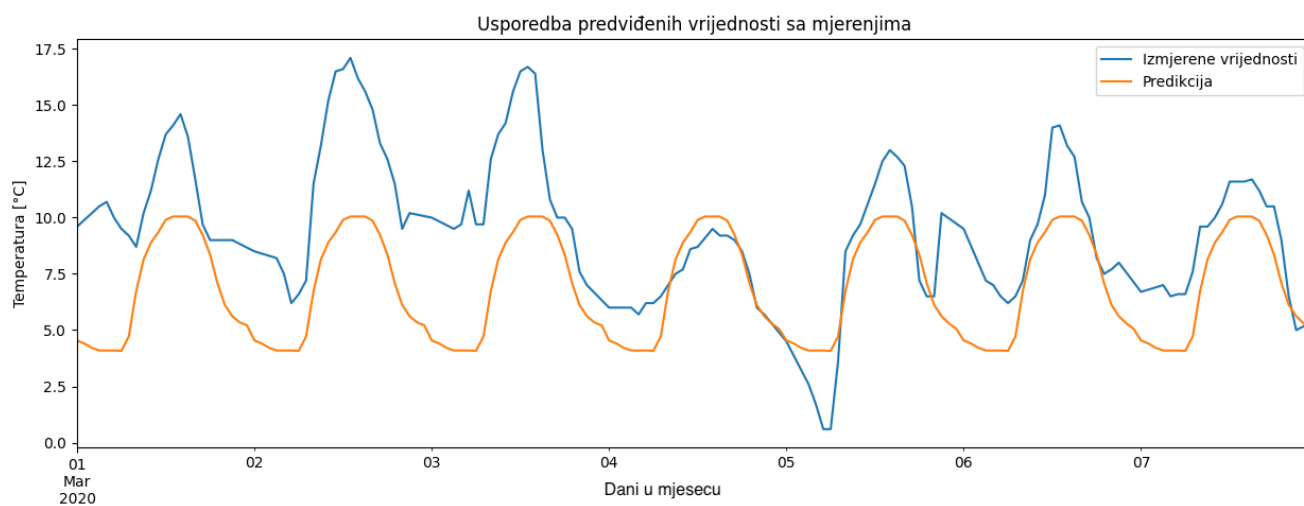
Sa slike 6.5. je vidljivo da se model sa trenutnim parametrima prekomjerno prilagođava podacima za treniranje jer se prosječno apsolutno odstupanje na podacima za treniranje s vremenom smanjuje, a prosječno apsolutno odstupanje na podacima za testiranje se s vremenom povećava (nakon 80-te iteracije). Stoga se model može dodatno optimizirati na način da se treniranje modela završi prije nego što prekomjerno prilagođavanje započne, smanjivanjem parametra ranog zaustavljanja na 10 iteracija. Optimizirani model ima prosječno apsolutno odstupanje od 3.32633 °C, a evolucija učenja je prikazana na slici 6.6. Na slici 6.7. prikazana je usporedba predikcije modela sa podacima mjerenja kroz godine, a na slici 6.8. prikazana je usporedba predikcije s podacima mjerenja za ožujak 2020. godine.



Slika 6.6. Evolucija učenja modela bez prekomjernog prilagođavanja

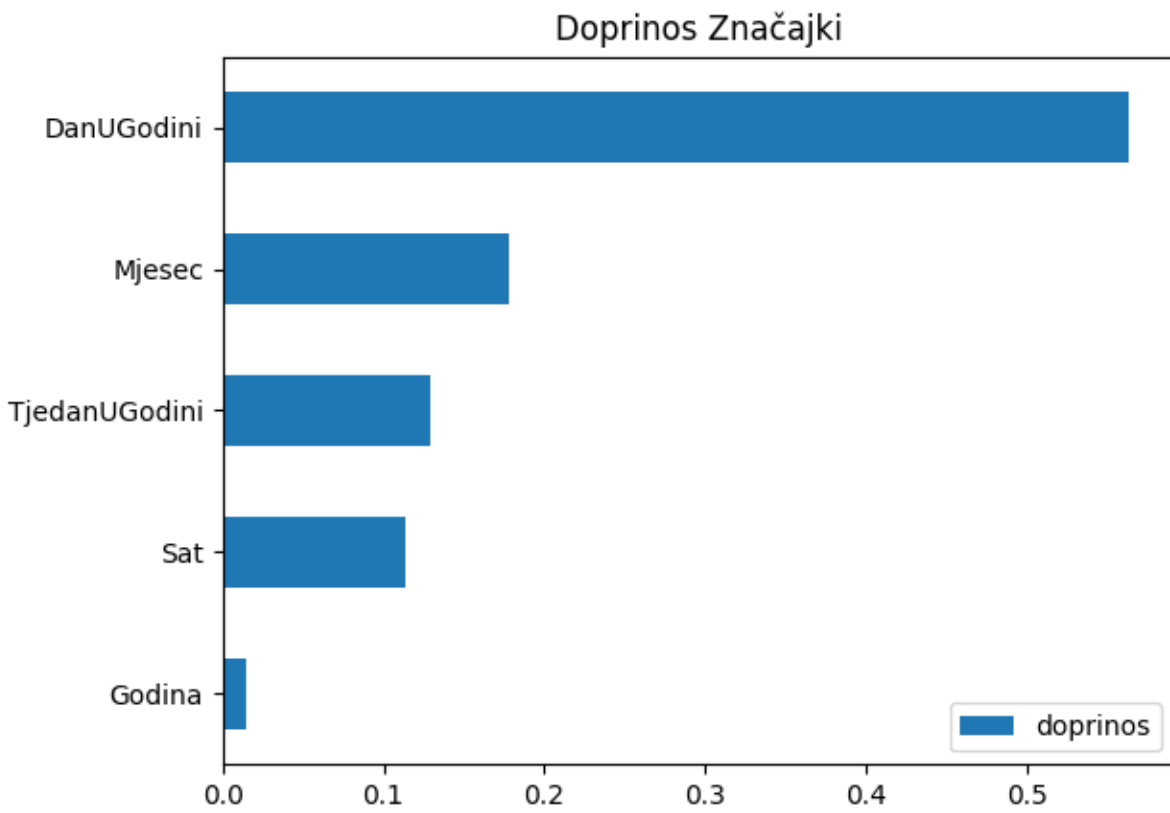


Slika 6.7. Usporedba predikcije modela sa mjerenjima kroz godine



Slika 6.8. Usporedba predikcije modela sa mjerenjima u ožujku 2020., interpolirani prikaz

Iz slika predikcije može se uočiti kako su predikcije za svaku godinu vrlo slične te da se predikcija ponavlja periodično svake godine. Razlog takvog ponašanja modela je u odabiru značajki, odnosno ulaznih vrijednosti modela. Trenutne značajke ('Sat', 'Mjesec', 'Godina', 'DanUGodini', 'TjedanUGodini') opisuju periodični vremenski trenutak u godini. Uvidom u usporedbu doprinosa značajki, na slici 6.9., može se uočiti da značajka *DanUGodini* najviše doprinosi rezultatu predikcije te da značajka *Godina* ima jako mali utjecaj. Odnosno, model kod treniranja i predikcije ne uzima u obzir trenutnu godinu, što rezultira godišnjoj periodičnosti predikcija. Uostalom, godišnja periodičnost je prisutna i u podacima mjerenja.



*Slika 6.9. Doprinos značajki modela*

### 6.2.3. Povećanje točnosti predikcije smanjenjem vremenskog prozora predikcije

Izradom značajki zaostatka (*eng.* lag features) moguće je povećati točnost modela uz uvjet da se vremenski prozor predikcije (vrijeme u budućnosti koje je moguće predvidjeti) smanji. Značajke zaostatka su koristan koncept u analizi podataka s vremenskom domenom i predviđanju istih. Značajke zaostatka podrazumjevaju uključivanje prošlih opažanja ciljane varijable u model kao značajke (ulazni podaci modela). Ideja je uhvatiti vremenske uzorke prisutne u podacima, gdje trenutna vrijednost varijable predviđanja može ovisiti o njezinim prošlim vrijednostima. Korištenjem značajki kašnjenja u modelu, moguće je iskoristiti povijesne uzorke i trendove u podacima kako bi se napravila točnija predviđanja budućih vrijednosti [19].

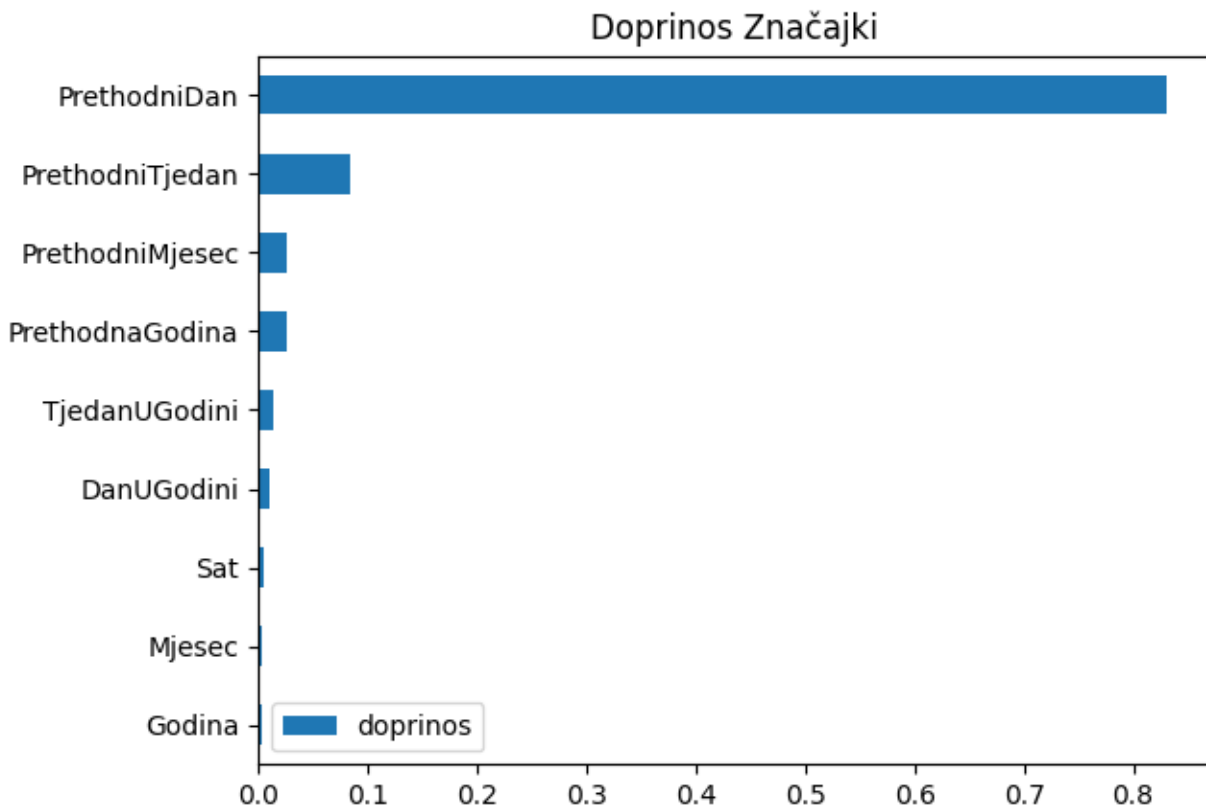
```
def lagZnacajke(df):
    df = df.copy()
    df["PrethodniSat"] = df["Temperatura"].shift(1)
    df["PrethodniDan"] = df["Temperatura"].shift(24)
    # Temperatura u istom satu u prethodnom tjednu
    df["PrethodniTjedan"] = df["Temperatura"].shift(24 * 7)
    # Temperatura u istom satu prije 30 dana
    df["PrethodniMjesec"] = df["Temperatura"].shift(24 * 30)
    # Temperatura u istom satu prošle godine
    df["PrethodnaGodina"] = df["Temperatura"].shift(24 * 365)
    return df
```

#### *Isječak koda 6.5. Funkcija za izradu značajki zaostatka*

U isječku koda 6.5. prikazana je izrada značajki zaostatka. Kreiraju se novi stupci koji, za svako mjerenje, sadrže vrijednosti temperature u prethodnim povijesnim trenucima. Ako se koristi samo značajka temperature u prethodnoj godini (*PrethodnaGodina*) onda je vremenski prozor predikcije ograničen samo na jednu godinu unaprijed, jer za svaki budući trenutak predikcije potrebno je znati temperaturu prije točno godinu dana od trenutka predikcije. Korištenjem značajke zaostatka za prethodni mjesec, vremenski prozor predikcije se ograničava na mjesec, 30, dana unaprijed. Te se tako vremenski prozor predikcije smanjuje ovisno o udaljenosti najbliže značajke zaostatka od trenutka predikcije.

Korištenje značajki *PrethodnaGodina*, *PrethodniMjesec* i *PrethodniTjedna* nije imalo značajan utjecaj na točnost modela, ali dodavanjem značajke *PrethodniDan*, odnosno, temperature 24 sata prije trenutka predikcije, točnost modela se povećava značajnije. Prosječno apsolutno odstupanje iznosi 2.4

°C, što je razlika u odnosu na prethodnih 3.5 °C. Iz slike 6.10. može se uočiti veliki utjecaj značajke *PrethodniDan* i ostalih značajki zaostatka na predikciju. Usporedbom sa utjecajem značajki iz prethodnog modela sa slike 6.9. može se zaključiti da se načini razumijevanja podataka između dva modela bitno razlikuju.

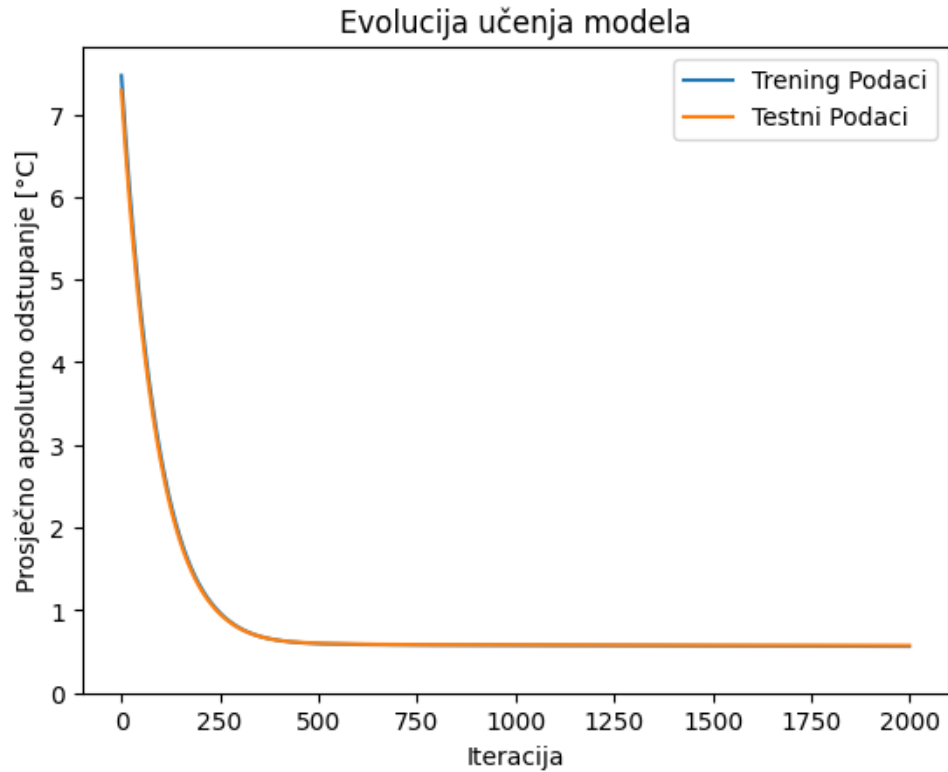


Slika 6.10. Doprinos značajki modela sa značajkama zaostatka

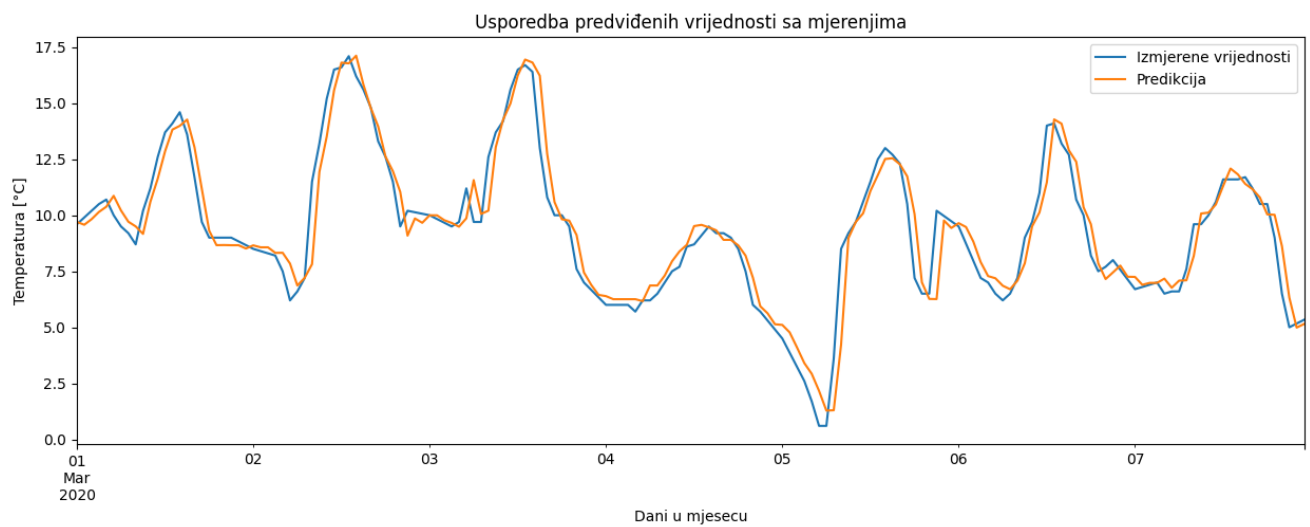
Ako se, dodatno, modelu doda značajka zaostatka koja prati temperaturu u jednom vremenskom trenutku prije trenutka predikcije, modelu se povećava točnost, prosječno apsolutno odstupanje iznosi 0.6 °C. Ali tada je vremenski prozor predikcije modela ograničen na jedan sat u budućnost. Evolucija učenja takvog modela prikazana je na slici 6.11. Može se uočiti da nema prekomjernog prilagođavanja modela te da model već u ranim iteracijama treniranja postaje točan. Na slici 6.12. prikazana je usporedba predikcija modela sa mjerenjima za ožujak 2020. godine. Može se učiti vrlo dobra točnost predikcije, ali bitno je uzeti u obzir da su prikazane predikcije, predikcije samo jednog sata unaprijed. Tablica 6.1. prikazuje točnosti promatranih modela s obzirom na udaljenost vremenskog prozora



predikcije. Može se zaključiti kako se točnost modela smanjuje s obzirom na povećanje vremenskog prozora predikcije.



Slika 6.11. Evolucija učenja modela sa vremenskim prozorom predikcije od jednog sata



Slika 6.12. Usporedba točnosti modela sa vremenskim prozorom predikcije od jednog sata, interpolirani prikaz

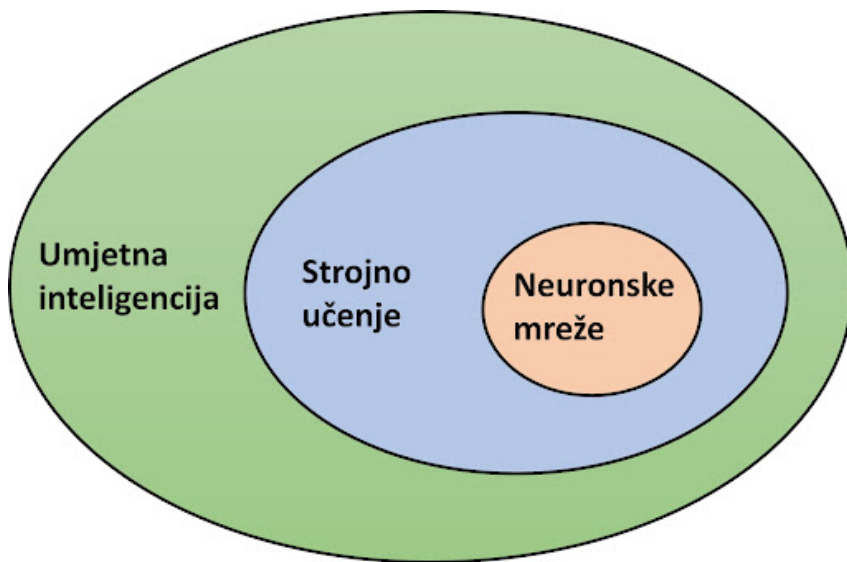
*Tablica 6.1. Usporedba točnosti predikcije i vremenskog prozora predikcije*

<b>Prosječno apsolutno odstupanje</b>	<b>Vremenski prozor predikcije</b>
3.5 °C	Neograničen
3.3 °C	Tjedan dana
2.4°C	Jedan dan
0.6 °C	Jedan sat

### 6.3. Predikcija temperature zraka pomoću neuronske mreže

U nastavku su sažeto opisane neuronske mreže te je opisan postupak predikcije temperature zraka pomoću LSTM rekurentne neuronske mreže.

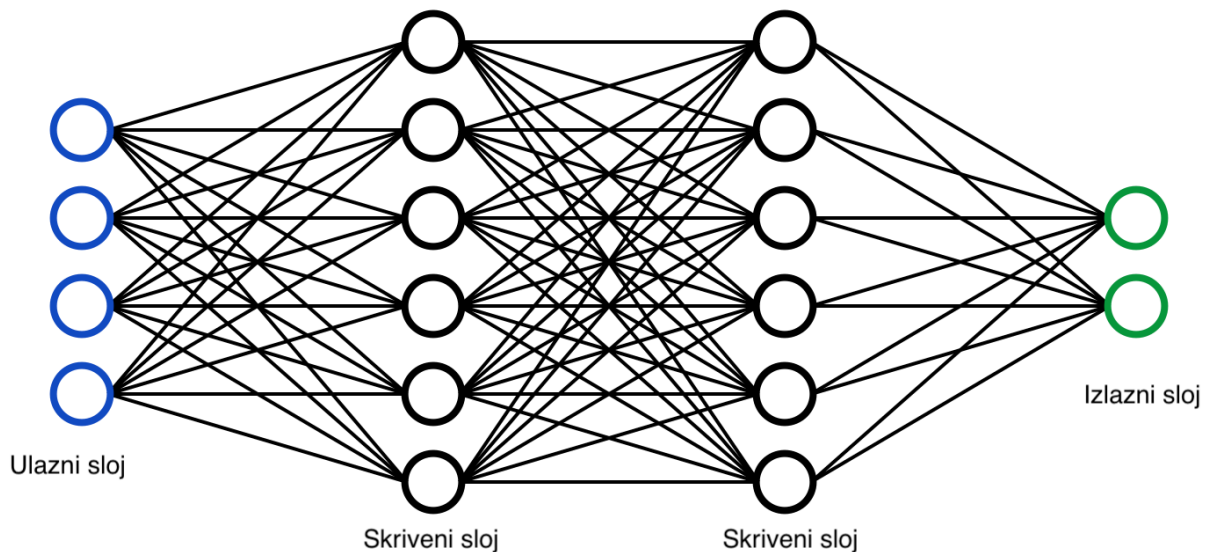
Neuronska mreža je arhitektura modela strojnog učenja inspirirana ljudskim mozgom. Neuronske mreže su podskup područja strojnog učenja i podskup područja umjetne inteligencije. Mreža prima ulaz te na temelju ulaza vraća izlaz. Na primjer, ulaz može biti slika, a izlaz može biti broj 0 ili 1, gdje jedinica predstavlja da se na ulaznoj slici nalazi auto, a nula predstavlja da na ulaznoj slici nema auta. Mreža se sastoji od mnogo međusobno povezanih elemenata koji se nazivaju neuroni. Neuroni su organizirani u slojeve, uključujući ulazni sloj, jedan ili više sakrivenih slojeva i izlazni sloj. Svaki neuron prima svoj ulaz, izvršava osnovne matematičke operacije nad ulazom i prenosi rezultat (izlaz) sljedećim neuronima. Veze između neurona povezane su s težinskim faktorima (*eng. weights*) koji se računaju i mijenjaju tijekom procesa treniranja mreže. Dodatni parametar koji se zove pristranost (*eng. bias*) prisutan je u svakom neuronu kako bi se izlaz neurona mogao preciznije podesiti.



Slika 6.13. Prikaz područja Neuronskih mreža u kontekstu područja Umjetne inteligencije

Neuronske mreže koriste aktivacijske funkcije za određivanje izlaza neurona na temelju njegovog ulaza. Uobičajene aktivacijske funkcije uključuju: sigmoidnu, hiperboličku tangensu i ispravljenu linearnu jediničnu funkciju. Mreža obrađuje podatke putem prosljeđivanja, gdje se podaci kreću od

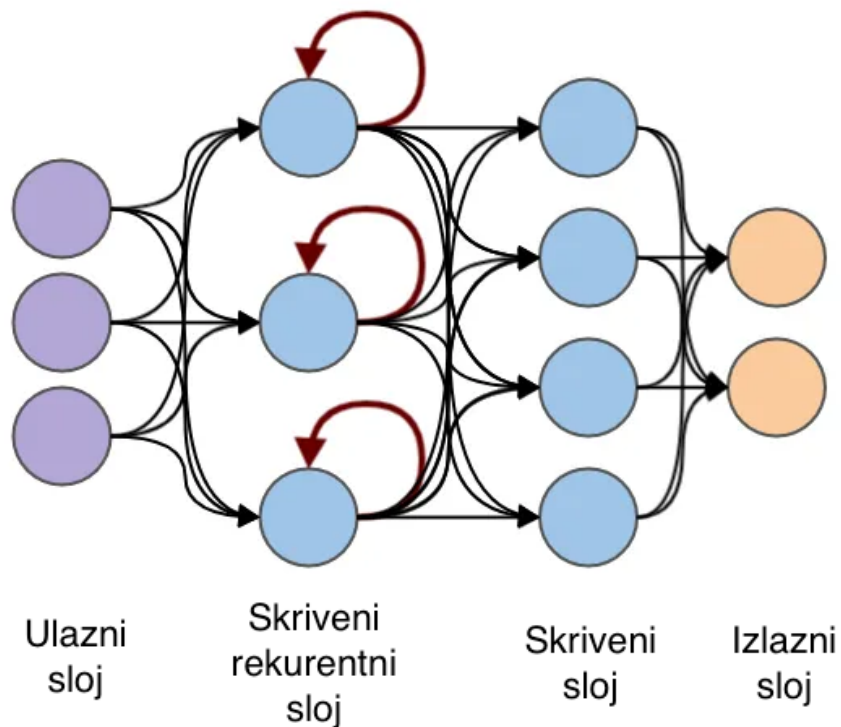
ulaznog sloja, kroz skrivene slojeve, do izlaznog sloja. Učinkovitost mreže procjenjuje se pomoću funkcije gubitaka (*eng.* loss function), koja mjeri razliku između predviđenih i stvarnih izlaza. Kako bi se poboljšala točnost izlaza mreže, mreža prolazi kroz fazu treniranja. Ključna tehnika kod treniranja je širenje unazad (*eng.* backpropagation). Širenje unazad uključuje izračunavanje nagiba funkcije gubitka s obzirom na svaki težinski faktor i prilagođavanje težinskog faktora kako bi se gubitak smanjio na minimum. Odnosno, težinski faktori mreže se podešavaju kako bi mreža na izlazu dala rezultat sa najvećom točnošću. Treniranje neuronske mreže obično uključuje iterativne algoritme, poput stohastičkog gradijentnog spuštanja. Postoje različite vrste neuronskih mreža, od kojih svaka odgovara različitim vrstama problema. Feedforward neuronske mreže su najjednostavniji oblik neuronskih mreža. Konvolucijske neuronske mreže imaju primjenu u prepoznavanju i klasificiranju slika. Rekurentne neuronske mreže dizajnirane su za sekvencijalne podatke, kao što su vremenske serije ili prirodni jezik [20].



Slika 6.14. Prikaz jednostavne neuronske mreže sa 4 sloja

### 6.3.1. Rekurentne neuronske mreže i LSTM mreže

Rekurentna neuronska mreža (*eng. Recurrent Neural Network*), skraćeno RNN, je vrsta neuronske mreže dizajnirane za obradu sekvencijalnih podataka. Za razliku od tradicionalnih neuronskih mreža, rekurentne mreže sadrže veze, između neurona, koje tvore usmjerene cikluse, što im omogućuje da imaju svojstvo memorije. Svojstvo memorije čine neuroni koji sadrže sakriveno stanje (*eng. hidden state*). Karakteristika pamćenja čini rekurentne mreže prikladnima za zadatke u kojima su važni kontekst i redoslijed, kao što je predviđanje vremenskih serija i obrada prirodnog jezika. Ulaz u svakom vremenskom koraku kombinira se s skrivenim stanjem iz prethodnog koraka kako bi se proizvelo novo skriveno stanje za trenutni vremenski korak te kako bi se izračunao izlaz trenutnog vremenskog koraka. Novo skriveno stanje se zatim koristi zajedno sa sljedećim ulazom iz niza kako bi se izračunao sljedeći izlaz i sljedeće skriveno stanje. Iteracija se nastavlja sve dok nije obrađen cijeli ulazni niz. Na taj način mreža pamti informacije o prethodnim ulazima dok obrađuje nove ulaze. Treniranje rekurentne mreže uključuje izračun težinskih faktora sa ciljem smanjivanja vrijednosti funkcija gubitka, slično kao i kod drugih neuronskih mreža [21].



Slika 6.15. Prikaz jednostavne rekurentne neuronske mreže

Long Short-Term Memory rekurentne neuronska mreža, skraćeno LSTM, specijalizirana je vrsta rekurentne neuronske mreže dizajnirane za bolje razumijevanje dugoročnih uzoraka u sekvencijalnim podacima. LSTM mreže rješavaju nedostatke tradicionalnih RNN mreža, konkretno, probleme nestajanja i raspršivanja gradijenata, koji usporavaju proces treniranja. LSTM ćelija se sastoji od ulaznih vrata, izlaznih vrata i vrata zaborava. Vrata reguliraju tok informacija na ulazu, izlazu i unutar ćelije. Ćelija sadrži svoju memoriju u kojoj pohranjuje relevantne informacije. Ulazna vrata kontroliraju koliko se novih informacija dodaje stanju ćelije, vrata zaborava određuju koje se informacije odbacuju iz stanja ćelije, a izlazna vrata upravljaju izlazom na temelju stanja ćelije i trenutnog ulaza. Tijekom procesa treniranja, LSTM ažurira svoja vrata i stanje ćelije pomoću algoritma širenja unazad (*eng.* backpropagation). Proces treniranja omogućuje mreži učenje bitnih informacija i zaborav nebitnih informacija. Mehanizam usmjeravanja informacija pomoću vrata, čini LSTM mreže vrlo efikasnim u prepoznavanju uzoraka u dugim nizovima podataka, što je ključno za zadatke kao što su jezično modeliranje, strojno prevođenje i predviđanje vremenskih serija [22].

### 6.3.2. Predikcija temperature pomoću LSTM mreže

Za izradu LSTM modela predikcije temperature korišten je isti NOAA skup podataka koji je korišten u prethodnom poglavlju za predikciju temperature pomoću gradient boosting algoritma.

Prije izrade i treniranja modela, potrebno je napraviti obradu podatka kako bi podaci bili u željenom formatu prikladnom za ulaz u neuronsku mrežu. Podaci se učitavaju u pandas podatkovni okvir, filtriraju se podaci samo sa mjerne postaje u Maksimiru te se odbacuju nepotrebni stupci u podacima. Vrijednosti stupca temperature se transformiraju u prikladni, ljudsko čitljivi, format, nepostojeće vrijednosti temperature se linearno interpoliraju te se dodaju nepostojeći datumi u indeks stupac podatkovnog okvira. Grafički prikaz podatka nakon inicijalne obrade je prikazan na slici 6.2.

Nakon inicijalne obrade podaci se dijele na skup podataka za treniranje i na skup podataka za testiranje, gdje trening podskup sadrži 75 % inicijalnih podataka, a testni podskup 25 % podataka. Podjela je grafički prikazana na slici 6.4.

Cilj modela je predviđanje temperature u sljedećem satu s obzirom na temperaturu u prethodnih 24 sata. Ulaz u model je definiran kao vremenska serija od 24 elementa, temperatura u prethodnih 24 sata, a izlaz modela je decimalni broj, predviđena temperatura. Stoga, potrebno je transformirati podatke za treniranje s obzirom na ulaz.

Prije sljedeće transformacije podataka, podaci su pretvoreni u TensorFlow skup podataka radi lakše transformacije i radi same kompatibilnosti podatka sa knjižnicom funkcija TensorFlow. Podaci se transformiraju u vremenske prozore gdje svaki vremenski prozor sadrži listu od 24 elemenata te vrijednost oznake kao zasebnu vrijednost odvojenu od sekvence ulaza. Prvi vremenski prozor je definiran sa 25 uzastopnih vrijednosti temperature počevši od prvog elementa unutar trening podskupa podataka, drugi vremenski prozor je isto definiran pomoću 25 uzastopnih elemenata, ali počevši od drugog elementa iz trening podskupa. Na taj način je definirano 149016 vremenskih prozora. U isječku koda 6.6. prikazan je jedan vremenski prozor. Programski, vremenski prozor je jedan uređeni par (*eng.* tuple) koji u sebi sadrži dva Tensor objekta. Prvi tenzor objekt sadrži polje podatka od 24 vrijednosti, a drugi tenzor objekt sadrži samo jednu decimalnu vrijednost. Prvi tenzor je ulaz u model, a drugi tenzor je oznaka (ciljana vrijednost).

```

(<tf.Tensor: shape=(24,), dtype=float64, numpy=
array([ 8.9      ,  8.8      ,  9.1      ,  8.8      ,  7.2      ,
        4.3      ,  2.8      ,  1.5      ,  0.6      ,  0.23333333,
       -0.13333333, -0.5      , -0.86666667, -1.23333333, -1.6      ,
       -1.9      , -2.2      , -2.3      , -1.8      ,  0.4      ,
        2.1      ,  3.4      ,  5.6      ,  8.9      ])>,
<tf.Tensor: shape=(), dtype=float64, numpy=8.6>)

```

*Isječak koda 6.6. Prikaz podataka jednog vremenskog prozora*

Nakon izrade vremenskih prozora prozori se grupiraju u hrpe (*eng.* batch). Batch grupira više vremenskih prozora u jednu logičku cjelinu. Jedan batch, u ovom slučaju, sadrži 887 vremenskih prozora. Stoga je cijeli skup podataka za treniranje sastavljen od 168 batch-eva, jedan batch je sastavljen od 887 vremenskih prozora, a jedan vremenski prozor ima 24 vrijednosti i jednu ciljanu vrijednost temperature.

Arhitektura modela neuronske mreže prikazana je u isječku koda 6.7. Model mreže je sekvencijalan te definira slojeve mreže gdje su slojevi povezani jedan za drugim u linearni niz. Prvi sloj je Lambda sloj koji primjenjuje lambda funkciju za proširenje ulaznih dimenzija. Točnije, dodaje dodatnu dimenziju ulaznom tenzoru, kako bi tenzor bio prikladnih dimenzija kao ulaz za sljedeći LSTM sloj. Nakon Lambda sloja slijede dva dvosmjerna LSTM sloja. Dvosmjerni LSTM-ovi su vrsta rekurentnog sloja neuronske mreže koji obrađuju podatke između vremenskih koraka, poboljšavajući sposobnost modela da nauči uzorke u sekvencijalnim podacima. Prvi dvosmjerni LSTM sloj sadrži 128 ćelija i postavljen je da vraća, što znači da vraća puni niz stanja ćelija za svaki element ulazne sekvence. Stanja ćelija vraća drugom dvosmjernom LSTM sloju sa 128 jedinica. Nakon LSTM slojeva, model sadrži gusti sloj s 256 jedinica i ReLU aktivacijskom funkcijom. Gusti sloj uvodi nelinearnost i omogućuje modelu da nauči složenije uzorke u podacima. Nakon gustog sloja, slijedi Dropout sloj koji nasumično gasi 40% neurona kako bi se spriječilo prekomjerno prilagođavanje i poboljšala sposobnost generalizacije modela. Završni sloj je još jedan gusti sloj sa jednim neuronom koji se koristi kao izlaz iz modela.



```

lstmModel = Sequential([
    Lambda(lambda x: tf.expand_dims(x, axis=-1), input_shape=[prozor]),
    Bidirectional(LSTM(128, return_sequences=True)),
    Bidirectional(LSTM(128)),
    Dense(256, activation='relu'),
    Dropout(0.4),
    Dense(1)
])

lstmModel.compile(
    loss=Huber(),
    optimizer=Adam(),
    metrics=['mae']
)

```

*Isječak koda 6.7. Prikaz arhitekture trening parametara LSTM neuronske mreže*

Kao funkcija gubitka koristi se Huberova funkcija gubitka koja je dizajnirana tako da bude manje osjetljiva na vrijednosti izvan raspona (*eng.* outliers) od funkcije prosječnog apsolutnog odstupanja. Za prilagođavanje parametara koristi se Adam algoritam, a kao metrika točnosti koristi se prosječno apsolutno odstupanje.

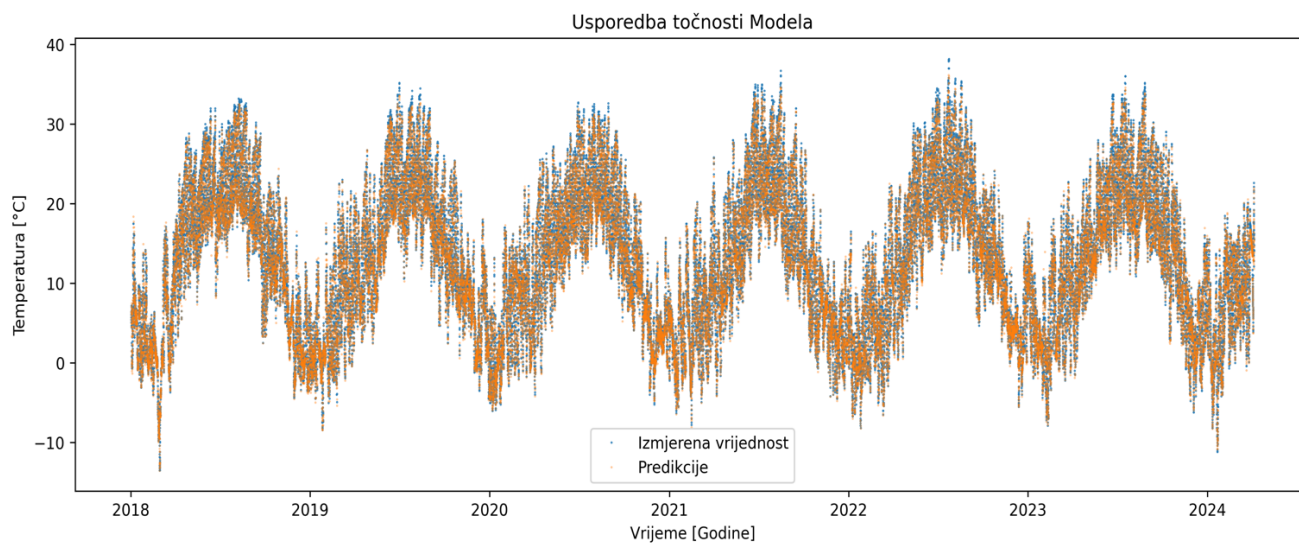
Model se trenira kroz 10 prolazaka kroz trening skup podataka. Loss funkcija na kraju treniranja iznosi 0.6166, a prosječno apsolutno odstupanje iznosi 1.0 °C.

Nakon treniranja modela, izrađuje se tensorflow skup podataka sa podacima za testiranje te se istrenirani model pokreće sa testnim podacima. Nakon pokretanja modela nad podacima za testiranje računa se prosječno apsolutno odstupanje nad izlazom iz modela te ono iznosi 0.6 °C.

Tablica 6.2. Usporedba predviđenih i izmjerenih vrijednosti temperature za Maksimir u Zagrebu

Vremenski trenutak	Izmjerena vrijednost	Predikcija
2024-04-03 01:00:00	5.1 °C	5.4 °C
2024-04-03 02:00:00	4.6 °C	4.7 °C
2024-04-03 03:00:00	4.1 °C	4.2 °C
2024-04-03 04:00:00	3.9 °C	3.7 °C
2024-04-03 05:00:00	4.2 °C	3.7 °C

U tablici 6.2. prikazan je podskup izlaza iz modela uspoređen sa izmjerenim vrijednostima. Bitno je naglasiti da je vrijednost predikcije u određenom vremenskom trenutku izračunata na temelju prethodnih 24 izmjerenih vrijednosti.



Slika 6.16. Usporedba točnosti modela nad podacima za testiranje

Na slici 6.16. prikazana je usporedba predviđenih vrijednosti sa izmjerenim vrijednostima nad cijelom testnom skupu podataka. Može se uočiti da predviđene vrijednosti prate periodički trend izmjerenih vrijednosti, što se može zaključiti i iz male vrijednosti prosječnog apsolutnog odstupanja od 0.6 °C.

Prethodne usporedbe ne opisuju dovoljno dobro rezultate razvijenog modela. Model kao ulaz prima 24 konsektivne vrijednosti temperature, odnosno, vrijednosti temperature u posljednja 24 sata. Te kao izlaz, model predviđa vrijednost temperature u sljedećem vremenskom trenutku (satu). Stoga je bitno usporediti predviđene vrijednosti u odnosu na prethodne vrijednosti, odnosno, usporediti izlaz u odnosu na ulaz modela. Usporedba je prikazana na grafovima na slici 6.16. gdje su prikazane 3 sekvence vrijednosti temperature i predikcija zadnje vrijednosti.



#### **6.4. Usporedba gradient boosting algoritma i LSTM neuronske mreže**

Model LSTM neuronske mreže i model gradient boosting algoritma pokazali su se kao poprilično dobri algoritmi za predikciju vremenskih podataka, odnosno, temperature. Iako se arhitektura dva modela bitno razlikuje, modeli se mogu primijeniti na istom problemu strojnog učenja, odnosno na predikciji vremenskih podataka. Ako usporedimo točnost LSTM neuronskog modela i gradient boosting algoritma sa vremenskim prozorom predikcije od jednog sata, vidi se da razlika nije velika. Prosječno apsolutno odstupanje gradient boosting algoritma iznosi  $0.6\text{ }^{\circ}\text{C}$ , dok prosječno apsolutno odstupanje neuronskog LSTM modela iznosi  $0.6\text{ }^{\circ}\text{C}$ . Treniranje modela gradient boosting algoritma, uz obradu ulaznih vrijednosti, zahtjeva i dodatnu optimizaciju hiper-parametara kao što su: stopa učenja, broj stabla, dubina stabla. Ali je zato treniranje modela gradient boosting algoritma puno brže od treniranja neuronske LSTM mreže te zahtjeva puno manje računalnih resursa. LSTM neuronska mreža zahtjeva grafički procesor za treniranje i izvršavanje, dok se model gradient boosting algoritma može trenirati i izvršavati na središnjoj jedinici za obradu (CPU). Također, interpretabilnost gradient boosting algoritma je veća od modela LSTM mreže.

Model gradient boosting algoritma i model LSTM neuronske mreže su se pokazali kao pouzdanima i prikladnima za predikciju vremenskih podataka temperature zraka sa duljinom vremenskog prozora od jednog sata. Gradient boosting algoritam pokazuje prednosti u interpretabilnosti i performansama dok LSTM neuronska mreža može uočiti razne kompleksne uzorke u vremenskim podacima.

## 7. ZAKLJUČAK

Cilj ovog rada bio je analizirati točnost različitih vremenskih prognoza te istražiti primjenu metoda strojnog učenja na problemu predikcije temperature zraka. Analizirana je točnost triju prognoza: subjektivna prognoza minimalne i maksimalne temperature koju izrađuju meteorolozi DHMZ-a, numerička prognoza temperature i količine padalina DHMZ-a i numerička prognoza Norveškog instituta za meteorologiju koja prognozira temperaturu, količinu padalina, relativnu vlažnost zraka, tlak zraka te brzinu i smjera vjetra. Prognoze DHMZ-a prognoziraju vrijeme za gradove i mjesta diljem Republike Hrvatske te su u analizi uzeta u obzir mjesta i gradovi za koje postoje mjerenja sa mjernih postaja. Numerička prognoza Norveškog instituta za meteorologiju nudi prognozu za bilo koju geografsku točku na planeti, a u sklopu ove analize uzeta je u obzir prognoza za Maksimir u Zagrebu. Za potrebe prikupljanja i pohranjivanja podataka izrađena je python skripta koja periodično sa mrežnih mjesta prognoza pohranjuje podatke prognoza u bazu podataka. Prikupljeni podaci obuhvaćaju period od 2.4.2024. do 10.6.2024. Analiza točnosti napravljena je usporedbom podataka prognoza sa podacima izmjerenih vrijednosti sa mjernih postaja DHMZ-a. Točnost vremenskih prognoza je izračunata kao prosječno apsolutno odstupanje (PAO) prognoziranih vrijednosti od izmjerenih vrijednosti. Greška, PAO, DHMZ subjektivne prognoze minimalne i maksimalne temperature za trenutni i naredni se kreće između 0.69 °C i 1.96 °C. Greška DHMZ numeričke prognoze temperature raste linearno s obzirom na porast vremenskog prozora predikcije te se greška kreće od 1.2 °C do 2.2 °C. DHMZ numerička prognoza količine padalina pokazuje veliku točnost sa PAO koji se kreće od od 5.5 mm do 7 mm. Numerička prognoza količine padalina od Norveškog instituta za meteorologiju ima manju točnost od DHMZ-ove prognoze. DHMZ numerička prognoza temperature ima veću točnost, nego prognoza Norveškog instituta za meteorologiju, u prvih 50 budućih vremenskih trenutaka, a zatim u sljedećim budućim trenutcima Norveška prognoza ima veću točnost. No bitno je uzeti u obzir da je skup svih analiziranih podataka vremenski obuhvatio samo 2 mjeseca te da su se DHMZ i YR prognoze uspoređivale samo za jedno geografsko mjesto, Maksimir u gradu Zagrebu. Greška numeričke prognoze tlaka zraka Norveškog instituta za meteorologiju raste eksponencijalno sa obzirom na rast vremenskog prozora predikcije. Numerička prognoza smjera vjetra Norveškog instituta za meteorologiju pokazala se kao netočnom sa samo 34.3 % točnih predikcija smjera vjetra za vremenski prozor predikcije od 24 sata. Dok greška predikcije brzine vjetra relativno mala te se kreće oko 1 m/s s obzirom na vremenski prozor predikcije. Greška numeričke prognoze vlage u zraku Norveškog instituta za meteorologiju raste linearno sa porastom vremenskog

prozora predikcije. Analizirane prognoze vremena imaju veliku točnost te se može zaključiti da su pouzdan izvor informacija o vremenu. Potencijalna buduća analiza točnosti meteoroloških prognoza uključivala bi analizu više različitih numeričkih modela te veći skup podataka sa većim vremenskim periodom i sa više geografskih mjesta. Na taj bi se način povećala točnost same analize te bi se moglo usporediti više različitih numeričkih modela.

Za predikciju temperature zraka izrađena i istrenirana su dva modela strojnog učenja. Modeli su trenirani na povijesnim podacima mjerenja temperature zraka sa mjerne postaje Maksimir u Zagrebu. Podaci obuhvaćaju period od 1.1.2000. godine do 5.4.2024. godine. Modeli predviđaju temperaturu zraka te rezolucija podataka ulaza i izlaza iz modela iznosi jedan sat. Prvi model strojnog učenja koristi gradient boosting algoritam, ima neograničeni vremenski prozor te može predvidjeti temperaturu zraka sa PAO od 3.5 °C. Smanjivanjem vremenskog prozora predikcije, povećava se točnost predikcije. Stoga model sa vremenskim prozorom predikcije od jednog sata ima PAO od 0.6 °C. Drugi promatrani model strojnog učenja koristi rekurentnu neuronsku mrežu. Model kao ulaz prima vrijednosti temperature u posljednja 24 sata te na izlazu vraća vrijednost predikcije temperature u 25. satu. Model ima PAO od 0.6 °C. Ako se usporedi točnost neuronskog modela i modela gradient boosting algoritma sa vremenskim prozorom predikcije od jednog sata, vidi se da razlika u točnosti ne postoji. Treniranje modela gradient boosting algoritma zahtjeva puno manje računalnih resursa od treniranja neuronskog modela. Također, interpretabilnost gradient boosting algoritma je veća od interpretabilnosti modela rekurentne neuronske mreže. Potencijalna buduća analiza algoritama strojnog učenja može uključivati analizu algoritama sa većim vremenskim prozorom predikcije te usporedbu sa raznim drugim algoritmima.

## 8. LITERATURA

- [1] „Weather forecasting“, s Interneta  
[https://en.wikipedia.org/wiki/Weather\\_forecasting?useskin=vector](https://en.wikipedia.org/wiki/Weather_forecasting?useskin=vector) 2. travnja 2024.
- [2] „O računalnim prognozama“, s Interneta  
[https://meteo.hr/prognoze.phpsection=prognoze\\_model&param=7d](https://meteo.hr/prognoze.phpsection=prognoze_model&param=7d) 29. ožujka 2024.
- [3] „Numerical weather prediction“, s Interneta  
[https://en.wikipedia.org/wiki/Numerical\\_weather\\_prediction?useskin=vector](https://en.wikipedia.org/wiki/Numerical_weather_prediction?useskin=vector) 29. ožujka 2024.
- [4] „Sedmodnevna prognoza“, s Interneta  
[https://meteo.hr/prognoze.php?section=prognoze\\_model&param=7d](https://meteo.hr/prognoze.php?section=prognoze_model&param=7d) 3. travnja 2024.
- [5] “HERS – High Resolution Forecasts”, s Interneta  
<https://confluence.ecmwf.int/display/FUG/Section+2.1.2.4+HRES+-+High+Resolution+Forecasts> 3. travnja 2024.
- [6] “Weather Forecasts on Yr – how are they made”, s Interneta <https://hjelp.yr.no/hc/en-us/articles/360004008874-Weather-forecasts-on-Yr-how-are-they-made> 3. travnja 2024.
- [7] „Python (programming language)“, s Interneta  
[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)?useskin=vector](https://en.wikipedia.org/wiki/Python_(programming_language)?useskin=vector) 4. travnja 2024.
- [8] „Beautiful Soup Documentation“, s Interneta <https://beautiful-soup-4.readthedocs.io/en/latest/> 4. travnja 2024.
- [9] „SQLAlchemy“, s Interneta <https://www.sqlalchemy.org/> 4. travnja 2024.
- [10] “Pandas”, s Interneta <https://pandas.pydata.org/docs/>, 10 travnja 2024.
- [11] “Matplotlib”, s Interneta <https://matplotlib.org/stable/index.html>, 10. travnja 2024.
- [12] „Machine learning“, s Interneta  
[https://en.wikipedia.org/wiki/Machine\\_learning?useskin=vector](https://en.wikipedia.org/wiki/Machine_learning?useskin=vector) 13. travnja 2024.
- [13] „A Guide to The Gradient Boosting Algorithm“, s Interneta  
<https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm> 13. travnja 2024.
- [14] „Gradient boosting“, s Interneta  
[https://en.wikipedia.org/wiki/Gradient\\_boosting?useskin=vector](https://en.wikipedia.org/wiki/Gradient_boosting?useskin=vector) 13. travnja 2024.
- [15] „Introduction to Boosted Trees“, s Interneta  
<https://xgboost.readthedocs.io/en/stable/tutorials/model.html> 13. travnja 2024.



- [16] „Optimizing XGBoost: A Guide to Hyperparameter Tuning“, s Interneta <https://medium.com/@rithpansanga/optimizing-xgboost-a-guide-to-hyperparameter-tuning-77b6e48e289d> 14. travnja 2024.
- [17] „Službena dokumentacija Scikit-learn biblioteke“, s Interneta [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) 14. travnja 2024.
- [18] „Službena dokumentacija XGBoost biblioteke“, s Interneta <https://xgboost.readthedocs.io/en/stable/> 14. travnja 2024.
- [19] „Basic Feature Engineering With Time Series Data in Python“, s Interneta, <https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/> 15. travnja 2024.
- [20] „Neural network (machine learning)“, s Interneta, [https://en.wikipedia.org/wiki/Neural\\_network\\_\(machine\\_learning\)?useskin=vector](https://en.wikipedia.org/wiki/Neural_network_(machine_learning)?useskin=vector) 5. lipnja 2024.
- [21] „The Unreasonable Effectiveness of Recurrent Neural Networks“, s Interneta <https://karpathy.github.io/2015/05/21/rnn-effectiveness/>, 5. lipnja 2024.
- [22] „Long short-term memory“, s Interneta, [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory?useskin=vector](https://en.wikipedia.org/wiki/Long_short-term_memory?useskin=vector) 5. lipnja 2024.
- [23] „Tensorflow dokumentacija“, s Interneta, <https://www.tensorflow.org/guide> 7. lipnja 2024.

## 9. SAŽETAK

Ovaj rad opisuje proces provođenja analize točnosti različitih vremenskih prognoza i proces izrade modela strojnog učenja za predikciju temperature zraka. Analizirana su i uspoređena dva izvora vremenske prognoze: Državni hidrometeorološki zavod Republike Hrvatske i Norveški institut za meteorologiju. Proces analize uključuje prikupljanje i pohranu meteoroloških podataka, analizu podataka te vizualizaciju i usporedbu rezultata. Rezultati su pokazali dobru točnost analiziranih prognoza te se numerička prognoza DHMZ-a pokazala boljom od prognoze Norveškog instituta za meteorologiju. Za predikciju temperature zraka izrađena su dva modela strojnog učenja: model algoritma pojačanja nagiba i model rekurentne neuronske mreže. Izračunata je točnost oba modela te su modeli međusobno uspoređeni. Modeli su pokazali dobru točnost kod predikcije temperature zraka.

### **Ključne riječi:**

Meteorološki modeli, Strojno učenje, Predikcija Temperature zraka, Neuronske mreže

### **SUMMARY**

This paper describes the process of analyzing the accuracy of various weather forecasts and the process of creating a machine learning models for air temperature prediction. Two sources of weather forecasts were analyzed and compared: the Hydrometeorological Institute of the Republic of Croatia and the Norwegian Institute of Meteorology. The analysis process includes the collection and storage of meteorological data, data analysis, and the visualization and comparison of results. The results showed a good accuracy of the analyzed forecasts, and the numerical forecast of DHMZ proved to be better than the forecast of the Norwegian Meteorological Institute. For air temperature prediction, two machine learning models were created: a gradient boosting algorithm model and a recurrent neural network model. The accuracy of both models was calculated and the models were compared with each other. The models showed good accuracy in air temperature prediction.

### **Keywords:**

Meteorological models, Machine Learning, Air Temperature Prediction, Neural Networks

## 8. POPIS OZNAKA I KRATICA

DHMZ	Državni hidrometeorološki zavod
XML	Extensible Markup Language
URL	Uniform Resource Locator
RH	Republika Hrvatska
HTTP	Hypertext Transfer Protocol
API	Application Programming Interface
CSV	Comma Seperated Valies
NaN	Not a Number
PAO	Prosječno Apsolutno Odstuapnje