Primjena konvolucijskih neuronskih mreža za modeliranje vremenskih nizova

Terzić, Teo

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI DIPLOMSKI STUDIJ

Teo Terzić

DIPLOMSKI RAD

Zagreb, rujan 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI DIPLOMSKI STUDIJ

Teo Terzić

PRIMJENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA MODELIRANJE VREMENSKIH NIZOVA

DIPLOMSKI RAD

Voditelj diplomskog rada: doc. dr. sc. Željka Ujević Andrijić

Članovi ispitnog povjerenstva:

doc. dr. sc. Željka Ujević Andrijić

doc. dr. sc. Miroslav Jerković

doc. dr. sc. Mario Lovrić

Zagreb, rujan 2023.

Ovaj diplomski rad izrađen je pod vodstvom doc. dr. sc. Marija Lovrića, te pod mentorstvom doc. dr. sc. Željke Ujević Andrijić sa Zavoda za mjerenja i automatsko vođenje procesa na Fakultetu kemijskog inženjerstva i tehnologije Sveučilišta u Zagrebu.

Prije svega zahvaljujem mentorici doc. dr. sc. Željki Ujević Andrijić na brojnim savjetima i strpljenju tijekom izrade diplomskoga rada. Također zahvaljujem vanjskome mentoru doc. dr. sc. Mariju Lovriću na strpljenju, predanome znanju i ponajviše njegovom izdvojenom vremenu.

Posebno zahvaljujem svojim roditeljima, punici i puncu na nesebičnoj pomoći. Najveću zahvalu dugujem svojoj supruzi Ivani koja mi je omogućila da sve stignem.

SAŽETAK

Visoke koncentracije dušikova dioksida (NO_2) u zraku mogu ozbiljno utjecati na ljudsko zdravlje. Dugotrajno izlaganje ovoj onečišćujućoj tvari može uzrokovati ili pogoršati respiratorne probleme, uzrokujući bolesti kao što su astma i bronhitis. U ovom radu razvijene su 1D konvolucijske neuronske mreže s ciljem predviđanja koncentracija NO_2 na urbaniziranom području grada Graza u Austriji. Za izradu modela korišten je programski jezik Python te pripadajuće biblioteke (Pytorch, Pandas, Numpy, itd.). Podatci su prikupljeni s mjerne postaje Zapad u Grazu. U svrhu optimizacije modela za predviđanje koncentracija NO_2 ispitivana je različita kombinacija hiperparametara modela kao što su veličina vremenskog prozora kod vremenskih nizova, stopa učenja i regularizacijski član *lambda*. Za razvoj modela koristile su se meteorološke, temporalne i *lag* značajke. Podskupovi podataka uzimani su u različitim vremenskim periodima: podskup za učenje (treniranje) modela, 1.1.2014 do 15.3.2018, podskup za validaciju, 15.3.2018 do 15.3.2019 te testni skup, 15.3.2019 do 15.3.2020. Razvijeni modeli pokazali su dobru sposobnost generalizacije, a najbolji rezultat modela konvolucijske neuronske mreže postignut je sa vremenskim prozorom 12, stopom učenja 0,0001 i regularizacijskim članom *lambda* 0,1 te postiže koeficijent determinacije R^2=0,62. Usporedno je razvijen i model slučajnih šuma (engl. *Random Forest*) koji postiže koeficijent determinacije R^2=0,65.

Ključne riječi: okolišni monitoring, konvolucijske neuronske mreže, duboko učenje

ABSTRACT

High concentrations of nitrogen dioxide (NO_2) in the air can seriously affect human health. Long-term exposure to this pollutant can cause or exacerbate respiratory problems such as asthma and bronchitis. In this study, 1D convolutional neural networks were developed with the aim of predicting NO_2 concentrations in the urban area of Graz, Austria. The Python programming language and corresponding libraries (Pytorch, Pandas, Numpy, etc.) were used to create the model. The data were collected from the West monitoring station in Graz. A combination of various hyperparameters, such as the size of the time window, the learning rate, and the regularization term *lambda*, was used for the optimization of the NO_2 concentration model. The features used for modeling included meteorological, temporal, and lag features. Subsets of data from different time periods were taken: the training subset from 1.1.2014 to 15.3.2018, the validation subset from 15.3.2018 to 15.3.2019, and the test subset from 15.3.2019 to 15.3.2020. These models showed good generalization ability, with the best result achieved with a time window of 12, a learning rate of 0.0001, and a regularization term *lambda* of 0.1, achieving a coefficient of determination R^2=0.62. A Random Forest model was also developed in parallel, achieving a coefficient of determination R^2=0.65.

Key words: environmental monitoring, convolutional neural networks, deep learning.

SADRŽAJ

1. UVOD

Zrak predstavlja jedan od osnovnih preduvjeta za opstanak i razvoj svakog života na Zemlji. Povećanjem ljudske populacije i podizanjem životnog standarda dolazi do povećane potrebe za potrošnjom energije. Većina energije dobiva se iz fosilnih goriva što dovodi do znatnih ekoloških i zdravstvenih problema.[1] Loša kvaliteta zraka, kako unutarnja tako i vanjska, prijeti pojavi brojnih respiratornih i kardiovaskularnih bolesti, s posljedičnim povećanjem globalne smrtnosti. Oko 2,4 milijarde ljudi kuha i grije svoje domove koristeći energiju dobivenu iz fosilnih goriva, a svake godine 3,2 milijuna ljudi prerano umre od posljedica onečišćenja zraka u kućanstvu. Više od 99% stanovništva živi u područjima gdje su prekoračene granične vrijednosti parametara koji ukazuju na onečišćenje zraka, a svake godine se 4,2 milijuna smrtnih slučajeva pripisuje onečišćenju zraka.[2] Ključnu ulogu u identificiranju i pokušaju smanjenja onečišćenja zraka ima okolišni monitoring. Kako bi se identificirali dugoročni trendovi koncentracije polutanata potrebno je kontinuirano i u realnom vremenu pratiti promjene onečišćenja zraka. U tu svrhu razvijaju se prediktivni modeli praćenja onečišćenja zraka. Odnos između ulaznih i izlaznih varijabli modela može biti vrlo složen, stoga je teško teorijskim modeliranjem dobiti podatke izlaznih vrijednosti. Duboko učenje jedan je od odličnih alata za opisivanje složenih odnosa između ulaznih i izlaznih vrijednosti modela. Praćenje kvalitete zraka primjenom metoda strojnog učenja pokazalo se kao kvalitetna praksa koja omogućava preciznije i efikasnije analize mjernih podataka. Strojno učenje doprinosi identifikaciji pojavljivanja karakterističnih uzoraka, predviđanju promjena i brzom otkrivanju odstupanja u kvaliteti zraka. [3]

U ovome radu ispitivane su 1D konvolucijske neuronske mreže za predviđanje izlaznih koncentracija NO_2 te su dobiveni rezultati uspoređivani s modelom nasumičnih šuma. Podatci korišteni za izradu modela (temperatura, relativna vlažnost zraka, brzina i smjer vjetra) prikupljeni su s mjerne postaje Zapad u Grazu. Za razvoj modela korišten je programski jezik Python.

2. TEORIJSKI UVOD

2.1. Onečišćenje zraka u urbanim sredinama

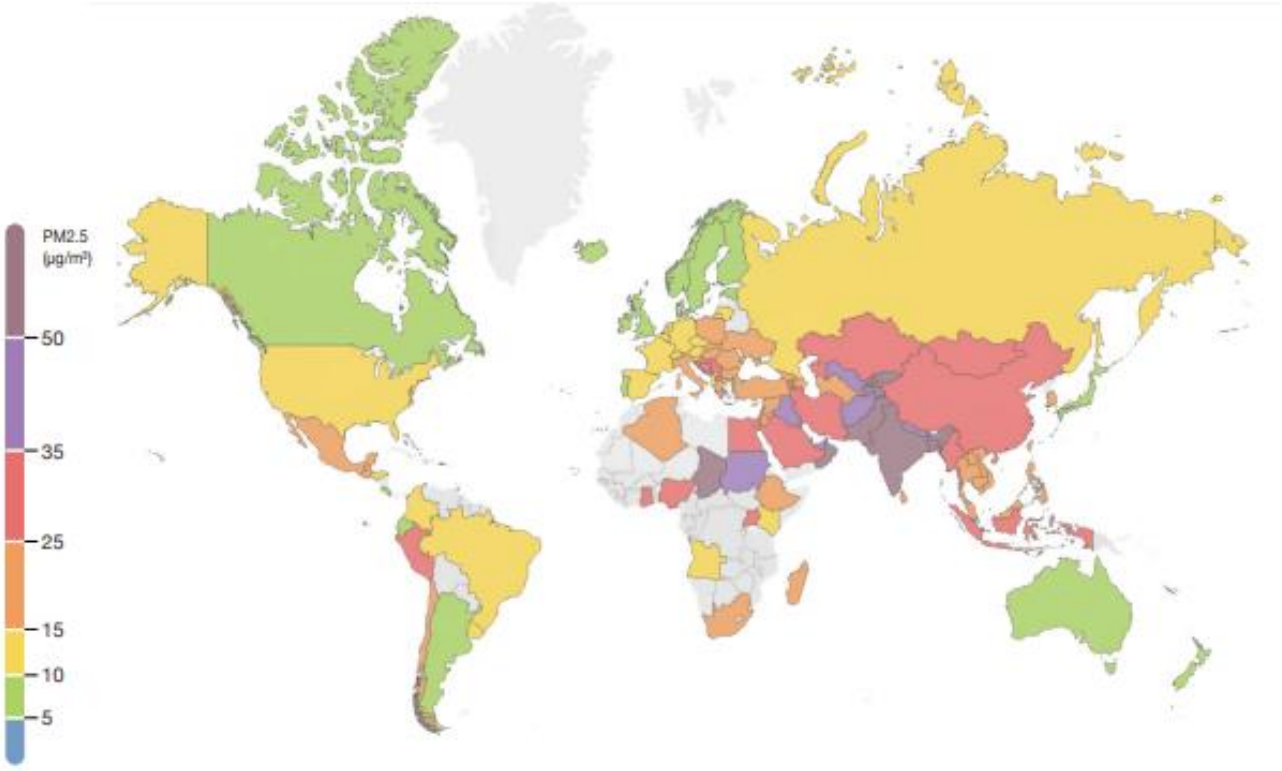
Danas, zbog povećane industrijalizacije, porasta broja osobnih automobila i izgaranja fosilnih goriva u industriji i kućanstvima ugrožena je kvaliteta zraka. Jedna od posljedica industrijske revolucije je i nastanak smoga. Smog je 1952. godine u Londonu uzrokovao najmanje 4000 smrtnih slučajeva unutar 5 dana.[4] Onečišćenje zraka uzrokuju smjese plinova i čestica u visokim koncentracijama koje ugrožavaju život na Zemlji. Izvori onečišćenja mogu biti prirodni i antropološki. Prirodni izvori onečišćenja su pojave koje se javljaju u prirodi poput erupcija vulkana i požara koje za posljedicu imaju ispuštanje onečišćujućih tvari u zrak. Antropogeni utjecaj najviše je izražen preko korištenja fosilnih goriva koja dominiraju u industrijskoj proizvodnji te grijanju u kućanstvima. Onečišćenje zraka izrazito utječe na ljudsko zdravlje i okoliš. Onečišćenje okoliša dovodi do pojava poput kiselih kiša, klimatskih promjena te globalnog zatopljenja.[1] U tablici 2.1. prikazano je 5 europskih gradova s najvećim onečišćenjem lebdećim česticama, $PM_{2,5}$ u zraku u 2021. godini.

Tablica 2.1. Pet europskih gradova s najvećim onečišćenjem lebdećim česticama, $PM_{2,5}$ u zraku u 2021. godini[5]

Grad, Zemlja	Koncentracija $\mu g/m^3$
Igdir, Turska	66,2
Krasnoyarsk, Rusija	49,8
Novi Pazar, Srbija	47,2
Foca, Bosna i Hercegovina	46,0
Duzce, Turska	44,4

2.2. Onečišćujuće tvari

Svakim danom sve je više dokaza negativnih utjecaja brojnih onečišćujućih tvari iz zraka na zdravlje ljudi. Identifikacija akutnih i dugoročnih štetnih učinaka onečišćenja zraka na zdravlje ljudi te određivanje graničnih koncentracija pri kojima dolazi do istih predstavlja velike izazove za znanstvenike.[1] Tvari koje najčešće uzrokuju onečišćenje zraka su: ugljikov monoksid, olovo, dušikovi oksidi, ozon, lebdeće čestice, sumporov dioksid.[6] Na slici 2.1. prikazane su srednje vrijednosti koncentracija lebdećih čestica $PM_{2,5}$ u svijetu.[5]



Slika 2.1. *Srednje vrijednosti koncentracija $PM_{2,5}$ čestica izražene u µg/m^3 za 2021. godinu*[5]

2.3. Dušikovi oksidi

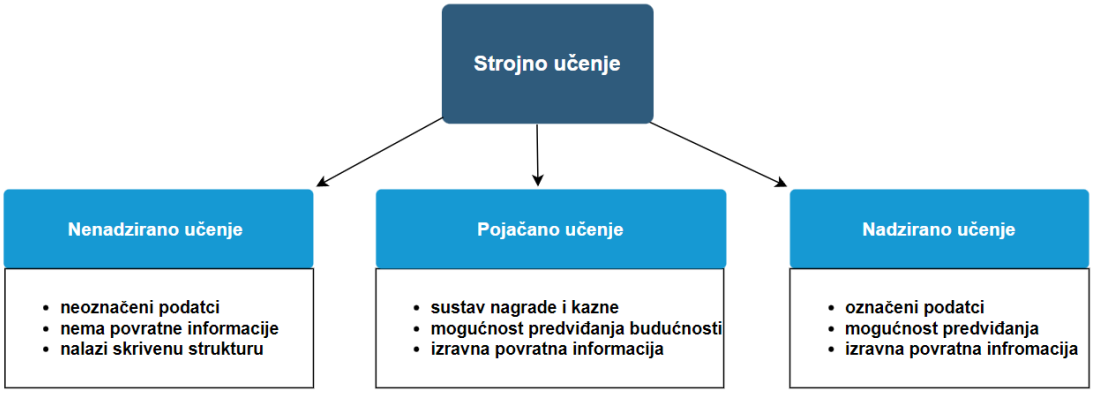
Dušikov dioksid, NO_2 je često sastavni dio složene smjese onečišćujućih tvari, s glavnim izvorima emisija koje potječu iz motornih vozila, termoelektrana, industrije te poljoprivrede. Zbog toga se NO_2 smatra značajnim uzročnikom emisija i njegov utjecaj na zdravlje ljudi izaziva značajan interes. Teško je izolirati utjecaj pojedinačnih onečišćujućih tvari u složenoj smjesi onečišćenog zraka. U zatvorenim prostorima NO_2 uglavnom nastaje izgaranjem plina bez ventilacije koji se koristi za kuhanje ili grijanje. NO_2 može poslužiti kao pokazatelj mješavine onečišćujućih tvari

koje nastaju izgaranjem plina. Osim toga, NO_2 ključni je prekursor za formiranje ozona u okolišu i podložan je atmosferskim transformacijama u različite vrste dušičnih kiselina i amonijak. Zbog toga je NO_2 značajan prekursor sekundarnih onečišćujućih tvari.[1]

Dugotrajno izlaganje NO_2 može doprinijeti razvoju bolesti kao što je astma i povećati osjetljivost na respiratorne infekcije. NO_2 i drugi dušikovi oksidi (NO_x) reagiraju s kemikalijama u zraku, stvarajući čestice i ozon, koji su također štetni kada se udišu zbog njihovih učinaka na dišne puteve. Nitratne čestice nastale iz NO_x čine zrak mutnim i smanjuju vidljivost te također doprinose onečišćenju obale i voda.[7] Podaci Svjetske zdravstvene organizacije (engl. *World Health Organization*, WHO) upućuju na to da je NO_2 povezan s više od 4 milijuna smrtnih slučajeva godišnje širom svijeta. Prema izvješću Europske agencije za okoliš (engl. *European Environment Agency*, EEA), NO_x uzrokuju smrt oko 75000 ljudi svake godine u Europi. Osim toga, onečišćenje NO_2 negativno utječe na okoliš, doprinoseći kiselosti tla i vode, što posljedično utječe na biljni i životinjski svijet. Stoga je od ključne važnosti kontinuirano praćenje i smanjenje emisija NO_2 kako bi se zaštitilo zdravlje ljudi te očuvala kvaliteta okoliša.[8]

2.4. Strojno učenje

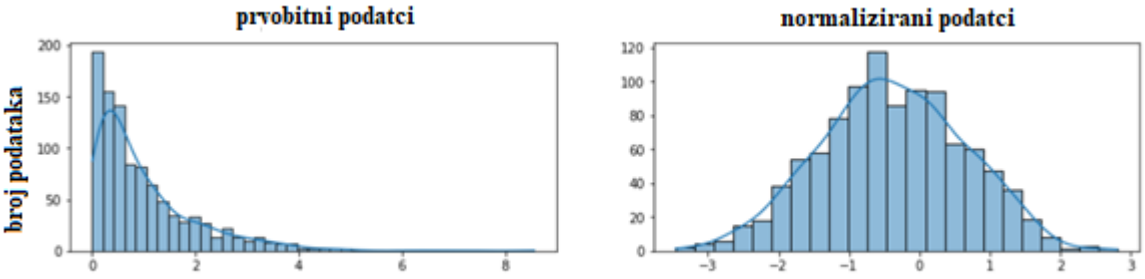
U doba moderne tehnologije dostupna je velika količina strukturiranih i nestrukturiranih podataka. U drugoj polovici 20. stoljeća, strojno učenje (engl. *machine learning*, ML) razvilo se kao područje umjetne inteligencije (engl. *artificial intelligence,* AI) koje uključuje algoritme samoučenja koji iz podataka izvlače znanje kako bi donosili određena predviđanja.[9] Postoje tri vrste strojnog učenja: nadzirano učenje, nenadzirano učenje te učenje pojačanjem (slika 2.2.). Nadzirano učenje temelji se na podatcima koje uz ulazne vrijednosti imaju i odgovarajuće izlazne vrijednosti, što omogućuje modelu da pronađe vezu između ulaza i izlaza. Nenadzirano učenje ne zahtijeva izlazne podatke. Umjesto toga, algoritmi traže skrivene strukture i veze unutar ulaznih podataka (npr. grupiranje sličnih vrijednosti). Učenje pojačanjem koristi agenta koji uči kako donositi odluke u nekom okruženju kako bi ostvario što veću nagradu. Učenje pojačanjem koristi se u problemima koji zahtijevaju optimalno donošenje odluka, kao što su igranje igara ili upravljanje robotom.[9]



Slika 2.2. *Različiti načini strojnog učenja*[9]

2.4.1. Predobrada podataka

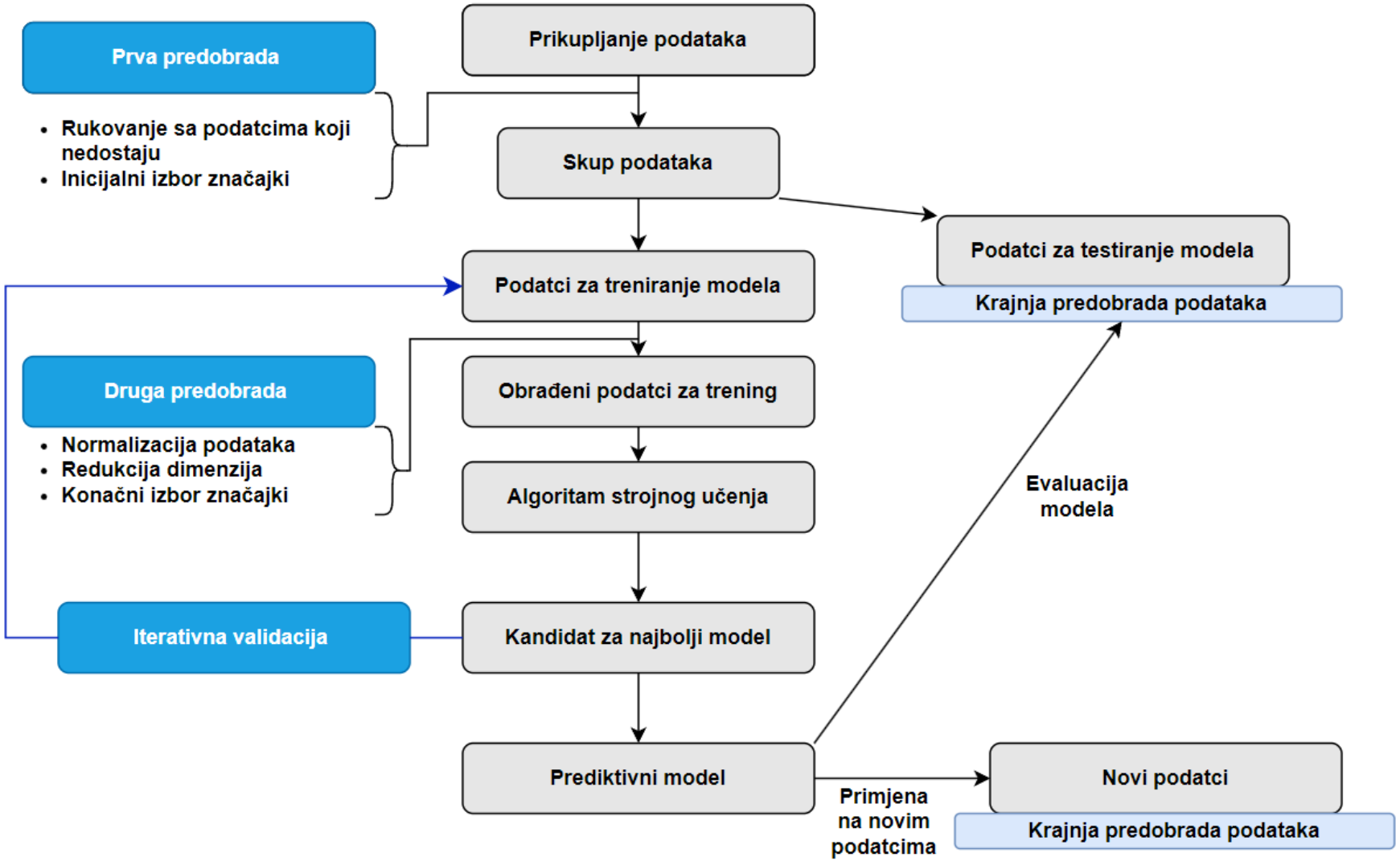
Dobiveni podaci često ne dolaze u obliku i stanju koji su potrebni za funkcioniranje algoritma učenja. Stoga je predobrada podataka jedan od najvažnijih koraka u bilo kojoj primjeni strojnog učenja. Cilj predobrade podataka je iz određenog skupa podataka izvući najkorisnije značajke za modeliranje.[9] Značajke u strojnome učenju predstavljaju ulazne varijable modela. Mnogi algoritmi strojnog učenja također zahtijevaju da odabrane značajke budu na istoj skali za optimalnu izvedbu modela, što se često postiže transformacijom značajki u rasponu [0, 1] ili standardnoj normalnoj distribuciji s nultom srednjom vrijednošću i jediničnom varijancom.[10] Prikaz promjene distribucije podataka nakon normalizacije prikazan je na slici 2.3.



Slika 2.3. *Podatci nakon normalizacije*

Neke od odabranih značajki mogu biti međusobno visoko korelirane i stoga suvišne za razvoj modela. U tim slučajevima korisne su tehnike smanjenja dimenzionalnosti gdje se značajke komprimiraju na niže dimenzionalni prostor ili se određene značajke izbacuju što utječe na smanjenje prostora za pohranu te algoritam učenja može raditi brže. U određenim slučajevima, smanjenje dimenzionalnosti

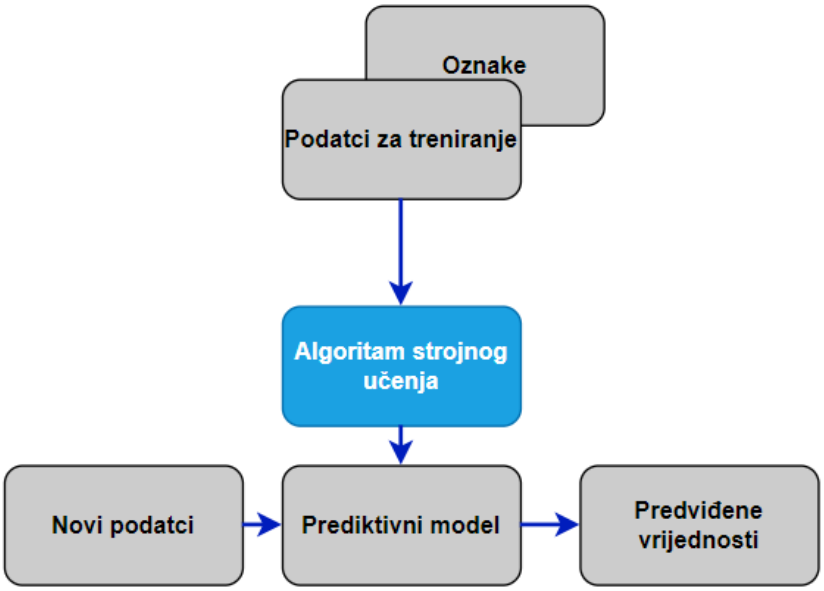
također može poboljšati predikcije modela ako skup podataka sadrži velik broj irelevantnih značajki ili šuma.[11] Da bi se utvrdila dobra funkcionalnost algoritma potrebno je nasumično podijeliti skup podataka na zasebne skupove podataka za treniranje i testiranje. Skup podataka za učenje ili treniranje modela koristi se za treniranje i optimizaciju modela strojnog učenja, dok se testni skup podataka čuva za procjenu konačnog modela.[9] Prikaz procesa strojnog učenja prikazan je na slici 2.4.



Slika 2.4. *Proces strojnog učenja* [9]

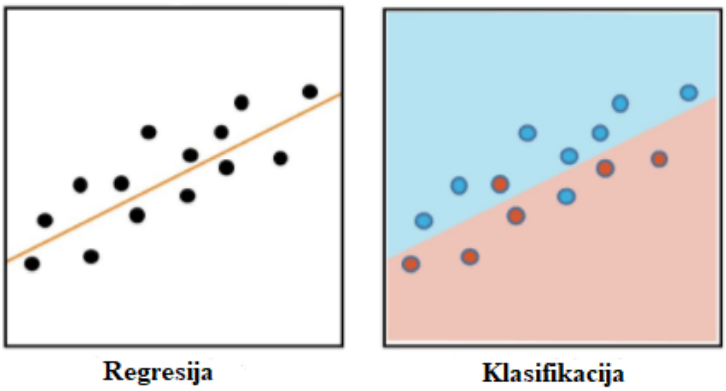
2.4.2. Nadzirano učenje

Nadzirano učenje koristi set podataka za treniranje za postizanje optimalnih parametara modela za željenu izlaznu vrijednost. Kod nadziranog učenja podatci za učenje sadrže odgovarajuće ulazne i izlazne vrijednosti koje omogućuju modelu da uči s vremenom. Algoritmu se mjeri točnost pomoću funkcije gubitka, koja korigira parametre modela sve dok funkcija gubitka ne poprimi minimalnu vrijednost (slika 2.5).[9]



Slika 2.5. *Prikaz procesa nadziranog učenja*[9]

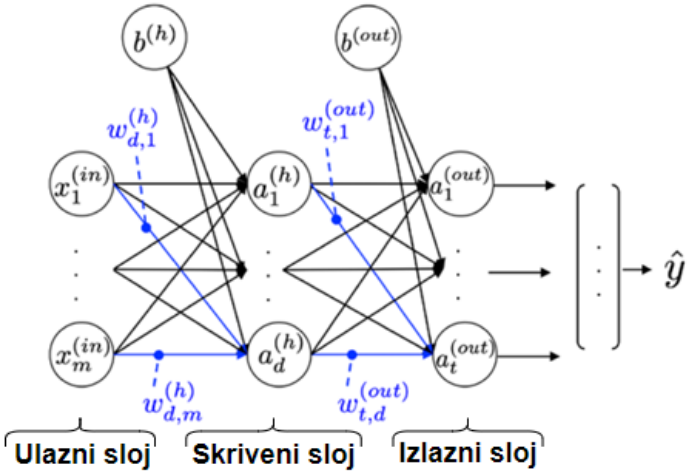
Postoje dva tipa nadziranog učenja, klasifikacija i regresija (slika 2.6.). Klasifikacija se koristi za predviđanje kategorije između diskretnih vrijednosti, dok se regresija koristi za predviđanje kontinuiranih vrijednosti izlazne varijable.[9]



Slika 2.6. *Prikaz razlike između regresije i klasifikacije*

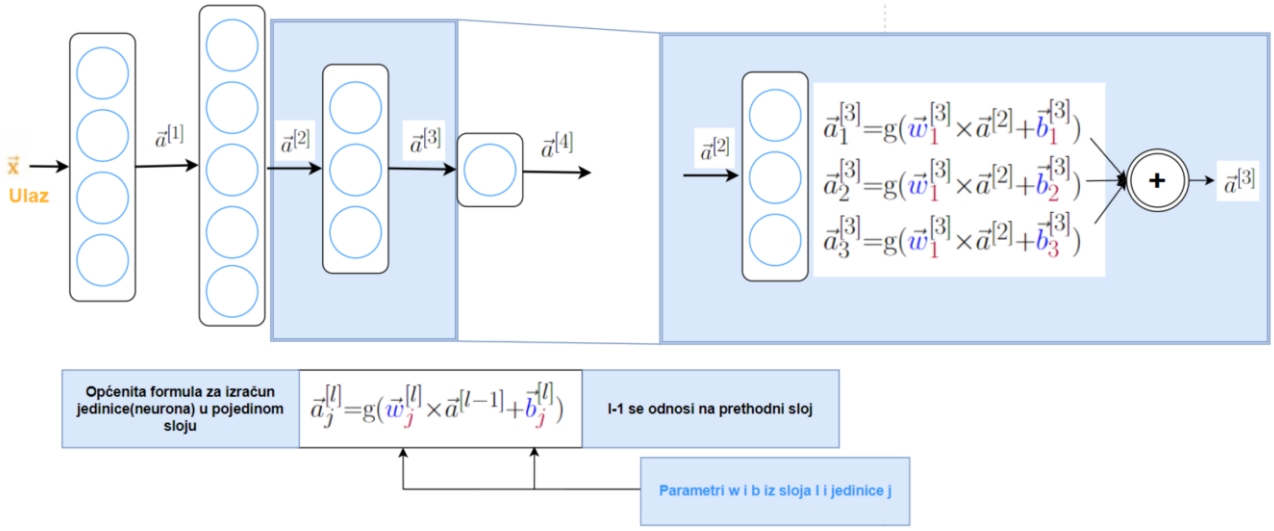
2.5. Neuronske mreže

Neuronske mreže preferiraju se ispred drugih metoda strojnog učenja zbog sposobnosti prilagođavanja složenim i nelinearnim problemima te sposobnosti učenja iz velikih količina podataka. Njihova robusnost i paralelno procesiranje čine ih pogodnima za primjenu u različitim industrijama gdje brzina proračuna i točnost igraju ključnu ulogu. Neuronske mreže su složene strukture koje se sastoje od tri osnovna sloja: ulaznog, skrivenog i izlaznog sloja (slika 2.7.). Proces koji se odvija između ulaznih i izlaznih podataka može se smatrati modelom crne kutije (engl. *black box model*) zbog svoje kompleksnosti. U svakom sloju mreže, aktivacijske funkcije koriste se za transformaciju i sumiranje informacija.[12]



Slika 2.7. *Prikaz neuronske mreže te njenih slojeva* [9]

Slojevi neuronskih mreža međusobno su povezani čvorovima. U svakom sloju „*l*“, čvorovi su povezani sa čvorovima iz sloja „*l+1*“ putem dva koeficijenta: težinskog koeficijenta (*w*) i pristranosti (*b*). Na primjer, veza između *k*-te jedinice u sloju „*l*“ i *j*-te jedinice u sloju „l + 1“ označena je kao $w_j^{(l)}$. Koeficijenti ili parametri mreže imaju ključnu ulogu u optimizaciji neuronske mreže. Čvorovi u mreži označeni su slovom „*a*“, a indeks čvora i sloja prikazuje se u zagradama: $a_i^{(l)}$ predstavlja *i*-ti čvor u *l*-tom sloju. Ovaj sustav označavanja omogućava jasnije razumijevanje strukture i povezanosti neuronskih mreža.[9] Također, zapis aktivacijskih jedinica i parametara može se pojednostaviti tako da se elementi svakog sloja pohrane u matricu. Ovaj pristup omogućuje promatranje parametara i aktivacijskih jedinica po slojevima.[9] Slika 2.8. prikazuje izračun za pojedine jedinice u neuronskoj mreži te pojednostavljeni način matričnog zapisa po slojevima.



Slika 2.8. *Detalji proračuna pojedinačnog neurona i čitavog sloja* [13]

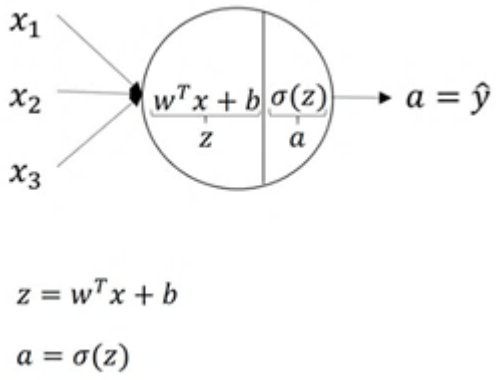
Proces treniranja neuronske mreže za izračunavanje izlaza modela može se sažeti u tri jednostavna koraka:

- Počevši od ulaznog sloja, unaprijedno se propagiraju uzorci trening podataka kroz mrežu kako bi se generirao izlaz mreže.
- Na temelju izlaza mreže, izračunava se pogreška koju treba minimizirati pomoću funkcije gubitka.
- Pogreška se propagira unatrag, pronalaze se njezine derivacije s obzirom na svaku težinu i jedinicu pristranosti u mreži, te se tako model ažurira.

Nakon prolaska navedenih koraka kroz više epoha, model sa istreniranim parametrima se koristi za izračunavanje izlaza mreže.[9]

2.5.1. Aktivacijske funkcije

Aktivacijske funkcije omogućuju neuronskim mrežama učenje i obavljanje složenih operacija. One predstavljaju nelinearne transformacije koje se primjenjuju na izlaz svakog neurona, omogućavajući modelima učenje i aproksimaciju nelinearne veze među podacima. Aktivacijske funkcije daju neuronskim mrežama mogućnost rješavanja problema koji se ne mogu riješiti samo linearnim modelima.[14] Na slici 2.9. prikazan je način izračunavanja vrijednosti jedinice te primjena aktivacijske funkcije.



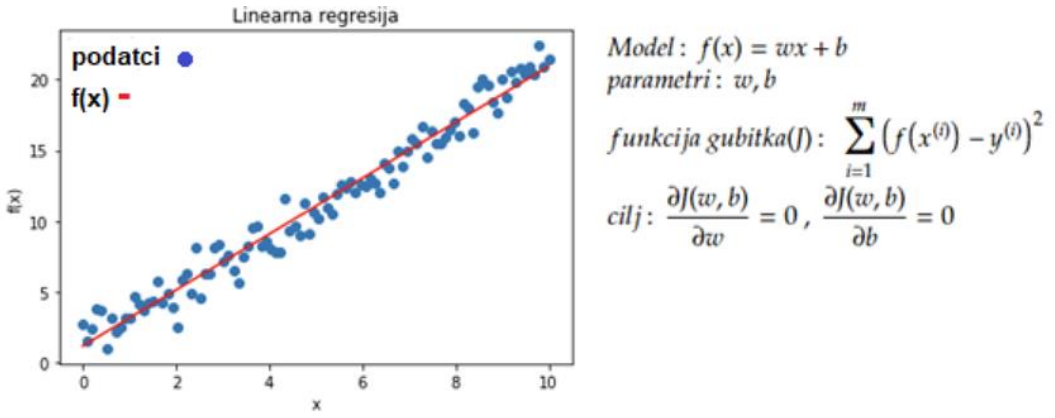
Slika 2.9. *Izračun aktivacijske funkcije* [13]

Postoji nekoliko vrsta aktivacijskih funkcija koje se koriste u praksi, a izbor funkcije ovisi o specifičnim zahtjevima. Najčešće korištene aktivacijske funkcije su sigmoidna, tangens hiperbolička (engl. *hyperbolic tangent*, tanh), ispravljena linearna aktivacijska funkcija (engl. *Rectified Linear Unit*, ReLU), ispravljena linearna aktivacijska funkcija s propuštanjem (engl. *Leaky Rectified Linear Unit*, *Leaky* ReLU) i normalizirana eksponencijalna funkcija (engl. *Softmax*). Sigmoidna i tangens hiperbolička su nelinearne funkcije koje mapiraju ulazne vrijednosti u određeni raspon. ReLU je jednostavna funkcija koja mapira sve negativne ulazne vrijednosti u nulu, a pozitivne vrijednosti ostavlja nepromijenjenima. *Leaky* ReLU je varijacija ReLU funkcije koja omogućuje malu negativnu vrijednost za negativne ulaze, čime se smanjuje problem nestajućeg gradijenta. *Softmax* funkcija se često koristi u zadnjem sloju klasifikacijskih neuronskih mreža jer pretvara izlazne vrijednosti u vjerojatnosti koje se zbrajaju na 1.[15] Aktivacijske funkcije su temeljni dio učenja neuronskih mreža jer omogućavaju propagaciju gradijenata pogreške unatrag kroz mrežu tijekom procesa optimizacije. Bez nelinearnosti koje aktivacijske funkcije uvode, neuronska mreža bila bi samo linearni model s ograničenom sposobnošću aproksimacije funkcija.[16]

2.5.2. Funkcija gubitka

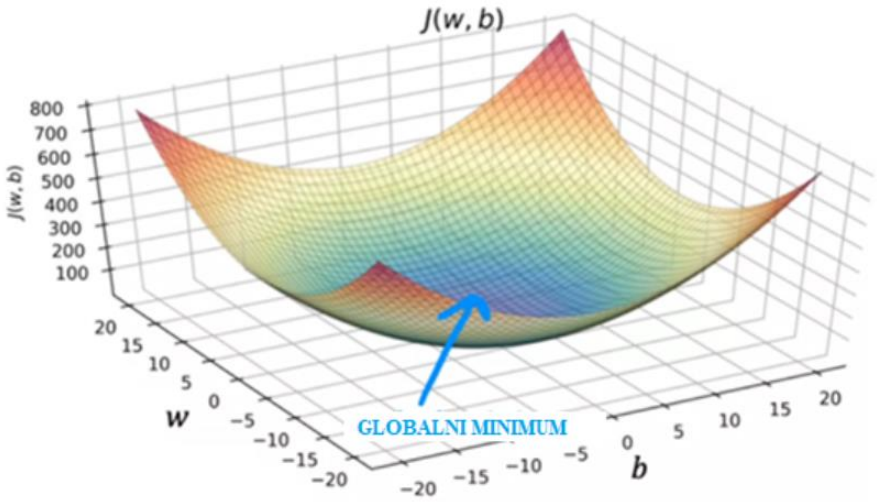
Funkcija gubitka (engl. *Loss function*) ima presudan značaj u procesu učenja neuronskih mreža. Ona kvantificira pogrešku modela, tj. razliku između predviđenih izlaza mreže i stvarnih ciljnih vrijednosti. Cilj optimiranja mreže je reducirati pogrešku modela prilagođavanjem parametara mreže. Najčešće korištene funkcije gubitka kod neuronskih mreža su srednja kvadratna pogreška (engl. *Mean squared*

error, *MSE*) i unakrsna entropija (engl. *Cross entropy*, CE). MSE se koristi za regresijske probleme u kojima je cilj smanjiti ukupnu kvadratnu razliku između predviđenih i stvarnih vrijednosti. S druge strane, CE se često primjenjuje u klasifikacijskim zadacima gdje CE omogućuje usporedbu vjerojatnosti pripadnosti različitim klasama.[17]



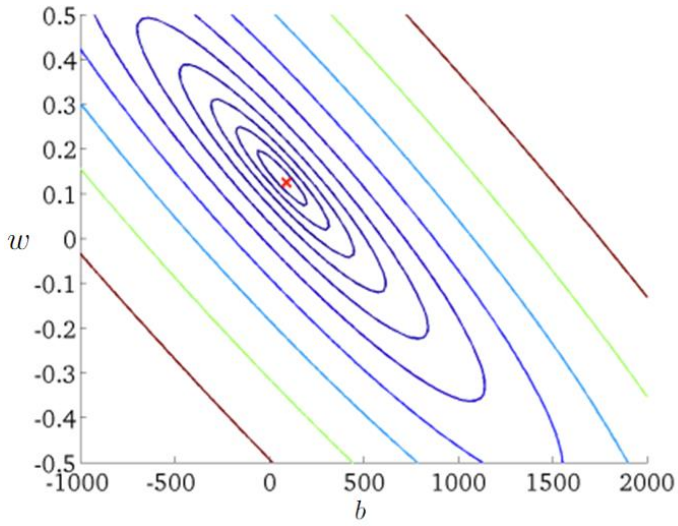
Slika 2.10. *Smisao funkcije gubitka na jednostavnom regresijskom problemu*[13]

Na slici 2.10. prikazan je jednostavan model linearne regresije. Vrijednosti modela $f(x^{(i)})$ uspoređuju se s vrijednostima y. Takav sustav može se opisati matematičkom funkcijom MSE. Model će dati najmanju vrijednost funkcije gubitka kada je suma udaljenosti po cijelom skupu podataka $f(x^{(i)})$ i $y^{(i)}$ najmanja. Ovisnost funkcije gubitka $J(w,b)$ o parametrima w i b govori za koje vrijednosti w i b funkcija poprima minimum to jest daje najmanju vrijednost funkcije gubitka(slika. 2.11).



Slika 2.11. *Prikaz MSE s dva parametra* [13]

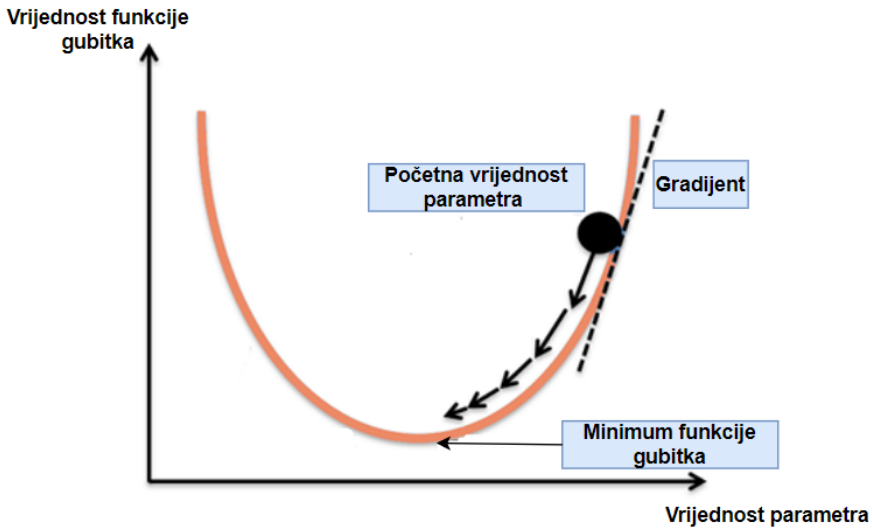
Funkciju gubitka s dva parametra moguće je prikazati i u 3D prostoru. Alternativni pristup vizualizacije funkcije gubitka je korištenje elipsi u 2D prostoru gdje svaka elipsa predstavlja jednaku vrijednost funkcije gubitka (slika 2.12.).



Slika 2.12. *Prikaz 2D funkcije gubitka s dva parametra* [13]

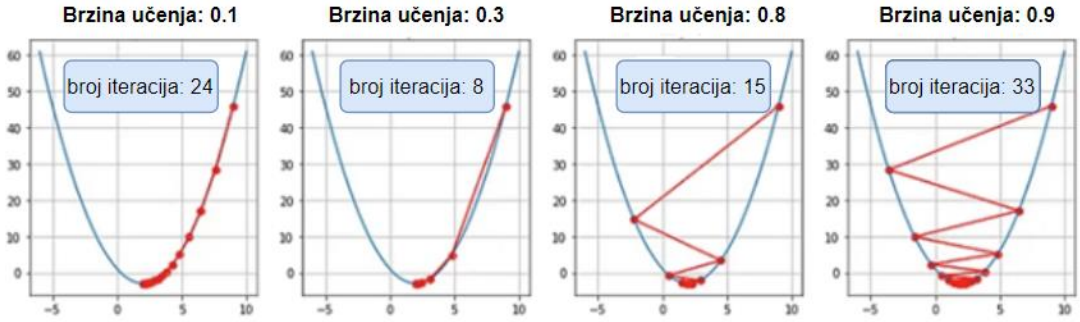
2.5.3. Metoda gradijentnog spusta

Metoda gradijentnog spusta (engl. *Gradient Descent*, GD) je iterativni optimizacijski algoritam prvog reda koji se koristi za pronalazak lokalnog minimuma/maksimuma zadane funkcije. Na slici 2.13 prikazan je proces gradijentnog spusta. Ova se metoda često koristi u strojnom učenju i dubokom učenju (engl. *Deep Learning*, DL) kako bi se smanjila vrijednost funkcije gubitka. Algoritam gradijentnog spusta ne djeluje za sve funkcije, već postoje dva specifična uvjeta. Funkcija mora biti diferencijabilna i konveksna. Ako je funkcija diferencijabilna, ona ima derivaciju za svaku točku u svojoj domeni. Sljedeći zahtjev je da funkcija mora biti konveksna, odnosno da je svaki lokalni minimum ujedno i globalni.[18] Gradijentni spust smanjuje trenutne vrijednosti parametara za derivacijski član (derivacija funkcije gubitka po danom parametru) koji je pomnožen s brzinom učenja.[18]



Slika 2.13. *Prikaz optimiranja postupkom gradijentnog spusta* [19]

Brzina učenja jedan je od hiperparametara modela koji utječu na proces treniranja u neuronskim mrežama. Ukoliko je brzina učenja prevelika sustav nestabilno uči i često ne uspijeva konvergirati u minimum funkcije. S druge strane, ako je brzina učenja premala sustavu je potrebno puno više epoha da dođe u minimum (slika 2.14).



Slika 2.14. *Različite brzine učenja te njihov utjecaj na stabilnost sustava* [18]

U dubokom učenju postoji puno više od jednog parametra modela te vrlo često funkcija gubitka nije konveksna. Stoga se u takvim sustavima često dogodi da izračunato rješenje ne daje globalni minimum funkcije gubitka. Taj problem se nastoji riješiti postavljanjem određenih početnih parametara te raznim unaprijeđenim tehnikama optimizacije.[18]

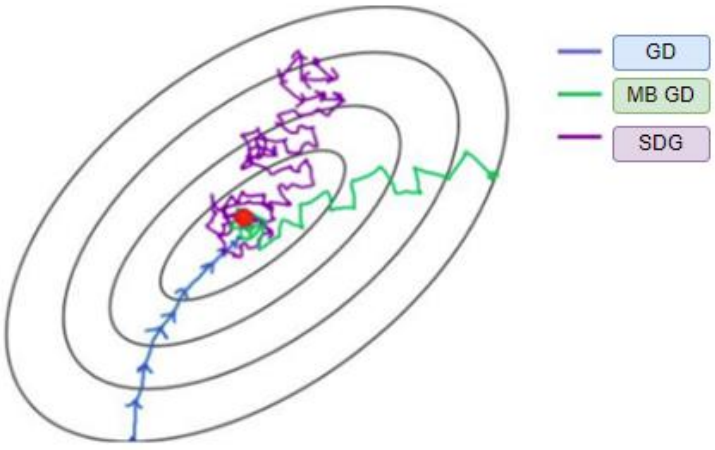
2.5.4. *Mini-grupa gradijent*

Ukoliko neuronska mreža mora obraditi ogromni skup podataka, računalo će provesti znatno vrijeme izvodeći izračune. Kako bi se ubrzao proces gradijentnog spusta, umjesto korištenja *m* podataka (cijeli skup podataka), može se koristiti manji uzorak podataka. Ako je veličina uzorka jednaka 1, tada govorimo o stohastičkom gradijentnom spustu (engl. *Stohastic Gradient Descent*, SGD), a ako je veličina uzorka *k*, uz uvjet *k* manje od *m*, tada govorimo o mini-grupi (engl. *Mini Batch*, MB) (slika 2.15).

$$J = \underbrace{\sum_{i=1}^{m} (h(x)-y)^2}_{\text{GD}} = \underbrace{\sum_{i=1}^{k} (h(x)-y)^2}_{\text{MB}} = \underbrace{(h(x)-y)}_{\text{SGD}}$$

Slika 2.15. *Razlike između izračuna funkcije gubitka pri različitim grupama*[20]

Bitno je istaknuti da iako se smanjenjem veličine uzorka ubrzava proces izračuna, narušava se stabilnost postizanja minimuma, odnosno, prisutan je dodatan šum. Stoga se pravo rješenje problema nalazi u korištenju mini-grupa (1 < veličina uzorka < *m*). Gradijentni spust, stohastički gradijentni spust i mini-grupa predstavljaju iste algoritme, s razlikom da se kod GD-a analizira cijela populacija, kod mini-grupe odabrani dio populacije, a kod SGD-a pojedinac.[21] Na slici 2.16 prikazana je konvergencija različitih grupa u procesu učenja.



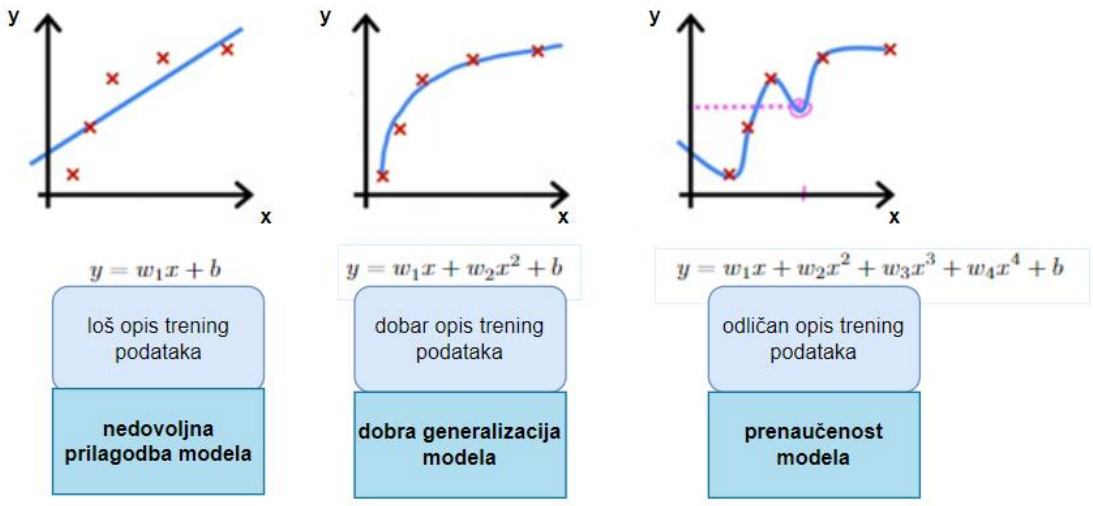
Slika 2.16. *Konvergencija pri različitim grupama* [20]

2.5.5. Adam optimizacijski algoritam

Adam optimizacijski algoritam (engl. *Adaptive Moment Estimation*) predstavlja naprednu tehniku optimizacije koja kombinira koncepte RMSprop i stohastičkog gradijentnog spusta s momentom. Adam algoritam je pogodan za implementaciju u razne modele koji koriste velike skupove podataka i puno parametara. U pogledu računalnih resursa, Adam zahtijeva manje memorije i vrlo je računalno učinkovit. Osim toga, algoritam je pogodan za gradijente podložne fluktuacijama i šumovima te koji sadrže mnogo nula ili blizu-nula vrijednosti. Adam optimizacijski algoritam obično zahtijeva vrlo malo podešavanja. Adam metoda zadaje zasebne adaptivne stope učenja za svaki parametar, te koristi jedinstvenu stopu učenja za sva ažuriranja težinskih faktora koja se ne mijenjaju tijekom treninga.[22]

2.5.6. Regularizacija

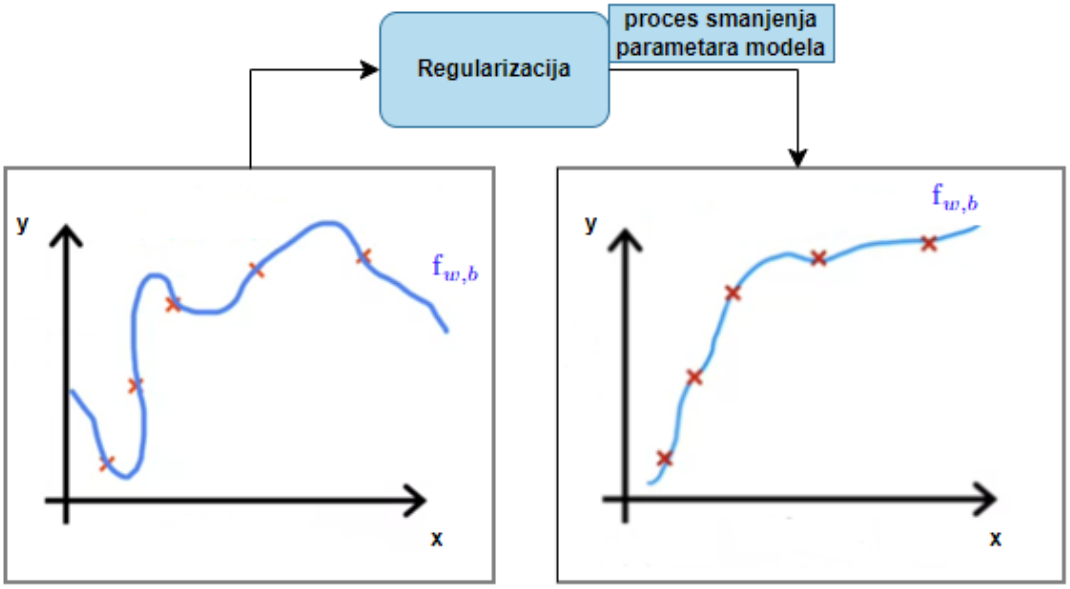
Regularizacija je tehnika koja se koristi za sprječavanje prenaučenosti ili pretreniranja (engl. *overfitting*) modela, poboljšavajući njegovu generalizacijsku sposobnost i osiguravajući stabilnost i bolje vladanje modela na novim neovisnim podacima. Problem prenaučenosti i nedovoljne prilagodbe (engl. *underfitting*) modela predstavljaju jedan od većih izazova u strojnom učenju. Cilj svakog modela je dobra prilagodba novim podacima. Kako bi se riješio ovaj problem, moguće je prikupiti više podataka za razvoj modela ili prilagoditi broj značajki. Ako nije moguće mijenjati broj podataka i značajki, prenaučenost modela se može riješiti procesom regularizacije, dok se nedovoljna prilagodba obično rješava povećanjem broja parametara modela. Na slici 2.17. podatci su opisani pomoću različitih jednadžbi gdje je vidljivo da će složeni modeli uvijek najbolje opisati podatke na skupu za treniranje, dok isto ne mora vrijediti i za podatke na skupu za testiranje.



Slika 2.17. *Primjeri dobre i loše generalizacije modela* [13]

Uvođenjem regularizacijskog člana $\left(\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2\right)$ u funkciju gubitka $\left(\frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right)-y^{(i)}\right)^2\right)$ omogućuje se uklanjanje nepotrebnih značajki ili smanjenje njihovih vrijednosti na minimum (jednadžba (1)). U funkciji gubitka definiran je novi parametar *lambda*, koji pomaže smanjenju prenaučenosti modela. Parametar *lambda* ima vrijednosti veće od nule. Funkcija gubitka može se podijeliti na dva dijela. Prvi se dio odnosi na prilagodbu podataka, dok drugi dio služi za sprečavanje prenaučenosti podataka. Ako je vrijednost parametra *lambda* prevelika, može doći do nedovoljne prilagodbe podataka, tj. regularizacijski član će znatno smanjiti parametre modela, zbog čega će model biti prejednostavan. S druge strane, ako je *lambda* jako mala, neće se spriječiti prenaučenost podataka, stoga se ovaj postupak u praksi često provodi iterativno, ispitujući različite vrijednosti parametra *lambda* (jednadžba 1).[23]

$$J\left(w,b\right)=\frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}\left(x^{(i)}\right)-y^{(i)}\right)^2+\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2 \qquad (1)$$



Slika 2.18. *Regularizacija modela* [13]

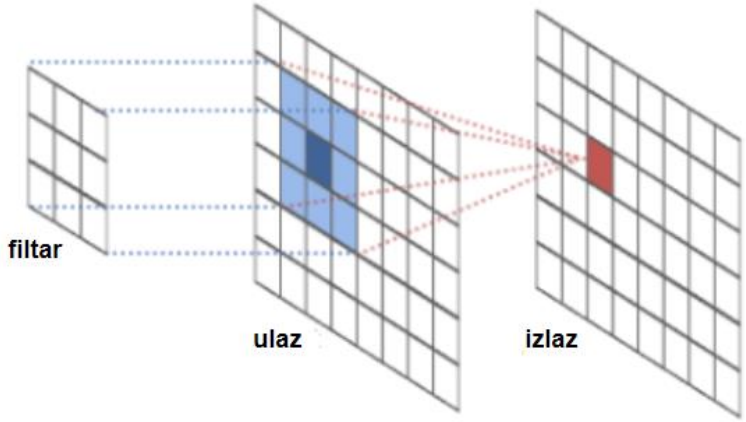
2.6. Konvolucijske mreže

Konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*, CNN) predstavljaju vrstu dubokih neuronskih mreža specijaliziranih za obradu slika ili vremenskih nizova. Operacija konvolucije kombinira dva signala ili dvije funkcije kako bi se dobila treća funkcija koja predstavlja njihovu zajedničku lokalnu interakciju. Operacija konvolucije uključuje klizanje filtra preko ulaznog signala, korak po korak, te izračunavanje elementarnog umnoška između pripadajućih elemenata filtra i ulazne slike. U kontekstu obrade slika ili signala, konvolucija se koristi za izdvajanje značajki iz ulaznih podataka.[15]

2.6.1. Konvolucija

Konvolucijski sloj predstavlja temeljnu komponentu konvolucijskih neuronskih mreža i odgovoran je za većinu transformacija podataka. Često se naziva i skraćeno, CONV sloj. Tipično se sastoji od dva dijela: konvolucije ulazne aktivacijske mape (slika ili signal) s određenim brojem filtara za prepoznavanje različitih značajki, te primjene aktivacijske funkcije na dobiveni rezultat. Iskustveno se ReLU aktivacijska funkcija pokazala najučinkovitijom u zadacima u kojima se primjenjivala konvolucijska mreža. Svaki konvolucijski sloj sadrži unaprijed odabrani broj filtara fiksne dimenzije koji pohranjuju vrijednosti težina *W*. Konvolucija filtara s mapom

značajki odvija se samo na prostornim koordinatama širine i visine mape značajki. U slučaju slika u boji, za svaki kanal boje obično se primjenjuje isti filtar, a rezultati se zbrajaju po kanalima mape značajki. Na slici 2.19. prikazan je proces konvolucije.



Slika 2.19. *Konvolucije za jedan kanal* [20]

Konvolucijski slojevi koji su bliže početku mreže uče prepoznati detalje i strukture, poput rubova objekata na slikama. Što se konvolucijski sloj nalazi dublje u neuronskoj mreži, on uči prepoznavati sve kompleksnije značajke. Vrijednosti težina pohranjene u filtrima predstavljaju parametre modela koje mreža može naučiti tijekom procesa učenja. Hiperparametri modela koji se definiraju u konvolucijskom sloju uključuju broj filtara (*d*), veličinu filtra (*f*), dopunjavanje (*p*) i korak (*s*). Veličina filtra u 1D konvolucijama definira se dužinom filtra, a u 2D konvolucijama definirana je visinom i širinom matrice filtra. Težine svih filtara u konvolucijskom sloju inicijalizirane su na proizvoljne vrijednosti prije početka učenja mreže.[15]

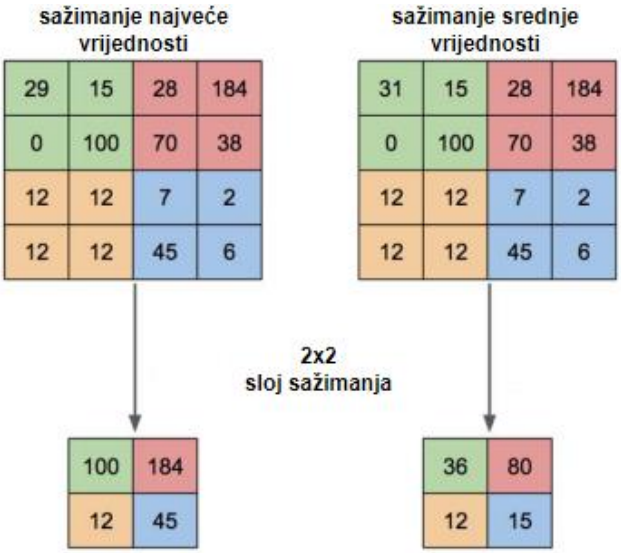
2.6.2. *Korak*

Korak (engl. *stride*) označava broj piksela koje filtar prelazi u horizontalnom i vertikalnom smjeru tijekom konvolucije aktivacijske mape.

2.6.3. *Sloj sažimanja*

Prilikom kreiranja konvolucijske neuronske mreže, postoje tri temeljne faze. U prvoj fazi, paralelne konvolucije stvaraju aktivacijske mape. U drugoj fazi, svaka aktivacijska mapa prolazi kroz proces nelinearizacije pomoću neke od aktivacijskih funkcija. Ova faza se ponekad naziva i faza detekcije. U trećoj fazi, koriste se sažimajuće funkcije kako bi se prilagodio aktivacijski sloj. Sloj sažimanja (engl. *Pooling layer*) uključuje primjenu funkcija sažimanja na aktivacijske mape iz

prethodnog sloja s ciljem smanjenja njihovih dimenzija. Operacija sažimanja na ulaznoj slici obično primjenjuje agregirane statističke podatke na lokalnom području definiranom veličinom filtra. Dva najčešća tipa sažimanja su sažimanje odabirom najveće vrijednosti (engl. *max-pooling*) i sažimanje prosječnom vrijednošću (engl. *average-pooling*) (slika 2.20.).[24]



Slika 2.20. *Prikaz sažimanja najveće i srednje vrijednosti* [20]

2.6.4. Dopuna

Dopuna (engl. *padding*) se odnosi na dodavanje nula oko granica izvorne mape značajki. Ovaj postupak povećava prostorni obujam mape, ali donosi i neka korisna svojstva. Na primjer, elementi na rubovima mape sada imaju veći utjecaj na prijenos informacija, jer sudjeluju više puta u konvoluciji s filtrom. Osim toga, dopunom se mogu kontrolirati dimenzije izlazne aktivacijske mape. Dvije vrsta dopuna s obzirom na dimenzije izlazne mape značajki su valjana dopuna i nepromjenjiva dopuna. U valjanoj dopuni (engl. *valid padding*) mapa značajki se ne dopunjuje nulama, što rezultira promjenom dimenzija izlazne aktivacijske mape. Dok u nepromijenjenoj dopuni (engl. *same padding*) mapa značajki se dopunjuje nulama oko rubova na način koji osigurava očuvanje dimenzija izlazne aktivacijske mape nakon konvolucije.[15]

2.7. Vrednovanje kvalitete modela

Vrednovanje kvalitete modela ključni je korak u procesu izvedbe i primjene modela. Vrednovanje modela omogućava procjenu točnosti, preciznosti i opće prikladnosti modela u objašnjavanju podataka. Vrednovanje modela uključuje analizu različitih statističkih pokazatelja, kao što su koeficijent determinacije (R^2), srednja kvadratna greška (MSE) i mnogih drugih. Ovi pokazatelji pokazuju koliko model adekvatno prati trendove i obrasce u podacima.

2.7.1 *Koeficijent determinacije*

Koeficijent determinacije, R^2 predstavlja statistički koncept koji se koristi u analizi varijance i regresijskoj analizi. Njime se mjeri postotak objašnjene varijance u podacima. Veća vrijednost R^2 ukazuje na bolju prilagođenost podacima. Primjenom modela s prediktivnim varijablama na podatke koji sadrže vrijednosti $y_1,...,y_n$ i odgovarajuće izlazne vrijednosti, R^2 definiran je jednadžbom (2) kao razlika između varijance podataka i varijance greške podijeljena s varijancom podataka. [25]

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2} \tag{2}$$

3. MATERIJALI I METODE

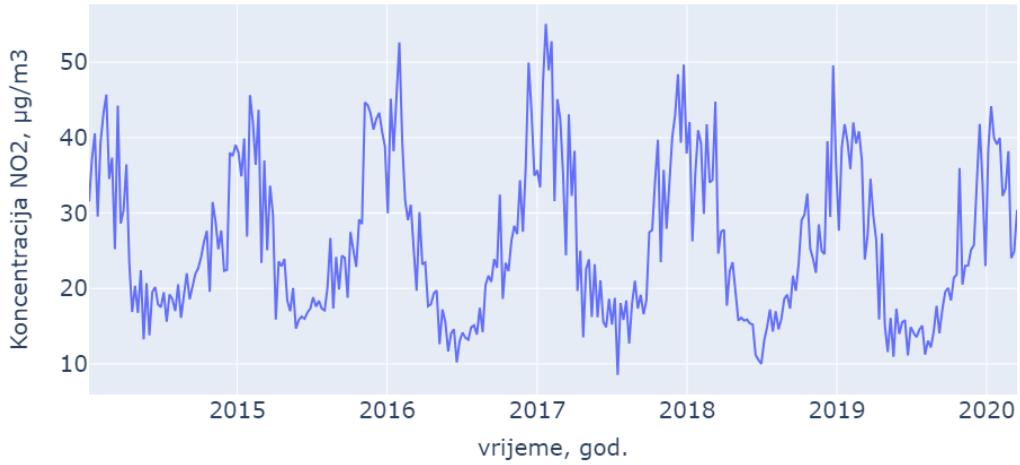
3.1. Prikupljanje podataka sa mjernih postaja u Grazu

Graz je drugi po veličini grad u Austriji s obzirom na stanovništvo i površinu s približno 280800 stanovnika. Grad se prostire na površini od oko 127 km² i smješten je u predalpskom prostoru, na sjevernom rubu Gradačkog polja. Kroz Graz protječe rijeka Mura, koja je u prošlosti imala značajnu ulogu u gospodarstvu kao prometnica. Graz se razvio uz važnu prometnicu koja povezuje talijanski i panonski prostor preko rijeke Mure. Ova prometnica križa se s cestom koja povezuje njemačke i hrvatske krajeve. Zbog svog izvrsnog položaja, Graz je od ranog razdoblja postao središte trgovine i obrta. Danas, u skladu s tom tradicijom, Graz je poznat po velesajmovima koji se održavaju svakog proljeća i jeseni.[26] Osim toga, grad predstavlja industrijsko i gospodarsko središte što utječe na kvalitetu zraka. Kako bi se stekao jasniji uvid kvalitete zraka, dugoročni podaci mjerenja u razdoblju od 1. siječnja 2014. do 15. ožujka 2020. analizirani su na temelju mjerne postaje Zapad smještene u Grazu. Primarni cilj ove analize je procjena vrijednosti koncentracija NO_2, uzimajući u obzir utjecaj drugih promatranih varijabli koje odražavaju kvalitetu zraka u gradu Grazu. Na slici 3.1. prikazana je karta grada s mjernom postajom Zapad.

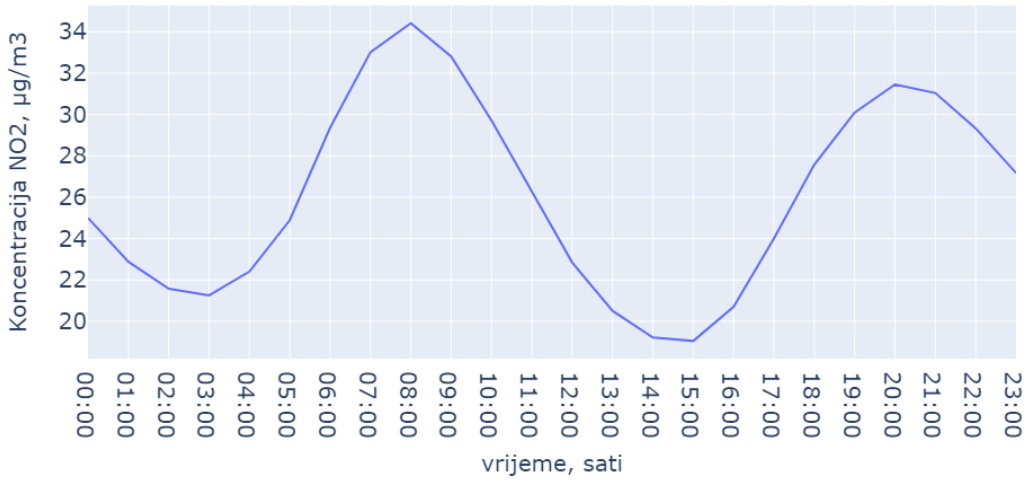
Slika 3.1. *Karta grada s mjernom postajom Zapad*

3.2. Koncentracije dušikovih oksida

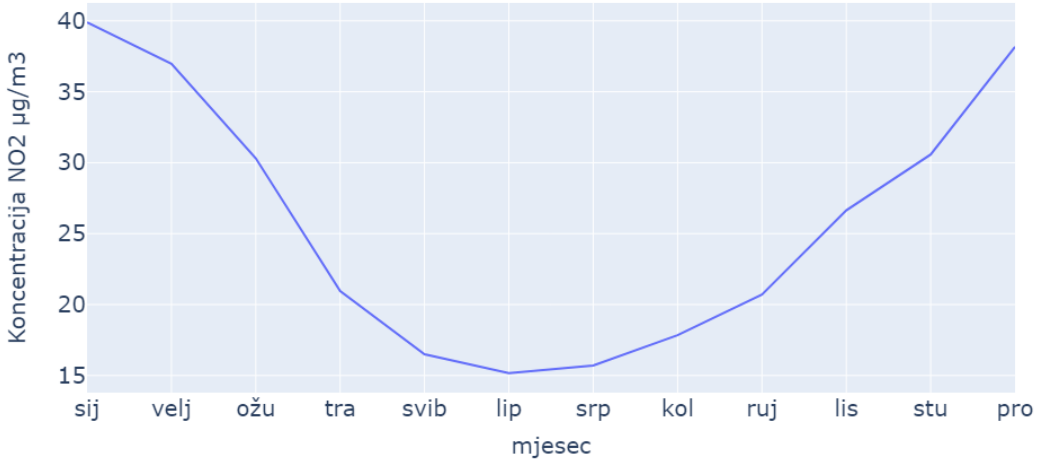
Na slici 3.2. prikazane su koncentracije NO_2 izražene u mikrogramima po kubnom metru ($\mu g/m^3$) u razdoblju od 01.01.2014. do 15.03.2020. na mjernoj postaji Zapad. Iz slika 3.2. i 3.4. je vidljivo kako postoji sezonska varijabilnost, što znači da razine NO_2 fluktuiraju ovisno o godišnjem dobu. Za vrijeme zime, razine NO_2 su više, 7dok su za vrijeme ljeta zabilježene niže vrijednosti. Na promatranom periodu primjećuje se da su tijekom 2017. godine koncentracije NO_2 najviše i najniže. Prema slici 3.3., može se primijetiti da se razine NO_2 povećavaju tijekom jutarnjih i večernjih prometnih špica, odnosno ujutro između 6 i 10 sati te navečer između 18 i 22 sata. Na slici 3.4. prikazane su koncentracije NO_2 u pojedinome mjesecu. Koncentracije NO_2 najviše su za vrijeme zime dok su najniže u ljeto.



Slika 3.2. *Srednje koncentracije NO_2 mjerene na tijekom promatranog razdoblja izražene na tjednoj osnovi*



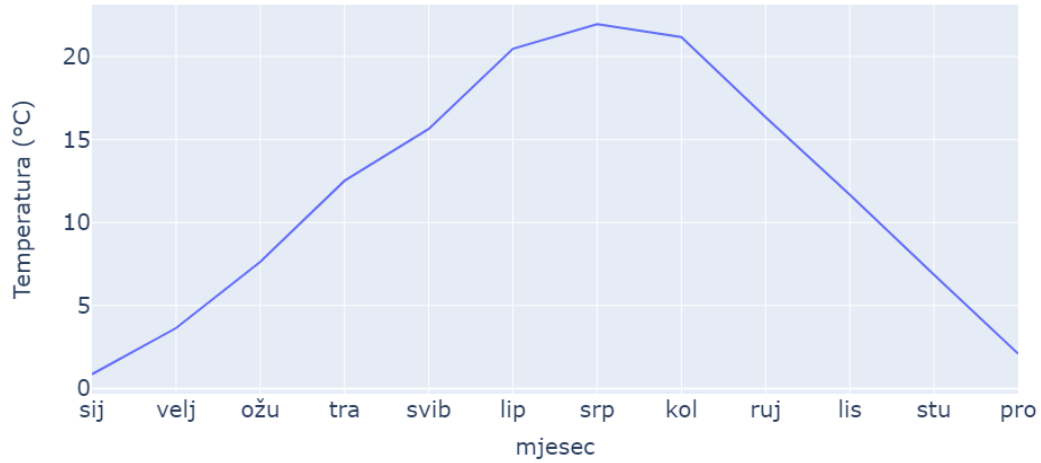
Slika 3.3. *Prikaz prosječnih koncentracija NO_2 po satu*



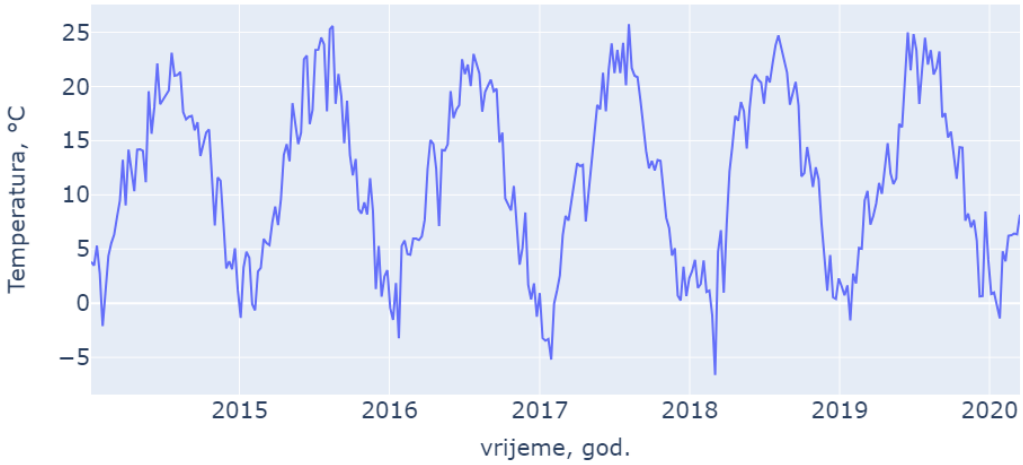
Slika 3.4. *Prikaz prosječnih koncentracija NO_2 po pojedinome mjesecu*

3.3. Temperatura zraka

Temperatura zraka predstavlja temperaturu prizemnog sloja atmosfere mjerenu na 2m visine, kako bi se izbjegao utjecaj toplinskog zračenja tla i okoline ili izravno zračenje Sunca. Ova temperatura ovisi o dobu dana i godini, s dnevnim promjenama koje ovise o količini naoblake, vjetru i poremećajima koji se javljaju tijekom dana. Godišnji tijek temperature zraka ovisi o položaju Zemlje u odnosu na Sunce i klimatskim promjenama. U Grazu je najhladniji mjesec siječanj, a najtopliji srpanj.[27] Analiza prikupljenih podataka potvrđuje tezu da je siječanj najhladniji mjesec godine, a srpanj najtopliji (slika 3.5.). Na slici 3.6 vidljivo je da su 2016. godine, temperature ljeti bile najniže. Također, na slici 3.6. se primjećuje godišnja cikličnost temperature sa najnižom temperaturom u zimi 2018. godine.



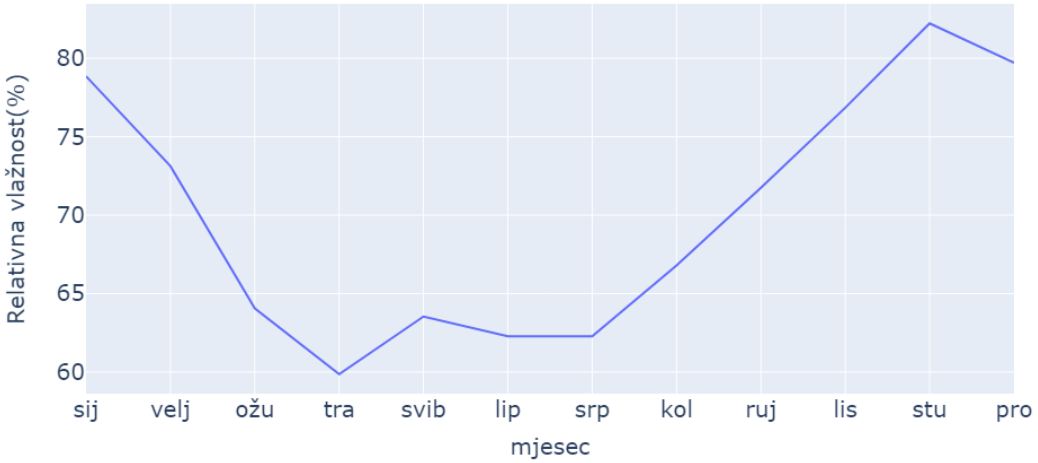
Slika 3.5. *Prosječne temperature u Grazu za svaki mjesec u godini*



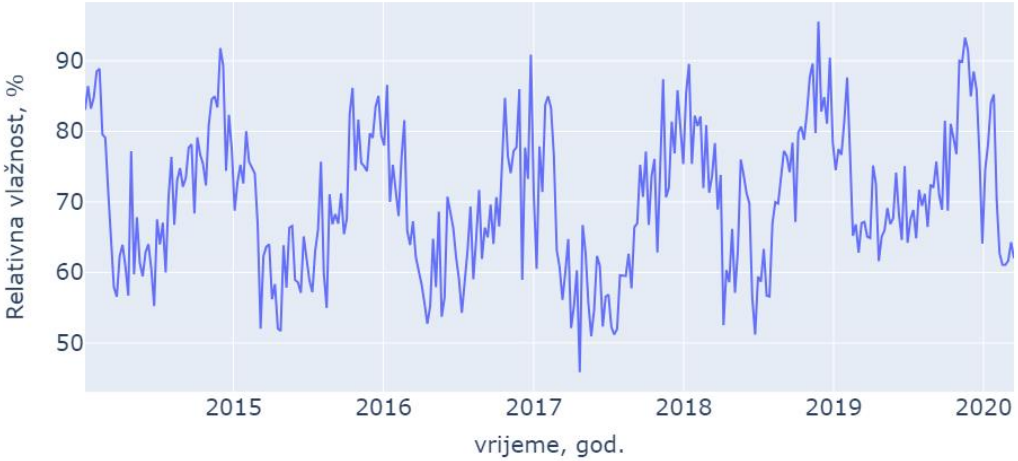
Slika 3.6. *Srednje vrijednosti tjednih temperatura*

3.4. Relativna vlažnost zraka

Relativna vlažnost zraka je fizikalni parametar koji opisuje količinu vodene pare prisutne u zraku u odnosu na maksimalnu količinu vodene pare koju zrak može zadržati pri određenoj temperaturi i tlaku. Relativna vlažnost zraka izražava se kao omjer parcijalnog tlaka vodene pare i parcijalnog tlaka zasićene vodene pare pri istim uvjetima. U istraživanju su prikupljeni podaci o relativnoj vlažnosti zraka prikazani u postotcima. Relativna vlažnost zraka je viša tijekom zimskih mjeseci, s najvišim vrijednostima u studenom, dok je najniža relativna vlažnost zraka zabilježena tijekom proljeća, u travnju (slika 3.7.). Tjedne vrijednosti relativne vlažnosti zraka prikazane su na slici 3.8. Minimalna vrijednost prosječne tjedne relativne vlažnosti je oko 45 % dok je maksimalna oko 90% .



Slika 3.7. *Prosječne vrijednosti relativne vlažnosti po mjesecima*

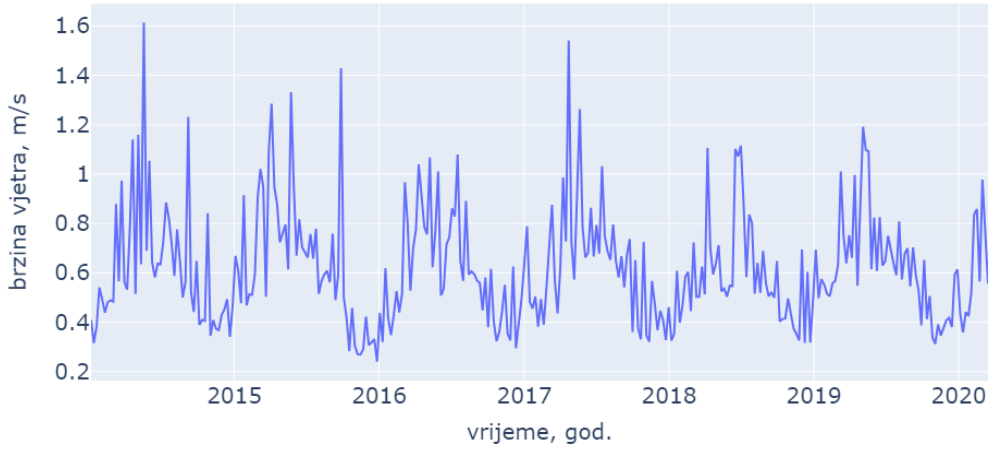


Slika 3.8. *Prosječne tjedne vrijednosti relativne vlažnosti*

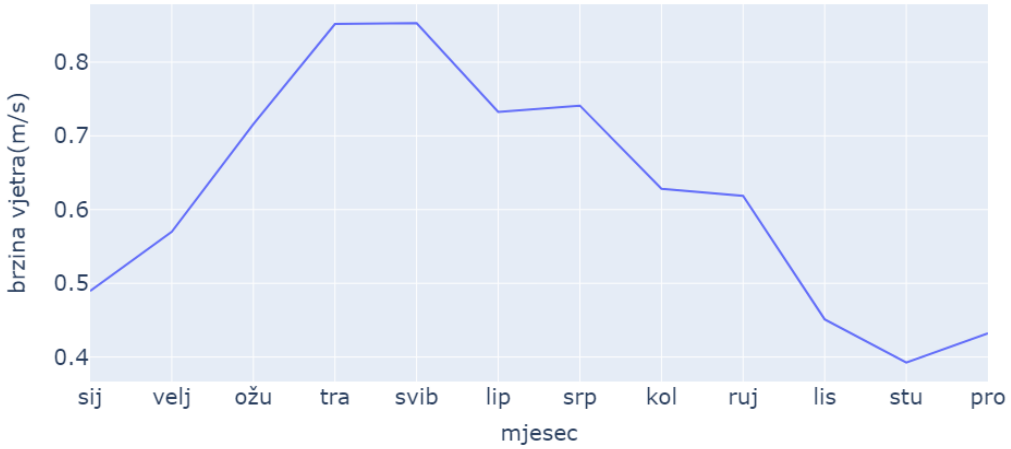
3.5. Vjetar

Vjetar nastaje uslijed složene interakcije nekoliko čimbenika koji utječu na njegovu brzinu i smjer. Jedan od tih čimbenika su razlike u atmosferskom tlaku između područja, što rezultira kretanjem zraka s višeg tlaka prema nižem tlaku. Brzina vjetra može se povećati s većom razlikom u tlaku. Zemljina rotacija također ima utjecaj na vjetar, mijenjajući smjer i brzinu vjetra na različitim geografskim područjima. Coriolisova sila, koja proizlazi iz Zemljine rotacije, dodatno mijenja smjer vjetra tako da skreće udesno na sjevernoj polutki i ulijevo na južnoj polutki. Kada su putanje čestica zraka zakrivljene, centrifugalna sila utječe na vjetar, mijenjajući njegovu brzinu i smjer. S druge strane, sila trenja s tla može usporiti vjetar i uzrokovati promjene u smjeru dok se kreće blizu tla. Uzimajući u obzir sve ove sile,

može se razumjeti zašto vjetar pokazuje složenost i varijabilnost na različitim mjestima i uvjetima.[28] Podaci o orijentaciji vjetra prikazani su u stupnjevima. Graz nije podložan snažnim vjetrovima. Brzina vjetra prikazana je kroz tjedni prosjek, što je ilustrirano na slici 3.9. Na njoj se može vidjeti kako je najveća brzina vjetra zabilježena 2014. godine. Također vidljivo je da brzina vjetra tijekom godina uglavnom ne pokazuje značajne promjene, iako se povremeno javljaju jači udari vjetra. Na slici 3.10. je vidljivo da je brzina vjetra u zimskim danima najmanja.



Slika 3.9. *Prosječne tjedne brzine vjetra*



Slika 3.10. *Prosječne brzine vjetrova za pojedini mjesec*

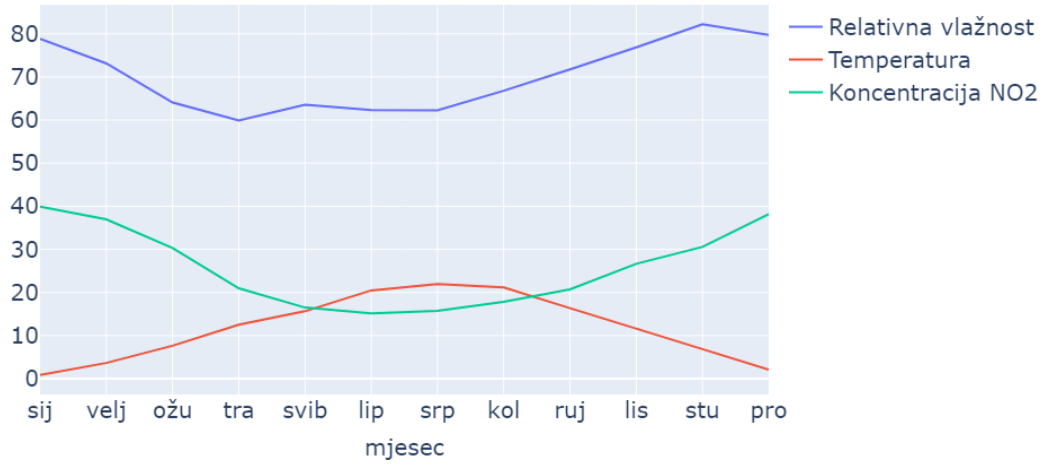
3.6. Temporalni podatci

U radu Lovrić et al. [3] naglašava se značaj vremenskih čimbenika poput dana u tjednu, godišnjih doba i praznika na koncentraciju dušikovih oksida u atmosferi. Slijedom toga, ovim istraživanjem predviđeno je da će se ti čimbenici temeljito razmatrati prilikom razrade modela. Pretpostavka je da vremenski elementi poput

godine, dana u godini, mjeseca, dana u tjednu, godišnjih doba, školskih praznika i nacionalnih praznika mogu utjecati na koncentraciju dušikovih oksida. Integriranjem podataka o ovim varijablama s drugim relevantnim informacijama, poput meteoroloških uvjeta i geografskog položaja, znanstvenici će dobiti sveobuhvatnu sliku različitih čimbenika koji utječu na kvalitetu zraka. Ovaj pristup omogućuje preciznije analize i omogućuje izvlačenje informiranih zaključaka o vezama između vremenskih čimbenika i koncentracija dušikovih oksida, čime se doprinosi boljem razumijevanju i upravljanju problemom onečišćenja zraka.

3.7. Ovisnost koncentracije NO_2 o vremenskim uvjetima

Promjene u koncentraciji dušikova dioksida tijekom godine su česte, a one su uglavnom uvjetovane fluktuacijama u meteorološkim uvjetima. Grafikoni sa Zapadne postaje daju uvid u općenitu vezu između meteorologije i koncentracije NO_2. Slika 3.11. ilustrira utjecaj temperature na koncentraciju NO_2, pri čemu se može uočiti inverzna veza između ove dvije varijable. Hladniji mjeseci su obilježeni s višim koncentracijama NO_2, dok su koncentracije NO_2 tijekom toplijih mjeseci niže. Nadalje, zapaženo je da se koncentracija NO_2 povećava s porastom relativne vlažnosti.



Slika 3.11. *Ovisnost relativne koncentracije NO_2 o temperaturi i relativnoj vlažnosti*

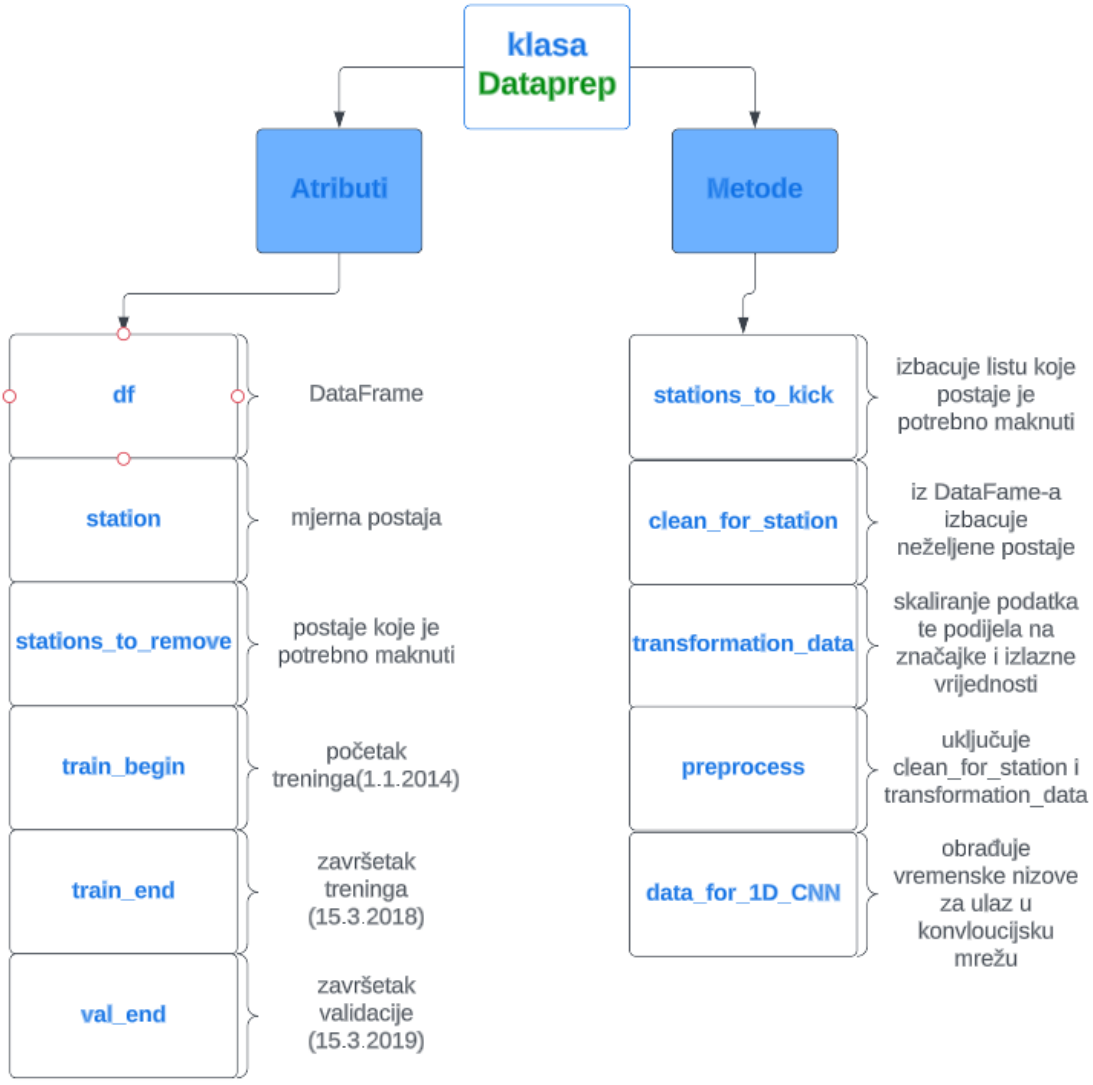
3.8. Alati za obradu podataka i izradu modela

Python je iznimno fleksibilan, dinamički programski jezik, poznat po svojoj primjeni u analizi podataka, zahvaljujući bibliotekama poput Pandas, NumPy i Matplotlib. Matplotlib je biblioteka koja se koristi za vizualizaciju podataka. Matplotlib omogućuje prikaz grafikona, histograma, spektara, te druge vrste vizualizacija. NumPy je biblioteka za numeričke proračune koja sadrži brojne funkcije za statistiku, linearnu algebru, Fourierove transformacije, generiranje nasumičnih brojeva. Pandas je biblioteka za obradu i analizu podataka u kojoj se radi sa "DataFrame-ovima", koji su u suštini tablice podataka. Sa DataFrame-ovima se može lako manipulirati, filtrirati, sortirati, grupirati itd. Pandas sadrži mnogo funkcija za rad sa datumima, rad sa nedostajućim podacima, za povezivanje tablica, čitanje i pisanje podataka u različitim formatima (CSV, Excel, SQL baze podataka, itd.). Pandas je vrlo moćan alat za analizu podataka i koristi se u kombinaciji sa Matplotlib i NumPy bibliotekama u mnogim aplikacijama.[29] Osim toga, korištena je Plotly biblioteka za vizualizaciju koja omogućava izradu interaktivnih vizualizacija.[30] U svrhu dubokog učenja korištena je biblioteka PyTorch koja predstavlja snažan alat za razvoj različitih modela strojnog učenja, uključujući između ostalih i konvolucijske neuronske mreže.[31]

4. EKSPERIMENTALNI DIO

4.1. Organizacija i skaliranje podataka

Prije razvoja jednodimenzionalnog konvolucijskog modela potrebna je adekvatna priprema podataka. Priprema podataka provedena je unutar klase radi preglednijeg koda. U objektno orijentiranom programiranju, klasa sadrži skup metoda i varijabli. Objekt je specifična postavka klase; sadrži realne vrijednosti umjesto varijabli. Na slici 4.1. prikazani su atributi i metode unutar klase.



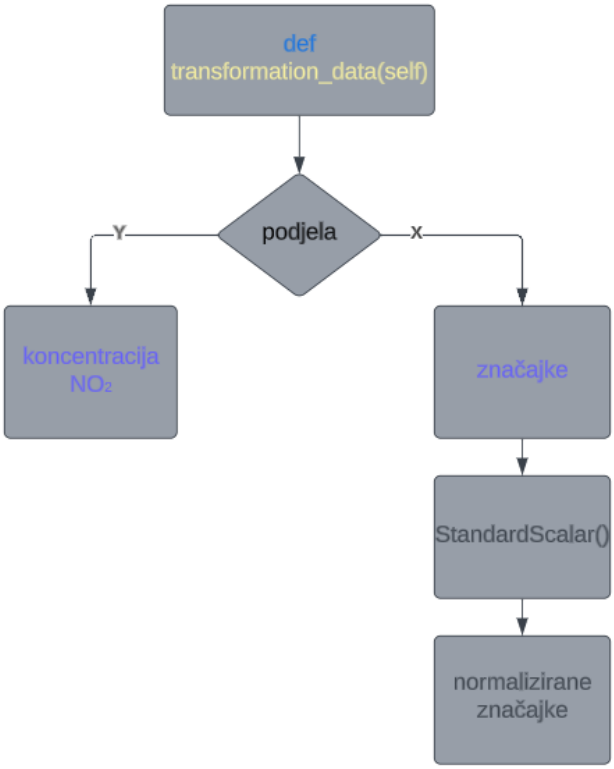
Slika 4.1. *Prikaz metoda i atributa unutar klase za predobradu vremenskog niza za 1D CNN mrežu*

Priprema podataka za razvoj modela se sastoji od sljedećih koraka:

1) **Normalizacija podataka**: Neuronske mreže obično bolje funkcioniraju kada su njihovi ulazni podaci normalizirani, što znači da su podaci skalirani tako da imaju srednju vrijednost 0 i standardnu devijaciju 1. To omogućava brže i stabilnije učenje mreže.
2) **Podjela na skupove**: učenje, validacija i testiranje. Kako bi se procijenilo koliko dobro model uči i generalizira na novim podacima, podatke je potrebno podijeliti na skupove za učenje, validaciju i testiranje.
3) **Formatiranje podataka**: Podaci moraju biti formatirani na način da su kompatibilni s ulaznim slojem CNN-a. U kontekstu 1D CNN-a, to znači da podaci moraju biti strukturirani kao jednodimenzionalni nizovi. Ako se radi s više od jedne značajke, svaka značajka treba biti organizirana kao odvojeni kanal.

4.1.1. Normalizacija podataka

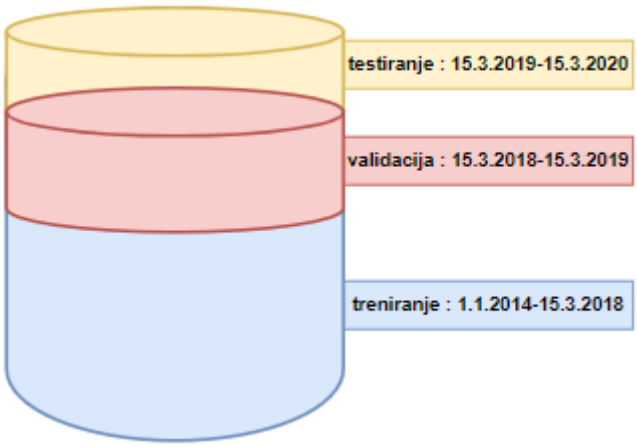
Na slici 4.2. prikazana je metoda za normalizaciju podatka i njene izlazne vrijednosti. Metoda koristi *StandardScaler* iz biblioteke Sklearn za normalizaciju podataka. *StandardScaler* normalizira značajke tako da imaju srednju vrijednost 0 i standardnu devijaciju 1. Ova metoda prvo briše ciljnu varijablu (u ovom slučaju, koncentraciju NO_2) iz skupa podataka. Zatim skalira podatke koristeći *StandardScaler* koji je prethodno inicijaliziran. Skalirani podaci se zatim pretvaraju u tenzore koristeći PyTorch *torch.tensor*. PyTorch tensor je višedimenzionalna struktura podataka slična matrici, ali s većom fleksibilnošću. Tenzori u PyTorch-u mogu imati bilo koju dimenzionalnost, što znači da mogu biti skalari, vektori, matrice ili tenzori viših redova. Ciljna varijabla (koncentracija NO_2) se izvlači iz originalnog skupa podataka i pretvara u tenzor.



Slika 4.2. *Algoritam metode za skaliranje podataka*

4.1.2. Podjela na skupove

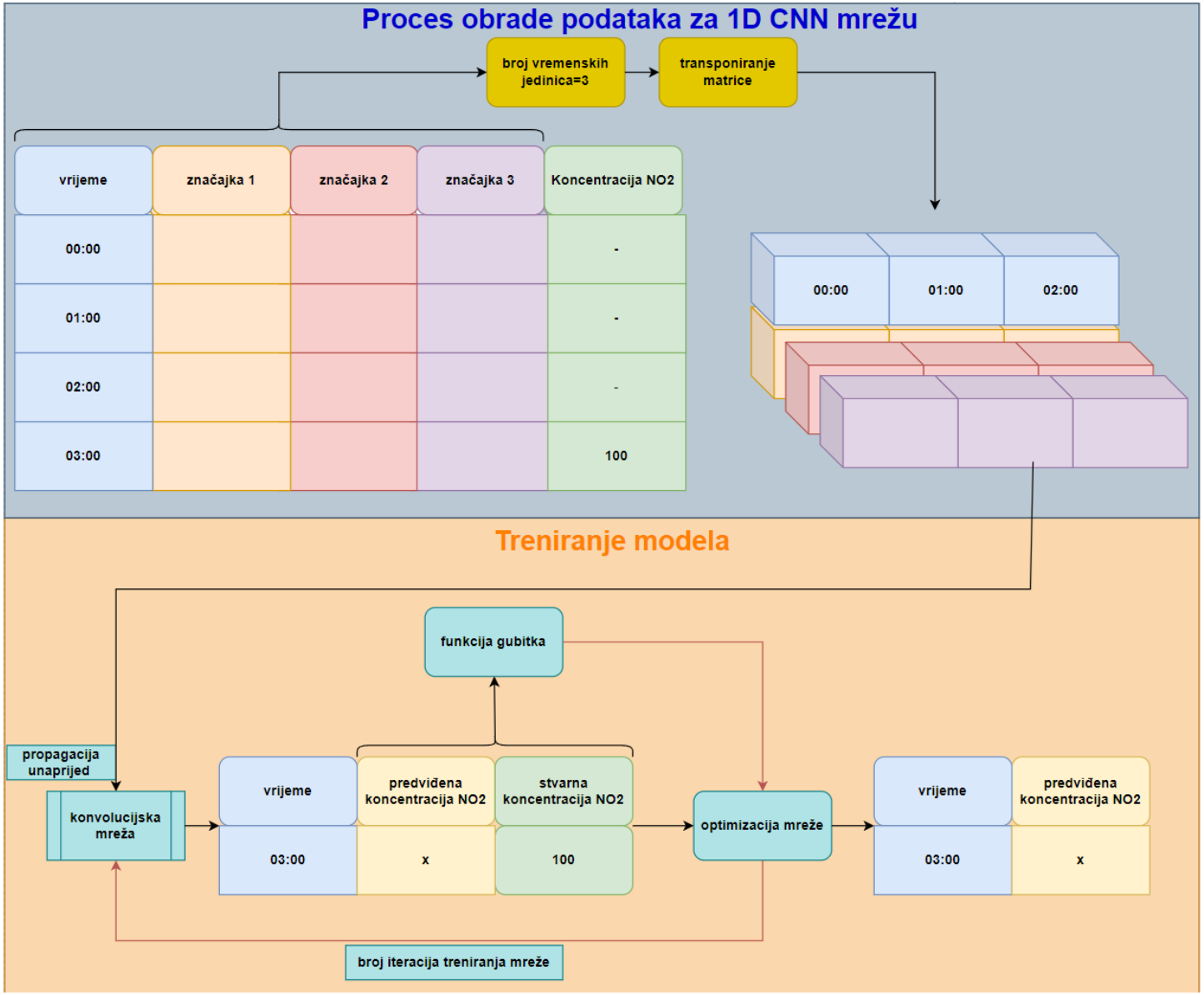
Definirana je podjela podataka na podskupove (slika 4.3.). Skup za treniranje podataka odnosi se na vremenski period od 1.1.2014 do 15.3.2018. Skup za validaciju modela pokriva vremenski period od 15.3.2018 do 15.3.2019, a skup za testiranje podataka odnosi se na period od 15.3.2019 do 15.3.2020.



Slika 4.3. *Prikaz podjele podataka na podskupove*

4.2. Predobrada i treniranje mreže

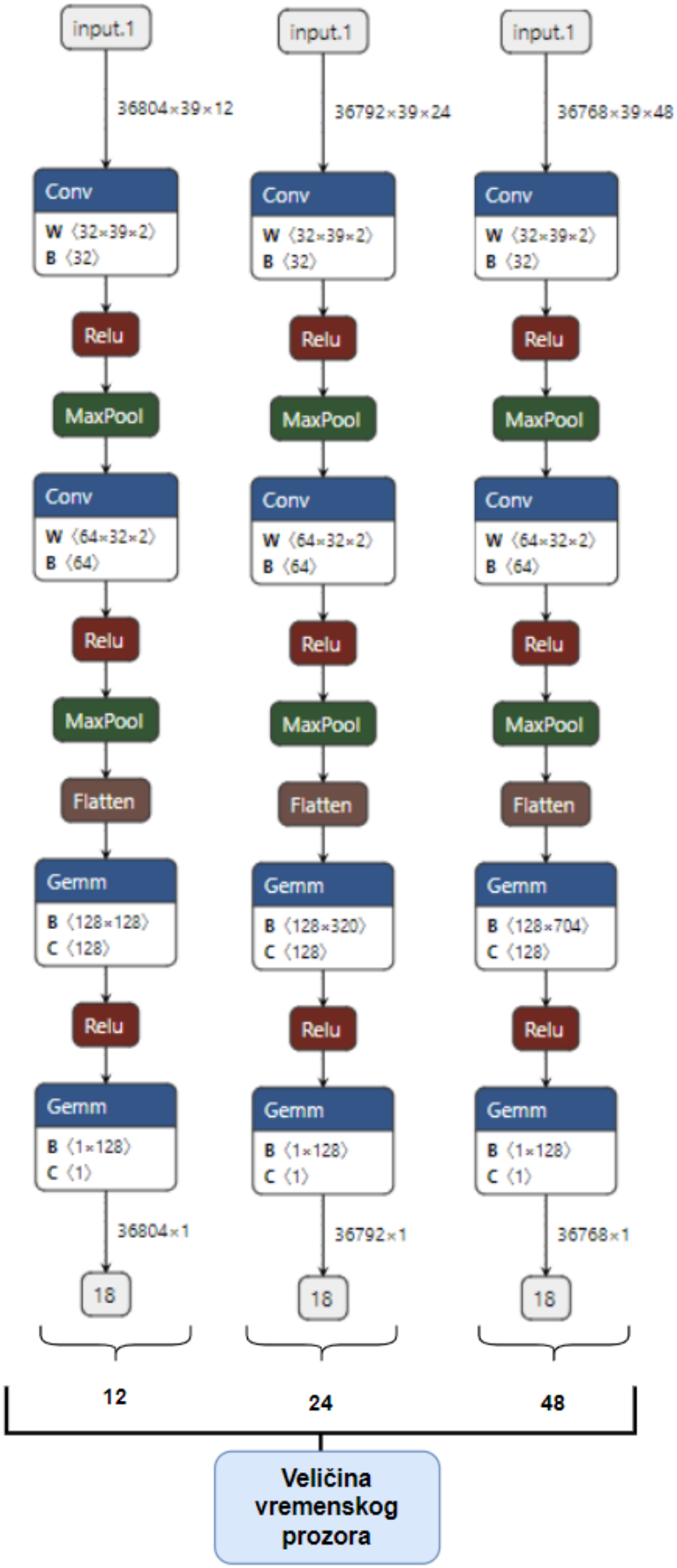
Na slici 4.4. prikazana je obrada podataka za treniranje, validaciju i testiranje 1D konvolucijske neuronske mreže. Podatci se strukturiraju u odgovarajući format za ulaz u 1D CNN. Prvobitno se definiraju skupovi podataka za treniranje, validaciju i testiranje. Slijedeći korak uključuje inicijalizaciju matrica za pohranu podataka. Svaki skup (treniranje, validacija, testiranje) ima svoj par tenzora x i y. Dimenzije x tenzora su broj uzoraka, broj značajki, broj vremenskih jedinica, gdje je broj vremenskih jedinica broj prošlih vremenskih vrijednosti koje pohranjuje u X kao značajke. Za svaki uzorak, tenzor x se puni s brojem prethodnih vremenskih koraka podataka, a tenzor y se puni s odgovarajućom ciljnom vrijednosti koja slijedi nakon vremenskih jedinica. Transpozicija se koristi za pravilno usklađivanje dimenzija podataka. Na slici 4.4. prikazan je proces predobrade podataka za 1D konvolucijsku mrežu sa vremenskom jedinicom 3. Proces je isti za svaki broj vremenskih jedinica gdje sustav uzima različit broj uzoraka iz prošlosti za predviđanje. Nakon predobrade značajki slijedi treniranje konvolucijske mreže. Tijekom svake epohe za svaki podskup podataka prethodno predobrađeni za 1D CNN u skupu za učenje izračunava se predviđanje. Gubitak između stvarne i predviđene vrijednosti se zatim izračunava koristeći funkciju gubitka (MSE). Regularizacija je implementirana dodavanjem apsolutne vrijednosti svih parametara modela, te množeći ukupan zbroj s unaprijed definiranim parametrom (l1_lambda). Nakon što se izračuna ukupan gubitak (uključujući i regularizacijski termin), izvodi se korak optimizacije. Na kraju svake epohe, prosječni gubitak po grupi dodaje se u listu *train_loss*, izračunava se gubitak na validacijskom skupu i mjera R^2 te se ažurira lista *val_loss*. Ako je mjera R^2 bolja od najbolje dosadašnje vrijednosti, model se sprema i brojač strpljivosti (engl. *early stop*) se resetira. U suprotnom, brojač strpljivosti se poveća. Ako brojač strpljivosti dosegne predefiniranu granicu, treniranje se zaustavlja.



Slika 4.4. *Prikaz obrade podataka za 1D CNN mrežu*

4.3. Arhitektura 1D CNN modela

Model je sastavljen od dva konvolucijska sloja (*self.conv1* i *self.conv2*), funkcije maksimalnog sažimanja (*self.pool*) koja smanjuje dimenzionalnost podataka, te funkcije za poravnanje (*self.flatten*) koja transformira multi-dimenzionalni izlaz konvolucijskih slojeva u jednodimenzionalni. Osim toga, definirana su dva potpuno povezana sloja (*self.fc1* i *self.fc2*), kao i sloj za ispuštanje (*self.dropout*) koji se koristi za sprječavanje prenaučenosti mreže. Dimenzionalnost prvog potpuno povezanog sloja varira ovisno o veličini vremenskog prozora, što omogućava fleksibilnost u modeliranju različitih vremenskih razdoblja. Funkcija unaprijed (engl. *forward pass*) definira način na koji se podaci propagiraju kroz mrežu. Aktivacijska funkcija ReLU koristi se nakon svakog konvolucijskog i prvog potpuno povezanog sloja. Sloj za ispuštanje primjenjuje se nakon prvog potpuno povezanog sloja, a izlazni sloj (*self.fc2*) generira konačnu predikciju koncentracije NO_2. Konfiguracije modela za različite veličine vremenskih prozora odstupaju minimalno. Razlog odstupanja je različiti broj vremenskih jedinica koji ulazi u mrežu stoga svaka mreža ima istu dubinu mreže, ali različit broj neurona nakon konvolucije. Na slici 4.5. prikazani su modeli za vremenske prozore 12, 24, 48 gdje GEMM (engl. *General matrix multiply*) označava matrične umnoške.



Slika 4.5. *Prikaz modela za svaki vremenski prozor*

4.4. Odabir hiperparametara i značajki modela

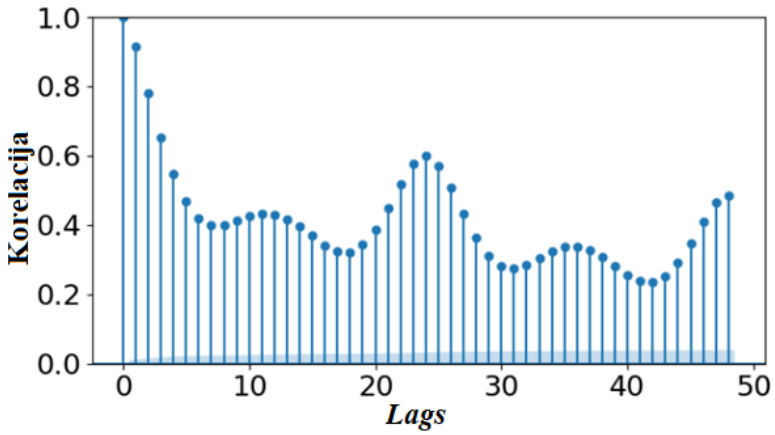
Prvo je potrebno provesti podešavanje veličine vremenskog prozora, određujući koliko prethodnih vremenskih koraka će se koristiti za predviđanje sljedećeg. Za svaku određenu veličinu vremenskog prozora, podaci se pripremaju posebno za konvolucijski model vremenskih serija. Drugo, provodi se prilagodba stope učenja, koja je ključna za optimizaciju parametara modela. Treće, prilagođava se parametar koji određuje jačinu regularizacije. Za svaku kombinaciju veličine vremenskog prozora, stope učenja i jačine regularizacije, model se trenira kroz određeni broj epoha, a zatim se model procjenjuje na validacijskom i testnom skupu. Ovaj dio predstavlja sveobuhvatan postupak za prilagodbu hiperparametara neuronske mreže pružajući detaljnu analizu kako različite vrijednosti ključnih hiperparametara utječu na kvalitetu modela. Kroz ovu analizu, moguće je identificirati optimalne hiperparametre koji rezultiraju najboljim vrijednostima R^2 modela na testnim podacima. Također uz vrijednosti R^2 i *MSE* (engl. *Mean Square Error*, MSE) praćena je vrijednost *MSE*-a tijekom treniranja odnosno praćena je funkcija gubitka za sve kombinacije hiperparametara.

Osim konvolucijskog modela vremenskih serija, razvijen je i model baziran na metodi slučajnih šuma (engl. *Random Forest*). Model slučajnih šuma treniran je na istom skupu podataka kao i CNN model, koristeći optimalne hiperparametre odabrane putem validacije. Svi hiperparametri za dane modele prikazani su u tablici 4.1.

Tablica 4.1. Vrijednosti hiperparametara korištenih za određeni algoritam

Algoritam	Hiperparametar	Vrijednosti
1D-CNN	vremenski prozor	[12,24,48]
	alpha	[0.001,0.01,0.1]
	learning_rate	[0.0001,0.0005,0.001]
	solver	Adam
	early_stop	TRUE
Random Forest	*max_features*	['auto',log2','sqrt']
	ccp_alpha	[0.1,0.01,0.001]
	max_depth	[6,7,8]
	min_samples_split	[3,4,5]
	n_estimators	[100,150,200]
	random_state	[42]

U ovom radu su za predviđanje koncentracija NO_2 korištene meteorološke, temporalne i *lag* značajke. Izbor ovih značajki temeljio se na ranijim istraživanjima i empirijskim mjerenjima prikazanim u prethodnim dijelovima ovog rada. Meteorološke značajke imaju značajan utjecaj na koncentraciju NO_2. Dakle, korištenje meteoroloških podataka kao značajki može poboljšati efikasnost modela. Temporalne značajke su ključne za modeliranje vremenskih serija poput koncentracija NO_2. Koncentracije NO_2 pokazuju snažnu sezonalnu komponentu i variraju ovisno o dobu dana[3]. Korištenjem temporalnih značajki, model može naučiti ove obrasce i koristiti ih za bolje predviđanje budućih koncentracija NO_2. Konačno, korištene su *lag* značajke. Autokorelacija opisuje situaciju gdje su vrijednosti vremenske serije u jednom trenutku povezane s vrijednostima u nekom prethodnom trenutku. Na slici 4.6. prikazan je graf autokorelacija za koncentraciju NO_2. *Lag* značajke su korištene kako bi se modelu omogućio „pogled unatrag“ u vremenu te se te informacije koriste za predviđanje budućih koncentracija NO_2. Vrijednosti koncentracije NO_2 unutar 3 sata koreliraju 80% te su prisutni skokovi u korelaciji svakih 12 i 24 sata. Najveći skok u korelaciji vidljiv je nakon 24 sata gdje koncentracije NO_2 koreliraju oko 60%. U tablici 4.2. prikazane su sve značajke korištene u modelima.

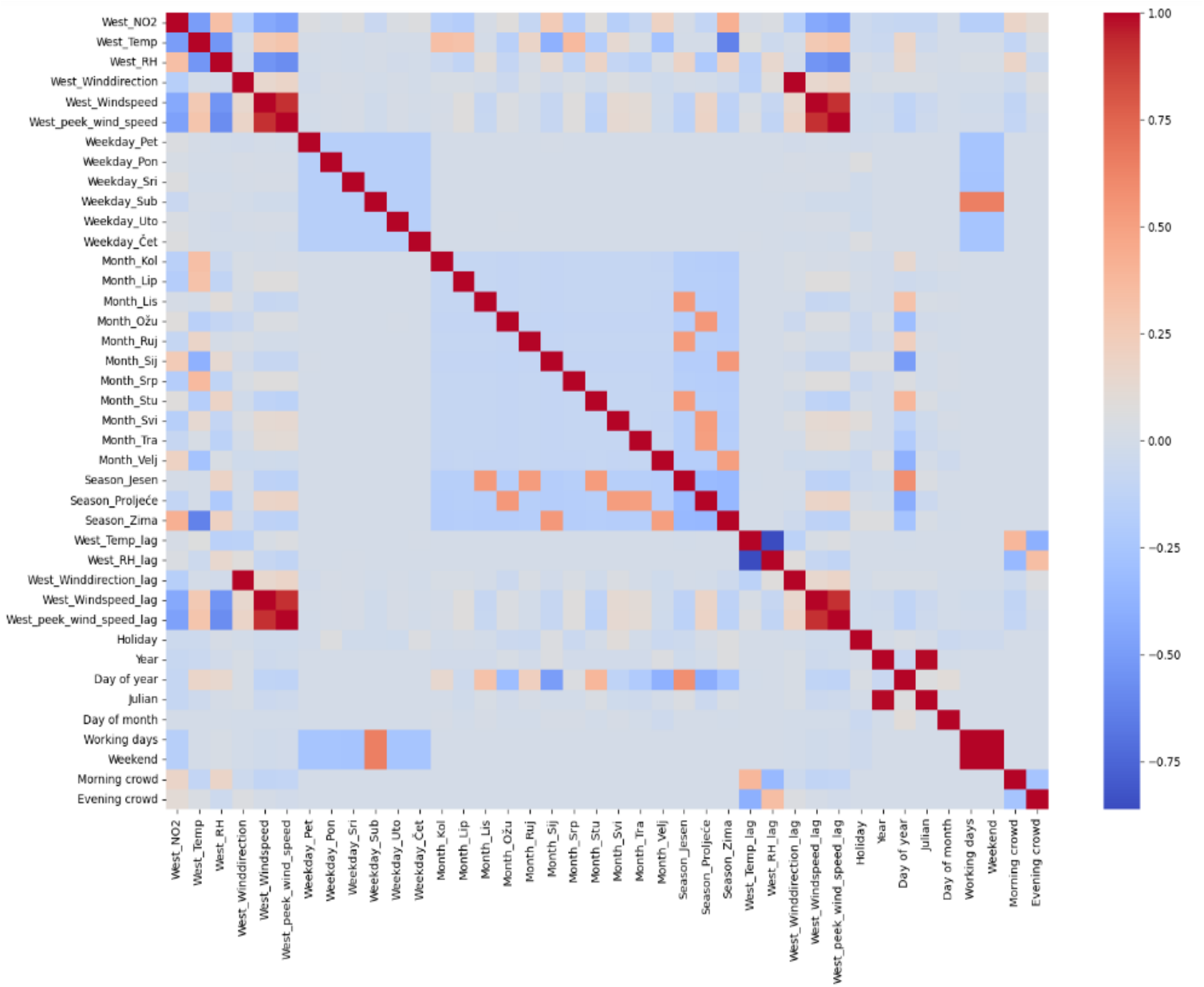


Slika 4.6. *Prikaz autokorelacije koncentracije NO_2*

Tablica 4.2. Prikaz značajki i tip značajke korišten u algoritmima

Tip značajke	Značajka	Mjesec
Temporalne značajke	praznik	godišnje doba
	godina	radni dan
	julianski dan	vikend
	dan u mjesecu	jutarnja gužva
	dan u tjednu	večernja gužva
Značajke vremenske prognoze	temperatura	temperatura_lag
	relativna vlažnost	relativna vlažnost_lag
	smjer vjetra	smjer vjetra_lag
	brzina vjetra	brzina vjetra_lag
	maksimalni naleti brzine vjetra	maksimalni nalet brzine vjetra_lag

Slika 4.7. prikazuje matricu korelacije značajki koje su upotrijebljene za 1D CNN model. Maksimalna pozitivna korelacija (+1.0) označena je crvenom bojom, dok je maksimalna negativna korelacija (-1.0) označena plavom bojom. Promjenom intenziteta boje prikazana je promjena u vrijednosti korelacija, tj. vrijednosti korelacija koje se približavaju nuli. S obzirom na to da je koncentracija NO_2 glavni prediktor u ovom radu, posebna je pozornost usmjerena na korelacije između značajki i koncentracije NO_2.



Slika 4.7. *Toplinska karta međusobnih korelacija značajki i koncentracije NO_2 korištenih za razvoj 1D CNN modela*

5. REZULTATI I RASPRAVA

Baza podataka obuhvaća 71928 uzoraka mjerenja koja su prikupljena od 1. siječnja 2014. do 17. ožujka 2022. Analiza i modeliranje temelje se na podacima koji su dostupni do 15. ožujka 2020. zbog početka pandemije koronavirusa. Dakle, broj uzoraka koji su uzeti u obzir za ovu analizu iznosi 54360. Primarni cilj ovog diplomskog rada je identifikacija najučinkovitijeg modela 1D konvolucijske neuronske mreže kroz iteracije i optimizacije vremenskih jedinica, stope učenja i regularizacije. Nakon što je određen najbolje postignut 1D CNN model, uspoređena je njegova učinkovitost s modelom temeljenim na metodi slučajnih šuma.

5.1. 1D CNN modeli

U ovom dijelu, cilj je detaljno ispitati utjecaj različitog broja vremenskih jedinica u odnosu na stopu učenja i regularizacijski član *lambda*. Svaki od ovih parametara ima značajan utjecaj na kvalitetu modela, a njihovo pažljivo prilagođavanje ključno je za optimizaciju modela. Istraživane su različite konfiguracije ovih parametara, uspoređujući njihove učinke za različite brojeve vremenskih jedinica. Pretpostavka je da će povećanje broja vremenskih jedinca pozitivno utjecati na kvalitetu modela. Takva teza proizlazi iz toga da veći broj vremenskih jedinica uzima veći broj podataka iz prošlosti za predviđanje budućnosti.

5.1.1 1D CNN 12 vremenskih jedinica

U ovom poglavlju su prikazani rezultati za modele sa brojem vremenskih jedinica 12 uz različitu stopu učenja i reguralizacijske članove *lambda*. U tablici 5.1. prikazane su dobivene vrijednosti za R^2 i *MSE* na testnom podskupu podataka.

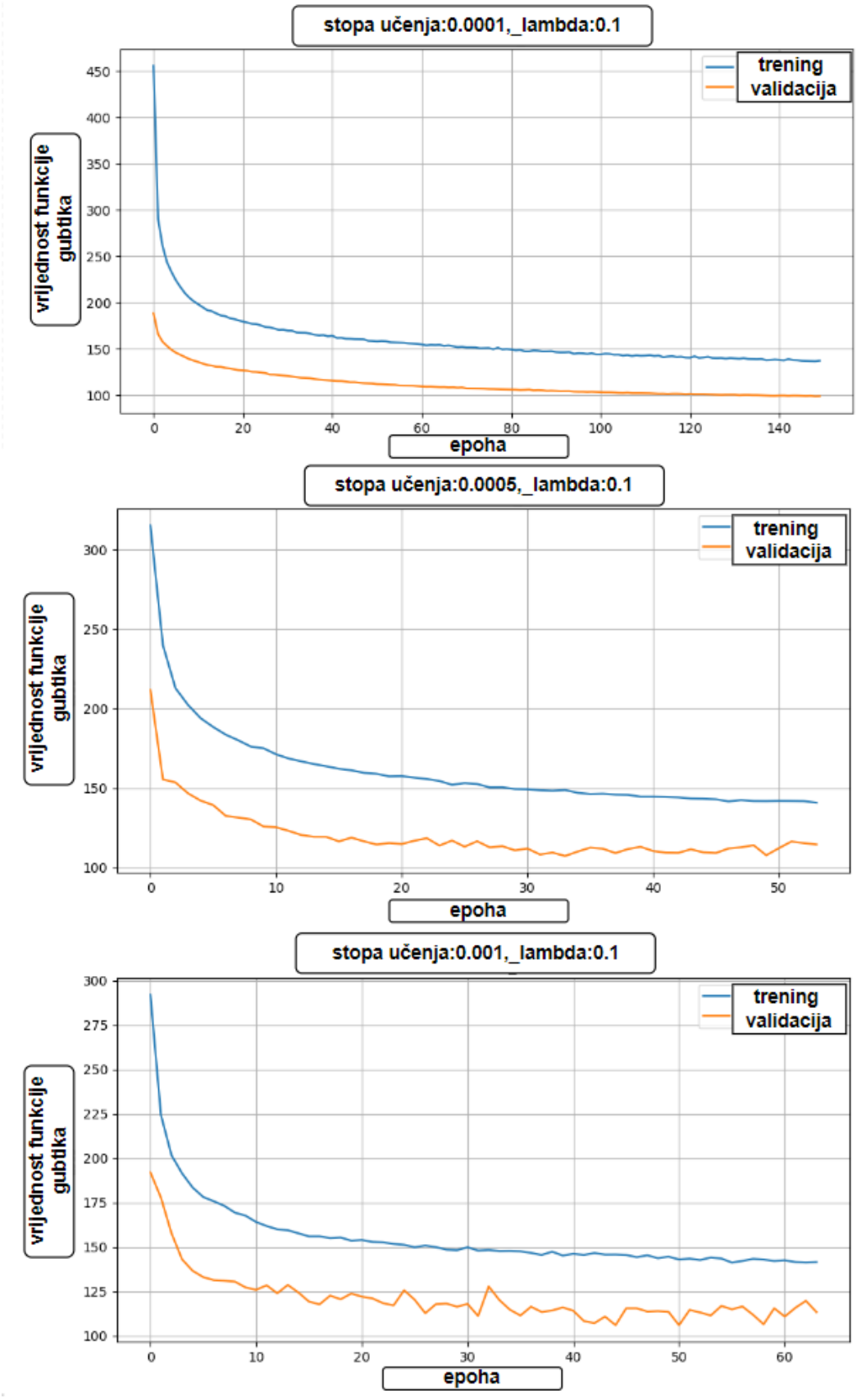
Tablica 5.1. Rezultati modela za broj vremenskih jedinica 12

broj vremenskih jedinica	stopa učenja	*lambda*	R^2	*MSE*
12	0,0001	0,001	0,57	112,51
		0,01	0,57	112,75
		0,1	0,62	98,12
	0,0005	0,001	0,50	130,38
		0,01	0,42	150,27
		0,1	0,51	127,48
	0,001	0,001	0,46	140,45
		0,01	0,42	150,83
		0,1	0,54	118,35

Model s najboljim vrijednostima R^2 i *MSE* dobiven je pri stopi učenja 0,0001 i *lambda* 0,1. Tijekom treniranja modela smanjuju se vrijednosti funkcije gubitka na trenirajućem i testnom skupu. Uz to treba naglasiti da je učenje brže u prvim epohama neovisno o stopi učenja. Također na slici 5.1 vidljivo je kako je stabilnost učenja modela veća pri manjim stopama učenja. Pri malim vrijednostima stope učenja modeli sporije uče, ali stabilnije dolaze u minimum. Na slici 5.1 vidljivo je skup za treniranje stabilnije dolazi u minimum u odnosu na validacijski skup. U tablici 5.2. vidljivo je da model za najmanje vrijednosti stope učenja ima najbolje vrijednosti R^2 i *MSE*.

Tablica 5.2. Prikaz vrijednost R^2 i *MSE* grupirane srednje vrijednosti po stopi učenja za broj vremenskih jedinica 12

broj vremenskih jedinica	stopa učenja	R^2	*MSE*
12	0,0001	0,59	107,79
	0,0005	0,48	136,04
	0,001	0,48	136,54



Slika 5.1. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 12 (preostali grafovi za broj vremenskih jedinica 12 dani su u Dodatku 1)*

5.1.2 1D CNN 24 vremenske jedinice

U ovom poglavlju su prikazani rezultati za modele sa brojem vremenskih jedinica 24 uz različite stope učenja i regularizacijske članove *lambda*. U tablici 5.3. prikazane su dobivene vrijednosti R^2 i *MSE* na testnom podskupu podataka.

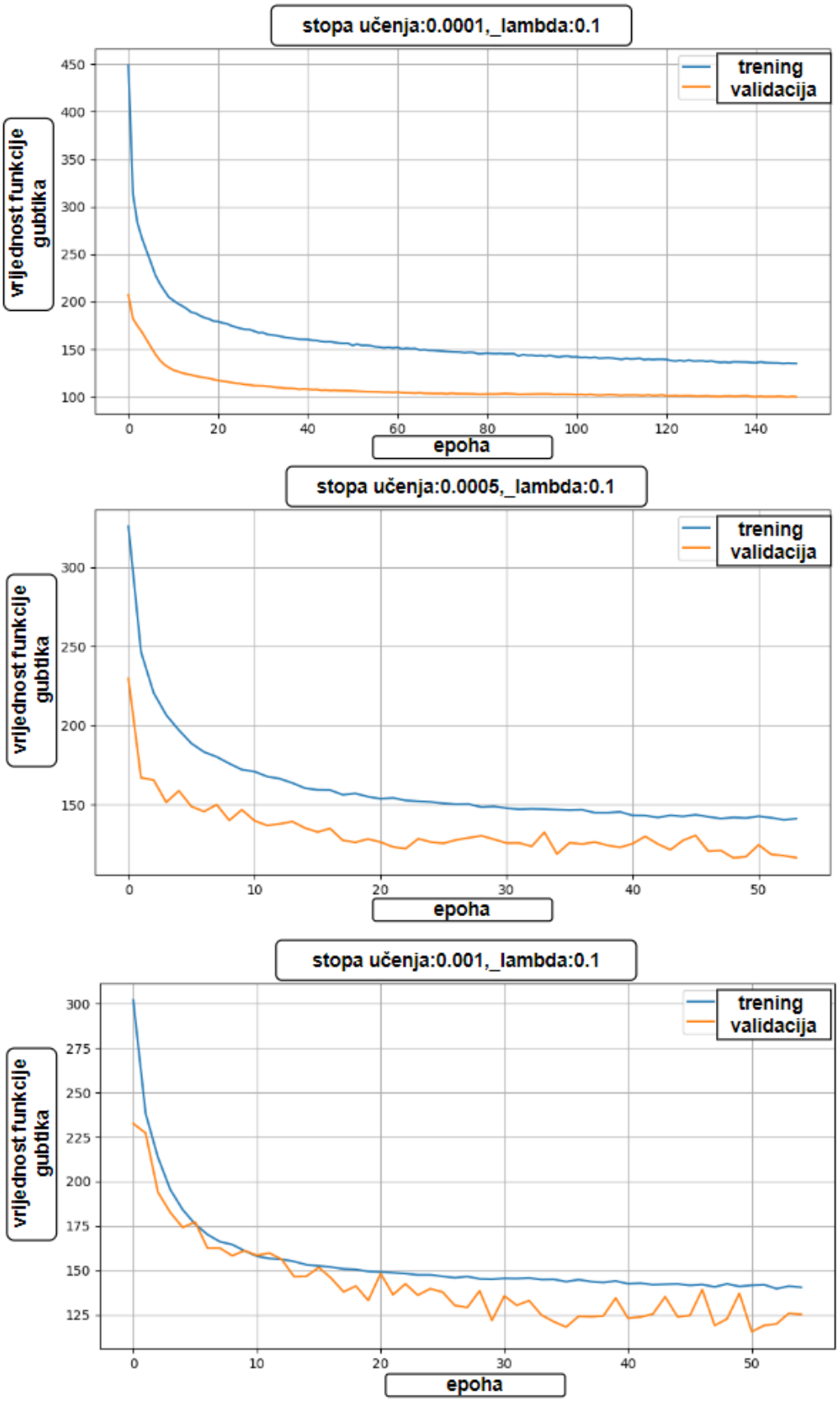
Tablica 5.3. Rezultati modela za broj vremenskih jedinica 24

broj vremenskih jedinica	stopa učenja	_lambda	R^2	*MSE*
24	0,0001	0,001	0,51	128,11
		0,01	0,56	115,39
		0,1	0,60	103,53
	0,0005	0,001	0,36	166,97
		0,01	0,39	159,99
		0,1	0,38	162,20
	0,001	0,001	0,00	259,89
		0,01	0,18	214,84
		0,1	0,35	170,81

Model s najboljim vrijednostima R^2 i *MSE* je dobiven pri stopi učenja 0,0001 i *lambda* 0,1. Na slici 5.2. vidi se proces učenja modela. Tijekom treniranja modela smanjuju se vrijednosti funkcije gubitka na trenirajućem i testnom skupu. Također kao i kod korištenja broja vremenskih jedinica 12 model sporije, ali stabilnije uči. U tablici 5.4. vidljivo je da model za najmanje vrijednosti stope učenja ima najbolje vrijednosti R^2 i *MSE*.

Tablica 5.4. Prikaz vrijednost R^2 i *MSE* grupirane srednje vrijednosti po stopi učenja za broj vremenskih jedinica 24

Broj vremenskih jedinica	stopa učenja	R^2	*MSE*
24	0,0001	0,56	115,67
	0,0005	0,38	163,05
	0,001	0,18	215,18



Slika 5.2. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 24 (preostali grafovi za broj vremenskih jedinica 24 dani su u Dodatku 1)*

5.1.3 1D CNN 48 vremenske jedinice

U ovom poglavlju su prikazani rezultati za modele sa 48 vremenskih jedinica uz različite stope učenja i regularizacijski član *lambda*. U tablici 5.5. su prikazane dobivene vrijednosti za R^2 i *MSE* na testnom podskupu podataka.

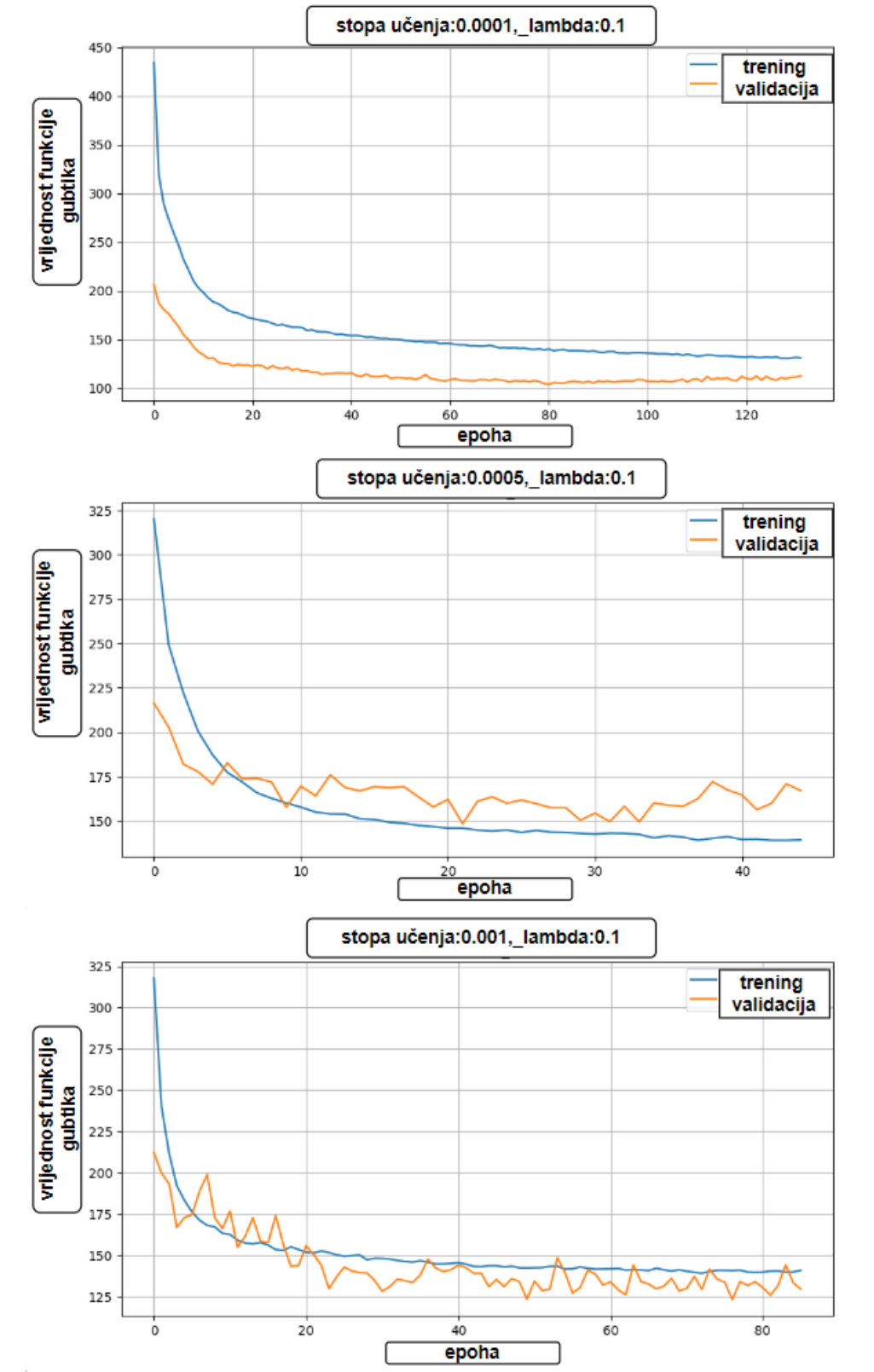
Tablica 5.5. Prikaz rezultata modela za broj vremenskih jedinica 48

broj vremenskih jedinica	stopa učenja	*lambda*	R^2	*MSE*
48	0,0001	0,001	0,49	132,61
		0,01	0,57	111,30
		0,1	0,38	161,55
	0,0005	0,001	0,31	179,01
		0,01	0,32	178,06
		0,1	0,21	206,17
	0,001	0,001	0,39	159,96
		0,01	0,22	203,73
		0,1	0,34	171,41

Model sa najboljim vrijednostima R^2 i *MSE* je dobiven pri stopi učenja 0,0001 i *lambda* 0,01. Na slici 5.3. vidi se proces učenja modela. Tijekom treniranja modela smanjuju se vrijednosti funkcije gubitka na trenirajućem i testnom skupu. Također kao i za brojeve vremenskih jedinica 12 i 24 model sporije, ali stabilnije uči pri nižim vrijednostima stope učenja. U tablici 5.6. vidljivo je da model za najmanje vrijednosti stope učenja ima najbolje vrijednosti R^2 i *MSE*.

Tablica 5.6. Prikaz vrijednosti R^2 i *MSE* grupirane srednje vrijednosti po stopi učenja za broj vremenskih jedinica 48

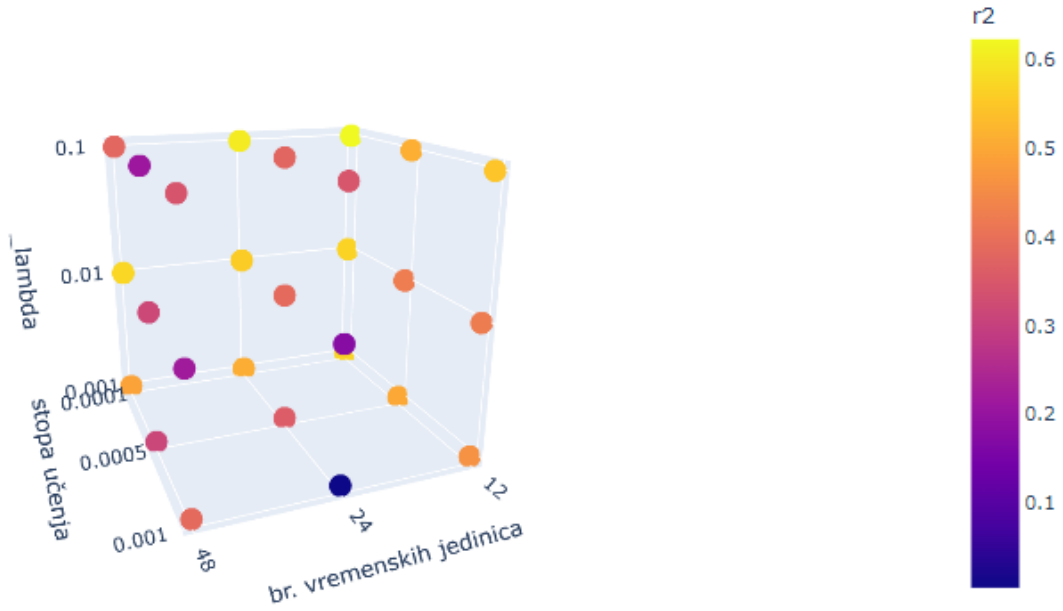
Broj vremenskih jedinica	Stopa učenja	R^2	*MSE*
48	0,0001	0,48	135,15
	0,0005	0,28	187,75
	0,001	0,32	178,37



Slika 5.3. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 48 (preostali grafovi za broj vremenskih jedinca 48 dani su u Dodatku 1)*

5.2. Međusobna usporedba 1D CNN modela

Na slici 5.4. prikazana je 3D mreža hiperparametara u kojoj je vidljivo da je najveća vrijednost R^2 od 0.62 dobivena pri broju vremenskih jedinica 12, *lambdi* 0,1 i stopi učenja 0,0001. Model s najmanjom vrijednosti R^2 je dobiven pri broju vremenskih jedinica 48, stopi učenja 0,001 i lambdi 0,1. U tablicama 5.2, 5.4. i 5.6 vidljivo je da su za svaku stopu učenja najveće vrijednosti R^2 pri broju vremenskih jedinica 12. Ovakvi rezultati mogu se opravdati arhitekturama modela. Budući da zbog konvolucijskih slojeva nije bilo moguće ujednačiti arhitekturu u potpunosti, tj. imati podjednaki broj težinskih koeficijenata, modeli sa većim brojem vremenskih jedinica (24 i 48) imaju veće razlike u broju neurona između dva skrivena sloja. Korištenjem takve arhitekture mreža može postati sklona preprilagođavanju ili nedovoljnome prilagođavanju, te neće dobro generalizirati na nove podatke ili neće uspjeti opisati složenost problema. Velika razlika u broju neurona između slojeva može dovesti do manje stabilnog učenja, otežavajući optimizaciju težina i pristranosti mreže i usporavajući konvergenciju. Neujednačeni broj neurona može izazvati problem nestajanja ili eksplodiranja gradijenta, što ometa proces učenja mreže.[15]



Slika 5.4. *Pregled 3D mreže hiperparametara modela te vrijednosti R^2*

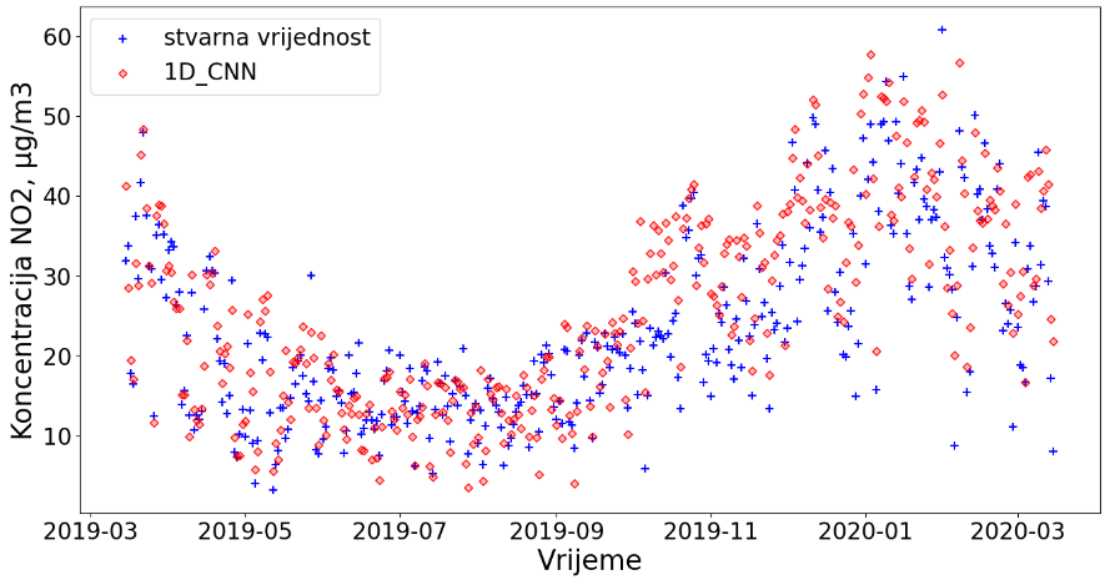
5.3. Usporedba 1D CNN modela sa *Random Forest* modelom

U tablici 5.7. uspoređene su vrijednosti najboljih modela za različit broj vremenskih jedinica s *Random Forest* modelom. *Random Forest* model bolji je od 1D CNN modela sa brojem vremenskih jedinica 24 i 48 dok je usporediv sa brojem vremenskih jedinica 12, ali i dalje poprima veću vrijednost R^2 i nižu *MSE* vrijednost.

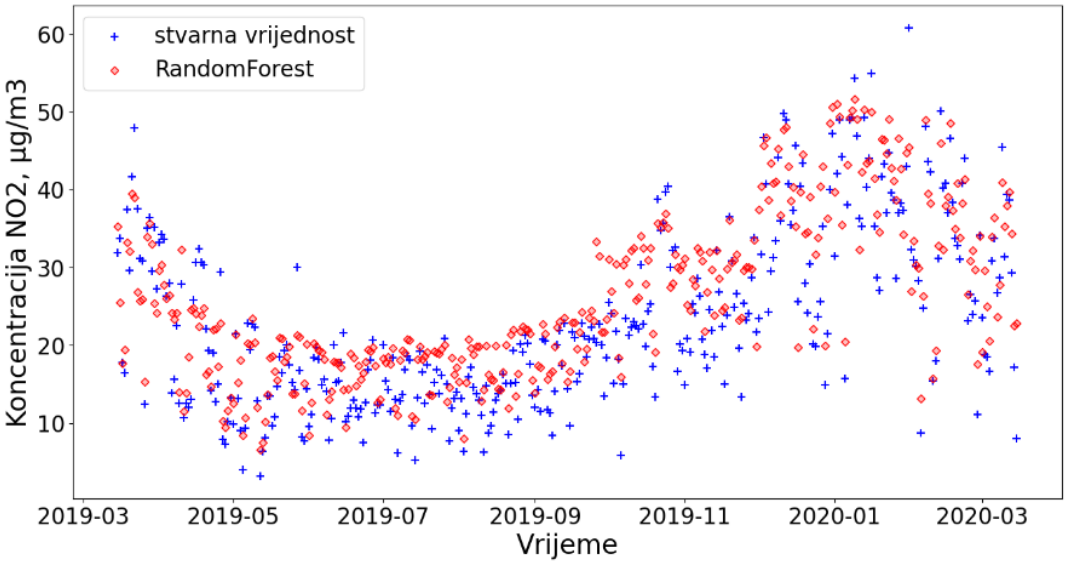
Tablica 5.7. Vrijednosti *MSE* i R^2 za najbolje modele pojedinog broja vremenskih jedinica i *Random Foresta*

Model	***Random Forest***	**Najbolji 1D CNN (br. vremenskih jedinica 12)**	**Najbolji 1D CNN (br. vremenskih jedinica 24)**	**Najbolji 1D CNN (br. vremenskih jedinica 48)**
MSE	91,2	98,12	103,53	111,30
$\boldsymbol{R^2}$	0,65	0,62	0,60	0,57

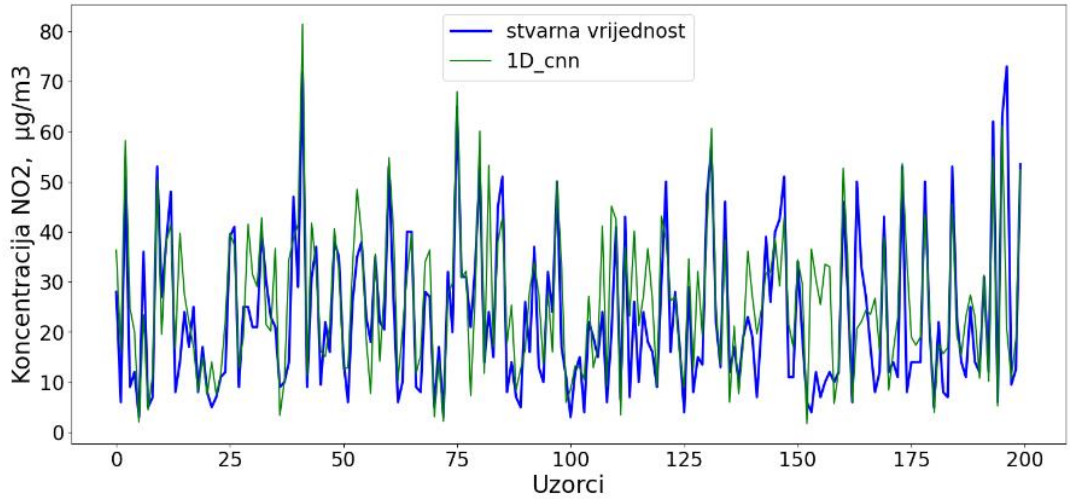
Na slikama 5.5. i 5.6. prikazani su najbolji 1D CNN i *Random Forest* modeli usporedno s pravim vrijednostima koncentracija NO_2 na testnome podskupu. Na slikama 5.7. i 5.8. prikazano je 200 slučajnih stvarnih vrijednosti koncentracija NO_2 i koncentracije koje su dobivene pomoću najboljeg 1D CNN modela i najboljeg *RandomForest* modela. Također iz slike 5.5 vidljivo je da model procjenjuje slične vrijednosti NO_2 sa sličnom točnošću.



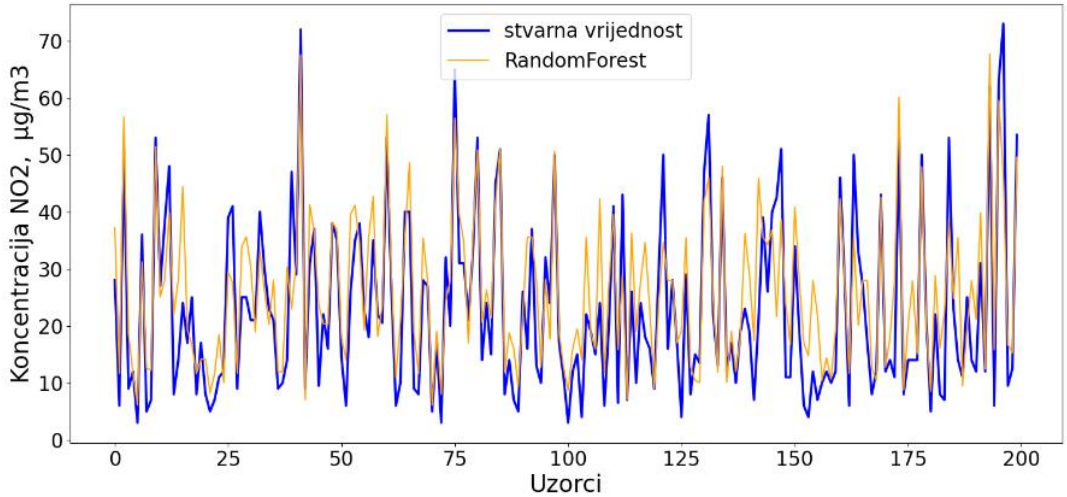
Slika 5.5. *Prikaz stvarnih vrijednosti u usporedbi sa najboljim 1D CNN modelom grupirane kao srednje vrijednosti po danu*



Slika 5.6. *Prikaz stvarnih vrijednosti u usporedbi sa Random Forest modelom grupirane kao srednje vrijednosti po danu*



Slika 5.7. *Prikaz stvarnih vrijednosti u usporedbi sa najboljim 1D CNN modelom*



Slika 5.8. *Prikaz stvarnih vrijednosti u usporedbi sa Random Forest modelom*

6. ZAKLJUČAK

U ovom istraživanju je prikazan razvoj modela konvolucijskih neuronskih mreža i modela nasumičnih šuma u svrhu procjene koncentracije NO_2 u zraku u gradu Grazu na lokaciji Zapad. Modeli u ovome radu razvijeni su koristeći podatke iz razdoblja od 15. ožujka 2019 do 15. ožujka 2020. Istraživanje je provedeno koristeći 71928 uzorka mjerenja, ali je za analizu i modeliranje korišteno 54360 uzoraka dostupnih do početka pandemije koronavirusa. Cilj diplomskog rada je bio identificirati najučinkovitiji model 1D konvolucijske neuronske mreže (1D CNN) pronalaženjem optimalnog broja vremenskih jedinica, stope učenja i regularizacije modela. Modeli vremenskih nizova su optimizirani za različite brojeve vremenskih jedinica, a utvrđeno je da se sporije, ali stabilnije učenje provodi pri nižim stopama učenja. Također skup za treniranje stabilnije dolazi u minimum u odnosu na validacijski skup. Najbolji rezultati (R^2=0,62, *MSE*=98,13) dobiveni su za konvolucijski model sa brojem vremenskih jedinica 12, stopom učenja 0,0001 i parametrom *lambda* 0,1. Modeli sa većim brojem vremenskih jedinica (24 i 48) imali su nešto lošije rezultate. S obzirom da veći broj vremenskih jedinica uzima za razvoj modela veći broj podataka, nije bilo za očekivati niže vrijednosti R^2. Ova nelogičnost u vladanju modela se može objasniti sa sveobuhvatno lošijom arhitekturom modela. Zbog usporedbe i računalnog ograničenja, mreže sa većim brojem vremenskih jedinica, imaju veliku razliku u broju neurona u zadnjim slojevima mreže što može dovesti do nestabilnijeg učenja mreže. Stoga bi u budućem istraživanju moglo biti korisno ispitati veću dubinu mreže za veći broj vremenskih jedinica kako bi se izbjegla spomenuta razlika. Iako su 1D CNN modeli pokazali zadovoljavajuće rezultate za konačnu primjenu, utvrđeno je da je *Random Forest* model nešto bolji od 1D CNN modela s brojem vremenskih jedinica 24 i 48. Modeli sa brojem vremenskih jedinica 12 su usporedivi s modelom nasumičnih šuma, ali je sveukupno model nasumičnih šuma ipak postigao bolje rezultate. Na temelju provedenog istraživanja zaključeno je da, iako 1D CNN modeli imaju potencijala za identifikaciju složenih obrazaca u korištenim podacima, modeli temeljeni na metodi slučajnih šuma pružaju nešto bolje rezultate u ovom specifičnom kontekstu. Također, za buduće istraživanje bilo bi korisno istražiti kako poboljšati optimizaciju hiperparametara 1D CNN modela.

7. POPIS SIMBOLA I KRATICA

a – *č*vorovi u mreži

Adam optimizacijski algoritam (engl. *Adaptive Moment Estimation*)

b – koeficijent pristranosti

d – broj filtara

Duboko učenje (engl. *Deep Learning*, DL)

Europska agencija za okoliš (engl. *European Environment Agency*, EEA)

f – veličina filtra

GEMM (engl. *General matrix multiply*)

Ispravljena linearna aktivacijska funkcija (engl. *Rectified Linear Unit*, relu),

Ispravljena linearna aktivacijska funkcija s propuštanjem (engl. *Leaky Rectified Linear Unit, Leaky* relu)

J(w,b) – funkcija gubitka

k – veličina uzorka

Konvolucijske neuronske mreže (engl. *Convolutional Neural Networks*, CNN)

l – sloj neuronske mreže

lambda, λ – parameter koji pomaže smanjenju prenaučenosti modela

m – cijeli skup podataka

Metoda gradijentnog spusta (engl. *Gradient Descent*, GD)

Metoda slučajnih šuma (engl. *Random Forest*, RF)

Mini-grupa (engl. *Mini Batch*, MB)

NO_2 – dušikov dioksid

NO_x – dušikovi oksidi

p – dopunjavanje

$PM_{2,5}$ – lebdeće čestice promjera 2,5 μm

R^2 – koeficijent determinacije

s – korak

Srednja kvadratna pogreška (engl. *Mean squared error*, MSE)

Stohastički gradijentni spust (engl. *Stohastic Gradient Descent*, SGD)

Strojno učenje (engl. *Machine learning*, ML)

Svjetska zdravstvena organizacija (engl. *World Health Organization*, WHO)

Tangens hiperbolička aktivacijska finkcija (engl. *Hyperbolic tangent*, tanh)

Umjetna inteligencija (engl. *Artificial intelligence*, AI)

Unakrsna entropija (engl. *Cross entropy*, CE)

w – težinski koeficijent

W – vrijednosti težina unutar konvolucijske mreže

$\mu g/m^3$ – mikrogrami po kubnom metru

$a_i^{(l)}$ – i-ti čvor u l-tom sloju

$w^{(l)}$ – veza između k-te jedinice u sloju „l“ i j-te jedinice u sloju „$l+1$“

8. LITERATURA

[1] L. Bai, J. Wang, X. Ma, H. Lu, Air pollution forecasts: An overview, *Int. J. Environ. Res. Public. Health,* **15**(4) (2018) 780.

[2] WHO, Household air pollution, dostupno online: https://www.who.int/news-room/fact-sheets/detail/household-air-pollution-and-health, (pristupljeno 17.5.2023.)

[3] M. Lovrić, K. Pavlović, M. Vuković, S. K. Grange, M. Haberl, R. Kern, Understanding the true effects of the COVID-19 lockdown on air pollution by means of machine learning, *Environ. Pollut.* **274** (2021) 115900.

[4] M. L. Bell, D. L. Davis, Reassessment of the lethal London fog of 1952: novel indicators of acute and chronic consequences of acute exposure to air pollution, *Environ. Health Perspect.* **109** (2001) 389–394.

[5] IQAir, 2021 World Air Quality Report, dostupno online: https://www.iqair.com/world-most-polluted-cities/world-air-quality-report-2021-en.pdf, (pristupljeno 17.5.2023.)

[6] WHO, Types of pollutants, dostupno online: https://www.who.int/teams/environment-climate-change-and-health/air-quality-and-health/health-impacts/types-of-pollutants, (pristupljeno 17.5.2023.)

[7] T. M. Chen, W. G. Kuschner, J. Gokhale, S. Shofer, Outdoor air pollution: nitrogen dioxide, sulfur dioxide, and carbon monoxide health effects, *Am. J. Med. Sci.* **333**(4) (2007) 249–256.

[8] WHO, Billions of people still breathe unhealthy air: new WHO data, dostupno online: https://www.who.int/news/item/04-04-2022-billions-of-people-still-breathe-unhealthy-air-new-who-data, (pristupljeno 17.5.2023.)

[9] S. Raschka, Y. Liu, V. Mirjalili, D. Dzhulgakov, Machine Learning with PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python, Packt Publishing Ltd., Birmingham, 2022.

[10] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, *Appl. Soft Comput.* **97** (2020) 105524.

[11] P. Singh, Dimensionality Reduction Approaches, 2020. dostupno online: https://towardsdatascience.com/dimensionality-reduction-approaches-8547c4c44334, (pristupljeno 1.6.2023.)

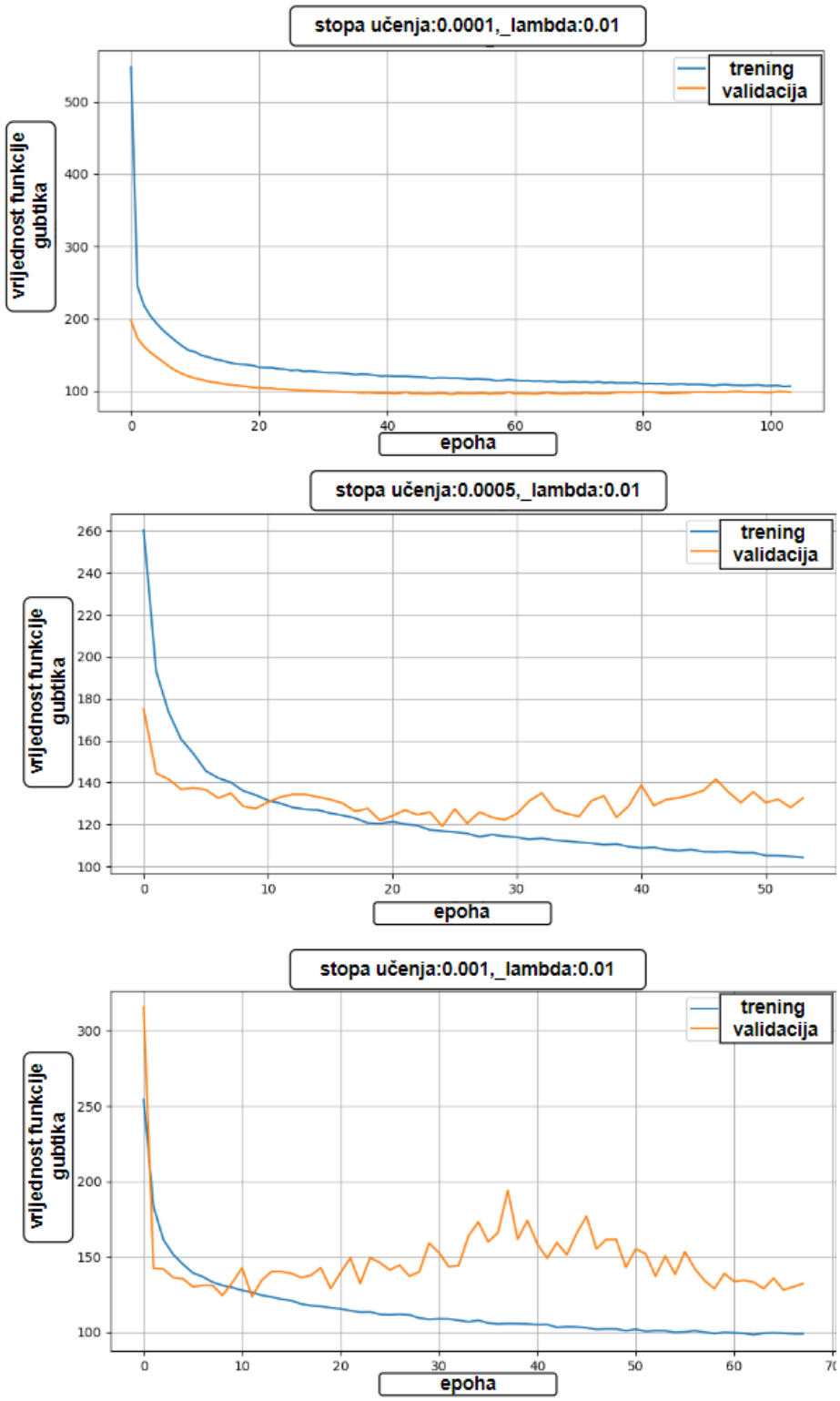
[12] J. Fumo, A Gentle Introduction To Neural Networks Series — Part 1, 2017. dostupno online: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc, (pristupljeno 7.6.2023.)

[13] Machine Learning Course, dostupno online: https://www.coursera.org/specializations/machine-learning-introduction, (pristupljeno 1.6.2023.)

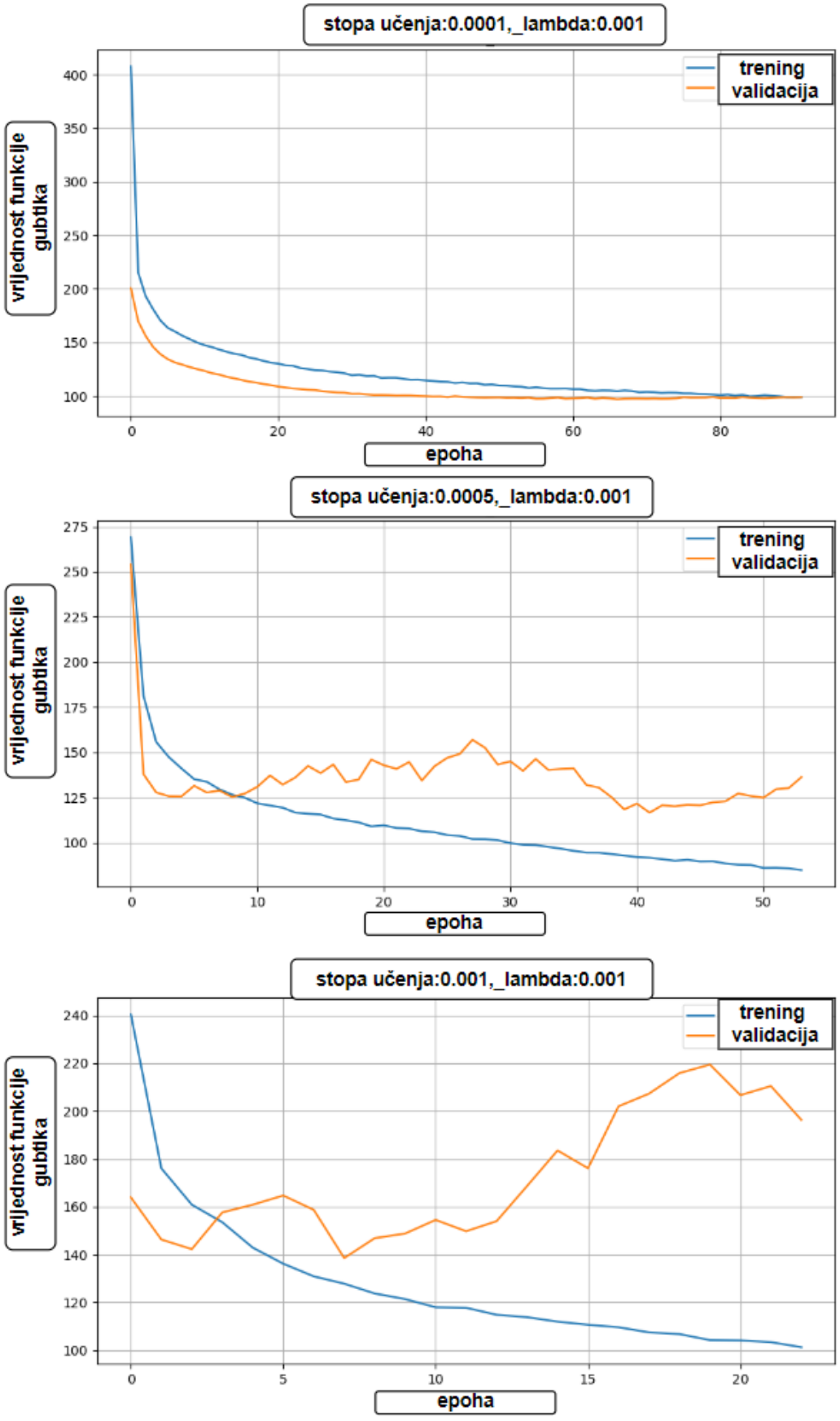
[14] S. Sharma, Activation Functions in Neural Networks, 2022., dostupno online: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, (pristupljeno 7.6.2023.)

[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, Cambridge, Massachusetts, 2016.
[16] Z. Brodtman, The Importance and Reasoning behind Activation Functions, 2021., dostupno online: https://towardsdatascience.com/the-importance-and-reasoning-behind-activation-functions-4dc00e74db41, (pristupljeno 20.6.2023.)
[17] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521**(7553) (2015) 436-444.
[18] R. Kwiatkowski, Gradient Descent Algorithm, a deep dive, 2021., dostupno online: https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21, (pristupljeno 21.6.2023.)
[19] Crypto1, How Does the Gradient Descent Algorithm Work in Machine Learning?, 2020., dostupno online: https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/, (pristupljeno 21.6.2023.)
[20] Deep Learning Course, dostupno online: https://www.coursera.org/specializations/deep-learning, (pristupljeno 1.6.2023.)
[21] S. Patrikar, Batch, Mini Batch & Stochastic Gradient Descent, 2019., dostupno online: https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a, (pristupljeno 21.6.2023.)
[22] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint*, (2014). arXiv:1412.6980.
[23] P. Gupta, Regularization in Machine Learning, 2017., dostupno online: https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a, (pristupljeno 20.6.2023.)
[24] M. Yani, S. S. M. Budhi Irawan, S. M. Casi Setiningsih, Application of transfer learning using convolutional neural network method for early detection of terry's nail, *J. Phys. Conf. Ser.*, **1201**(1) (2019) 012052.
[25] A. Di Bucchianico, Coefficient of determination (R^2), u F. Ruggeri, R.S. Kenett, F.W. Faltin, Encyclopedia of statistics in quality and reliability, Wiley, 2008.
[26] Wikipedija, Graz, dostupno online: https://hr.wikipedia.org/wiki/Graz, (pristupljeno 25.5.2023.)
[27] Weather Spark, Graz Climate, Weather By Month, Average Temperature (Austria), dostupno online: https://weatherspark.com/y/79331/Average-Weather-in-Graz-Austria-Year-Round, (pristupljeno 25.5.2023.)
[28] Hrvatska enciklopedija, Vjetar, dostupno online: https://www.enciklopedija.hr/natuknica.aspx?ID=64995, (pristupljeno 25.5.2023.)
[29] H. Parra, The Data Science Trilogy: NumPy, Pandas and Matplotlib basics, 2021., dostupno online: https://towardsdatascience.com/the-data-science-trilogy-numpy-pandas-and-matplotlib-basics-42192b89e26, (pristupljeno 1.6.2023.)
[30] Plotly, dostupno online: https://plotly.com/api/, (pristupljeno 1.6.2023.)
[31] S. Bhutani, PyTorch Basics in 4 Minutes, 2019., dostupno online: https://medium.com/dsnet/pytorch-basics-in-4-minutes-c7814fa5f03d, (pristupljeno 1.6.2023.)

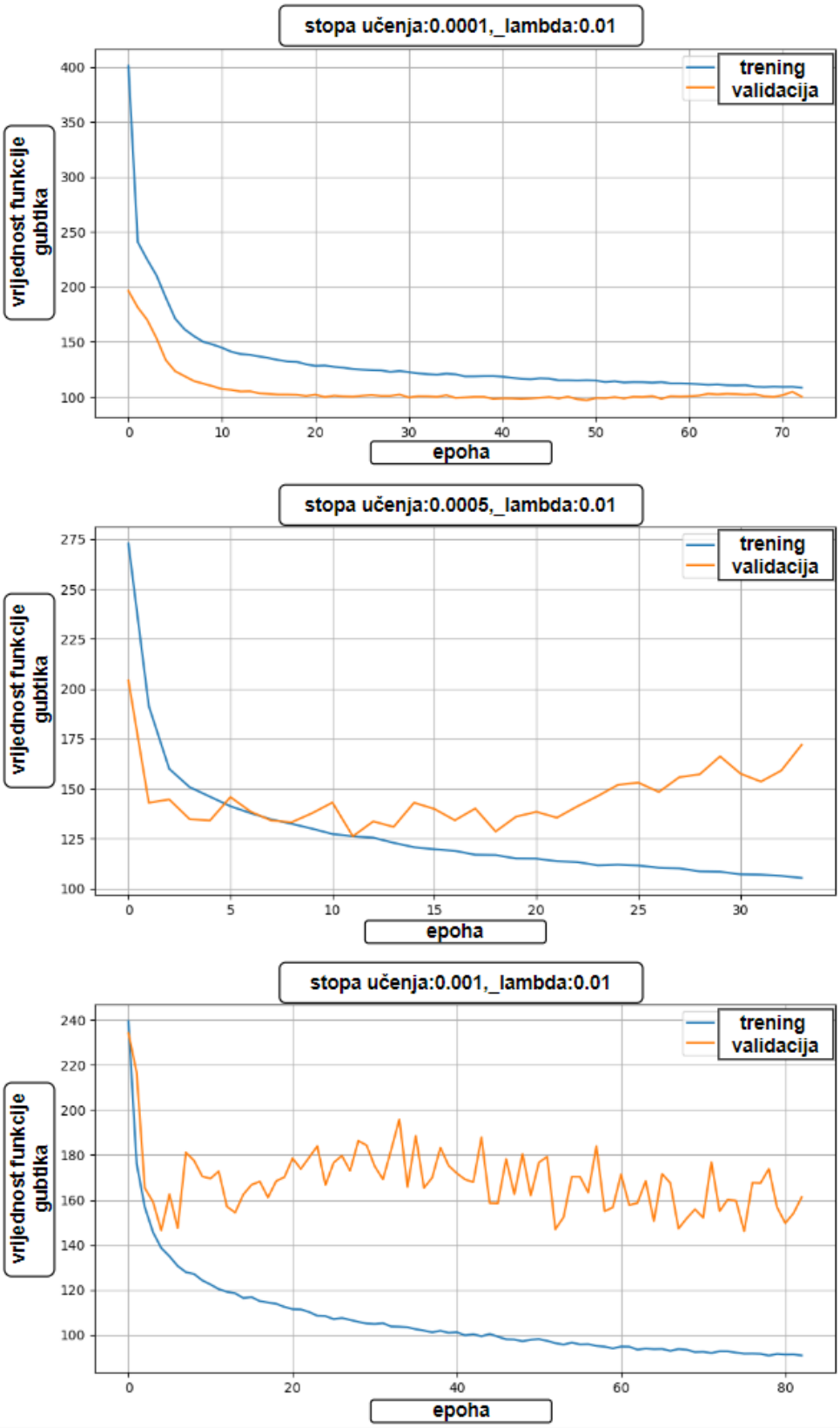
9. DODATAK 1



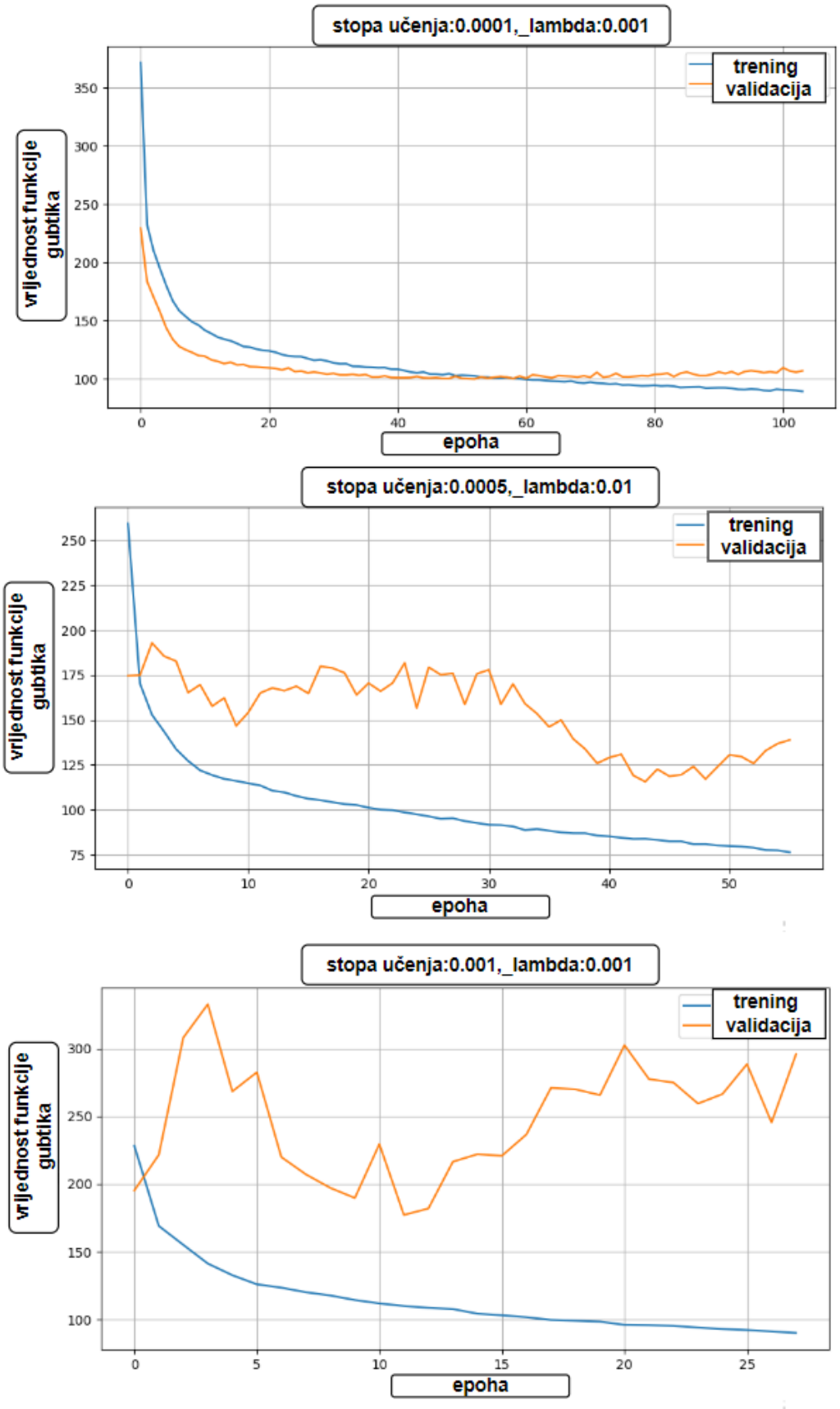
Slika 9.1. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 12*



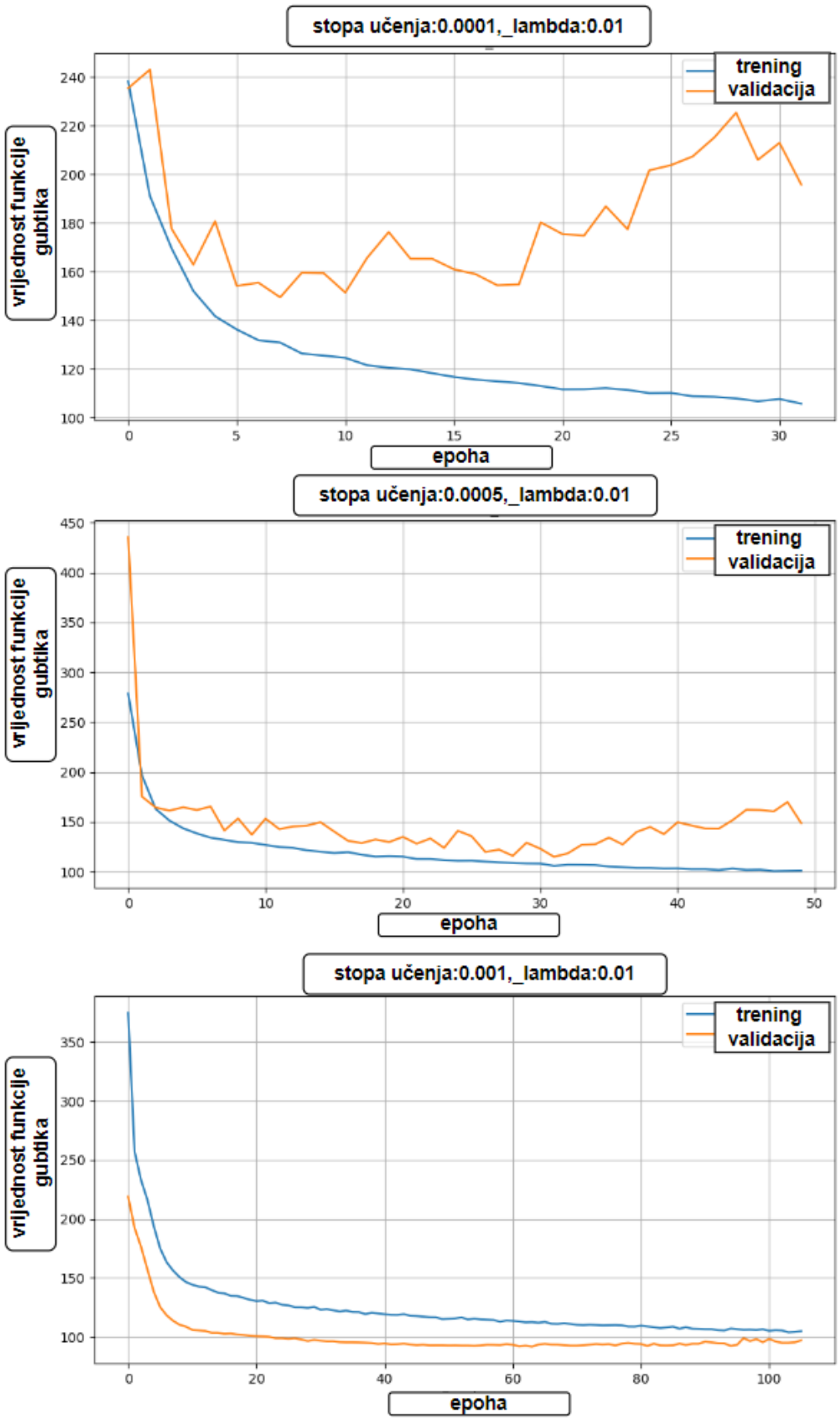
Slika 9.2. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 12*



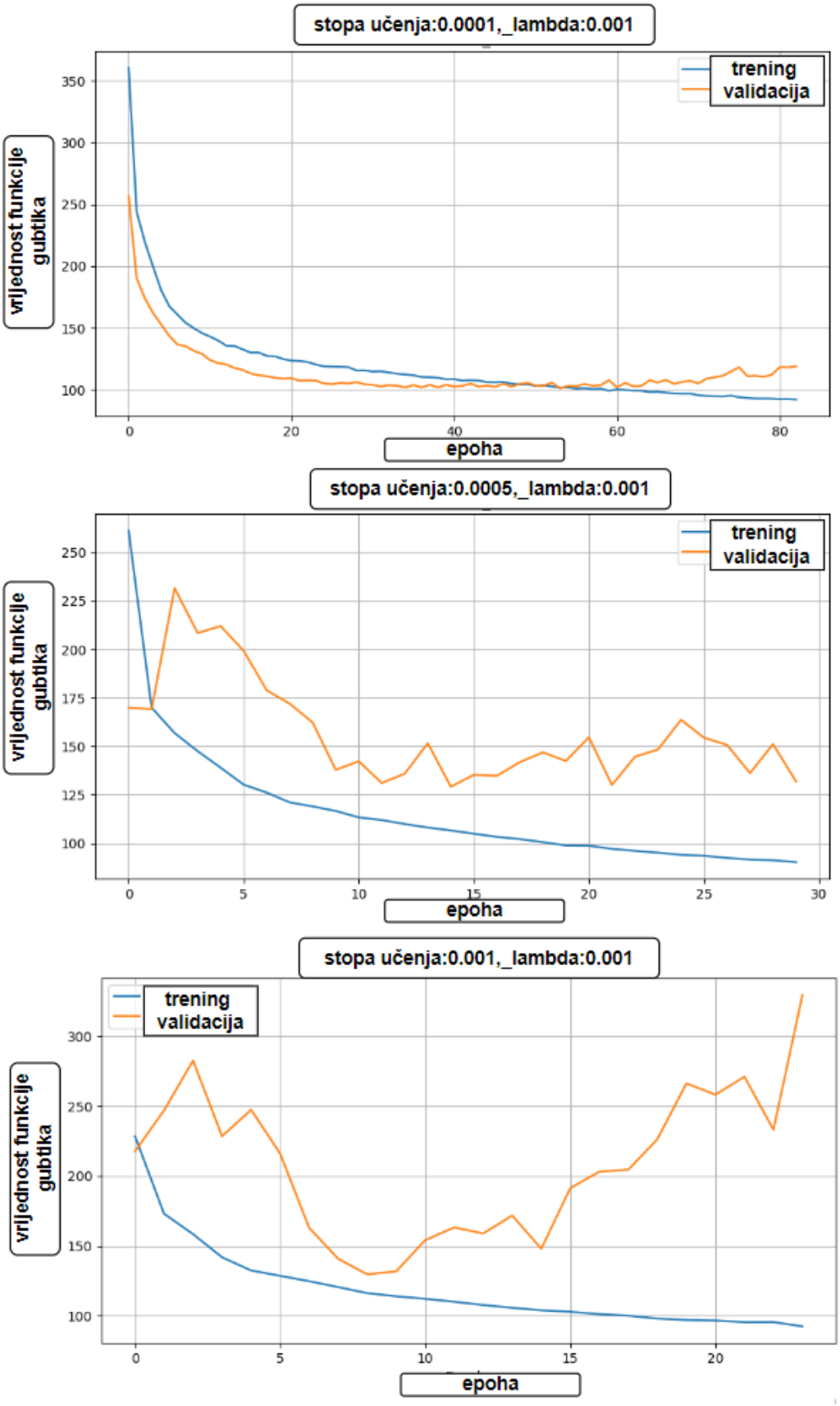
Slika 9.3. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 24*



Slika 9.4. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 24*



Slika 9.5. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 48*



Slika 9.6. *Proces učenja za različite hiperparametre za broj vremenskih jedinica 48*

10. ELEKTRONIČKI DODATAK

Cjelokupna programska implementacija predstavljena u ovom diplomskom radu dostupna je za pregled na *GitHub*-u. Za brz i jednostavan pristup ovom repozitoriju generiran je QR kôd.

Slika 10.1. *QR kôd GitHub repozitorija*

ŽIVOTOPIS

Teo Terzić, [redacted] započeo je svoj obrazovni put u Osnovnoj školi Vladimir Dešćak u Rakitju. Nakon sticanja osnovnih znanja i vještina, školovanje je nastavio u Srednjoj školi Vladimir Prelog, koju je završio 2018. godine s vrlo dobrim uspjehom. Daljnju edukaciju nastavio je na Fakultetu kemijskog inženjerstva, gdje je 2020. godine uspješno obranio završni rad na temu „Priprava i karakterizacija celuluzno acetatnih membrana“. Stručnu praksu odradio je u pogonu farmaceutske tvrtke PLIVA u Savskom Marofu, gdje se upoznao s sigurnosnim mjerama specifičnim za farmaceutsku industriju. Dodatno, radio je kao demonstrator na vježbama iz opće kemije i programiranja.

U sklopu Erasmus+ programa, obavio je studentsku praksu u Know-Center-u u Grazu, gdje je bio zadužen za postavljanje senzorskih sustava koristeći ESP8266 mikroupravljač. Prikupljao je različite vrste okolišnih podataka, uključujući razine PM_{10}, vlažnost, tlak i temperaturu. Izvršio je kalibraciju jeftinih senzora u usporedbi s referentnim postajama, koristeći tehnike strojnog učenja kako bi osigurao točnost prikupljenih podataka. Obradu, analizu i modeliranje podataka izvodio je pomoću programskog jezika Pythona.

Osim akademskih obaveza, puno vremena provodi baveći se sportom. Trenirao je golf u zagrebačkom klubu i košarku u klubu Zagreb. Nakon preseljenja u Svetu Nedelju, odlučio je golf zamijeniti tenisom.

Trenutno je zaposlen kao student u firmi Chemical Codes, gdje radi kao *data scientist*, i u firmi LicenseSpring baveći se analizom podataka i implementacijom API-ja na ESP32 pločicama u kontekstu IoT sustava.