SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1732

Prognoza vremenskih serija u senzorskim tokovima podataka

Antonio Ivčec

Zagreb, lipanj 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 9. ožujka 2018.

DIPLOMSKI ZADATAK br. 1732

Pristupnik: **Antonio Ivčec (0036475156)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Prognoza vremenskih serija u senzorskim tokovima podataka**

Opis zadatka:

Svjedoci smo dramatične tranzicije našeg suvremenog informacijski usmjerenog svijeta kojom iz razdoblja samo jednog uređaja povezanog na Internet po osobi prelazimo u razdoblje Interneta stvari u kojem po osobi postoji mnoštvo povezanih uređaja. Mnogi od ovih uređaja su senzori koji generiraju specifičnu vrstu strujećih podataka koji su najčešće velike frekvencije, ali uglavnom male veličine.
Vaš zadatak je proučiti i detaljno opisati tehnike strojnog učenja koje se koriste za prognozu vremenskih serija s posebnim naglaskom na mogućnost primjene takvih metoda na strujeće senzorske podatke. Na studijskom slučaju stvarnih senzorskih tokova podataka o vodostajima, protjecajima i padalinama primijenite i evaluirajte odgovarajuće tehnike strojnog učenja za prognozu vremenskih serija.
Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Zadatak uručen pristupniku: 16. ožujka 2018.
Rok za predaju rada: 29. lipnja 2018.

Mentor:

Izv. prof. dr. sc. Krešimir Pripužić

Predsjednik odbora za diplomski rad profila:

Prof. dr. sc. Siniša Srbljić

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Zahvaljujem mentoru prof. dr. sc. Krešimiru Pripužiću na pomoći pri izradi ovog diplomskog rada. Također zahvaljujem svojoj obitelji, kolegama i prijateljima te svima drugima koji su mi bili podrška tijekom studija.

Sadržaj

1. Uvod

Prognoza vremenskih serija je od interesa u mnogim područjima ljudskog djelovanja. Bilo da je od interesa kretanje cijene dionica, prognoza prometa u komunikacijskim mrežama ili predikcija potrošnje električne energije. Danas to područje dobiva sve veći značaj jer smo suočeni s paradigmom Interneta stvari, odnosno s velikom količinom strujećih podataka visoke frekvencije. Prognoza serija stoga prikazuje veliki potencijal za sve organizacije koje posjeduju podatke od značaja.

U klasičnoj statističkoj analizi vremenskih serija, dugi niz godina se prognoziralo korištenjem autoregresivnih modela, poput ARIMA modela. U zadnjim desetljećima porasla je popularnost za metodama strojnog učenja, a posebno za neuronskim mrežama, koje su poznate po tome da rade bolje s velikim količinama podataka. Zbog toga se postavlja pitanje mogu li se ti, novi i uzbudljivi modeli primijeniti na prognoziranje vremenskih serija.

Kroz ovaj rad, opisana je tehnika prognoziranja povratnim neuronskim mrežama kao predstavnik strojnog učenja, te vektorski autoregresivni model, kao predstavnik statističkih modela za prognozu. Nakon toga, implementirane su i eksperimentalno evaluirane obje metode nad stvarnim povijesnim podatcima o padalinama, protjecajima i vodostajima, kako bi se izradila prognoza vodostaja i protjecaja Kupe. U svrhu izrade eksperimenata korišten je programski jezik *Python* te programske knjižnice *Keras* i *Statsmodels*.

U prvom poglavlju su opisane vremenske serije, polja analize i prognoze vremenskih serija, te modeliranje vremenskih serija u problem koji se može rješavati strojnim učenjem. U drugom poglavlju su opisane povratne neuronske mreže. U trećem poglavlju opisani su autoregresivni modeli. U četvrtom poglavlju opisani su izrađeni eksperimenti te prikazana dobivena rješenja.

2. Vremenske serije

Vremenska serija je slijed brojčanih podataka promatrane varijable, koji su odijeljeni jednakim vremenskim intervalima te poredani u vremenskom redoslijedu. U klasičnim problemima strojnog učenja, model je suočen sa skupom primjera, točnije s vektorom $\vec{X}$. Model sve primjere tretira jednako i na temelju njih dolazi do određenih zaključaka. Vremenske serije u učenje modela dovode još jednu vezu između primjera, vremensku dimenziju. Ta poveznica predstavlja ograničenje na model učenja, ali služi i kao izvor dodatnih informacija. Iz ovog razloga, prognoza vremenskih serija se smatra zasebnim i složenijim problemom od drugih problema strojnog učenja, poput regresije i klasifikacije.

Ovisno o tome jesmo li zainteresirani u samu strukturu vremenske serije ili nas zanima prognoza za budućnost, razlikujemo područja analize i prognoze vremenskih serija.

2.1. Analiza vremenskih serija

Primarni interes analize vremenskih serija je objašnjenje i opis serije. Ona se izvodi s pretpostavkom da podatci u seriji sadrže neku dijeljenu unutarnju strukturu. Vremenska serija se dijeli na svoje komponente te se na temelju podataka i relacija između njih izvode različite pretpostavke i gradi model koji opisuje seriju.

Komponente vremenskih serija su:

- **Trend**
  Pravilna uzlazna ili silazna promjena vrijednosti varijable kroz vrijeme. Promjena može biti linearna, eksponencijalna ili neka druga.

- **Sezonalnost**
  Pravilno odstupanje od srednje vrijednosti serije koje se ponavlja unutar nekog redovitog razdoblja koje je kraće od jedne godine. Na primjer, godišnje ili mjesečno.

- **Ciklusi**

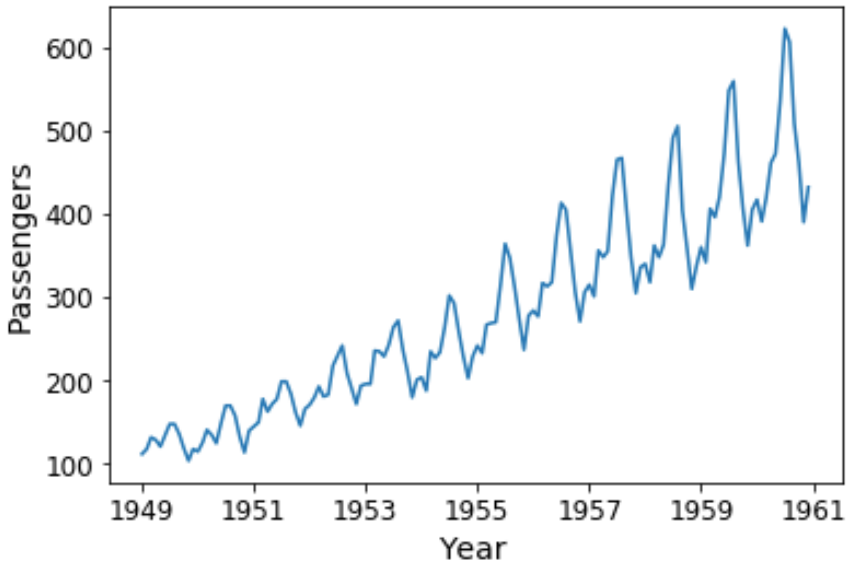
  Pravilno odstupanje od srednje vrijednosti serije koje se događa unutar vremena duljeg od godine dana.

- **Šum**

  Nasumične promjene u seriji u čijem nastajanju ne postoji pravilnost. Šum je najteži za prognoziranje i modeliranje. U nekim literaturama se ova komponenta naziva i rezidualna komponenta.

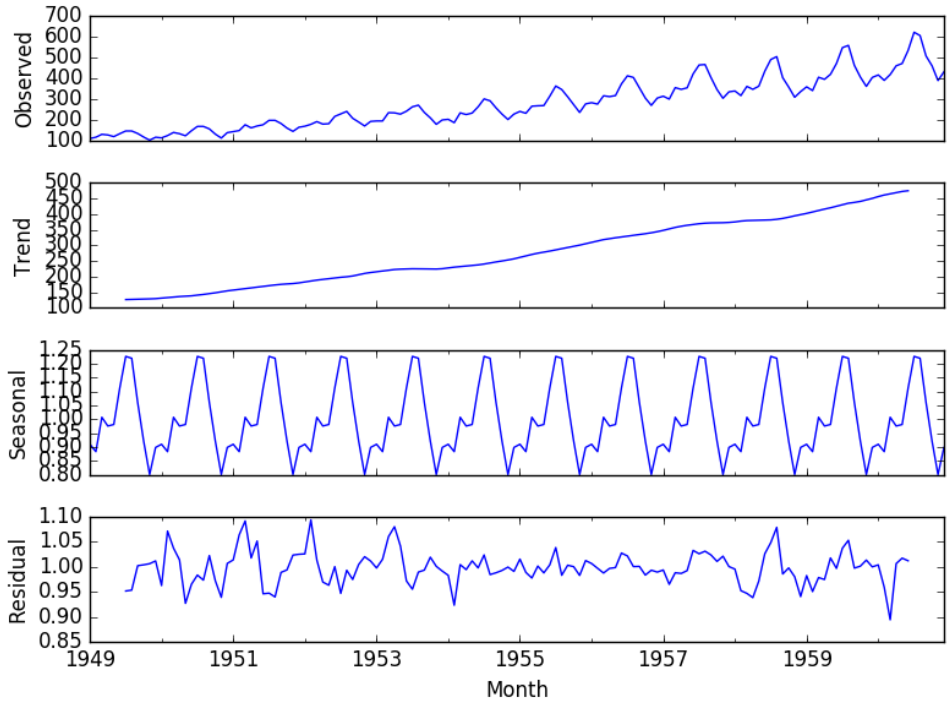
Model se dakle dobiva zbrajanjem ili množenjem pojedinih komponenti vremenske serije. Na primjer:

$$y(t) = \text{trend} + \text{sezonalnost} + \text{šum} \tag{2.1}$$



Slika 2.1 Vremenska serija broja putnika avionom kroz godine [9]

Na slici 2.1 je prikazan skup podataka koji se u analizi vremenskih serija često koristi u edukativne svrhe. Radi se o broju putnika avionom od 1949. do 1960. godine. Skup se sastoji od 144 mjesečnih podataka, te zorno prikazuje komponente trenda i sezonalnosti.



Slika 2.2 Dekompozicija vremenske serije broja putnika avionom kroz godine [9]

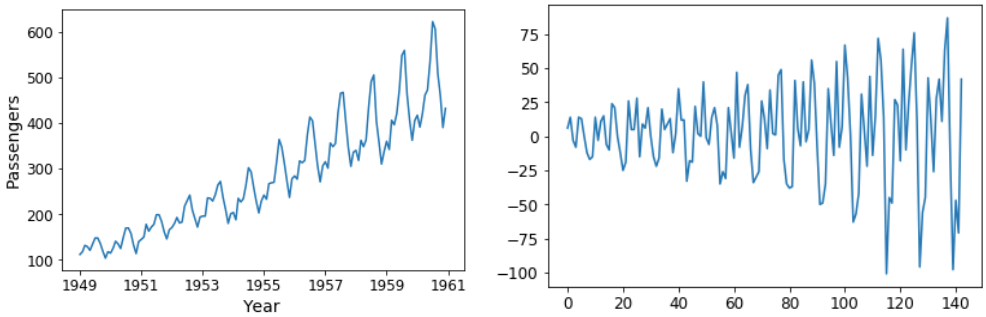
Na slici 2.2 je prikazana dekompozicija serije na trend, sezonalne te rezidualne efekte. Komponente trenda i sezonalnosti su jasno vidljive, no također je interesantna i rezidualna komponenta. Ona pokazuje periode velikih promjena pri početku i kraju originalne serije.

2.1.1. Stacionarnost vremenske serije

Stacionarne vremenske serije su one serije čije su statističke značajke poput srednje vrijednosti, varijance i autokorelacije, konstantne kroz vrijeme. Odnosno takve serije nemaju trend, cikluse niti sezonalne efekte. Većina metoda za prognoziranje i analizu vremenskih serija se ne zna nositi s nestacionarnim vremenskim serijama te pretpostavlja da su ulazne serije stacionarne. Kako bi se mogla izvršiti analiza ili prognoza nad nestacionarnom serijom, potrebno je izvesti transformaciju serije u stacionarnu. Tako postoje različite transformacije koje se primjenjuju na seriju, na primjer: transformacije trenda ili transformacije sezonalnosti. Na

slici 2.2 je prikazana jedna od transformacija, to je dekompozicija serija na komponente.

Jedna od najpopularnijih metoda transformacije je diferenciranje serije. Ona podrazumijeva izračunavanje razlike između susjednih primjera serije. Tako originalna vremenska serija primjera postaje serija razlika između primjera te se nad njom izvodi model. Kako bi se izbjegao trend, često se koristi diferenciranje prvog reda, odnosno diferenciranje prvih susjednih primjera. Tako se mogu vrlo efektivno suzbiti rastuće i padajuće komponente serije. Za suzbijanje sezonalnosti je potrebno odrediti trajanje sezone, te raditi diferenciranje onog reda koliko traje jedna sezona.



Slika 2.3 Serije prije i nakon transformacije diferenciranjem prvog reda [9]

Na slici 2.3 opet je prikazana serija putnika avionom kroz godine, koja očito ima trend i sezonalne efekte. Diferenciranjem prvog reda dobivena je nova serija koja se lakše modelira. Iz diferencirane serije se i dalje vidi da su prisutni sezonalni efekti koji prolaskom vremena imaju sve veće amplitude.

Ako se radi prognoza vremenske serije, nakon učenja modela i prognoze nad podatcima potrebno je napraviti inverznu transformaciju da bi se dobili stvarni podatci npr. inverzno diferenciranje serije.

2.1.2. Prošireni Dickey-Fuller test

Ponekad nam vizualizacija vremenske serije putem grafova ne daje previše informacija, odnosno teško nam je odrediti postoji li trend ili sezonalnost. U tom slučaju je dobro provjeriti jesu li statističke značajke poput srednje vrijednosti serije približno slične u različitim podskupovima serije. Ako još uvijek nismo sigurni, potrebno je napraviti neki test kojim možemo otkriti stacionarnost serije. U tu svrhu se koristi prošireni Dickey-Fuller test jediničnih korijena.

Test počinje od modela:

$$X_t = \varphi X_{t-1} + e_t \tag{2.2}$$

Koji prikazuje vremensku seriju kao funkciju prethodnog koraka serije, te rezidualne komponente u trenutnom koraku. Za istinitost se koriste hipoteze:

- H_0: serija posjeduje jedinični korijen, $\varphi = 1$
- H_1: serija je stacionarna, $\varphi < 1$

Nul-hipoteza tog testa pretpostavlja da dekomponirana vremenska serija sadrži jedinični korijen čime se potvrđuje da je serija nestacionarna. Intuicija iza testa jest da stacionarne serije imaju tendenciju zadržavanja konstantne srednje vrijednosti. Tako bi nakon velikih razina vrijednosti serije trebale slijediti manje vrijednosti. Analiziranjem vremenske serije se na temelju ove pretpostavke dodjeljuju negativne i pozitivne prediktorske vrijednosti te test rezultira vjerojatnosti potvrđivanja nul-hipoteze.

2.2. Prognoza vremenskih serija

Prognoza vremenskih serija se izvodi s ciljem predviđanja budućih vrijednosti serije. Ona se izvodi tako da se gradi model nad povijesnim podatcima te se korištenjem modela predviđaju vrijednosti serije u budućnosti.

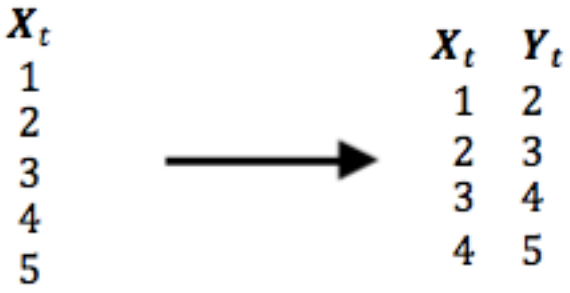
Svaka prognoza vremenske serije je specifična, stoga je na početku postupka prognoziranja bitno detaljno analizirati podatke, izvući zaključke te na temelju njih identificirati najprikladniji model za zadani problem. Pošto je prognoza vremenskih serija područje istraživanja koje se užurbano razvija, bitno je potražiti slične probleme koji su već riješeni od strane stručnjaka kako bi se moglo ubrzati vrijeme odabira modela.

Isto tako, bitno je prikupiti čim više relevantnih i kvalitetnih podataka. Podatci vremenskih sljedova često zahtijevaju čišćenje, skaliranje ili transformaciju. Zatim je bitno odrediti koliki vremenski raspon se predviđa, pošto to direktno utječe na odabir i parametre modela. Također je bitno odrediti frekvenciju podataka koja će se predviđati. Ponekad je potrebno raditi uzorkovanje serije kako bi model bolje generalizirao. Prognoza nad senzorskim podatcima nerijetko zahtijeva zaključivanje na temelju više odvojenih vremenskih serija različitih senzora koji ne moraju biti iste frekvencije. U takvim situacijama je potrebno pažljivo rukovati ulazima u model prognoze. Također, nakon čišćenja podataka i konačnog odabira modela bitno je podijeliti podatke na skup za učenje, validaciju i test, te temeljem njih namjestiti parametre modela. Odabir *hiperparametara* modela je iznimno važan jer on određuje koliko dobro će model prognozirati nad novim i neviđenim primjerima, odnosno kolika mu je moć generalizacije.

2.3. Prognoza vremenskih serija strojnim učenjem

2.3.1. Prognoza kao problem nadziranog učenja

Nadzirano učenje (engl. *supervised learning*) je oblik strojnog učenja u kojemu su nam podatci dani u obliku $(ulaz, izlaz) = (x, y)$, a cilj je pronaći preslikavanje y= $f(x)$ koje opisuje vezu između podataka. Ako je y diskretna varijabla onda govorimo o klasifikaciji, a ako je kontinuirana onda se radi o problemu regresije. Kako bi mogli raditi prognozu nad podatcima, potrebno je vremensku seriju transformirati u problem nadziranog učenja. Serija je slijedni niz podataka, s opcionalnim parametrom vremenske oznake, a u prognozi nas na temelju sadašnje vrijednosti zanima buduća vrijednost varijable. Ovisno o tome koliko koraka želimo gledati u budućnost, toliko koraka serije iz budućnosti možemo dodati u izlaz.



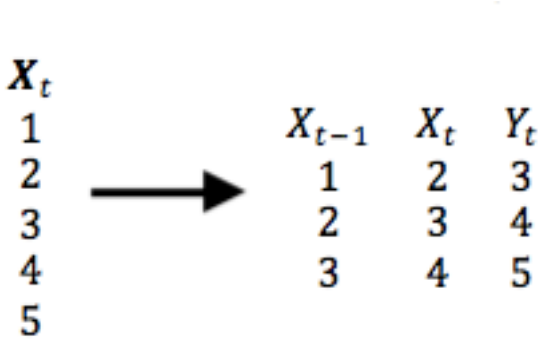
Slika 2.4 Transformacija vremenske serije u problem nadziranog učenja

Kao što je običaj i kod ostalih problema nadziranog učenja, preporučeno je podatke za učenje podijeliti u tri dijela, skup za učenje, skup za validaciju i skup za testiranje. Skup za učenje se koristi za treniranje modela, a na skupu za validaciju se radi evaluacija rješenja kako bi se odabrali idealni *hiperparametri* modela. Nakon što je određen najbolji model, trenira se novi model s istim parametrima, ali na skupu kojeg čine i skup za učenje i skup za validaciju. Nakon toga se koristi skup za testiranje kako bi se konačno odredile performanse modela. Kod klasičnih problema nadziranog učenja, podatci se često i pomiješaju prije nego započne učenje.

Kod prognoziranja vremenskih serija se to očito ne radi, već se podatci slijedno šalju u model, a podatci iz zadnjeg perioda se koriste za testiranje.

2.3.2. Prognoza korištenjem klizećeg prozora

Za razliku od prognoze na temelju trenutne vrijednosti serije, korisno je raditi prognozu na temelju više slijednih vrijednosti serije iz prošlosti. Tako će model imati više informacija o vremenskom kontekstu varijable te često može dolaziti do boljih zaključaka. Ova metoda naziva se metoda klizećeg prozora. Duljina prozora odnosno broj zaostalih primjera koje se gleda, često označavana s *W*, jedan je od *hiperparametara* modela koji se mora podesiti kako bi model dobro prognozirao.



Slika 2.5 Transformacija vremenske serije u problem korištenjem klizećeg prozora

Kao što se vidi, navedene transformacije imaju direktan utjecaj na broj primjera za učenje, o čemu treba voditi računa prilikom izvršavanja postupka učenja.

2.3.3. Strategije za prognozu više koraka unaprijed

U prošlom poglavlju je pokazano kako se problem prognoze vremenskih serija može svesti na problem nadziranog učenja. Međutim, često nam nije dovoljno prognozirati samo jednu sljedeću vrijednost na temelju prošlih. U

mnogim primjenama traži se vrijednost varijable kroz sljedeći tjedan, mjesec ili kroz godinu dana. Ovdje ćemo navesti strategije koje se koriste za prognozu više koraka unaprijed. Pri tome u nastavku smatramo da se prognoza radi za H koraka unaprijed.

2.3.3.1. Direktna strategija

U direktnoj strategiji stvara se H zasebnih modela, od kojih svaki predviđa vremensku seriju u sljedećem koraku. Odnosno svaki model uči prognozu za određeni korak $h, h \in \{1, \dots H\}$.

$$y(t+1)=model1(x(t),x(t-1),\dots,x(t-n)) \tag{2.3}$$

$$y(t+2)=model2(x(t),x(t-1),\dots,x(t-n))$$

…

Ova strategija iziskuje dosta računalne snage, a unosi i dodatnu kompleksnost u izgradnju, zbog učenja H odvojenih modela. Također, pošto se za prognozu koriste odvojeni modeli, ovakav model ne uspijeva raspoznati vezu između susjednih koraka. To je velika mana u prognozi pošto je važna pretpostavka gotovo svake prognoze upravo korelacija između slijednih koraka.

2.3.3.2. Rekurzivna strategija

U rekurzivnoj strategiji, stvara se jedan model koji radi predikciju za jedan korak. Model prognozira vrijednost sljedećeg koraka, a nakon toga se izlaz modela za taj korak koristi kao ulaz u isti model za prognozu sljedećeg koraka. Prognoze se tako dalje stvaraju za H koraka unaprijed.

$$y(t+1) = model(x(t), x(t-1), \dots ,x(t-n)) \tag{2.4}$$

$$y(t+2) = model(y(t+1), x(t), x(t-1), ..., x(t-n+1))$$

Pošto se kao ulaz koriste predviđene vrijednosti, a ne stvarne, greške prognoze se uglavnom gomilaju iz koraka u korak te se performanse ovakvog modela pogoršavaju sa svakim uzastopnim predviđenim korakom.

2.3.3.3. Hibridna direktno-rekurzivna strategija

Hibridna direktno-rekurzivna strategija je nastala udruživanjem elemenata direktne i rekurzivne strategije, a pokušava smanjiti ograničenja koje one zasebno imaju.

Kao i u direktnoj strategiji, stvara se H različitih modela, za svaki korak posebno. No kao ulaz za svaki sljedeći korak se unosi izlaz koraka ispred njega.

$$y(t+1) = model(x(t), x(t-1), \dots, x(t-n)) \tag{2.5}$$

$$y(t+2) = model2(y(t+1), x(t), \dots, x(t-n+1))$$

$$\dots$$

Tako se uspijeva modelirati veza između susjednih primjera, iako je i dalje potrebno dosta računalne snage za izračunavanje modela.

2.3.3.4. Strategija višestrukog izlaza

Strategija višestrukog izlaza (engl. *Multiple output strategy*) zasniva se na izgradnji jednog modela koji odjednom radi prognozu za sve korake u budućnosti. Modeli koji koriste ovu strategiju su u pravilu kompleksniji jer modeliraju korelaciju između nekoliko koraka ulaza i izlaza. Takvi modeli su uglavnom sporiji pri učenju te zahtijevaju mnogo podataka kako bi izbjegli prenaučenost.

$$y(t+1), y(t+2) = model(x(t), x(t-1), \dots, x(t-n)) \tag{2.6}$$

2.3.4. Mjere točnosti prognoze

Tijekom i nakon izgradnje modela prognoze, vrlo je bitno znati ocijeniti točnost modela. Postoje različite mjere koje nam daju različite informacije o naučenom modelu te nas mogu usmjeriti na daljnja poboljšanja.

Najčešće korištene mjere točnosti prognoze vremenskih serija su:

- **Prosječno apsolutno odstupanje**

  Prosječno apsolutno odstupanje (engl. *Mean absolute error, MAE*) dobiva se usrednjavanjem apsolutne vrijednosti greške svakog pojedinog prognoziranog primjera.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{2.7}$$

  Prosječno apsolutno odstupanje nam je zanimljivo zato što je dobiveno odstupanje prikazano u istim jedinicama u kojima je izražena i prognozirana varijabla. Cilj učenja modela je što više smanjiti ovu grešku na skupu za učenje.

- **Korijen srednjeg kvadratnog odstupanja**

  Korijen srednjeg kvadratnog odstupanja *(Root mean squared error, RMSE)* je vrlo često korišten u prognozama serija. Osim što daje iznos odstupanja koji je izražen u jedinicama tražene varijable, kvadriranje odstupanja ima efekt dodavanja težina lošim primjerima. Tako se pridodaje važnost primjerima za koje model jako loše predviđa te se lakše mogu spoznati nedostaci modela.

$$RMSE = \frac{1}{n} \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{2.8}$$

- **Prosječni postotak apsolutnog odstupanja (Mean absolute percentage error, MAPE)**

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \tag{2.9}$$

Prosječni postotak apsolutnog odstupanja ima jednostavnu interpretaciju, govori nam u kojem postotku model, u prosjeku, odstupa od prave vrijednosti. Ipak, ova mjera se rjeđe koristi u praksi, pošto se ne može mjeriti za vremenske serije koje sadrže nulu kao vrijednost. Tako na primjer programska knjižnica *Scikit-Learn* niti ne sadrži implementaciju ove mjere. Još jedan bitan problem jest činjenica da greške prognoza koje su manje od tražene varijable ne mogu prijeći iznos od 100%, dok za greške prognoza koje su veće od tražene varijable ne postoji gornja granica postotka odstupanja. Tako se još kaže i da je MAPE asimetrična mjera, jer zadaje veće kazne prevelikim prognozama, nego što zadaje premalim prognozama. Iz ovog razloga ova metoda nije primjerena za usporedbu različitih modela učenja, čime je ponekad teže odrediti dobar model.

Mjere za prosječno apsolutno odstupanje te srednje kvadratno odstupanje mogu se pronaći u *Python* knjižnici *Scikit-Learn*, te u mnogim drugim knjižnicama za ostale programske jezike.

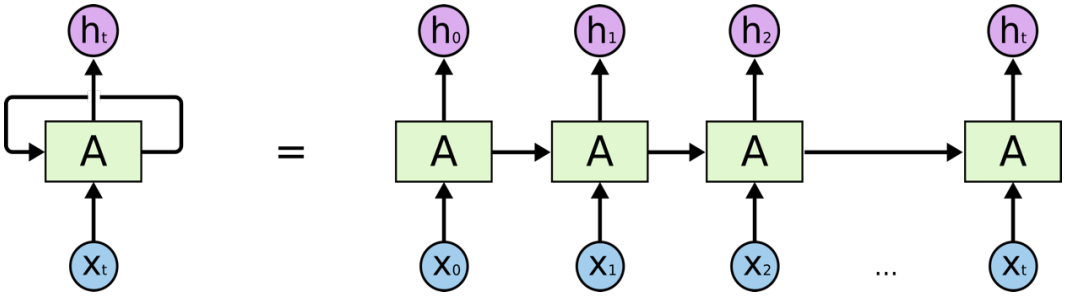
3. Neuronske mreže

Neuronske mreže su u posljednjih nekoliko godina postale jako popularne za rješavanje svih problema s kojima se suočava strojno učenje. Tako su danas duboki modeli neizbježni u poljima poput računalnog vida, raspoznavanju govora te razvoja umjetne inteligencije za samoupravljajuće automobile. Razlog tome jest poboljšanje i pojeftinjenje računalne opreme koja može efektivno izvoditi kompleksne izračune težina neuronskih mreža te povećanje dostupnosti velikog broja podataka s različitih senzorskih i drugih očitanja.

Klasične unaprijedne mreže (engl. *feed forward neural networks)* se mogu nositi s obiljem podataka i vezama između ulaza i izlaza, ali ne uspijevaju pronalaziti korelaciju između slijednih podataka. Mnogi problemi zahtijevaju poznavanje slijednog konteksta, poput predviđanja riječi tijekom pisanja rečenice te prognoze vremenski slijednih podataka. Kako bi u problem unijeli i slijednu, odnosno temporalnu dimenziju, potrebno je koristiti neuronske mreže koje sadrže povratne veze. U ovom poglavlju su objašnjene povratne neuronske mreže, ćelije s dugoročnom memorijom te optimizacijski postupci za učenje takvih mreža.

3.1. Povratne neuronske mreže

Povratne neuronske mreže su posebne vrste neuronskih mreža koje sadrže petlje, odnosno veze unatrag. U zadanom trenutku, izlaz mreže se ne računa samo na temelju trenutnog ulaza mreže, već i na temelju izlaza mreže u prošlom koraku. Tako odluke mreže iz prošlosti utječu na izlaz mreže u budućnosti. Često kažemo da ovakve mreže imaju neku vrstu memorije u kojoj sadrže informacije vezane uz samu seriju podataka. Povratnu mrežu možemo zamišljati kao niz kopija iste mreže, pri čemu ranije kopije sljedećima šalju poruke. Povratne mreže često tako i vizualiziramo, a taj postupak nazivamo “odmatanjem” mreže.



Slika 3.1 Odmatanje povratne neuronske mreže [8]

Prijenos memorije u mreži možemo matematički zapisati kao:

$$h_t = \phi(Wx_t + Uh_{t-1}) \tag{3.1}$$

Gdje je h_t skriveno stanje koje u nekim slučajevima može biti i izlaz mreže. Ono je funkcija ulaza u danom trenutku, x_t koji je pomnožen za težinskom matricom W, te skrivenog stanja h_{t-1} koje je pomnoženo sa svojom težinskom matricom U. Suma ulaza i skrivenog stanja je predana kao ulaz aktivacijskoj funkciji ϕ, za koju se uglavnom koristi logistička sigmoidalna funkcija ili tangens hiperbolni. Kao i ostale neuronske mreže, mreže s povratnim vezama su stohastični algoritmi. Treniranje, odnosno rješenja modela ovise o inicijalnom odabiru težina mreže tako da je pri evaluaciji potrebno nekoliko puta pokrenuti treniranje, te izvršiti usrednjavanje rezultata. Također, kako bi se izgradio model koji dobro rješava konkretan problem, potrebno je odrediti brojne *hiperparametre* mreže, poput broja neurona, broja slojeva te načina inicijalizacije težina u mreži. Za težine ulaznih veza se uglavnom koristi nasumična uniformna distribucija, a iznosi pristranosti se inicijalno postavljaju na nulu. Za težine koje povezuju skriveno stanje između različitih koraka mreže se pak koristi generiranje nasumične ortogonalne matrice.

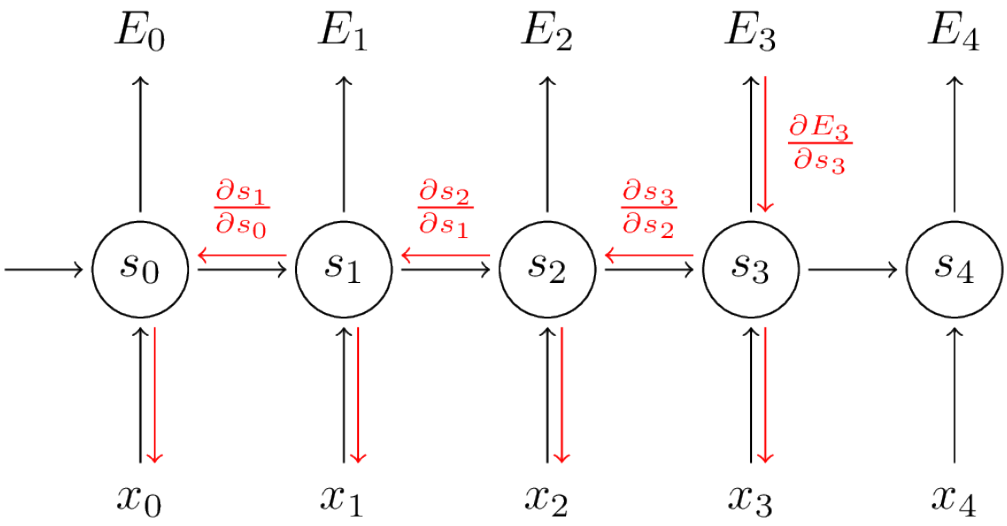
3.1.1. Treniranje povratnih neuronskih mreža

Kod osnovnih unaprijednih mreža, težine skrivenog sloja učimo na osnovu primjera koristeći algoritam propagacije unatrag (engl. *Backpropagation*). U njemu se prvo radi propagacija primjera kroz mrežu čime se dobiva neki izlaz. Nakon toga se računa greška, uspoređujući dobiveni izlaz i stvarni izlaz primjera. Zatim se radi propagacija greške unatrag, računajući derivacije greške o odnosu na težine mreže, te se ovisno o veličini i smjeru greške podešavaju težine mreže. Težine se podešavaju na način koji je svojstven optimizacijskom postupku koji je odabran, a u najjednostavnijem slučaju se radi o gradijentnom spustu. U njegovom radu se težine pomiču u smjeru negativnog gradijenta koji množimo s nekom stopom učenja. Ovaj postupak se radi za sve ispitne primjere. U praksi se češće koristi skupna varijanta ovog postupka (engl. *batch* ili *mini-batch*), u kojoj se kroz mrežu provode skupovi primjera, računa se gradijent greške na temelju skupine viđenih primjera te se tek onda radi promjena težina. Takav algoritam dovodi do nešto stabilnijih rješenja, te zahtjeva manje računalnih operacije jer se promjene težina ne računaju za svaki pojedini primjer.

Mreže s povratnim vezama uče algoritmom koji se malo razlikuje od klasičnog algoritma propagacije unatrag. Unaprijedne neuronske mreže kao ulaz za učenje očekuju tenzor dimenzija *(batch_size, input_dimensions),* odnosno za svaki primjer unutar podskupa primjera za učenje, kao ulaz se očekuje vektor sa svim značajkama (eng. *feature*) u tom primjeru. Za razliku od toga, mreže s povratnim vezama za učenje očekuju trodimenzionalni tenzor oblika: *(batch_size, timesteps, input_dimensions).* Odnosno, jedan primjer za učenje se može sastojati i od više ulaznih koraka.

Razlika u učenju ovakvih mreža jest što se pogreška računa zasebno za svaki vremenski korak, odnosno za svaki ulazni korak serije. Za svaki sljedeći korak, mreža se “odmota” kako bi se izračunao utjecaj greške u trenutnom koraku na težine skrivenog sloja u prošlim koracima. Nakon što

su akumulirane greške kroz svaki korak, radi se ažuriranje težina mreže. Konceptualno, svaki dodatni korak u mreži se tijekom učenja tretira kao dodatni sloj mreže. Ovaj algoritam postaje računalno sve zahtjevniji s povećanjem koraka.



Slika 3.2 Propagacija unatrag kroz vrijeme [6]

Ovaj algoritam nazivamo propagacijom unatrag kroz vrijeme (engl. *Backpropagation through time, BPTT*). Postoji i skraćena varijanta ovog algoritma (engl. *Truncated backpropagation through time, TBPTT*) koja je računalno lakša za izvođenje jer se propagacija unatrag kroz vrijeme ne izvodi za svaki pojedini korak. Ona podrazumijeva odabir parametara *k1* i *k2*. Parametar *k1* određuje koliko koraka se ide unaprijed prije izračuna propagacije unatrag, a parametar *k2* određuje koliko koraka se ide unatrag pri kalkulaciji greške. Algoritam se označava kao TBTT(*k1*, *k2*), a TBTT(*n*,*n*), gdje je n ukupna duljina koraka ulaza, je zapravo klasični algoritam BPTT.

3.1.2. Aktivacijske funkcije

Aktivacijske funkcije su funkcije koje odlučuju hoće li neuron biti "aktiviran" ili ne, odnosno odlučuju o izlazu pojedinog neurona. One se evaluiraju nad sumom umnoška vektora težina i ulaza te pristranosti (engl. *bias*) u tom sloju.

$$y = \phi(Wx + b) \tag{3.2}$$

Aktivacijske funkcije u neuronske mreže uvode nelinearnost, odnosno omogućavaju pronalaženje nelinearnih veza među podatcima.

Aktivacijske funkcije koje se koriste u povratnim neuronskim mrežama su:

1. **Sigmoidalna funkcija**

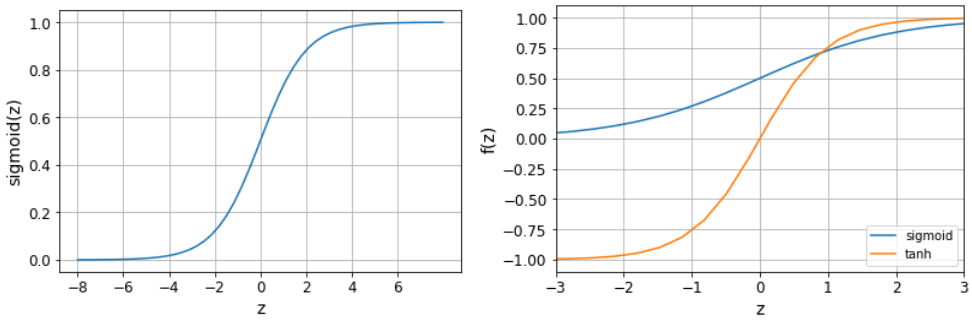
   Sigmoidalna funkcija je funkcija koja preslikava realnu domenu u skup (0,1) čime je dobra za klasifikacijske primjene jer ima probabilističku interpretaciju. Ona je monotona i derivabilna na cijelom području domene.

$$\phi(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.3}$$

2. **Tangens hiperbolični**

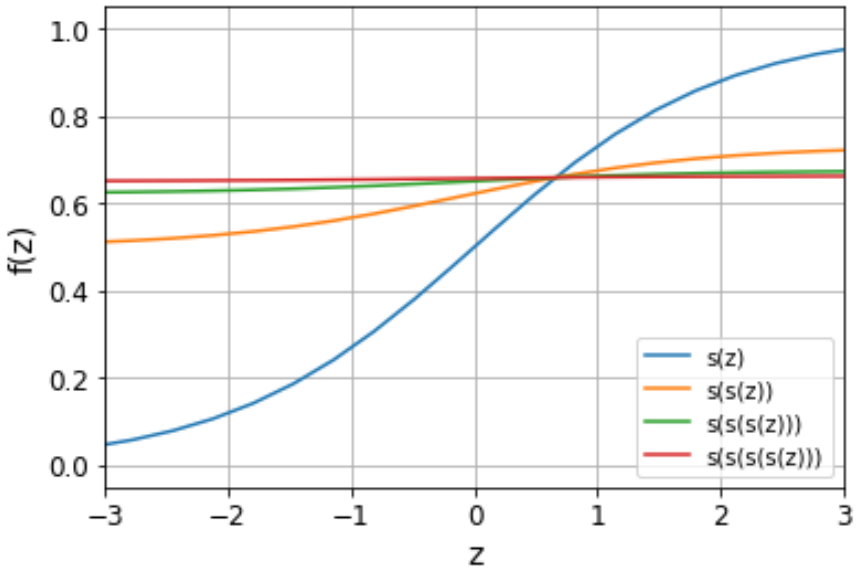
   Tangens hiperbolični je zapravo sigmoidalna funkcija skalirana na interval (-1, 1). Ona je također monotona i derivabilna na cijelom području domene i oko nule je strmija od sigmoidalne funkcije. Ponekad nam je ona prikladnija jer nam je potrebno vidjeti razliku između negativnih ulaza i ulaza koji su blizu nule.

$$\phi(z) = \tanh(z) = \frac{2}{1 + e^{-2z}} - 1 = 2\sigma(2z) - 1 \tag{3.4}$$



Slika 3.3 Simgoidalna funkcija(lijevo) i hiperbolična funkcija (desno)

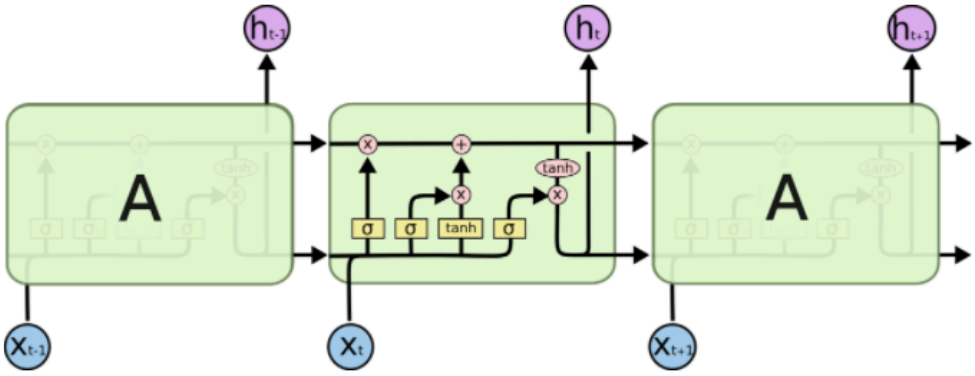
Povratne neuronske mreže imaju problema s takozvanim nestajućim gradijentom do kojeg dolazi jer se pomak gradijenta računa koristeći pravilo ulančavanja. Konkretno, u propagaciji unazad kroz vrijeme, gradijenti aktivacijske funkcije se mnogo puta množe. Pošto su gradijenti uvijek manji od jedan, uzastopno množenje takvih vrijednosti dovodi do jako malih pomaka težina. Također, za sigmoidalnu funkciju, sve vrijednosti koje su blizu 0 ili 1 imaju vrlo mali gradijent te još više umanjuju iznos pomaka. Ova pojava predstavlja veliki problem za duboke mreže.



Slika 3.4 Uzastopno evaluiranje sigmoidalne funkcije

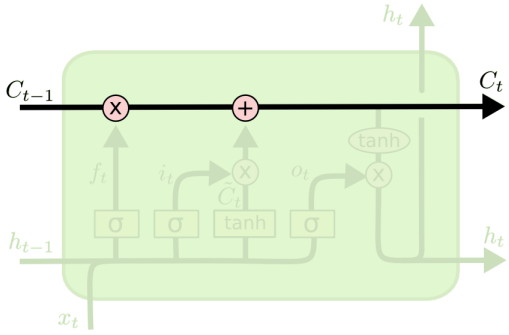
Na slici 3.4 je prikazan izgled sigmoidalne funkcije, s uzastopnim primjenjivanjem iste sigmoidalne funkcije na njezin izlaz. Vidimo da se s daljnjim primjenjivanjem funkcije nagib smanjuje, odnosno nestaje.

3.2. Ćelija s dugoročnom memorijom



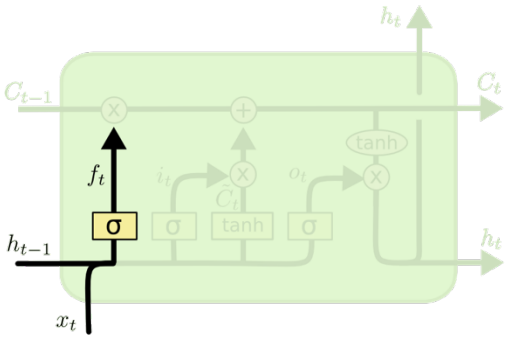
Slika 3.5 Ćelija s dugoročnom memorijom [8]

Ćelija s dugoročnom memorijom (engl. *Long Short Term Memory, LSTM*) je poseban građevni blok za stvaranje povratne neuronske mreže koji omogućuje učenje dugotrajnih zavisnosti u podatcima. Klasična ćelija povratnih mreža koje smo do sada spominjali, sadrži samo jedan neuronski sloj s određenom aktivacijskom funkcijom. LSTM ćelija u sebi sadrži složeniju arhitekturu koja se sastoji od četiri različitih slojeva



Slika 3.6 Stanje unutar LSTM ćelije [8]

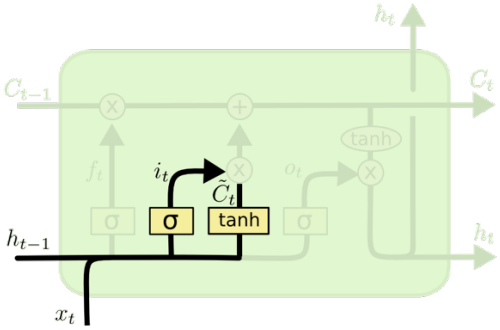
Glavna ideja iza LSTM ćelije jest modifikacija stanja unutar ćelije, koje je prikazano gornjom trakom na slici 3.6. LSTM ćelija ima mogućnost dodavanja i oduzimanja informacija iz stanja, putem struktura koje nazivamo propusnice ili vrata (engl. *gates*). Propusnice se sastoje od neuronskog sloja sa sigmoidalnom aktivacijom, čiji je izlaz između 0 i 1, te one određuju u kojoj mjeri se informacija propušta u stanje. LSTM sadrži 3 ovakvih propusnica.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Slika 3.7 Propusnica zaboravljanja [8]

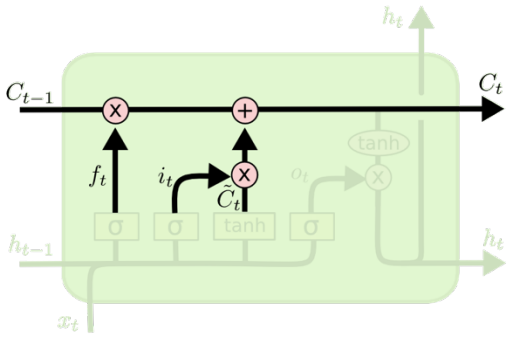
Prva propusnica, koju još nazivamo i propusnicom zaboravljanja (engl. *forget gate*), sastoji se od sigmoidalnog sloja koji kao ulaz dobiva spojeni vektor vrijednost od trenutnog ulaza x_t, te izlaza iz prošlog koraka h_{t-1}. Njegova funkcija je odbacivanje informacija iz prethodnog stanja, odnosno modifikacija dugoročne memorije.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Slika 3.8 Ulazna propusnica [8]

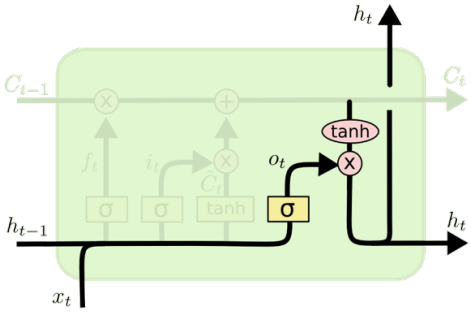
U sljedećem koraku se biraju nove informacije koje će se zapisati u stanje ćelije. Ovo se odvija u dva dijela. Prvo se koristi sigmoidalni sloj kojeg nazivamo ulaznom propusnicom (engl. *input gate*). Ona odabire koje vrijednosti ćemo ažurirati. Nakon toga se nalazi jedan sloj s aktivacijskom funkcijom tangens hiperbolični koji pak stvara vektor vrijednosti kandidata C_t, koje bi mogle biti dodane u trenutno stanje. Kombinacija izlaza ta dva sloja se dodaje u trenutno stanje, čime se ažurira memorija ćelije.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Slika 3.9 Izračunavanje stanja u LSTM ćeliji [8]

Na temelju propusnice zaboravljanja i ulazne propusnice izračunava se novo stanje C_t koje direktno putuje u ćeliju sljedećeg trenutka.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Slika 3.10 Izlazna propusnica [8]

Izlaz ćelije se dobiva kao funkcija trenutnog stanja C_t i trenutnog ulaza. Spojeni vektor ulaza i izlaza iz prošlosti opet prolazi kroz sigmoidalni sloj, te se množi s trenutnim stanjem koje prolazi sloj s aktivacijom tangens hiperbolični. Ovu propusnicu nazivamo i izlaznom propusnicom (engl. *output gate*).

3.3. Optimizacijski postupci za učenje neuronskih mreža

Učenje neuronske mreže se svodi na optimizaciju funkcije pogreške, npr. MSE ili MAE. Težine u neuronskim slojevima se mijenjaju dok se ne minimizira ta funkcija. U najosnovnijem slučaju, koji je opisan i ranije, koristi ste gradijentni spust. On u svakom koraku podrazumijeva pomak u negativnom smjeru gradijenta, pri čemu se korak množi s parametrom α ili stopom učenja. Isti parametar se koristi cijelo vrijeme i za sve težine mreže. Pokazalo se da to nije najefektivniji način, te danas postoje bolji i brži načini optimizacije funkcije pogreške.

3.3.1. Adam

Optimizacijski postupak Adam (engl. *Adaptive moment estimation*) je alternativa gradijentnom spustu koja je jednostavna za implementaciju, računalno efektivna te vrlo brzo dolazi do dobrih rješenja. Adam je nastao spajanjem elemenata dva također učinkovita optimizacijska postupka:

- **AdaGrad (Adaptive Gradient Algorithm)**

  Algoritam koji koristi posebne stope učenja za svaki parametar te tako donosi bolje rezultate s rijetkim gradijentima. Ima veliku primjenu u procesiranju prirodnog jezika te računalnom vidu.

- **RMSProp (Root Mean Square Propagation)**

  Algoritam koji isto koristi posebne stope učenja za pojedine parametre koje se mijenjaju tijekom učenja, ovisno o trenutnom prosjeku veličina gradijenta. Stopa učenja se dakle prilagođava kroz vrijeme čime ovaj algoritam uspijeva brže putovati kroz padine funkcija cilja. Ovaj algoritam je vrlo dobar na problemima za koje se radi *online* učenje ili sadrže puno šuma.

Algoritam Adam također koristi zasebne stope učenja za svaki parametar, ali ih ažurira na temelju prvog i drugog momenta gradijenta. On koristi parameter α, koji služi kao početna stopa učenja svih težina, te uvodi parametre β_1 i β_2 . Algoritam izračunava eksponencijalni pomični prosjek gradijenta i kvadratnog gradijenta, a ti parametri služe kao stope propadanja tih prosjeka. Za parametar α se uzimaju niske vrijednosti (npr. 0.001). Više vrijednosti dovode do bržeg inicijalnog učenja, dok težine još nisu promijenjene, a to nam nije uvijek dobro jer problem može zaglaviti u lokalnom optimumu. Za beta parametre se uzimaju vrijednost iz skupa $\langle 0.9, 1 \rangle$.

4. Statistički modeli za prognozu

U statistici i ekonometriji se za analizu i prognozu vremenskih serija već dugi niz godina koristi model ARIMA (engl. *Autoregressive integrated moving average*).

On se sastoji od tri dijela:

- **Autoregresivni model (AR)** – model koji pretpostavlja da se vremenska serija ponaša kao linearna funkcija svojih zaostalih (engl. *lagged*) vrijednosti
- **Integrirana komponenta (I)** – dio modela koji radi diferencijaciju nad originalnom vremenskom serijom
- **Model pomičnog prosjeka (MA, engl. Moving average)** – model koji pretpostavlja da se greška, odnosno rezidualna komponenta serije ponaša kao linearna funkcija grešaka iz prošlosti

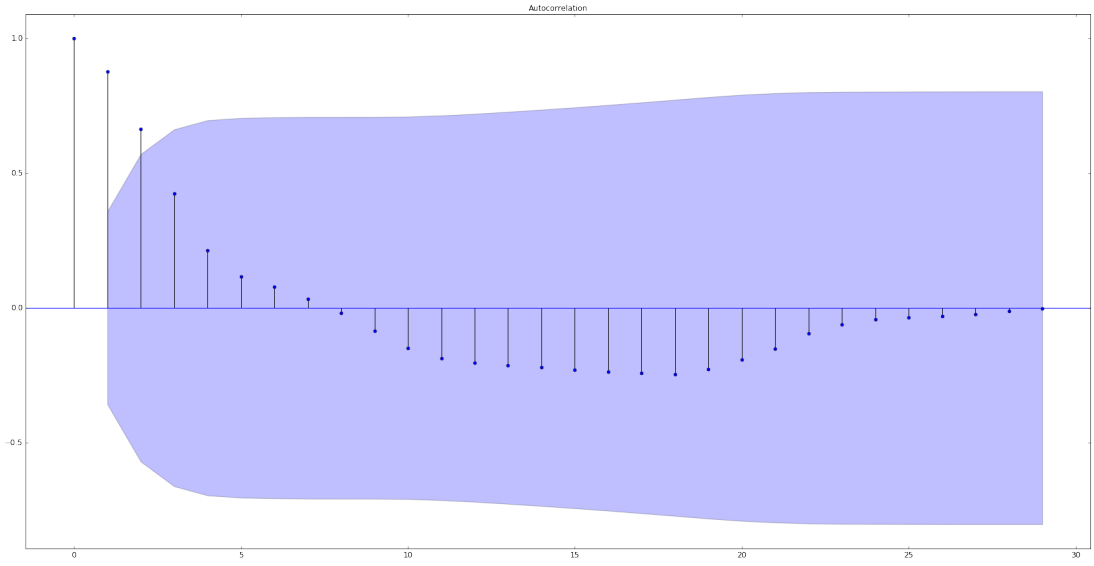
Ovaj model se često specificira kao ARIMA(p,q,d), pri čemu su p, q i d parametri pojedinih komponenata modela. P je broj zaostalih vrijednosti serije nad kojima se vrši autoregresivni model, q je broj zaostalih grešaka iz povijesti nad kojima se pokreće model pomičnog prosjeka, a d je stupanj diferencijacije vremenske serije. Također postoji i sezonalna verzija modela ARIMA koja se zove SARIMA, te sadrži još tri parametra koji se odnose na sezonalnu verziju prije opisanih parametara. Također, ponekad nije potrebno raditi sve modele u kombinaciji, već se mogu koristiti i samo AR ili MA modeli. Ako pak radimo s više varijabli, odnosno s vektorom vremenskih serija, mogu se koristiti VAR, VMA ili VARIMA modeli.

Najveći problem pri analizi i prognozi korištenjem modela ARIMA jest odabir parametara p, q, d, koji bi trebali biti odabrani od strane stručnjaka na temelju analize komponenata serije. Učenje VARMA modela se pokazalo kao jako dugotrajan i računalno zahtjevan proces, pa je sklopu ovog diplomskog rada korišten model VAR.

4.1. Autokorelacija i parcijalna autokorelacija

Važna metoda koja se koristi za odabir parametra p u autoregresivnom modelu, jest analiziranje grafova autokorelacijskih i parcijalnih autokorelacijskih funkcija.

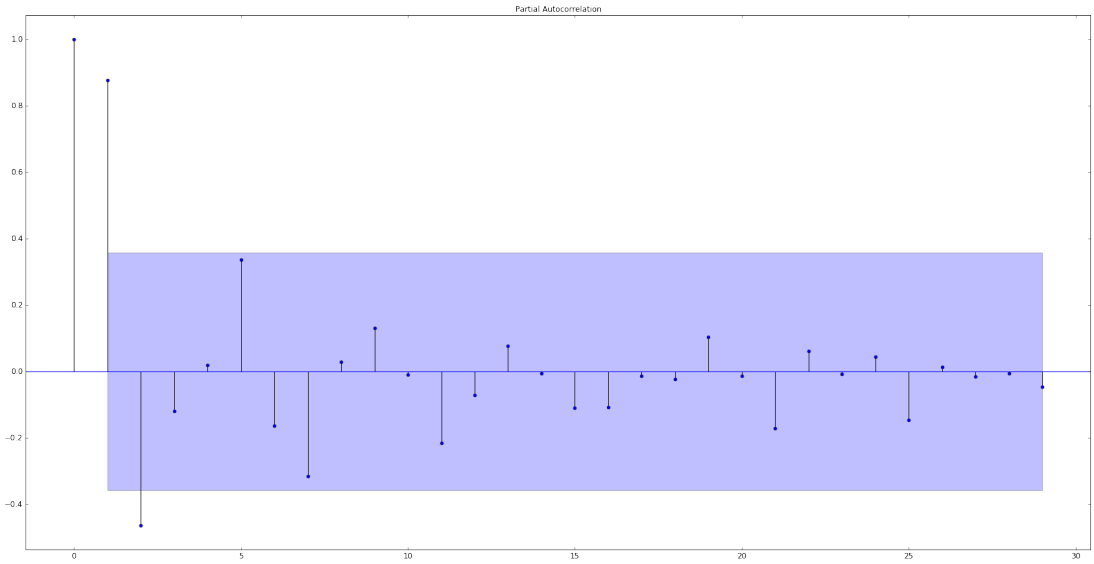
Autokorelacijska funkcija (ACF) nam govori o korelaciji vremenske serije sa zaostalim vrijednostima same sebe. ACF nam za svaki korak unazad daje vrijednost između -1 i 1, odnosno za svaki korak unazad govori u kojoj mjeri su vremenski odijeljene vrijednosti serije korelirane. Autokorelacijska funkcija tipično ima veliku vrijednost za prvih nekoliko koraka, a nakon toga polako pada prema nuli.



Slika 4.1 Primjer ACF grafa

Parcijalna autokorelacijska funkcija (PACF) također određuje korelacije između dva koraka unutar serije, ali to radi tako da poništava efekte svih koraka između njih. Za razliku od grafa ACF, graf PACF nakon koraka koji ima visoku korelaciju vrlo brzo pada u nulu, jer se poništavanjem utjecaja koraka između njih smanjuje lančana reakcija koju prijašnji koraci

imaju na sljedeće. Tako se može bolje opaziti značaj pojedinog zaostalog koraka serije.



Slika 4.2 Primjer PACF grafa

Na slikama 4.1. i 4.2 su prikazani ACF i PACF grafovi iste serije. Na slikama se vide intervali pouzdanosti koji su obojani ljubičastom bojom. Oni sugeriraju da su koraci čije se vrijednosti nalaze izvan obojanog područja značajno korelirani s trenutnom vrijednošću varijable.

4.2. Autoregresivni model

Autoregresivni model pretpostavlja da izlazna varijabla linearno ovisi o svojim prošlim vrijednostima, te rezidualnoj komponenti u trenutnom koraku. Parametar p modela AR(p) označava broj zaostalih varijabli nad kojima se evaluira model.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t, \tag{4.1}$$

Pri tome su $\varphi_1, \ldots, \varphi_p$ parametri modela, c je konstanta, a ε_t je rezidualna komponenta. Formula (4.1) izgleda kao obična linearna regresija nad više varijabli, pri čemu su varijable zapravo vrijednosti serije X iz prethodnih vremenskih koraka.

Postoje različiti načini učenja AR modela, no najčešće korišteni su metoda najmanjih kvadrata te maksimizacija funkcije izglednosti. Velike prednosti autoregresivnog modela su kratko vrijeme učenja te dobre performanse čak i nad malim skupovima podatka. Pošto AR modelira veze između koraka serije u različitim vremenskim trenucima, on radi i nad nestacionarnim vremenskim serijama, odnosno ne zahtijeva transformacije nad originalnim vremenskim sljedovima.

4.3. Autoregresija vektora

Autoregresija vektora ili VAR model je generalizacija AR modela, koja pronalazi linearnu vezu između slijednih koraka na osnovu više različitih vremenskih serija. On tretira sve varijable jednako, odnosno pretpostavlja da sve ulazne varijable imaju efekt jedna na drugu. Takve varijable nazivamo još i endogenim varijablama. Za modeliranje VAR-om nije potrebno veliko poznavanje statistike unaprijed, već je jedino bitno imati skup varijabli za koje se pretpostavlja da imaju međusobno djelovanje jedna na drugu kroz vrijeme.

VAR(p) model se gradi zbrajanjem vektorskih komponenti:

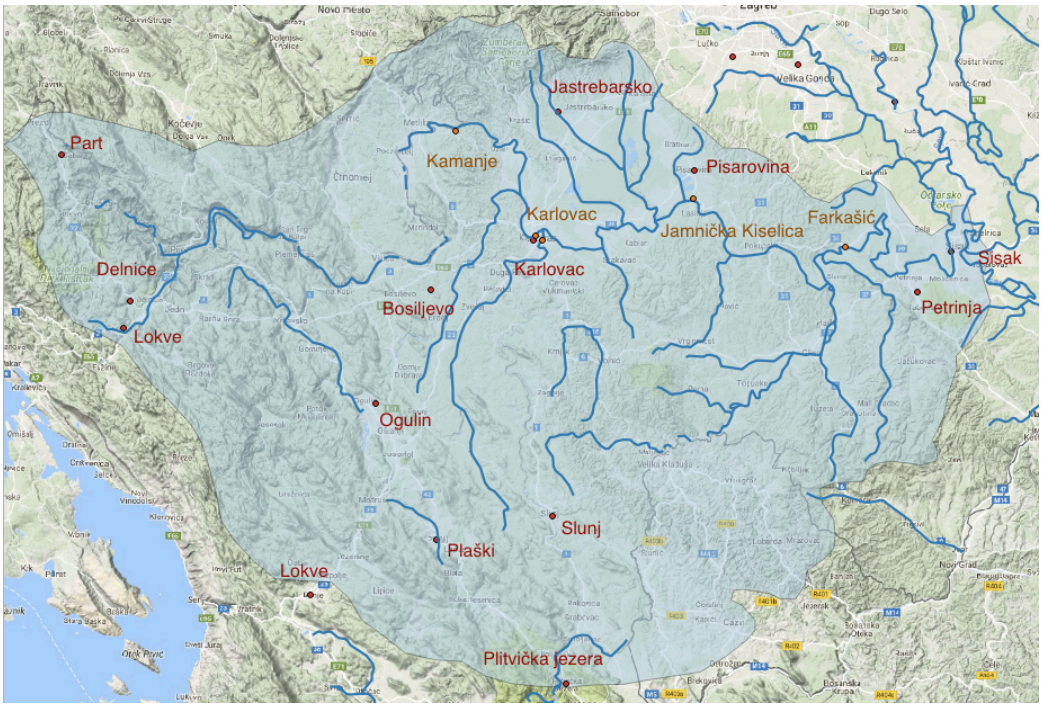
$$X_t = c + \sum_{i=1}^{p} A_i X_{t-i} + \varepsilon_t \tag{4.2}$$

Pri tome su $A_1, \dots, A_p$ matrice koje sadrže parametre modela, c je konstanta, a ε_t je rezidualna komponenta u trenutnom koraku. Model VAR zapravo za svaku varijablu stvara posebnu jednadžbu, a do rješenja jednadžbi dolazi metodom najmanjih korijena

Jednom izgrađeni VAR model može se koristiti za prognozu. Pri tome on generira prognozirane vrijednosti za svaku varijablu posebno. Pri prognozi unaprijed koristi rekurzivnu strategiju. Odnosno, vektor prognoza u trenutku t se koristi za prognoziranje vrijednosti vektora u trenutku t+1. *Hiperparametri* modela autoregresije vektora su broj koraka unazad koji se gledaju pri stvaranju jednadžbe *p*, te broj endogenih varijabli *K*. U praksi je pokazano da se bolje prognozira korištenjem manjeg broja varijabli. U sklopu ovog rada korištena je implementacija modela VAR iz programske knjižnice *Statsmodels*.

5. Prognoza vodostaja i protjecaja Kupe

Zadatak praktičnog dijela ovog rada bio je primijeniti tehnike prognoziranja vremenskih serija na prognozu vodostaja i protjecaja Kupe. Pri tome su korišteni podatci dobiveni od Državnog Hidrometeorološkog zavoda.



Slika 5.1 Hidrološke i meteorološke stanice u porječju Kupe

Korišteni su podatci s 4 hidrološke stanice duž toka rijeke Kupe u Republici Hrvatskoj. To su stanice Kamanje, Karlovac, Jamnička Kiselica i Farkašić. Podatci se sastoje od dnevnih mjerenja vodostaja (izraženih u cm) i protjecaja na stanicama (izraženih u m^3/s), od 2009. godine, zaključno s 2014. godinom. Uz to, za prognozu su bili dostupni i podatci o dnevnim oborinama (izraženi u mm), u istom periodu, izmjerenim na brojnim meteorološkim stanicama uključivši Delnice, Lokve, Bosiljevo, Ogulin, Karlovac, Jastrebarsko i druge. Ideja je bila koristiti podatke uzvodno kako bi se prognozirao vodostaj i protjecaj nizvodno kroz tok rijeke. Na slici 5.1. prikazano je porječje Kupe u Republici Hrvatskoj te korištene stanice. Crvenom bojom su označene meterološke, a narančastom hidrološke stanice.

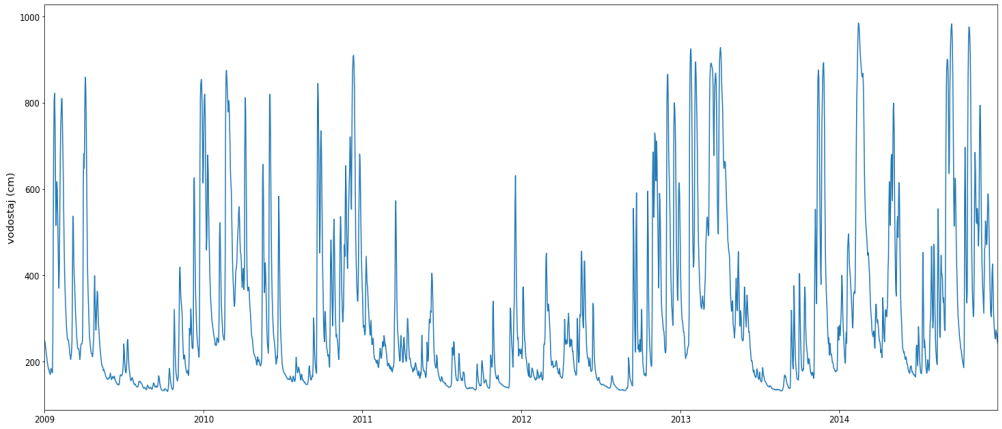
Tablica 5.1 Hidrološke postaje i direktno uzvodne meteorološke postaje

Hidrološka postaja	Meteorološke postaje
Kamanje	Parg, Delnice, Lokve, Bosiljevo
Karlovac	Ogulin, Karlovac
Jamnička Kiselica	Plaški, Slunja, Pisarovina
Farkašić	-

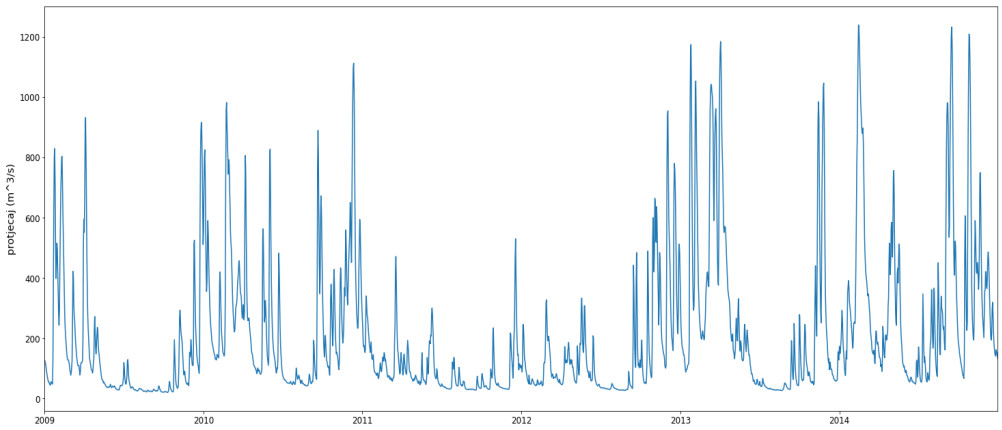
U tablici 5.1 su navedene korištene hidrološke postaje, poredane nizvodno u odnosu na tok, te meteorološke postaje koje su njima direktno uzvodno. Osim tih meteoroloških postaja, za predviđanje vremenskog slijeda određene postaje, na raspolaganju su i sve hidrološke postaje koje su njoj uzvodno, te meteorološke postaje koje utječu na uzvodne hidrološke postaje. Na ovaj način smo dobili velik broj ulaznih sljedova i mogućih različitih konfiguracija prognoze. Cilj je dakle bio raditi multivarijatnu prognozu vremenskih sljedova.

5.1. Analiza i obrada podataka

Predobrada podataka je podrazumijevala čišćenje, identifikaciju i odabir podataka koji su potpuni i relevantni za problem kao i oblikovanje podataka u format koji je pogodan za analizu i prognoziranje. Za rukovanje podatcima korištene su strukture podataka iz knjižnice *Pandas*. Nakon toga je izvršena detaljna analiza podataka kako bi se izvukle dobre pretpostavke za izradu prognoze.

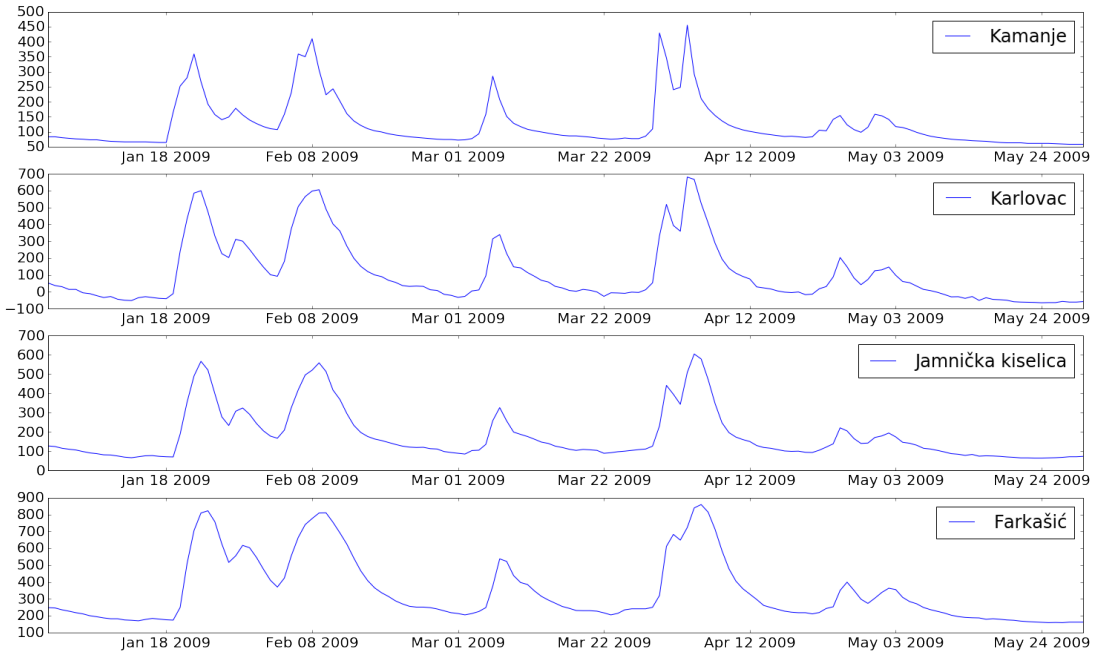


Slika 5.2 Vremenska serija vodostaja Kamanje



Slika 5.3 Vremenska serija protjecaja Kamanje

Na slikama 5.2 i 5.3 su prikazane serije vodostaja i protjecaja Kupe na hidrološkoj stanici Kamanje, kroz svih šest godina analize. Na serijama se ne vide obilježja trenda niti sezonalnih pojava, već ih karakteriziraju samo povremena visoka i kratkotrajna odstupanja od srednjih vrijednosti. Osim toga, može se uočiti da su serije protjecaja i vodostaja vrlo slične, odnosno imaju odstupanja u istim vremenskim trenucima. Ova opažanja nas navode na mišljenje da pri multivarijatnoj prognozi ne moramo koristiti baš sve vremenske sljedove, jer će se modeli lakše trenirati ako maknemo redundantne značajke.



Slika 5.4 Prvih 5 mjeseci nizvodno poredanih vremenskih serija vodostaja

Iscrtani su i grafovi svih serija duž toka rijeke i iz početka su podatci izgledali vrlo slični. Smanjenjem vremenske skale na jednu godinu uočeno je da se na serijama itekako opaža odljev vode s ranijih stanica na daljnje. Kako bi utvrdili da su vremenske serije stacionarne, te da nad njom nisu potrebne dodatne transformacije prije obrade modelom, izračunata je srednja vrijednost i varijanca nad polovicama podataka iz svake vremenska serije.

Tablica 5.2 Statističke značajke podskupova vremenskih serija

Postaja	**Podatci iz 2009.-2011.**		**Podatci iz 2012.-2014.**	
	Srednja vrijednost	Varijanca	Srednja vrijednost	Varijanca
Kamanje	281.0	29776	346.5	46257
Karlovac	136.9	27077	199.32	45119
Jamnička kiselica	57.17	24947	113.36	39570
Farkašić	102.65	5155	125.62	7324

Tablica 5.2 nam govori da ipak postoji neki pomak, odnosno povećanje u srednjim vrijednostima, u drugoj polovici promatranog perioda. Međutim, velike promjene u varijanci nas navode na mišljenje da je u drugoj polovici perioda bilo više pojedinačnih odstupanja, odnosno povremene rezidualne pojave su bile ili češće ili većeg intenziteta (ili oboje).

Za svaki slučaj, nad svakom ulaznom serijom izvršen je i prošireni Dickey-Fuller test, kako bi bili sigurni da se radi o stacionarnim serijama. On nam je dao jednoznačne rezultate za svaku vremensku seriju iz našeg skupa serija. Izlazne p-vrijednost svakog testa bile su jako blizu nuli. P-vrijednost testa je zapravo vjerojatnost da je nul-hipoteza potvrđena, odnosno da je serija nestacionarna. Kako su naše vrijednosti sve jako blizu nule, možemo zaključiti da su sve serije stacionarne. Ovaj podatak nam je vrlo važan jer nam govori kako nije potrebno raditi diferenciranje nad serijom, već je dovoljno raditi prognozu nad originalnim podatcima.

Iako je u ovom poglavlju opisana samo analiza serija vodostaja Kupe, na isti su način obrađeni i podatci o protjecajima te padalinama. Utvrđeno je da oni također predstavljaju stacionarne serije.

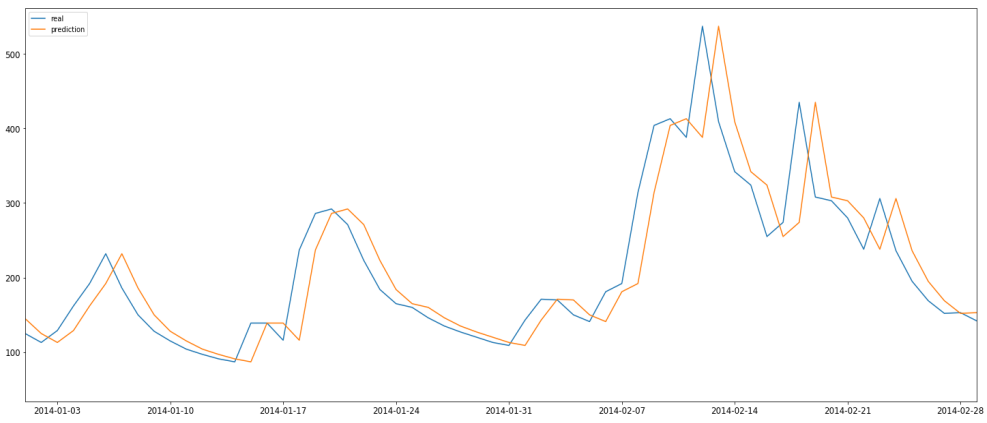
Prije učenja neuronskom mrežom, svi podatci su normalizirani, odnosno skalirani na interval [0,1] zbog olakšavanja učenja modela. Za učenje vektorskom autoregresijom skaliranje nije bilo potrebno.

5.2. Prognoza jednog koraka

Pošto imamo 6 godina podataka, za učenje modela ćemo koristiti prve 4 godine podataka, 5. godinu ćemo koristiti kao skup podataka za validaciju, a zadnja godina će se koristiti za testiranje izgrađenog modela.

5.2.1. Osnovica prognoze

Prije nego što se počne s implementacijom prognoze, dobro je napraviti neku vrstu osnovne (engl. *baseline*) odnosno naivne prognoze do koje se dolazi jednostavno, a predstavlja nam gornju granicu pogreške naše prognoze. Uspoređujući dobivene rezultate i rezultate osnovne prognoze, možemo brzo dobiti dojam koliko dobro naš model radi, te odbaciti loše modele. Kao osnovicu za prognozu sam koristio model ustrajnosti (engl. *persistence model*). On je vrlo jednostavan, jer za svaki sljedeći dan serije prognozira da će vrijednost serije biti ista kao stvarni iznos serije prošlog dana. Nakon što je napravljen takav model, izračunate su greške koje takav model radi na skupu podataka za učenje, kako bi imali referentnu vrijednost za procjenu izgrađenih modela.



Slika 5.5 Osnovica prognoze vodostaja Kamanje

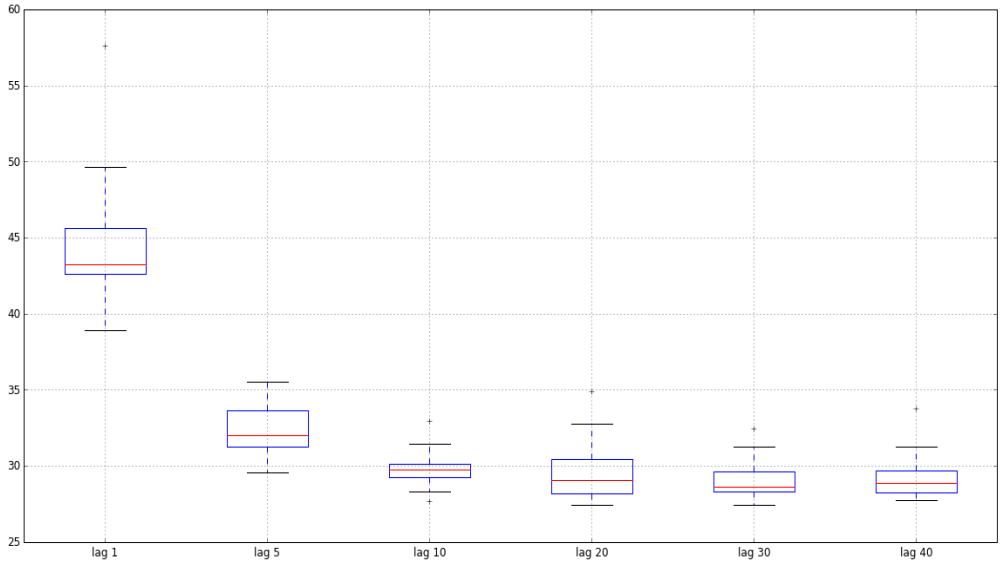
Na slici 5.5. je prikazana osnovna prognoza vodostaja Kamanje. Narančastom bojom je prikazana prognozirana, a plavom originalna serija.

5.2.2. Prognoza neuronskom mrežom

Za svaku prognozu je izgrađen model približno slične arhitekture. Modeli su se sastojali od jednog skrivenog LSTM sloja s različitim brojem neurona (eksperimentalno određeno za pojedini model), te jednog potpuno povezanog sloja koji izlaze iz skrivenog sloja transformira u realnu vrijednost. Pokušane su i arhitekture s dubljim mrežama (ugniježđeni LSTM slojevi), no nisu opažena neka značajna poboljšanja. Modeli su izgrađeni korištenjem programske knjižnice *Keras*, koji interno koristi knjižnicu *Tensorflow*. Pokretanje se izvršavalo na procesoru, odnosno na serveru s 8 jezgri brzine 2.0 GHz te 32 GB RAM memorije.

5.2.2.1. Odabir veličine prozora

Prvi hiperparametar kojeg je bilo potrebno odrediti jest veličina ulaznog prozora podataka. Cilj je bio pronaći dovoljno malu veličinu prozora koja daje dobre rezultate, pošto se povećanjem prozora povećava i računalna složenost, odnosno vrijeme treniranja modela. Testirane su performanse modela s 1, 5, 10, 20, 30 i 40 ulaznih koraka. Testiranje je izvršeno za vremensku seriju vodostaja Farkašić, a kao ulaz su korišteni svi podaci iz hidroloških stanica uzvodno te nekoliko uzvodnih meteoroloških postaja. Testovi su izvođeni nad serijom vodostaja Farkašić s 200 epoha treniranja.

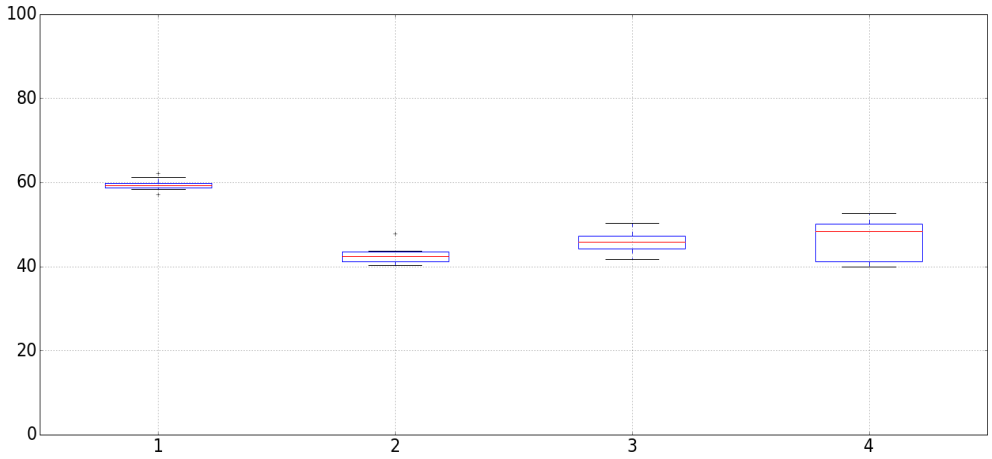


Slika 5.6 Greška prognoze u ovisnosti o veličini prozora

Pošto je treniranje neuronske mreže stohastičan proces, za svaki odabir prozora je pokrenuto 20 treninga modela te je na temelju svih podataka iscrtan kutijasti dijagram (engl. *box-plot*) koji je prikazan na slici 5.6. Na y osi modela se nalazi greška RMSE. Dijagram pokazuje da model ima najmanju grešku s 30 koraka, te je upravo ta veličina prozora korištena u daljnjoj prognozi.

5.2.2.2. Odabir ulaznih serija za prognozu

Pošto smo imali velik broj ulaznih podataka, bilo je pitanje koji od tih podataka su nam stvarno potrebni za prognozu. Svaki dodatni ulazni niz usporava treniranje te unosi nove zavisnosti u model, koje mu potencijalno otežavaju treniranje. Također, mnoge serije su vrlo slične čime možda imamo dosta redundantnih ulaznih podataka.



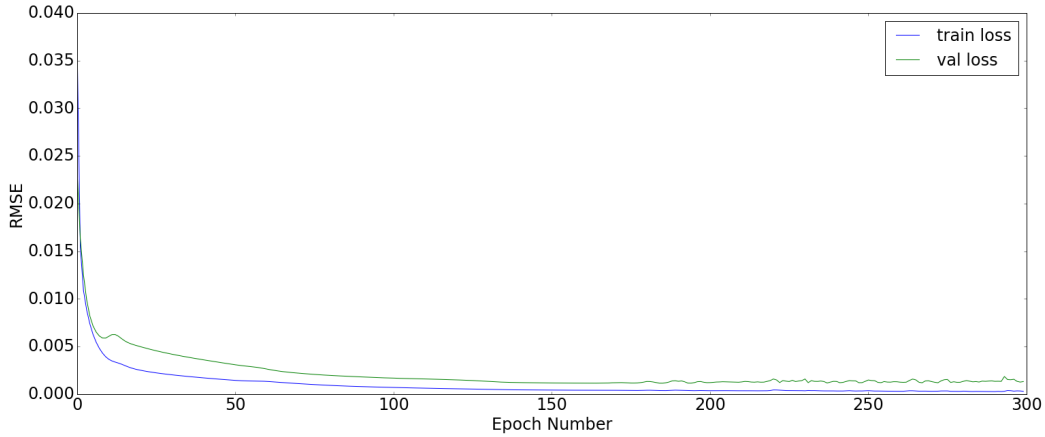
Slika 5.7 Greške modela s različitim ulaznim nizovima

Ovaj graf prikazuje greške na modelu prognoze vodostaja Jamnička Kiselica s različitim izborima ulaznih sljedova. Model 1 dobivao je na ulaz samo podatke vodostaja i protjecaja u postaji Jamnička kiselica, model 2 je na ulaz dodao i sve podatke o hidrološkim stanicama uzvodno. Model 3 je na ulaz još dodao i podatke s direktnih uzvodnih meteoroloških stanica (za Jamničku Kiselicu su to Plaški, Slunj i Pisarovina), a model 4 je koristio sve uzvodne podatke. Ovaj eksperiment nam je pokazao da nam pretpostavka kako sve nizvodne serije ovise o prethodnim ne dovodi nužno do boljih rješenja, a uvodi i dodatnu varijancu u moguća rješenja. Bolja rješenja u modelu 2 u odnosu na model 3 nas navode na ideju da je možda potrebno povećati kapacitet modela. Također se pokazalo da model ne pokazuje velika poboljšanja s uvođenjem ulaznih sljedova protjecaja, ako se prognozira vodostaj, i obrnuto. Tako da se uglavnom prognoziralo samo na temelju istovrsnih uzvodnih podataka te direktnih meteoroloških serija.

5.2.2.3. Odabir ostalih hiperparametara

Na slici 5.8 je prikazan odnos između greške na skupu za učenje i skupu za validaciju tijekom epoha učenja mreže s 30 neurona u prvom sloju. Vidi se da već za oko 100 epoha treniranje dođe do zasićenja, te daljnje treniranje ne dovodi do značajnog poboljšanja. Ovo nas navodi na mišljenje da model

nema dovoljan kapacitet. Iznos greške na prikazanom grafu je mali jer se prije dovođenja podataka u neuronsku mrežu radilo skaliranje vrijednosti ulaza.



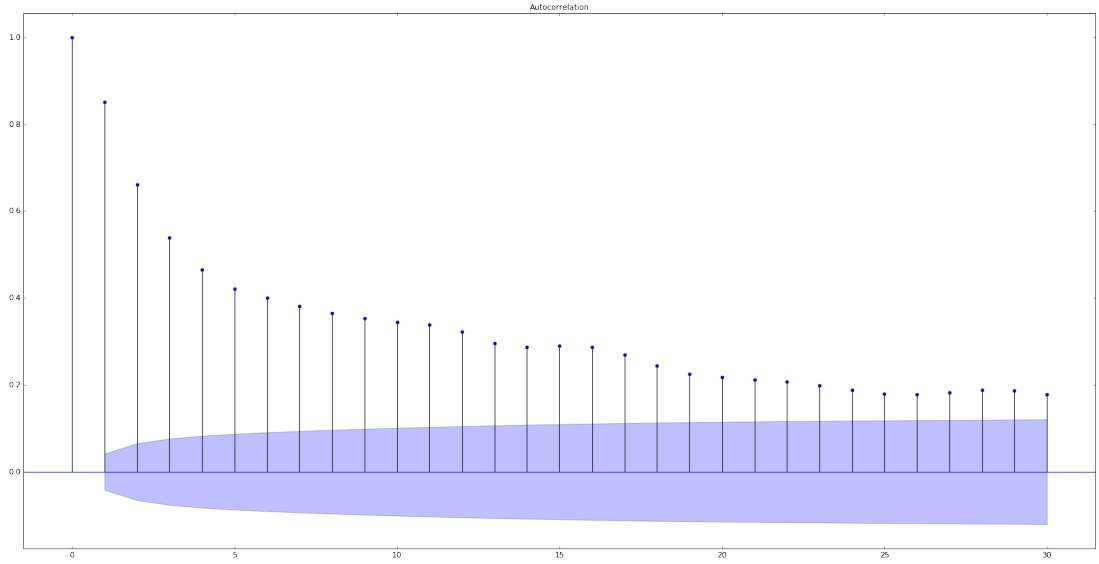
Slika 5.8 Greška na skupu za učenje i skupu za validaciju kroz epohe

Kako bi povećali kapacitet isprobane su različite arhitekture neuronske mreže. No povećanje broja neurona, niti dodavanje još skrivenih slojeva nije uvelo značajna poboljšanja. Na kraju je ipak korišteno rano zaustavljanje, odnosno zaustavljanje treniranja modela u trenu kada greška na skupu za validaciju počne rasti. Tom metodom se gubi određeni dio podataka koji bi se inače koristio na treniranje, sa svrhom bolje generalizacije modela nad novim podatcima.

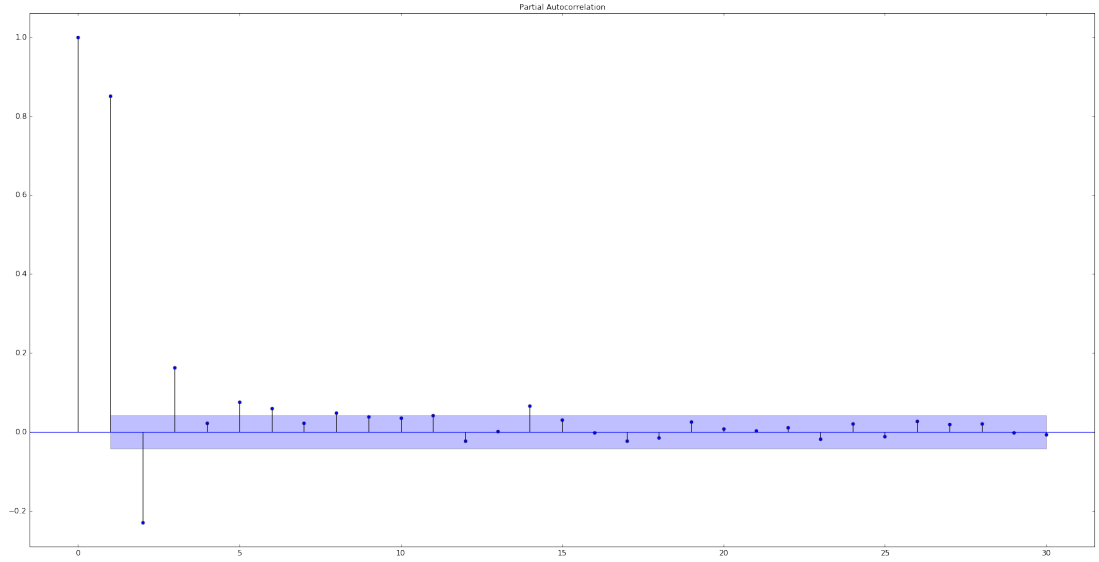
Za treniranje se koristio optimizacijski algoritam Adam s parametrima $\alpha = 0.001, \beta_1 = 0.9\ i\ \beta_2 = 0.999$, te srednjim kvadratnim odstupanjem (MSE) kao funkcijom cilja.

5.2.3. Prognoza autoregresijom vektora

Prvi korak izgradnje modela vektorske autoregresije bio je određivanje parametra p, odnosno količine koraka iz povijesti koji bi se koristili za regresiju vektora. Iz tog razloga iscrtani su ACF i PACF grafovi za svaki vremenski slijed.

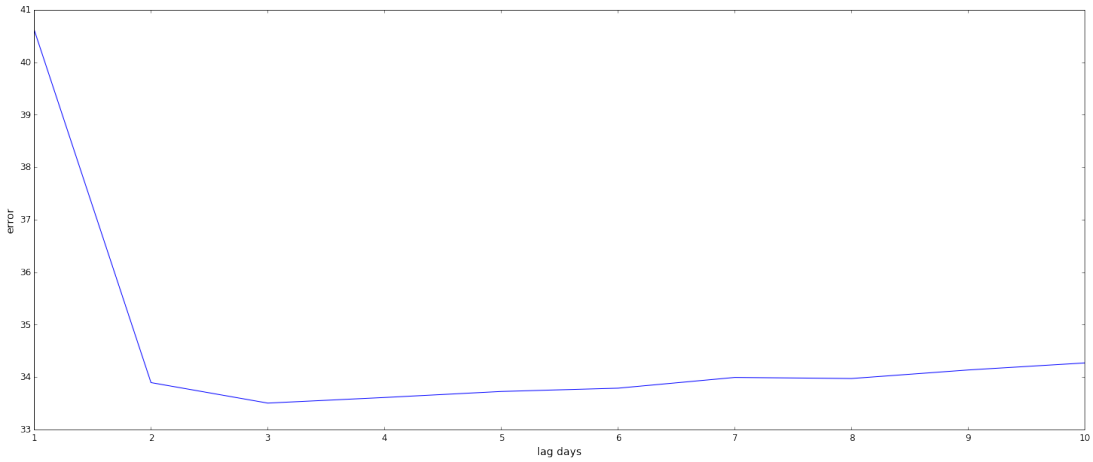


Slika 5.9 Graf ACF za vremensku seriju vodostaja Kamanje



Slika 5.10 Graf PACF za vremensku seriju vodostaja Kamanje

Na slici 5.9 se vidi da serija vodostaja Kamanje ima višu pozitivnu korelaciju s čak preko trideset koraka iz svoje prošlosti, no PACF graf sa slike 5.10 nam sugerira da koraci nakon četvrtog iz povijesti više nisu toliko važni. Udaljavanjem kroz povijest serije, vrijednosti imaju sve manji značaj. Ovi grafovi nam daje neke grube pretpostavke za odabir parametra, odnosno govore nam da parametar p tražimo u blizini trećeg koraka.



Slika 5.11 Greška VAR modela u ovisnosti o redu modela

Pošto se VAR model vrlo brzo trenira, te je treniranje deterministično, isprobane su vrijednosti modela između 1 i 10, kako bi vidjeli koji odabir parametra dovodi do boljih rješenja. Testovi su nam pokazali da model ima najmanju grešku s gledanjem 3 dana unatrag, te smo upravo taj parametar koristili pri prognozi. Za odabir ulaznih serija se koristila intuicija koja je objašnjena prije, odnosno odabirane su sve uzvodne hidrološke stanice te direktne uzvodne meteorološke stanice.

U svakom koraku prognoze, model je ponovno treniran tako da mu je kao ulaz predana cijela povijest vremenske serije, uključivši i taj trenutak. Tako je model u svakom koraku izgradio korelacijsku matricu koja pokazuje korelaciju između svih serija u skupu za učenje.

Tablica 5.3 Korelacija između vremenskih tokova vodostaja

	Farkašić	Jamnička Kiselica	Karlovac	Kamanje
Farkašić	1.0	0.79	0.74	0.57
Jamnička K.	0.79	1.0	0.93	0.71
Karlovac	0.74	0.93	1.0	0.87
Kamanje	0.57	0.71	0.87	1.0

U tablici 5.3. je prikazana korelacija između serija vodostaja. Vidi se da stanice koje su bliže imaju veću pozitivnu korelaciju.

5.2.4. Rezultati

U tablici 5.4 su prikazani dobiveni rezultati svih izgrađenih modela za prognozu jednog koraka.

Tablica 5.4 Rezultati prognoze 1 koraka za serije vodostaja

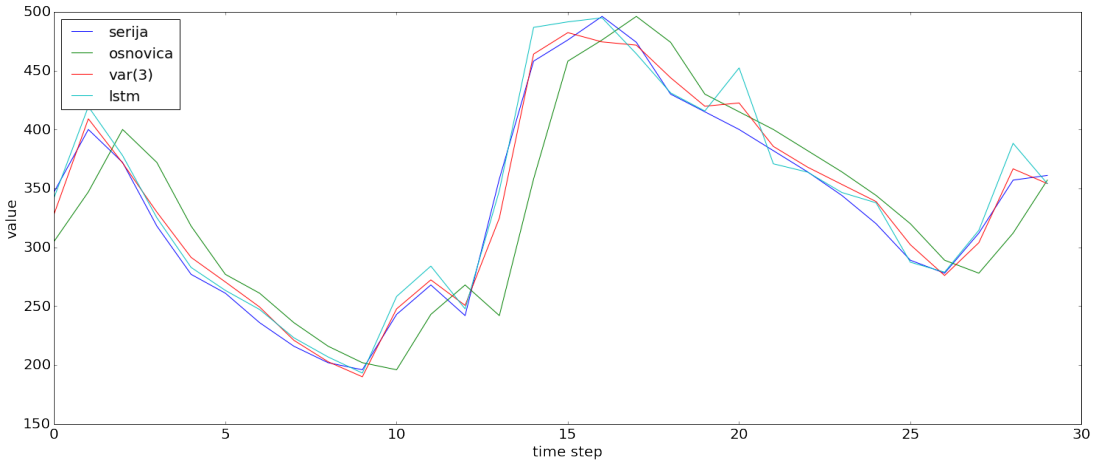
Vremenski slijed	Osnovica		LSTM		AR(3)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Kamanje	27.7	46.9	24.0	39.6	34.4	63.4
Karlovac	49.3	77.3	27.9	49.9	37.6	57.3
Jamnička Kiselica	31.9	53.2	22.3	32.7	22.6	36.9
Farkašić	50.3	76.2	16.9	26.8	20.2	33.7

Na temelju dobivenih rezultata, možemo zaključiti da su prognoze bolje za stanice niže kroz vodostaj. Taj rezultat ima smisla jer je svaka sljedeća stanica imala više ulaznih podataka za učenje. Izuzev prve stanice Kamanje, modeli su uvijek imali bolje rezultate od osnovne prognoze, što nam također govori da prognoza jako ovisi o kvaliteti i količini ulaznih podataka. Isto tako, iako su zahtijevali više posla oko obrade podataka i odabira *hiperparametara*, LSTM modeli su uvijek dovodili do boljih rješenja.

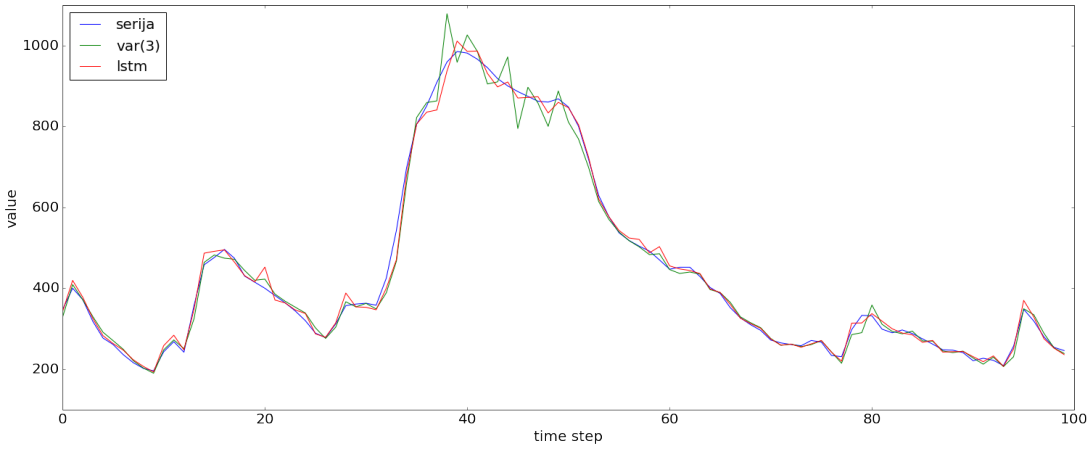
Tablica 5.5 Rezultati prognoze 1 koraka za serije protjecaja

Vremenski slijed	**Osnovica**		**LSTM**		**AR(3)**	
	MAE	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
Kamanje	37.8	70.5	23.8	41.5	26.6	42.0
Jamnička Kiselica	45.1	71.6	30.7	42.9	31.0	50.2
Farkašić	41.7	62.5	24.2	37.3	25.9	38.6

Na osnovu rezultata prognoze protjecaja iz tablice 5.5, možemo također zaključiti da su nizvodne stanice bila lakše za prognozirati. Rezultati bi vjerojatno bili i bolji da smo kao ulazni niz imali i podatke o protjecaju sa stanice Karlovac.



Slika 5.12 Prognoza prvih 30 dana svim modelima



Slika 5.13 Prognoza prvih 100 dana modelima VAR(3) i LSTM

Slike 5.12 i 5.13 prikazuju originalni niz vodostaja s postaje Farkašić te prognozirane vrijednosti za jedan korak unaprijed korištenjem LSTM modela i VAR(3) modela. Oba modela se dobro nose s regresijom vrijednosti, odnosno prate rast i pad varijable, no VAR model pokazuje veća odstupanja pri vrhovima vrijednosti serije što je na kraju dovelo do veće ukupne greške.

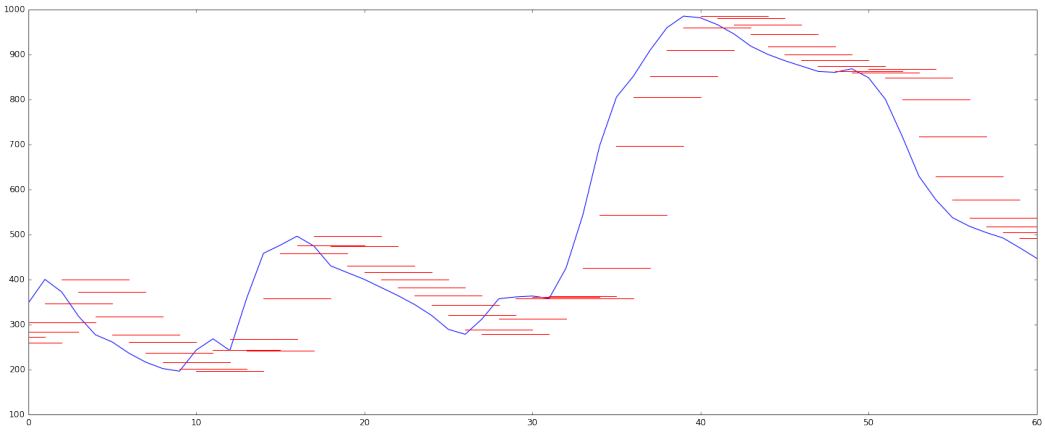
5.3. Prognoza više koraka unaprijed

Prognoza više koraka unaprijed je teži i zanimljiviji problem. Klasična pojava kod prognoze unaprijed jest gomilanje greške sa svakim daljnjim korakom u budućnosti. Za potrebe našeg zadatka, fokusirali smo se na prognozu 5 dana unaprijed. Ideja je bila pravovremeno dobiti informaciju o brzom povećaju vodostaja ili protjecaja rijeke.

Prognoziranje multivarijatnih vremenskih sljedova je još teži problem, pogotovo ako se koristi rekurzivna strategija prognoziranja. Naime, kako se ulaz u model sastoji od više vremenskih sljedova, bilo bi potrebno prognozirati svaki od njih, te izlaz tog modela koristiti kao ulaz za model u sljedećem koraku. Ipak, isproban je VAR model koji radi upravo to, a također je izgrađen i LSTM model u kojem je korištena direktna strategija prognoze. Prognoza više koraka unaprijed je izrađena samo za seriju vodostaja u postaji Farkašić, jer su se na toj stanici pokazala najbolja rješenja.

5.3.1. Osnovica prognoze

Kao i za prognozu jednog koraka unaprijed, izrađena je osnovica prognoze više koraka. Korišten je model ustrajnosti koji, slično kao i kod prognoze za jedan dan unaprijed, za sve sljedeće dane prognozira zadržanu trenutnu vrijednost serije. Ovo je naravno, naivan postupak, ali nam zadaje referentnu gornju granicu greške za naš izgrađeni model.



Slika 5.14 Osnovica prognoze za 5 koraka unaprijed

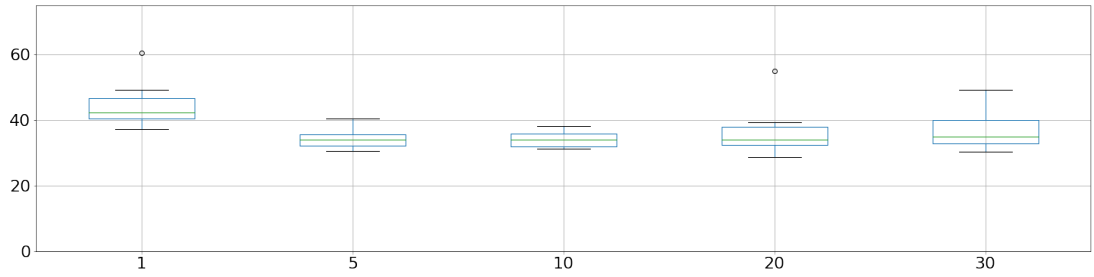
Tablica 5.6 Greške po koracima unaprijed koje je ostvarila osnovna prognoza

	1	2	3	4	5
RMSE	62.9	111.6	145.8	169.6	186.3
MAE	42.0	76.2	101.7	119.4	132.4

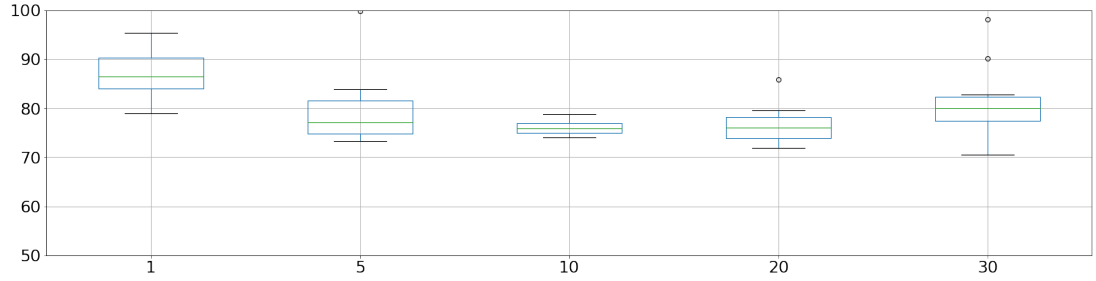
Tablica 5.6 pokazuje pojedine greške koje čini osnovni model, po danima prognoze. Vidi se da greška svakim daljnjim korakom raste.

5.3.2. Prognoza neuronskom mrežom

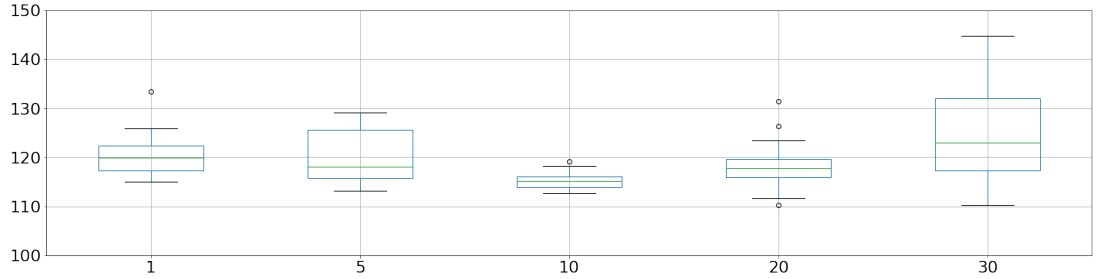
Za prognoziranje 5 dana unaprijed korištena je direktna strategija prognoze. To smo postigli tako da samo u izlazni sloj neuronske mreže stavili potpuno povezani sloj s 5 neurona. Tako smo stvorili jedan model, ali smo opet svaki sljedeći izlaz modelirali neovisno o prethodnom izlazu. Ovakav model je nešto teži za naučiti, pa smo povećavali kapacitet mreže tako da smo dodavali još neurona i slojeva. Efektivnijim se ipak pokazalo smanjiti broj ulaznih koraka u mrežu te tako smanjiti broj težina koje neuronska mreža mora naučiti.



Slika 5.15 Greške prognoze prvog dana s različitim veličinama ulaznog prozora

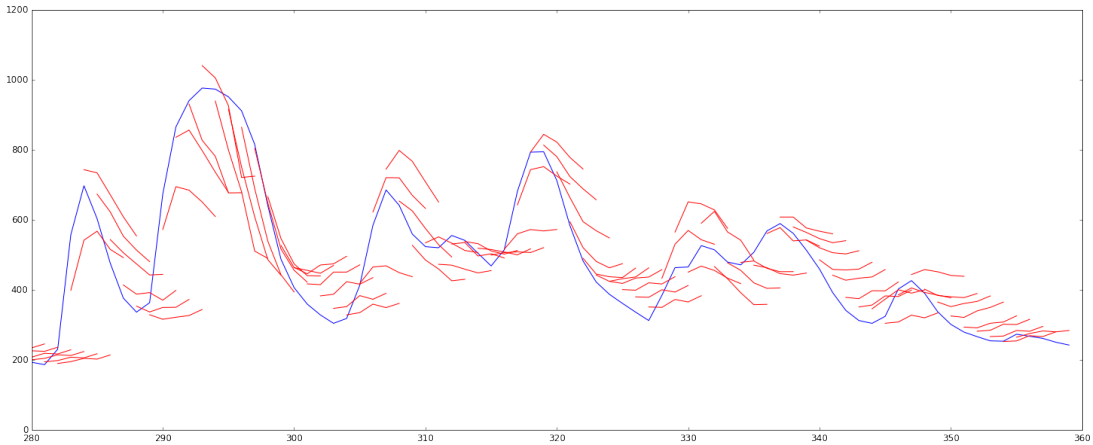


Slika 5.16 Greške prognoze drugog dana s različitim veličinama ulaznog prozora



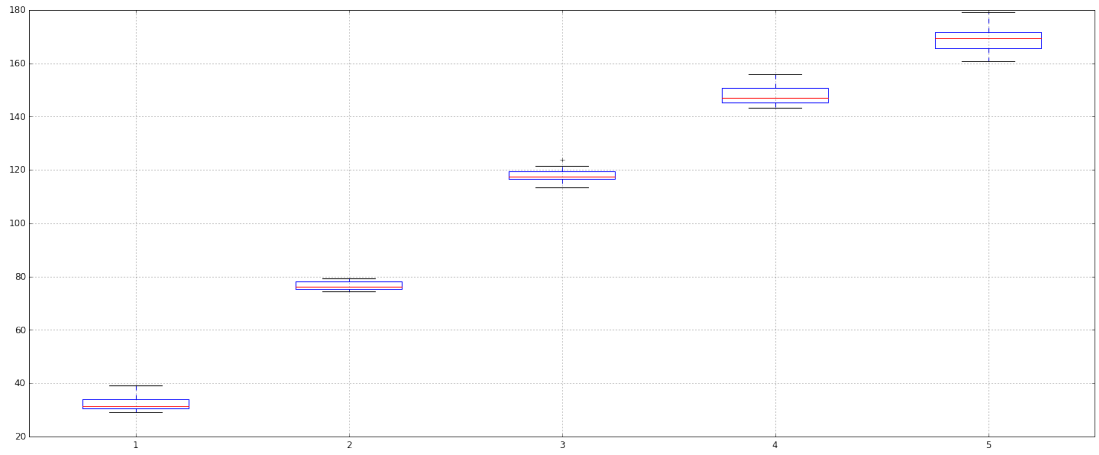
Slika 5.17 Greške prognoze trećeg dana s različitim veličinama ulaznog prozora

Na slikama 5.15, 5.16 i 5.17, prikazane su greške modela za prva tri dana prognoze s različitim veličinama ulaznog prozora. Vidi se da model ne radi najbolje s 30 ulaznih dana, kao što je bio slučaj kod prognoze jednog koraka. Uzrok ove pojave jest činjenica da veliki prozori uvode veliku varijancu u prostor rješenja, a to više dolazi do izražaja kada imamo i veću dimenziju izlaza. Zbog najmanje greške, kao i najmanje varijance u iscrtanim rješenjima, u daljnoj prognozi smo koristili veličinu prozora 10.



Slika 5.18 Zadnjih 80 dana prognoze 5 dana unaprijed nad vremenskim slijedom vodostaj Farkašić

Na slici 5.18 prikazana je prognoza 5 dana unaprijed koja je izvršena LSTM modelom s 10 neurona te s ulaznim prozorom od 10 dana. Ostali hiperparametri modela su isti kao i oni korišteni kod prognoze za 1 dan. Vidi se da model često ne uspijeva pratiti porast vrijednosti serije što nas ne čudi jer nije modelirana veza između slijednih koraka prognoze.



Slika 5.19 Greške po danima prognoze nad vremenskom serijom vodostaja Farkašić

Na slici 5.19 prikazane su greške po danima prognoze. Također je prisutno gomilanje greške s odmicanjem koraka, ali su rezultati bolji od prognoze osnovnim modelom.

Tablica 5.7 Greške po koracima unaprijed koje je ostvario LSTM model

	1	**2**	**3**	**4**	**5**
RMSE	30.7	77.8	115.2	147.0	174.8
MAE	19.7	55.9	84.4	108.2	127.9

U tablici 5.7. prikazani su rezultati najboljeg izgrađenog modela prognoze vodostaja Farkašić.

5.3.3. Prognoza autoregresijom vektora

Za prognozu modelom VAR radi se isti postupak kao i kod prognoze jednog koraka, samo se pri uzimanju izlaza navodi broj koraka koji se želi prognozirati. Broj zaostalih varijabli koji je korišten za prognozu je također bio 3. Interno model VAR radi rekurzivnu strategiju prognoze. Pošto model u učenju već koristi matricu koja izračunava korelaciju između svih pojedinih vremenskih sljedova, istu koristi za prognoziranje svake pojedine serije u trenutku *t*. Nakon toga, vrijednost svih serija u trenutku *t* koristi za prognoziranje vrijednosti u trenutku *t+1.* Isti postupak se radi i za sve ostale korake.

Tablica 5.8 Greške po koracima unaprijed koje je ostvario VAR(3) model

	1	**2**	**3**	**4**	**5**
RMSE	34.2	80.2	117.8	145.1	163.7
MAE	20.4	54.2	82.3	103.5	118.1

U tablici 5.8. prikazani su iznosi grešaka modela VAR(3) po danima. Vidi se da su iznosi grešaka niži nego oni ostvareni s osnovnom prognozom. Ovdje je također zanimljivo primijetiti kako su za rane dane rezultati gori nego kod prognoze korištenjem neuronske mreže, ali prolazom u budućnost postaju bolji. To nas navodi na mišljenje da je VAR model bolji u autoregresivnim problemima, te da nelinearnost koju smo uveli neuronskom mrežom ne donosi uvijek bolja rješenja. Također, u ovom problemu je očito da postoji veza između slijednih koraka serije koju s LSTM-om uopće nismo modelirali, a VAR model je barem u maloj mjeri uspijeva pogoditi što se odražava i na rezultatima pri prolazu u budućnost.

6. Zaključak

Kroz ovaj rad, prikazane su metode multivarijatne prognoze senzorskih vremenskih serija. Konkretno, opisana je prognoza temeljena na povratnim neuronskim mrežama te na modelu vektorske autoregresije. Nakon toga je analiziran i prognoziran vodostaj i protjecaj rijeke Kupe na osnovu povijesnih podataka. Iz dobivenih rezultata možemo zaključiti da su se u našem primjeru neuronske mreže pokazale kao bolje rješenje problema. Ipak, model vektorske autoregresije je davao dobra rješenja te se pokazuje kao dobra alternativa zbog jednostavnosti modeliranja kao i zato što ne zahtijeva mnogo ulaznih podataka.

Rezultati za prognozu nekoliko koraka unaprijed nam govore kako je prognoza unaprijed i dalje težak problem za čije rješavanje je potrebno iscrpno pretraživanje parametara i strategija prognoze, kao i mnogo računalne snage.

Kao dodatno istraživanje bilo bi zanimljivo usporediti rješenja s modelima poput običnih unaprijednih mreža te konvolucijskih mreža, koje se također koriste u prognozi vremenskih sljedova. Također, bilo bi interesantno vidjeti kako se ponašaju izgrađeni modeli kod prognoze istih serija na dnevnoj bazi, odnosno s vrijednostima kroz sate u danu.

Literatura

[1] Robert H. Shumway, David S. Stoffer. Time Series Analysis and Its Applications: with R examples. 2011.

[2] Rob J. Hyndman: Forecasting: principles and practice. 2013. URL https://www.otexts.org/fpp

[3] Jason Brownlee, Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future. 2017.

[4] Gianluca Bontempi, Souhaib Ben Taieb, Yann-Aël Le Borgne. Machine Learning Strategies for Time Series Forecasting. 2013.

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. 2016.

[6] Denny Britz. Backpropagation Through Time and Vanishing Gradients. 2015. URL http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/

[6] Aurélien Géron. Hands-On Machine Learning with Scikit-Learn and TensorFlow. 2017

[7] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014. URL https://arxiv.org/abs/1412.6980

[8] Christopher Olah. Understanding LSTM Networks. 2015. URL http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[9] Javno dostupan podatkovni skup International Airline Passengers: monthly total in thousands. Jan 49 - Dec 60. URL https://datamarket.com/data/set/22u3/international-airline-passengers-monthly-totals-in-thousands-jan-49-dec-60#!ds=22u3

Sažetak

Prognoza vremenskih serija u senzorskim tokovima podataka

Kroz ovaj rad napravljen je pregled metoda analize i prognoze vremenskih serija. Objašnjene su metode prognoziranja vremenskih serija strojnim učenjem te strategije prognoze nekoliko koraka unaprijed. Detaljno su objašnjene metode prognoziranja neuronskim mrežama te statističkim autoregresivnim modelima. Na studijskom slučaju stvarnih senzorskih tokova vodostaja, protjecaja i padalina, implementirane su i evaluirane tehnike multivarijatne prognoze povratnim neuronskim mrežama te modelom autoregresije vektora. Izvršena je analiza nad senzorskim tokovima, a svi dobiveni rezultati su prikazani grafički.

Ključne riječi: vremenske serije, senzorski tokovi, povratne neuronske mreže, autoregresivni model, vektorska autoregresija, analiza vremenskih serija, prognoza vremenskih serija

Summary

Time Series Forecasting in Sensor Data Streams

In this thesis the methods of time series analysis and time series forecast are analyzed. Machine learning methods for forecasting and multi step strategies are described. Neural networks and statistical autoregressive models for forecasting are also described in detail. Recurrent neural network models and vector autoregression models are implemented and evaluated on the case study of actual sensory streams of water levels, sprains and rainfall. Analysis and forecast results are presented and visualized.

Key words: time series, sensory streams, recurrent neural networks, autoregressive model, vector autoregression, time series analysis, time series forecast

Privitak A

Upute za pokretanje izgrađenih *Jupyter* bilježnica

U sklopu ovog rada napravljeno je nekoliko pomoćnih programa u jeziku *Python*, te nekoliko *Jupyter* bilježnica u kojima se može pokrenuti treniranje modela te iscrtati rješenja.

Prije pokretanja *Jupyter* bilježnica, potrebno je instalirati sljedeće alate:

- **Programski jezik Python (3.6)**
- **Razvojno okruženje Anaconda (5.2)**
- **Python knjižnice za duboko učenje Tensorflow (1.1.0) i Keras (2.1.5)**
- **Python knjižnica za statističke strukture podataka Pandas (0.22.0)**
- **Python knjižnica sa statističkim modelima Statsmodels (0.9.0)**

Prikaz rješenja te izgrađenih modela je moguć putem sljedećih *Jupyter* bilježnica:

- **Baseline-one-step.ipynb**
  Osnovica prognoze za jedan korak unaprijed.
- **Baseline-multi-step.ipynb**
  Osnovica prognoze za više koraka unaprijed.
- **Series-analysis.ipynb**
  Vizualizacija i statistička analiza serija.
- **VAR-models.ipynb**
  VAR modeli za prognozu jednog i više koraka unaprijed.
- **LSTM-one-step.ipynb**
  LSTM model za prognozu jednog koraka unaprijed.
- **LSTM-multi-step.ipynb**
  LSTM model za prognozu više koraka unaprijed.

Navedene bilježnice koriste izgrađene programske datoteke *utils.py*, *lstm.py* te *baseline.py*, za brzu izgradnju modela te vizualizaciju podatka. Putem navedenih bilježnica moguće je graditi modele za bilo koju od dostupnih postaja. Ulazni podatci s hidroloških postaja nalaze se u mapi "*vod-pro*". Pri tome podatci o vodostajima imaju sufiks 'H', a podatci o protjecajima sufiks 'Q'. Podatci o padalinama se nalaze u mapi '*padaline*'.