

Prediktivna dijagnostika industrijske opreme

Cvjetović, Vladimir

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Chemical Engineering and Technology / Sveučilište u Zagrebu, Fakultet kemijskog inženjerstva i tehnologije**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:149:974912>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-14**



Repository / Repozitorij:

[Repository of Faculty of Chemical Engineering and Technology University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI DIPLOMSKI STUDIJ

Vladimir Cvjetojević

DIPLOMSKI RAD

:

Zagreb, rujan 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
POVJERENSTVO ZA DIPLOMSKE ISPITE

Kandidat **Vladimir Cvjetojević**

Predao je izrađen diplomski rad dana: 17. rujna 2021.

Povjerenstvo u sastavu:

Doc. dr. sc. Željka Ujević Andrijić, Fakultet kemijskog
inženjerstva i tehnologije, Sveučilište u Zagrebu

Prof. dr. sc. Nenad Bolf, Fakultet kemijskog inženjerstva i
tehnologije, Sveučilište u Zagrebu

Izv. prof. dr. sc. Igor Dejanović, Fakultet kemijskog
inženjerstva i tehnologije, Sveučilište u Zagrebu

Dr. sc. Matija Cvetnić, poslijedoktorand, Fakultet kemijskog
inženjerstva i tehnologije, Sveučilište u Zagrebu (zamjena)

povoljno je ocijenilo diplomski rad i odobrilo obranu diplomskog
rada pred povjerenstvom u istom sastavu.

Diplomski ispit održat će se dana: 22. rujna 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET KEMIJSKOG INŽENJERSTVA I TEHNOLOGIJE
SVEUČILIŠNI DIPLOMSKI STUDIJ

Vladimir Cvjetojević

PREDIKTIVNA DIJAGNOSTIKA INDUSTRIJSKE OPREME
PREDICTIVE DIAGNOSTICS OF INDUSTRIAL EQUIPMENT

DIPLOMSKI RAD

Voditelj rada:

doc.dr.sc. Željka Ujević Andrijić

Članovi ispitnog povjerenstva:

doc.dr.sc. Željka Ujević Andrijić

prof.dr.sc. Nenad Bolf

izv.prof.dr.sc. Igor Dejanović

Zagreb, rujan 2021.

Zahvaljujem se mentorici doc.dr.sc Željki Ujević Andrijić na predloženoj temi koja mi je omogućila da se upoznam sa načelima rada umjetne inteligencije, kao i na njenom vodstvu, savjetima, sugestijama i uloženom vremenu koje su mi pomogle da studij privedem kraju.

Posebno bih se želio zahvaliti asistentu mag.ing.oecing. Nikoli Rimcu za strpljenje, uloženo vrijeme kao i znanje programiranja bez kojega rad ne bi bio moguć u ovome obliku.

Najveće zahvale i zasluge idu mojim roditeljima Ranku i Vlasi radi podrške, oslonca i svega što su uložili u mene tijekom zadnjih 18 godina formalnog obrazovanja, a čega nisam svjestan u ovim godinama.

Sadržaj

| | |
|--|----|
| Sažetak..... | 1 |
| Abstract | 2 |
| 1.Uvod | 3 |
| 2.Teorijski dio | 4 |
| 2.1. Preventivno i prediktivno održavanje..... | 4 |
| 2.2. Izmjenjivači topline | 6 |
| 2.2.1. Bilance topline | 7 |
| 2.2.2. Koeficijent prijenosa topline | 8 |
| 2.3. Naslage | 9 |
| 2.3.1. Metode smanjenja nastanka naslaga | 10 |
| 2.3.2. Metode praćanja nastajanja naslaga..... | 10 |
| 2.4. Razvoj modela..... | 11 |
| 2.5. Neuronske mreže..... | 12 |
| 2.5.1. Učenje neuronske mreže..... | 16 |
| 2.5.2. Slijedni podaci..... | 19 |
| 2.5.3. Dinamičke neuronske mreže | 19 |
| 2.5.4. Rekurentne (povratne) neuronske mreže | 19 |
| 2.5.5. Učenje rekurentnih neuronskih mreža..... | 24 |
| 2.5.6. Čelija s dugoročnom memorijom – LSTM | 25 |
| 2.6. Korišteni softver za razvoj modela | 30 |
| 2.6.1. Python | 30 |
| 3. Eksperimentalni dio..... | 32 |
| 4. Metodika | 35 |
| 4.1. Konfiguracije modela neuronske mreže u Python-u | 35 |

| | |
|-------------------------------|----|
| 4.2. Optimizacija mreže | 35 |
| 5. Rezultati i rasprava | 37 |
| 5. Zaključak | 49 |
| 6. Popis oznaka | 50 |
| 7. Popis literature: | 52 |
| Životopis | 53 |

Sažetak

Djelotvornost izmjenjivača topline opada s vremenom te ga je nakon nekog vremena potrebno servisirati, tj. mehanički ili kemijski očistiti od nakupljenih naslaga. Djelotvornost počinje padati nakon nekog vremena zbog nastajanja naslaga koje djeluju kao otpor prijenosu topline (tzv. *fouling*). Kako bi se mogao organizirati rad postrojenja i odrediti termin servisiranja nužno je poznavati trenutno stanje promatranog izmjenjivača topline, odnosno potrebno je kontinuirano pratiti nastajanje naslaga na stijenkama. Mehanizam nastajanja naslaga je uglavnom nepoznat i nedovoljno istražen stoga se pribjegava empirijskim modelima za procjenu nastajanja naslaga. Jedan od modela koji se pokazao primjenjiv, ali još uvijek nedovoljno istražen za praćenje nastajanja naslaga je model neuronskih mreža iz područja umjetne inteligencije. U ovome radu primijenjen je model neuronskih mreža za procjenu stanja industrijskog izmjenjivača topline iz postrojenja u naftno petrokemijskoj industriji. Kako su procesni uvjeti u industrijskim procesima izrazito nestacionarni i nelinearni, ispitivana je primjena rekurentnih (povratnih) neuronskih mreža. Statistički kriteriji vrednovanja kvalitete modela kao i dobiveni grafički rezultati ukazuju na to da se rekurentne neuronske mreže mogu koristiti za kontinuirano praćenje nastajanja naslaga u izmjenjivačima topline.

Ključne riječi:

Izmjenjivač topline, nastajanje naslaga, otpor prijenosu topline, neuronska mreža, nestacionarni uvjeti, rekurentna neuronska mreža

Abstract

Heat exchanger's efficiency diminishes over time, hence there is a need for its maintenance in order to remove the deposits using mechanical and/or chemical procedures. Efficiency starts to decline due to deposit formation on the surface of the exchanger which acts like heat transfer resistance. In order to schedule the working time of such plant, there is a need for heat exchanger's diagnostics, in other words there is a need for continuous monitoring of deposit formation. The mechanism of deposit formation is yet quite unknown, therefore, use of empirical models is unavoidable. In this Master's thesis, neural network model for monitoring the state of the heat exchanger in petrochemical industry is used. Due to non-stationary and nonlinear conditions in industrial processes, use of the recurrent neural network has been tested. Statistical criterion of the model's quality, as well as given graphical results show that recurrent neural network can be used for continuous monitoring of deposit formation in heat exchangers.

Key words:

Heat exchanger, fouling, deposit formation, heat transfer resistance, neural network, non-stationary conditions, recurrent neural network

1.Uvod

Napretkom tehnologije industrijski kompleksi postaju sve veći i složeniji. Kako raste kompleksnost takve infrastrukture tako se povećava i vjerojatnost pojave kvara na nekom od uređaja. Kako bi se moglo pravodobno reagirati i otkloniti problem nameće se potreba za praćenjem procesa i naprednom dijagnostikom procesne opreme u industriji. Pogreška u kontekstu praćenja procesa može se interpretirati kao nedozvoljena devijacija jedne ili više karakterističnih veličina koje ne dozvoljavaju postrojenju da izvrši zadatak za koji je predviđeno [1]. Umjetna inteligencija pokazuje veliki potencijal kao alat za procjenu stanja sustava, a posebice umjetne neuronske mreže (engl. *Artificial Neural Networks*) koje nalaze vrlo uspješnu primjenu u modeliranju procesa, kod predikcije svojstava produkta i radnih uvjeta, a pogotovo u vođenju procesa i dijagnostici grešaka [2]. Jedan od najvećih problema u naftnim rafinerijama je nastajanje naslaga u izmjenjivačima topline [3]. Ako se izmjenjivač topline demontira i očisti prije nego mu opada djelotvornost tada dolazi do nepotrebnog zaustavljanja postrojenja i povećanja pogonskih troškova. Ako se pak, on počne čistiti prekasno tada se javljaju veliki gubici energije zbog njegove smanjene efikasnosti. Tradicionalne metode praćenja nastajanja naslaga na izmjenjivačima topline ne mogu se provoditi pouzdano i *online*, te je često nepoznat i mehanizam procesa nastajanja naslaga. Dostupnost velikih količina podataka u postrojenjima u moderno vrijeme čini strojno učenje praktičnom metodom rješavanja problema u praćenju i predviđanju stanja opreme i procesa. Jedna od metoda strojnog učenja su rekurentne neuronske mreže koje su se pokazale iznimno korisne u brojnim područjima kao što su prevođenje jezika, prepoznavanje glasa, skladanje glazbe, autonomna vozila, te nalaze primjenu i u procesnoj industriji. Dobiveni podaci iz Rafinerije nafte Rijeka su podaci u vremenu koji se konstantno mijenjaju zbog dinamičkog vladanja procesa, stoga se nastoji utvrditi u kojoj mjeri prijašnji podaci utječu na sadašnje stanje u procesu. Ako takva zavisnost postoji tada je realno za očekivati da će dinamičke neuronske mreže dati bolje rezultate od statičnih. Jedne od najčešće upotrebljivanih dinamičkih mreža su Hopfieldove, Elmanove, NARX (engl. *nonlinear autoregressive network with exogenous inputs*) i rekurentne neuronske mreže. Rekurentne neuronske mreže imaju karakteristiku zadržavanja konteksta među podacima, a uz pomoć koncepta LSTM (engl. *Long Short Term Memory*) imaju sposobnost dugoročnog pamćenja stoga su predmet istraživanja u ovome radu.

2. Teorijski dio

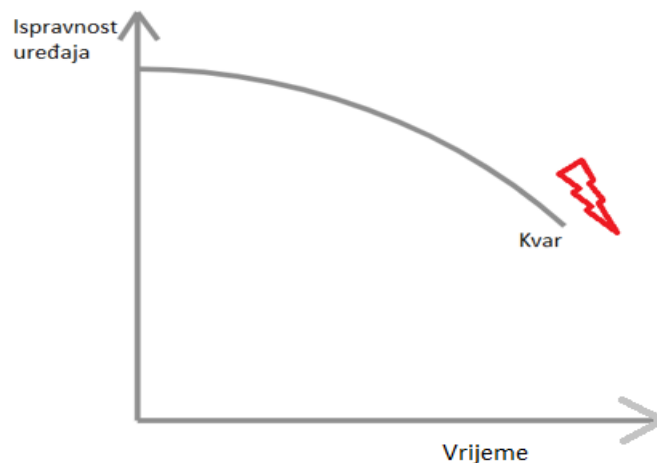
2.1. Preventivno i prediktivno održavanje

Kako bi postrojenje radilo pri optimalnim uvjetima s maksimalnom iskoristivosti, ona postaju sve složenija i potpuno integrirana u kontekstu interneta stvari (engl. IoT – *Internet of Things*). U takvim kompleksnim sustavima pojava greške/kvara na pojedinom dijelu postrojenja može dovesti do zastoja, ozbiljnih problema i potencijalno vrlo opasnih situacija. Kako se svaka oprema neizbježno vremenom pokvari ili joj se smanji djelotvornost, potrebno je razviti sustav dijagnostike opreme. Prvi korak dijagnostike procesa i procesne opreme je detektiranje prave greške/kvara, a ne lažne greške, zatim slijedi izolacija greške kako bi odredili njenu vrstu, mjesto i vrijeme pojave. Posljednji korak čini identifikacija greške.

Riječ „prediktivno“, dolazi od latinske riječi „pre“ što znači prije i „dicare“ što znači proglasiti, dok „održavanje“ podrazumijeva sačuvati nešto u postojećem stanju te zaštititi od kvarova i od propadanja. [4] Cilj održavanja je produžiti rok trajanja uređaja i procesne opreme kako bi postrojenje dugoročno optimalno funkcioniralo.

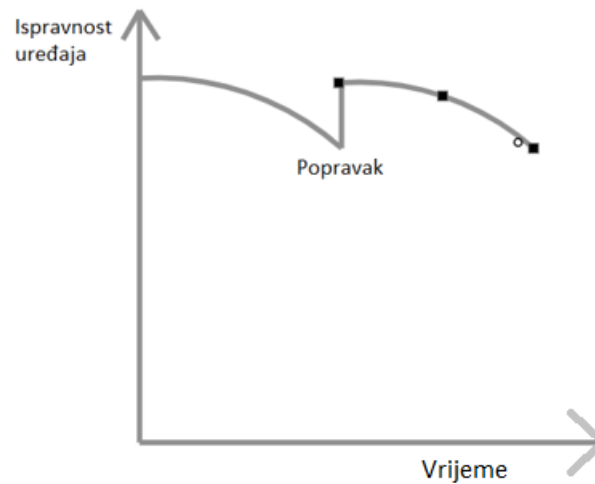
Pod održavanjem se podrazumijeva niz radnji kojima se nastoji proizvodni proces održati u optimalnom stanju što duže vrijeme. Pritom se razlikuju tri kategorije održavanja:

- **Reaktivno održavanje** (eng. *Reactive maintenance*) – uređaj/procesna oprema se koriste do samog kraja, intervenira se tek nakon što je kvar uzrokovao zastoj (slika 1). Takav pristup je koristan kada su dijelovi uređaja pristupačni po niskim cijenama čiji kvar snosi minimalne posljedice, npr. žarulja u kućanstvu.



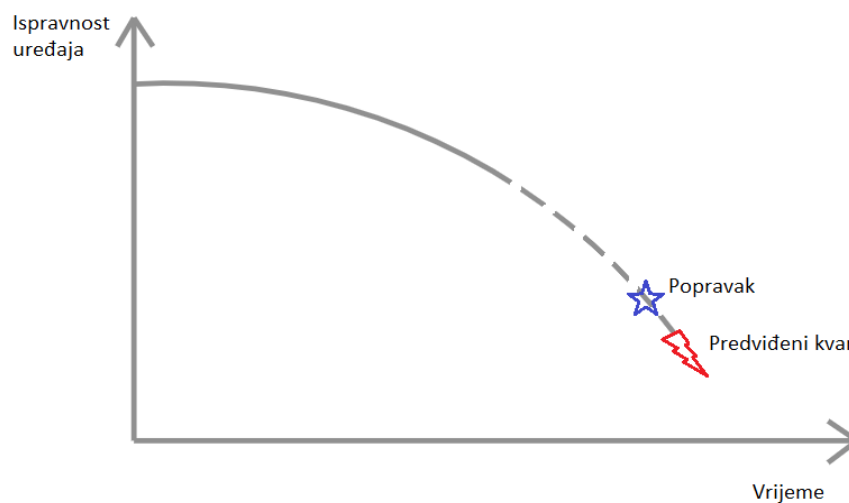
Slika 1. Shema reaktivnog održavanja

- **Preventivno održavanje** (eng. *Preventive maintenance*) – oprema se popravlja prije nego što je procijenjeno vrijeme nastanka kvara. Procjena se vrši na temelju iskustva i redovnim pregledima. Problem kod takvog pristupa je što se ne može pouzdano odrediti vrijeme pojave kvara, stoga se popravak izvršava prerano što uzrokuje dodatne troškove. Ukoliko se čeka predugo kvar može prouzročiti ozbiljne probleme i još veću materijalnu štetu.



Slika 2. Shema preventivnog održavanja

- **Prediktivno održavanje** (eng. *Predictive maintenance*) – ukoliko je moguće predvidjeti vrijeme nastanka kvara, tada se popravak može zakazati nešto prije samog otkazivanja. Time se može u potpunosti iskoristiti sav potencijal koji oprema posjeduje pri čemu se mogu postići znatne uštede.



Slika 3. Prikaz prediktivnog održavanja

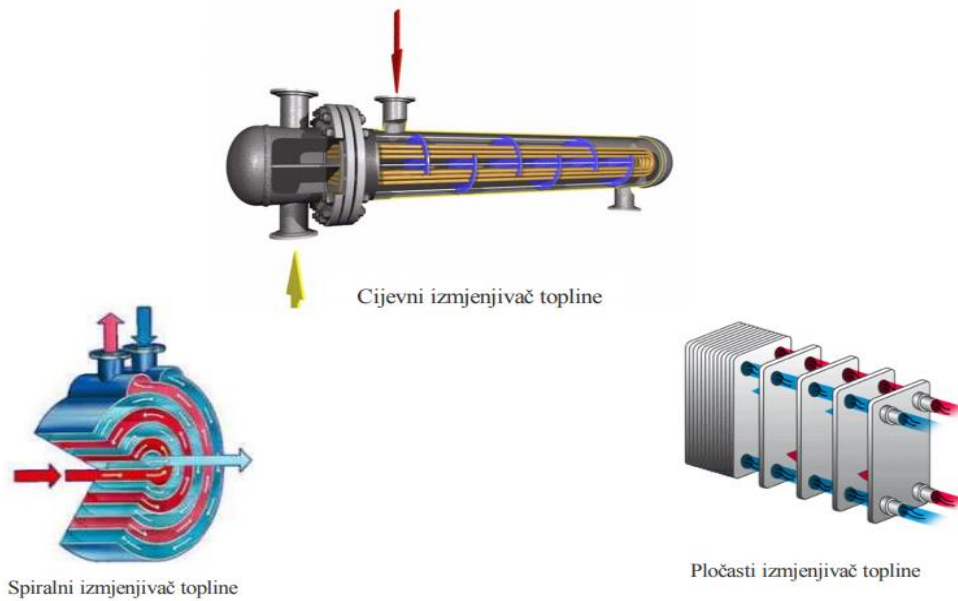
Dakle, ključno je pravovremeno odrediti vrijeme u kojem će doći do kvara ili problema u radu procesne opreme. U tu svrhu se razvija **algoritam prediktivnog održavanja**, koji se sastoji od nekoliko koraka:

- Prikupljanje podataka - Ako podaci nisu dostupni, ili su nepotpuni, oni se mogu generirati pomoću matematičkog modela.
- Predobrada prikupljenih podataka.
- Prepoznati koje se procesne veličine mogu povezati sa stanjem opreme.
- Razvoj modela procesne opreme.
- Primjena razvijenog modela [5]

Kako bi postojao kvalitetan pristup podacima, poželjno je mjerenja u postrojenju provoditi kontinuirano i *on-line* te pritom kontinuirano pratiti stanje procesne opreme. Na taj način u svakom trenutku može postojati valjana indikacija stanja opreme.

2.2. Izmjenjivači topline

Izmjenjivači topline su uređaji u kojima se toplina prenosi sa toplijeg na hladniji fluid putem zajedničke stijenke dobre toplinske vodljivosti. Mogu se klasificirati po raznim kategorijama, a s obzirom na konstrukciju mogu biti cijevni, spiralni i pločasti (slika 4.). S obzirom na smjer strujanja fluida postoje istostrujni, protustrujni i križni izmjenjivači topline. Toplina se može prenositi kondukcijom, konvekcijom i radijacijom. Uglavnom se kroz fluid toplina prenosi konvekcijom, a kroz stijenku kondukcijom, pri čemu je onaj mehanizam koji u sebi sadrži više otpora kontrolirajući. Svakom izmjenjivaču topline sa vremenom opada efikasnost te počne obavljati sve manju toplinsku dužnost, dok u jednom trenutku više nije sposoban obavljati svoju funkciju. Kako fluid teče kroz cijev/i, vremenom dolazi do nastajanja taloga koji se nakuplja na stijenkama izmjenjivača. Talog nastaje iz nekoliko razloga kao što je kristalizacija, odvijanje kemijske reakcije, taloženje čestica, biološki procesi i korozija. Nakon što se izmjenjivač rastavi uočavaju se naslage na stijenkama koje pružaju otpor prijenosu topline. Te naslage, odgovorne za pad efikasnosti izmjenjivača, poznate su još i kao *fouling*.



Slika 4. Vrste izmjenjivača topline

2.2.1. Bilance topline

U idealnom slučaju vrijedi da je toplina koju hladniji fluid primi jednaka toplini koju topliji fluid prenese.

Bilančna jednačina:

$$Q_t = Q_h \quad (1)$$

$$m_h * c_{p,h} * (T_{h,izl} - T_{h,ul}) = m_t * c_{p,t} * (T_{t,izl} - T_{t,ul}) \quad (2)$$

Pri čemu je:

- m - maseni protok [kg/s]
- c_p - specifični toplinski kapacitet [kJ/kg⁻¹K⁻¹]
- Q - izmijenjena toplina [W]
- T - temperatura [K]

Kinetička jednačina:

$$Q = K * A * \Delta T_{lm} \quad (3)$$

Pri čemu je:

- Q – izmijenjena toplina [W]
- K – koeficijent prolaza topline [$W\ m^{-2}\ K^{-1}$]
- A – površina izmjene topline [m^2]
- ΔT_{lm} – pokretačka sila [K]

2.2.2. Koeficijent prijenosa topline

Koeficijent prijenosa topline se definira kao recipročna vrijednost sume svih otpora koji postoje u sustavu, prema jednadžbama (4), (5), (6) i (7).

$$K = \frac{1}{\sum_i R_i} \quad (4)$$

$$K = \frac{1}{\sum_j R_{konv,j} + \sum_k R_{kond,k} + \sum_l R_{f,l}} \quad (5)$$

$$R_{konvekcijski} = \frac{1}{\alpha} \quad (6)$$

$$R_{kondukcijski} = \frac{L}{\lambda} \quad (7)$$

Pri čemu je:

- α – koeficijent prijelaza topline, $W\ m^{-1}\ K^{-1}$
- λ – koeficijent toplinske vodljivosti, $W\ m^{-2}\ K^{-1}$
- R_f , fouling faktor
- $R_{konvekcijski}$ – faktor otpora prijenosu topline konvekcijom
- $R_{kondukcijski}$ - faktor otpora prijenosu topline kondukcijom
- K – koeficijent prijenosa topline

Suma kondukcijskog, konvekcijskog i otpora koji dolazi od naslaga jest ukupni otpor prijenosu topline. Detekcijom povećanog otpora prijenosu topline u izmjenjivaču dobiva se informacija da je vjerojatno došlo do nastajanja naslaga.

2.3. Naslage

Fouling se može definirati kao pad sposobnosti izmjenjivača topline da obavlja toplinsku dužnost za koju je projektiran. Njegova nesposobnost da izvršava svoju zadaću se može kvantitativno prikazati pomoću veličine koja se naziva *fouling* faktor. Određuje se eksperimentalno uspoređujući izmjenjivač topline sa i bez naslaga pomoću jednadžbe (8):

$$R_f = \frac{1}{K_{sa\ naslagama}} - \frac{1}{K_{bez\ naslaga}} \quad (8)$$



Slika 5. Zaprljani izmjenjivač topline [6]

Proces nastanka naslaga se odvija u nekoliko stupnjeva:

- Indukcija (eng. *Induction*) – odvija se na čistoj površini stijenke izmjenjivača te je potrebno neko vrijeme da bi do nje došlo.
- Rast naslaga (eng. *Fouling growth*) – u pravilu je trend rasta linearan ili eksponencijalan, a ovisi o mehanizmu nastanka foulinga, namjeni izmjenjivača i drugim parametrima.
- Granica rasta naslaga (eng. *Fouling growth limit*) – u nekom trenutku naslage dostižu svoju maksimalnu debljinu, dolazi do uspostavljanja ravnoteže između rasta i otkidanja naslaga. Ukoliko je strujanje turbulentno, ravnoteža se postiže nešto ranije.
- Uklanjanje naslaga (eng. *Fouling removal*) – nakon nekog vremena naslage se uklanjaju, obično je to za vrijeme zaustavljanja rada cijelog ili jednog dijela postrojenja radi remonta.

2.3.1. Metode smanjenja nastanka naslaga

Nastajanje naslaga se ne može spriječiti, no može se usporiti njihovo nastajanje i ublažiti njihov učinak sa ciljem produženja vijeka trajanja izmjenjivača topline.

- Dodatak *antifouling* kemikalija – npr. dodatak biocida, otrova koji ubija sve organizme u vodi i tako sprečava nastanak bioloških naslaga.
- Održavanje turbulentnog režima rada – vrlo je poželjno održavati režim strujanja u turbulentnim uvjetima kako bi se spriječio nastanak debelih slojeva na stijenkama, ukoliko to kompenzira dodatni trošak pumpe koja omogućuje strujanje fluida pri visokim brzinama.
- Predimenzioniranje – izmjenjivač topline koji obavlja toplinsku dužnost nešto veću od zadane će i nakon nastanka naslaga neko vrijeme moći efektivno nastaviti sa radom.

2.3.2. Metode praćanja nastajanja naslaga

Klasične metode određivanja nastanka naslaga na stijenkama uključuju:

- Određivanje koeficijenta prijenosa topline
- Mjerenje pada tlaka i protoka
- Mjerenje temperature na ulazu i izlazu toplih i hladnih struja
- Ultrazvučne i električne metode
- Vaganje ploče

Prve tri metode zahtijevaju da se proces vodi pri stalnim procesnim uvjetima što često nije slučaj u realnim postrojenjima. Ultrazvučne i električne metode imaju nedostatak što mogu otkriti samo naslage koje su se lokalno nakupile, a vaganje ploče izmjenjivača zahtijeva prestanak rada postrojenja što često ne dolazi u obzir [6]. Javlja se potreba sakupljanja podataka u realnom vremenu koji se mogu iskoristiti za procjenu stanja izmjenjivača topline.

Prikupljanjem podataka od relevantnih procesnih varijabli netom poslije čišćenja i praćenjem stanja rada kada je izmjenjivač bio djelotvoran može se razviti model izmjenjivača topline koji bi pokazivao rad čistog izmjenjivača. Usporedbom s podacima, koji su dobiveni *online* u realnim uvjetima, može se pratiti razlika među njima, te na taj način izvršiti procjena stanja izmjenjivača. Više o tome bit će objašnjeno u poglavlju Razvoj modela i Metodika.

2.4. Razvoj modela

Kod razvoja modela cilj je pronaći funkcionalnu vezu između ulaznih i izlaznih veličina modela. Modeli mogu biti:

- Teorijski – primjenjivi na široko područje radnih uvjeta, ali mogu biti jako složeni i skupi za razvoj, a vrlo često i nemogući.
- Poluempirijski – lakši su za razvoj od teorijskih modela jer uključuju i podatke dobivene eksperimentalno.
- Empirijski – relativno su jednostavni za razvoj jer koriste podatke iz realnog procesa, međutim problem se javlja pri ekstrapolaciji modela na radne uvjete koji su drukčiji od onih koji su vladali kod podataka korištenih za razvoj.

U ovome radu se razvija model neuronske mreže, za čiji razvoj su neophodni eksperimentalni podaci, a s obzirom da će se pri proračunu fouling faktora koristiti i fundamentalni model, riječ je o poluempirijskom modelu.

Krajnji cilj modela neuronske mreže u ovom radu je procjena fouling faktora izmjenjivača topline. Fouling faktor, jednom kada su poznati koeficijenti prijenosa topline se lako izračuna pomoću jednadžbe (8). Veličine $K_{sa\ naslagama}$ i $K_{bez\ naslaga}$ se računaju pomoću jednadžbi (9) i (10):

$$K_{sa\ naslagama} = \frac{\dot{m} * c_p * (T_{h,i} - T_{h,o})}{A} * F * \frac{(T_{h,i} - T_{c,o}) - (T_{h,i} - T_{c,o})}{\ln \left(\frac{T_{h,i} - T_{c,o}}{T_{h,o} - T_{c,i}} \right)} \quad (9)$$

$$K_{bez\ naslaga} = \frac{\dot{m} * c_p * (T_{h,i} - T_{h,o,model})}{A} * F * \frac{(T_{h,i} - T_{c,o,model}) - (T_{h,i} - T_{c,o,model})}{\ln \left(\frac{T_{h,i} - T_{c,o,model}}{T_{h,o,model} - T_{c,i}} \right)} \quad (10)$$

Pri čemu je:

- $K_{sa\ naslagama}$ – koeficijent prijenosa topline izmjenjivača u kojem postoje naslage
- $K_{bez\ naslaga}$ – koeficijent prijenosa topline čistog izmjenjivača
- \dot{m} - maseni protok fluida
- c_p – specifični toplinski kapacitet fluida
- $T_{h,i}$ – ulazna temperatura hladne struje

- $T_{h,o}$ – izlazna temperatura hladne struje
- $T_{c,i}$ – ulazna temperatura tople struje
- $T_{c,o}$ – izlazna temperatura tople struje
- $T_{h,o,model}$ – izlazna temperatura hladne struje procijenjena modelom neuronske mreže
- $T_{c,o,model}$ – izlazna temperatura tople struje procijenjena modelom neuronske mreže
- F – korekcijski faktor
- A – površina izmjene topline

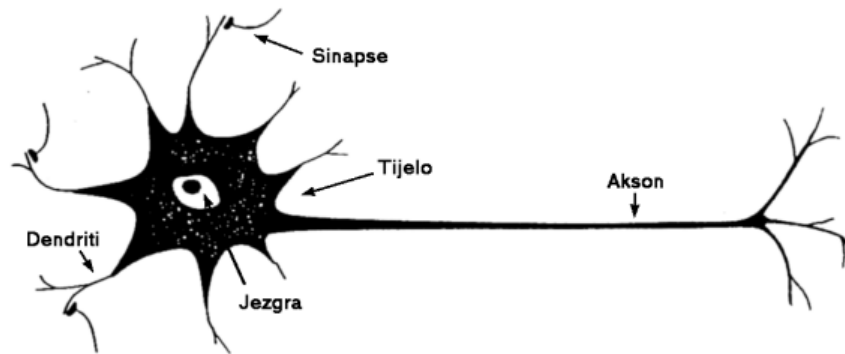
Važno je napomenuti da koeficijent prijenosa topline bez naslaga ($K_{bez\ naslaga}$) uzima u obzir temperature na izlazu iz izmjenjivača koje su dobivene modelom jer se model razvija pomoću podataka iz vremena kada još nije došlo do nastajanja naslaga, za razliku od koeficijenta prijenosa topline sa naslagama ($K_{sa\ naslagama}$) koji u obzir uzima stvarne temperature na izlazu iz izmjenjivača.

2.5. Neuronske mreže

Tijekom stotina milijuna godina evolucije usporedo sa tijelom se razvijao i ljudski mozak te je poprimio karakteristike koje povezujemo sa inteligentnim ponašanjem. Neke od bitnih karakteristika koje se povezuju sa inteligencijom su:

- Paralelno djelovanje
- Distribuirano predočavanje i obrada
- Sposobnost učenja
- Sposobnost uopćavanja
- Prilagodljivost
- Suvislo postupanje s informacijama
- Tolerancija na pogreške i nepotpune informacije (filtriranje podataka) [11]

Neuronska mreža se sastoji od skupa umjetnih neurona koji međusobno komuniciraju i pokazuju neke od gore navedenih karakteristika. Pojam „neuron“ se koristi zato jer se može povući paralela između načina na koji funkcioniraju biološki i umjetni neuron pri čemu svaki dio živčane stanice ima neku svoju matematičku interpretaciju. U ljudskome mozgu neuroni dobiju ulazni signal koji aktivira neuron te počnu međusobno komunicirati i na kraju daju izlazni signal. Ljudska živčana stanica je prikazana pomoću slike 7.



Slika 7. Biološka živčana stanica [12]

Pri čemu svaki dio stanice ima svoju funkciju:

- Sinapse povezuju akson prethodne sa dendritom sljedeće stanice, imaju funkciju oslobađanja neurotransmitera, tvari koja omogućuje prijenos signala od jednog neurona do drugog.
- Dendriti prenose signal u samu stanicu gdje se on obrađuje te ukoliko je dovoljno jak tada se neuron pobuđuje.
- Tijelo stanice u sebi sadrži jezgru koja nosi informacije o nasljednim značajkama organizma.
- Akson prenosi obrađeni signal do drugih neurona.

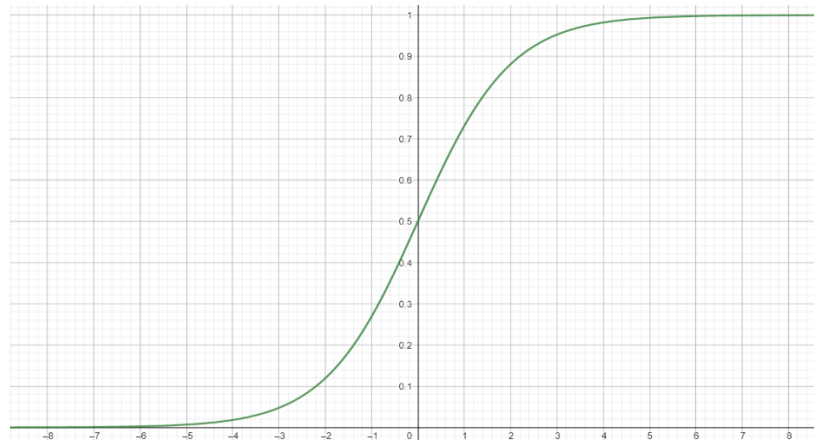
1943. godine američki neurofiziolog i kibernetičar sa Sveučilišta *Illionis* u *Chicagu* Warren McCulloch, te logičar i kognitivni psiholog Walter Pitts su objavili rad pod nazivom „Logički račun neizbježnih procesa u neuronskoj aktivnosti“ gdje uvode pojam „McCulloch-Pitts neuron“ što je prvi matematički model neuronske mreže [13]. Prema tom modelu neuroni su podijeljeni u tri sloja: ulazni, izlazni i skriveni sloj. Ulazni sloj poprima vrijednosti ulaznih veličina [12], dakle broj neurona u ulaznom sloju je jednak broju ulaznih varijabli. Pri tome se svaki ulaz množi sa koeficijentom koji se naziva težinski koeficijent i ima raspon vrijednosti uglavnom od 0 do 1 ili veći od 1. Takvi otežani signali se sumiraju te im se još pridodaje vrijednost praga umjetnog neurona b (11).

$$I = \sum_{i=1}^n x_i * w_i + b_i \quad (11)$$

Ta sumirana vrijednost postaje domena nelinearne aktivacijske funkcije (12), koja daje izlazni signal neurona.

$$Y = f(I) \quad (12)$$

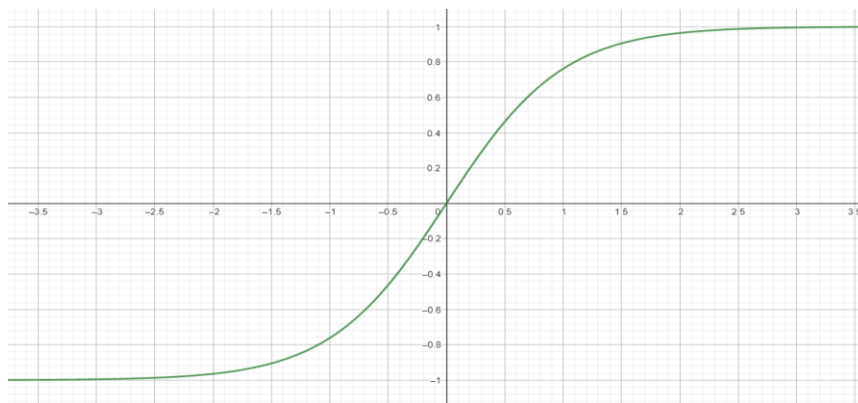
Aktivacijske funkcije predstavljaju osnovu za učenje mreže, one su te koje omogućuju njenu nelinearnost. Za aktivacijsku ili prijenosnu funkciju obično se koristi tangens-hiperbolni ili sigmoida, no odabir funkcije ovisi o zadatku koji je postavljen pred neuronsku mrežu.



Slika 8. Sigmoidna aktivacijska funkcija

$$S(x) = \frac{1}{1+e^{-x}} \quad (13)$$

Sigmoidna funkcija (13) uvodi nelinearnost u model neuronske mreže pri čemu izlazne veličine poprimaju vrijednosti od 0 do 1 (slika 8.). Sigmoidna funkcija donosi aktivaciju na jedan od krajeva krivulje (za vrijednosti više od 2 i manje od -2) [14]. Drugim riječima, gradijent u tom području je gotovo ništavan čime je onemogućeno učenje tj. promjena težinskih koeficijenata. Vrlo je raširena upotreba sigmoidne funkcije kod problema klasifikacije. Zbog faktora e^x vrlo je zahtjevna za računanje pa mreža može sporo konvergirati.



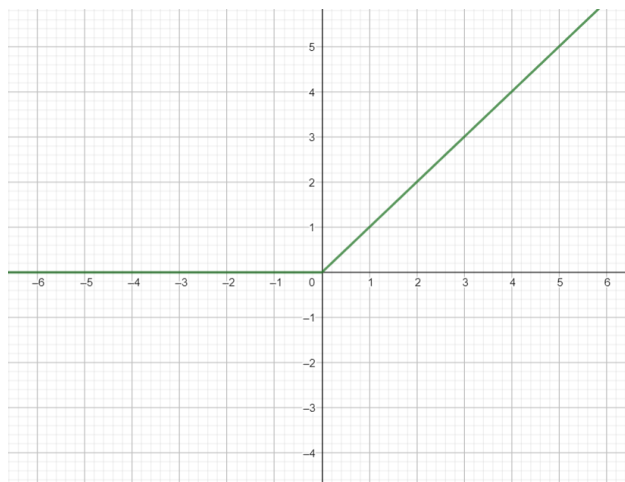
Slika 9. Tangens hiperbolni aktivacijska funkcija

$$f(x) = \text{th}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (14)$$

Kod funkcije tangens hiperbolni (14) bitno je napomenuti da je gradijent veći nego kod sigmoidne funkcije [14]. Problemi koji se javljaju kod funkcije tangens hiperbolni:

- Zasićeni neuron (kada su izlazne vrijednosti neurona bliske ekstremnim vrijednostima -1 i 1) uzrokuje nestanak gradijenta (slika 9.)
- Zbog člana funkcije e^x zahtijeva puno računalne snage [15]

Funkcija tangens hiperbolni koristi se uglavnom kada je riječ o problemu klasifikacije.

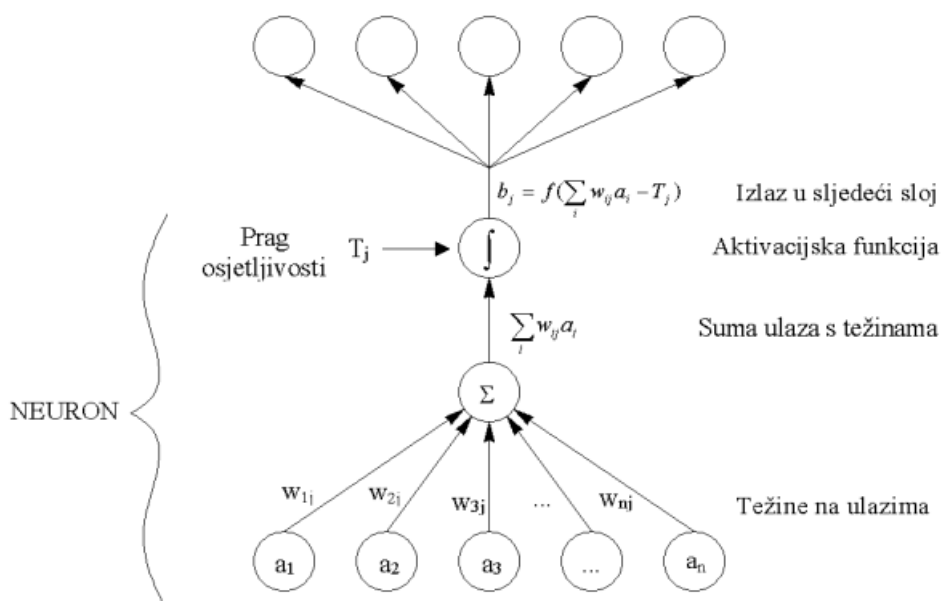


Slika 10. ReLU aktivacijska funkcija

$$f(x) = \frac{x + |x|}{2} \quad (15)$$

$$f(x) = \max(0, \sum_{i=1}^{i=n} w_i x_i + b) \quad (16)$$

ReLU funkcija (15) za sve negativne vrijednosti daje nulu, a za sve pozitivne vrijednosti izbacuje istu tu vrijednost. Zbog svoje jednostavnosti često je korištena kao aktivacijska funkcija. Prednost je što ne dolazi do zasićenja neurona kod velikih vrijednosti ulaza [15]. Nedostatak ReLU funkcije je što ne unosi nelinearnost u mrežu i stoga ne može opisati nelinearne relacije među podacima, te ukoliko suma svih otežanih ulaza ima veću negativnu vrijednost nego što iznosi prag (engl. *bias*) (16) tada ne dolazi do aktivacije neurona.



Slika 11. Shema perceptrona [8]

Imajući na umu sustav kojim funkcioniraju umjetni i biološki neuron može se izvući nekoliko poveznica među njima:

- Dendriti imaju ulogu težinskih koeficijenata, oni signal pojačavaju ili ga prigušuju.
- Tijelo stanice ima ulogu postavljanja praga, tj. odlučuje da li je signal dovoljno jak da pobudi stanicu.
- Akson pretvara dani ulaz u izlaz, prema tome on je analogan prijenosnoj funkciji kod perceptrona.

2.5.1. Učenje neuronske mreže

Neuronska mreža je specificirana kada su zadani neuroni u sva tri sloja, definiran prag (*bias*), kao i prijenosna funkcija te težinski koeficijenti među neuronima. Upravo su težinski koeficijenti nosioci „znanja“ u neuronskoj mreži, a do njihovih vrijednosti se dolazi treniranjem, tj. učenjem mreže na određenom skupu ulaznih podataka. Pojam treniranje ili učenje neuronske mreže predstavlja iteracijski postupak podešavanja veza među neuronima. Podešavanje težinskih koeficijenata odvija se pomoću signala pogreške (engl. *Loss*) između izlaza koji je dobiven modelom i stvarne vrijednosti izlazne veličine izmjerene u laboratoriju ili dobivene iz informacijskih sustava s postrojenja.

Učenje neuronske mreže odvija se prema nekom pravilu učenja kao što je pravilo širenja unatrag (eng. *Backpropagation*). Razlika vrijednosti dobivene modelom i stvarne vrijednosti služi za korekciju težinskog koeficijenta pomoću tzv. Delta pravila:

$$w_{ji}^{novi} = w_{ji}^{stari} + \Delta w_{ji} \quad (17)$$

Pri čemu w_{ji} označava neuron j povezan sa neuronom i .

$$\Delta w_{ji} = \eta y_{cj} \varepsilon_i \quad (18)$$

Pri čemu je η koeficijent učenja (engl. *learning rate*), y_{cj} vrijednost izlaza koji dolazi od neurona j , ε_i vrijednost pogreške izračunate prema (19):

$$\varepsilon_i = \text{vrijednost koju daje neuronska mreža} - \text{stvarna vrijednost} \quad (19)$$

Globalna greška mreže se računa kao:

$$E = \frac{1}{2} \sum_k (d_k - x_k)^2 \quad (20)$$

pri čemu je k broj neurona u izlaznom sloju, d_k vrijednost dobivena eksperimentom, a x_k vrijednost veličine dobivene pomoću neuronske mreže.

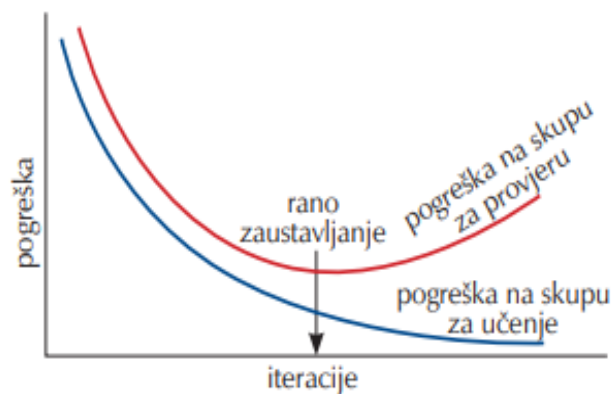
Koeficijent učenja je zadani parametar koji diktira brzinu promjene težinskih koeficijenata, uzimajući u obzir vrijednost pogreške ε (18). Važno je napomenuti da premala vrijednost koeficijenta učenja može rezultirati presporim učenjem, a ukoliko se postavi prevelika vrijednost tada se nikada ne može pronaći lokalni minimum funkcije pogreške što se može vidjeti na odnosu pogreške kroz vrijeme učenja na slici 12. Na apscisi se nalazi broj epoha koja označava jedan prolaz podataka kroz cijelu mrežu.



Slika 12. Utjecaj vrijednosti koeficijenta brzine učenja na treniranje mreže [20]

Da bi se dobiveni ulazni podaci mogli upotrijebiti za treniranje često ih je potrebno prethodno filtrirati kako bi se uklonile smetnje i varijabilnosti karakteristične za realna mjerenja [19].

Nakon toga se obrađeni podaci dijele u tri skupa: skup za učenje, skup za provjeru i skup za testiranje. Podaci iz skupa za učenje služe za ugađanje vrijednosti težinskih faktora među neuronima sve dok pogreška izlaza ne dosegne neku minimalnu vrijednost. Skup za provjeru služi za provjeru modela u smislu da li je došlo do pretreniranosti mreže, što je slučaj kada neuronska mreža daje odlične rezultate na skupu podataka na kojem je učila, dok na novim podacima pokazuje vrlo loše rezultate.



Slika 13. Tehnika ranog zaustavljanja [8]

U svrhu sprječavanja pretreniranja mreže koristi se tehnika „ranog zaustavljanja“ (engl. *early stopping*). Treniranje mreže valja zaustaviti onda kada pogreška na skupu za učenje i dalje pada, ali pogreška na skupu za provjeru dostiže svoj minimum što je prikazano na slici 13.

2.5.2. Slijedni podaci

U realnim sustavima procesne veličine su uglavnom zavisne o vremenu, stoga ih možemo odvojiti po sekvencama (slijedovima). Sekvencijalni podaci su oni podaci koji imaju svoj kontekst uzimajući u obzir druge podatke koji se nalaze u nizu. Naprimjer, to mogu biti cijene dionica na tržištu, vremenska prognoza, ili pak pojedini pikseli unutar jedne fotografije. Podaci koji su dobiveni iz rafinerije su kontinuirani i kronološki tj. imaju jasno definiran vremenski redoslijed, dakle to su strukturirani podatci poredani u vremenskom slijedu. Sekvencijalni podaci se nalaze svuda u svijetu oko nas. Kada ljudi razgovaraju riječi koje dolaze od sugovornika dobivaju značenje u kontekstu riječi koje su bile izgovorene prije, ali također i onih koje će sugovornik tek izgovoriti. Vrlo zoran primjer je običan tekst, poput ovoga, riječi u ovome tekstu same za sebe imaju vrlo ograničeno značenje te vrlo teško mogu prenijeti neku informaciju. Ponegdje taj kontekst ovisi o npr. mjestu u rečenici (ili u tekstu), ponekad ovisi o drugim pikselima koji se nalaze oko promatranog dijela fotografije (kod grafičkog prepoznavanja), a ponekad je kontekst vremenski zavisan. Za razvoj modela s podacima koji su vremenski zavisni (a onda je i kontekst u neku ruku vremenski) najbolje su se pokazale dinamičke neuronske mreže.

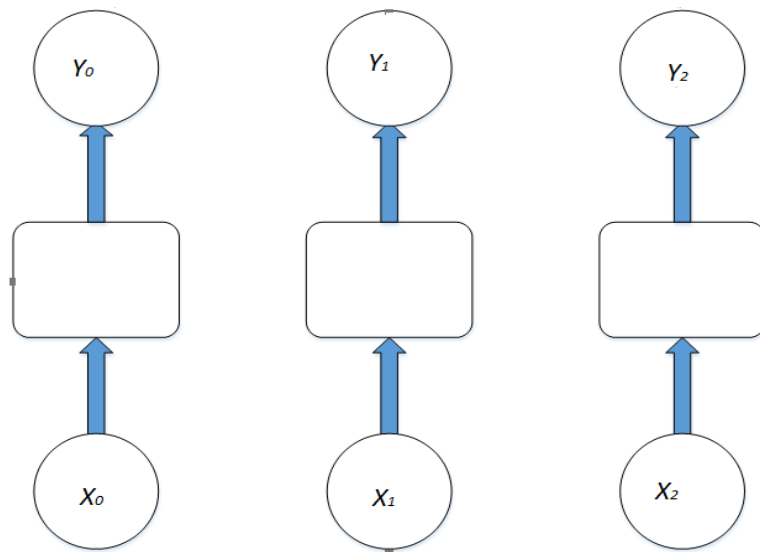
2.5.3. Dinamičke neuronske mreže

Osnovna ideja dinamičkog neurona je da uz prijenosnu funkciju posjeduje i određeno pamćenje vlastitih prošlih vrijednosti i na taj način pamti određeni kontekst. Dinamičke neuronske mreže su stoga korisne za obradu sekvencijalnih podataka kod kojih je bitan kontekst u vremenu, te za obradu podataka vremenskih nizova (engl. *time series*).

2.5.4. Rekurentne (povratne) neuronske mreže

Da bi se objasnile rekurentne neuronske mreže, prikladno je za početak razmotriti načela statičke neuronske mreže. Kada bi statičkoj neuronskoj mreži (pod pretpostavkom da je ona već na odgovarajući način trenirana) za ulazne podatke predali podatke koji su vremenski

zavisni, imali bi niz podataka od kojih se svaki odnosi na jedan vremenski trenutak. Radi zornog objašnjenja načelo rada pokazat će se na mrežama koje imaju jedan ulaz i jedan izlaz.

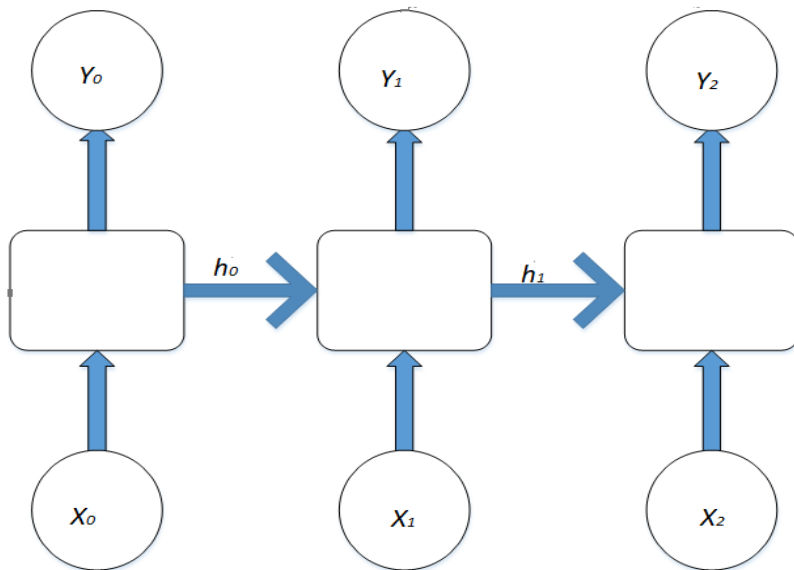


Slika 14. Statička neuronska mreža

$$y_t = f(x_t) \quad (21)$$

Obzirom da slijedni podaci zavise jedni o drugima, to znači da vrijednost y_2 ovisi ne samo o vrijednosti x_2 , kako je to prikazano pomoću (21), već i o vrijednosti ulaza u prethodnim vremenskim intervalima x_1 i x_0 . Međutim, postavlja se pitanje kako sve te prošle vrijednosti povezati.

Osnovna ideja dinamičke neuronske mreže je povezivanje vrijednosti različitih vremenskih intervala pomoću interne memorije (engl. *internal memory*), h_t kako je to prikazano jednadžbom (22) (slika 15).



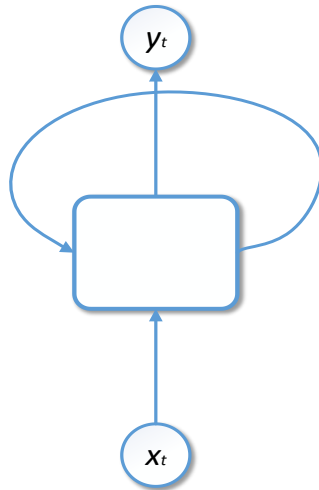
Slika 15. Interna memorija

$$y_t = f(x_t, h_{t-1}) \quad (22)$$

Pri čemu je:

- y_t – izlaz koji mreža daje za neki trenutak t
- x_t – ulaz u mrežu u trenutku t
- h_{t-1} – interna memorija

Interna memorija u sebi sadrži kontekst koji dolazi od svih prethodnih podataka. Pri tome, nakon svakog vremenskog trenutka interna memorija se ažurira, poprima novu vrijednost koja se koristi u sljedećem trenutku. Za neurone koji imaju u sebi zatvorenu petlju u kojoj prošla vrijednost (interna memorija) ulazi u sadašnji trenutak se kaže da posjeduju povratnu ili rekurentnu vezu (slika 16.), stoga se takve mreže nazivaju rekurentne neuronske mreže.



Slika 16. Rekurentni neuron

Nova ažurirana vrijednost interne memorije koja se koristi u vremenskom intervalu t , računa se na temelju ulaznih vrijednosti u tom trenutku te vrijednosti interne memorije u prethodnom vremenskom trenutku (jednadžba 23):

$$h_t = f_w(x_t, h_{t-1}) \quad (23)$$

Pri čemu je f_w funkcija sa definiranim setom parametara (jednom kad je mreža trenirana) koji definiraju odnos između ulaza x_t , prethodne vrijednosti interne memorije h_{t-1} i nove vrijednosti interne memorije h_t . Ista funkcija i iste vrijednosti težinskih faktora dobivene treniranjem se koriste za sve vremenske korake. Drugim riječima, interno stanje se ažurira temeljem prošlog internog stanja i trenutačnog ulaza.

Funkcija sa definiranim setom parametara je zapravo matrica sa pripadajućim težinskim koeficijentima. Prije izračuna vrijednosti izlaza, mreža računa novu vrijednost interne memorije koja u sebi sadrži vrijednost stare interne memorije i trenutne vrijednosti ulaza. Ta vrijednost nove ažurirane interne memorije koristi se za izračun vrijednosti trenutnog izlaza:

Ulazni vektor: x_t

↓

$$\text{Ažuriranje interne memorije: } h_t = f(W_{hh}^T h_{t-1} + W_{xh}^T x_t) \quad (24)$$

↓

$$\text{Izlazni vektor: } y_t = W_{hy}^T h_t \quad (25)$$

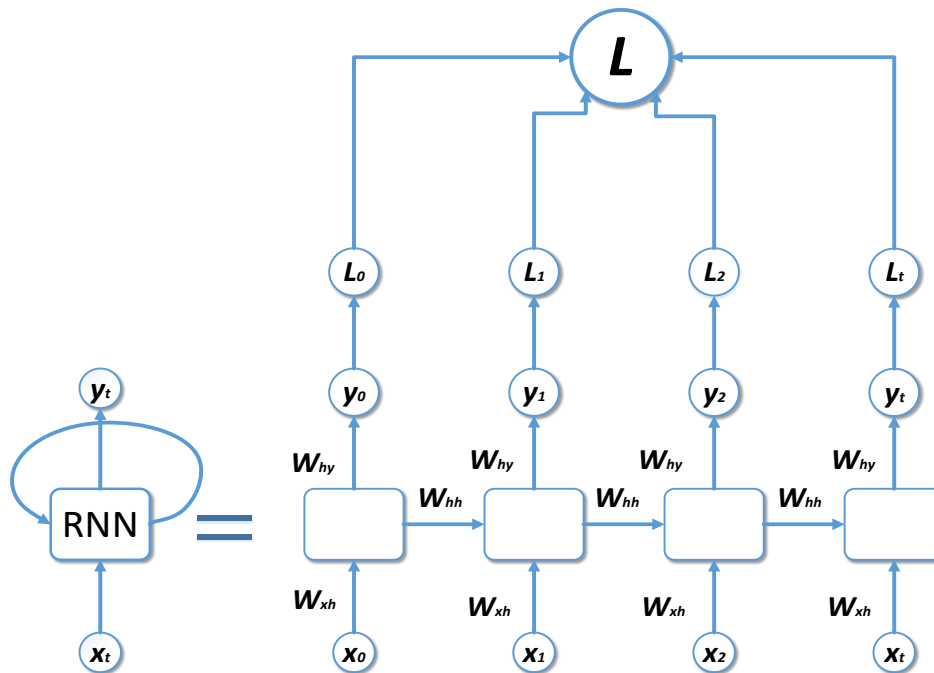
Matrice težinskih koeficijenata različite su za ulazne parametre h_{t-1} i x_t .

Pri čemu su:

- W_{xh}^T – matrica težinskih koeficijenata koja definira kako će ulazi, za svaki vremenski interval, biti transformirani u internu memoriju h_t .
- W_{hh}^T – matrica težinskih koeficijenata koja definira vezu između prošle i nove vrijednosti interne memorije.
- W_{hy}^T – matrica težinskih koeficijenata koja transformira novu vrijednost interne memorije u izlaz za dani vremenski korak.
- h_{t-1} – stara vrijednost vektora interne memorije
- x_t – trenutna vrijednost vektora ulaza

Bitno je naglasiti da jednom kad se ugone težinski koeficijenti u matricama, oni ostaju nepromijenjeni, te se mogu mijenjati samo ponovnim treniranjem mreže.

2.5.5. Učenje rekurentnih neuronskih mreža



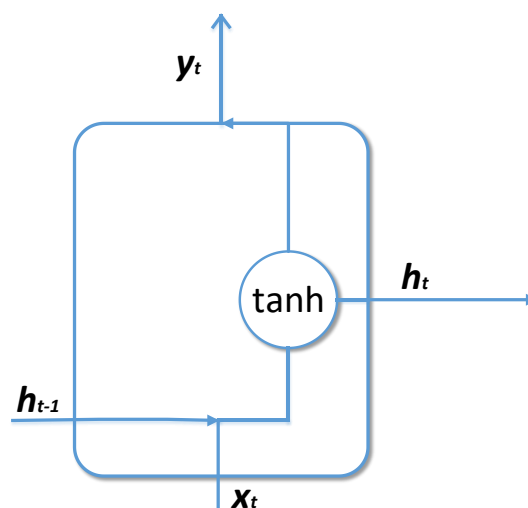
Slika 17. Rekurentna neuronska mreža odmotana u vremenu

U svakom vremenskom koraku lokalno se računa pogreška $L_0, L_1 \dots$ (engl. *Loss*) koja se zbraja kako bi se dobila globalna pogreška cijelog slijeda podataka L . Ta pogreška se koristi kao „pokretačka sila“ za ažuriranje težinskih koeficijenata unatražnom propagacijom kroz vrijeme (engl. *backpropagation*). Pogreška se tako prenosi od trenutka t sve do početnog trenutka koji je najdalje u prošlosti. Međutim, velika vremenska udaljenost između slijednih podataka može dovesti do smanjenja, ili pak enormnog povećanja gradijenta pogreške (koji služi kao pokretačka sila učenja), pa se tako mogu javiti dva problematična slučaja. Prvi je pojava nestajućeg, tj. jako opadajućeg gradijenta (engl. *vanishing gradient*) pogreške, a drugi je pak pojava eksplodirajućeg gradijenta pogreške, tj. on počne rasti prema ogromnim vrijednostima (engl. *exploding gradient*). Smanjenje gradijenta pogreške može se javiti ako je slijed podataka suviše dugačak jer pokretačka sila promjene težinskog koeficijenta svakog prethodnog vremenskog koraka ovisi o promjeni u koraku koji je ispred njega u vremenu. Dakle, ako u nekom vremenskom intervalu iz bilo kojeg razloga nastupi mala promjena, tada će u prethodnom trenutku promjena biti još manja, u trenutku prije toga još manja itd., sve dok promjena ne postane nezatna, a učenje nemoguće. Posljedica toga je nezatna promjena težinskih koeficijenata kroz vremenske korake iz dalje prošlosti što ne doprinosi

više učenju mreže te se zato učenje mreže prekida. Problem smanjenja gradijenata pogreške izrazito je bitan jer se zbog malenih vrijednosti može izgubiti bitna informacija (vrijednost) između vremenskih koraka, što može značiti da neka bitna informacija ili parametar iz povijesti neće sudjelovati u izračunavanju buduće vrijednosti. Dakle, mreža više ne pamti što je bilo prije dva ili više koraka u vremenu, a što je u biti glavna prednost dinamičkih neuronskih mreža.

2.5.6. Čelija s dugoročnom memorijom – LSTM

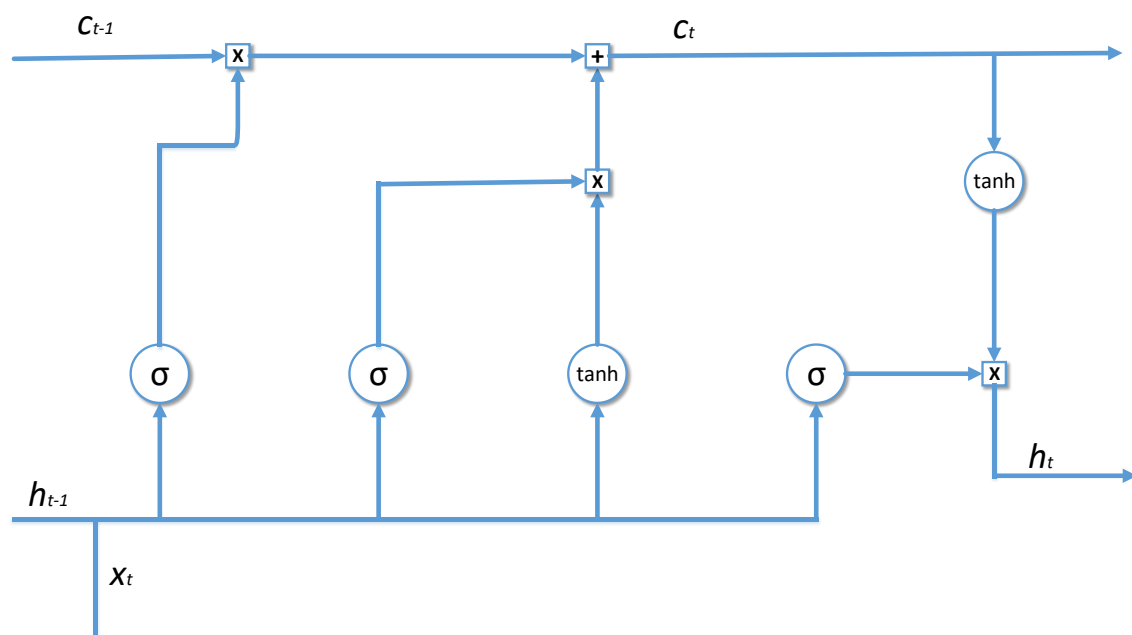
Kako bi se riješio problem eksplodirajućeg/nestajućeg gradijenta, tj. problem učenja onih neurona koji se nalaze u ranim vremenskim intervalima uvodi se koncept pod nazivom *dugotrajno kratkoročno pamćenje*, engl. *Long Short Term Memory* (LSTM). Najvažnija odlika LSTM mreža je da mogu upravljati protokom informacija. LSTM je posebna vrsta rekurentnih neuronskih mreža koje su prvi put predstavili Hochreiter & Schmidhuber (1997) i od tada su postale vrlo popularne [21]. Glavna karakteristika LSTM mreža je da se informacija koja se prosljeđuje internoj memoriji može pohraniti ili ukloniti pomoću strukture koja se naziva 'propusnica' (eng. *gate*). Propusnice predstavljaju zapravo još jednu neuronsku mrežu koja je nakon odgovarajućeg treniranja u stanju odlučiti koja informacija je bitna, tj. koja informacija prolazi (1), a koja ne prolazi (0). Zahvaljujući svojoj unutarnjoj strukturi LSTM može upravljati protokom informacija, stoga može prenijeti bitne informacije i kroz dugački slijed podataka.



Slika 18. Struktura rekurentnog neurona

Kod obične rekurentne ćelije (slika 18.) ulaz i prethodna vrijednost interne memorije čine vektor koji u sebi sadrži informaciju sadašnjeg i prošlih ulaza. Taj vektor prolazi kroz \tanh aktivacijsku funkciju, koja omogućava da vrijednosti ostanu između -1 i 1, pri čemu se dobije nova vrijednost interne memorije.

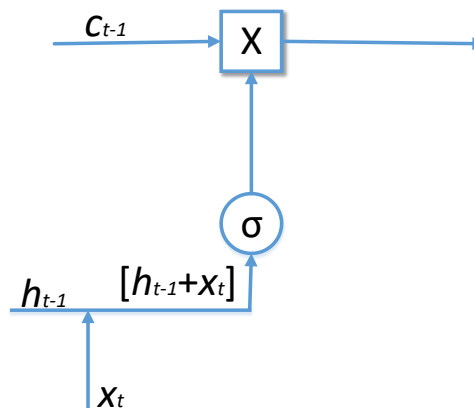
Sušтина LSTM ćelija je da mogu obrađivati podatke koji su daleko u vremenu, a razlika u odnosu na običnu rekurentnu ćeliju je u načelu samo u operacijama koje se provode (slika 19). Kako bi se razumjelo kako LSTM ćelija funkcionira potrebno je uvesti pojam **stanja ćelije**, c_t (engl. *Cell State*). Stanje ćelije je veličina koja u sebi sadrži i kojom se prenose relevantne informacije kroz slijed podataka, a može se shvatiti i kao neka vrsta memorije mreže. Stanje ćelije na svome 'putu' dobiva ili ne dobiva informacije pomoću strukture koja se naziva **vrata** ili **propusnica** (engl. *gate*). Svaka od tih vrata su zapravo još jedna neuronska mreža unutar LSTM rekurentnog neurona, koji donosi odluku što smije proći u stanje ćelije [16]. Tokom učenja mreža propusnice također 'uče', tj. optimiraju svoje vrijednosti težinskih koeficijenata. Načelo rada stanja ćelije se bazira na tome da se prvo prethodno stanje ćelije množi sa vektorom zaboravljanja, a zatim se tako dobivena vrijednost zbraja sa vrijednosti koja je dobivena pomoću ulazne propusnice, što sveukupno daje novu vrijednost stanja ćelije. Drugim riječima stanje ćelije se može predočiti kao „put“ kojom relevantne informacije putuju kroz niz vremenskih intervala. Što će od informacija utjecati na stanje ćelije, je zadaća propusnica.



Slika 19. Struktura LSTM ćelije

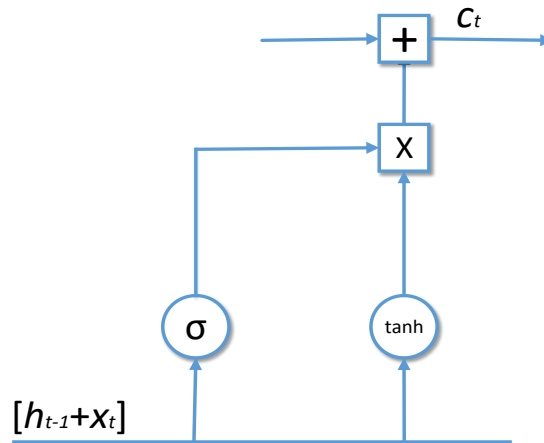
U strukturi LSTM ćelije postoje četiri propusnice:

- **Propusnica zaboravljanja** (engl. *Forget gate*) – zaboravlja nebitnu povijest tako što se prethodna vrijednost interne memorije i trenutna vrijednost ulaza propuste kroz sigmoidnu funkciju koja skalira vrijednosti od 0 do 1. Vrijednosti koje su bliže 0 se 'zaboravljaju', tj. ne propuštaju, a vrijednosti bliže 1 se zadržavaju. Izlaz sigmoidne funkcije se pomnoži sa prijašnjim stanjem ćelije c_{t-1} .



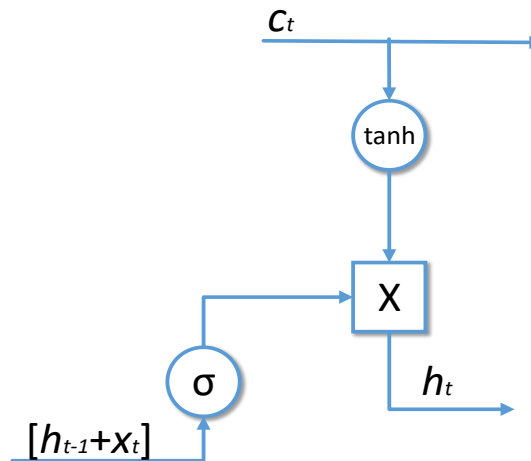
Slika 20. Propusnica zaboravljanja

- **Ulazna propusnica** (engl. *Input Gate*) – služi za ažuriranje stanja ćelije. Sastoji se od dvije mreže, tj. u njemu su dvije aktivacijske funkcije, sigmoidna i tangens hiperbolna od kojih svaka ima specifičnu ulogu. Sigmoidna funkcija odlučuje što će se propustiti dalje na ažuriranje, a funkcija tangens hiperbolni transformira veličine u rasponu od -1 do 1. Vrlo je bitno skalirati veličine u rasponu od -1 do 1 kako ne bi tokom više vremenskih koraka neke vrijednosti jako porasle (njihov red veličine bi se dosta povećao) zbog čega bi se ostale mogle učiniti nebitnim. Izlaz sigmoidne funkcije odlučuje koji dio izlaza funkcije tangens hiperbolni je bitan putem operacije množenja vrijednosti. Tako dobivena vrijednost se zbraja sa izlazom iz propusnice zaboravljanja i dobiva se novo stanje ćelije c_t .



Slika 21. Ulazna propusnica

- **Izlazna propusnica** (engl. *Output Gate*) – odgovorna je za generiranje nove vrijednosti skrivenog stanja h_t koje u sebi sadrži informacije od ulaza u prijašnjim trenucima. Prošla vrijednost skrivenog stanja i vrijednost sadašnjeg ulaza čine vektor koji ulaze u sigmoidnu funkciju. Nova vrijednost stanja ćelije c_t prolazi kroz *tanh* funkciju. Ta dva izlaza se množe, pri čemu izlaz sigmoidne funkcije ima ulogu odlučivanja koje informacije će sadržavati nova vrijednost skrivenog stanja h_t .



Slika 22. Izlazna propusnica

Načelo rada propusnica može se prikazati i pomoću Python **pseudokoda**:

```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
```

```

it = input_layer(combine)
candidate = candidate_layer (combine)
Ct = prev_ct*ft + candidate*it
ot = output_layer (combine)
ht = ot*tanh(Ct)
Ct = [0,0,0]
Ht = [0,0,0]
for input in inputs:
    ct,ht = LSTMCELL(ct,ht,input) [16]

```

1. Prethodno skriveno stanje i sadašnji ulaz čine vektor $[h_{t-1}+x_t]$, što se naziva combine.
2. combine ulazi u forget layer, čija je uloga uklanjanje nebitnih podataka.
3. candidate layer je kreiran pomoću naredbe combine. candidate u sebi sadrži potencijalne vrijednosti koje će postati dio stanja ćelije.
4. combine ulazi u input layer koji odlučuje koji podaci iz candidate će ući u novo stanje ćelije.
5. Nakon izračuna forget layer, candidate layer i input layer, stanje ćelije se računa pomoću njih i prethodnog stanja ćelije.
6. Izračuna se vrijednost output.
7. Množenjem vrijednosti output sa novim stanjem ćelije (koje je prošlo kroz tanh funkciju) dobiva se novo skriveno stanje. [16]

Načela LSTM mreže:

- Koriste „propusnice“ radi kontrole protoka informacija tako što zaborave nebitne i spremne bitne informacije od ulaznog vektora.
- Omogućuje unatražnu propagaciju kroz vrijeme sa nesmetanim protokom gradijenta pogreške što je ključno za ostvarenje efikasnog učenja mreže čak i kod dugačkih sljedova podataka.

2.6. Korišteni softver za razvoj modela

Kako bi se razvio i primijenio model neuronske mreže u ovome radu korišten je programski jezik Python te Keras i *NumPy* kao njegove programske knjižnice.

2.6.1. Python

Python je programski jezik općenite namjene razvijen tokom kasnih 1980-ih kao nadogradnja ABC programskog jezika. Koristi se za razvoj softvera, izradu web stranica, te se vrlo često koristi za primjenu metoda znanosti o podacima (engl. *data science*).

Keras

Keras je programsko sučelje (engl. *Application Programming Interface, API*) razvijeno za potrebe dubokog učenja koje u svojoj pozadini koristi platformu TensorFlow, CNTK ili Theano. Razvijen je kako bi se omogućio brz i jednostavan razvoj modela dubokog učenja. Ključne odlike Kerasa su da omogućava pokretanje koda pomoću CPU (engl. *Central Processing Unit*) ili pomoću GPU (engl. *Graphics Processing Unit*) istovremeno. Keras ima relativno jednostavno korisničko sučelje zbog čega omogućava relativno brzu izradu modela dubokog učenja, unaprijednih neuronskih mreža, rekurentnih neuronskih mreža ili njihove kombinacije. [17]

NumPy

NumPy je knjižnica u Pythonu koja se koristi za rad sa poljima, a također može raditi u području linearne algebre, Fourierovih transformacija i matrica. U Pythonu već postoje liste koje služe za rad s poljima, međutim sam proces je vrlo spor. Radi toga je smišljen NumPy koji iste operacije može izvršiti 50 puta brže. Polja NumPy-a su spremljena na memoriji za razliku od lista, stoga procesi mogu vrlo efikasno pristupiti i manipulirati poljima [18].

Spyder

Spyder (engl. *Scientific Python Development Environment*) je IDE (engl. *integrated development environment*) platforma za programiranje unutar *Python* programskog jezika. U sebi sadrži pakete poput NumPy, SciPy, Matplotlib, pandas, IPython, SymPy i Cython.

3. Eksperimentalni dio

Prva primjena procesa hidrokrekinga seže u godinu 1959. u Chevronovoj rafineriji u Richmondu [9]. Porast cijena sirove nafte nakon 1973. godine te sve teže raspoložive nafte dovele su do masovne upotrebe hidrokreking procesa. Proces hidrokrekinga se odvija pri visokom tlaku vodika te je vrlo fleksibilan pa se udjeli konačnih produkata mogu mijenjati ovisno o potrebama tržišta. Proizvodi hidrokrekinga su:

- UNP
- Laki benzin (C₅-85 °C)
- Benzin (85-180 °C)
- Petrolej
- Plinsko ulje
- Ostatak

Postoje tri tipa hidrokreking procesa koji se međusobno razlikuju po procesnim uvjetima:

- Blagi hidrokreking BHK
- Oštri hidrokreking OHK
- Hidrokreking teških ostataka HKO

Procesni uvjeti su tlak, temperatura, prostorna brzina, omjer H₂/sirovina i vrsta katalizatora.

Utjecaj tlaka:

- Povećava kvalitetu proizvoda, ali također raste temperatura procesa i potrošnja vodika.
- Ukoliko je tlak previsok može oštetiti ili ozbiljno smanjiti životni vijek katalizatora.

Temperatura:

Na kraju ciklusa temperatura je viša nego na početku jer dolazi do postepene deaktivacije katalizatora. Ona se definira za pojedinu sirovinu i traženi stupanj konverzije. [9]

Prostorna brzina:

Ovisi o vrsti sirovine te o traženoj konverziji [9]. Veća prostorna brzina znači manje zadržavanje vodika tj. veći protok svježije struje stoga se povećanjem prostorne brzine

povećava i životni vijek katalizatora. Ukoliko je riječ o niskim vrijednostima tada su potrebni veliki reaktori i velike količine katalizatora.

Potrošnja vodika:

Kako je sirovina koja ulazi u proces veće prosječne molekulske mase tada je potrebno više vodika da bi dobili visokovrijedne naftne derivate. Kod procesa blagog hidrokrekinga potrošnja vodika iznosi 70-200 Nm³/m³ sirovine, dok kod hidrokrekinga vakuum ostatka taj je omjer 600-1000 Nm³/m³ sirovine[9].

Konverzija:

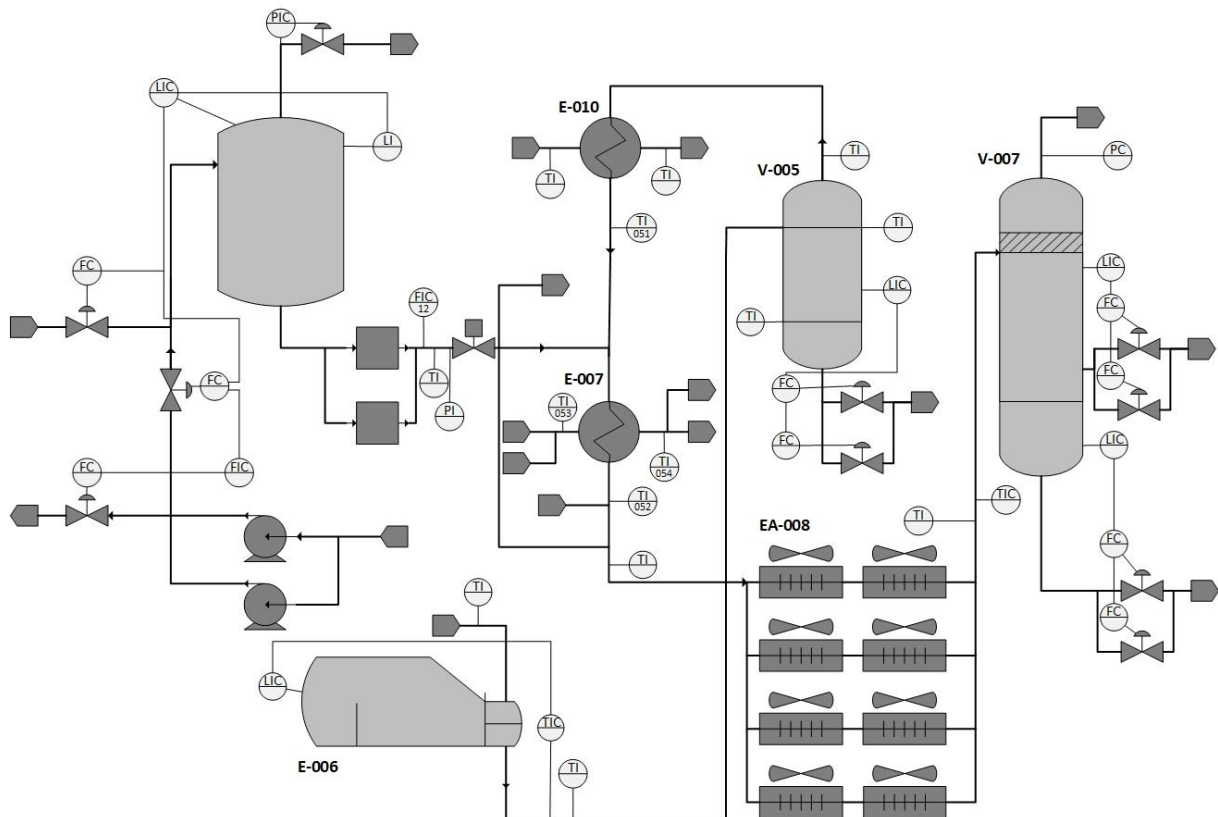
Svi gore navedene procesne varijable na kraju utječu na konverziju pa ukoliko je želimo povećati treba se povećati temperatura, parcijalni tlak vodika i omjer H₂/sirovina, a smanjiti prostornu brzinu [9]. Radi smanjenja prostorne brzine potrebno je povećati postrojenje ili smanjiti njegov kapacitet. Uz veću konverziju također raste i brzina deaktivacije katalizatora.

Kod hidrokrekiranja odvijaju se paralelno dva procesa, katalitičko krekiranje (razbijanje molekula) i katalitička hidrogenacija (obogaćivanje vodikom). Najpoznatiji katalitički sustavi su sintetički zeoliti uz metalne okside ili sulfide, od kojih su vrlo djelotvorni oksidi Co-Mo i Ni-Mo te sulfidi Ni-W [10].

Izmjenjivač topline na kojemu se provodila analiza nastajanja naslaga se nalazi unutar postrojenja za hidroobradu i hidrokrekiranje pod oznakom E-007. Sirovina se prvo predgrijava toplinom dizelskog produkta koji dolazi iz sekcije frakcionacije, zatim odlazi u filter gdje se uklanjaju velike čestice koje mogu prouzročiti probleme u radu reaktora. U procesnoj peći se sirovina zagrijava do ciljane temperature. Na vrhu reaktora nalazi se sloj katalizatora na bazi Ni-Mo kako bi se odvijala reakcija hidroobrade (uklanjanje sumpora i dušika) prije samog hidrokrekinga.

Nakon predgrijavanja sirovine efluent reaktora se koristi za proizvodnju visokotlačne pare. Tako ohlađeni efluent reaktora ulazi u HHPS separator (eng. *Hot High Pressure Separator*). Ulazna struja se separira na vodikom bogatu paru i ukapljene ugljikovodike. Vruća para iz HHPS separatora se hladi i djelomično kondenzira predgrijavanjem recikliranog plina iz reaktora u izmjenjivaču E-007. Tijekom hlađenja HHPS pare dolazi do taloženja amonij

bisulfida (NH_4SH) u cijevima izmjenjivača topline. U izmjenjivaču su ugrađene posebne mlaznice koje povremeno ispiru nastali talog s vodom. HHPS para nastavlja u CHPS separator (eng. *Cold High Pressure Separator*) gdje se dijeli na vodikom bogatu paru, ulje i kisele vode. CHPS para odlazi u visokotlačni amin apsorber gdje se otopinom MDEA uklanja H_2S . Čisti vodik se komprimira i vraća u reaktor. Na slici 23. prikazan je dio postrojenja hidrokrekinga u kojem se nalazi ispitivani izmjenjivač topline E-007.



Slika 23. Shema procesa hidrokrekinga – dio postrojenja s ispitivanim izmjenjivačem topline

4. Metodika

4.1. Konfiguracije modela neuronske mreže u Python-u

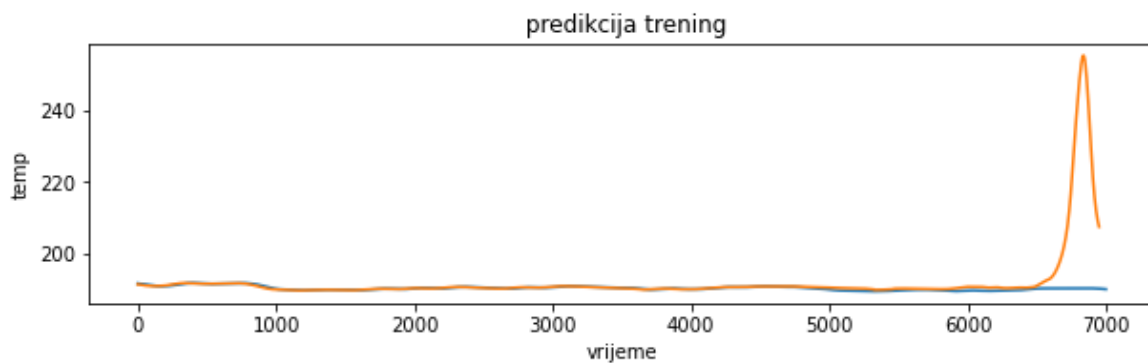
Arhitektura neuronske mreže u Python-u zadaje se pomoću funkcije `model`, a ona predstavlja način organizacije slojeva. U ovome radu se pomoću naredbe `model=Sequential()` koristio sekvencijalni model. Sekvencijalni model predstavlja linearni stog slojeva i pogodan je za upotrebu tamo gdje svaki sloj ima po jedan tenzor na ulazu i na izlazu [23]. Slojevi se dodavaju pomoću naredbe `model.add()` koja kao parametar unutar zagrade prima tip sloja. U ovome slučaju radi se o sloju LSTM i sloju *Dense* koji predstavlja izlazni sloj. Radilo se sa dva LSTM sloja pri čemu su ispitivane kombinacije mreža i sa samo jednim slojem. Kod dodavanja LSTM ćelije u skriveni sloj važno je zadati i prijenosne (aktivacijske) funkcije koje će se koristiti unutar samog LSTM, a to su *activation* (odgovara stanju ćelije i skrivenom stanju) i *recurrent activation* (odgovara prijenosnoj funkciji koja se koristi u propusnici zaborava, ulaza i izlaza). Jednom kada je zadana arhitektura mreže, tada se definira proces učenja pomoću naredbe `model.compile()` gdje se definira metoda optimiranja (tzv. optimizator) i funkcija pogreške. Optimizator je algoritam ili metoda optimiranja koja se koristi za procjenu parametara neuronske mreže kao što su težinski koeficijenti i koeficijent učenja kako bi se minimizirala funkcija pogreške [21]. U ovome radu koristio se veći broj dostupnih optimizatora kao što su *Nadam*, *Adam*, *Adadelta* i *RMSprop*. Za treniranje modela koristi se naredba `model.fit()` u okviru koje se zadaju parametri kao što su broj epoha (broj prolaza svih podataka kroz mrežu) te *batch size* (broj uzoraka sljedova podataka koji se obrađuju kao jedna cjelina).

4.2. Optimizacija mreže

Za procjenu uspješnosti razvijene mreže koristio se kriterij srednjeg kvadratnog odstupanja *MSE* (engl. *Mean squared error*) (26). Pri tome su se pratila dva koeficijenta korelacije od kojih se jedan odnosi na set za učenje, a drugi na validaciju (vrednovanje). Na slici 24. je dan primjer pretreniranja neuronske mreže po kojemu izrazito dobro slaganje na skupu za učenje uzrokuje vrlo loše slaganje na skupu za validaciju (žuto je prikazana predikcija modela, plava boja označava eksperimentalne podatke).

Kako bi se pronašla optimalna (ili barem dovoljno dobra za primjenu) konfiguracija LSTM mreže korištena je metoda pokušaja i pogreške pri čemu se mijenjao broj koraka ulaznih varijabli u prošlost, tj. n_steps i to redom 3, 4, 5, 6, 7, 8, 9, 10, 20 i 50. Za svaki pojedini n_steps prvo su ispitivane konfiguracije sa samo jednim skrivenim slojem tako što su mijenjane prijenosne funkcije (tanh, sigmoid i ReLU) za svaki broj neurona u skrivenom sloju koji je povećavan redom 3, 5, 10, 15, 25, 30, 35, 40 i 50. Zatim je za sve konfiguracije sa jednim skrivenim slojem koje su imale srednje kvadratno odstupanje na skupu za učenje i validaciji barem 85% napravljena daljnja optimizacija uvođenjem drugog skrivenog sloja pri čemu je optimiran drugi skriveni sloj kombinirajući tri navedene prijenosne funkcije za svaki broj neurona u tom skrivenom sloju redom 3, 5, 10, 20, 30 i 40. Bitno je naglasiti pretpostavku da nema smisla u drugi sloj dodavati manje neurona nego u prvi. Ukoliko je uvođenje dodatnih neurona u drugi sloj pokazalo jasno smanjenje srednjeg kvadratnog odstupanja ispitivanje je prekinuto i ranije.

$$MSE = \frac{1}{n} \sum_{i=1}^n (d_k - x_k)^2 \quad (26)$$



Slika 24. Pretrenirani model

Tako je za svaki n_steps dobiveno po nekoliko najboljih mreža. Dobivene konfiguracije mreža (za hladnu struju) koje su imale srednje kvadratno odstupanje barem 90% na skupu za učenje i na skupu za validiranje su se dalje ispitivale mijenjanjem četiri navedena optimizatora. Važno je napomenuti da se svi navedeni rezultati odnose na hladnu struju (osim tablice 20.) pri čemu se pretpostavilo da će odabrane konfiguracije vrijediti i za toplu struju. Deset najboljih konfiguracija za hladnu struju je primijenjeno na razvoj modela za toplu struju, od toga je izabrana konfiguracija koja je dala najbolje rezultate (tablica 20.).

5. Rezultati i rasprava

Za hladni tok izmjenjivača topline ispitivale su se različite neuronske mreže promjenom broja koraka ulaznih varijabli u prošlost, dodavanjem drugog skrivenog sloja neurona te mijenjanjem broja neurona u svakom skrivenom sloju kao i mijenjanjem prijenosnih funkcija tangens hiperbolni, sigmoidna i ReLU što je prikazano u Tablicama 1-20.

Tablica 1. Najbolje LSTM mreže za $n_steps = 3$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}$ / % | $R_{validacija}$ / % |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------------|-------------------|----------------------|
| <i>sigmoid</i> | <i>relu</i> | 35 | <i>sigmoid</i> | <i>relu</i> | 50 | <i>Nadam</i> (0.001) | 98,1 | 92,2 |
| <i>tanh</i> | <i>tanh</i> | 35 | <i>sigmoid</i> | <i>sigmoid</i> | 40 | <i>Nadam</i> (0.001) | 98,1 | 93,5 |

Tablica 2. Najbolje LSTM mreže za $n_steps=4$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}$ / % | $R_{validacija}$ / % |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------------|-------------------|----------------------|
| <i>sigmoid</i> | <i>relu</i> | 35 | <i>sigmoid</i> | <i>sigmoid</i> | 50 | <i>Nadam</i> (0.001) | 99,1 | 92,2 |
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>tanh</i> | 30 | <i>Nadam</i> (0.001) | 98,6 | 91,7 |

Tablica 3. Najbolje LSTM mreže za $n_steps=5$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv u 2. sloju | rec_aktiv u 2. sloju | Neuron i u 2. sloju | Optimizator | $R_{trening}$ / % | $R_{validacija}$ / % |
|-------------------|----------------------|--------------------|------------------|----------------------|---------------------|----------------------|-------------------|----------------------|
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>relu</i> | <i>sigmoid</i> | 30 | <i>Nadam</i> (0.001) | 99,1 | 91,6 |
| <i>tanh</i> | <i>tanh</i> | 30 | <i>sigmoid</i> | <i>sigmoid</i> | 30 | <i>Nadam</i> (0.001) | 98,3 | 90,6 |

Tablica 4. Najbolje LSTM mreže za $n_steps=6$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>relu</i> | 40 | <i>sigmoid</i> | <i>relu</i> | 40 | Nadam(0.001) | 98,9 | 91,4 |
| <i>sigmoid</i> | <i>relu</i> | 40 | <i>sigmoid</i> | <i>tanh</i> | 40 | Nadam(0.001) | 98,6 | 91,5 |

Tablica 5. Najbolje LSTM mreže za $n_steps=7$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>relu</i> | <i>sigmoid</i> | 25 | <i>sigmoid</i> | <i>tanh</i> | 25 | Nadam(0.001) | 98,7 | 91,4 |
| <i>sigmoid</i> | <i>relu</i> | 50 | <i>sigmoid</i> | <i>tanh</i> | 50 | Nadam(0.001) | 97,8 | 91,9 |

Tablica 6. Najbolje LSTM mreže za $n_steps=8$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>tanh</i> | <i>tanh</i> | 3 | <i>sigmoid</i> | <i>tanh</i> | 5 | Nadam(0.001) | 98,2 | 91,7 |
| <i>sigmoid</i> | <i>relu</i> | 25 | <i>sigmoid</i> | <i>relu</i> | 50 | Nadam(0.001) | 99,2 | 91,2 |

Tablica 7. Najbolje LSTM mreže za $n_steps=9$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>tanh</i> | 30 | <i>tanh</i> | <i>tanh</i> | 30 | Nadam(0.001) | 99,3 | 90,7 |
| <i>sigmoid</i> | <i>tanh</i> | 30 | <i>relu</i> | <i>sigmoid</i> | 50 | Nadam(0.001) | 98,1 | 92,4 |

Tablica 8. Najbolje LSTM mreže za $n_steps=10$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>tanh</i> | 25 | <i>relu</i> | <i>sigmoid</i> | 30 | Nadam(0.001) | 98 | 91,9 |
| <i>sigmoid</i> | <i>tanh</i> | 25 | <i>tanh</i> | <i>sigmoid</i> | 40 | Nadam(0.001) | 98 | 92,9 |

Tablica 9. Najbolje LSTM mreže za $n_steps=20$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>relu</i> | <i>sigmoid</i> | 40 | <i>tanh</i> | <i>sigmoid</i> | 50 | Nadam(0.001) | 99,8 | 89,6 |
| <i>sigmoid</i> | <i>relu</i> | 50 | <i>tanh</i> | <i>relu</i> | 50 | Nadam(0.001) | 99,2 | 98,9 |

Tablica 10. Najbolje LSTM mreže za $n_steps=50$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>tanh</i> | 40 | Nadam(0.001) | 99,5 | 90 |
| <i>relu</i> | <i>sigmoid</i> | 50 | Nadam(0.001) | 99 | 88,7 |

Tablica 11. Konačna optimizacija LSTM mreže za $n_steps=3$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>tanh</i> | <i>tanh</i> | 35 | <i>sigmoid</i> | <i>sigmoid</i> | 40 | Nadam(0.001) | 98,1 | 93,5 |
| <i>sigmoid</i> | <i>relu</i> | 35 | <i>sigmoid</i> | <i>relu</i> | 50 | Nadam(0.001) | 98,1 | 92,2 |
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>tanh</i> | 40 | Nadam(0.001) | 98,5 | 90,1 |

Tablica 12. Konačna optimizacija LSTM mreže za $n_steps=4$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------|------------------|---------------------|
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>tanh</i> | 30 | RMSprop(0.001) | 97,9 | 93,1 |
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>tanh</i> | 30 | Nadam(0.001) | 98,6 | 91,7 |
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>relu</i> | 50 | Nadam(0.001) | 98,8 | 91,4 |

Tablica 13. Konačna optimizacija LSTM mreže za $n_steps=5$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------|------------------|---------------------|
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>relu</i> | <i>sigmoid</i> | 30 | RMSprop(0.001) | 99 | 90,9 |
| <i>tanh</i> | <i>tanh</i> | 30 | <i>sigmoid</i> | <i>sigmoid</i> | 30 | Nadam(0.001) | 98,3 | 90,6 |
| <i>sigmoid</i> | <i>relu</i> | 25 | <i>relu</i> | <i>sigmoid</i> | 25 | Nadam(0.001) | 99 | 90,4 |

Tablica 14. Konačna optimizacija LSTM mreže za $n_steps=6$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>relu</i> | 40 | <i>sigmoid</i> | <i>tanh</i> | 40 | Nadam(0.001) | 98,6 | 91,5 |
| <i>sigmoid</i> | <i>relu</i> | 40 | <i>sigmoid</i> | <i>relu</i> | 40 | Nadam(0.001) | 98,9 | 91,4 |
| <i>sigmoid</i> | <i>relu</i> | 40 | <i>sigmoid</i> | <i>relu</i> | 40 | Adam(0.001) | 99,5 | 90,6 |

Tablica 15. Konačna optimizacija LSTM mreže za $n_steps=7$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>relu</i> | 50 | <i>sigmoid</i> | <i>tanh</i> | 50 | Nadam(0.001) | 97,8 | 91,9 |
| <i>relu</i> | <i>sigmoid</i> | 25 | <i>sigmoid</i> | <i>tanh</i> | 25 | Nadam(0.001) | 98,7 | 91,4 |
| <i>sigmoid</i> | <i>tanh</i> | 40 | <i>sigmoid</i> | <i>relu</i> | 40 | Nadam(0.001) | 98,9 | 90,6 |

Tablica 16. Konačna optimizacija LSTM mreže za $n_steps=8$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizer | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>tanh</i> | <i>tanh</i> | 3 | <i>sigmoid</i> | <i>tanh</i> | 5 | Nadam(0.001) | 98,2 | 91,7 |
| <i>sigmoid</i> | <i>relu</i> | 25 | <i>sigmoid</i> | <i>relu</i> | 50 | Nadam(0.001) | 99,2 | 91,2 |
| <i>relu</i> | <i>sigmoid</i> | 15 | <i>relu</i> | <i>sigmoid</i> | 15 | Nadam(0.001) | 95,7 | 91,2 |

Tablica 17. Konačna optimizacija LSTM mreže za $n_steps=9$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizer | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>tanh</i> | 30 | <i>relu</i> | <i>sigmoid</i> | 50 | Nadam(0.001) | 98,1 | 92,4 |
| <i>sigmoid</i> | <i>tanh</i> | 30 | <i>tanh</i> | <i>tanh</i> | 30 | Nadam(0.001) | 99,3 | 90,7 |

Tablica 18. Konačna optimizacija za $n_steps=10$

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizer | $R_{trening}/\%$ | $R_{validacija}/\%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|--------------|------------------|---------------------|
| <i>sigmoid</i> | <i>tanh</i> | 25 | <i>tanh</i> | <i>sigmoid</i> | 40 | Nadam(0.001) | 98 | 92,9 |
| <i>sigmoid</i> | <i>relu</i> | 25 | <i>sigmoid</i> | <i>tanh</i> | 25 | Nadam(0.001) | 99,7 | 92,2 |
| <i>sigmoid</i> | <i>tanh</i> | 25 | <i>relu</i> | <i>sigmoid</i> | 30 | Nadam(0.001) | 98 | 91,9 |

Pokazalo se da nema smisla uzimati više od 3-4 koraka ulaza u prošlost jer rezultati postaju lošiji, a uvođenjem više od 10 koraka u prošlost točnost modela značajno pada.

Najboljim mrežama je dodatno mijenjan optimizator kako bi se pronašla najbolja konfiguracija. Optimizator Nadam se u pravilu pokazao najpovoljnijim, Adam i RMSprop daju nešto lošije, dok Adadelta daje uvjerljivo najlošije rezultate. Tako je dobivena najbolja mreža za hladnu struju (tablica 19.) dok je za toplu struju primijenjeno deset najboljih konfiguracija za hladne struje i tako dobivena najbolja neuronska mreža prikazana u tablici 20.

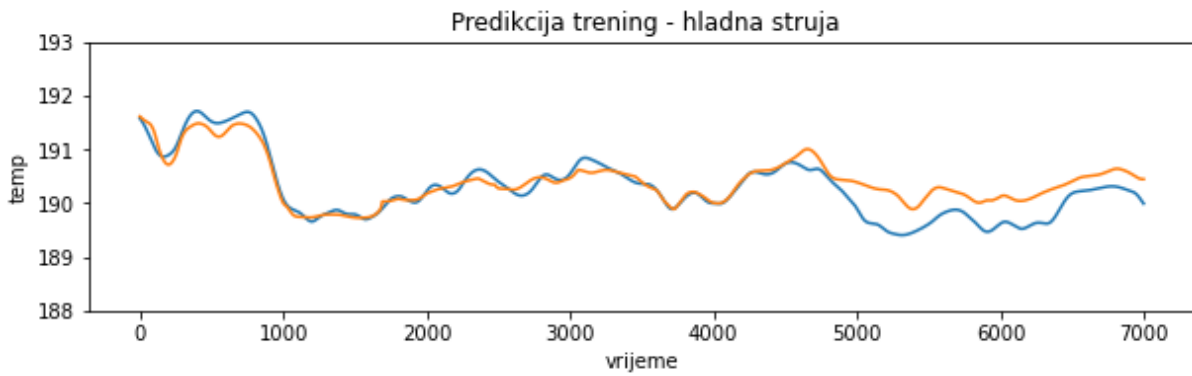
Tablica 19. Najbolja konfiguracija za hladnu struju

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{\text{trening}}/ \%$ | $R_{\text{validacija}} / \%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------|--------------------------|------------------------------|
| <i>sigmoid</i> | <i>relu</i> | 30 | / | / | / | RMSprop(0.001) | 98,3 | 93,6 |

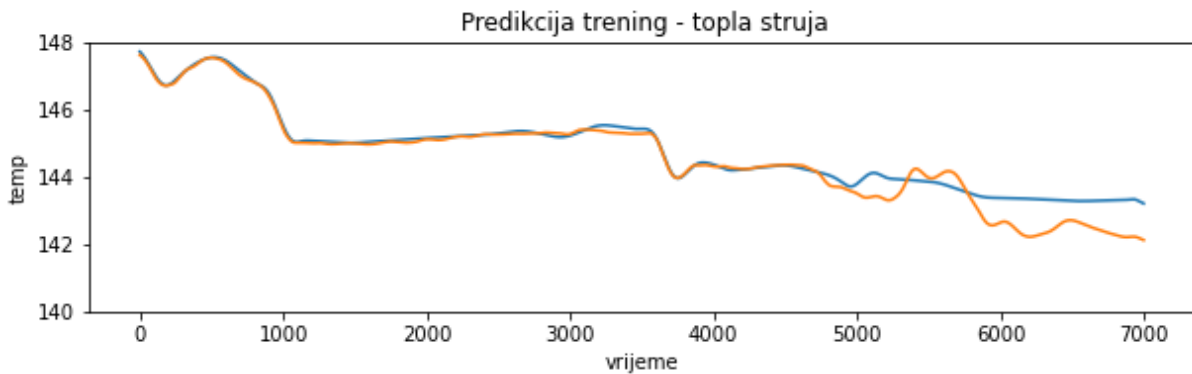
Tablica 20. Najbolja konfiguracija za toplu struju

| Aktiv. u 1. sloju | rec_aktiv u 1. sloju | Neuroni u 1. sloju | Aktiv. u 2. sloju | rec_aktiv u 2. sloju | Neuroni u 2. sloju | Optimizator | $R_{\text{trening}}/ \%$ | $R_{\text{validacija}} / \%$ |
|-------------------|----------------------|--------------------|-------------------|----------------------|--------------------|----------------|--------------------------|------------------------------|
| <i>sigmoid</i> | <i>relu</i> | 30 | <i>sigmoid</i> | <i>tanh</i> | 30 | RMSprop(0.001) | 99,8 | 87,4 |

Od kombinacija prijenosnih funkcija najbolje rezultate daju *sigmoid relu* i *tanh tanh* pri čemu se prva funkcija odnosi na stanje ćelije i skriveno stanje (*activation*), a druga funkcija se odnosi na propusnicu zaborava, ulaza i izlaza (*recurrent activation*).

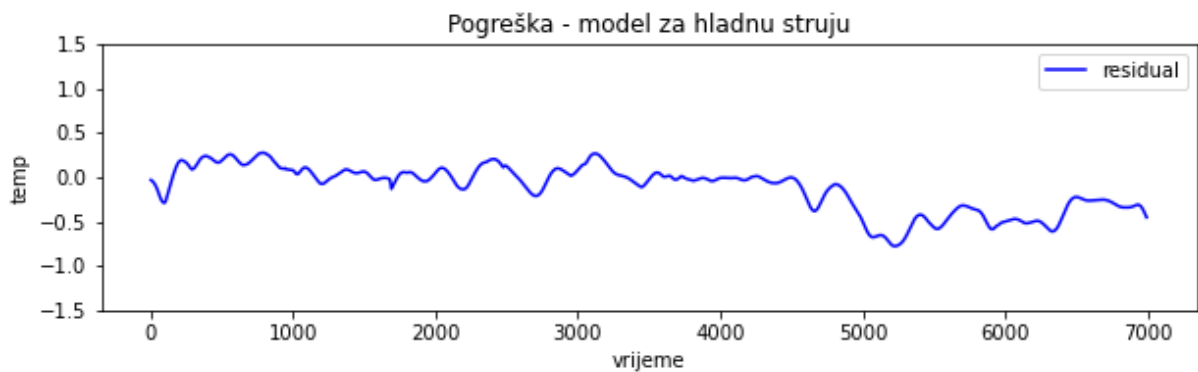


Slika 25. Usporedba temperatura dobivenih modelom i realnih temperatura na skupu za razvoj modela kod hladne struje

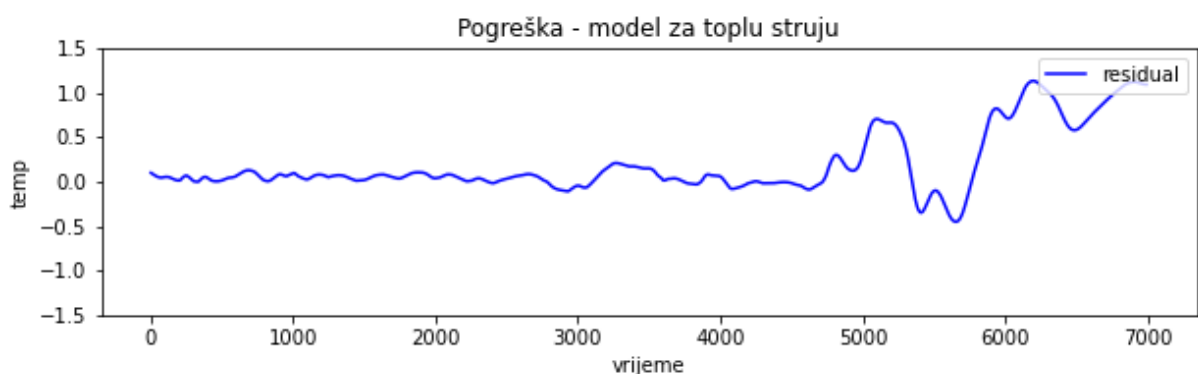


Slika 26. Usporedba temperatura dobivenih modelom i realnih temperatura na skupu za razvoj modela kod tople struje

Na slikama 25. i 26. prikazani su rezultati odabranih modela LSTM mreža na skupu za razvoj modela za hladnu te za toplu struju. Prvih 4550 podataka odnosi se na treniranje modela dok se preostalih 2550 odnosi na validaciju modela. Slaganje je sveukupno dobro iako kao što je očekivano temperature modela se nešto bolje slažu sa stvarnim podacima na skupu za treniranje nego sa onima koji su upotrjebljeni za validaciju modela.

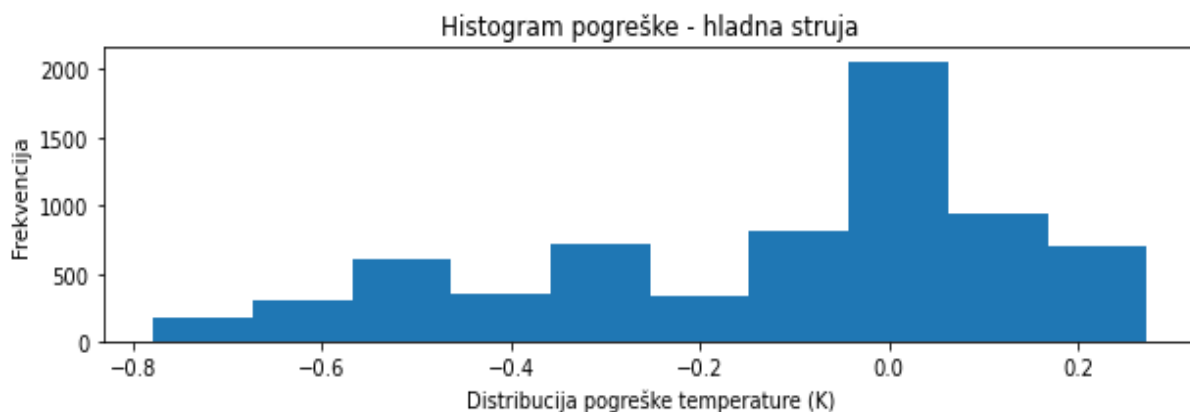


Slika 27. Pogreška na skupu za razvoj modela hladne struje

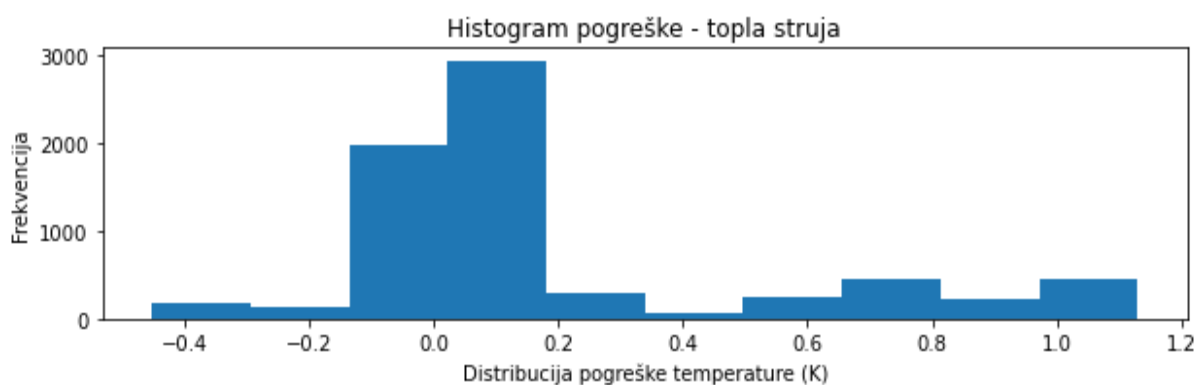


Slika 28. Pogreška na skupu za razvoj modela za toplu struju

Na slikama 27. i 28. prikazana je pogreška modela iz koje se vidi daje pogreška modela veća na skupu za validaciju nego od one na skupu za učenje. Pogreška na skupu za učenje oscilira oko nule u rasponu $\pm 0.3^{\circ}\text{C}$, dok na skupu za validaciju pokazuje negativan trend do -0.7°C (slika 27.). Kod modela za tople struju na skupu za učenje pogreška oscilira oko nule u rasponu $\pm 0.3^{\circ}\text{C}$ dok se na skupu za validaciju kreće u pozitivnom smjeru sa maksimalnim iznosom 1.2°C (slika 28.). Pretpostavka tokom izvođenja eksperimenta da će dobra konfiguracija LSTM neuronske mreže za hladnu struju odgovarati onoj za toplu struju se ipak pokazala pogrešna što je uzrok nešto manje točnosti modela za toplu struju.



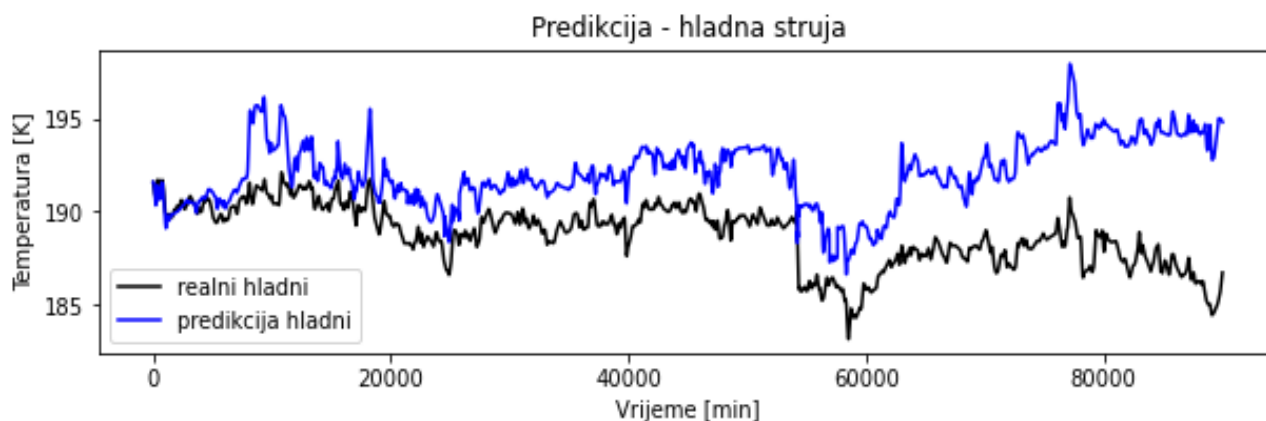
Slika 29. Histogram pogreške na skupu za razvoj modela za hladnu struju



Slika 30. Histogram pogreške na skupu za razvoj modela za toplu struju

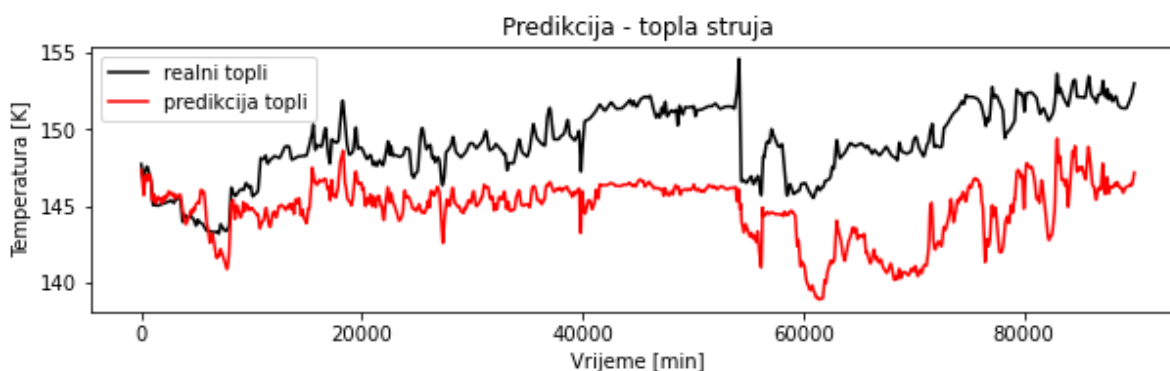
Slika 29. pokazuje histogram pogrešaka predviđenih izlaznih temperatura hladne struje na skupu za razvoj modela. Pogreške su blago raspodijeljene prema negativnim vrijednostima izlazne temperature, ali su vrlo malog iznosa što ukazuje na neznatno prljanje izmjenjivača tokom perioda 7000 podataka. Analogan zaključak se može izvesti za toplu struju čiji je histogram pogreške na skupu za razvoj modela prikazan na slici 30.

Navedeni model neuronske mreže je zatim primijenjen na novom skupu od 90 000 podataka što odgovara vremenskom periodu oko 3 mjeseca tokom kojeg je izmjenjivač topline radio između dva čišćenja. Na prvih 7000 podataka od čišćenja se pretpostavlja da nema nastajanja naslaga.



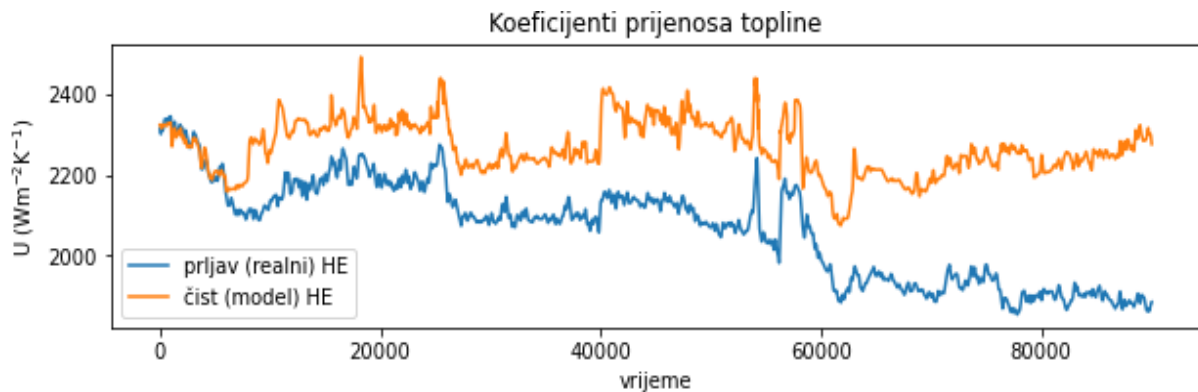
Slika 31. Usporedba temperatura na izlazu iz izmjenjivača dobivenih modelom i realnih temperatura za hladnu struju kroz period od 90 000 podataka

Na slici 31. prikazana je usporedba temperatura dobivenih modelom i realnih temperatura za hladnu struju kroz period 90 000 podataka. Vidljivo je dobro slaganje temperatura dobivenih modelom i realnih podataka na početku tj. za period 7000 podataka. Razlika između dviju krivulja u početku se pripisuje samo pogrešci modela, kako je pogreška tada gotovo zanemariva može se reći da model dobro oponaša rad čistog izmjenjivača. Kako vrijeme odmiče tako raste razlika u temperaturama te realni podaci temperature na izlazu iz izmjenjivača pokazuju nižu temperaturu od one predviđene modelom, iz razloga jer se hladna struja nije dovoljno zagrijala kao što se to očekuje zbog otpora prijenosu topline. Rastom naslaga na stijenci izmjenjivača raste otpor prijenosu topline što se na slici 31. očituje kao sve veća razlika u temperaturama.



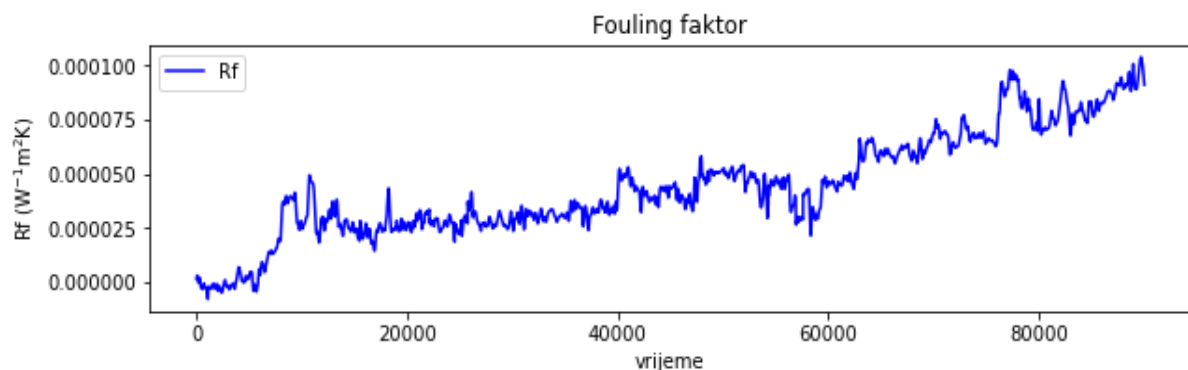
Slika 32. Usporedba temperatura na izlazu iz izmjenjivača dobivenih modelom i realnih temperatura za toplu struju kroz period od 90 000 podataka

Na slici 32. prikazana je usporedba temperatura na izlazu iz tople struje izmjenjivača dobivenih modelom i realnih temperatura za toplu struju kroz period od 90 000 podataka. Topla struja pokazuje višu temperaturu na izlazu iz izmjenjivača od očekivane, dakle nije uspjela prenijeti projektiranu toplinu zbog naslaga koje daju otpor prijenosu topline.



Slika 33. Koeficijenti prijenosa topline za prljavi (realni) i čisti (procjenjeni modelom) izmjenjivač topline

Na slici 33. prikazani su koeficijenti prijenosa topline dobiveni pomoću jednadžbi (9) i (10). Jasno je vidljiv trend opadanja koeficijenta prijenosa topline kod realnih podataka dok oni dobiveni modelom osciliraju, uslijed dinamičkih uvjeta procesa, unutar relativno stabilnog područja promjene. Sve veća razlika i rastući trend razlike između ta dva koeficijenta daje zaključiti da dolazi do smanjenja djelotvornosti izmjenjivača zbog otpora prijenosu topline, na što u konačnici ukazuje i sve veća vrijednost Fouling faktora prikazana na slici 34. Nakon čišćenja izmjenjivača, tj. u početku vrijednost Fouling faktora oscilira oko nule, a zatim, raste sve vrijeme sve do idućeg čišćenja izmjenjivača. Sporouzlazni rast dobivenog Fouling faktora ukazuje na polagano nastajanje naslaga u ispitivanom izmjenjivaču topline.



Slika 34. Porast fouling faktora kroz period 90 000 podataka

5. Zaključak

Cilj ovog rada bio je razviti model rekurentne LSTM neuronske mreže na primjeru industrijskog izmjenjivača topline radi kontinuiranog praćenja nastajanja naslaga na stijenkama izmjenjivača topline.

Pretpostavka ovoga rada bila je da rekurentne LSTM neuronske mreže zbog sposobnosti obrade sekvencijalnih podataka mogu poslužiti za razvoj modela dinamičkih procesa kao što je praćenje stvaranja naslaga u izmjenjivaču.

U programskom jeziku *Spyder* razvijen je model LSTM rekurentne neuronske mreže pri čemu se mijenjao broj koraka ulaznih varijabli u prošlost, dodavao novi skriveni sloj, mijenjao broj neurona u skrivenom sloju te mijenjao optimizator. Dobiveni rezultati su pokazali da nije potrebno uzimati ulazne podatke više od 3-4 koraka iz prošlosti jer rezultati postaju lošiji, a uvođenjem više od 10 koraka u prošlost točnost modela počinje značajno padati.

Na temelju provedenih simulacija te dobivenih statističkih i grafičkih rezultata može se zaključiti da samo podaci iz bliže prošlosti utječu na promjenu izlaznih varijabli modela, a njihov utjecaj slabi kako se udaljavaju u vremenu.

Na temelju dobivenog modela neuronske mreže za procjenu temperatura na izlazu iz izmjenjivača topline dobiveni su koeficijenti prijenosa topline sa i bez naslaga na osnovu kojih je izračunata vrijednost *fouling faktora*. Trend kontinuiranog rasta *fouling faktora* ukazuje na stvaranje naslaga u izmjenjivaču topline.

Primjenom razvijenih modela na postrojenju omogućila bi se prediktivna dijagnostika izmjenjivača topline tj. omogućilo bi se pravovremeno čišćenje i servisiranje izmjenjivača uslijed čega se može očekivati stabilniji rad cjelokupnog postrojenja, kao i značajne financijske uštede.

6. Popis oznaka

Q_h - toplina koju primi hladniji fluid, W

Q_t - toplina koju topliji fluid prenese, W

m_h - maseni protok hladne struje, kg s^{-1}

m_t - maseni protok tople struje, kg s^{-1}

$c_{p,t}$ - specifični toplinski kapacitet tople struje, $\text{J K}^{-1} \text{kg}^{-1}$

$c_{p,h}$ - specifični toplinski kapacitet hladne struje, $\text{J K}^{-1} \text{kg}^{-1}$

$T_{h,ul}$ - temperatura hladne struje na ulazu, K

$T_{h,izl}$ - temperatura hladne struje na izlazu, K

$T_{t,ul}$ - temperatura tople struje na ulazu, K

$T_{t,izl}$ - temperatura tople struje na izlazu, K

Q - izmijenjena toplina, W

K - koeficijent prijenosa topline, $\text{W m}^{-2} \text{K}^{-1}$

A - površina izmjene topline, m^2

ΔT_{lm} - logaritamska razlika temperature, K

α - koeficijent prijelaza temperature, $\text{W m}^{-1} \text{K}^{-1}$

λ - koeficijent toplinske vodljivosti, $\text{W m}^{-2} \text{K}^{-1}$

R_f - fouling faktor

$R_{konvekcijski}$ - faktor otpora prijenosu topline konvekcijom

$R_{kondukcijski}$ - faktor otpora prijenosu topline kondukcijom

$K_{sa\ naslagama}$ - ukupni koeficijent prijenosa topline zaprljanog izmjenjivača

$K_{bez\ naslaga}$ - ukupni koeficijent prijenosa topline čistog izmjenjivača

$T_{h,o,model}$ - izlazna temperatura hladne struje procijenjena modelom neuronske mreže

$T_{c,o,model}$ - izlazna temperatura tople struje procijenjena modelom neuronske mreže

F - korekcijski faktor

x - ulazna veličina u neuron i

w - težinski koeficijent

b - prag (engl. bias)

I - suma otežanih ulaza u neuron

$\sigma, S(x)$, sigmoid - sigmoidna funkcija

$th(x), tanh$ - funkcija tangens hiperbolni

η - koeficijent učenja (engl. learning rate)

y - izlaz neurona

ε – vrijednost pogreške

E – globalna pogreška mreže

d_k – vrijednost dobivena eksperimentom

x_k – vrijednost koju daje izlazni sloj mreže

y_t – izlaz koji mreža daje za trenutak t

x_t – ulaz u mrežu u trenutku t

y_t – izlaz koji mreža daje za neki trenutak t

h_{t-1} – interna memorija, skriveno stanje u prijašnjem trenutku

h_t – interna memorija, skriveno stanje u sadašnjem trenutku

f_w – funkcija koja povezuje internu memorije u sadašnjem trenutku sa memorijom u prošlom trenutku i sadašnjim ulazom u neuron

W_{xh}^T – matrica težinskih koeficijenata koja povezuje ulaz sa internom memorijom

W_{hh}^T – matrica težinskih koeficijenata koja povezuje prošlu i novu vrijednost interne memorije

W_{hy}^T – matrica težinskih koeficijenata koja povezuje internu memorije sa izlazom iz neurona

L – loss funkcija, pogreška

c_t – stanje ćelije u sadašnjem trenutku

c_{t-1} – stanje ćelije u prijašnjem trenutku

$relu$ – ReLU aktivacijska funkcija

MSE – srednje kvadratno odstupanje

$R_{trening}$ – srednje kvadratno odstupanje modela na skupu za razvoj

$R_{validacija}$ – srednje kvadratno odstupanje modela na skupu za validaciju

7. Popis literature:

- [1] R. Isermann, *Fault-Diagnosis Systems*, 1st ed., Germany: Springer, 2006.
- [2] <https://www.intechopen.com/chapters/75690> 31.08.2021. 20:00
- [3] Željka Ujević Andrijić, Nenad Bolf, Nikola Rimac, Adriana Brzović, *Fouling Detection in Industrial Heat Exchanger Using Neural Network Models*, Heat Transfer Engineering, 2021
- [4] Joel Levitt, *Complete Guide to Preventive and Predictive Maintenance*, Industrial Press Inc.
- [5] <https://www.youtube.com/watch?v=RmVWKLbLq2Y> 27.03.2021. 14:00
- [6] O. Gudmundsson, "Detection of fouling in heat exchangers using model comparison" Ph.D. dissertation, Faculty of Industrial Engineering, Mechanical Engineering and Computer Science University of Iceland, Reykjavik, Iceland, 2015
- [7] Aleksandra Sander, „Izmjenjivači topline (interno predavanje)“
- [8] Toni Palčić, Roko Perković, Filip Taradi, *Napredna analiza i dijagnostika rada procesa*, Kemijsko inženjerske vježbe, FKIT, 2017.
- [9] Emil Cerić, *Nafta procesi i proizvodi*, Biblioteka INA, Zagreb 2006.
- [10] Zvonimir Janović, *Naftni i petrokemijski procesi i proizvodi*, Drugo izdanje, Zagreb 2011.
- [11] Nenad Bolf, *Modeliranje procesa*, interno predavanje, FKIT
- [12] Željka Ujević-Andrijić, *Umjetne neuronske mreže*, Kemija u industriji 68 (5-6) [2019]
- [13] <https://historyofinformation.com/detail.php?entryid=782> 04.09.2021. 13:15
- [14] <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0> 04.09.2021. 15:25
- [15] <https://towardsdatascience.com/tagged/tanh?p=e11649256209> 04.09.2021 11:25
- [16] <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> 28.05.2021. 14:00
- [17] Adriana Brzović, *Process Monitoring and Predictive Diagnostics*, Final thesis, FKIT, Zagreb, September 2019
- [18] https://www.w3schools.com/python/numpy/numpy_intro.asp 28.05.2021. 14:30
- [19] Nenad Bolf, Ivica Jerbić, *Primjena umjetnih neuronskih mreža pri identificiranju i vođenju procesa*, Kemija u industriji 55 (11)
- [20] <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10> 27.08.2021. 18:15
- [21] <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> 20.08.2020.12:11
- [22] <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6> 01.09.2021. 9:00
- [23] https://keras.io/guides/sequential_model/ 20.08.2021. 13:20

Životopis

████████████████████ Pohađao sam Osnovnu školu Julija Kempfa u Požegi. Nakon osmogodišnjeg školovanja upisao sam Gimnaziju u Požegi prirodoslovno matematičkog usmjerenja 2011. godine. Srednju školu sam završio 2015. godine, a sljedeće godine sam upisao Fakultet kemijskog inženjerstva i tehnologije Sveučilišta u Zagrebu. Stručnu praksu odradio sam u HEP d.d. u Centralnom kemijsko – tehnološkom laboratoriju (CKTL). Završni rad sam pisao pod mentorstvom prof. dr. sc. Veljka Filipana na Zavodu za termodinamiku, strojarstvo i energetiku sa temom Radne tvari za organske Rankine procese, čime sam stekao akademsku titulu Sveučilišni prvostupnik inženjer kemijskog inženjerstva. Sudjelovao sam na XIII. Susretu mladih kemijskih inženjera sa posterom na temu „Istraživanje hidrodinamike u različitim izvedbama anularnog fotokatalitičkog reaktora“ pod mentorstvom izv.prof.dr.sc. Vanja Kosar. Diplomski rad iz područja umjetne inteligencije izradio sam pod mentorstvom doc. dr. sc. Željke Ujević Andrijić na Zavodu za mjerenja i automatsko vođenje procesa 2021. godine.