SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6544

PREDVIĐANJE VRIJEDNOSTI FINANCIJSKIH PODATAKA TEMELJENO NA REKURENTNIM NEURONSKIM MREŽAMA

Matej Luburić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6544

PREDVIĐANJE VRIJEDNOSTI FINANCIJSKIH PODATAKA TEMELJENO NA REKURENTNIM NEURONSKIM MREŽAMA

Matej Luburić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 13. ožujka 2020.

ZAVRŠNI ZADATAK br. 6544

Pristupnik:	**Matej Luburić (0036508139)**
Studij:	Računarstvo
Modul:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Predviđanje vrijednosti financijskih podataka temeljeno na rekurentnim neuronskim mrežama**

Opis zadatka:

Rekurentne neuronske mreže koriste se za uspješno rješavanje klasifikacijskih i regresijskih problema u području umjetne inteligencije koji koriste vremenske nizove mjerenja. U literaturi je navedeno više tipova arhitektura rekurentnih neuronskih mreža. Cilj ovog završnog rada je implementacija i primjena rekurentne neuronske mreže na problem predviđanja vrijednosti financijskih podataka. U radu je potrebno ostvariti implementaciju jednog algoritma za rekurentne neuronske mreže koji je najbolje primijenjiv na dani problem te ukratko opisati i značajke ostalih algoritama rekurentnih neuronskih mreža. Implementaciju algoritma potrebno je primijeniti i vrednovati na nekoliko skupova slobodno dostupnih financijskih podataka (npr. industrijski indeksi, cijene dionica).

Rok za predaju rada: 12. lipnja 2020.

Bogu hvala!

Al‘ onima što se u Jahvu uzdaju snaga se obnavlja, krila im rastu kao orlovima, trče i ne sustaju, hode i ne more se. Iz 40, 31!

Veliko hvala mentoru dr. sc. Alanu Jovići na strpljenju i svim savjetima za vrijeme izrade ovog rada!

Veliko hvala mojim roditeljima, braći i sestrama na velikoj podršci za vrijeme cijelog preddiplomskog studija!

Veliko hvala svim kolegama koji su mi na bilo koji način pomogli za vrijeme studiranja!

SADRŽAJ

POPIS SLIKA

POPIS TABLICA

1. Uvod

Predviđanje vrijednosti financijskih podataka izazovno je područje za istraživače i ljude iz poslovnog svijeta. Predviđanje može biti zasnovano na raznim temeljima, kao npr. intuiciji ili pak na egzaktnim činjenicama i pokazateljima. Ekonomija je dosta temeljena na nelinearnim odnosima parametara koji utječu na cijenu imovine te postoji velika nepredvidivost kretanja cijene.

Istraživači koji se bave strojnim učenjem nastoje automatizirati predviđanje vrijednosti financijskih podataka na temelju modela koji se uče na povijesnim podacima kako bi iz njih naučili neka generalna svojstva. Dakle, cilj je napraviti model koji uči generalne obrasce iz povijesnih podataka kako bi model bio primjenjiv na više podatkovnih skupova, tj. kako bi se izbjegla prevelika prilagođenost podacima na kojima model uči (engl. *overfitting*). Automatizacijom ovog postupka uvelike bi se olakšao posao poslovnim ljudima koji moraju uspoređivati mnoštvo tehničkih parametara koji utječu na cijenu imovine.

U strojnom učenju postoje razni pristupi ovom kompleksnom problemu. Osnovni pristupi su:

1. **autoregresija** – predviđanje buduće cijene isključivo na osnovu povijesnih cijena

2. **multivarijabilna regresija** – predviđanje buduće cijene na osnovu kombinacije povijesne cijene i drugih izračunatih parametara. Neki od parametara mogu biti informacije iz novinskih članaka [5] ili pak numerički kao što su kretanje srednje vrijednosti (engl. *moving average*), Bollingerova traka (engl. *Bollinger band*), itd.

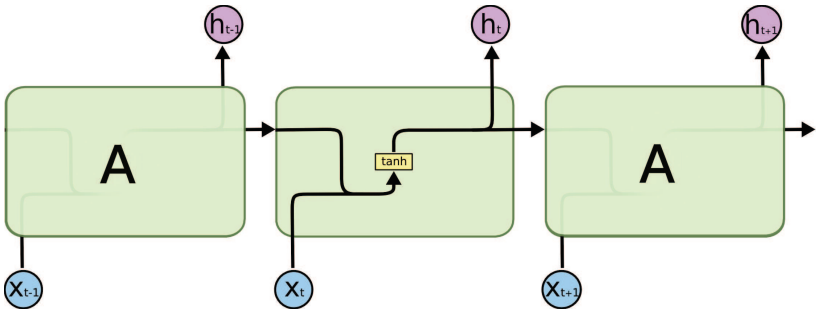
Mnogi modeli kombiniraju ova dva pristupa kako bi povećali uspješnost predikcije. Primjer je model ARIMA (engl. *autoregressive integrated moving average*) koji kombinira autoregresiju i varijablu kretanja srednje vrijednosti. Zbog postojanja šuma u financijskim podacima i kako bi se u obzir uzele samo kvalitetne informacije iz podataka, u novije vrijeme nastaju sve kompleksniji modeli koji kombiniraju neku tehniku

predobrade podataka za uklanjanje šuma, te jednog ili više jednostavnih modela za učenje. Jedan primjer je model ICA-CCA-SVR, koji kombinira neovisnu komponentnu analizu (engl. *independent component analysis*) kao predobradu podataka, kanonsku korelacijsku analizu (engl. *canonical correlation analysis*) i stroj potpornih vektora za regresiju kao osnovne modele [8].
Ovaj rad se temelji na predviđanju financijskih podataka temeljem niza ćelija s dugom kratkoročnom memorijom (engl. *long short-term memory, LSTM*). Ćelije LSTM su podtip rekurentnih neuronskih mreža (engl. *recurrent neural network, RNN*).

Ovaj rad je organiziran na sljedeći način: u poglavlju 2 je prikaz tri vrste arhitektura rekurentnih neuronskih mreža, a to su: osnovna ćelija RNN, ćelija LSTM i propusna povratna ćelija (engl. *gated recurrent unit, GRU*). U poglavlju 3 nalazi se formalni opis modela, u poglavlju 4 opis programske implementacije i prikaz rezultata, te na kraju zaključak.

2. Vrste rekurentnih neuronskih mreža

Jednostavan prikaz jedne ćelije rekurentne neuronske mreže prikazan je na slici (2.1). Velika prednost rekurentnih u odnosu na unaprijedne neuronske mreže je modeliranje slijedova različite duljine, dok unaprijedne mogu raditi samo s vektorima fiksne duljine i zato se rekurentne neuronske mreže dosta primjenjuju u problemima raspoznavanja govora, rukom pisanog teksta, semantike slika, prevođenju teksta i dr.



Slika 2.1: Prikaz razmotane rekurentne neuronske mreže [10]

Rekurentne neuronske mreže [10] su vrsta umjetnih neuronskih mreža koje imaju sposobnost pamćenja više prethodnih stanja jer je trenutno skriveno stanje h_t funkcija trenutnog ulaza i prethodnog skrivenog stanja (2.1), dok je prethodno stanje ovisi o stanju prije njega i tako proizvoljan broj puta unatrag. Iz formule (2.2) je vidljivo da trenutno skriveno stanje h_t ovisi o proizvoljno velikom vremenskom nizu veličine $t-1$ koje je pohranjeno u h_{t-1} i trenutnom ulazu x_t.

$$h_t = f(Ux_t + Wh_{t-1}) = f(a_t) \tag{2.1}$$

$$\begin{aligned} h_t &= z(x_t, x_{t-1}, ..., x_2, x_1) \\ &= f(h_{t-1}, x_t) \end{aligned} \tag{2.2}$$

Funkcija f u navedenim formulama je nelinearna funkcija, a najčešće su to sigmoida

ili tangens hiperbolni. Parametri U, W su težinske matrice koje se ažuriraju učenjem mreže.

2.1. Problem nestajućeg i eksplodirajućeg gradijenta

Problem nestajućeg i eksplodirajućeg gradijenta nastaje zbog opetovanog množenja matricom W iz formule (2.1) kod propagacije pogreške unatrag.

$$\frac{\partial h_t}{\partial h_{t-1}} = W^T diag(\frac{\partial h_t}{\partial a_t}) \tag{2.3}$$

Iz formule (2.3) $diag$ pretvara vektor u dijagonalnu matricu. Zbog $||AB|| \leq ||A||||B||$ slijedi:

$$\frac{\partial h_t}{\partial h_{t-1}} \leq ||W^T||||diag(\frac{\partial h_t}{\partial a_t})|| \leq \gamma\lambda_1 \tag{2.4}$$

gdje je λ_1 najveća svojstvena vrijednost matrice W, a γ gornja granica od $diag(\frac{\partial tanh(a_t)}{\partial a_t})$.

Generalizacijom formule (2.4) na više vremenskih koraka preko pravila ulančavanja dobiva se formula:

$$\frac{\partial h_T}{\partial h_{t_0}} \leq (\gamma\lambda_1)^{T-t_0} \tag{2.5}$$

Za broj koraka $T-t_0$ mnogo veći od 0, gradijent može biti eksplodirajući, nestajući ili stabilan:

$$(\gamma\lambda_1)^{T-t_0} \begin{cases} \infty & ako \quad je \quad \gamma\lambda_1 > 1 \quad (eksplodirajuci) \\ 0 & ako \quad je \quad \gamma\lambda_1 < 1 \quad (nestajuci) \\ 1 & ako \quad je \quad \gamma\lambda_1 \approx 1 \end{cases}$$

Kako bi gradijent bio stabilan matrica W mora zadovoljavati stroge uvjete.

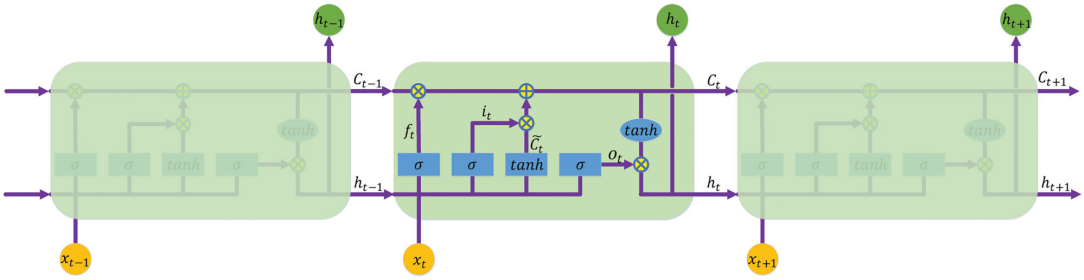
Nestajući gradijent se manifestira tako da osnovni RNN ima kratko pamćenje. Što niz podataka koji dolazi na ulaz postaje veći, mreža će svojstva naučena na početku više zaboravljati. Razlog zaboravljanja je nestajući gradijent jer velikim povećanjem duljine sekvence, povećava se i broj slojeva odmotane rekurentne mreže. Tako propagacijom pogreške unatrag dolazi do jako male ili nikakve promjene težine prvih slojeva, tako ti slojevi postaju beznačajni za učenje.

Navedeni problem rješavaju ćelije LSTM i GRU.

2.2. Ćelija sa dugom kratkoročnom memorijom

Ćelija sa dugom kratkoročnom memorijom (LSTM) je među najkorištenijim ćelijama rekurentnih neuronskih mreža. Naziva se duga memorija jer može pamtiti ovisnosti iz dalje prošlosti pomoću stanja ćelije C_t, dok kratkoročna memorija se odnosi na trenutni ulaz x_t i skriveno stanje h_{t-1} koji u manjoj mjeri mijenjaju stanje ćelije C_t. (Slika 2.2)

U kojoj mjeri će prethodno stanje C_{t-1}, prethodno skriveno stanje h_{t-1} i trenutni ulaz



Slika 2.2: Prikaz ćelije s dugom kratkoročnom memorijom [2]

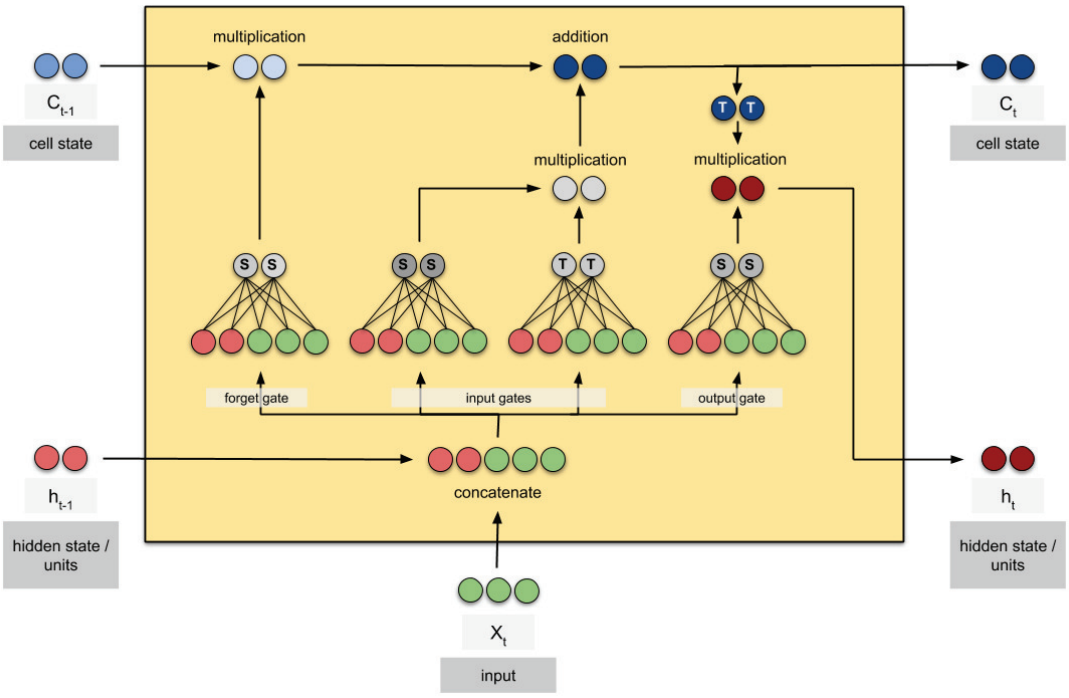
x_t utjecati na novo stanje ćelije C_t ovisi o dvije propusnice:

- propusnica zaboravljanja (engl. *forget gate*),
- propusnica novog ulaza (engl. *input gate*),
- izlazna propusnica (engl. *output gate*).

Propusnica zaboravljanja računa koliki dio informacije će se "zaboraviti" iz prethodnog stanja. Iz slike (2.3) se vidi da je propusnica zaboravljanja zapravo neuronska mreža. Prvo dolazi do spajanja vektora x_t i h_{t-1} koji čine ulazni sloj neuronske mreže, zatim se svaka vrijednost vektora množi s odgovarajućom težinom. Izlazni sloj ove neuronske mreže ima onoliko čvorova kolika je dimenzionalnost vektorâ x_t, h_t i C_t. U izlaznom sloju primjenjuje se sigmoidna funkcija koja je ključna za "zaboravljanje" jer su njezine vrijednosti isključivo nenegativne iz intervala [0, 1]. Ako kao vrijednost f_t dobijemo 0, to znači da potpuno zaboravljamo prethodno stanje jer se prethodno stanje ćelije C_{t-1} množi s vektorom koji ima sve vrijednosti 0. Dobijanjem vrijednosti 1 za f_t znači da cijela informacija očuvana iz prethodnog stanja ćelije, a koliko će se odraziti na novo stanje ćelije C_t ovisi o propusnici novog ulaza.

$$f_t = \sigma(U_f x_t + W_f h_{t-1} + b_f) = \sigma(a_{ft}) \tag{2.6}$$

Propusnica novog ulaza određuje u kojoj će mjeri trenutni ulaz zajedno sa prethodnim skrivenim stanjem utjecati na novo stanje ćelije. Ulaz je jednak u obje neuronske



Slika 2.3: Detaljan prikaz ćelije s dugom kratkoročnom memorijom [9]

mreže u propusnici novog sloja. Razlika između ove dvije mreže je što je aktivacijska funkcija za lijevu neuronsku mrežu sigmoida (u *pytorchu* i *tensorflowu* je tangens hiperbolni), a za desnu tangens hiperbolni i što imaju različite vrijednosti vektora težina. Vektor i_t određuje koliki dio informacije $\tilde{C}$ će utjecati na novo stanje ćelije C_t i to se postiže Hadamardovim umnoškom ova dva vektora.

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i) = \sigma(a_{it}) \tag{2.7}$$

$$\tilde{C}_t = tanh(U_c x_t + W_c h_{t-1} + b_c) \tag{2.8}$$

Ukupni utjecaj navedene dvije propusnice čini novo stanje ćelije C_t i računamo ga po formuli (2.9) gdje operator $*$ predstavlja Hadamardov umnožak – umnožak vektora po komponentama.

$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1} \tag{2.9}$$

Neuronska mreža izlazne propusnice ima sigmoidnu aktivacijsku funkciju i određuje koliki će dio informacije novog stanja C_t prenijeti kao novo skriveno stanje h_t na sljedeći vremenski trenutak ćelije.

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + b_o) = \sigma(a_{ot}) \tag{2.10}$$

Konačno, skriveno stanje ćelije h_t je Hadamardov umnožak izlazne propusnice i stanja C_t koji je prije množenja prošao kroz neuronski sloj s aktivacijskom funkcijom tangens hiperbolni.

$$h_t = o_t * tanh(C_t) \tag{2.11}$$

Uvođenjem ćelije LSTM kod računanja novog stanja ćelije iz formule (2.10) uklonjeno je opetovano množenje sa matricom W_o koje je postojalo kod osnovne RNN [11].

$$\frac{\partial C_t}{\partial C_{t-1}} = f_t = \sigma(a_{ft}) \in [0, 1] \tag{2.12}$$

Slijedi, preko pravila ulančavanja

$$\frac{\partial C_T}{\partial C_{t_0}} = \prod_{t=t_0}^{T} f_t \leq 1 \tag{2.13}$$

Iz formule (2.13) je vidljivo da nije moguća pojava eksplodirajućeg gradijenta. Ipak problem eksplodirajućeg gradijenta nije potpuno eliminiran kod ćelije LSTM, ali se u praksi rijetko događa [11]. Eksplozija je moguća kod propagacije pogreške unatrag kod skrivenog stanja h_t zbog opetovanog množenja matricom W_o.

$$\frac{\partial h_t}{\partial h_{t-1}} = \frac{\partial h_t}{\partial a_{ot}} \frac{\partial a_{ot}}{\partial h_{t-1}} = W_o^T \frac{\partial h_t}{\partial a_{ot}} = ... \tag{2.14}$$

2.3. Propusna povratna ćelija

Propusnu povratnu ćeliju (GRU) uveo je Cho i suradnici tek 2014. godine. Ćelija GRU je izvedena iz ćelije LSTM. Ćelija GRU nema stanje ćelije C_t već cijelu informaciju prenosi preko skrivenog stanja h_t (Slika 2.4). Sastoji se od dvije propusnice [6]:

- propusnica ažuriranja (engl. *update gate*),
- propusnica resetiranja (engl. *reset gate*).

Propusnicu ažuriranja se računa po formuli (2.15). Prvo dolazi do spajanja prošlog skrivenog stanja h_{t-1} i novog ulaza x_t. Nakon množenja sa matricama težina U_z i W_z primjenjuje se sigmoidna funkcija.

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z) \tag{2.15}$$

Propusnica resetiranja određuje koliki dio informacije skrivenog stanja će se zaboraviti. Računa se po formuli (2.16).

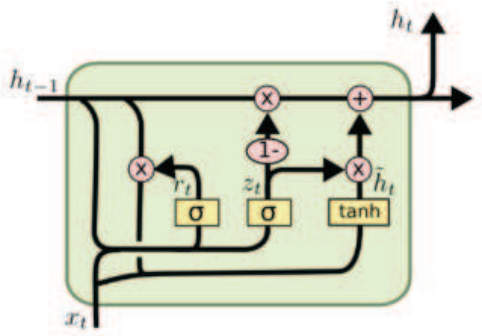
$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r) \tag{2.16}$$

U formulama (2.17) i (2.18) operator $*$ označava Hadamardov umnožak. Vrijednost $\tilde{h_t}$ koja je prijedlog budućeg skrivenog stanja. Iz formule (2.17) je vidljivo da prije primjene aktivacijske funkcije tangensa hiperbolnog prethodno skriveno stanje se filtrira kroz propusnicu ažuriranja.

$$\tilde{h}_t = \sigma(U_h x_t + W_h(r_t * h_{t-1}) + b_h) \tag{2.17}$$

Konačno, novo skriveno stanje se računa prema formuli (2.18). Izlaz propusnice ažuriranja z_t dosta određuje buduću vrijednost skrivenog stanja. Vrijednost z_t je iz intervala [0, 1] i ako je njena vrijednost bliže 0 to veći dio informacije prethodnog stanja postaje novo skriveno stanje, dok se toliko zanemaruje prijedlog skrivenog stanja $\tilde{h}_t$. Vrijedi i obratno.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{2.18}$$



Slika 2.4: Općeniti prikaz propusne povratne ćelije (GRU) [10]

Ćelija GRU je jednostavnija od ćelije LSTM jer nema stanje ćelije C_t i zbog toga ima manje parametara za ažurirati pri učenju. U radu [4] provedena je usporedba ova

tri tipa RNN mreža na više podatkovnih skupova. Zaključeno je da ćelije LSTM i GRU su bolje od obične RNN, a da razlika između LSTM i GRU nije statistički signifikantna.

Postoji još vrsta RNN koje imaju samo male preinake u odnosu na ćeliju LSTM, kao npr. LSTM sa špijunkama (engl. *peepholes*), multiplikativna LSTM, LSTM s pažnjom i dr.

3. Model za predviđanje financijskih podataka

U ovom poglavlju detaljnije se razrađuje model na osnovu kojeg je izrađena implementacija. Model koji je korišten jednak je dijelu modela LSTM iz modela WSAE-LSTM iz članka [2].

3.1. Podatkovni skup

Podatkovni skup jednak je skupu koji je korišten za navedeni model WSAE-LSTM. Sastoji se od podataka o šest različitih dioničkih indeksa, a to su: S&P 500, koji predstavlja 500 najvećih američkih kompanija i zajedno sa DJIA indeksom predstavljaju dobar indikator stanja američkog gospodarstva, Nikkei 225 koji predstavlja dobar indikator stanja japanskog gospodarstva, CSI 300 i HangSeng indeksi dobar indikator stanja kineskog gospodarstva i Nifty 50 dobar indikator stanja indijskog gospodarstva. Podatkovni skupovi imaju atribut datum i ostale atribute koji su vidljivi na slici (3.1). Podijeljeni su u tri kategorije podataka.

Prva kategorija poodataka odnosi se na dnevne podatke, a to su cijena u trenutku otvaranja (engl. *open price*), zatim cijena u trenutku zatvaranja (engl. *close price*), najviša (engl. *high price*) odnosno najniža cijena u danu (engl. *low price*) i dnevni volumen razmjena izražen u broju razmjena.
Druga kategorija odnosi se na pokazatelje tehničke analize. MACD (engl. *moving average convergence divergence*) se odnosi na kretanje srednje vrijednosti i predstavlja indikator jačine trenda, CCI (engl. *commodity channel index*) pomaže pri otkrivaju promjene trenda, BOLL (engl. *Bollinger band*) predstavlja volatilnost, tj. standardnu devijaciju (nekad se uzima jedna, a nekad dvije) oko srednje vrijednosti, EMA 20 (engl. *exponential moving average*) prikazuje kretanje srednje vrijednosti s naglaskom na novije podatke, MA5/MA10 (engl. *moving average*) je srednja vrijednost zadnjih 5

odnosno 10 vrijednosti zadnje cijene. MTM6/MTM12 se odnosi na akceleraciju promjene cijene, ROC je postotna promjena cijene u odnosu na cijenu iz prošlosti, SMI pokazuje relativni odnos zadnje cijene sa medianom, a WVAD stavlja u odnos volumen razmjena i cijenu dionica.

Treća kategorija su makroekonomski podaci koji također imaju utjecaj na cijenu dionica. Uzeti su indeks jačine američkog dolara i kamatna stopa države iz koje dolazi indeks.

Name	Definition/Implication
Panel A. Daily Trading Data	
Open/Close Price	nominal daily open/close price
High/Low Price	nominal daily highest/lowest price
Trading volume	Daily trading volume
Panel B. Technical Indicator	
MACD	Moving average convergence divergence: displays trend following characteristics and momentum characteristics.
CCI	Commodity channel index: helps to find the start and the end of a trend.
ATR	Average true range: measures the volatility of price.
BOLL	Bollinger Band: provides a relative definition of high and low, which aids in rigorous pattern recognition
EMA20	20 day Exponential Moving Average
MA5/MA10	5/10 day Moving Average
MTM6/MTM12	6/12 month Momentum: helps pinpoint the end of a decline or advance
ROC	Price rate of change: shows the speed at which a stock's price is changing
SMI	Stochastic Momentum Index: shows where the close price is relative to the midpoint of the same range.
WVAD	Williams's Variable Accumulation/Distribution: measures the buying and selling pressure.
Panel C. Macroeconomic Variable	
Exchange rate	US dollar Index
Interest rate	Interbank Offered Rate

Slika 3.1: Prikaz atributa podatkovnog skupa, preuzeto iz [2]

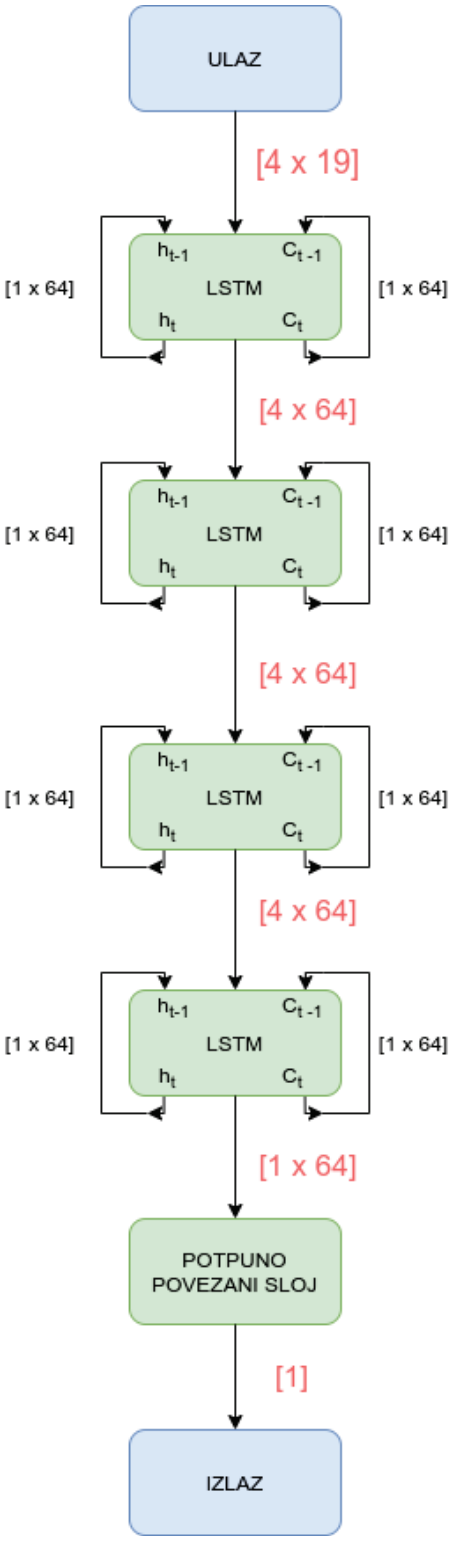
3.2. Priprema podataka

Prije ulaza podataka u model iz skupa atributa je izbačena oznaka datuma kako bi se poništio utjecaj nezavisne varijable vremena, a kao ciljna varijabla je postavljena cijena u trenutku zatvaranja. Cijeli podatkovni skup podijeljen je na manje podatkovne skupove od 30 mjeseci podataka, a svaki od njih je dalje bio podjeljen na skup za učenje (engl. *training set*), koji je činio 80% podskupa, odnosno 24 mjeseca, skup za validaciju (engl. *validation set*) i skup za testiranje (engl. *test set*) koji su činili po 10% svaki od njih, odnosno po 3 mjeseca svaki. Nakon učenja i testiranja na jednom manjem podatkovnom skupu, model bi se ponovno učio i testirao na novom manjem

podatkovnom skupu čiji je početak skupa za učenje pomaknut za 3 mjeseca udesno od početka skupa za učenje od prethodnog podatkovnog skupa. Tim se postiže ažuriranje težina u modelu i stavlja se naglasak na novije podatke.
Nakon podjele podatkovnog skupa, svi atributi osim ciljne varijable su normirani kako bi se ubrzao gradijentni spust.
Zatim, podatkovni skup je preoblikovan tako da je jedna sekvenca u modelu imala oblik $[x_{t-3}, x_{t-2}, x_{t-1}, x_t]$ -> y_{t+1}, što znači da se koristila prognoza jedan korak unaprijed (engl. *one-step-ahead*) na temelju vrijednosti atributa u posljednja četiri dana. Broj prošlih dana predlaže [2] i uzet je tehnikom pokušaja i pogreške.

3.3. Arhitektura modela implementacije

Na slici (3.2) je detaljnije objašnjenje modela izloženog u [2], a ovdje implementiranog. Model se sastoji od četiri slijedne ćelije LSTM, kod kojih je izlaz jedne ćelije ulaz u slijedeću ćeliju. Na kraju je potpuno povezani sloj koji preslikava zadnje skriveno stanje zadnje ćelije LSTM u cijenu.
Jednodimenzionalni vektor skrivenog stanja i stanja ćelije svih LSTM–ova je veličine 64, što znači da može toliku količinu obrazaca pamtiti. U početku učenja sve vrijednosti ovih vektora su postavljene na 0.
Na ulaz modela dolazi sekvenca 4 vektora koji prestavljaju zadnja 4 dana vrijednosti svih atributa. U model ulazi primjerak po primjerak iz sekvence koji predstavljaju x_t vektor. Daljni tijek učenja odvija se po postupku objašnjenom u poglavlju (2.2). Važno je naglasiti da je dimenzija izlaza prve tri ćelije LSTM 4 puta 64, jer se šalje sekvenca od zadnja 4 skrivena stanja, a svaki od njih je vektor duljine 64. Izlaz zadnje ćelije LSTM je jednodimenzionalni vektor veličine 64, jer ona na ulaz potpuno povezanog sloja šalje zadnje skriveno stanje ćelije. Aktivacijska funkcija potpuno povezanog sloja je linearna $y(x) = x$, koja na izlaz šalje predviđenu cijenu. Ovakav model ima oko 121 tisuću parametara za učenje i oni predstavljaju težine u neuronskim mrežama.



Slika 3.2: Prikaz implementiranog modela

4. Eksperimenti

U ovom poglavlju izložen je praktični dio rada. Prikazani su korišteni alati pri programskoj izvedbi te tablično i grafički prikazani dobiveni rezultati.

4.1. Korišteni alati i programska izvedba

Programska izvedba napisana je u programskom jeziku Python uz korištenje više biblioteka i modula.

Najkorištenija biblioteka u projektu je biblioteka *TensorFlow* koja se oslanja na biblioteku *Keras*. *TensorFlow* je Pythonova biblioteka otvorenog kôda (engl. *open source library*) razvijena od Googlea. Iz nje je korišten razred LSTM za konstrukciju modela, optimizacijska funkcija Adam i *tf.data.Dataset* aplikacijsko programsko sučelje.
Za razne funkcije sa podatkovnim okvirima korištena je biblioteka *pandas*.
Za iscrtavanje grafova korišten je modul *pyplot* iz biblioteke *matplotlib*. Modul *StandardScaler* iz biblioteke *sklearn* korišten je za normiranje podatkovnog skupa.

Aplikacijsko programsko sučelje *tf.data.Dataset* omogućuje stvaranje efektivnih cjevovoda podataka. Ono omogućuje stvaranje podatkovnog skupa od ulaznih podataka, transformacije nad podatkovnim skupom i iteriranje nad skupom.
Prva linija iz isječka kôda (4.1) prikazuje spajanje vektora značajki x_train sa vektorom ciljnih vrijednosti y_train. Druga linija prikazuje mješanje (engl. *shuffle*) sekvenci nakon kojeg se formiraju grupe (engl. *batch*). Metoda $repeat()$ ponavlja formiranje grupa beskonačno, pa je u metodi fit potrebno postaviti varijablu $steps\_per\_epoch$ na broj grupa po epohi koliki je moguće napraviti iz podatkovnog skupa. Miješanje sekvenci je napravljeno zbog toga da svaka epoha uči na novim skupovima grupa kako bi se izbjeglo da neuronska mreža zaglavi u lokalnim minimumima. Miješanje sekvenci je moguće zbog prethodnog preoblikovanja podataka na oblik $[x_{t-3}, x_{t-2}, x_{t-1}, x_t]$

-> y_{t+1} [1].

Isječak kôda 4.1: Prikaz primjene *tf.data.Dataset* u programskoj izvedbi

```
train_examples = tf.data.Dataset
        .from_tensor_slices((x_train, y_train))
train_examples = train_examples.cache().shuffle(1000)
        .batch(size_of_batch).repeat()

        ...

model.fit(train_set,
        epochs=500,
        validation_data=val_set,
        steps_per_epoch=(len(x_train) // size_of_batch),
        validation_steps=1,
        verbose=0)
```

Prema preporukama iz [2] korištena je veličina grupa od 60 primjeraka za skupove za učenje i validaciju.
Korišteno je srednje kvadratno odstupanje (engl. *mean square error, MSE*) kao funkcija gubitka jer su nam stršeće vrijednosti (engl. *outlier*) bitne kod promatranja financijskih podataka (formula 4.1). MSE daje velik značaj stršećim vrijednostima kod propagacije pogreške unatrag [7].

$$MSE = \frac{\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}{N} \tag{4.1}$$

Kao optimizacijska funkcija gradijentnog spusta korištena je Adam funkcija (engl. *Adaptive Moments*). Broj epoha izosio je 500, dok faktor učenja (engl. *learning rate*) 0.05.

4.2. Prikaz dobivenih rezultata

Odabrane metrike za interpretaciju dobivenih rješenja su: srednja apsolutna postotna pogreška (engl. *mean absolute percentage error, MAPE*), Pearsonov koeficijent korelacije (engl. *pearson's correlation coefficient*) (oznaka R), Theilov koeficijent U.

Srednja apsolutna postotna pogreška (MAPE) računa se po formuli (4.2).

$$MAPE = \frac{\sum_{t=1}^{N} |\frac{(y_t - \hat{y}_t)}{y_t}|}{N} \tag{4.2}$$

Pearsonov koeficijent korelacije [12] računa se prema formuli (4.3) i označava linearnu vezu između dvije varijable. Moguće vrijednosti koeficijenta su iz intervala [-1, 1]. Vrijednost 0 označava da ne postoji linearna veza između varijabli. Vrijednost 1 označava idealnu povezanost dvije varijable. Ako je R negativan, onda se radi o antikorelaciji dvaju nizova.

$$R = \frac{\sum_{t=1}^{N}(y_t - \bar{y}_t)(\hat{y}_t - \bar{\hat{y}}_t)}{\sqrt{\sum_{t=1}^{N}(y_t - \bar{y}_t)^2(\hat{y}_t - \bar{\hat{y}}_t)^2}} \tag{4.3}$$

Theilov koeficijent U iz formule (4.4) predstavlja točnost predikcije i može biti iz intervala [0, 1]. Vrijednost koeficijenta 0 označava točnu predikciju za sve primjere, dok vrijednost 1 označava da postoji obrnuta proporcionalnost ili jedna varijabla jednaka 0 za sve primjere [3].

$$Theilov \quad U \quad koeficijent = \frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t)^2} + \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t)^2}} \tag{4.4}$$

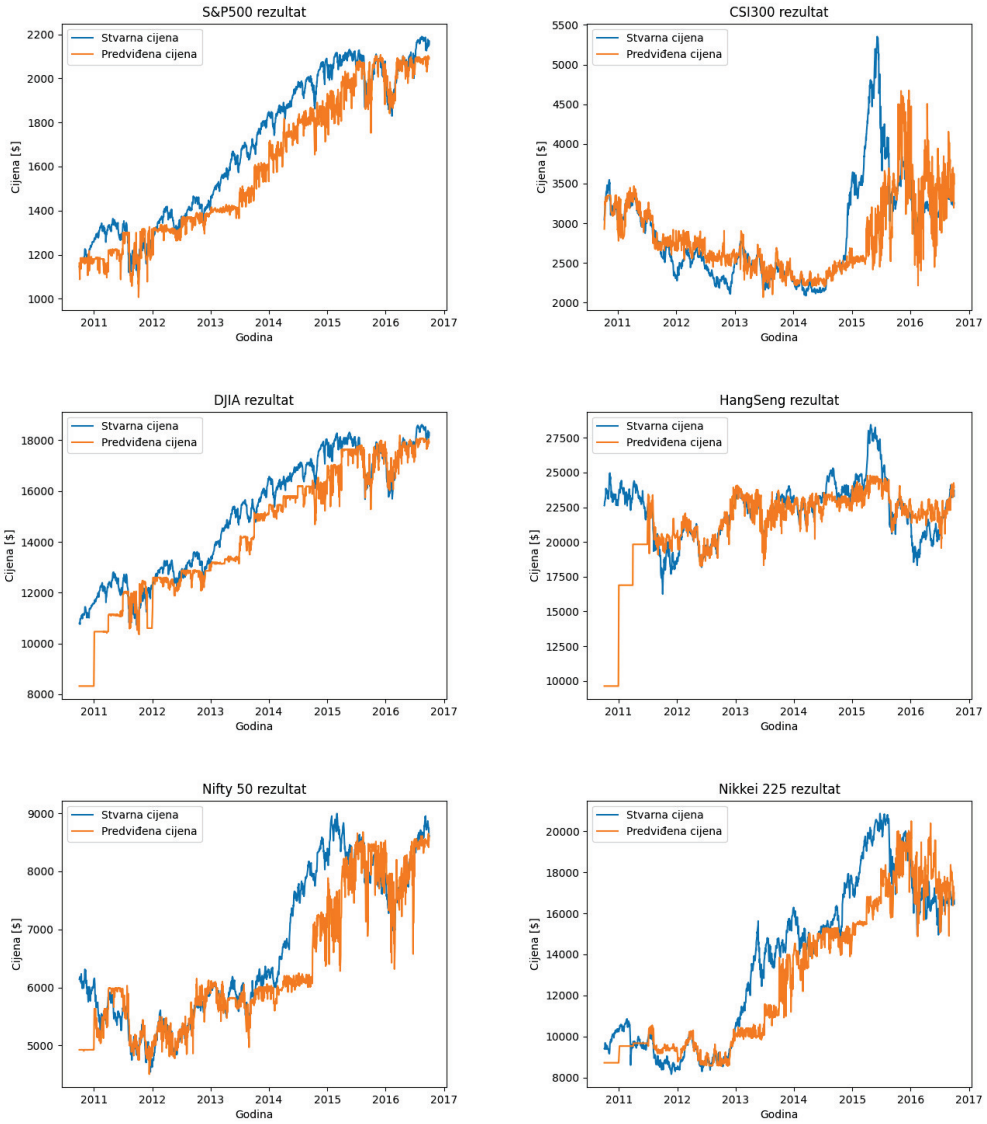
Dakle, predikcija je bolja što su njezine vrijednosti MAPE i Theilov koeficijent U bliži 0, dok je vrijednost koeficijenta R što bliže 1.

U tablici (4.1) su izračunate vrijednosti MAPE, R i Theilov koeficijent U po godinama za sve godine iz podatkovnog skupa. Iz navedene tablice se može zaključiti da su vrijednosti MAPE i Theilov koeficijent U za sve indekse poprilično dobri, dok vrijednost koeficijenta R varira. Razlog odstupanja od rezultata iz [2] jednim dijelom je i zbog činjenice da je uzet broj epoha jedan red veličine manji. Također, ako dođe do naglog skoka cijena na tromjesečnom periodu na kojem se model testira, a model je naučen na podacima iz zadnje 2 godine na kojima nije bilo naglih promjena, model ne može predvidjeti takvu iznenadnu promjenu. To je najbolje vidljivo kod CSI 300 indeksa sa slike (4.1). Moguće rješenje ovog problema je smanjenje perioda za testiranje, tj. provoditi ponovno učenje češće, periodom koji je manji od 3 mjeseca.

Na slici (4.1) nalaze se grafovi na kojima na osi apcisa se nalaze datumi između 2011. i 2017. godine, dok su na osi ordinata pripadne vrijednosti za svaki pojedini indeks. Plavom linijom na grafovima je prikazana stvarna cijena, a narančastom linijom predviđena cijena navedenim modelom.

Rezultati su dobiveni korištenjem ćelija LSTM bez predobrade podataka nekom od transformacija za uklanjanje šuma u podacima. Konkretno, u modelu WSAE-LSTM [2] kao predobrada podataka se koristila valićna transformacija (engl. *wavelet transform*), nakon uklonjenog šuma podaci su ulazili u niz autoenkodera koji izvlače generalizirana svojstva na nenadzirajući način. Na kraju, podaci sa izlaza niza autoenkodera dolaze na ulaz ćelije LSTM. Navedeni model postiže bolje rezultate od korištenja niza ćelija LSTM.

Ovakav model u stvarnosti bi se primjenjivao tako što bi se svaki dan u model ubacila sekvenca 4 vektora sa svim izračunatim vrijednostima atributa (iz tablice 3.1) prethodna 4 dana, te bi se kao izlaz dobila vrijednost sutrašnje cijene u trenutku zatvaranja. Rezultat koji se dobije modelom, može poslužiti korisniku modela kao pomoć u poslovnoj odluci, bez da je sam utrošio ikakvo vrijeme u razmatranje povijesnog kretanja cijena i drugih povezanih varijabli koje utječu na cijenu, a sastavni su dio ovog modela. Svakih 3 mjeseca (ili češće) korisnik bi ponovno pokrenuo model kako bi ažurirao model s novim podacima iz protekla 3 mjeseca, zadržavajući podatke iz prethodne dvije godine.



Slika 4.1: Prikaz stvarne i predviđene cijene u trenutku zatvaranja za svaki pojedini indeks

Tablica 4.1: Metrike za detaljnije objašnjenje rezultata

Indeks	Godina	MAPE	R	Theil U
S&P 500	2011	0.0684	0.2428	0.0408
	2012	0.0345	0.6712	0.0208
	2013	0.1057	0.8409	0.0589
	2014	0.0879	0.8118	0.0478
	2015	0.0467	0.2176	0.0304
	2016	0.0225	0.8716	0.0140
CSI 300	2011	0.0423	0.8503	0.0265
	2012	0.0958	0.7146	0.0493
	2013	0.0361	0.6574	0.0232
	2014	0.0517	0.8344	0.0499
	2015	0.2382	-0.2260	0.1654
	2016	0.1044	0.4763	0.0601
DJIA	2011	0.0820	-0.0353	0.0488
	2012	0.0308	0.5702	0.0183
	2013	0.0731	0.7736	0.0416
	2014	0.0595	0.6804	0.0333
	2015	0.0307	0.5452	0.0207
	2016	0.0195	0.8897	0.0116
Nikkei 225	2011	0.0609	0.4345	0.0359
	2012	0.0245	0.8458	0.0152
	2013	0.1753	0.6532	0.1076
	2014	0.0629	0.4096	0.0443
	2015	0.1180	0.2083	0.0721
	2016	0.0710	0.4525	0.0405
Nifty 50	2011	0.0407	0.7672	0.0258
	2012	0.0251	0.8688	0.0159
	2013	0.0275	0.7020	0.0169
	2014	0.1423	0.7342	0.0891
	2015	0.0702	-0.4049	0.0513
	2016	0.0311	0.7951	0.0205
Hang Seng	2011	0.1357	-0.4055	0.0927
	2012	0.0210	0.8776	0.0137
	2013	0.0304	0.6678	0.0185
	2014	0.0304	0.5728	0.0194
	2015	0.0497	0.8448	0.0358
	2016	0.0697	0.5697	0.0387

5. Zaključak

U ovom radu obrađena je primjena rekurentnih neuronskih mreža za predviđanje vrijednosti cijene u trenutku zatvaranja (engl. *close price*) više indeksa. Model temeljen na nizu ćelija LSTM se pokazao kao solidan za ovu vrstu problema. Navedeni model je malo spor na nagle promjene u vrijednosti indeksa što bi se moglo popraviti izloženim modelom s manjim intervalom testiranja, tj. češćim provođenjem učenja.

Model predviđa 1 dan unaprijed što je korisno za dnevne trgovce (engl. *day trader*) na burzi koji dnevno mogu učiniti jednu kupnju ako model predviđa rast cijene za idući dan, odnosno prodaju ako model predviđa pad cijene. Model je neupotrebljiv za one koji žele neke dugoročne prognoze na financijkom tržištu.

Pokazana je prednost ćelija GRU i LSTM u odnosu na osnovni oblik ćelije RNN u rješavanju eksplodirajućeg i nestajućeg gradijenta kod rekurentnih neuronskih mreža.

Navedene rezultate moguće je poboljšati primjenom neke metode predobrade podataka koja će ukloniti šum iz podataka. Također, dodatno unaprijeđenje moglo bi se postići kombinacijom s drugim modelima strojnog učenja.

LITERATURA

[1] Shervine Amidi Afshine Amidi. A detailed example of how to use data generators with keras. https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly, 19. 5. 2018. Pristupljeno: lipanj 2020.

[2] Wei Bao, Jun Yue, i Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE*, 12(7):1–24, 07 2017. doi: 10.1371/journal.pone.0180944. URL https://doi.org/10.1371/journal.pone.0180944.

[3] Friedhelm Bliemel. Theil's forecast accuracy coefficient: A clarification. *Journal of Marketing Research*, 10(4):444–446, 1973. ISSN 00222437. URL http://www.jstor.org/stable/3149394.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, i Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[5] Xiao Ding, Yue Zhang, Ting Liu, i Junwen Duan. Deep learning for event-driven stock prediction. U *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, stranica 2327–2333. AAAI Press, 2015. ISBN 9781577357384.

[6] Georgios Drakos. What is a recurrent neural networks (rnns) and gated recurrent unit (grus). https://gdcoder.com/what-is-a-recurrent-neural-networks-rnns-and-gated-recurrent-unit-grus/. Pristupljeno: lipanj 2020.

[7] Prince Grover. 5 regression loss functions all machine learners should know. https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0, 5. 6. 2018. Pristupljeno: lipanj 2020.

[8] Zhiqiang Guo, Huaiqing Wang, Quan Liu, i Jie Yang. A feature fusion based forecasting model for financial time series. *PLOS ONE*, 9(6):1–13, 06 2014. doi: 10.1371/journal.pone.0101113. URL https://doi.org/10.1371/journal.pone.0101113.

[9] Raimi Karim. Animated rnn, lstm and gru. https://towardsdatascience.com/animated-rnn-lstm-and-gru-ef124d06cf45, 14. 12. 2018. Pristupljeno: lipanj 2020.

[10] Cristopher Olah. Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/#fn1, 27. 8. 2015. Pristupljeno: lipanj 2020.

[11] Martin Tutek. Napredne povratne neuronske mreže. http://www.zemris.fer.hr/~ssegvic/du/du6recurrent2.pdf, 13. 5. 2020. Pristupljeno: lipanj 2020.

[12] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, i Keying Ye. *Probability & statistics for engineers and scientists*. Pearson Education, Upper Saddle River, 8th izdanju, 2007.

Predviđanje vrijednosti financijskih podataka temeljeno na rekurentnim neuronskim mrežama

Sažetak

U ovom radu obrađena je analiza financijskih podataka pomoću neuronske mreže koja koristi ćelije s dugom kratkoročnom memorijom (engl. long short-term memory, LSTM). Detaljno su objašnjene tri vrste rekurentnih neuronskih mreža: osnovna rekurentna neuronska mreža (RNN), LSTM i propusna povratna ćelija (engl. gated recurrent unit, GRU). Objašnjena je prednost ćelija LSTM i GRU naspram jednostavne RNN zbog rješavanja problema eksplodirajućeg i nestajućeg gradijenta. Prikazani su dobiveni rezultati i njihova interpretacija pomoću srednje apsolutne postotne pogreške, Pearsonovog koeficijenta i Theilovog koeficijenta U.

Ključne riječi: strojno učenje, rekurentne neuronske mreže, predviđanje, financije, LSTM, GRU

Prediction of Financial Data Values based on Recurrent Neural Networks

Abstract

This paper deals with the processing of financial data using a long short–term memory (LSTM). Three types of recurrent neural networks are explained in detail: the simple recurrent neural network (RNN), LSTM and the gated recurrent unit (GRU). The advantage of LSTM and GRU over simple RNN due to solving the problem of exploding and vanishing gradient is explained. The obtained results and their interpretation are presented using the mean absolute percentage error (MAPE), Pearson's coefficient and Theil's U coefficient.

Keywords: machine learning, recurrent neural network (RNN), prediction, finance, LSTM, GRU