SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5944

Tehnička analiza financijskih podataka s ciljem predviđanja budućih vrijednosti

Filip Pažanin

Zagreb, lipanj 2019.

Sadržaj

1. Uvod

Burza je organizirano mjesto koje nam omogućava da kupimo odnosno prodamo jedinice vlasništva tvrtke koje nazivamo dionice. Ako zarada tvrtke čije dionice posjedujemo bude pozitivna, mi posjedujemo dio te zarade. Isto tako, ako zarada bude negativna, odnosno tvrtka bude u gubitku, i mi ćemo biti u gubitku. U slučaju da „pogodimo" vrijeme ulaska u kupnju neke dionice, možemo ostvariti izniman profit u vrlo kratkom periodu. Naravno, pitanje koje se nameće je možemo li na neki način predvidjeti kretanje dionica na burzi? Je li moguće detaljnom analizom podataka iz prošlosti doći do zaključaka o kretanju cijene u budućnosti?

Grana računarstva, znanost o podacima (engl. *data science*) bi nam mogla pokušati dati odgovor na navedeno pitanje. No, prema hipotezi efikasnog tržišta, cijene dionica su slučajne i nepredvidive. Da tržište dionica ipak nije sasvim nepredvidivo dokazuju najveće tvrtke koje se bave ulaganjem i općenito financijama kao što su JPMorgan i Goldman Sachs koje zapošljavaju kvalificirane brokere već godinama, koji grade modele za predviđanje budućih vrijednosti dionica na temelju kretanja dionica unazad određeni period. Osim prošlih cijena dionice, u obzir možemo uzeti i opće raspoloženje i mišljenje o dionici u društvu odnosno medijima (engl. sentiment analysis), podatke o količini trgovanja dionicom na burzi, dividendama koje tvrtka isplaćuje te mnoge druge. U okviru ovog završnog rada, za predikciju cijena dionica bit će uzete u obzir samo prošle cijene dionica da bi se smanjila složenost programskog dijela projekta.

Postoje mnogi modeli po kojima možemo analizirati te potom predviđati financijske vremenske nizove. U ovom radu bit će obrađena tri modela po mome izboru. Dva od tri odabrana modela koristit će strojno učenje koje postaje sve popularnije u svim područjima ljudskog djelovanja, pogotovo u financijama. Odabrani modeli su model ARIMA koji se pokazuje iznimno jednostavan no vrlo uspješan u predikciji vremenskih nizova, duboka neuronska mreža (engl. *Deep LSTM Neural Network*) čiju ćemo uspješnost usporediti s prethodno navedenim modelom te ćemo za kraj izraditi simulaciju u kojoj

istreniranom agentu davasmo određeni početni kapital kojeg on pokušava uvećati trgovanjem. No, prije nego krenemo na obrađivanje gore navedenih modela, nužno je savladati temeljne termine i pozadinu računalnih financija.

1.1. Financijski vremenski nizovi

Najprije se valja upoznati s definicijom financijskih vremenskih nizova. Financijski vremenski nizovi su nizovi cijena nekog financijskog dobra na definiranom vremenskom intervalu. Ovaj rad sadrži elemente tehničke analize, matematičkog modeliranja i računalne znanosti te analizira kretanja jednih od pet najlikvidnijih i „najaktivnijih" dionica na Američkoj burzi.

Financijske vremenske nizove analiziramo da bismo predvidjeli buduću vrijednost te tako ostvarili profit ispravno ulagajući određeni kapital. Kada se govori o analizi financijskih vremenskih nizova, tada se uglavnom misli na osnovne dvije analize: fundamentalnu i tehničku analizu, no od velike je važnosti i analiza psihologije investitora. Fundamentalna analiza bazira se na konkretnim podacima o poslovanju odnosno financijskim izvješćima neke kompanije te ukupnom ekonomskom analizom sposobnosti menadžmenta kompanije, sektora u kojem se ona nalazi te sveukupnog mikro i makroekonomskog okruženja u kojem posluje. Tehnička analiza, s druge stane, nastoji uočiti obrasce kretanja vrijednosti cijene dionice pomoću analize statistika dobivenih iz aktivnosti tržišta, poput povijesnih podataka i volumena trgovanja. Iako bismo mogli pomisliti da bismo se primarno baš ovim analizama trebali voditi pri odlučivanju o bilo kakvim akcijama na tržištu, istraživanje je pokazalo da 73% japanskih institucionalnih investitora daje prednost psihologiji pred fundamentima tržišta. Ponašanje mase na tržištu je prilično jednostavno objasniti. U vrijeme rasta tržišta i općenitog optimizma, ljudi su skloni stvoriti vrlo nerealna očekivanja. Kako cijene rastu, tako većina ulagača kupuje sve više, dižući tako cijene dionica čime privlače nove ulagače koji opet svojim kupnjama - nadajući se zaradi - dižu cijene još više i tako skoro unedogled. Svaki tržišni zakon kaže da se taj trend mora prekinuti prije ili kasnije, a onda obično nastaju veliki gubici zbog panike koja nastaje preokretanjem trenda. [8] Tehnička analiza ima veoma dugu povijest, još od kraja 19. stoljeća, kada je Charles Dow, osnivač Dow Jonesa, iznio svoju teoriju za identificiranje trendova u cijenama dionica. Veoma dugo, tehnička analiza nije bila cijenjena od strane ekonomista, za razliku od fundamentalne analize. Neki su ove analize

uspoređivali s odnosom astrologije i astronomije. Veliki udarac tehničkoj analizi bio je eksperiment proveden 1973. godine u kojem su majmuni bacanjem strelica odabrali profitabilnija ulaganja od investicijskih stručnjaka koji su ulaganja birali tehničkom analizom. Još od šezdesetih godina prošlog stoljeća ekonomisti vjeruju, više ili manje, u teoriju efikasnosti tržišta. Tvrde da na efikasnim tržištima cijene reflektiraju sve dostupne informacije, pa kretanje cijena u prošlosti dionica ne može ništa korisno savjetovati glede budućeg kretanja cijene. No bez obzira na sve, tehnička analiza je više umjetnost, a manje znanost, koju koriste i najuspješniji investitori. [1]

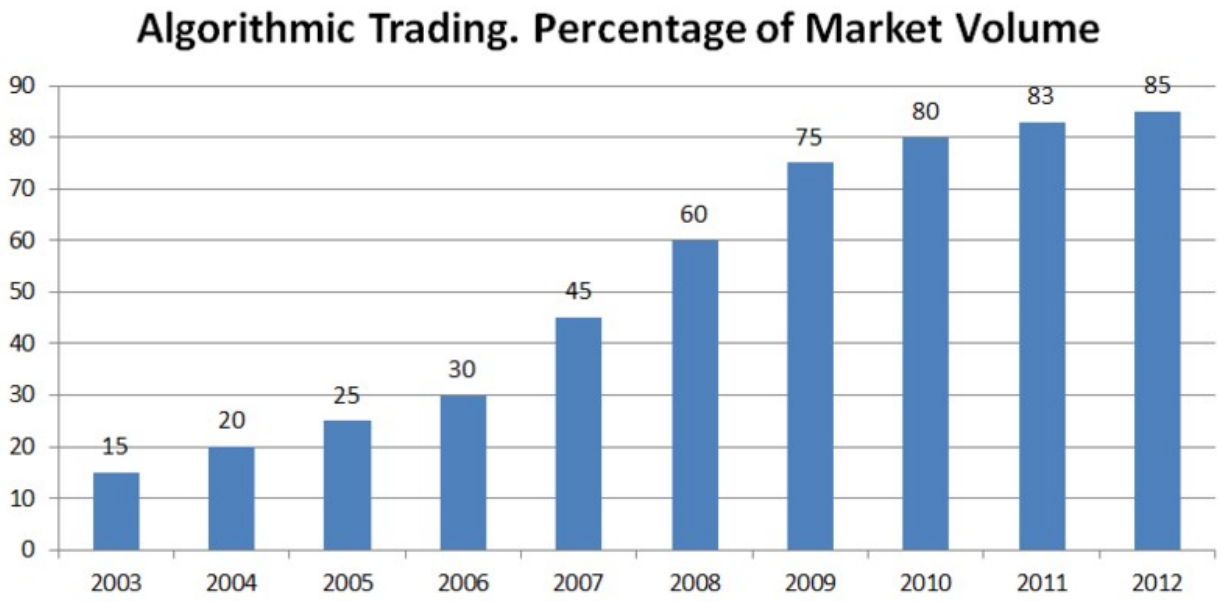
1.2. Cilj i motivacija

Prije svega nužno je znati da ne postoji jedan univerzalan model koji će savršeno raditi nad svim dionicama, te da se pri ulaganju ne bi trebali oslanjati samo na dostupne nam prediktivne modele, pogotovo ne na one koje koriste samo cijene dionica u prošlosti kao zavisnu varijablu te vrijeme kao nezavisnu varijablu. Također bitno je imati na umu da ako želimo iskoristiti naše modele za predikciju kratkog nadolazećeg vremenskog perioda te to iskoristiti u stvarnom trgovanju, ti se algoritmi moraju izvršavati iznimno brzi, što znači da se isključuje korištenje umjetne inteligencije odnosno strojnog učenja zbog prevelike složenosti. No ako želimo prognozirati događanja na duži period te to iskoristiti u stvarnom svijetu, korištenje strojnog učenja bi trebalo davati bolje rezultate.

Glavna motivacija za bavljenjem ovom temom leži u samoj kompleksnosti zadatka. Potrebno je spojiti znanja ekonomije, matematike i računarske znanosti da bismo se mogli baviti problematikom predviđanja financijskih podataka.

Ako pogledamo izvještaje financijske industrije danas, primijetit ćemo da su tržišta dionica, obveznica te raznih indeksa u velikoj količini automatizirana. Jedan od izvještaja je prikazan na slici 1, gdje se vidi kako kroz period od 2003. godine do 2012. godine, postotak algoritamskih transakcija u ukupnom broju transakcija na financijskom tržištu, raste. Treba istaknuti da su određeni algoritmi koji se koriste pri trgovanju zaradili određenim korporacijama milijarde dolara. Iz navedenog možemo zaključiti da se znanje iz

ovog područja može jako dobro iskoristiti što nam može biti jedan jak motiv za istraživanjem i usavršavanjem vještina analize i predviđanja. Također, poboljšavanjem performansi i padom cijena računalnog hardvera te popularizacije kulture otvorenog koda (engl. *open source*) mi kao pojedinci možemo upoznati i koristiti se raznim kompleksnim algoritmima za koje su prije jedino velike korporacije imale hardver potreban za izvršavanje. [7]



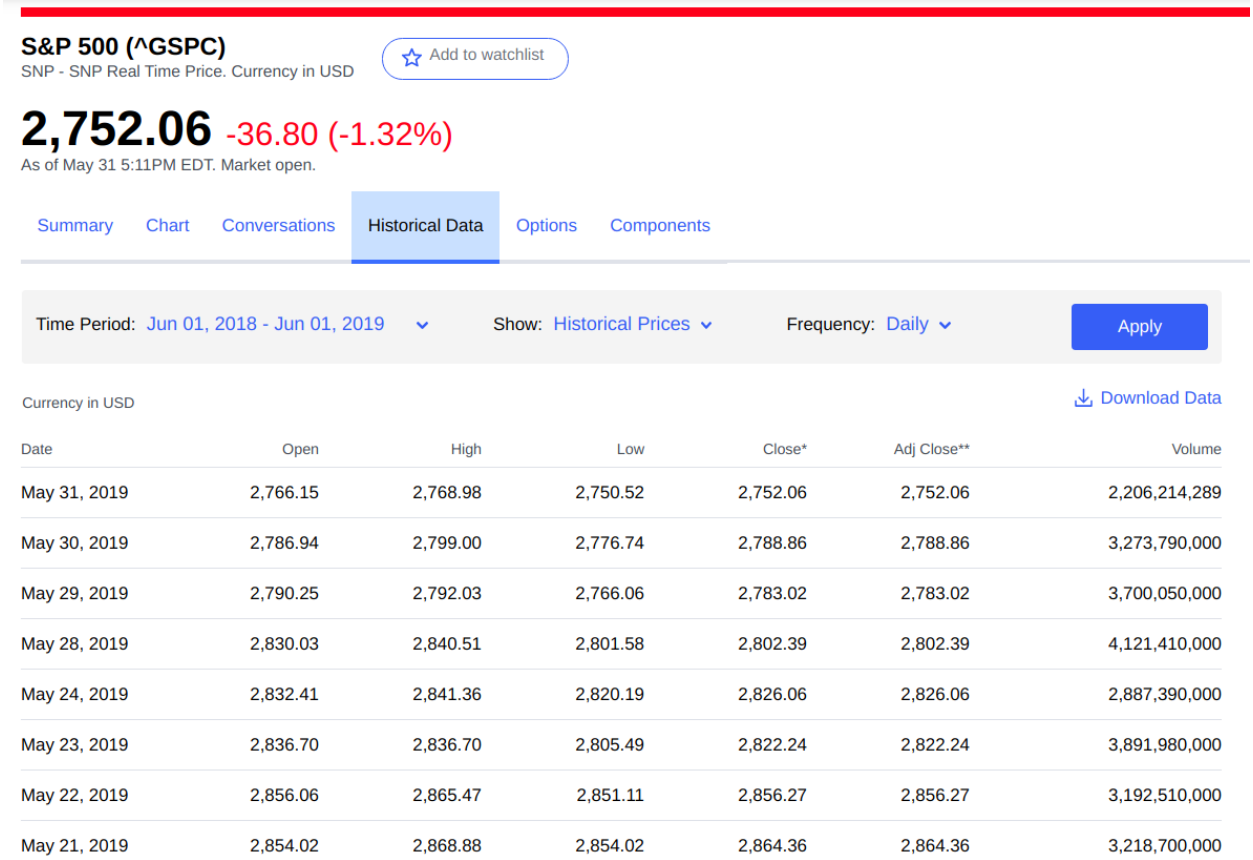
Slika 1: Broj algoritamskih transakcija kroz povijest [7]

2. Podaci

Svi podaci za analizu preuzeti su sa stranice finance.yahoo.com (slika 2). Dionice čiji se podatci promatraju, predviđaju i testiraju u ovom radu su sljedeće:

1. **Bank of America Corporation (BAC)**
2. **Ford Motor Company (F)**
3. **Microsoft Corporation (MSFT)**
4. **Alphabet Inc. (GOOG)**
5. **S&P 500 (^GSPC)**
6. **International Business Machines Corporation (IBM)**
7. **QUALCOMM Incorporated (QCOM)**

Za svaki dan trgovanja pojedine dionice imamo dostupno pet cijena (prvu, najvišu, najnižu, zadnju cijenu i prilagođenu zadnju cijenu). Prva cijena je cijena dionice prilikom otvaranja burze u 9:30 h pojedinog radnog dana, zadnja cijena je zaključna cijena na zatvaranju burze u 16:00 h dok su najviša i najniža cijena vrijednosti najviše i najniže postignute cijene tijekom dana. Također imamo prilagođenu zadnju cijenu koja ne predstavlja zadnju cijenu po kojoj je dionica istrgovana već uzima u obzir i dividende, u međuvremenu, isplaćene u gotovini i dionicama te moguće promijene u ukupnom broju dionica te prema tim parametrima i zadnjoj cijeni formira vrijednost. Zadnja stavka podataka je obujam trgovanja dionice, odnosno broj, količina dionica određene tvrtke koja se istrgovala toga pojedinog dana.



S&P 500 (^GSPC)
SNP - SNP Real Time Price. Currency in USD
Add to watchlist

2,752.06 -36.80 (-1.32%)
As of May 31 5:11PM EDT. Market open.

Summary | Chart | Conversations | Historical Data | Options | Components

Time Period: Jun 01, 2018 - Jun 01, 2019 Show: Historical Prices Frequency: Daily Apply

Currency in USD Download Data

Date	Open	High	Low	Close*	Adj Close**	Volume
May 31, 2019	2,766.15	2,768.98	2,750.52	2,752.06	2,752.06	2,206,214,289
May 30, 2019	2,786.94	2,799.00	2,776.74	2,788.86	2,788.86	3,273,790,000
May 29, 2019	2,790.25	2,792.03	2,766.06	2,783.02	2,783.02	3,700,050,000
May 28, 2019	2,830.03	2,840.51	2,801.58	2,802.39	2,802.39	4,121,410,000
May 24, 2019	2,832.41	2,841.36	2,820.19	2,826.06	2,826.06	2,887,390,000
May 23, 2019	2,836.70	2,836.70	2,805.49	2,822.24	2,822.24	3,891,980,000
May 22, 2019	2,856.06	2,865.47	2,851.11	2,856.27	2,856.27	3,192,510,000
May 21, 2019	2,854.02	2,868.88	2,854.02	2,864.36	2,864.36	3,218,700,000

Slika 2: Podaci dostupni na web stranici yahoo finance

https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC

U svim analizama koristit ćemo zaključne cijene dionica iako se u mnogim analizama i financijskim tehničkim sustavima koriste druge dnevne cijene pa čak i njihove kombinacije. Zaključna cijena je prosječna cijena ponderirana količinom svih transakcija dionice sklopljenih određeni trgovinski dan unutar knjige ponuda.

Financijske varijable koje se najčešće analiziraju su:

- Cijene - (pojedine cijene, kombinacije cijena, indeks cijena, tečajevi)
- Prinosi – (prinosi dionica, indeksi prinosa, kamatne stope)
- Volatilnost – (pokazatelja rizika)

- Količina prometa - (obujam)

Financijski podaci imaju određena svojstva koja utječu na odabir i primjenu statističkih metoda u istraživanjima kao što su:

- Netipične vrijednosti (*engl.outliers*),
- Trendovi (*engl. trends*),
- Grupiranje volatilnosti (*engl. volatility clustering*)

Osnovni statistički pokazatelji u financijskoj analizi:

- Prosječna vrijednost varijable (prosječna vrijednost, median, mod)
- Varijabilnost podataka s obzirom na mjere centralne tendencije (standardna devijacija, varijanca)
- Distribucija podataka
- Relacija između varijabli (histogrami, korelacije, kovarijance, poligon frekvencija)

Osnovne transformacije podataka:

- Logaritmiranje
- Izračunavanje prinosa
- Centriranje (*engl. de-meaning*)
- Uklanjanje trenda(*engl. de-trending*)
- Pomaci varijabli (unazad *engl. lagging*, unaprijed *engl. leading*)

Osnovne metode ocjenjivanja predviđanja podataka:

- *Root Mean Squared Error (RMSE)*
- *Mean Absolute Percent Error (MAPE)*
- *Mean Absolute Scale Error (MASE)*

3. Modeli

U okviru ovog završnog rada izradit ćemo i opisati tri modela za predviđanje kretanja cijene u budućnosti. Samo modeliranje kretanja vrijednosti dionice leži na pretpostavci da se cijene dionica ne kreću posve nasumično već po određenim obrascima koje možemo odrediti analizom povijesnih podataka.

3.1. Model ARIMA

ARIMA je razred statističkih modela namijenjenih za analizu i predviđanje vremenskih nizova podataka. Ona eksplicitno pokriva skup standardnih struktura u podacima vremenskih serija i kao takva pruža jednostavnu, ali moćnu metodu za izradu vještih predikcija vremenskih serija. ARIMA je akronim koji znači *Auto-Regressive Integrated Moving Averages.* Model ARIMA za predviđanje stacionarnih vremenskih serija nije ništa drugo nego linearna (kao linearna regresija) jednadžba. Predikcija ovisi o parametrima (p, d, q) modela ARIMA:

1. **AR**: Autoregresija (engl. Autoregression) (p). Autoregresija označuje zavisnost vrijednosti promatrane varijable o određenom broju vrijednosti varijabla koje su joj prethodile. Uzmimo za primjer da p iznosi 5. U tom slučaju za izračun vrijednosti x(t) u obzir će se uzimati vrijednosti x(t-1) ... x(t-5).

2. **I**: Integrated (d). Parametar koji koristimo kako bi diferencirali „sirove" podatke u svrhu stacioniranja podataka. Npr. može se izvesti tako da oduzmemo vrijednost trenutno promatranog opažanja s vrijednosti prethodnog opažanja.

3. **MA**: Pomični prosjek (engl. Moving Average) (q). Pomični prosjek je model koji koristi ovisnost između promatrane varijable i rezidualne pogreške dobivene modelom pomičnog prosjeka koji se računa na nekoliko prijašnjih vrijednosti trenutno promatrane varijable. Na primjer, ako parametar q iznosi 5 tada će se

predikcija za x(t) računati s obzirom na e(t-1) ... e(t-5) gdje je e(i) razlika između pomičnog prosjeka kod i-tog slučaja i stvarne vrijednosti i-tog slučaja. [9]

Podešavanjem gore navedenih parametara konstruira se linearni regresijski model koji uključuje određena obilježja. Podaci koji se koriste pri modeliranju diferencirani su kako bi ih se učinilo stacionarnim odnosno kako bi se uklonio trend i sezonske strukture koje inače negativno utječu na regresijski model.

Vrijednost 0 može se koristiti za parametar, što ukazuje da se taj element modela ne smije koristiti. Tako, model ARIMA može biti konfiguriran za obavljanje funkcije ARMA modela, pa čak i jednostavnog AR, I ili MA modela.

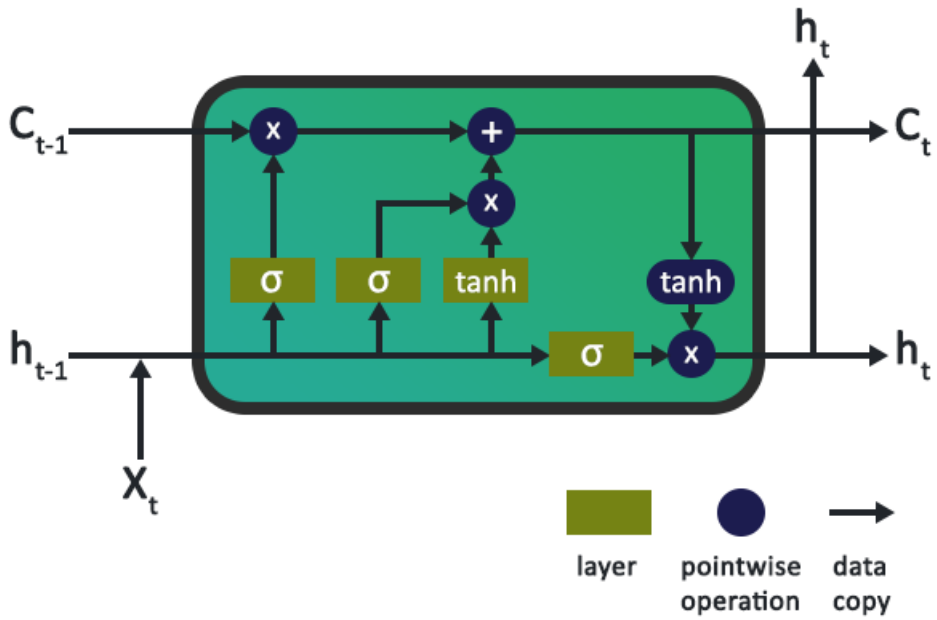
3.2. Duboke neuronske mreže LSTM

Jedan od temeljnih problema tradicionalnih arhitektura neuronskih mreža, dugo vremena, bila je nesposobnost interpretiranja sekvenci ulaza, tj. nizova podataka koji su bili međusobno zavisni odnosno imali kontekst kao cjelina. Takva vrsta podataka su rečenice odnosno jezik sam po sebi, koji funkcionira na principu fraza, nakon neke pojedine riječi možemo s velikom točnošću pretpostaviti koja će riječ ili izraz slijediti. Jednostavno rečeno, tradicionalne neuronske mreže uzimaju svaki put samostalni vektor podataka i nemaju koncept memorije da im pomogne u zadacima u kojima je potrebno pamćenje.

Rani pokušaj rješavanja ovoga bio je korištenje jednostavnog pristupa povratne sprege za neurone u mreži gdje je izlaz bio uključen u ulaz kako bi se osigurao kontekst na posljednjim viđenim ulazima. Taj pristup se naziva rekurentne neuronske mreže (RNN). Dok su ti RNN-ovi radili do određenog stupnja, imali su prilično velik nedostatak, a to je da bi svaka značajnija upotreba istoga dovela do problema koji se zove problem nestajućeg gradijenta (engl. *Vanishing Gradient Problem*). Taj pojam možemo opisati kao problem neuronske mreže u kojem kako dodajemo više slojeva koji koriste određenu aktivacijsku funkciju, gradijent funkcije gubitka (engl. *loss function gradient*) se približava nuli te tako čine mrežu iznimno tešku za treniranje. Neuroni mreže *Long Short Term Memory* (LSTM)

isto kao i oni mreže RNN omogućavaju rješavanje sekvencijskih i vremenskih problema uz pomoć memorije unutar svojih cjevovoda (engl. *pipeline*) no rješavaju i problem nestajućeg gradijenta te su tako postale veoma pogodne za široku upotrebu.

Na slici 3 prikazan je dijagram tipičnog djelovanja neurona LSTM. Sastoji se od nekoliko slojeva i operacija u točkama koje djeluju kao vrata za unos, izlaz i zaboravljanje podataka koji utječu na stanje ćelije LSTM. Ćelija LSTM je ta koja omogućava odnosno zadržava dugoročnu memoriju i kontekst u mreži. [5]



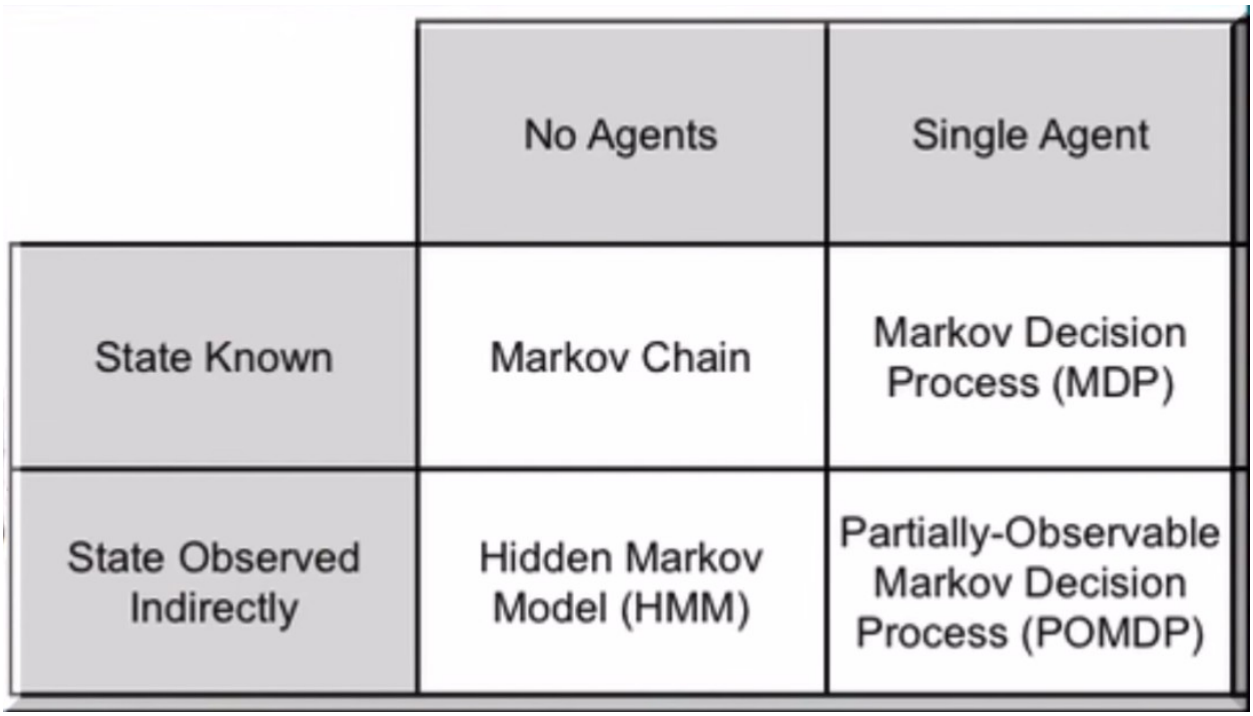
Slika 3: Djelovanje neurona LSTM [4]

3.3. Q-učenje

Q-učenje je algoritam potpornog učenja bez izravnog znanja modela (engl. *model-free reinforcement learning algorithm*). Koristeći potporno učenje (engl. *reinforcement learning*), problem predikcije cijena dionica s obzirom na vrijeme možemo promatrati kao Markovljev proces odlučivanja (engl. *Markov Decision process*), proces koji se sastoji od

stanja, akcija i nagrada. Agent koji se nalazi u našoj zadanoj okolini, može izvršiti akciju. U našem slučaju, postoje tri moguće akcije, kupnja (engl. *buy*), prodaja (engl. *sell*) i zadrži (engl. *hold*). Aktiviranjem akcije kupnja, agent će kupiti onoliko dionica koliko može s obzirom na cijenu dionice i količinu novaca koje posjeduje na računu, dok će akcija prodaja, na odabranoj dionici, biti izvršena tako da se prodaju sve dionice pojedine tvrtke te se sav novac dobiven od iste prodaje, doda na račun u obliku gotovine. Ako kupujemo dionice više tvrtki onda ćemo novac koji posjedujemo i koji možemo upotrijebiti za kupnju, jednoliko podijeliti. Ako agent odabere akciju zadrži tada se ništa ne događa. U svakom trenutku, agent koji se nalazi u stanju s može odabrati 3^n akcija, gdje je *n* broj različitih dionica koje posjedujemo odnosno promatramo. Stanje (engl. *state*) možemo opisati kao red koji sadrži podatke o tome koje i koliko dionica posjedujemo, kolika je cijena tih dionica te koliki nam je iznos gotovine na računu.

Osim našeg agenta, u trgovinskom okruženju koje smo definirali, postoje i drugi agenti odnosno trgovci s vlastitim računima i s vlastitim taktikama trgovanja koji direktno utječu na cijenu dionica. Nažalost, mi kao agent odnosno trgovac pojedinac nemamo pristup njihovim podacima. Zbog takvog stanja okruženja, prema slici 4, mi se zapravo bavimo sa POMDP-om (engl. *Partially-Observable Markov Decision Process*). Shodno tome, mi ne znamo kako izgleda funkcija nagrade (engl. *reward function*) niti funkcija prijelaza (engl. *transition function*) koje su nam potrebne da bi definirali potporno učenje (engl. *reinforcement learning*). U slučaju da znamo ove dvije funkcije bavili bismo se dinamičkim programiranjem da bismo izračunali optimalnu politiku koja bi agentu govorila koje akcije je potrebno poduzeti pod kojim okolnostima. Optimalna bi bila u smislu da maksimizira očekivanu vrijednost ukupne nagrade dobivenu poduzimanjem odabranih akcija koje vode iz početnog stanja u završno stanje. Iako su nam potrebne funkcije nagrade i funkcije prijelaza, mi ih ne moramo eksplicitno definirati. U našem slučaju, poslužit ćemo se Q-učenjem gdje ne učimo eksplicitno funkcije nagrade i prijelaza već direktno učimo preslikavanje stanja i akcija te tako računamo optimalnu politiku.



Slika 4: Stohastički modeli [3]

Q-učenje definira funkciju Q(s,a) koja označava očekivanje sume neposredne vrijednosti nagrade pri prijelazu u sljedeće stanje i diskontirane vrijednosti Q-funkcije sljedećeg stanja. U našem slučaju, Q-funkciju možemo opisati kao najveće moguće stanje na računu koje možemo imati na kraju epizode treninga nakon obavljanja akcije a u stanju s. Q(s,a) prikazuje koliko je dobra neka određena akcija obavljena u stanju s. U našem slučaju, Q-funkcija će ocjenjivati funkcije kupi, prodaj i zadrži u svakom stanju s. Nakon izračuna, odabrat ćemo akciju koja ima najveću Q-funkciju te tako zapravo odabrati optimalnu. Način na koji računamo Q-funkciju je sljedeći :

$$Q(s,a) \sim r + \gamma * Q'(s', a') \qquad (3.1)$$

- s : trenutno stanje agenta
- a : trenutno optimalna akcija
- γ : faktor propadanja

- r : neposredna nagrada kod prijelaza Q --> Q'
- s' : sljedeće optimalno stanje
- a' : optimalna akcija u sljedećem stanju

Navedeni izraz se naziva Bellmanova jednadžba te prikazuje ovisnost trenutne Q-vrijednosti o Q-vrijednosti sljedećeg stanja. Točnije, Bellmanova jednadžba definira maksimalnu buduću nagradu za trenutno stanje i akciju kao zbroj neposredne nagrade i maksimalne buduće nagrade budućeg stanja (slika 8).

U najjednostavnijem scenariju, Q-funkcija je implementirana kao tablica, gdje su stanja redovi, a akcije stupci. Nasumično inicijaliziramo Q-funkciju odnosno stanje koje će biti red (engl. *array*) te zatim iterativno računamo sljedeća četiri koraka. Odabiremo i izvršavamo akciju, promatramo nagradu i novo stanje, dobivene tri vrijednosti ubacujemo u Bellmanovu jednadžbu i ažuriramo Q-funkciju te potom postavljamo sljedeće stanje kao trenutno. Isti postupak ponavljamo do kraja epizode čije smo trajanje sami prethodno definirali.

Ako želimo poboljšati predviđanja, možemo proširiti značajke našeg stanja, na primjer dodati oznaku koja će predstavljati jesu li novosti o tvrtki koje izlaze na portalima dobre ili loše. Pri proširenju stanja, proširujemo cjelokupni prostor stanje-akcija te tako činimo izračun Q-funkcije eksponencijalno vremenski složenijim. U tom slučaju za aproksimaciju Q-funkcije koristimo neuronske mreže, te se tada naša metoda zove duboko Q-učenje (engl. *Deep Q Learning*). [3]

4. Procjena točnosti modela

Najčešće korištena metode za procjenu točnosti predviđanja vremenskih nizova su RMSE (engl. *Root Mean Squared Error*) , MAPE (engl. *Mean Absolute Percent Error*) i MASE (engl. *Mean Absolute Scale Error*). [2]

RMSE je standardna devijacija reziduala (pogreške predviđanja). Reziduali su mjera koliko su udaljene stvarne točke podataka od pretpostavljene regresijske linije. RMSE prikazuje raspršenost reziduala, odnosno govori nam koliko su podaci koncentrirani oko pretpostavljene regresijske linije.

RMSE se, vidljivo je iz formule 4.1, računa kao korijen od sume svih kvadrata reziduala podijeljenih s brojem vrijednosti tj. točaka koje uzimamo u obzir u izračunu.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(z_{fi} - z_{\delta i} \right)^2}{N}} \qquad (4.1)$$

- z_{fi} predstavlja stvarnu vrijednost promatrane varijable u točki *i*
- $z_{f\delta}$ predstavlja očekivanu vrijednost promatrane varijable u točki *i*
- N je broj vrijednosti odnosno točaka koje promatramo

MAPE je mjera koja neovisno o skali, predstavlja omjer pogreške prema stvarnim vrijednostima u obliku postotka. Računa se (formula 4.2.) tako da svaki rezidual,

podijelimo s vrijednosti regresijske linije u promatranoj točki te dobivenu sumu pomnožimo sa 100% i podijelimo s brojem reziduala.

$MAPE = \frac{100\,\%}{n} \sum_{t=1}^{n} \frac{A_t - F_t}{A_t}$	(4.2)

- A_t predstavlja stvarnu vrijednost u točki *t*
- F_t predstavlja vrijednost predviđanja u točki *t*
- n je broj točaka koje promatramo

MASE je konceptualno drukčija mjera ispravnosti prediktivnog modela. Naime, ona, kao što možemo primijetiti iz formule 4.3, uspoređuje koliki je omjer između pogreške našeg modela u odnosu na stvarnu vrijednost i pogreške obične naivne predikcije. MASE ima mnoga povoljna svojstva u usporedbi s drugim metodama za izračunavanje pogrešaka te se stoga često preporučuje njeno korištenje u svrhu uspoređivanja točnosti različitih modela.

$MASE = \frac{\sum_{t=1}^{T} \lvert e_t \rvert}{\frac{T}{T-1} \sum_{t=2}^{T} \lvert Y_t - Y_{t-1} \rvert}$	(4.3)

- e_t predstavlja razliku između stvarne vrijednosti i očekivane vrijednosti dobivene nekim određenim modelom u točki *t*
- Y_t predstavlja stvarnu vrijednost varijable u točki *t*

- $Y_t - Y_{t-1}$ predstavlja pogrešku koja se dobiva računanjem predikcije naivnim modelom, odnosno pogrešku koju dobijemo pretpostavimo li da je vrijednost u točki *t* ista kao i vrijednost u točki *t-1*
- T je broj točaka koje promatramo

5. Implementacija modela i normalizacija podataka

Cijeli završni rad odnosno problem predikcije budućih vrijednosti implementiran je u programskom jeziku Python. Modeli ARIMA i LSTM koriste Python 3.6.8, dok je Q-učenje implementirano uz korištenje Pythona 2.7.15rc1.

Model ARIMA izgrađen je pomoću modula *statsmodels*. On nam je omogućio jednostavnu implementaciju pomoću funkcije ARIMA() kojoj smo predali povijesne podatke kao prvi parametar te postavke modela ARIMA kao drugi parametar. S obzirom na postavljene parametre u našem modelu ARIMA (slika 5), p=2, d=1 i q=0, izgrdit će se autoregresijski model koji će u svrhu predviđanja vrijednosti cijene dionice za određeni dan, u obzir uzimati vrijednosti prethodna dva dana. Parametar d=1 osigurat će nam normalizaciju podataka, na kojima smo prethodno izveli log funkciju da bi smanjili međusobne razlike u vrijednosti i tako učinili normalizaciju i samo predviđanje uspješnijim. [6]

```
while (t < 100):

    model = ARIMA(history, order=(2, 1, 0))
    model_fit = model.fit(disp=-1)

    output = model_fit.forecast(steps=20)

    pred_value = output[0]
    pred_value = dragon.exp(pred_value)

    for i in range(20):

        original_value = test_arima[t + i]
        history.append(test_arima[t + i])

        original_value = dragon.exp(original_value)

        predictions.append(float(pred_value[i]))
        originals.append(float(original_value))

    t = t + 20
```

Slika 5: Implementacija modela ARIMA

Izgrađeni model LSTM koristi razne pakete jezika Python, jedni od značajnijih su *Tensorflow* i *Keras* koji su izgradili duboke neuronske mreže LSTM. U datoteci *config.json* postavili smo metapodatke našeg modela, npr. od koliko i od kojih neurona se sastoji naša mreža, koja je veličina serije (engl. *batch size*) tj. koliko se primjera obrađuje u jednoj iteraciji, također tu postavljamo nazive značajki koje uzimamo u obzir u svrhu predviđanja (u našem primjeru zadnja cijena i volumen trgovanja). Naša mreža LSTM sastoji se od šest slojeva, tri sloja LSTM u kojima se u svakom nalazi po 100 neurona, dva prekidna sloja koja sprečavaju „pretreniranje" (engl. *overfitting*) te jednog izlaznog sloja. Parametri izgrađene mreže LSTM prikazani su na slici 7. Normalizacija podataka se izvršava posebno za svaki vremenski prozor podataka, koji u našem slučaju iznosi 20 dana. Normalizacija se obavlja dijeljenjem iznosa dionice s iznosom koji se nalazi na samom početku vremenskog prozora (formula 5.1). [5]

$n_i = \left(\frac{p_i}{p_0}\right) - 1$	(5.1)

- n_i predstavlja normalizirani podatak
- p_i predstavlja neobrađeni, sirovi podatak na i-tom mjestu u prozoru
- p_0 predstavlja neobrađeni, sirovi podataka na 0. početnom mjestu u prozoru

```python
def predict_sequences_multiple(self, data, window_size, prediction_len):
    #Predict sequence of 20 steps before shifting prediction run forward by 20 steps
    print('[Model] Predicting Sequences Multiple...')
    prediction_seqs = []
    for i in range(int(len(data)/prediction_len)):
        curr_frame = data[i*prediction_len]
        predicted = []
        for j in range(prediction_len):
            predicted.append(self.model.predict(curr_frame[newaxis,:,:])[0,0])
            curr_frame = curr_frame[1:]
            curr_frame = np.insert(curr_frame, [window_size-2], predicted[-1], axis=0)
        prediction_seqs.append(predicted)
    return prediction_seqs
```

Slika 6: Uzastopno predviđanje slijeda od 20 koraka korištenjem istreniranog modela LSTM

```
"model": {
    "loss": "mse",
    "optimizer": "adam",
    "save_dir": "saved_models",
    "layers": [
        {
            "type": "lstm",
            "neurons": 100,
            "input_timesteps": 19,
            "input_dim": 2,
            "return_seq": true
        },
        {
            "type": "dropout",
            "rate": 0.2
        },
        {
            "type": "lstm",
            "neurons": 100,
            "return_seq": true
        },
        {
            "type": "lstm",
            "neurons": 100,
            "return_seq": false
        },
        {
            "type": "dropout",
            "rate": 0.2
        },
        {
            "type": "dense",
            "neurons": 1,
            "activation": "linear"
        }
    ]
}
```

Slika 7: Parametri modela LSTM

Q-učenje smo implementirali pomoću *Keras Sequential* API-a. On nam je omogućio stvaranje neuronske mreže koja u našem slučaju ima jedan skriveni sloj (engl. *hidden layer*) s 32 neurona. Izrađeni model neuronske mreže, agent je koristio za izračun Q-tablice (slika 8). Za normalizaciju podataka poslužili smo se *Scikit-learn* knjižnicom, preciznije metodom *StandardScaler*. U tablici 1 prikazan je postupak normalizacije vrijednosti x pomoću funkcije *StandardScaler*, gdje x predstavlja cijenu dionice. Postupak normalizacije podataka ključan je preduvjet za precizno treniranje modela kojeg ćemo kasnije koristiti u svrhu predikcije. [3]

Standardizacija: $z=\frac{x-\mu}{\sigma}$	(5.2)
Srednja vrijednsot: $\mu=\frac{1}{N}\sum_{i=1}^{N}(x_i)$	(5.3)
Standardna devijacija: $\sigma=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$	(5.4)

Tablica 1: Matematika u pozadini funkcije StandardScaler

```
def replay(self, batch_size=32):
    """ vectorized implementation; 30x speed up compared with for loop """
    minibatch = random.sample(self.memory, batch_size)

    states = np.array([tup[0][0] for tup in minibatch])
    actions = np.array([tup[1] for tup in minibatch])
    rewards = np.array([tup[2] for tup in minibatch])
    next_states = np.array([tup[3][0] for tup in minibatch])
    done = np.array([tup[4] for tup in minibatch])

    # Q(s', a)
    target = rewards + self.gamma * np.amax(self.model.predict(next_states), axis=1)
    # end state target is reward itself (no lookahead)
    target[done] = rewards[done]

    # Q(s, a)
    target_f = self.model.predict(states)
    # make the agent to approximately map the current state to future discounted reward
    target_f[range(batch_size), actions] = target

    self.model.fit(states, target_f, epochs=1, verbose=0)

    if self.epsilon > self.epsilon_min:
        self.epsilon *= self.epsilon_decay
```

Slika 8: Kod zaslužan za izradu Q-tablice

6. Rezultati

Testiranje modela ARIMA i modela baziranog na dubokim neuronskim mrežama LSTM proveli smo na vremenskom periodu od 100 dana. Period, za koje su rađane predikcije, su dani od 28. prosinca 2018 do 22. svibnja 2019. Model ARIMA za svaki pojedini period od 20 dana, od njih ukupno pet u 100 dana, obavlja predikciju na temelju modela kojeg trenira na podacima o cijeni dionice koje su prethodile periodu kojeg računamo unazad nešto više od deset godina (slika 5). Model LSTM smo učili na periodu od 20 dana, na podacima do 28. prosinca 2018, unazad deset godina. Da bismo ga testirali i usporedili s modelom ARIMA bilo je potrebno napraviti predikciju pet vremenskih perioda od 20 dana. Metoda *predvidi* (engl. *predict*) modela LSTM vraća vrijednost koju predviđa za određeni dan na temelju informacije o kretanju cijene dionice u prethodnih 20 dana. Tako je za potrebe predikcije perioda od 20 dana bilo potrebno 20 puta pozvati metodu *predvidi*, predajući joj informacije o cijeni za prethodnih 20 dana, što stvarnih, što predviđenih. Tako smo recimo za predikciju prvog dana u vremenskom prozoru od 20 dana vrijednost računali na temelju 20 prethodnih stvarnih vrijednosti dok smo za vrijednost cijene zadnjeg, 20-tog dana u vremenskom prozoru od 20 dana, u obzir uzimali jednu stvarnu vrijednost i 19 prethodno predviđenih (slika 6).

Testiranje modela ARIMA i LSTM navedenih modela proveli smo na pet različitih dionica, to su dionice: Bank of Amerika Corporation, Ford Motor Company, Microsoft Corporation, Alphabet Inc. i S&P 500. Odabrane su navedene dionice zato što su jedne od najaktivnijih dionica na američkoj burzi te međusobno pokrivaju različite segmente poslovanja.

Rezultate modela ARIMA i LSTM predočili smo tako da smo za svaku od pojedinih dionica nacrtali graf sa stvarnim kretanjem i s očekivanim kretanjem vrijednosti dionice u testnom periodu od 100 dana. Također uz svaki graf kretanja dionice priložili smo i tablicu s rezultatima izračunavanja sljedećih pogrešaka : MAPE, RMSE, MASE.

Slike 9 - 13 i tablice 2 – 6 prikazuju rezultate modela ARIMA, dok slike 14 - 18 i tablice 7 - 11 prikazuju rezultate modela LSTM.

Što se tiče konkretnih rezultata, oba modela najtočnije predviđaju dionicu Microsofta (slika 11 i slika 16), dok do najvećih pogrešaka kod modela ARIMA dolazi kod predviđanja Bank of America dionice (slika 9), a kod LSTM modela kod predviđanja dionice Forda (slika 15). Ako usporedimo pogreške koje generiraju modeli ARIMA i LSTM, vidjet ćemo da za svaku dionicu osim dionice Bank of America, ARIMA stvara manje pogreške pri predikciji, unatoč svojoj konceptualnoj i implementacijskoj jednostavnosti u odnosu na LSTM model. Ako pogledamo graf dionice Bank of America (slika 9 i slika 14) i usporedimo ga s grafovima ostalih dionica, uočit ćemo da je dionica BAC poprilično volatilna u usporedbi s ostalim dionicama koje testiramo. To uzevši u obzir, možemo naslutiti da će model LSTM bolje predviđati vrijednosti nestabilnih dionica u odnosu na model ARIMA.

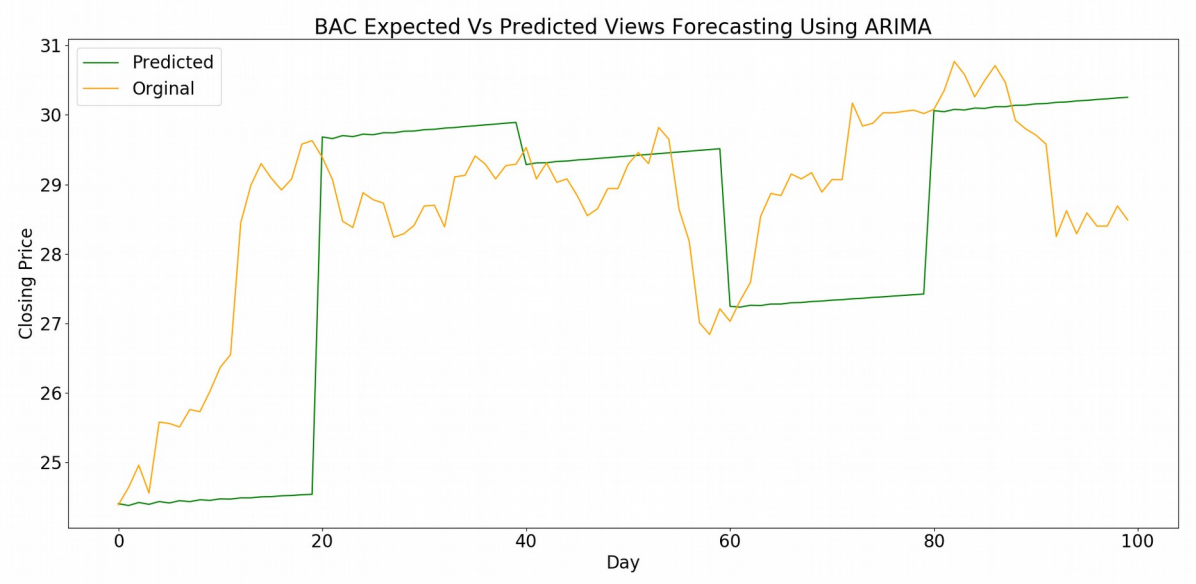
U rezultatima modela ARIMA i LSTM možemo primijetiti zanimljivost koju krije pogreška MASE. Ona nam govori u kojoj je mjeri određeni model kojeg testiramo bolji od primitivne predikcije. Ako pogreška MASE iznosi više od jedan, to znači da primitivna predikcija dava bolje rezultate od promatranog modela. Jedini slučaj u kojem se to dogodilo u našim rezultatima je kod predikcije dionica S&P 500 indeksa korištenjem modela LSTM (tablica 11). Inače je poznato da su indeksne dionice vrlo mirne te je to, uz jako lošu predikciju modela LSTM, razlog zašto je vrijednosti MASE veća od jedan.

Testiranje potpornog učenja izveli smo tako da smo promatrali kretanje vrijednosti računa kojim agent upravlja, čiju smo početnu vrijednost postavili na 20 000 $. Navedeno kretanje promatrali smo kroz vremenski period od 2 000 dana. U tih 2 000 dana agent je svakodnevno odlučivao o akcijama koje će izvršavati s dionicama triju tvrtki, IBM, Microsoft i Qualcomm, čije smo mu podatke, unazad 14 godina, dali da bi se mogao istrenirati. Rezultati su prikazani na grafu (slika 19) gdje se na x-osi nalaze dani, a na y-osi ukupna vrijednost računa. Također je na grafu označena i početna vrijednost, crvenom linijom te konačna vrijednost odnosno vrijednost računa na zadnji dan testiranja, zelenom linijom. Možemo primijetiti da je stanje računa kroz vrijeme veoma volatilno, u tolikoj mjeri da nalikuje na šum. U jednom trenutku agent je uspio utrostručiti vrijednost svog

kapitala, premašivši vrijednost od 60 000 $. No, nedugo nakon, pogrešnim akcijama, agent je smanjio vrijednost svog kapitala na 10 000 $ što je prepolovljena vrijednost početnog kapitala. Iz svega navedenog možemo zaključiti da Q-učenje nije postiglo zadovoljavajuće rezultate usprkos kratkotrajnim uspjesima.

6.1. Rezultati modela ARIMA

1. Bank of America Corporation (BAC)

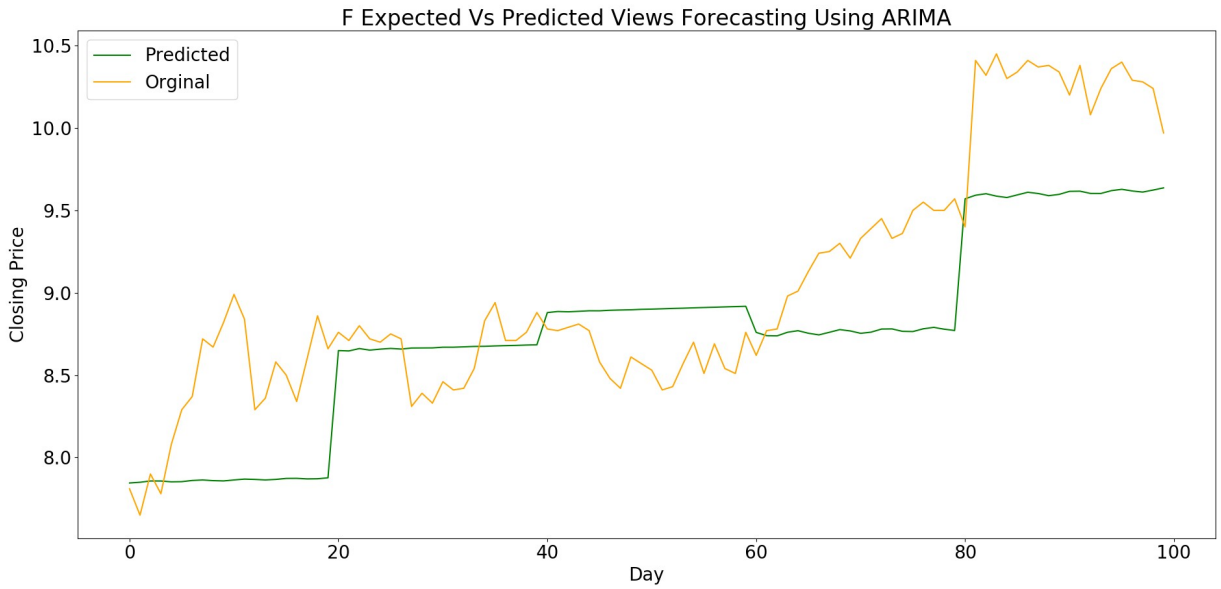


Slika 9: Prikaz stvarnih cijena dionica Bank of America i predviđanja temeljenih na modelu ARIMA

MAPE	4.940635%
RMSE	6.995253
MASE	0.983992

Tablica 2: Pogreške u rezultatima predviđanja cijena dionica Bank of America temeljenih na modelu ARIMA

2. Ford Motor Company (F)

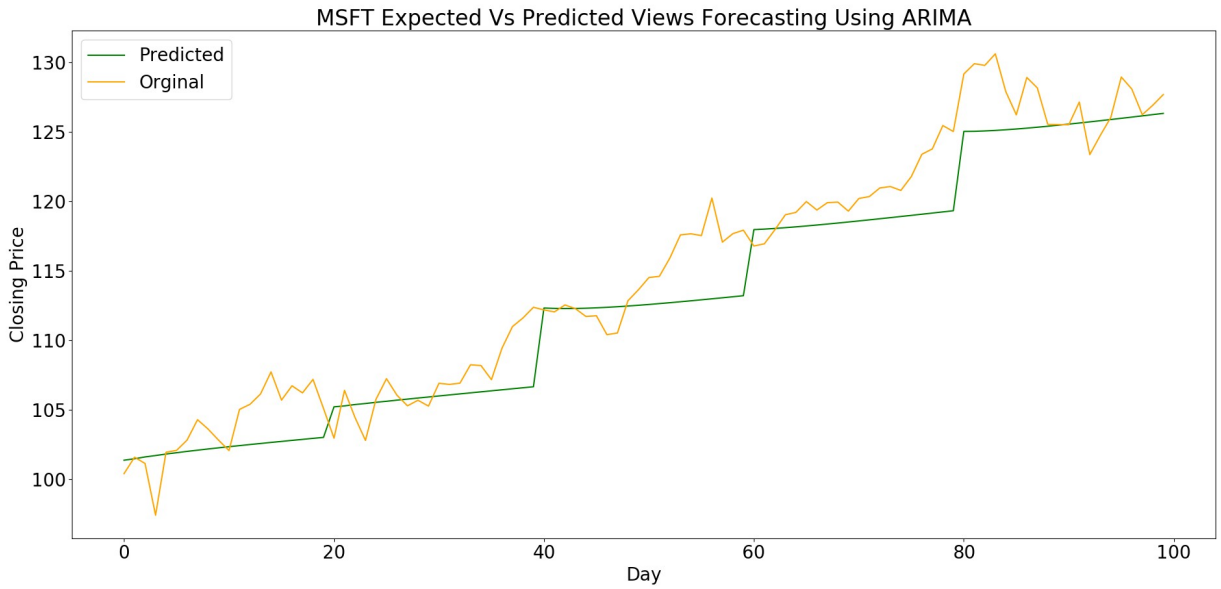


Slika 10: Prikaz stvarnih cijena dionica Forda i predviđanja temeljenih na modelu ARIMA

MAPE	4.717857%
RMSE	5.585456
MASE	0.854190

Tablica 3: Pogreške u rezultatima predviđanja cijena dionica Forda temeljenih na modelu ARIMA

3. Microsoft Corporation (MSFT)

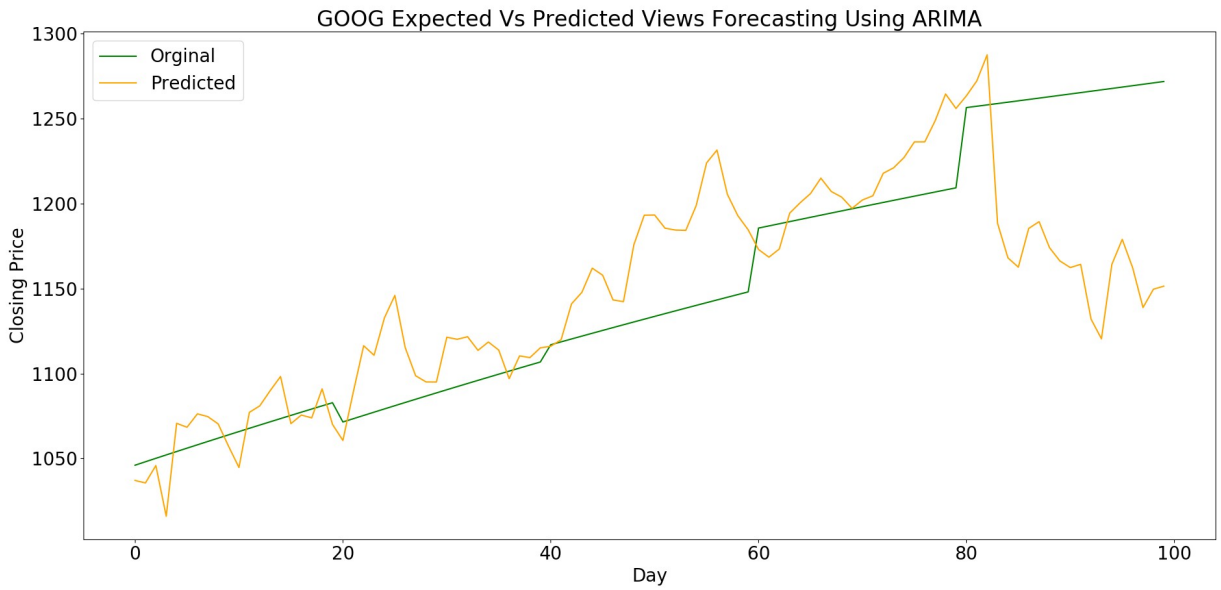


Slika 11: Prikaz stvarnih cijena dionica Microsofta i predviđanja temeljenih na modelu ARIMA

MAPE	1.840582%
RMSE	2.375391
MASE	0.615501

Tablica 4: Pogreške u rezultatima predviđanja cijena dionica Microsofta temeljenih na modelu ARIMA

4. Alphabet Inc. (GOOG)

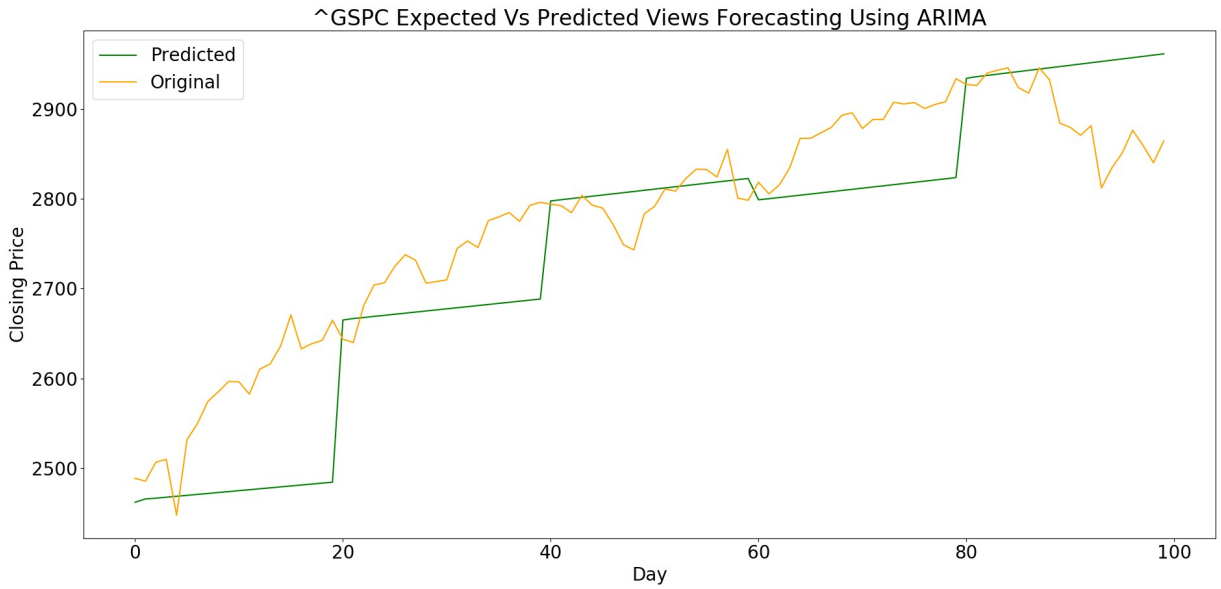


Slika 12: Prikaz stvarnih cijena dionica Googla i predviđanja temeljenih na modelu ARIMA

MAPE	3.212627%
RMSE	4.467371
MASE	0.664628

Tablica 5: Pogreške u rezultatima predviđanja cijena dionica Googla temeljenih na modelu ARIMA

5. S&P 500 (^GSPC)



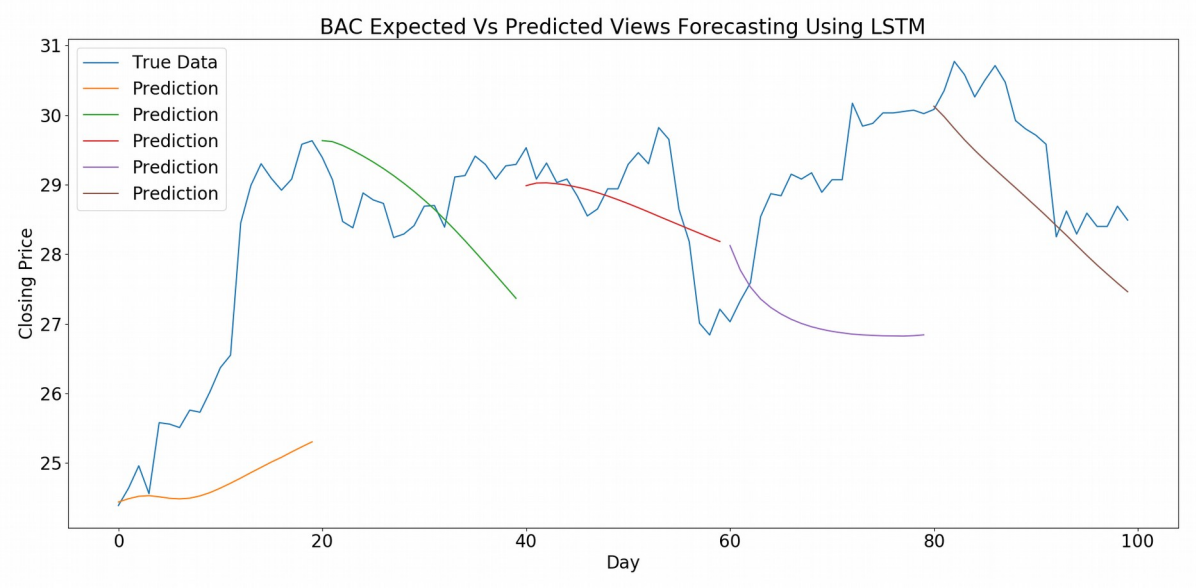
Slika 13: Prikaz stvarnih cijena dionica S&P 500 indeksa i predviđanja temeljenih na modelu ARIMA

MAPE	2.247690%
RMSE	2.843947
MASE	0.960594

Tablica 6: Pogreške u rezultatima predviđanja cijena dionica S&P 500 indeksa temeljenih na modelu ARIMA

6.2. Rezulati modela LSTM

1. Bank of America Corporation (BAC)

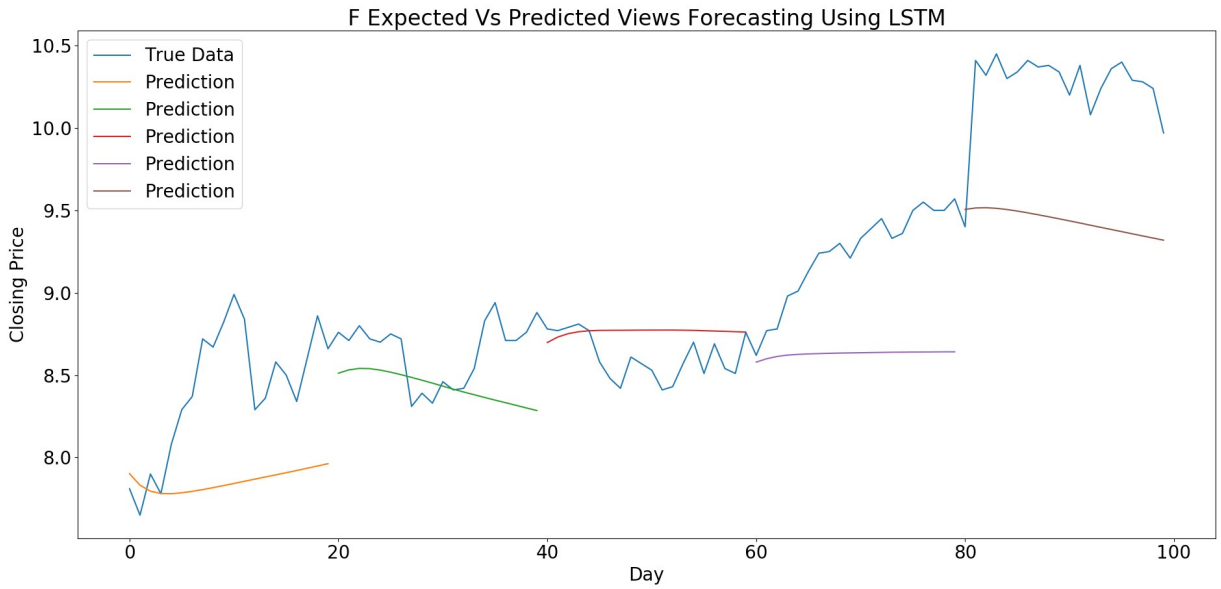


Slika 14: Prikaz stvarnih cijena dionica Bank of America i predviđanja temeljenih na modelu LSTM.

MAPE	4.526437%
RMSE	6.062859
MASE	0.906853

Tablica 7: Pogreške u rezultatima predviđanja cijena dionica Bank of America temeljenih na modelu LSTM.

2. Ford Motor Company (F)

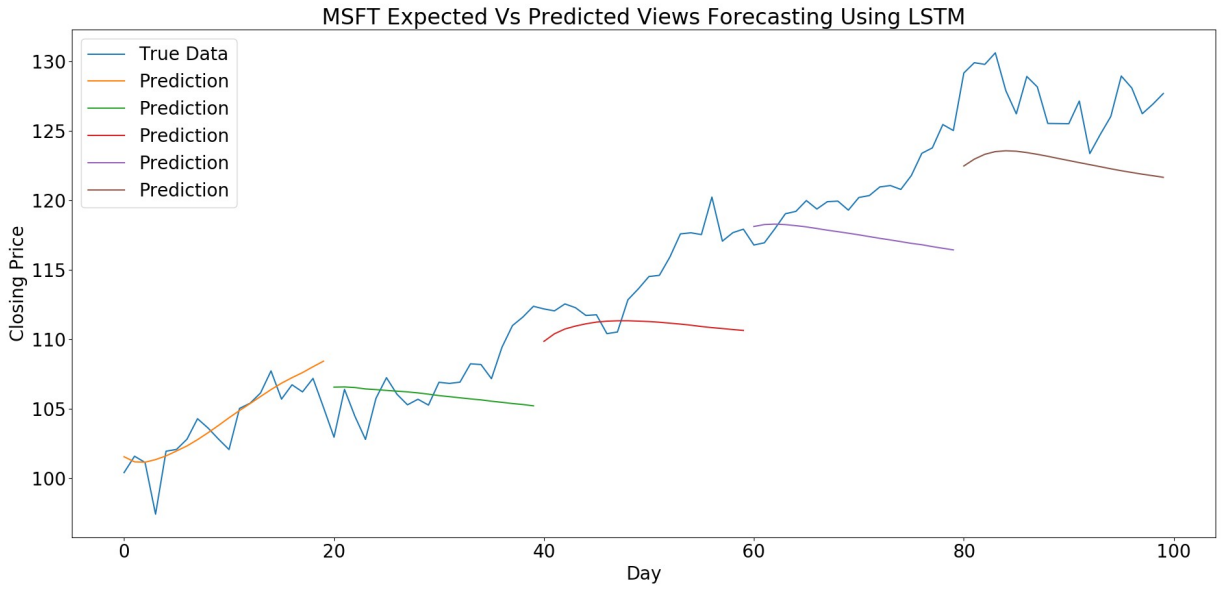


Slika 15: Prikaz stvarnih cijena dionica Forda i predviđanja temelienih na LSTM modelu.

MAPE	5.232005%
RMSE	6.255682
MASE	0.956810

Tablica 8: Pogreške u rezultatima predviđanja cijena dionica Forda temeljenih na modelu LSTM.

3. Microsoft Corporation (MSFT)

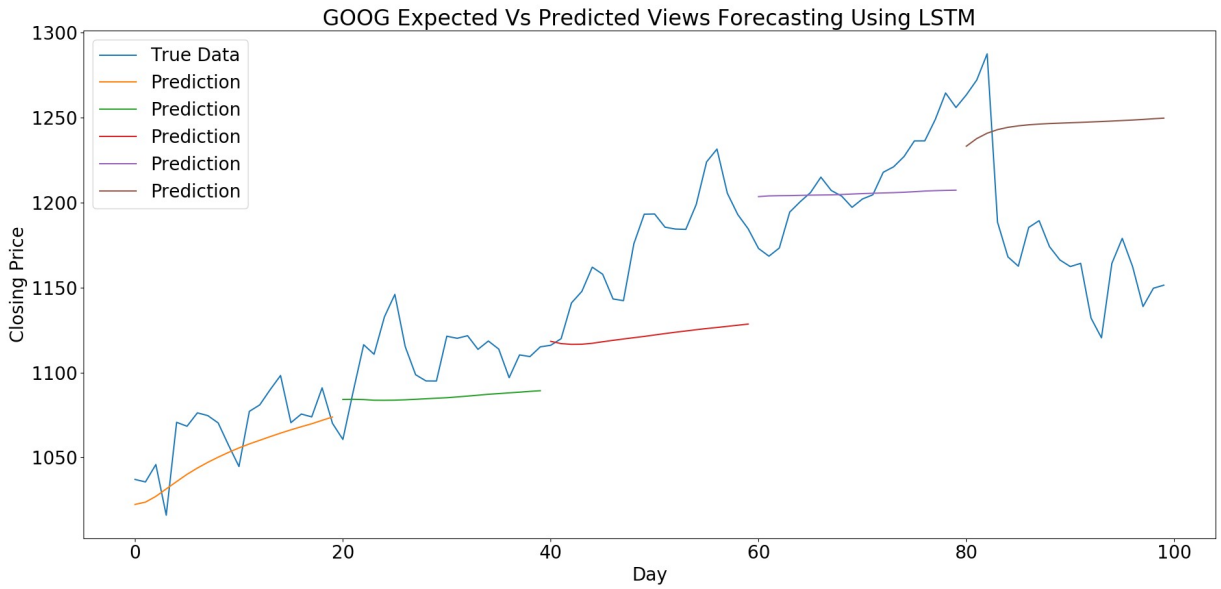


Slika 16: Prikaz stvarnih cijena dionica Microsofta i predviđanja temelienih na modelu LSTM.

MAPE	2.535055%
RMSE	3.236672
MASE	0.867978

Tablica 9: Pogreške u rezultatima predviđanja cijena dionica Microsofta temeljenih na modelu LSTM.

4. Alphabet Inc. (GOOG)

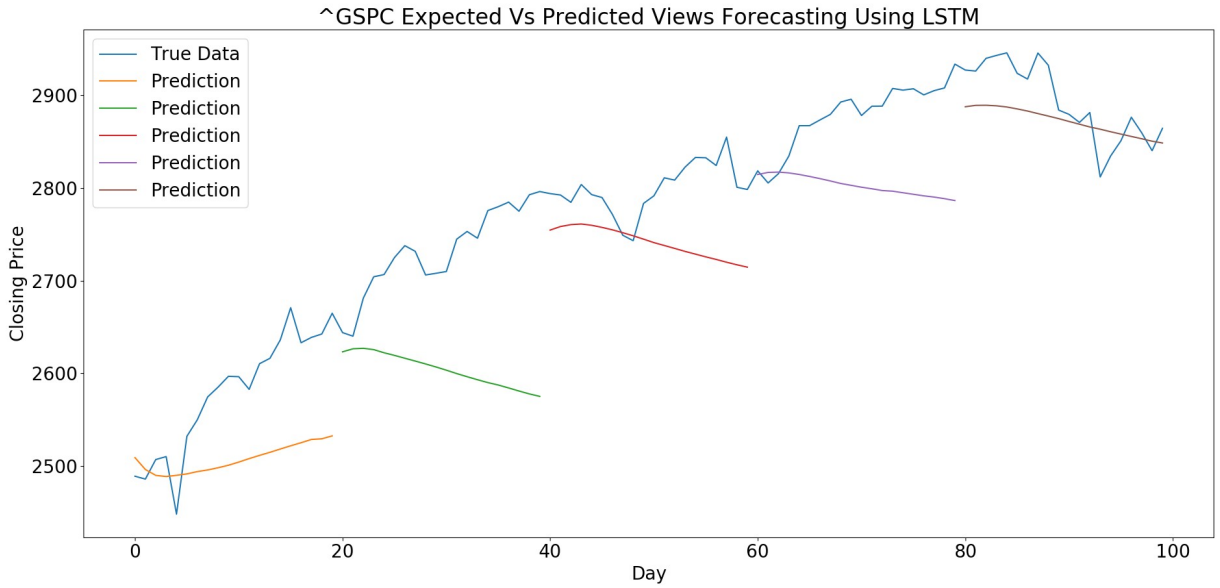


Slika 17: Prikaz stvarnih cijena dionica Gooala i predviđania temelienih na modelu LSTM.

MAPE	3.350211%
RMSE	4.251106
MASE	0.692589

Tablica 10: Pogreške u rezultatima predviđanja cijena dionica Googla temeljenih na modelu LSTM.

5. S&P 500 (^GSPC)

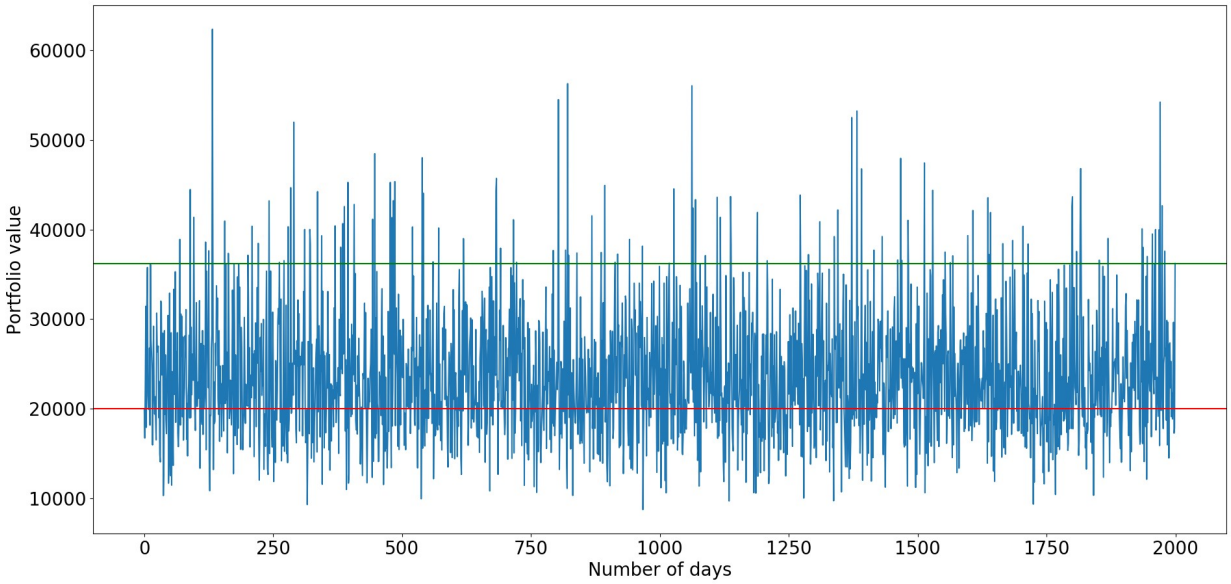


Slika 18: Prikaz stvarnih cijena dionica S&P 500 indeksa i predviđanja temeljenih na modelu LSTM.

MAPE	2.689809%
RMSE	3.307280
MASE	1.157881

Tablica 11: Pogreške u rezultatima predviđanja cijena dionica S&P 500 indeksa temeljenih na modelu LSTM.

6.3. Rezultati Q-učenja



Slika 19: Vrijednost računa agenta Q-učenja.

7. Zaključak

Korištenjem opisanih modela u izgradnji sustava za predviđanje cijena dionica nismo postigli posve zadovoljavajuće rezultate. Razlog tome je ne uzimanje u obzir značajke poput volumena trgovanja, isplate dividenti, vijesti koje se izravno ili neizravno tiču pojedinih dionica, tehničke analize i sl. Prostora za unapređenje, kao što smo naveli, ima jako puno te se nadamo da će ovaj rad poslužiti kao motivacija drugim studentima koji imaju interese za istraživanja na polju računalnih financija. Izrada modela za predviđanja cijena dionica je svakako zahtjevan zadatak ali problemi koji se postavljaju na tom putu pružaju još veću motivaciju u radu, a osjećaj i iskustvo stečeni na tom putu neprocjenjivi su.

8. Literatura

[1] Maslać M., Primjena optimizaciskih postupaka u analizi financijskih vremenskih nizova, završni rad, FER, 2014.

[2] Roman Josue de las Heras Torres, 7 Ways Time Series Forecasting Differs from Machine Learning, 29.5.2018, https://www.datascience.com/blog/time-series-forecasting-machine-learning-differences, 1.6.2019

[3] Siraj Raval, Q Learning for Trading, 25.9.2018, https://www.youtube.com/watch?v=rRssY6FrTvU, 1.6.2019

[4] Siraj Raval, Stock Price Prediction | AI in Finance, 28.4.2018, https://www.youtube.com/watch?v=7vunJlqLZok&feature=youtu.be, 1.6.2019

[5] Jakob Aungiers, TIME SERIES PREDICTION USING LSTM DEEP NEURAL NETWORKS, 1.9.2018, https://www.altumintelligence.com/articles/a/Time-Series-Prediction-Using-LSTM-Deep-Neural-Networks, 1.6.2019

[6] Ayushi Asthana, Bitcoin Price Prediction Using Time Series Forecasting, 27.6.2018, https://towardsdatascience.com/bitcoin-price-prediction-using-time-series-forecasting-9f468f7174d3, 1.6.2019

[7] Nepoznat autor, The Future of Algorithmic Trading, 24.7.2017, https://www.experfy.com/blog/the-future-of-algorithmic-trading, 1.6.2019

[8] Ante Pavić, Postavke ekonomije su psihološki nerealne, 12.11.2008, http://www.poslovni.hr/trzista/postavke-ekonomije-su-psiholoski-nerealne-98383, 1.6.2019

[9] Jason Brownlee, How to Create an ARIMA Model for Time Series Forecasting in Python, 9.1.2017, https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/, 1.6.2019

Sažetak i ključne riječi

Naslov:

Tehnička analiza financijskih podataka s ciljem predviđanja budućih vrijednosti

Sažetak:

U okviru završnog rada ispitana je primjenjivost triju modela u svrhu analize i predviđanja financijskih vremenskih nizova. Dok se prvi model, ARIMA, bazira na jednostavnijim statističkim postupcima, druga dva modela su nešto složeniji, sadržavajući nadzirano odnosno potporno učenje. Analiza je provedena za 5 vrlo aktivnih dionica sa Wall Streeta. Rad sadrži dva računalna modela koji konstruiraju vremenske nizove u trajanju od 20 dana na temelju podataka iz prošlosti. Spomenuti modeli dobivaju informacije na dnevnoj bazi te tako služe za dugoročna predviđanja. Treći model, koji implementira duboko Q-učenje, pokazuje nam kako bi nam računalo kojem predamo neki kapital u određenom okruženju te sa određenim postavkama, moglo stvoriti novčanu dobit. U procesu stvaranja modela rađale su se brojne ideje za unapređenje sustava koje predlažemo kao budući rad.

Ključne riječi:

Financijski vremneski nizovi, računalni modeli, dionice, ARIMA, LSTM, Q-učenje, normalizacija, evaluacija

Summary and keywords

Title:

Predicting future values of financial data using technical analysis

Summary:

This project analyzes the application of the tree models for the purpose of analysis and forecasting financial time series. While the first model, ARIMA, is based on simpler statistical procedures, the other two models are somewhat more complex, containing supervised and reinforcment learning. The analysis was conducted on 5 very active shares of Wall Street. This work contains two computational models that construct 20 day time series based on past data. These models get information on a daily basis and serve as the long-term prediction. The third model, implementing deep Q learning, shows that a computer in a particular environment and with certain settings could generate profit. During the process of the research, we got many ideas for the improvement of the models that could be made in further development.

Keywords:

Financial time series, computer modeling, stocks, ARIMA, LSTM, Q learning, data normalization, evaluation metrics