SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2818

PREVIĐANJE CIJENE KRETANJA FINANCIJSKIH INSTRUMENATA OPTIMIZACIJOM PERFORMANSI METODA STROJNOG UČENJA IZ TOKOVA PODATAKA I METODA DUBOKOG UČENJA

Mislav Križan

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2818

PREVIĐANJE CIJENE KRETANJA FINANCIJSKIH INSTRUMENATA OPTIMIZACIJOM PERFORMANSI METODA STROJNOG UČENJA IZ TOKOVA PODATAKA I METODA DUBOKOG UČENJA

Mislav Križan

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 2818

Pristupnik:	**Mislav Križan (0036507558)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Previđanje cijene kretanja financijskih instrumenata optimizacijom performansi metoda strojnog učenja iz tokova podataka i metoda dubokog učenja**

Opis zadatka:

Klasifikacija smjera kretanja cijene financijskih instrumenata je vrlo izazovno ali jako zahvalno područje u slučaju pronalaska uspješne strategije. U ovom diplomskom radu razmatrat će se optimizacija točnosti predikcije kretanja cijene financijskih instrumenata (npr. cijene dionice na javnim burzama) te kako se performanse klasičnih algoritama klasifikacije tokova podataka odnose s obzirom na modele dubokog učenja. Cilj algoritama je klasifikacija kretanja cijene (rast ili pad) dionica u određenom trenutku s određenim vremenskim periodom razmaka između trenutne i nadolazeće cijene. U tu svrhu, prvotno je potrebno prikupiti podatke o cijenama financijskih instrumenata (npr. cijena dionica) te ih obraditi i očistiti kako bi se nad njima provela dubinska analiza. Temeljem tih podataka, prvo će se optimirati performanse klasičnih algoritama klasifikacije tokova podataka (npr. VFDT stabala), a zatim će se optimirati performanse dubokih modela strojnog učenja (npr. LSTM, višeslojni perceptron, itd.) te će se oni međusobno usporediti. Kao skup podataka koristit će se više različitih vremenskih raspona cijene financijskih instrumenata (npr. 15 sekundi, 1 min, 15 min, 1 sat, itd.). Također, model će razmatrati dodatne informacije o financijskom instrumentu (npr. kod dionica, uključivanje fundamentalnih podataka te dionice). Analiza će se napraviti nad većim brojem financijskih instrumenata (npr. na više različitih dionica). Očekivani rezultat rada će biti vrijednosti performansi pojedinih algoritama kroz više različitih vremenskih raspona i više različitih financijskih instrumenata te njihova međusobna usporedba.

Rok za predaju rada: 27. lipnja 2022.

Sadržaj

Uvod

Klasifikacija smjera kretanja cijene financijskih instrumenata je vrlo izazovno ali jako zahvalno područje u slučaju pronalaska uspješne strategije. U ovom diplomskom radu razmatrat će se optimizacija točnosti predikcije kretanja cijene financijskih instrumenata (npr. cijene dionice na javnim burzama), te kako se performanse klasičnih algoritama klasifikacije tokova podataka odnose s obzirom na modele dubokog učenja.

U prvom dijelu diplomskog rada predstavit će se neke osnovne informacije vezane za financijske instrumente. Kao financijski instrument nad kojim će se napraviti analiza, odabrane su dionice. Dionice su odabrane radi dostupnosti podataka cijene dionice kroz povijest i radi mnogo razvijenih indikatora koji se često koriste u industriji tijekom analize cijene dionice. Jedan dio tih indikatora će također biti uključeni u analizu kao dodatni parametri.

Kao skup podataka koristit će se cijene nekoliko dionica iz različitih industrija. Budući da se cijene dionica generalno prate u više različitih vremenskih koraka, u radu će se razmatrati vremenski koraci duljine od jednog dana pa sve do petnaest-minutnog razmaka. Uz početnu cijenu dionice dodat će se razni indikatori koji se često koriste u industriji kao dodatni parametri tijekom predviđanja smjera kretanja dionice. Kao ciljna klasa bit će rast ili pad cijene sljedećeg vremenskog koraka s obzirom na trenutni.

Nakon prikupljanja navedenih podataka provest će se optimizacija klasičnih algoritama klasifikacije toka podataka te će se izračunati njihove performanse nad nekoliko dionica i nekoliko vremenskih koraka. Zatim će se provesti optimizacija i učenje standardnih modela dubokog učenja i vrijednosti njihovih performansi kako bi se završno mogli usporediti s ostalim algoritmima i modelima.

1. Uvod u financijske instrumente

Financijski instrumenti su imovina kojom se može trgovati, a oni se također mogu gledati i kao paketi kapitala kojima se može trgovati. Ti instrumenti su fizički ili virtualni dokumenti koji reprezentiraju legalni dogovor koji uključuje bilo kakvu monetarnu vrijednost. Postoji više načina podjela financijskih instrumenata. S obzirom na klasu imovine možemo ih podijeliti na instrumente temeljene na kapitalu ili vlasništvu (engl. *equity*) i na instrumente temeljene na dugu (engl. *debt*) [1].

Instrumenti temeljeni na vlasništvu su dionice. Također, u ovu kategoriju se mogu uključiti i dioničke opcije [1].

Instrumenti temeljeni na dugu se mogu dodatno podijeliti na kratkoročne i dugoročne. Kratkoročna zaduženja traju manje od jedne godine dok dugoročna traju duže od jedne godine. Glavni primjer instrumenata temeljenih na dugu su obveznice (engl. *bonds*) i zajmovi (engl. *loans*) [1].

Kao financijski instrument na kojem će se napraviti analiza su odabrane dionice tj. cijene dionica kroz vrijeme.

1.1. Dionice

Dionica je vlasnički vrijednosni papir koji predstavlja vlasništvo određenog dioničkog društva. Vlasnik dionice ima prava na dio imovine dionice kao i dio profita s obzirom koliko dionica posjeduje [5]. Najčešće, korporacije izdaju dionice kako bi prikupile sredstava za vođenje poslovanja. Posjedovanje dionica omogućuje pravo glasa na dioničarskim skupštinama (engl. *shareholder meeting*), primanje dividende te samu mogućnost prodaje tih dionica nekom drugom.

U slučaju da kompanija postane javna kompanija (engl. *public company)* trgovanje njezinim dionicama se obavlja na tržištu dionica (engl. *stock market*). Tržište dionica se odnosi na kolekciju burza (engl. *stock exchange*) gdje se odvija kupovina i prodaja dionica. Ono omogućuje reguliran način sudjelovanja u transakcijama razmjene dionica kompanija.

Tržište dionica omogućuje kupcima i prodavateljima međusobnu interakciju, gdje ta interakcija omogućuje otkrivanje vrijednosti pojedine dionice koja indirektno evaluira i vrijednost cjelokupne kompanije. Zbog velikog broja trgovaca na tržištima dionica (naročito na velikim burzama), kupci i prodavači imaju veće osiguranje da trguju dionicom po pravoj vrijednosti cijene [6]. Neke od najpopularnijih i najvećih burza su NYSE (*New York stock exchange*), LSE (*London stock exchange*) i TSE (*Tokyo stock exchange*). Također postoji i hrvatska burza ZSE (*Zagreb stock exchange*) na kojoj se mogu kupovati dionice hrvatskih kompanija.

Radi toga što cijena dionice vrlo često indirektno reprezentira vrijednost kompanije kojoj pripada, potrebno je spomenuti i temeljne podatke kompanije. Temeljni podaci pojedine dionice se sastoje od informacija koje opisuju financijsko ili ekonomsko stanje kompanije, valute ili bilo kakvog sličnog instrumenta [19]. Temeljna analiza kompanija najčešće sagledava prihode, rashode, stanje imovine i ostalo što se može pronaći na bilanci i drugim izvješćima kako bi se dobila bolja slika o intrinzičnoj vrijednosti kompanije naspram vrijednosti temeljem cijene dionice te kompanije. Javne kompanije koje trguju na burzama imaju obavezu informirati investitore o stanju kompanije, te se radi toga objavljuju informacije o bilanci (engl. *balance sheet)*, financijsko izvješće (engl. *financial statement*), novčani tijek (engl. *cash flow*). U budućem istraživanju bilo bi korisno ubaciti ove podatke u analizu radi moguće koristi kod predviđanja cijene dionice s malo dužim vremenskim korakom predviđanja. Također, cijena dionice je često pod utjecajem ekonomskog okruženja industrije ili čak države u kojem posluje, gdje pozitivno okruženje promovira rast dionice, dok negativno promovira pad.

1.2. Ulaganje u dionice

Postoje mnogi sudionici na tržištu dionica. Najčešće, ulagači trguju dionicama na burzi putem brokera. Brokeri su licencirani predstavnici ulagača za kupuju i prodaju dionice (vrijednosnice) u ime ulagača.

Postoji više načina kako uložiti kapital u dionicu. Najosnovniji način ulaganja kapitala je kupiti dionicu s ciljem da cijena dionice raste. Unutar industrije ta pozicija kapitala s očekivanjem rasta cijene se naziva duga pozicija (engl. *long*). Uz dugu poziciju ponekad postoji i mogućnost kratke pozicije (engl. *short*). Najuobičajenija pozicija je pozicija *long* , kada kupujemo neku dionicu, postajemo vlasnici vrijednosnih papira. Međutim u nekim tržištima postoji i mogućnost pozicije *short*.

Kada ulagač ima poziciju *long* u nekoj dionici to najčešće označava da je on kupio dionice (engl. *shares*) određene kompanije i time postao vlasnik tih vrijednosnih papira. Takav tip pozicije omogućuje ulagaču da poveća vrijednost kapitala temeljem rasta cijene. Kratka (engl. *short*) pozicija omogućuje ulagaču mogućnost zarade temeljem pada cijene dionice. U poziciji *short* ulagač posuđuje dionice od brokera pri trenutnoj cijeni tih dionica, koje pri zatvaranju pozicije kupuje nazad kako bi ih vratio broker i time zaradio na razlici cijena [2]. Radi tih dviju vrsta pozicija moguće je razviti strategije temeljene na predviđanju rasta i pada cijene dionice zbog čega je korisna točnost klasifikacije ne samo rasta nego i pada. Treba naglasiti da je pozicija *short* ovisna o samom brokeru i kompaniji koju je cilj *shortati.*

Primjer zarade putem pozicije *long*:

1. Kupiti 1 dionicu po cijeni od 100 $
2. Prodati za 101 $
3. Zarada od 1 $.

Primjer zarade putem pozicije *short*:

1. Posuditi 1 dionicu od brokera, te je prodati po tadašnjoj cijeni od 100 $.
2. Kupiti nazad tu dionicu po trenutnoj cijeni od 99 $.
3. Vratiti kupljenu dionicu nazad brokeru.
4. Zaraditi 1 $ na razlici cijena posudbe i vraćanja.

Ulaganje s obzirom na vremenski rok ulaganja možemo podijeliti na nekoliko vrsta:

- Dugoročno ulaganje – ulaganje u kompaniju na duži vremenski rok, od nekoliko mjeseci do nekoliko godina.
- Kratkoročno ulaganje – ulaganje u kompaniju na kraći vremenski rok, od nekoliko minuta do nekoliko dana. Radi kratkog roka ulaganja postoji veći potencijal zarade ali isto tako nosi i veći rizik.

Unutar ovog rada glavni fokus će biti na kratkoročno ulaganje zbog cilja predviđanja cijene u udaljenosti maksimalno jednog dana.

U slučaju da se provedba strategije mora izvoditi u malom vremenskom roku, preko više financijskih instrumenata ne bi bilo povoljno da to radi čovjek, zbog čega treba spomenuti algoritamsko trgovanje. Algoritamsko trgovanje je način trgovanja gdje se zahtjevi za kupnju i prodaju dionice generiraju temeljem isprogramiranog algoritma koji donosi odluku o kupnji i prodaji temeljem podataka o cijeni, volumenu i ostalim indikatorima. Većina današnjeg algoritamskog trgovanja spada pod trgovanje s visokom frekvencijom (engl. *high frequency trading*), gdje algoritam postavlja mnogo zahtjeva kupnje ili prodaje na više burza i dionica u malom vremenskom periodu [22].

Neke prednosti algoritamskog trgovanja:

- Zahtjevi za kupnju ili prodaju dionice se često obavljaju na najboljoj mogućoj cijeni naspram ručnog postavljanja zahtjeva.
- Veća šansa egzekucije na željenoj cijeni.
- Vrlo često manji troškovi obavljanja transakcije (ovisno o brokeru).
- Manja šansa grešaka postavljanju zahtjeva u usporedbi s ljudskim ulagačem koji mogu biti pod emocionalnim i ostalim psihološkim utjecajima.

2. Priprema podataka

Kako bi se mogla provesti što bolja klasifikacija cijene dionica, prvo je potrebno pronaći kvalitetni izvor podataka o cijenama dionica kako bi mogli naučiti modele i algoritme na prijašnjim vrijednostima cijene. Vrlo često za kvalitetnije predviđanje kretanja cijene dionice ulagači koriste dodatne parametre kod predikcije. To mogu biti neki vanjski parametri ili derivacije osnovnih parametara kako bi se poboljšala rezolucija pojedinog uzorka u prošlost. Na povećanje rezolucije se misli u slučaju ako se neki parametar računa putem prošlih nekoliko vrijednosti cijene, on daje veću rezoluciju jer uzima u obzir i prijašnje vrijednosti cijene u svoju kalkulaciju. Također, potrebno je i razmotriti kvalitetan način skaliranja podataka radi problema konstantnog rasta maksimuma vrijednosti cijene. U slučaju da se želi raditi klasifikacija putem regresije dolazi do problema radi toga što često maksimalna vrijednost predikcije buduće cijene ne može biti veća od neke prije viđene (u slučaju ako su podaci skalirani, model će rijetko dati vrijednost veću od 1 što može stvarati problem kod predviđanja rasta cijene).

2.1. Vrste podataka

Glavni podaci koji su potrebni za provedbu analize su vrijednosti cijene pojedine dionice kroz vrijeme. Podaci o cijeni dionice su najčešće dostupni od postojanja pojedine kompanije na burzi do današnjeg trenutka. Cijena dionice se često prati u različitim vremenskim intervalima. Neki od najčešćih intervala su intervali od 1, 5, 15 i 30 minuta, te interval od 1 sat i 1 dan. U ovom radu najviše će se razmatrati intervali od 15 i 30 minuta te intervali od jedan sat i jedan dan. Podaci za svaki interval dolaze u 5 vrijednosti: *datetime, open, high, low, close i volume*.

- *Datetime* – datum i vrijeme koji označuju kada su prikupljeni podaci i određuju početak intervala.
- *Open* – početna cijena tog intervala.
- *High* – najviša dostignuta cijena unutar tog intervala.
- *Low* – najniža dostignuta cijena unutar tog intervala.
- *Close* – završna cijena tog intervala.

- *Volume* – ukupni broj dionica (engl. *shares)* kojima se trgovalo unutar tog intervala.

Cilj klasifikacije je odrediti vrijednost buduće cijene sljedećeg intervala naspram trenutnog intervala, koji se računa putem izraza (1). Pozitivna klasifikacija će označavati slučaj u kojemu je *close* vrijednost sljedećeg intervala veća od *close* vrijednosti trenutnog intervala, dok će negativna klasifikacija označavati slučaj u kojemu je *close* vrijednost sljedećeg intervala manja ili jednaka *close* vrijednosti trenutnog intervala.

$$f(t) = \begin{cases} 1, & close(t+1) > close(t) \\ 0, & close(t+1) \leq close(t) \end{cases} \quad (1)$$

$$t - interval$$

Podaci cijene dionice su dobiveni preko API-ja stranice TwelveData (https://twelvedata.com/). Stranica omogućuje preuzimanje podataka većine američkih kompanija s burza NYSE i NASDAQ. Podaci za jednodnevne vrijednosti se mogu preuzeti u potpunosti od datuma postojanja dionica kompanije na burzi, dok podaci vrijednosti intervala cijene unutar dana (5 min, 15 min...) se mogu preuzeti u rasponu do dvije godine od današnjeg datuma. Primjer podataka za dionicu Googl prikazan je u tablici 1.

Tablica 1. Primjer podataka cijene dionice Googl s intervalom jednog dana

datetime	open	high	low	close	volume
19-08-04	50.05005	52.08208	48.02803	50.22022	44659096
20-08-04	50.55556	54.5946	50.3003	54.20921	22834343
23-08-04	55.43043	56.7968	54.57958	54.75475	18256126
24-08-04	55.67567	55.85586	51.83684	52.48749	15247337
25-08-04	52.53253	54.05405	51.99199	53.05305	9188602
26-08-04	52.52753	54.02903	52.38238	54.00901	7094898
27-08-04	54.1041	54.36436	52.8979	53.12813	6211782
30-08-04	52.69269	52.7978	51.05606	51.05606	5196798
31-08-04	51.2012	51.90691	51.13113	51.23624	4917877
01-09-04	51.4014	51.53654	49.88488	50.17517	9138253

2.2. Skaliranje podataka

Kod skaliranja podataka postoji mnogo problema. Jedan od problema je konstantan rast maksimalne vrijednosti dionice što može uzrokovati lošu klasifikaciju, jer prije nije viđena ta vrijednost. Drugi problem je pitanje je li potrebno uzimati sve prijašnje podatke za skaliranje. Možda je potrebno raditi skaliranje putem pomičnog prozora. Unutar ovog rada razmatrat će se klasifikacija bez skaliranja, i skaliranje s pomičnim prozorom određene veličine. Skaliranje će se provoditi putem normalizacije min-max (2).

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

2.3. Tehnički indikatori

Uz osnovne informacije kretanja cijene unutar intervala, dodat će se neki tehnički indikatori koji su često korišteni u industriji tijekom predviđanja smjera cijene. Tehnički indikatori su signali ili heuristike koji su dobiveni od osnovnih informacija cijene i volumena određene dionice. Te indikatore koriste ulagači koji se bave tehničkom analizom financijskih instrumenata. Analizirajući povijesne podatke cijene, indikatori mogu poslužiti u predviđanju buduće cijene [8]. Postoji mnogo tipova ovih indikatora, u ovom radu će biti razmatrani neki od najpopularnijih, s ciljem da svaki od njih ima drugu ulogu, to jest da daje drugačiji pogled na cijenu.

Odabrani indikatori:

- *Simple moving average (SMA)* – indikator koji se računa putem prosječne vrijednosti cijene dionice. Indikator može poslužiti u određivanju hoće li instrument nastaviti trenutni smjer kretanja ili ga promijeniti. Vrlo često se koristi kombinacija dužeg vremenskog intervala i kraćeg, gdje se u slučaju križanja javlja signal. Unutar rada koristit će se SMA 20 koji se računa putem formule (3) kao prosječna cijena (*close* vrijednost) od zadnjih 20 vremenskih intervala [9].

$$SMA\ (n) = \frac{\sum x_i}{n} \tag{3}$$

- *Exponential moving average (EMA)* – indikator sličan indikatoru SMA s glavnom razlikom da indikator EMA stavlja veću vrijednost na recentnije podatke nego indikator SMA. Unutar rada koristit će se EMA 20, koji računa EMA od cijene (vrijednost *close*) zadnjih 20 vremenskih intervala [10].
- *Bollinger Band*® - indikator koji se sastoji od gornje i donje granice. Granice se računaju tako da se indikatoru SMA zbroje dvije standardne devijacije cijene u pozitivnom i negativnom smjeru. Tradicionalno se smatra da bliža vrijednost cijene gornjoj granici može označavati da je instrument prekupljen ili ako je cijena bliža donjoj granici da je instrument preprodan. Unutar rada koristit će se BOLU (engl. *Bollinger Band Up)* i BOLD (engl. *Bollinger Band Down)* koje se računaju zbrajanjem/oduzimanjem dviju standardnih devijacija na indikator SMA 20 u periodu od 20 intervala [11].
- *Relative strength index (RSI)* – indikator momenta koji označava magnitudu recentne promjene cijene. RSI je prikazan kao oscilator između najveće i najniže vrijednosti (0-100). Jedan je od popularnijih indikatora momenta, gdje tradicionalno vrijednost ispod 30 često označava da je instrument podcijenjen te bi moglo doći do promjene kretanja cijene, isto tako u slučaju da je vrijednosti iznad 70 može označavati da je instrument precijenjen. U ovom radu koristit će se RSI 14 koji se računa od zadnjih 14 intervala [12].
- *Stochastic Oscillator* – indikator momenta koji uspoređuje cijenu zatvaranja (engl. *close*) s rasponom vrijednosti cijene u odabranom periodu. Slično kao indikator RSI, služi za generiranje signala precijenjenosti i podcijenjenosti, također je u rasponu 0 do 100. U ovom radu će se koristiti RSI 14 koji se računa od zadnjih 14 intervala [13].
- *Moving Average Convergence Divergence (MACD)* – indikator momenta koji računa odnos između dvije vrijednosti EMA. Najpopularniji način računanja vrijednosti MACD je oduzimajući EMA 26 od EMA 12. Nakon toga se često računa 9-vrijednosni EMA od same vrijednosti MACD(12,26) kao signalna linija. Vrijednost MACD se može gledati kao signal kada se signalna linija prekriži s vrijednosti MACD(12, 26) [14].
- *On Balance Volume (OBV)* – indikator koji označava moment volumena kao način predviđanja promjene cijene instrumenta. Sama vrijednost indikatora nije bitna, jer

je vrijednost drugačija ovisno o tome kada se počela računati, bitnija je promjena indikatora OBV kroz vrijeme [15].

- *Accumulation/Distribution (A/D)* – indikator koji koristi volumen kako bi pokazao da li se instrument akumulira ili distribuira. Koristi se zajedno s cijenom instrumenta. Tradicionalno, rast vrijednosti indikatora A/D može služiti kao potvrda trenda rasta cijene, dok pad indikatora može potvrditi trend pada cijene [16].
- *Average Directional Index (ADX)* – indikator koji služi kao mjera snage trenda kretanja cijene. ADX se sastoji od pozitivnog i negativnog indikatora smjera. Često kada je vrijednost indikatora iznad 25 to označava jaki trend, dok vrijednost ispod 20 označava slabi. U ovom radu koristit će se ADX 14 koji se računa od zadnjih 14 intervala [17].
- *Aroon oscillator* – indikator koji pokušava odrediti snagu trenutnog trenda cijene instrumenta. Računa se preko indikatora *Aroon Up* i *Aroon down*. Oni se računaju putem broja perioda (unutar određenog intervala) od prijašnje najviše ili najniže vrijednosti [18]. U ovom radu koristit će se *Aroon* 14 koji se računa od zadnjih 14 intervala.

2.4. Implementacija

Implementacija je rađena u programskom jeziku Python s dodatnim knjižicama. Za učitavanje i manipulaciju podataka korištena je knjižnica *pandas*. Unutar knjižice *scikit-multiflow* su implementirani razni algoritmi klasifikacije tokova podataka kao i razni detektori promjene u distribuciji podataka. Za duboke modele korištena je knjižica *keras*. Kalkulacija tehničkih indikatora je samostalno implementirana.

Implementacija se sastoji od preuzimanja podataka, dodavanja dodatnih indikatora, te same klasifikacije putem algoritama toka podataka ili dubokih modela.

2.4.1. Implementacija preuzimanja podataka

Za preuzimanje podataka koristi se API dostupan od strane *TwelveData*. Prije preuzimanja podataka, unutar koda se postavljaju željeni parametri koji određuju dionicu i vremenski korak za koji će podaci biti preuzeti (Kod 2.1 – Parametri za preuzimanja podataka). Parametar `symbol` određuje simbol dionice za koju preuzimamo podatke (npr. GOOGL, TSLA...). Parametar `interval` određuje željeni vremenski korak za koji će se preuzeti podaci (npr. 5 min, 15 min, 1 h, 1 day...). Parametri `start_date` i `end_date` određuju početni i krajnji datum raspona za koje će se preuzeti podaci. Parametar `apikey` je potreban kako bi se moglo pristupiti API-ju stranice *TwelveData.*

```
parameters = {}

parameters["symbol"] = symbol

parameters["interval"] = interval

parameters["apikey"] = apikey

parameters["start_date"] = start_date

parameters["end_date"] = end_date
```

Kod 2.1 – Parametri za preuzimanja podataka

Zatim se preuzimaju podaci putem API zahtjeva Pythonove knjižnice *request*. Preuzeti podaci (vrijednosti *open, close, high, low, volume*) se nalaze unutar varijable `data` te se ti podaci spremaju na računalo (Kod 2.2 – Program preuzimanja podataka).

```
response =
requests.get("https://api.twelvedata.com/time_series",pa
rams=parameters)

print("STATUS: ", response.status_code)

data = response.json()

data = data["values"]

...
```

```
df.to_csv(filePath + symbol + "/" + symbol + "_" +
interval + "RAW.csv", index=False)
```

Kod 2.2 – Program preuzimanja podataka

2.4.2. Implementacija tehničkih indikatora

Kako bi se povećala šansa točnog predviđanja, dodani su tehnički indikatori iz drugog poglavlja. Tehnički indikatori se računaju putem prije preuzetih osnovnih podataka pojedine dionice.

Dodavanje tehničkih indikatora na osnovne podatke implementirano je u funkciji `addTechnicalIndicators()`(Kod 2.3 – Funkcija `addTechnicalIndicators()`).

```
def addTechnicalIndicators():
    df = pd.read_csv(filePath + symbol + "/" + symbol +
    "_" + interval + "RAW.csv",index_col=False)

    #Simple moving average
    SMA(df,period=20)



    #exponential moving average
    EMA(df,period=20)


    #BollingerBands
    bollingerBands(df, period=20, m=2)
    ...
```

Kod 2.3 – Funkcija `addTechnicalIndicators()`

Primjer računanja indikatora SMA (*simple moving average*) (Kod 2.4 – Funkcija izračuna tehničkod indikatora SMA).

```
def SMA(df, period = 20):
    df["SMA"+str(period)]=df['close'].rolling(period)
    .mean()
```

Kod 2.4 – Funkcija izračuna tehničkod indikatora SMA

Kao krajnja vrijednost koja se dodaje na preuzete podatke je vrijednost labele klase podataka. Vrijednosti klase se određuju kako je prije navedeno, putem uspoređivanja vrijednosti *close* trenutne i buduće cijene.

```
if(df.iloc[index+1]["close"]>df.iloc[index]["close"]):
    newVal = 1
```

3. Klasični algoritmi tokova podataka

Tok podataka se može definirati kao predan slijed podataka koji dolazi kroz vrijeme, radi toga cijenu dionice možemo sagledati kao tok podataka. Postoji generalna podjela tokova podataka na stacionarne i nestacionarne podatke. Glavno obilježje stacionarnih podataka je da se distribucija koja ih generira ne mijenja ili jako malo mijenja s obzirom na vrijeme, to jest da postoji jako mala šansa za pomak koncepta (engl. *concept drift*). Kod nestacionarnih podataka distribucija iz koje se generiraju podaci se mijenja. Najčešće, cijena dionica se smatra nestacionarnim oblikom toka podataka zbog konstantno evoluirajuće distribucije koja ovisi o mnogo vanjskih faktora. Radi toga pri izboru tradicionalnih algoritama, više će biti razmatrani oni s mogućnosti prilagodbe na pomak koncepta.

Postoji dva glavna pristupa rješavanju pomaka koncepta: evoluirajući i adaptivni modeli. Evoluirajući modeli ažuriraju parametre u regularnim intervalima bez da gledaju da li je došlo do kakvih promjena u distribuciji toka podataka. Kako bi riješili problem pomaka koncepta, evoluirajući modeli najčešće koriste pomični prozor (engl. *sliding window*) s pretpostavkom da su novi podaci više reprezentativni za predviđanje nego stariji. Glavna mana takvog pristupa je što se promjene događaju u regularnim intervalima i time se parametri koji su prije bili aktualni mijenjaju neovisno o tome je li došlo do promjene u podacima. Takav pristup može dovesti do lošijih rezultata ako su ti intervali promjene loše podešeni. Takvi modeli su najuspješniji kada su pomaci koncepta postupni i inkrementalni.

S druge strane, adaptivni modeli eksplicitno rade detekciju promjene tokova podataka, to jest detektiraju je li došlo do pomaka koncepta i u slučaju da je, ažuriraju model sa svježim podacima. Glavna prednost takvog pristupa je što su ažuriranja modela puno rjeđa naspram evoluirajućih modela i time je sveukupno manje vrijeme ažuriranja modela. Izazov takvog pristupa je kvalitetno napraviti detekciju pomaka kako bi uspješnost modela bila konzistentna.

Neki od dostupnih detektora pomaka koncepta u knjižici *scikit-multiflow* su:

- ADWIN (*Adaptive sliding window)* – detektor pomaka koncepta koji određuje veličinu prozora koju algoritam uzima u obzir i time je algoritam više otporniji na pomak koncepta. Veličina prozora je odlučena putem kalkulacije statistike kroz više potprozora gdje u slučaju velike razlike u prozorima, detektira se promjena te se uzima ažurniji prozor [23].

- DDM (*Drift detection method)* – detektor pomaka koji prati količinu pogrešaka klasifikatora s pretpostavkom da u slučaju ako je distribucija podataka stacionarna da količina pogrešaka bi se trebala smanjivati kroz vrijeme. U slučaju da distribucija nije stacionarna, količina pogrešaka bi trebala rasti. Kada količina pogrešaka pređe izračunatu granicu, algoritam će najprije upozoriti da je mogući dolazak promjene u distribuciji a ako prijeđe još višu granicu, bit će detektiran pomak koncepta.

U ovom radu pretežno će se koristiti detektor pomaka koncepta ADWIN kod adaptivnih algoritama.

3.1. Korišteni algoritmi

Knjižica *scikit-multiflow* nudi mnogo različitih algoritama klasifikacije toka podataka. U ovom radu razmatrat će se:

- *HoeffdingTreeClassifier* – VFDT (engl. *very fast decision tree*) algoritam baziran na Hoeffdingovoj granici koji gradi stablo temeljem podataka, gdje se samo pamte određene statistike tijekom grananja stabla. Radi toga, jako je koristan u slučaju velikog broja podataka gdje ih svede na relativno malo stablo. Velika prednost Hoeffdingovog stabla kod klasifikacije toka podataka (radi inkrementalnog učenja) je što ima jamstvo uspješnosti jer koristi Hoeffdingovu granicu koja pokazuje da na

dovoljno velikom skupu podataka rezultat gotovo identičan rezultatu bez inkrementalnog učenja.

- *HoeffdingAdaptiveTreeClassifier* – Nadogradnja algoritma *HoeffdingTreeClassifier* gdje se dodatno koristi algoritam ADWIN koji prati uspješnost pojedinih grana. U slučaju da se uspješnost grane smanji, bit će zamijenjena s novom granom koja ima veću uspješnost i time čini algoritam otpornijim na pomak koncepta.
- *KNNADWINClassifier* – algoritam koji određuje klasu putem računanja najbližih susjeda unutar svog prozora, te s obzirom na klasu najbližih susjeda određuje klasifikaciju traženog podatka. Najbliži susjedi se računaju putem parametara podataka. Postoje razne mjere računanja udaljenosti dvaju podataka. Većina uobičajenih mjera se može predstaviti formulom Minkowskijeve udaljenosti $D(X,Y) = (\sum_{i=0}^{n}|x_i - y_i|^p)^{1/p}$ gdje vrijednost u slučaju $p = 1$ odgovara udaljenosti Manhattan, dok $p = 2$ odgovara vrijednosti euklidske udaljenosti. Glavna nadogradnja naspram regularnog algoritma KNN je dodatak detekcije pomaka koncepta. *KNNADWINClassifier* za detekciju pomaka koncepta koristi algoritam detekcije promjene ADWIN, te on ažurira prozor s primjerima koje će algoritam uspoređivati kod pretraživanja najbližih susjeda i tako povećava otpornost na promjenu u distribuciji.

3.2. Implementacija

Za klasifikaciju putem algoritama toka podataka, prvotno se učitava datoteka s prije izračunatim vrijednostima tehničkih indikatora i vrijednostima klase pojedinog primjera. Kako bi se cijena dionica reprezentirala kao tok podataka, datoteka se učitava putem klase *FileStream* iz knjižice *scikit-multiflow* (Kod 3.1 – Učitavanje datoteke u klasu). U realnoj primjeni bi se vrijednosti indikatora morale računati pri svakom dolasku podataka, što u slučaju vremenskih razmaka većih od 5 minuta ne bi trebalo predstavljati problem.

```
symbol = "GOOGL"

interval = "15min"
```

```
filePath = folderPath + symbol + "/" + symbol + "_" +
interval + "_TI_CLASS.csv"

stream = mf.data.FileStream(filePath)
```

Kod 3.1 – Učitavanje datoteke u klasu *FileStream*

Nakon učitavanja datoteke u klasu *FileStream*, odabire se željeni algoritam s određenim parametrima s kojim će se provoditi klasifikacija, npr. `KNNADWINClassifier` iz knjižnice *scikit-multiflow.lazy* (Kod 3.2 – Odabir algoritma za klasifikaciju).

```
classificator=
mf.lazy.KNNADWINClassifier(n_neighbors=5,max_window_size
=300)
```

Kod 3.2 – Odabir algoritma za klasifikaciju

Zatim se radi predučenje klasifikatora s određenim brojem primjera, te se izvodi glavna petlja koja reprezentira tok podataka dolaskom pojedinog podatka na klasifikaciju. Nakon klasifikacije, u algoritam se dodaje pridošli podatak kako bi algoritam sadržavao što svježije informacije. Unutar glavne petlje se računa točna klasifikacija odabranog algoritma kako bi se na kraju algoritmi mogli međusobno usporediti (Kod 3.3 – Glavna petlja klasifikacije podataka algoritmima klasifikacije toka podataka).

```
x,y = stream.next_sample(100)

classificator.partial_fit(x,y,classes=[0,1])


while stream.has_more_samples():
     x,y = stream.next_sample()
     pred = classificator.predict(x)
     classificator.partial_fit(x,y)
```

Kod 3.3 – Glavna petlja klasifikacije podataka algoritmima klasifikacije toka podataka

Također, postoji mogućnost skaliranja putem pomičnog prozora. Skaliranje putem pomičnog prozora se radi kroz program *scalerDataWindow.py*. Program skalira podatke s

određenom veličinom pomičnog prozora, te se skalirani podaci spremaju u datoteku koja se zatim može učitati za klasifikaciju na isti način kao datoteka s neskaliranim podacima.

3.3. Rezultati

Analiza je napravljena na podacima triju dionica unutar različitih industrija. To su dionice američkih kompanija *Alphabet Inc* (GOOGL), *Tesla Inc* (TSLA) te *Darling Ingredients Inc* (DAR). Razmatrali su se podaci bez skaliranja i podaci skalirani s pomičnim prozorom veličine 250. Unutar tablica 2 – 5 prikazani su rezultati dobiveni klasifikacijom na podacima skaliranim s pomičnim prozorom veličine 250. Rezultati su dobiveni klasifikacijom 1000 nadolazećih vrijednosti te kako bi algoritam imao najažurnije podatke tijekom klasifikacije pojedinog primjera, svaki primjer bi bio dodan u znanje algoritma nakon što bi bio klasificiran. Uz osnovne informacije cijene, za klasifikaciju su se koristili i tehnički indikatori. Bazna točnost je dobivena putem modela koji klasificira svaki primjer kao pozitivnu klasu.

Radi međusobne usporedbe algoritama unutar rada, algoritmi su optimirani na dionicu DAR te su se primijenili na druge dionice s istim parametrima.

Rezultati pokazuju da u slučaju klasifikacije svakog nadolazećeg podatka nije optimalno primijeniti jedan algoritam nad svim dionicama i vremenskim koracima. Za najbolje rezultate potrebno je raditi individualnu optimizaciju s obzirom na dionicu i vremenski korak jer svaka dionica ima drugačije ponašanje radi drugačije valuacije i industrije u kojoj operira.

Međutim, najbolji rezultati su se pokazali kada se radila klasifikacija na onim primjerima koji imaju veliku sigurnost predikcije. U ovom slučaju kada je sigurnost predikcije bila veća od 0.7 za pozitivne ili manja od 0.3 za negativne, primjer bi se uzeo u rezultate (Tablica 5 – Neki od boljih rezultata tijekom klasifikacije primjera s velikom sigurnošću (>0.7)). Polje BROJ PODATAKA označava broj klasificiranih primjera kojima je sigurnost predikcije bila veća od 0.7 ili manja od 0.3 od ukupne klasifikacije 1000 nadolazećih primjera.

Tablica 2 – Rezultati tradicionalnih algoritama na podacima dionice GOOGL

GOOGL	BAZNA TOČNOST	Hoeffding AdaptiveTree (grace_period=50)	KNNADWIN (max_window_size=250)	KNNADWIN (n_neighbors=15, max_window_size=250, metric="l1")
1 day	51.45%	48.35%	49.65%	49.85%
1 h	52.15%	52.05%	50.15%	47.75%
30 min	51.95%	51.95%	51.85%	53.15%
15 min	53.25%	53.55%	48.45%	49.05%

Tablica 3 – Rezultati tradicionalnih algoritama na podacima dionice TSLA

TSLA	BAZNA TOČNOST	Hoeffding AdaptiveTree (grace_period=50)	KNNADWIN (max_window_size=250)	KNNADWIN (n_neighbors=15, max_window_size=250, metric="l1")
1 day	49.85%	49.85%	47.95%	49.15%
1 h	53.95%	53.55%	50.65%	49.85%
30 min	50.25%	50.25%	51.55%	49.85%
15 min	52.45%	52.45%	51.05%	49.05%

Tablica 4 – Rezultati tradicionalnih algoritama na podacima dionice DAR

DAR	BAZNA TOČNOST	Hoeffding AdaptiveTree (grace_period=50)	KNNADWIN (max_window_size=250)	KNNADWIN (n_neighbors=15, max_window_size=250, metric="l1")
1 day	54.74%	50.68%	50.38%	52.63%
1 h	54.25%	49.75%	50.05%	50.55%
30 min	52.25%	48.85%	51.35%	51.65%
15 min	48.35%	50.85%	48.05%	48.25%

Tablica 5 – Neki od boljih rezultata tijekom klasifikacije primjera s velikom sigurnošću (>0.7) tradicionalnim algoritmima

DIONICA	SKALIRANJE	INTERVAL	ALGORITAM	TOČNOST	BROJ PODATAKA
GOOGL	WINDOW-250	1 day	HoeffdingAdaptiveTreeClassifier (grace_period=250)	56.45%	62
GOOGL	WINDOW-250	30 min	KNNADWINClassifier (n_neighbors=15,max_window_size=250,metric="l1")	53.70%	108
GOOGL	WINDOW-250	15 min	HoeffdingAdaptiveTreeClassifier (grace_period=100)	55.66%	106
TSLA	WINDOW-250	30 min	KNNADWINClassifier (max_window_size=250)	53.94%	343
TSLA	WINDOW-250	30 min	KNNADWINClassifier (n_neighbors=15,max_window_size=250,metric="l1")	56.38%	94
TSLA	BEZ	1 day	KNNADWINClassifier (n_neighbors=15,max_window_size=250,metric="l1")	55.26%	76
TSLA	BEZ	1 h	KNNADWINClassifier (n_neighbors=15,max_window_size=250,metric="l1")	61.29%	93
TSLA	BEZ	15 min	KNNADWINClassifier (max_window_size=250)	57.20%	264

DAR	WINDOW-250	1 day	KNNADWINClassifier (max_window_size=250)	54.27%	199
DAR	WINDOW-250	1 h	HoeffdingAdaptiveTreeClassifier (grace_period=50)	52.61%	230
DAR	BEZ	30 min	HoeffdingAdaptiveTreeClassifier (grace_period=100)	53.38%	414
DAR	BEZ	15 min	HoeffdingAdaptiveTreeClassifier (grace_period=50)	53.38%	266

4. Modeli dubokog učenja

Uspješnost jednostavnih algoritama strojnog učenja poput logističke regresije i naivnog Bayesovog algoritma je jako ovisna o reprezentaciji podataka. Ako znamo kakve reprezentacije značajki su povezane na željenu vanjsku vrijednost, tada su jednostavni algoritmi strojnog učenja dovoljni. Međutim, u slučaju da ne znamo koje su točno značajke bitne, ili na koji je način potrebno reprezentirati bitne značajke, dolazimo do problema sa standardnim algoritmima strojnog učenja. Duboko učenje stvara jaku strukturu nadziranog učenja. Dodavanjem više slojeva, duboka mreža je sposobna reprezentirati jako kompleksne funkcije, što omogućuje puno kompleksniju reprezentaciju značajki.

4.1. Modeli

Unutar rada koristit će se dva tipa modela dubokog učenja: Duboke unaprijedne mreže i jedan oblik povratnih dubokih mreža.

Duboke unaprijedne mreže (engl. *deep feedforward networks*) – još se nazivaju i višeslojni perceptroni, su najosnovniji modeli dubokog učenja. Glavni cilj unaprijedne mreže je aproksimirati funkciju f, tako da za dani ulaz x daje željeni izlaz y s određenim parametrima θ, $y = f(x; \theta)$. Ime unaprijedna mreža dolazi radi toga što informacija teče unaprijed od ulaznih vrijednosti do izlaznih gdje ćelije nemaju nikakvih povratnih veza. Ime mreža dolazi zbog toga što se modeli mogu reprezentirati kompozicijom više različitih funkcija, gdje svaki sloj predstavlja funkciju koja ulazi u sljedeći sloj $f(x) = f^3\left(f^2\left(f^1(x)\right)\right)$, gdje f^1 označava prvi sloj, f^2drugi sloj i f^3 treći sloj [24].

Često se takve mreže nazivaju i neuronske mreže radi inspiracije neuronima u mozgu. Motivacija tih mreža je proizašla iz neuroznanosti, to jest matematičkim modelima funkcioniranja neurona u mozgu. Svaki neuron u unaprijedno povezanoj mreži prima ulaze od svih prijašnjih neurona te daje izlaz. U slučaju da prima i ulaze od budućih neurona (onih koji se nalaze ispred njega u mreži) onda se ta mreža naziva povratna. Glavne prednosti dubokih modela naspram jednostavnih linearnih modela poput logističke regresije je mogućnost reprezentacije kompleksnih funkcija. Linearni modeli mogu aproksimirati samo

linearne funkcije te ne mogu raditi poveznice interakcije između dviju ulaznih varijabla. Međutim, tijekom učenja, linearni modeli su puno manje zahtjevni od dubokih zbog zatvorene forme i konveksne optimizacije.

Svaki algoritam strojnog učenja se sastoji od tri glavne komponente: modela, funkcije pogreške i optimizacijskog postupka. Glavni problem kod neuronskih mreža je što često funkcija gubitka prestaje biti konveksna što znači da nema rješenje u zatvorenoj formi, te se mora raditi iterativna optimizacija funkcije gubitka, to jest postupno smanjivati pogrešku funkcije gubitka. Gradijentni spust nad ne-konveksnom funkcijom gubitka nema garanciju konvergencije u minimum te je vrlo osjetljiva s obzirom na inicijalne vrijednosti parametara.

Učenje neuronskih mreža je najčešće slično učenju bilo kojeg drugog modela strojnog učenja koji se uči gradijentnim spustom. Neuronske mreže se najčešće uče putem širenjem pogreške unazad (engl. *error backpropagation*). To znači da se računa gradijent funkcije gubitka te se ta pogreška prosljeđuje unazad kroz svaki sloj.

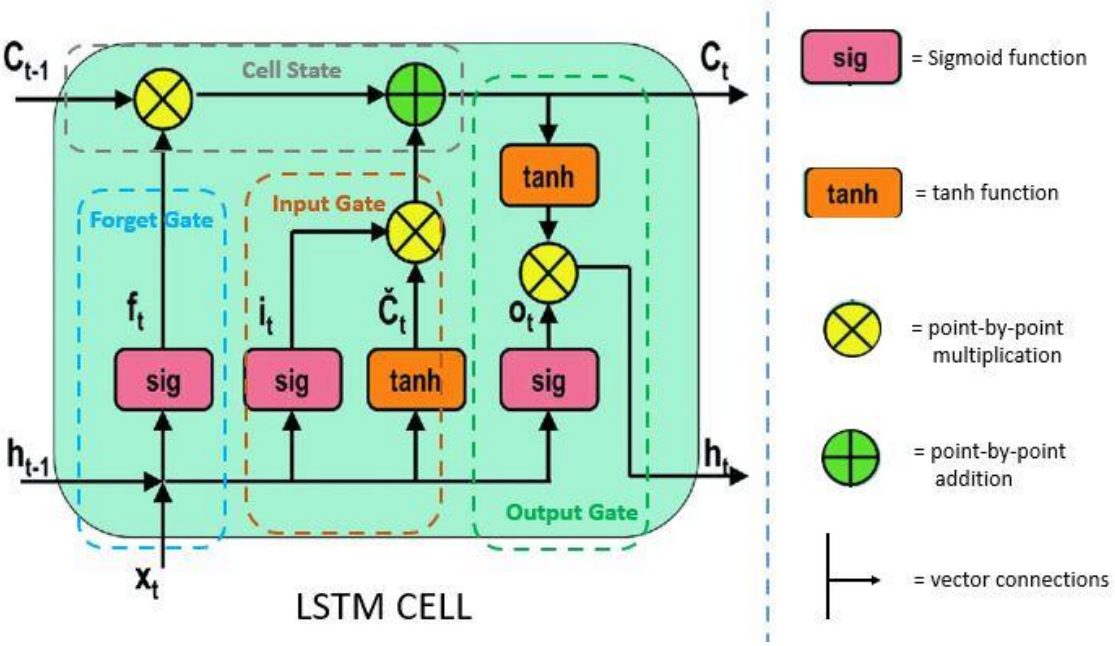
Radi gledanja kretanje cijena instrumenata kao sekvence cijena, vrlo često se koriste modeli dubokog učenja koji primaju kao ulazne vrijednosti sekvencu vrijednosti. Radi povratne ovisnosti pojedinog modela povratne neuronske mreže su specijalizirane za takav tip podataka.

Povratne neuronske mreže (engl. *recurrent neural networks*, RNN) su vrsta neuronskih mreža koje se koriste za obradu slijednih podataka x^0, x^1, x^2 ... Kod slijednih podataka postoji ovisnost budućih podataka o prošlim, a povratne mreže sadrže koncept pamćenja prošlih podataka, to jest one sadrže neki oblik povratne veze između jedinki, što ih razlikuje od standardne unaprijedne mreže [24]. Postoji više načina oblikovanja povratne neuronske mreže:

- Povratne mreže koje daju izlaznu vrijednost nakon svakog vremenskog koraka i imaju povratne veze između skrivenih jedinki.
- Povratne mreže koje daju izlaznu vrijednost nakon svakog vremenskog koraka i imaju vezu samo iz izlaza jednog vremenskog koraka prema skrivenim jedinkama sljedećeg vremenskog koraka.
- Povratne mreže s povratnim vezama između skrivenih jedinki koje čitaju cijelu sekvencu podataka, te generiraju jedan izlaz.

Jedna od vrsti povratne mreže je mreža s dugim kratkoročnim pamćenjem (engl. *long short-term memory,* LSTM).

Mreža s dugim kratkoročnim pamćenjem (engl. *long short-term memory,* LSTM) je vrsta povratnih neuronskih mreža koja ima unutarnje mehanizme koji reguliraju kretanje informacija. Ti mehanizmi se zovu vrata (engl. *gate*). Ćelija LSTM sastoji se od ulaznih vrata (engl. *input gate*), izlaznih vrata (engl. *output gate*) i vrata zaborava (engl. *forget gate)*. Uloga tih mehanizma je da mreža nauči koji dio podataka u sekvenci treba pamtiti, a koji dio zaboraviti [25].



Slika 1. Primjer ćelije LSTM [27]

4.2. Implementacija

Klasifikacija putem dubokih modela se radi preko knjižnice *keras* i *jupyter notebooka*. Prvotno se učitavaju podaci koji se sastoje od osnovnih podataka i tehničkih indikatora. Podaci se učitavaju u *pandas dataframe*.

```
Df = pd.read_csv(pathToData)
```

Radi toga što je učenje dubokih modela puno zahtjevniji proces od klasičnih algoritama, razmatra se mogućnost testiranja na većem broju podataka prije ponovnog učenja. Optimalni

proces bi bio učiti model nakon svakog pridošlog podatka, ali radi dugog vremena učenja radit će se testiranje na više podataka prije ponovnog učenja modela.

Prije početka programa, postavljaju se parametri koji određuju broj iteracija (varijabla `NUM_ITER`) izvršavanja petlje, te pomak podataka nakon svake petlje (varijabla `OFFSET`). Varijabla `OFFSET` istovremeno određuje i količinu podataka koje će model klasificirati prije ponovnog učenja. Varijabla `splitIndexStart` određuje veličinu prozora skaliranja te istovremeno označava količinu podataka na kojima će se model učiti dok varijabla `splitIndex` određuje od kojeg primjera se kreće s klasifikacijom (u slučaju da je potrebno krenuti od kasnijeg primjera u sklopu analize). Također, u slučaju da se radi o povratnom model postoji i parametar `timeStep` koji određuje koliko će se koraka u prošlost gledati pojedini primjer.

```
OFFSET = 10

NUM_ITER = 10

NUM_EPOCHS = 1000

#na koliko krece

splitIndex = 1995

#na koliko prijasnjih primjera se uči

splitIndexStart = 250

#data za lstm

timeStep = 5
```

Unutar glavne petlje, podaci se prvo skaliraju s *MinMaxScalerom* te se podijele na skup za učenje i za testiranje. U slučaju da se koriste povratne mreže, prije podjele na skup učenja i testiranja, putem funkcije *TimeseriesGenerator()* (knjižica *keras.preprocessing.sequence*) generiraju se trodimenzionalni podaci s određenim brojem koraka (engl. *timestep*).

Zatim se definira model, te se kreće s učenjem modela. Nakon završetka radi se klasifikacija skupa za testiranje podataka, nakon čega se cijeli skup podataka pomiče za količinu klasifikacije podataka (varijabla `OFFSET`) (Kod 4.1 – Glavna petlja učenja modela dubokog učenja).

```
scaler = MinMaxScaler()
scaler.fit(df[splitIndex-splitIndexStart:splitIndex])
scaledDf = scaler.transform(df)

#za stvaranje podataka s vremenskim korakom
data_gen=TimeseriesGenerator(X,yShift,length=timeStep,sa
mpling_rate=1,batch_size=10000)

#podjela na skup za učenje i za testiranje
X_train = X_seq[splitIndex-splitIndexStart:splitIndex]
X_test =X_seq[splitIndex:]
y_train = y_seq[splitIndex-splitIndexStart:splitIndex]
y_test =y_seq[splitIndex:]

#definicija modela
model = defineModel()

#učenje
history=model.fit(X_train,y_train,batch_size=batchSize,
epochs=NUM_EPOCHS,verbose=1,shuffle=False)

#predviđanje
saved_model=model
predicted = saved_model.predict(X_test)
```

Kod 4.1 – Glavna petlja učenja modela dubokog učenja

4.3. Rezultati

Analiza se radila na podacima triju dionica unutar različitih industrija. To su dionice američkih kompanija *Alphabet Inc* (GOOGL), *Tesla Inc* (TSLA) te *Darling Ingredients Inc* (DAR). Svi modeli su se učili na osnovnim podacima cijene i tehničkim indikatorima, gdje su podaci skalirani putem pomičnog prozora veličine 250. Modeli su se učili na prijašnjih 250 podataka gdje je broj epoha učenja kod unaprijednih modela bio 500, dok je kod povratnih modela bio 1000 kako bi se izbjegla prenaučenost. Bazna točnost je dobivena putem modela koji klasificira svaki primjer kao pozitivnu klasu.

Rezultati su dobiveni temeljem klasifikacije 100 nadolazećih podataka. Radi dužeg učenja dubokih modela naspram tradicionalnih algoritama, učenje se izvodilo nakon klasifikacije svakih 10 primjera kako bi se povećao broj klasificiranih podataka, a opet održala ažurnost podataka. U realnoj situaciji najbolji način bi bio učiti model nakon svakog pridošlog podatka. Slično kao kod tradicionalnih algoritama klasifikacije, za najbolje rezultate potrebno je optimirati modele za svaku dionicu i vremenski korak zasebno. Radi lakše međusobne usporedbe, modeli su optimirani na dionici DAR te su primijenjeni na sve dionice i vremenske korake s tim parametrima.

Može se vidjeti da je najgora točnost na dionici GOOGLa (39% i 40% točnost modela), što pokazuje da su rezultati dubokih modela jako osjetljivi s obzirom na parametre modela (broj epoha učenja, stopa učenja...) gdje u slučaju primjene modela s parametrima optimiranih na drugoj dionici može dovest do lošijih rezultata (moguće da je došlo do prenaučenosti, podnaučenosti ili čak da je model bio premalog kapaciteta). S druge strane, najbolji rezultati se pokazuju na dionici TSLA (61% i 56% točnost modela) unatoč tome da se optimizacija parametara vršila na dionici DAR, što se može objasniti time da se model uspio dovoljno naučiti na podatke a opet da se nije prenaučio.

Međutim, slično kao i kod tradicionalnih algoritma, najbolji rezultati su se pokazali kada se radila klasifikacija na onim primjerima koji imaju veliku sigurnost predikcije (kada je sigurnost predikcije bila veća od 0.65 za pozitivne ili manja od 0.35 za negativne) (Tablica 9 – Neki od boljih rezultata tijekom klasifikacije primjera s velikom sigurnošću (>0.65) dubokim modelima). U nekim slučajevima se pokazuju jako dobri rezultati od 72% točnosti na 10 primjera ili 58% točnosti na 55 primjera kod dionice TSLA. Slično kao i prije, polje BROJ PODATAKA označava broj klasificiranih primjera kojima je sigurnost predikcije bila veća od 0.65 ili manja od 0.35 od ukupne klasifikacije 100 nadolazećih primjera.

Tablica 6 – Rezultati dubokih modela na podacima dionice GOOGL

GOOGL	BAZNA TOČNOST	FFNN (20,5)	LSTM(3,2)
1 day	53.00%	39.00%	40.00%
1 h	56.00%	52.00%	55.00%
30 min	55.00%	47.00%	53.00%
15 min	51.00%	43.00%	48.00%

Tablica 7 – Rezultati dubokih modela na podacima dionice TSLA

TSLA	BAZNA TOČNOST	FFNN (20,5)	LSTM(3,2)
1dDay	52.00%	50.00%	54.00%
1 h	51.00%	53.00%	54.00%
30 min	52.00%	49.00%	52.00%
15 min	55.00%	61.00%	56.00%

Tablica 8 – Rezultati dubokih modela na podacima dionice DAR

DAR	BAZNA TOČNOST	FFNN (20,5)	LSTM(3,2)
1 day	58.00%	50.00%	50.00%
1 h	56.00%	54.00%	53.00%
30 min	50.00%	53.00%	45.00%
15 min	58.00%	51.00%	52.00%

Tablica 9 – Neki od boljih rezultata tijekom klasifikacije primjera s velikom sigurnošću (>0.65) dubokim modelima

DIONICA	SKALIRANJE	INTERVAL	ALGORITAM	TOČNOST	BROJ PODATAKA
GOOGL	WINDOW-250	1 h	LSTM(3,2)	55.56%	72
GOOGL	WINDOW-250	30 min	LSTM(3,2)	54.05%	74
TSLA	WINDOW-250	1 day	LSTM(3,2)	58.18%	55
TSLA	WINDOW-250	1 h	FFNN (20,5)	56.36%	55
TSLA	WINDOW-250	15 min	FFNN (20,5)	72.73%	11
DAR	WINDOW-250	30 min	FFNN (20,5)	56.00%	25
DAR	WINDOW-250	15 min	FFNN (20,5)	55.17%	58
DAR	WINDOW-250	30 min	LSTM(3,2)	55.26%	38

5. Usporedba metoda

Postoji znatna vremenska razlika tijekom provedbe klasifikacije toka podataka tradicionalnih algoritama naspram algoritama dubokog učenja.

Tradicionalni algoritmi klasifikacije toka podataka su dosta brži prilikom uključivanja dospjelih podataka u znanje algoritma, te su radi toga manje vremenski zahtjevniji. Radi toga, testni skup kod tradicionalnih algoritama je bio veličine 1000 primjera dok je kod dubokih modela iznosio 100.

Rezultati pokazuju da je i tradicionalne algoritme i duboke modele potrebno posebno optimirati za svaku dionicu i vremenski korak. Međutim, duboki modeli se čine puno osjetljiviji s obzirom na dionicu. S obzirom na to da je veliki problem kod modela dubokog učenja prenaučenost, potrebno je drugačije podesiti parametre količine učenja i stope učenja s obzirom na dionicu i vremenski korak. Isto tako, radi puno većeg vremenskog zahtjeva dubokih modela, rezultati klasifikacije su dobiveni kompromisom između ažurnosti modela i količinom podataka za adekvatnu analizu. U realnoj situaciji, svaki model bi se učio s najboljim parametrima i najažurnijim podacima prije klasifikacije svakog podatka. Još jedan problem dubokih modela je nemogućnost prilagodbe na promjenu u distribuciji podataka to jest dolazak do pomaka koncepta.

Međutim, i tradicionalni algoritmi i modeli dubokog učenja pokazuju dobre rezultate prilikom klasificiranja primjera s velikom sigurnošću. Tu se duboki modeli u nekim slučajevima pokazuju bolji. Duboki modeli mogu kompleksnije reprezentirati podatke, te mogu raditi međusobne interakcije među značajkama. Tradicionalni modeli većinom rade preko prijašnjih statistika koje se ažuriraju prilikom dolaska pojedinog podatka. Na drugu ruku, tradicionalni algoritmi su značajno brži tijekom klasifikacije podataka što može biti ključno u slučajevima kada je potrebna klasifikacija unutar kratkog vremenskog roka.

Radi svega toga, najbolji rezultat bi mogao biti dobiven kombinacijom klasifikacije tradicionalnih algoritama i modela dubokog učenja putem npr. ansambl metode glasanja gdje bi se klasa primjera odredila s obzirom na klasu s najvećom količinom klasifikacije među modelima. Isto tako može se razmatrati dodavanje detektora pomaka koncepta na duboke modele, kako bi se duboki model učio na što reprezentativnijim podacima trenutne distribucije.

Zaključak

Unutar rada predstavljene su neke osnovne informacije vezane uz financijske instrumente s naglaskom na dionice. Prikupljeni su podaci nekoliko dionica u nekoliko vremenskih koraka, te su dodani dodatni indikatori kako bi se poboljšala vjerojatnost točne klasifikacije. Na kraju je napravljena međusobna usporedba performansi odabranih algoritama kroz više vremenskih intervala na više dionica. Kod tradicionalnih algoritama korištenih na klasifikaciji tokova podataka jako je korisna mogućnost predikcije pomaka koncepta. Također, radi jednostavnijeg učenja naspram dubokog modela, tradicionalni algoritmi su značajno brži tijekom klasifikacije nadolazećih podataka što ih čini jako korisnim u slučaju kratkog vremenskog intervala između nadolazećih podataka.

S druge strane, duboki modeli mogu puno kompleksnije reprezentirati podatke te raditi razne interakcije među parametrima radi višeslojne arhitekture, što može dovesti do boljih predikcija u nekim slučajevima. Glavna mana dubokih modela kod klasifikacije tokova podataka je prilagođavanje na pomak koncepta. Tijekom učenja, duboki modeli gledaju svaki primjer jednako gdje bi u slučaju pomaka koncepta recentniji primjeri bili važniji. Radi toga u budućnosti bi bilo korisno razmatrati duboki model koji koristi neku vrstu detekcije pomaka koncepta, gdje bi u slučaju dolaska do pomaka, model se ponovno učio na podacima koji bolje reprezentiraju trenutnu distribuciju podataka. Isto tako može se razmatrati kombinacija klasifikacije standardnih algoritama i klasifikaciju dubokih modela putem ansambla algoritama.

Također, klasični algoritmi i modeli dubokog učenja se pokazuju puno bolji kada se kod kalkulacije točnosti uzimaju samo oni primjeri koje je algoritam klasificirao s velikom sigurnošću. U slučaju razvijanja investicijske strategije, potencijalno bi bolje bilo uzimati samo primjere s visokom vjerojatnošću klasifikacije nasuprot ulaganja tijekom svakog vremenskog intervala.

Tijekom usporedbe točnosti klasifikacije vidi se razlika među dionicama i vremenskim koracima. Takvo ponašanje se može objasniti time da svaka dionica posluje drugačije, ima drugačije temeljne podatke, te operira u drugačijoj industriji što može uzrokovati drugačije ponašanje cijene. Radi toga idealno bi bilo parametre svakog modela i algoritma prilagođavati na određenu dionicu, čak i na određeni vremenski korak koji se klasificira. Isto

tako, bilo bi korisno razmatrati predikciju više kompanija unutar iste industrije te takve klasifikacije kombinirati s klasifikacijom same dionice te industrije.

Radi toga što u financijskim tržištima postoje mnogi igrači, cijene dionica često sadrže puno šuma, naročito popularnije (npr. GOOGL), što može raditi probleme tijekom klasifikacije na manjim vremenskim intervalima. Također, radi ovisnosti dionice o generalnom stanju ekonomije u kojoj operiraju, potrebno je konstantno prilagođavati algoritme i modele tom evoluirajućem tržištu. Zbog toga bilo bi korisno u buduću analizu uključiti dodatne parametre koje opisuju trenutno stanje ekonomije i tržišta, te potencijalno i neke temeljne podatke dionice.

Literatura

[1] Kenton W., *Financial Instrument*, Investopedia, (2021, kolovoz). Poveznica: https://www.investopedia.com/terms/f/financialinstrument.asp ; pristupljeno 15. svibnja 2022.

[2] Kramer L., *Long Position vs. Short Position: What's the Difference?*, Investopedia, (2021, lipanj). Poveznica: https://www.investopedia.com/ask/answers/100314/whats-difference-between-long-and-short-position-market.asp ; pristupljeno 15. svibnja 2022.

[3] Segal T., *Short-Term Investments*, Investopedia, (2021, travanj). Poveznica: https://www.investopedia.com/terms/s/shorterminvestments.asp ; pristupljeno 15. svibnja 2022.

[4] Chen J., *Algorithmic Trading*, Investopedia, (2022, sječanj). Poveznica: https://www.investopedia.com/terms/a/algorithmictrading.asp ; pristupljeno 15. svibnja 2022.

[5] Hayes A., *Stock Definition*, Investopedia, (2022, ožujak). Poveznica: https://www.investopedia.com/terms/s/stock.asp ; pristupljeno 15. svibnja 2022.

[6] Chen J., *Stock Market*, Investopedia, (2022, ožujak). Poveznica: https://www.investopedia.com/terms/s/stockmarket.asp ; pristupljeno 15. svibnja 2022.

[7] Chen J., *Index*, Investopedia, (2022, ožujak). Poveznica: https://www.investopedia.com/terms/i/index.asp ; pristupljeno 15. svibnja 2022.

[8] Chen J., *Technical Indicator*, Investopedia, (2021, rujan). Poveznica: https://www.investopedia.com/terms/t/technicalindicator.asp ; pristupljeno 15. svibnja 2022.

[9] Hayes A., *Simple Moving Average (SMA)*, Investopedia, (2022, veljača). Poveznica: https://www.investopedia.com/terms/s/sma.asp ; pristupljeno 15. svibnja 2022.

[10] Chen J., *Exponential Moving Average (EMA)*, Investopedia, (2022, ožujak). Poveznica: https://www.investopedia.com/terms/e/ema.asp ; pristupljeno 15. svibnja 2022.

[11] Hayes A., *Bollinger Band®*,, Investopedia, (2022, srpanj). Poveznica: https://www.investopedia.com/terms/b/bollingerbands.asp ; pristupljeno 15. svibnja 2022.

[12] Fernando J., *Relative Strength Index (RSI)*, Investopedia, (2022, veljača). Poveznica: https://www.investopedia.com/terms/r/rsi.asp ; pristupljeno 15. svibnja 2022.

[13] Hayes A., *Stochastic Oscillator*, Investopedia, (2022, lipanj). Poveznica: https://www.investopedia.com/terms/s/stochasticoscillator.asp ; pristupljeno 15. svibnja 2022.

[14] Fernando J., *Moving Average Convergence Divergence (MACD)*, Investopedia, (2022, ožujak). Poveznica: https://www.investopedia.com/terms/m/macd.asp ; pristupljeno 15. svibnja 2022.

[15] Hayes A., *On-Balance Volume (OBV) Definition*, Investopedia, (2022, srpanj). Poveznica: https://www.investopedia.com/terms/o/onbalancevolume.asp ; pristupljeno 15. svibnja 2022.

[16] Mitchell C., *Accumulation/Distribution Indicator (A/D)*, Investopedia, (2021, svibanj). Poveznica: https://www.investopedia.com/terms/a/accumulationdistribution.asp ; pristupljeno 15. svibnja 2022.

[17] Mitchell C., *Average Directional Index (ADX)*, Investopedia, (2022, kolovoz). Poveznica: https://www.investopedia.com/terms/a/adx.asp ; pristupljeno 15. svibnja 2022.

[18] Mitchell C., *Aroon Oscillator*, Investopedia, (2021, kolovoz). Poveznica: https://www.investopedia.com/terms/a/aroonoscillator.asp ; pristupljeno 15. svibnja 2022.

[19] Majaski C., *Fundamentals*, Investopedia, (2021, svibanj). Poveznica: https://www.investopedia.com/terms/f/fundamentals.asp ; pristupljeno 15. svibnja 2022.

[20] Kenton W., *The S&P 500 Index: Standard & Poor's 500 Index*, Investopedia, (2022, veljača). Poveznica https://www.investopedia.com/terms/s/sp500.asp ; pristupljeno 15. svibnja 2022.

[21] Kuepper J., *Cboe Volatility Index (VIX)*, Investopedia, (2022, svibanj). Poveznica https://www.investopedia.com/terms/v/vix.asp ; pristupljeno 15. svibnja 2022.

[22] Shobhit S., *Basics of Algorithmic Trading: Concepts and Examples*, Investopedia, (2021, svibanj). Poveznica: https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp ; pristupljeno 2. lipnja 2022.

[23] *skmultiflow.drift_detection.ADWIN*, Scikit-multiflow, (2021, svibanj). Poveznica: https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.drift_detection.ADWIN.html ; pristupljeno 2. lipnja 2022.

[24] Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016.

[25] Phi M., *Illustrated Guide to LSTM's and GRU's: A step by step explanation*, Towards Data Science, (2018, rujan). Poveznica: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 ; pristupljeno 2. lipnja 2022.

[26] *Long short-term memory*, Wikipedia, Poveznica: https://en.wikipedia.org/wiki/Long_short-term_memory ; pristupljeno 2. lipnja 2022.

[27] *Introduction to LSTM Units in RNN*, Pluralsight, Poveznica: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn, pristupljeno 2. lipnja 2022.

Sažetak

Naslov:

Previđanje cijene kretanja financijskih instrumenata optimizacijom performansi metoda strojnog učenja iz tokova podataka i metoda dubokog učenja

Sažetak:

Klasifikacija smjera kretanja cijene financijskih instrumenata je vrlo izazovno, ali jako zahvalno područje u slučaju pronalaska uspješne strategije. Cilj prezentiranih algoritama je klasifikacija kretanja cijene (rast ili pad) dionica u određenom trenutku s određenim vremenskim periodom razmaka između trenutne i nadolazeće cijene. Unutar rada predstavljene su neke osnovne informacije vezane uz financijske instrumente s naglaskom na dionice. Prikupljeni su podaci nekoliko dionica u nekoliko vremenskih koraka, te su dodani dodatni indikatori kako bi se poboljšala vjerojatnost točne klasifikacije. Na kraju se napravila međusobna usporedba performansi odabranih algoritama strojnog i dubokog učenja kroz više vremenskih intervala, na više dionica.

Ključne riječi: klasifikacija, dionice, tok podataka, duboki modeli

Summary

Title:

Financial Price Data Prediction by Optimizing Stream-based Machine Learning Methods and Deep Learning Methods

Summary:

Classifying the direction of price movements of financial instruments is a very challenging, but also a very lucrative area in case of finding a successful strategy. The goal of the presented algorithms is to classify the movement of prices (rise or fall) of a stock at a given time with a certain period of time between the current and the upcoming price. In this thesis, some basic information related to financial instruments is presented, with an emphasis on stocks. Data from several stocks and several time steps were collected, and additional indicators were added to improve the probability of accurate classification. Finally, the performance of the selected machine learning and deep learning algorithms was compared over several time intervals and on several stocks.

Keywords: classification, stocks, streaming data, deep models