SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 452

SKALABILNO PREDVIĐANJE VREMENSKIH NIZOVA NA FINANCIJSKIM TRŽIŠTIMA U STVARNOM VREMENU ZASNOVANO NA DUBOKOM UČENJU

Stjepan Ruklić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 452

SKALABILNO PREDVIĐANJE VREMENSKIH NIZOVA NA FINANCIJSKIM TRŽIŠTIMA U STVARNOM VREMENU ZASNOVANO NA DUBOKOM UČENJU

Stjepan Ruklić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 452

Pristupnik:	**Stjepan Ruklić (0036516410)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentor:	izv. prof. dr. sc. Alan Jović

Zadatak: **Skalabilno predviđanje vremenskih nizova na financijskim tržištima u stvarnom vremenu zasnovano na dubokom učenju**

Opis zadatka:

Predviđanje vremenskih nizova (engl. time series forecasting) ima za cilj odrediti buduće vrijednosti vremenskog niza podataka na temelju njegovih prošlih vrijednosti. Uspješnim modelom predikcije možemo donositi informiranije odluke na financijskom tržištu koje mogu rezultirati financijskom koristi. Danas postoji nekoliko pristupa predviđanju, a jedan od njih je i duboko učenje. U ovom radu potrebno je koristiti aktualne podatke iz financijskih tržišta temeljem tokova podataka. Tokove podataka potrebno je obraditi korištenjem platforme Apache Kafka te ih nakon toga proslijediti na ulaz modela primjenom tehnike slijednog strojnog učenja (engl. sequential machine learning, online machine learning). Programski sustav u okviru diplomskog rada treba podržati učitavanje podataka financijskih vremenskih nizova u stvarnome vremenu, izlučivanje značajki te izgradnju modela na temelju jedne ili više metoda strojnog učenja, uključujući duboke neuronske mreže, s ciljem postizanja što veće uspješnosti na odabranom problemu. Kako bi se model mogao primijeniti na različitim vremenskim nizovima korištenjem različitih hiperparametara modela strojnog učenja potrebno je upotrijebiti orkestracijski alat k8s. U radu je potrebno razmotriti nekoliko financijskih vremenskih nizova prema vlastitom izboru te kvantitativno usporediti rezultate modela s nekim od srodnih istraživanja.

Rok za predaju rada: 28. lipnja 2024.

Zahvaljujem se svome mentoru profesoru dr. sc. Alanu Joviću na stručnoj pomoći i suradnji. Također, zahvaljujem se svojoj obitelji i prijateljima na podršci tijekom studiranja.

Sadržaj

1. Uvod

U ovom diplomskom radu opisan je postupak izrade sustava za ovladavanje financijskim tržištima i njihovim nestabilnostima primjenom predikcijskih modela.

Za prognoziranje cijena valuta financijskih tržišta potrebne su nam povijesne informacije o cijeni istih. Prikupljeni podaci zatim se pripremaju za proces učenja modela.

Da bi se dobila sveobuhvatna ideja o performancama modela uspoređene su performance statističkih modela kao što su ARIMA, SARIMA i Prophet te performance modela dubokog učenja. Performanca modela se mjeri usporedbom prognoziranih cijena na povijesnim podatcima.

Konačno, opisana je arhitektura koja je korištena kako bi se pokrenule ove komponente rada u jednoj funkcionalnoj cjelini.

U nastavku rada objasnit će se što je sve potrebno da bi se držalo strategije *"Buy low, sell high"*.

2. Financijska tržišta

Financijska tržišta (engl. *financial markets*) su tržišta u kojima se sredstva prenose od ljudi koji imaju višak dostupnih sredstava prema ljudima koji imaju nedostatak. Financijska tržišta, poput tržišta obveznica i dionica, ključna su za promicanje veće ekonomske učinkovitosti usmjeravanjem sredstava od ljudi koji nemaju produktivnu upotrebu za njih do onih koji je imaju. Zapravo, dobro funkcionirajuća financijska tržišta ključni su faktor za postizanje visokog gospodarskog rasta, a loše funkcionirajuća financijska tržišta jedan su od razloga zašto mnoge zemlje u svijetu ostaju očajno siromašne. Aktivnosti na financijskim tržištima također imaju izravne učinke na osobno bogatstvo, ponašanje poduzeća i potrošača te cikličke performance gospodarstva [1].

Financijske institucije su ono što omogućava funkcioniranje financijskih tržišta. Bez njih, financijska tržišta ne bi mogla preusmjeravati sredstva od ljudi koji štede prema ljudima koji imaju produktivne investicijske mogućnosti [1].

Funkcije financijskih tržišta su:

- **Alokacija kapitala**: Učinkovito usmjeravanje kapitala prema najproduktivnijim investicijama.
- **Održavanje likvidnosti**: Omogućavanje brzog i jednostavnog pretvaranja vrijednosnih papira u gotovinu.
- **Cijene informacija**: Omogućavanje transparentnosti i ažurnih informacija o cijeni i vrijednosti različitih financijskih instrumenata.
- **Diverzifikacija rizika**: Investitori mogu diversificirati svoja ulaganja kako bi smanjili rizik.

Dionica predstavlja udio vlasništva u korporaciji. To je vrijednosni papir koji predstavlja pravo na zaradu i imovinu korporacije. Izdavanje dionica i njihova prodaja javnosti način je na koji korporacije prikupljaju sredstva za financiranje svojih aktivnosti. Tržište dionica, na kojem se trguje pravima na zaradu korporacija (udjelima u dionicama), najšire je praćeno financijsko tržište u gotovo svakoj zemlji koja ga ima; zato se često jednostavno naziva "tržište" [1].

Kriptovalute su digitalne ili virtualne valute koje koriste kriptografiju za sigurnost. Za razliku od tradicionalnih valuta koje izdaju vlade (fiat valute), kriptovalute djeluju na decentraliziranim mrežama temeljenima na tehnologiji *blockchain*. Neke od ključnih karakteristika kriptovaluta su [2]:

- **Digitalna priroda**: Kriptovalute postoje samo u digitalnom obliku i nemaju fizički ekvivalent poput kovanica ili novčanica. Pohranjuju se u digitalnim novčanicima i mogu se trgovati na raznim internetskim platformama
- **Decentralizacija**: Većina kriptovaluta djeluje na decentraliziranoj mreži računala (čvorova), što znači da niti jedan entitet ne kontrolira cijelu mrežu. To je u suprotnosti s tradicionalnim valutama, koje izdaju i reguliraju središnje banke.
- **Sigurnost**: Kriptovalute koriste kriptografske tehnike za osiguranje transakcija, kontrolu stvaranja novih jedinica i provjeru prijenosa imovine. To ih čini otpornima na prijevare i hakiranje.
- **Ograničena ponuda**: Mnoge kriptovalute imaju ograničenu ponudu, što znači da postoji određena količina te valute u opticaju. Ovom karakteristikom valute valuta se bori protiv inflacija i spontano stvara vrijednost.

3. Skup podataka

Po definiciji, globalna financijska tržišta su istovremeno složena i vrlo raznolika. Ulažu se znatni napori kako bi se prikupile relevantne informacije povezane s tržištima. Izvori koje obično koriste financijske institucije uključuju [3]:

- burze (engl. *exchanges*)
- web portali
- dobavljači tržišnih podataka
- interni doprinosi
- vladine agencije
- novinske agencije

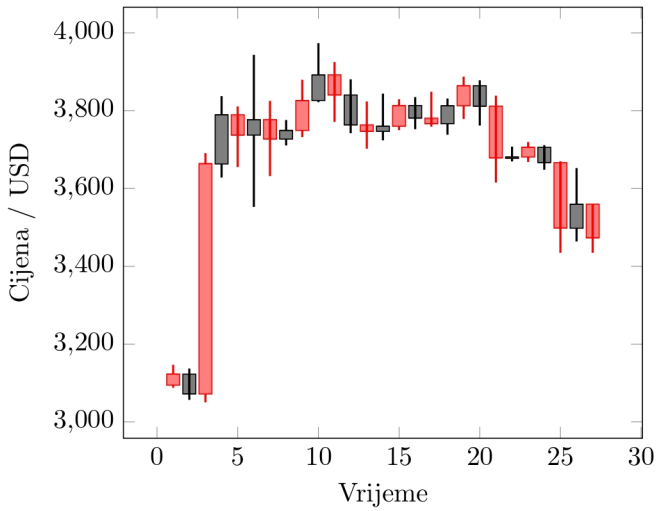
3.1. Izvor podataka

Podatci o vrijednostima valuta financijskih tržišta za ovaj sustav su preuzeta putem usluge Binance. Binance pruža uslugu preuzimanja povijesnih zapisa o vrijednostima valuta financijskih tržišta kao i uslugu praćenja aktualnih podataka putem web-socketa.

Na trgovačkoj burzi imovina (engl. *assets*) trguje se u parovima npr. ETH/BTC ili EUR/USD. Jedna (osnovna) imovina (engl. *base asset*) trguje se protiv druge (kotirajuće) imovine (engl. *quote asset*). *Spot*-tečaj (*spot* cijena) odražava koliko je kotirajuće imovine potrebno za kupnju 1 jedinice osnovne imovine. Na primjeru para ETH/USD, Ethereum (ETH) je osnovna imovina, a američki dolar (USD) je kotirajuća imovina. Ako je spot cijena 2000 USD, to znači da je potrebno 2000 američkih dolara za kupnju jednog Ethereuma.

3.2. Opis podataka

Svijeća (engl. *candle*), poznat i kao zapis OHLC (*open*, *high*, *low*, *close*), je metoda za prikazivanje kretanja cijena u trgovanju financijskim imovinama. Svaka svijeća prikazuje cijene u određenom vremenskom periodu, pružajući vizualnu informaciju o prvoj (*open*), najvišoj (*high*), najnižoj (*low*) i zadnjoj (*close*) cijeni za taj period. Dijagram svijećnjaka se vizualizira kao niz pozitivnih (crvenih) ili negativnih (crnih) svijeća što se vidi na slici 3.1.



Slika 3.1. Dijagram svijećnjaka kriptovalute Ethereum [4]

Servis Binance za svaki period dostavlja podatke svijećnjaka uključujući i podatake o volumenu trgovanja te količini odrađenih transakcija:

- **open time**: vremenska oznaka početka perioda u formatu unix
- **open**: početna cijena valute tog perioda
- **high**: najviša cijena valute tog perioda
- **low**: najniža cijena valute tog perioda
- **close**: konačna cijena valute tog perioda
- **volume**: ukupan broj dionica kojima je trgovano

- **close time**: vremenska oznaka kraja perioda u formatu unix
- **quote volume**: ukupan broj kotirajuće valute koja je korištena za kupnju ove valute
- **count** ukupan broj transakcija u periodu
- **taker buy volume**: ukupna količina naloga za kupnju koje su ispunili kupci unutar perioda
- **taker buy quote volume**: ukupna količina naloga za kupnju kotirajuće valute koje su ispunili kupci unutar perioda
- **ignore**: označava treba li se vrijednost ignorirati ili ne

Primjer zapisa koju dostavlja usluga Binance može se vidjeti u tablici 3.1.

open time	open	high	low	close	volume	close time	quote volume	count	taker buy volume	taker buy quote volume	ignore
1717923600	69352.1	69397.1	69341.9	69369.9	195.9	1717927199	13589024.6	14803	75.1	5214728.3	0
1717927200	69369.9	69373.2	69333.0	69346.4	236.8	1717930799	16426387.5	15935	87.5	6070642.9	0
1717930800	69346.5	69389.0	69333.0	69367.6	225.6	1717934399	15651519.3	18782	88.3	6126493.3	0
1717934400	69367.6	69760.0	69351.1	69712.9	864.7	1717937999	60123067.0	40933	558.6	38835556.1	0
1717938000	69712.9	69777.0	69250.0	69467.0	1517.1	1717941599	105425078.2	50246	764.5	53113677.4	0
1717941600	69467.0	69559.0	69264.0	69490.1	679.9	1717945199	47214532.2	30504	352.3	24472890.9	0
1717945200	69490.1	69590.3	69420.4	69508.6	716.4	1717948799	49809354.8	25953	318.1	22116243.4	0
1717948800	69508.6	69636.9	69500.0	69622.8	386.1	1717952399	26867439.0	26070	222.8	15500601.5	0
1717952400	69622.8	69698.4	69570.0	69660.5	281.7	1717955999	19619563.5	26988	121.6	8468759.3	0
1717956000	69660.5	69782.9	69611.2	69751.0	468.2	1717959599	32648840.9	13810	258.0	17994009.4	0

Tablica 3.1. Primjer zapisa koji pruža web-usluga Binance za par BTC/USDT

Granulacija zapisa koju usluga Binance pruža seže od jedne sekunde do jednog dana. Ta granulacija predstavlja proteklo vrijeme između prve cijene perioda (*open*) i zadnje cijene perioda (*close*).

3.3. Obrada podataka

3.3.1. Skaliranje

Za potrebe ovog rada korištena su samo podatci o početnoj cijeni perioda (Open) i vremenskoj oznaci tog perioda (Open time). Vrijednost valute skalirana je korištenjem standardizacijske metode *StandardScaler* programske knjižice *scikit* koja može biti izražena kao:

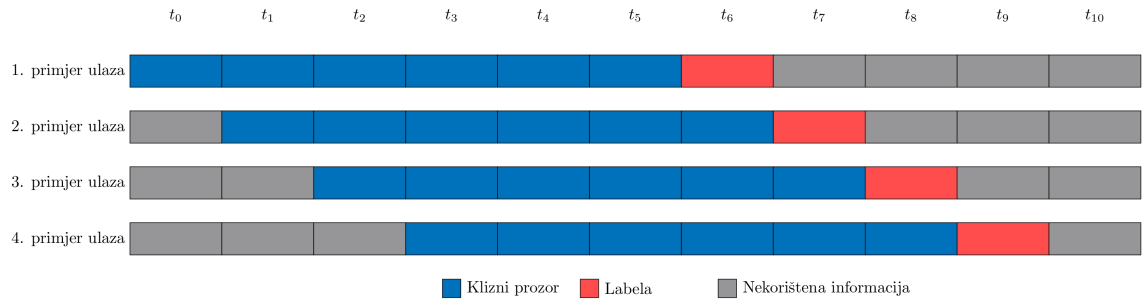
$$z = \frac{x - \mu}{\sigma} \tag{3.1}$$

gdje je μ srednja vrijednost skupa podataka za treniranje , a σ označava standardnu devijaciju istog skupa.

3.3.2. Podjela podataka

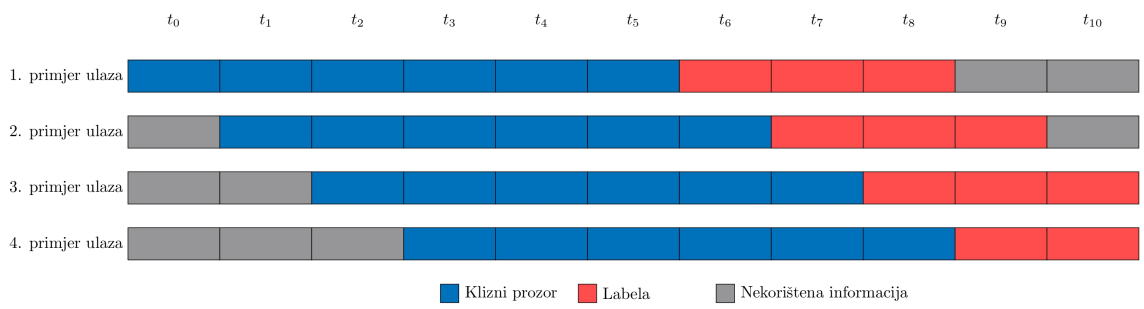
Ulazni skup podataka je raspodjeljen metodom kliznog prozora (engl. *data windowing*). Razlog zbog kojega želimo pretvoriti vremenski niz u skup kliznih prozora je da predviđanje budućih vrijednosti vremenskog niza pretvorimo u problem nadziranog učenja [5].

Duljina prozora predstavlja broj vremenskih zapisa koje koristimo za predikciju horizonta. Na slikama 3.2. i 3.3. obojano plavom bojom.



Slika 3.2. Podjela podataka s duljinom prozora šest vremenskih zapisa i duljinom horizonta od jednog vremenskog zapisa

Horizont (engl. *horizon*) u kontekstu kliznih prozora predstavlja količinu budućih vremenskih zapisa za koje želimo napraviti predikciju, a nalaze se neposredno nakon kliznog prozora. Označeno je crvenom bojom na slikama 3.2. i 3.3.



Slika 3.3. Podjela podataka s duljinom prozora šest vremenskih zapisa i duljinom horizonta od tri vremenska zapisa

4. Predikcijski modeli

4.1. Statistički modeli

4.1.1. Nesezonalni modeli

Autoregressive Integrated Moving Average (ARIMA)

Autoregresivni integrirani klizni prosjek, ili ARIMA, je model statističke analize koji koristi podatke vremenske nizove za bolje razumijevanje skupa podataka ili za predviđanje budućih trendova.

1. **Prošle vrijednosti**: Jasno je da je prošlo ponašanje dobar prediktor budućnosti. Glavno pitanje je koliko prošlih vrijednosti bismo trebali koristiti. Model koristi zadnjih p vrijednosti, gdje je p hiperparametar modela.

2. **Pogreške**: Model na osnovu zadnjih q vrijednosti može odrediti koliko je točan. Također, q je hiperparametar modela.

Pomoću tri parametra p, d i q definiramo model tri glavne komponente modela.

1. **Integrirani (I u ARIMA)**: Broj razlika potrebnih za postizanje stacionarnosti zadan je parametrom d. Neka su izvorne značajke Y_t gdje je t indeks niza. Kreiramo stacionarni vremenski niz koristeći sljedeće transformacije za različite vrijednosti d.

   Za $d = 0$

   $$y_t = Y_t \tag{4.1}$$

   Za $d = 1$

   $$y_t = Y_t - Y_{t-1} \tag{4.2}$$

Za $d = 2$

$$y_t = Y_t - Y_{t-1} - (Y_{t-1} - Y_{t-2}) = Y_t - 2 * Y_{t-1} + Y_{t-2} \tag{4.3}$$

2. **Autoregresivni (AR u ARIMA)**: Parametar p označava koliko prošlih vrijednosti treba uzeti u obzir za izraz trenutne vrijednosti. U suštini, uči se model koji predviđa vrijednost u trenutku t kao:

$$\overline{y_t} = C + \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t \tag{4.4}$$

3. **Klizni prosjek (MA u ARIMA)**: Koliko pogrešaka predviđanja u prošlosti treba uzeti u obzir. Time se dobiva:

$$\overline{y_t} = C + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{4.5}$$

Pogreške prošlih predviđanja:

$$\epsilon_t = y_t - \epsilon_{t-1} \tag{4.6}$$

Kombinacijom ovih triju komponenti dobiva se model $ARIMA(p, d, q)$.

Generalizirani izraz za model ARIMA:

$$\overline{y_t} = C + \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{4.7}$$

4.1.2. Sezonalni modeli

Seasonal Autoregressive Integrated Moving Average (SARIMA)

Sezonalni autoregresivni integrirani klizni prosjek, SARIMA ili sezonalna ARIMA, je proširenje modela ARIMA koje eksplicitno podržava univarijantne vremenske nizove sa sezonskom komponentom. Dodaje tri nova hiperparametra za određivanje autoregresije (AR), razlike (I) i klizni prosjek (MA) za sezonsku komponentu niza, kao i dodatni parametar za razdoblje sezonalnosti.

Predstavljanjem operator pomaka unazad (engl. *backshift operator*):

$$B^n y_t = y_{t-n} \tag{4.8}$$

pojednostavljujemo izraz za autoregresiju 4.4 (AR) modela ARIMA u:

$$\Phi(B) = 1 + \phi_1 B^1 + ... + \phi_p B^p \tag{4.9}$$

Izraz za klizni prosjek 4.5 (MA) postaje:

$$\Theta(B) = 1 + \phi_1 B^1 + ... + \phi_q B^q \tag{4.10}$$

Uporabom tih izraza, opći izraz za model ARIMA se pretvara u:

$$\Phi(B)\nabla^d y_t = \Theta(B)\epsilon_t \tag{4.11}$$

gdje je ∇^d operatora razlike $(1 - B)^d$ [6].

1. **Sezonalno integrirani (SI u SARIMA):** Parametar D slično kao i u ne sezonalnom modelu uzima u obzir razliku potrebnu za uklanjanje sezonalnosti iz niza

$$\nabla_S^D = (1 - B^S)^D \tag{4.12}$$

2. **Sezonalno autoregresivni (SA u SARIMA):** Ova komponenta preko parametra

P bilježi odnos između trenutne vrijednosti vremenske nizove i njegovih prošlih vrijednosti, posebno u sezonskim kašnjenjima (engl. *lag*).

$$\Phi_S(B) = 1 + \phi_1 B^S + ... + \phi_P B^{PS} \tag{4.13}$$

3. **Sezonalno klizni prosjek (SMA u SARIMA):** Parametar *Q* predstavlja ovisnost između trenutne vrijednosti i preostalih pogrešaka prethodnih predviđanja sa sezonskim kašnjenjima.

$$\Theta_S(B) = 1 + \theta_1 B^S + ... + \theta_Q B^{QS} \tag{4.14}$$

Kombinacijom ovih triju komponenti dobiva se model $ARIMA(p,d,q)ARIMA(P,D,Q)$, ili $SARIMA(p,d,q)(P,D,Q)S$.

Generalizirani izraz za model SARIMA:

$$\nabla^d \nabla_S^D y_t = \frac{\Theta(B) \times \Theta_S(B)}{\Phi(B) \times \Phi_S(B)} \epsilon_t \tag{4.15}$$

Prophet

Facebook Prophet (FB Prophet) je model predviđanja vremenskih nizova oblikovan za rukovanje zajedničkim značajkama poslovnih vremenskih nizova [7].

Model se sastoji od tri glavne komponente:

- trend (engl. *trend*)
- sezonalnost (engl. *seasonality*)
- blagdani (engl. *holidays*)

Zajedno formiraju sljedeću jednadžbu:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t \tag{4.16}$$

- ***g(t)*** predstavlja funkciju trenda koja modelira neperiodične promjene u vrijednosti vremenske nizove.
- ***s(t)*** prikazuje funkciju sezonalnosti (tjedno, mjesečno i godišnje).
- ***h(t)*** predstavlja učinke praznika koji se pojavljuju na potencijalno neredovitim rasporedima tijekom jednog ili više dana.

4.2. Neuronske mreže

Duboko učenje je podskup strojnog učenja koji se fokusira na izgradnju modela na arhitekturi neuronskih mreža. Prednost dubokog učenja je u tome što obično daje bolje rezultate kada je dostupno više podataka, što ga čini izvrsnim izborom za predviđanje vremenskih nizova s visokim dimenzijama [8].

Koristeći povijesne podatke, modeli dubokog učenja uče funkcionalni odnos između ulaznih značajki i budućih vrijednosti ciljne varijable. Rezultirajući model može pružiti predviđanja za ciljanu varijablu u budućim vremenskim točkama, koja je opisana nizom $x_1, x_2, ..., x_T$ gdje je x_t vektor od m ulaznih značajki mjerenih u trenutku t [9].

Naš zadatak je razviti model koji će predvidjeti vrijednost y_{t+k} u nekoj budućoj vremenskoj točki $t+k$ koristeći povijesne vrijednosti. Da bi ulaz u model bio uniformne duljine koristimo klizni vremenski prozor fiksne duljine w. Odnos naučen modelima strojnog učenja može se prikazati kao [9]:

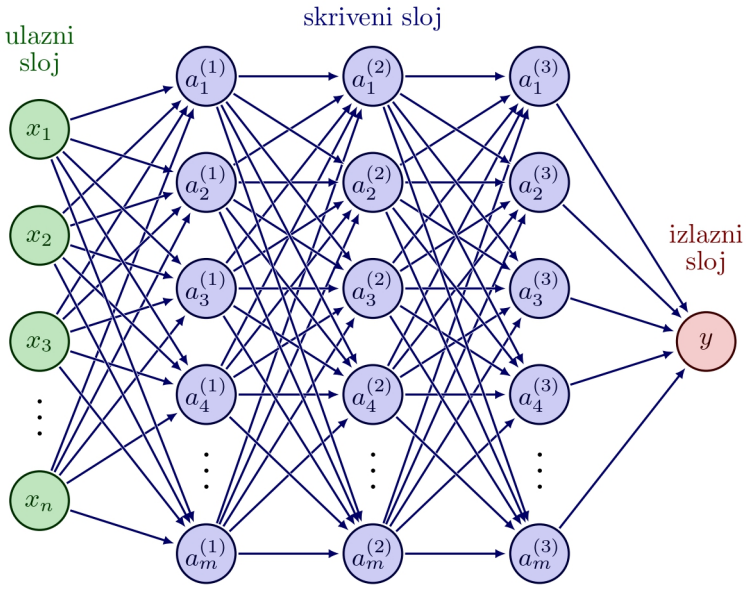
$$\hat{y}_{t+k} = f_k(x_{t-w}, ..., x_{t-1}, y_{t-w}, ..., y_{t-1}) \tag{4.17}$$

gdje je $\hat{y}_{t+k}$ ciljna varijabla predviđanja za vrijeme $t+k$; k opisuje broj vremenskih koraka u budućnost varijable koju predviđamo; $y_{t-w}, ..., y_{t-1}$ su ciljne vrijednosti od trenutka $t-w$ do trenutka $t-1$; $x_{t-w}, ..., x_{t-1}$ su vektori ulaznih vremenskih nizova od trenutka $t-w$ do trenutka $t-1$; f_k je funkcija naučena modelom dubokog učenja.; m je broj ulaznih značajki; w je veličina klizajućeg prozora korištena na ulaznim podacima [9].

4.2.1. Potpuno povezane neuronske mreže

Duboke unaprijedne mreže, također često nazvane unaprijedne neuronske mreže ili višeslojni perceptroni (MLPs), su osnovni modeli dubokog učenja.

Lijevi sloj naziva se **ulazni sloj**, a neuroni unutar sloja nazivaju se ulazni neuroni. Neuron ovog sloja je posebne vrste jer nema ulaz i samo izlazno daje vrijednost x_j za j-tu značajku. Desni ili **izlazni sloj** sadrži izlazne neurone (samo jedan ovdje). Srednji sloj naziva se **skriveni sloj**, jer neuroni u ovom sloju nisu ni ulazi ni izlazi. Ne promatra se (izravno) što izlazi iz ovog sloja [10].



Slika 4.1. Primjer potpuno povezane neuronske mreže [11]

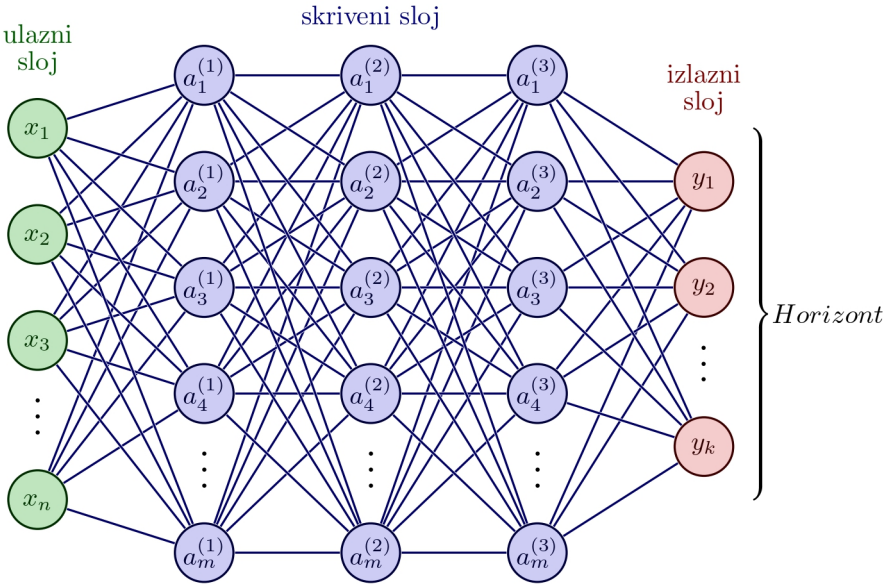
Slika 4.1. prikazuje duboku neuronsku mrežu s tri skrivena sloja gdje je jedan neuron u izlaznom sloju. Veličina izlaznog sloja ovisi o broju budućih vremenskih perioda za koje želimo napraviti predikciju. Tako da za ulazne podatke u obliku prikazane slikom 3.2. odgovara duboka neuronska mreža prikazana na slici 4.1.

Cilj unaprijedne mreže je aproksimirati neku funkciju f^*. Unaprijedna mreža definira preslikavanje $y = f(x;\theta)$ i uči vrijednosti parametara θ koje rezultiraju najboljom aproksimacijom funkcije [10].

Funkcija f sastoji se od lanca funkcija:

$$f = f^{(k)}(f^{(k-1)}(...f^{(1)})) \tag{4.18}$$

gdje se $f^{(1)}$ naziva prvi sloj, i tako dalje. Dubina mreže je k [10].



Slika 4.2. Primjer potpuno povezane neuronske mreže [11]

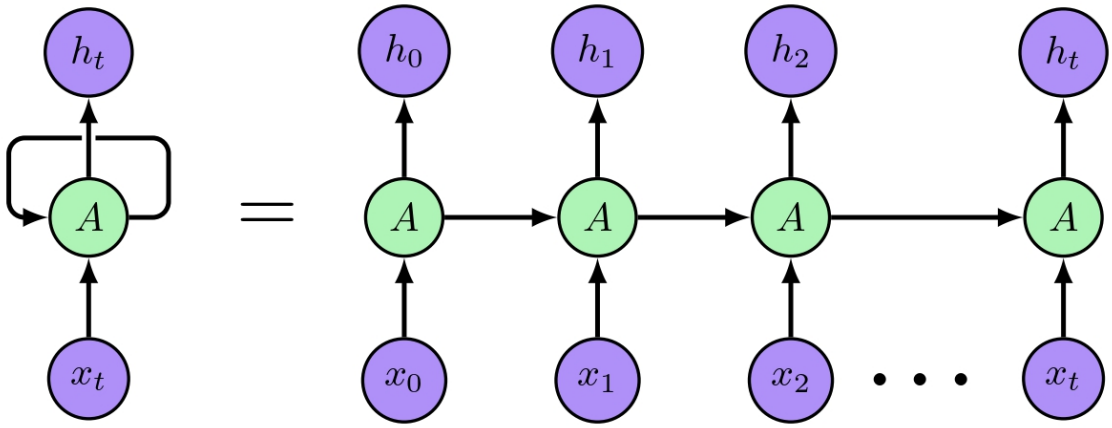
Za neuronsku mrežu 4.2. više odgovaraju ulazni podatci prikazani slikom 3.3. gdje vrijedi $k = 3$ za dimenziju izlaznog sloja.

Ovim jednostavnim primjerom možemo vidjeti kako je problem prognoziranja vremenskih nizova povezan s arhitekturom modela dubokih neuronskih mreža.

4.2.2. Povratne neuronske mreže

Povratna neuronska mreža (engl. *recurrent neural network*, RNN) je vrsta umjetne neuronske mreže koja koristi sekvencijalne podatke ili vremenske nizove. Ovi algoritmi dubokog učenja obično se koriste za serijske ili vremenske probleme, kao što su prijevod jezika, obrada prirodnog jezika (engl. *natural language processing*, NLP), prepoznavanje govora i opisivanje slika. Odlikuju se svojim "pamćenjem" jer uzimaju informacije iz prethodnih ulaza kako bi utjecali na trenutni ulaz i izlaz, kao što je prikazano na slici 4.3.

Dok tradicionalne duboke neuronske mreže pretpostavljaju da su ulazi i izlazi neovisni jedan o drugom, izlaz rekurentnih neuronskih mreža ovisi o prethodnim elementima unutar niza [12].



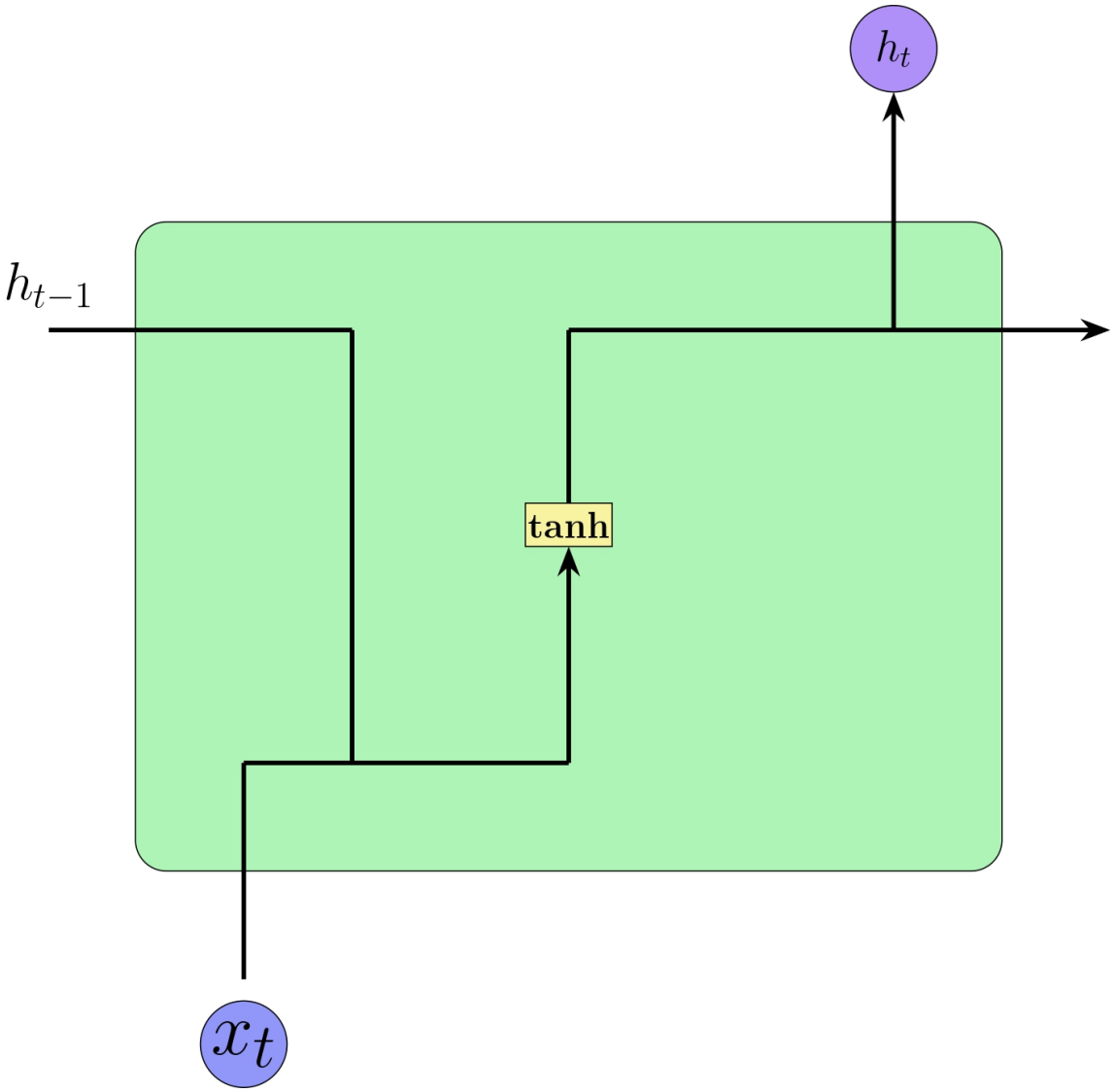
Slika 4.3. Primjer 2D konvolucije

Kao što je prikazano na slici 4.3., RNN uči na povijesnim podacima vremenskog niza fokusirajući se na prijelaze od skrivenog vremenskog stanja $t-1$ u t. Model je opisan matricama W_x, W_h i W_y te vektorima pomaka b_h i b_y. Izlaz $h^{(t)}$ ovisi o stanju, $h^{(t)}$, koje ovisi i o ulaznom podatku $x^{(t)}$ i o prošlom stanju $h^{(t-1)}$ [9]. Slika 4.4. vizualizira tu ovisnost.

$$h^{(t)} = tanh(W_{hh}h^{(t-1)} + W_{xh}x^{(t)} + b_h) \tag{4.19}$$

$$o^{(t)} = W_{hy}h^{(t)} + b_o \tag{4.20}$$

gdje je $x_t \in \mathbb{R}^m$ ulazni vektor m značajki u trenutku t; $W_{xs} \in \mathbb{R}^{n\times(m+n)}$; n predstavlja broj neurona u sloju RNN-a; $b_s \in \mathbb{R}^N$ i $b_y \in \mathbb{R}^N$ su vektori pomaka trenutnog stanja i izlaza; $h^{(t)}$ je trenutno skriveno stanje [9].

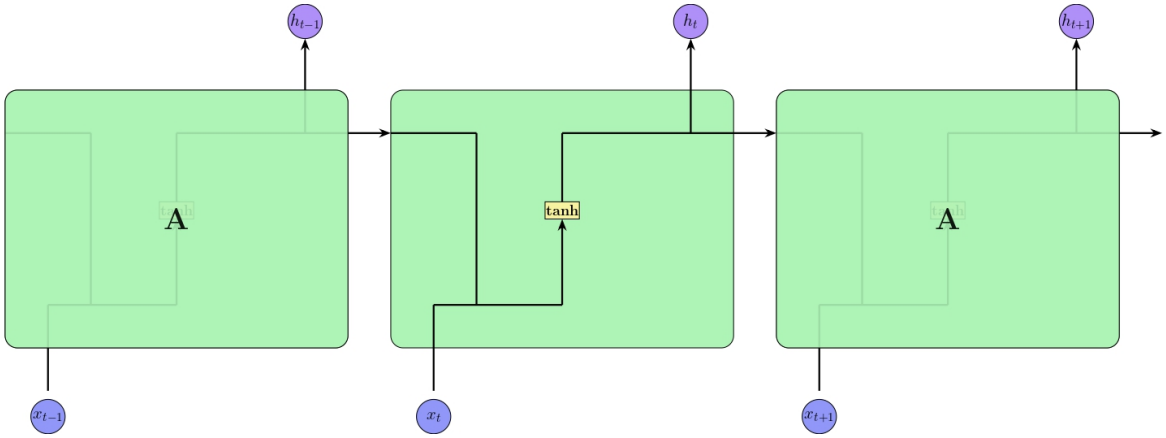


Slika 4.4. Standardni RNN modul [13]

Najveći nedostatak RNN-ova je to što množenjem matrica težina dolazi do iščezavanja gradijenta [9]. Ideja gradijentnog spusta jest da se, krenuvši od neke početne točke (početnog vektora w), postepeno "spuštamo" niz površinu funkcije $E(w \mid D)$ u smjeru suprotnome od gradijenta u točki w. To se ponavlja dok gradijent ne dođe do nule ili dovoljno blizu nuli. Formalno, težine w u svakoj iteraciji gradijentnog spusta ažuriraju se na ovaj način:

$$w \leftarrow w - \eta \bigtriangledown E(w \mid D) \tag{4.21}$$

gdje je η stopa učenja koja odreduje veličinu koraka koje radimo spuštajući se prema minimumu [14]. Iz jednadžbe se vidi da iščezavanjem gradijenta novi parametri modela ostaju jednaki starima te model prestaje učiti [15].



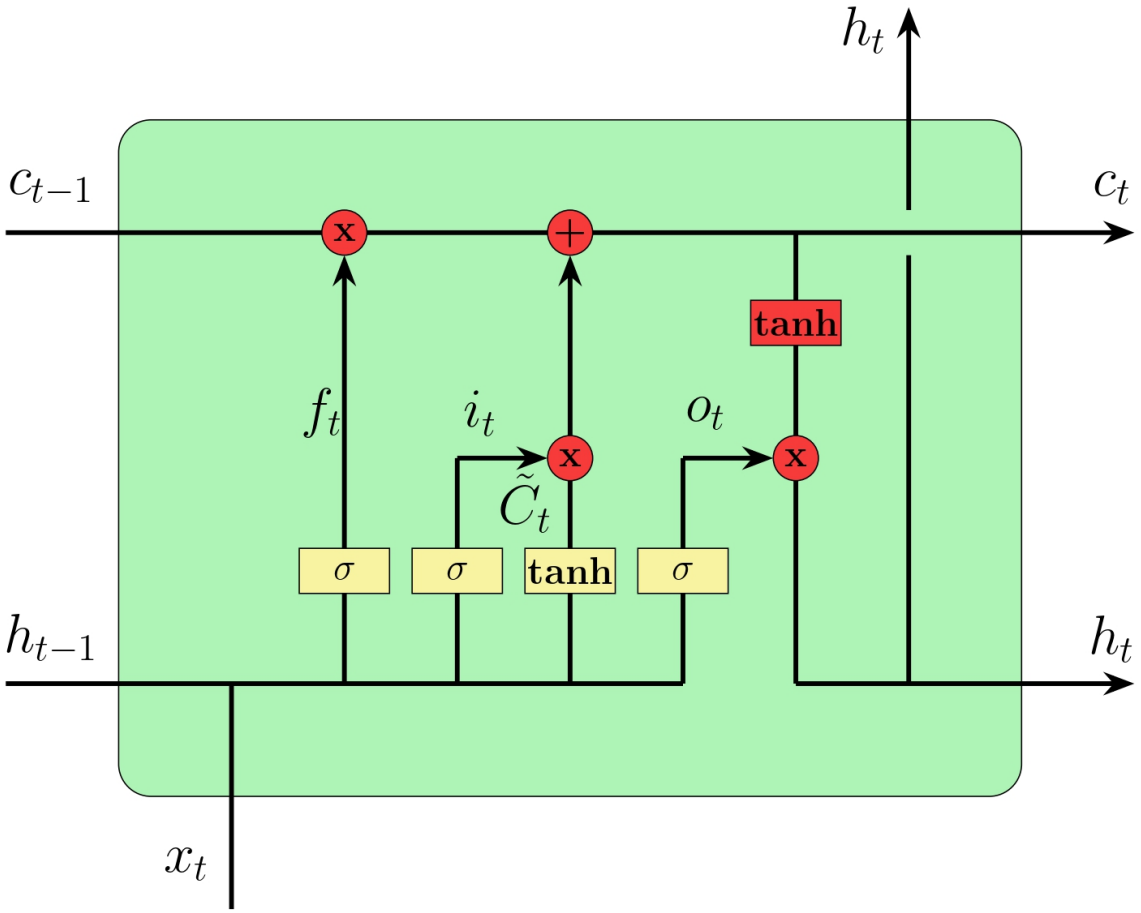
Slika 4.5. Niz standardnih RNN modula

Mreža s dugom kratkoročnom memorijom

Mreža s dugom kratkoročnom memorijom (engl. *long short term memory*, LSTM) posebna je vrsta RNN-a, sposobna učiti dugoročne ovisnosti. Uveli su ih Hochreiter i Schmidhuber (1997.), a mnogi su ih ljudi kasnije unaprijedili i popularizirali. Izvrsno funkcionira na velikom broju problema i danas se koriste za rješavanje čitavog spektra problema [15].

LSTM-ovi su izričito oblikovani kako bi izbjegli problem dugoročne ovisnosti. Pamćenje informacija kroz duže vremenske periode praktički je njihovo zadano ponašanje, a ne nešto s čime se bore da nauče [15]!

Sve povratne neuronske mreže imaju oblik lanca ponavljajućih modula neuronske mreže. U standardnim RNN-ovima ovaj ponavljajući modul ima vrlo jednostavnu strukturu, kao što je jedan sloj tanh [15]. To je vidljivo na slikama 4.4. i 4.5.



Slika 4.6. Modul LSTM

Ćelija LSTM-a zaboravlja dio informacije iz prethodnog stanja:

$$f_t = \sigma(W_{fhh}h^{(t-1)} + W_{fxh}x^{(t)} + b_{fh}) \tag{4.22}$$

Ćelija LSTM-a propušta samo podskup ulaza:

$$i_t = \sigma(W_{ihh}h^{(t-1)} + W_{ixh}x^{(t)} + b_{ih}) \tag{4.23}$$

Izraz za računanje doprinosa stanju ćelije $\hat{c}$:

$$\hat{c}^{(t)} = tanh(W_{chh}h^{(t-1)} + W_{cxh}x^{(t)} + b_{ch}) \tag{4.24}$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \hat{c}^{(t)} \tag{4.25}$$

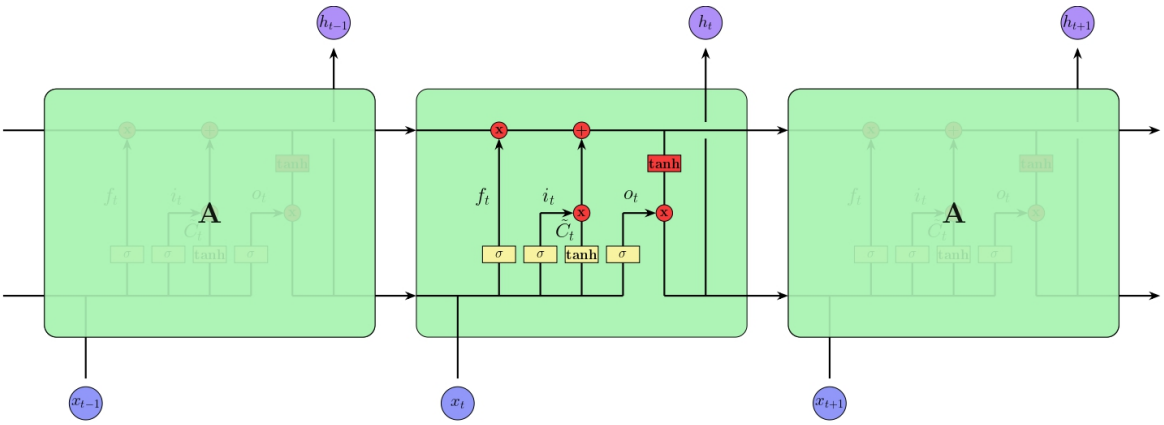
Logički vektor $o^{(t)}$ nazivamo izlaznim vratima:

$$o_t = \sigma(W_{ohh} h^{(t-1)} + W_{oxh} x^{(t)} + b_{oh}) \tag{4.26}$$

Skriveno stanje računamo kao funkciju stanja ćelije:

$$h^{(t)} = \sigma(W_{ohh} h^{(t-1)} + W_{oxh} x^{(t} + b_{oh}) \tag{4.27}$$

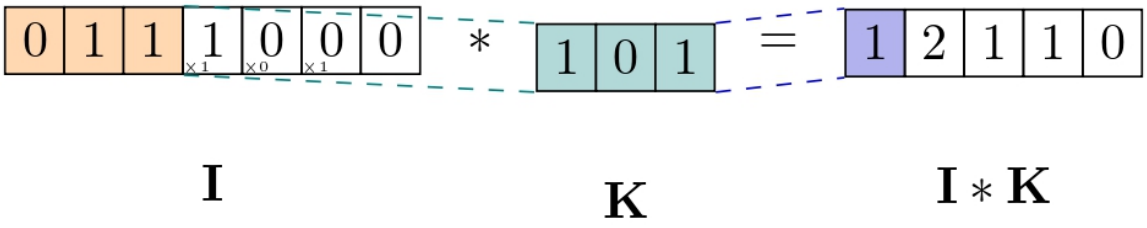
LSTM-ovi također imaju ovu lančanu strukturu (slika 4.7.), ali ponavljajući modul ima drugačiju strukturu. Umjesto da ima samo jedan sloj neuronske mreže, postoje četiri sloja, koji međusobno djeluju na vrlo poseban način [15]. S četiri dodatna sloja (žuto obojani na slikama 4.6. i 4.7.) imamo i toliko puta više parametara naspram jednostavnih modela povratnih neuronskih mreža.



Slika 4.7. Niz LSTM modula

4.2.3. Konvolucijske neuronske mreže

Konvolucija je operacija nad dva vektora, matrice ili tenzora, koja vraća treći vektor, matricu ili tenzor [10]. Slike 4.8. i 4.9. ilustriraju dva koraka te operacije.



Slika 4.8. Prvi korak 1D konvolucije nad ulaznim nizom I [16]

U strojnom učenju, pod konvolucijom najčešće podrazumijevamo unakrsnu korelaciju [17]:

$$h(t) = (w * x)(t) = \int_{D(w)} w(\tau)x(t+\tau)d\tau \tag{4.28}$$

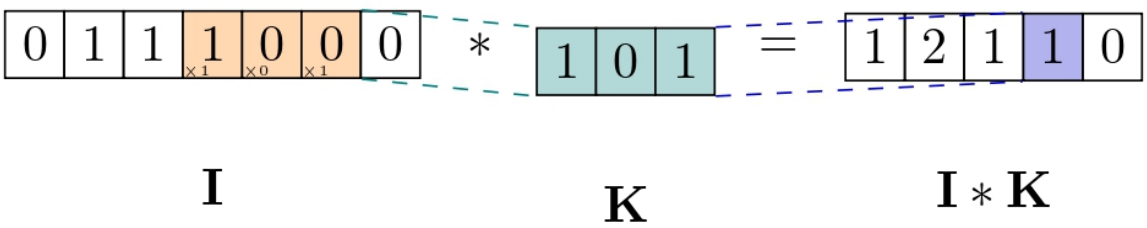
U diskretnom slučaju umjesto integrala koristimo zbroj [17]:

$$h(t) = (w * x)(t) = \sum_{\tau=-\infty}^{\infty} w(\tau)x(t+\tau) \tag{4.29}$$

Pretpostavljamo da je domena ulaza i jezgre konačan skup, tj. da su funkcije x i w izvan domene jednake 0 [17]:

$$h(t) = (w * x)(t) = \sum_{\tau=\tau_{min}}^{\tau_{max}} w(\tau)x(t+\tau) \tag{4.30}$$

Konvolucija (odnosno unakrsna korelacija) nam je zanimljiva kao diferencijabilna operacija sa slobodnim parametrima. Jednadžbe pokazuju da jezgru w možemo koristiti za ekstrakciju lokalnih značajki iz signala x [17].

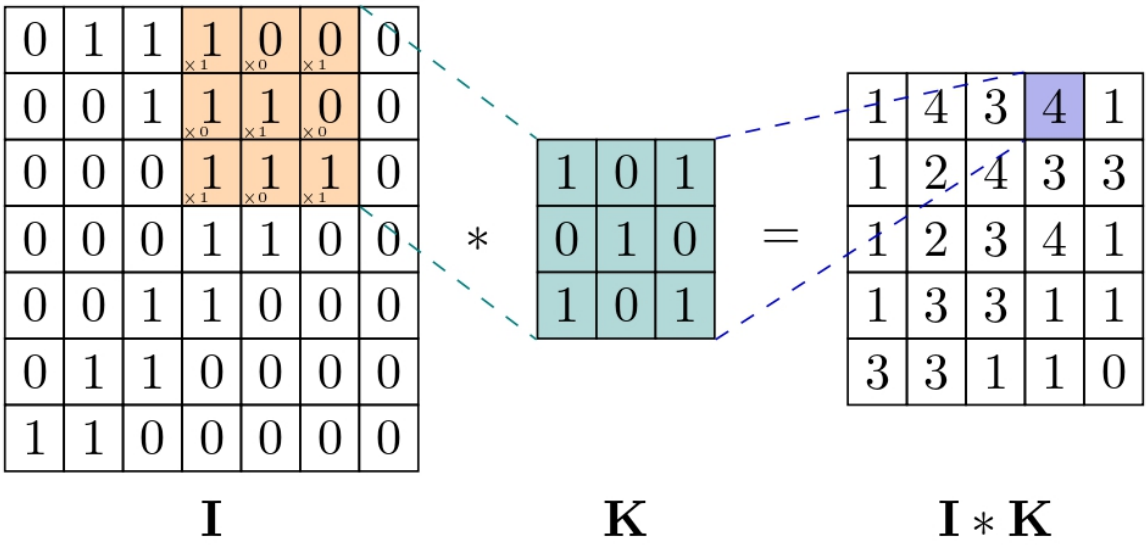


Slika 4.9. Četvrti korak 1D konvolucije nad ulaznim nizom I [16]

Korelacija se može primjenjivati kroz više dimenzija, npr. ako je argument ulaza dvodimenzionalan, kao u slučaju slika [17]:

$$S(i,j) = (K * I)(i,j) = \sum_{m=m_{min}}^{m_{max}} \sum_{n=n_{min}}^{n_{max}} K(m,n) \cdot I(i+m, j+n) \tag{4.31}$$

Primjer takve konvolucije može se vidjeti na slici 4.10., gdje postoji dvodimenzionalni 7×7 ulaz I i dvodimenzionalnu 3×3 jezgru K. Rezultat konvolucije je dvodimenzionalna matrica 5×5, u oznaci $I * K$.



Slika 4.10. Primjer 2D konvolucije [16]

5. Arhitektura sustava

Sustav je podjeljen u dvije glavne komponente:

- proizvođač (engl. *producer*)
- radnik (engl. *worker*)

Proizvođač (na slici 5.1. *crypto-bot-producer*) zadužen je za:

- stvaranje konfiguracije
- brisanje konfiguracije
- proizvodnju podataka u sustavu za prijenos podataka Kafka

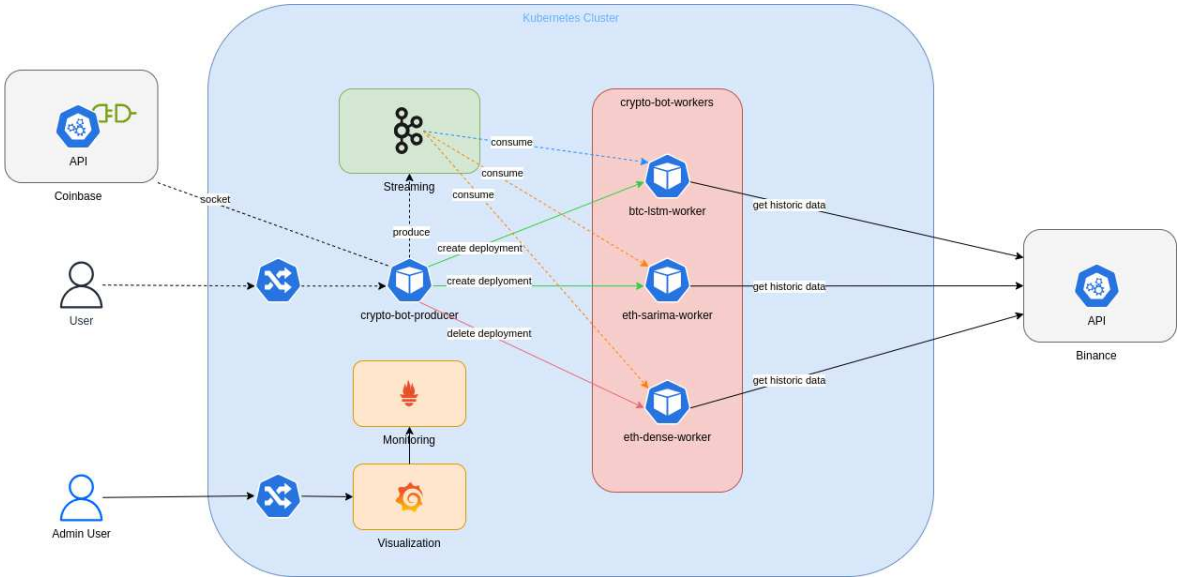
Gore navedena konfiguracija sadrži nekoliko informacija o modelu i izabranoj valuti koju pratimo.

Kod kreiranja konfiguracije dinamički se kreira i jedan Kubernetes *deployment* s jednom instancom aplikacije radnika.

Radnik (na slici 5.1. *crypto-bot-worker*) nakon što je kreiran od strane prozvođača sekvencijalno odrađuje sljedeće zadatke:

- dohvat povijesnih zapisa o odabranoj valuti
- priprema podataka
- stvaranje i učenje modela
- predikcija budućih cijena valute

Opisana arhitektura najsličnija je arhitekturi mikrousluga.



Slika 5.1. Arhitektura sustava

5.1. Helm chart

Helm chart je kolekcija datoteka koja opisuje resurse okoline Kubernetes. Sastoji se od četiri glavne komponente [18]:

- ***chart*** - *Chart.yaml* definira meta podatke kao što su ime *charta*, verzije, ovisnosti...
- **vrijednosti** - *values.yaml* je skup predefinirane konfiguracije.
- **obrasci** - *templates/* sadrži obrasce resursa koji se prvo popunjuju vrijednostima iz datoteke *values.yaml* te onda kreiraju kod instalacije *charta*.
- **direktorij *chartova*** - *charts/* pohranjuje sve *chartove* o kojima smo zavisni i koji se sami instaliraju prilikom instalacije našeg *charta*.

U idućem isječku koda 1 može se vidjeti korištena konfiguracija.

Isječak koda 1: Datoteka *values.yaml*

```yaml
apiVersion: v2
name: cryptobot
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: "1.0.2"
dependencies:
  - name: kafka
    repository: https://charts.bitnami.com/bitnami
    tags:
      - bitnami-kafka
    version: 26.4.1
    condition: kafka.install
  - name: kube-prometheus-stack
    repository: https://prometheus-community.github.io/helm-charts
    tags:
      - kube-prometheus-stack
    version: 54.2.2
    condition: prometheus.install
  - name: common
    repository: https://charts.bitnami.com/bitnami
    tags:
      - bitnami-common
    version: 2.13.1
```

5.1.1. Proizvodnja podataka

Kafka je sustav za upravljanje i obradom tokovima podataka u stvarnom vremenu. Kafka osigurava skalabilan i visoko dostupan sustav koji osigurava sekvencijalnu i nerepetitivnu dostavu poruke.

Sustav Kafka sastoji se od sljedećih ključnih komponenti: proizvođača koji šalje podatke, potrošača koji čita podatke, poslužitelja (*broker*) koji pohranjuje i poslužuje podatke. U ovom radu proizvođač je komponenta *crypto-bot-producer*, a potrošač je *crypto-bot-worker*.

Isječak koda 2: Stvaranje poruke

```
    def produce(self, topic, message):
        logging.info("sending message to topic: " + topic)
        (self.producer
         .send(topic=topic, value=message)
         .add_callback(on_send_success)
         .add_errback(on_send_error))
        self.producer.flush()
```

Isječkom koda 2 je prikazan kod za stvaranje poruka u sustavu Kafka, a isječkom koda 3 je prikazano konzumiranje istih.

Isječak koda 3: Konzumiranje poruke

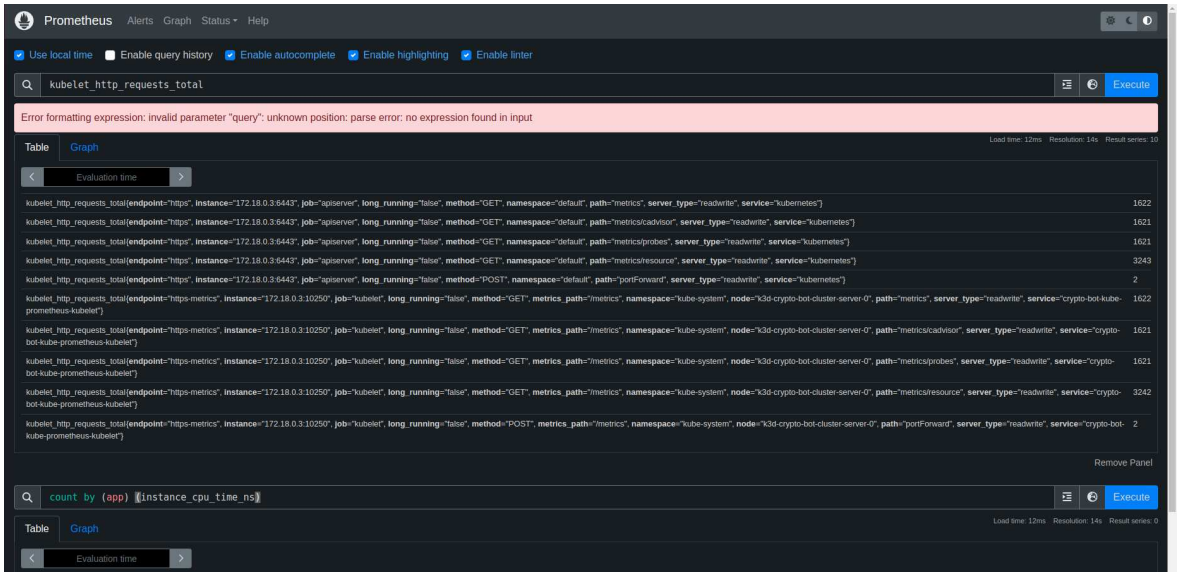
```
    async def consume(self):
        await self.consumer.start()

        try:
            async for message in self.consumer:
                logging.info("message received to topic: " +
                 ↪  message.topic)
                logging.info("message: " + message.value)
        except Exception as e:
            logging.error("failed while consuming messages", e)
            await self.consumer.stop()

```

5.1.2. Praćenje performanci sustava

Prometheus je programski alat za praćenje, analizu i pohranu podataka o performancama sustava.

Prometheus je u ovaj sustav dodan kao zavisan *chart* "kube-prometheus-stack" što je vidljivo u njegovoj definiciji, prema isječku koda 1.

Slika 5.2. prikazuje sučelje alata Prometheus.



Slika 5.2. Sučelje programskog alata Prometheus

5.1.3. Vizualizacija

Grafana je jedan od zavisnih *chartova charta* "kube-prometheus-stack" koji pruža kompletnu uslugu prikupljanja i vizualizacije podataka.

Grafana vizualizira sve podatke prikupljene od strane Prometheusa i pruža interaktivno i dinamično sučelje, kao npr. 5.3.

Slika 5.3. Sučelje programskog alata Grafana

6. Skalabilnost

Skalabilnost se ponekad definira kao "lakoća s kojom se sustav ili komponenta mogu modificirati kako bi odgovarali problematičnom području." Skalabilan sustav ima tri jednostavne karakteristike [19]:

- Sustav može prihvatiti povećanu upotrebu,
- Sustav može prihvatiti povećani skup podataka,
- Sustav je održiv i radi s razumnim performansama.

Skalabilnost nije samo brzina. Izvedba i skalabilnost sustava se razlikuju, ali su povezane jedna s drugom. Izvedba mjeri koliko brzo i učinkovito sustav može izvršiti određene računalne zadatke, dok skalabilnost mjeri trend izvedbe [19].

Vertikalno skaliranje (engl. *vertical scaling*) ili skaliranje prema gore odnosi se na maksimizaciju resursa jedne jedinice kako bi se povećala njezina sposobnost za rukovanje sve većim opterećenjem. U sklopovskim terminima, to uključuje dodavanje procesorske snage i memorije fizičkom stroju na kojem radi poslužitelj. U programskim terminima, skaliranje prema gore može uključivati optimizaciju algoritama i aplikacijskog koda. Optimizacija sklopovskih resursa, kao što je paralelizacija ili optimiran broj pokrenutih procesa, također se smatra tehnikama skaliranja prema gore [19].

Horizontalno skaliranje (engl. *horizontal scaling*) ili skaliranje prema van odnosi se na povećanje resursa dodavanjem jedinica u sustav. To znači dodavanje više jedinica manje kapaciteta umjesto dodavanja jedne jedinice većeg kapaciteta. Zahtjevi za resursima tada se raspoređuju na više jedinica, čime se smanjuje prekomjerno opterećenje na jednom stroju [19].

Programsko okruženje Kubernetes pruža ručno i automatsko sučelje za skaliranje.

Ručno skaliranje izvodi se pomoću naredbi u sučelju komandne linije.

Automatsko skaliranje je realizirano dodavanjem resursa *HorizontalPodAutoscaler* za horizontalno skaliranje ili *VerticalPodAutoscaler* za vertikalno skaliranje.

U radu je realizirana verzija horizontalnog skaliranja korištenjem resursa *HorizontalPodAutoscaler*.

7. Rezultati i rasprava

Performance modela mjerene su nad tri statistička modela (ARIMA, SARIMA i Prophet) te nad četiri duboke neuronske mreže (potpuno povezana duboka neuronska mreža (anotirana s *Dense*), konvolucijska neuronska mreža (anotirana s *Conv*) te dvije povratne neuronske mreže (LSTM i GRU). Za referentnu točku korišten je naivan pristup (anotiran s *Naive*) koji za predikciju budućih vrijednosti koristi zadnju viđenu vrijednost.

Za mjerenje performanci modela korišten je skup podataka o paru BTC/USDT od 17. kolovoza 2017. do 26. lipnja 2024. Skup podataka se sastoji od 59.945 zapisa u razmaku od sat vremena opisanih u poglavlju 3. Ispitivanje modela je odrađeno korištenjem jednake duljine vremenskih prozora duljine 24 zapisa dok je vrijednost horizonta kroz ispitne slučajeve postepno rasla. Rezultati modela vidljivi su u tablici 7.1. Predikcije modela za duljinu horizonta od jedne informacije prikazane su slikom 7.1.

Za funkciju gubitka korištena je formula srednje kvadratne pogreške (engl. *mean squared error*, MSE). Dodatno je mjerena srednja apsulutna pogreška (engl. *mean absolute error*, MAE), srednja apsolutna postotna pogreška (engl. *mean absolute percentage error*, MAPE) te srednja kvadratna logaritamska pogreška (engl. *mean squared logarithmic error*, MSLE).

Jednadžbe korištenih metrika su prikazane u nastavku:

Srednja apsolutna pogreška (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7.1}$$

Srednja kvadratna pogreška (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7.2}$$

Srednja apsolutna postotna pogreška (MAPE)

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7.3}$$

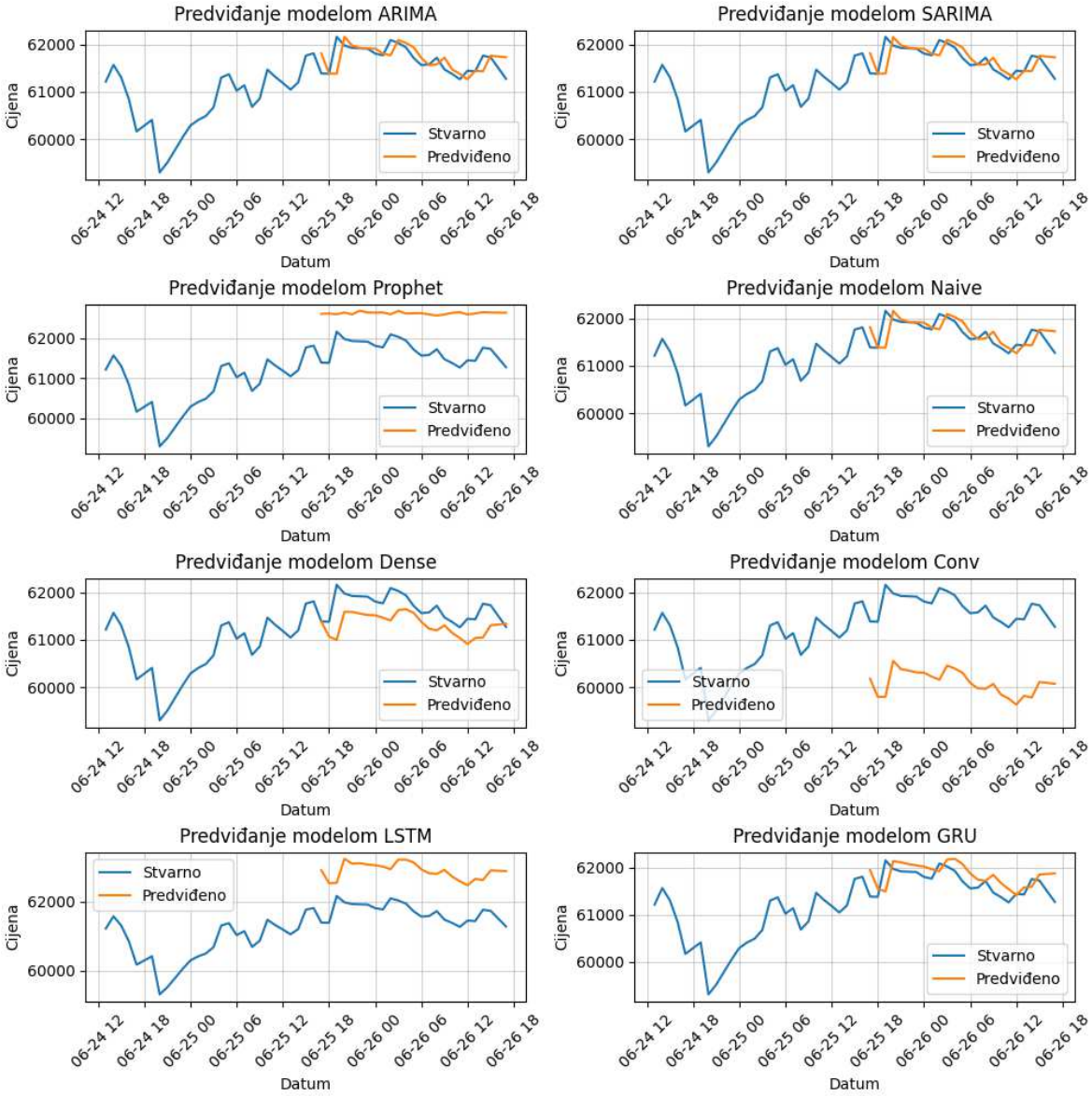
Srednja kvadratna logaritamska pogreška (MSLE)

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(1 + y_i) - \log(1 + \hat{y}_i))^2 \tag{7.4}$$

Tablica 7.1. Mjere perfromanci za različitu vrijednost horizonta

Horizont	Model	MAE	MSE	MAPE	MSLE
			Statistički modeli		
	ARIMA	53,625	5771,073	0,000833	0,00000139
	SARIMA	**53,295**	**5735,074**	**0,000828**	**0,00000138**
	Prophet	1746,235	3060319,338	0,0271	0,000759
1			**Duboke neuronske mreže**		
	Naive	**111,522**	**40639,994**	**0,0033**	0,0000295
	Dense	200,014	92792,798	0,0046	0,0000392
	Conv	809,453	800840,552	0,0226	0,000591
	LSTM	1550,223	3574462,676	0,0359	0,001445
	GRU	173,605	72083,340	0,0041	**0,0000333**
			Statistički modeli		
	ARIMA	146,653	**40699,070**	0,002306	0,000010067
	SARIMA	**146,397**	40711,143	**0,002302**	**0,000010064**
	Prophet	1234,974	1858592,441	0,0192	0,000461
2			**Duboke neuronske mreže**		
	Naive	34203,606	1170247379,666	0,5349	0,586203
	Dense	470,071	407392,180	0,0101	0,000143
	Conv	19447,229	586453495,842	0,9014	0,447139
	LSTM	20759,757	685160023,775	1,0736	0,569308
	GRU	**226,892**	**115755,546**	**0,0052**	**0,0000483**
			Statistički modeli		
	ARIMA	190,393	79830,555	**0,0030**	0,0000199
	SARIMA	**189,989**	**79704,010**	**0,0030**	**0,0000198**
	Prophet	1105,515	1611531,435	0,0172	0,000400
4			**Duboke neuronske mreže**		
	Naive	34041,722	1159315410,337	0,5337	0,582225
	Dense	345,100	242333,291	0,0078	0,0000970
	Conv	21022,353	695504146,034	1,1017	0,586584
	LSTM	20599,932	681351942,381	1,0605	0,562117
	GRU	**200,646**	**109324,131**	**0,0047**	**0,0000500**
			Statistički modeli		
	ARIMA	248,936	120491,367	0,0039	0,0000299
	SARIMA	**248,573**	**120269,968**	**0,0039**	**0,0000299**
	Prophet	1178,458	1735282,943	0,0184	0,000431
6			**Duboke neuronske mreže**		
	Naive	34111,852	1164060688,252	0,5342	0,583899
	Dense	241,291	147651,908	0,0060	0,0000733
	Conv	20764,979	681185687,582	1,0690	0,564899
	LSTM	20704,171	683718655,377	1,0691	0,566839
	GRU	**211,941**	**133940,220**	**0,0050**	**0,0000620**

U tablici 7.1. vidi se kako statistički model SARIMA konzistentno najbolji za sve vrijednosti horizonta. Kod dubokih modela najboljim se pokazao model povratnih neuronskih mreža GRU. Modeli LSTM i *Conv* pokazuju usporedive vrijednosti metrika s ostalim modelima za duljinu horizonta od jedne informacije dok su za veće iznose horizonta daleko lošiji. Zanimljivo je kako je naivan model bolji od dubokih neuronskih mreža za duljinu horizonta od jedne informacije.



Slika 7.1. Predikcije modela

8. Zaključak

Iz prikazanih rezultata može se zaključiti kako su statistički modeli konzistentniji od dubokih neuronskih mreža.

Specijalizirani modeli za vremenske nizove kao što je GRU ostvaruje bolje rezultate od ostalih dubokih neuronskih mreža.

Problem prognoziranja budućih cijena valuta financijskih tržišta pokazao se problematičan za većinu dubokih neuronskih mreža. Razlog je taj što financijski skupovi podataka pokazuju svojstva koja krše standardne pretpostavke primjene strojnog učenja [20] kao što je stacionarnost vremenskog niza kod ARIMA modela ili ovisnost budućih trendova o prošlima.

Arhitektura sustava za upogonjavanje modela i stvaranja automatizirane trgovine valutama financijskih tržišta se pokazala dosljednom zahtjevima predikcijskih modela.

Literatura

[1] S. G. E. Frederic S. Mishkin, *Financial markets and institutions*, 5. izd. Addison Wesley, 2005. [Mrežno]. Adresa: http://gen.lib.rus.ec/book/index.php?md5=591c1aae8c5a486f461d7232365f4b17

[2] Crypto.com, https://crypto.com/university/what-is-cryptocurrency, [mrežno; stranica posjećena: lipanj 2024.].

[3] M. Alvarez, *Market Data Explained: A Practical Guide to Global Capital Markets Information. (The Elsevier and Mondo Visione World Capital Markets)*, 1. izd., 2006. [Mrežno]. Adresa: http://gen.lib.rus.ec/book/index.phpmd5=08ee56d925a46f5af0aae5108a994c96

[4] FoxCharles, https://tex.stackexchange.com/questions/553833/plotting-financial-data-with-pgfplots, [mrežno; stranica posjećena: lipanj 2024.].

[5] Peter Foy, https://blog.mlq.ai/time-series-tensorflow-windows-horizons/, [mrežno; stranica posjećena: lipanj 2024.].

[6] J. Liu, F. Yu, i H. Song, "Application of sarima model in forecasting and analyzing inpatient cases of acute mountain sickness", *BMC Public Health*, 2023.

[7] S. J. Taylor i B. Letham, "Forecasting at scale", 2017., str. 5–7.

[8] M. Peixeiro, *Time Series Forecasting in Python*. Manning, 2022. [Mrežno]. Adresa: http://gen.lib.rus.ec/book/index.php?md5=49FD63FD2D4BFE75FCD83B03020AEB42

[9] J. Shi, M. Jain, i G. Narasimhan, *Time Series Forecasting (TSF) Using Various Deep Learning Models*, 2022. [Mrežno]. Adresa: https://arxiv.org/abs/2204.11115

[10] B. Liquet, S. Moka, i Y. Nazarathy, *Mathematical Engineering of Deep Learning*. CRC Press, 2024.

[11] Izaak Neutelings, https://tikz.net/neural_networks/, [mrežno; stranica posjećena: lipanj 2024.].

[12] IBM, https://www.ibm.com/cloud/learn/recurrent-neural-networks, [mrežno; stranica posjećena: lipanj 2024.].

[13] user121799, https://tex.stackexchange.com/questions/494139/how-do-i-draw-a-simple-recurrent-neural-network-with-goodfellows-style, [mrežno; stranica posjećena: lipanj 2024.].

[14] Jan Šnajder, https://www.fer.unizg.hr/_download/repository/SU1-2021-P06-LogistickaRegresija.pdf, [mrežno; stranica posjećena: lipanj 2024.].

[15] Christopher Olah, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, [mrežno; stranica posjećena: lipanj 2024.].

[16] Janosh Riebesell, https://tikz.net/conv2d/, [mrežno; stranica posjećena: lipanj 2024.].

[17] J. Krapac i S. Šegvić, "Konvolucijski modeli", prezentacije predmeta Duboko učenje 1.

[18] Red Hat, https://www.redhat.com/en/topics/devops/what-is-helm, [mrežno; stranica posjećena: lipanj 2024.].

[19] D. C. Rajapakse, *A Fresh Graduate's Guide to Software Development Tools and Technologies*. National University of Singapore, 2012.

[20] M. L. de Prado, "The 10 reasons most machine learning funds fail", [mrežno; stranica posjećena: lipanj 2024.].

Sažetak

Skalabilno predviđanje vremenskih nizova na financijskim tržištima u stvarnom vremenu zasnovano na dubokom učenju

Stjepan Ruklić

Ovaj rad opisuje razvoj sustava za ovladavanje kompleksnim financijskim tržištima primjenom predikcijskih modela. Analizirani su povijesni podaci o cijenama te su uspoređene performanse statističkih modela (ARIMA, SARIMA, Prophet) i modela dubokog učenja. Predložena arhitektura razdvaja odgovornosti prikupljanja podataka i predikcije od orkestriranja i upravljanja resursima i omogućuje skalabilno predviđanje u stvarnom vremenu. Pored toga, sustav je oblikovan da se prilagodi različitim tržišnim uvjetima, povećavajući njegovu pouzdanost i efikasnost. Primjena statističkih modela se pokazala sigurnijom, jednostavnijom i u većini slučajeva točnijom od dubokih od dubokih neuronskih mreža.

Ključne riječi: vremenski nizovi podataka; duboko učenje; neuronske mreže; skalabilnost; financijska tržišta; dionice; kriptovalute

Abstract

Scalable real-time forecasting of financial markets time series based on deep learning

Stjepan Ruklić

This paper describes the development of a system for mastering complex financial markets using prediction models. Historical price data were analyzed and the performance of statistical models (ARIMA, SARIMA, Prophet) and deep learning models were compared. The proposed architecture separates data collection and prediction responsibilities from orchestration and resource management, and enables scalable and real-time forecasting. In addition, the system is designed to adapt to different market conditions, increasing its reliability and efficiency. The application of statistical models proved to be safer, simpler and more accurate than deep neural networks.

Keywords: time series data; deep learning; neural networks; scalability; financial markets; stocks; cryptocurrencies