



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ



Данило Каћански

**Примена алгоритама машинског  
учења за предикцију вредности  
Ethereum криптовалуте**

ДИПЛОМСКИ РАД  
- Основне академске студије -

Нови Сад, 2024.

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска публикација		
Тип записа, ТЗ:	Текстуални штампани документ/ЦД		
Врста рада, ВР:	Завршни-bachelor рад		
Аутор, АУ:	Данило Каћански		
Ментор, МН:	Проф. др Дарко Чапко		
Наслов рада, НР:	Примена алгоритама машинског учења за предикцију вредности Ethereum криптовалуте		
Језик публикације, ЈП:	Српски (латиница)		
Језик извода, ЈИ:	Српски/Енглески		
Земља публиковања, ЗП:	Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2024		
Издавач, ИЗ:	Ауторски репрнт		
Место и адреса, МА:	Факултет Техничких Наук (ФТН), Д. Обрадовића 6, 21000 Нови Сад		
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/спика/графика/прилога)	10/33/27/6/23/12/1		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарство и аутоматика		
Предметна одредница/Кућне речи, ПО:	Машинско учење, предикција, временске серије, криптовалуте		
УДК			
Чува се, ЧУ:	Библиотека ФТН, Д. Обрадовића 6, 21000 Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	Напредак у области предикције временских серија отворио је могућност за примену различитих алгоритама у финансијском сектору, и на тржишту криптовалута. Циљ истраживања је да утврди ефикасност различитих модела у предвиђању кретања цене Ethereum-a (ETH) и да се анализира како сложеност модела и дужина периода предикције утичу на њихове перформансе. Ово истраживање за циљ има да допринесе бољем разумевању примене алгоритама машинског учења и да понуди практичне смернице за финансијске аналитичаре и инвеститоре.		
Датум прихватања теме, ДП:			
Датум одbrane, ДО:	25.09.2024.		
Чланови комисије, КО:	Председник:	др Срђан Вукмировић, редовни професор	
	Члан:	др Александар Ердељан, редовни професор	Потпис ментора
	Члан, ментор:	др Дарко Чапко, редовни професор	

## KEY WORDS DOCUMENTATION

Accession number, <b>ANO:</b>		
Identification number, <b>INO:</b>		
Document type, <b>DT:</b>	Monographic publication	
Type of record, <b>TR:</b>	Textual material, printed/CD	
Contents code, <b>CC:</b>	Bachelor thesis	
Author, <b>AU:</b>	Danilo Kaćanski	
Mentor, <b>MN:</b>	Darko Čapko, PhD, full professor	
Title, <b>TI:</b>	Application of Machine Learning Algorithms for Predicting the Value of Ethereum Cryptocurrency	
Language of text, <b>LT:</b>	Serbian (latin script)	
Language of abstract, <b>LA:</b>	Serbian/English	
Country of publication, <b>CP:</b>	Serbia	
Locality of publication, <b>LP:</b>	Vojvodina	
Publication year, <b>PY:</b>	2024	
Publisher, <b>PB:</b>	Author reprint	
Publication place, <b>PP:</b>	Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad	
Physical description, <b>PD:</b> <small>(chapters/pages/ref./tables/pictures/graphs/applications)</small>	10/33/27/6/23/12/	
Scientific field, <b>SF:</b>	Electrical and Computer Engineering	
Scientific discipline, <b>SD:</b>	Control and Automation Engineering	
Subject/Key words, <b>S/KW:</b>	Machine learning, prediction, time series, cryptocurrencies	
<b>UC</b>		
Holding data, <b>HD:</b>	Library of the Faculty of Technical Sciences, D. Obradovića 6, 21000 Novi Sad	
Note, <b>N:</b>		
Abstract, <b>AB:</b>	Advancements in the field of time series prediction have opened up possibilities for the application of various algorithms in the financial sector and the cryptocurrency market. The aim of this research is to determine the effectiveness of different models in predicting the price movements of Ethereum (ETH) and to analyze how model complexity and the length of the prediction period affect their performance. This research aims to contribute to a better understanding of the application of machine learning algorithms and to offer practical guidelines for financial analysts and investors.	
Accepted by the Scientific Board on, <b>ASB:</b>		
Defended on, <b>DE:</b>	25.09.2024,	
Defended Board, <b>DB:</b>	President: Srđan Vukmirović, PhD, full professor Member: Aleksandar Erdeljan, PhD, full professor Member, Mentor: Darko Čapko, PhD, full professor	
		Mentor's sign

	<b>УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6</b> <b>ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА</b>	Датум:   Лист/Листова:   3/38
---	---	---

(Податке уноси предметни наставник - ментор)

Врста студија:	<input type="checkbox"/> Основне академске студије		
Студијски програм:	Рачунарство и аутоматика		
Руководилац студијског програма:	Проф. др Милан Видаковић		

Студент:	Данило Каћански	Број индекса:	PA26 - 2020
Област:	Машинско учење		
Ментор:	Проф. др Дарко Чапко		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;
- литература

### НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

**Примена алгоритама машинског учења за предикцију вредности Ethereum криптовалуте**

### ТЕКСТ ЗАДАТКА:

- Упознati сe сa различитим алгоритмима за предикцију временских серија
- Пројектовати, имплементирати и тестирати алгоритме над конкретној временској серији кретања цене Ethereum-а (ETH)
- Анализа и упоређивање добијених резултата предикције за једнодневну, десетодневну и тридесетодевну прогнозу
- Давање препорука за примену ових алгоритама у реалним условима

Руководилац студијског програма:	Ментор рада:

Примерак за:  - Студента;  - Ментора

## Spisak korišćenih skraćenica

<i>Skraćenica</i>	<i>Značenje skraćenice</i>
ETH	Ethereum
ARIMA	AutoRegressive Integrated Moving Average
SARIMAX	Seasonal AutoRegressive Integrated Moving Average with eXogenous variables
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
MLP	Multi-Layer Perceptron
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
SMA	Simple Moving Average
EWMA	Exponentially Weighted Moving Average
SES	Simple Exponential Smoothing

## Sažetak

Tržište akcija danas ima izrazito značajan uticaj na globalnu ekonomiju. Promene u cenama akcija igraju ključnu ulogu u određivanju zarade investitora, što čini analizu i predviđanje njihovog kretanja veoma važnom i primenljivom oblasti istraživanja. Ovaj rad se fokusira na predikciju kretanja cena kriptovaluta korišćenjem dnevnih zatvornih cena (*Closing price*). Cilj je da se što preciznije predviđi buduća cena kriptovalute, što može značajno doprineti donošenju investicionih odluka. Predložene metode i modeli pružaju dubinski uvid u dinamiku tržišta kriptovaluta i potencijalne pravce njihovog kretanja.

## Sadržaj

Sažetak .....	1
Uvod .....	4
Predikcija na dnevnom nivou .....	6
Naivno predviđanje .....	6
Moving Average tehnike .....	6
Simple Moving Average (SMA) .....	7
Exponentially-Weighted Moving Average (EWMA) .....	8
Simple Exponential Smoothing (SES) .....	8
Autoregressive Integrated Moving Average (ARIMA) .....	9
AR(p) .....	9
MA(q) .....	9
ARMA(p,q) .....	10
I(d) .....	10
ARIMA(p, d, q) .....	10
Stacionarnost i selekcija hiperparametara .....	10
Hiperparametar d .....	11
Hiperparametar q .....	13
Hiperparametar p .....	15
Auto ARIMA .....	17
Duboko učenje .....	18
Artificial Neural Network (ANN) .....	18
Izdvajanje i skaliranje podataka .....	20
Kreiranje skupa podataka za učenje sa nadzorom .....	20
Kreiranje ANN modela .....	21
Podešavanje hiperparametara i predikcija .....	22
Sređivanje podataka i evaluacija modela .....	22
Recurrent Neural Network (RNN) .....	22
Otpakovana RNN i Nestajući(Eksplodirajući) gradijent .....	23
Long Short-Term Memory Neural Network (LSTM) .....	24
Rezultati jednodnevne predikcije .....	26
Desetodnevna predikcija .....	26
Rezultati desetodnevne predikcije .....	27

Tridesetodnevna predikcija.....	27
Rezultati tridesetodnevne predikcije.....	28
Zajednički rezultati.....	28
Grafici modela višednevnih predikcija sa najboljim performansama.....	29
Desetodnevna predikcija .....	29
Tridesetodnevna predikcija.....	29
Zaključak .....	30
Reference .....	31
Podaci o kandidatu.....	33

## Uvod

Fluktacije cena su neizvesne zbog mnogih međusobno povezanih faktora koje utiču na njih. Neki od mogućih faktora su globalno stanje ekonomije, stopa nezaposlenosti, politike trgovanja u određenim zemljama, prirodne katastrofe itd. Svi ljudi koji se bave trgovinom deonica imaju za cilj da naprave profit i smanje rizike pomoći procene prošlog, trenutnog i budućeg stanja na berzi. Glavni zadatak je sakupiti raznolike informacije iz više izvora, spojiti ih u jedan sistem i kreirati pouzdan model koji će napraviti što preciznije predviđanje.

Predviđanje se može definisati kao prognoza nekog budućeg događaja, koja se zasniva na analizi istorijskih podataka. Obuhvata mnoge oblasti kao što su poslovanje, industrija, ekonomija, nauke o životnoj sredini i finansije, što je i fokus ovog rada.

Problemi predviđanja mogu se klasifikovati u tri kategorije:

- **Kratkoročno predviđanje** (od nekoliko sekundi do nekoliko meseci)
- **Srednjoročno predviđanje** (od jedne do dve godine)
- **Dugoročno predviđanje** (više od dve godine)

Frekvencija podataka koji se obrađuju, preuzetih sa sajta Yahoo Finance [1], je jedan podatak dnevno. Zbog toga je očigledno donje ograničenje predviđanja jednodnevna predikcija, dok gornja granica ne postoji. Ovaj rad se bavi kratkoročnim predviđanjem, preciznije jednodnevnom, desetodnevnom i tridesetodnevnom predikcijom.

Praktično svaki problem predviđanja uključuje analizu vremenskih serija. Vremenska serija je hronološki niz opservacija izabrane promenjive. U ovom slučaju, promenjiva je dnevna vrednost kriptovalute (*Ethereum/ETH*). [2] Analiza vremenskih serija pomaže u identifikovanju obrasca, trendova i ciklusa prisutnih u podacima. U slučaju berze, rano prepoznavanje „bull“ ili „bear“ marketa [3] pomaže u mudrom ulaganju novca. Daljom analizom obrazaca mogu se utvrditi najbolje kompanije za ulaganje u određenom periodu i zato ovo čini analizu vremenskih serija i predviđanje izuzetno važnim područjem istraživanja.

Postojeće metode za predviđanje cena akcija mogu se klasifikovati kao:

- **Fundamentalna analiza**
- **Tehnička analiza**
- **Predviđanje vremenskih serija** [4]

**Fundamentalna analiza** je vrsta investicione analize gde se vrednosti akcija kompanije procenjuju analizom njenih prodaja, prihoda, profita i drugih ekonomskih faktora. Ova metoda je najprikladnija za dugoročna predviđanja.

**Tehnička analiza** koristi istorijsku cenu akcija za identifikaciju buduće cene. Pokretni prosek (Moving average) je često korišćen algoritam za tehničku analizu. On uzima poslednjih  $n$  podataka vremenske serije i predviđa  $n+1$  na osnovu srednje vrednosti pređašnjih  $n$  podataka. Zatim, nakon njega, prvo logično proširenje je Eksponencijalno ponderisani pokretni prosek (Weighted Moving Average), gde se ponderišu poslednjih  $n$  podataka kako bi došli do boljih rezultata. Metode poput ovih su najprikladnije za kratkoročna predviđanja na nivou od 1 do 3 dana.

Treća metoda je analiza vremenskih serija, ono na čemu će biti akcenat u ovom radu. Ona uključuje dve klase algoritama:

- **Linearni modeli**
- **Nelinearni modeli**

**Linearni modeli** poput AR, ARMA, ARIMA i drugih koriste unapred definisane jednačine za prilagođavanje matematičkog modela. Glavni nedostatak linearnih modela u predviđanju vremenskih serija je što ne uzimaju u obzir skrivenu dinamiku prisutnu u podacima. Korelacije između različitih akcija nisu identifikovane ovim modelima, a takođe ukoliko je model identifikovan za jednu vremensku seriju, neće biti odgovarajući za druge vremenske serije. Iz tih razloga nije moguće identifikovati obrasce prisutne u podacima u celini. Iz tog razloga, za predviđanje sa najvećim uspehom se koriste nelinearni modeli.

**Nelinearni modeli** uključuju metode poput ARCH-a, GARCH-a, kao i algoritme mašinskog učenja. Duboke neuronske mreže se mogu smatrati nelinearnim aproksimatorima funkcija koje su sposobne da mapiraju nelinearne funkcije. U zavisnosti od potrebe koriste se različite vrste arhitektura neuronskih mreža. Ovo uključuje višeslojne perceptrone (MLP), rekurzivne neuronske mreže (RNN), dugoročno-kratkoročne memoriske neuronske mreže (LSTM), konvolucione neuronske mreže (CNN) itd. One su izuzetno primenjivane u raznim oblastima poput obrade slika, obrade prirodnog jezika i analize vremenskih serija.

**Burton Gordon Malkiel** [5], profesor ekonomije na Prinstonu, 1973. godine objavljuje knjigu pod nazivom „**A Random Walk Down Wall Street**” [6] u kojoj piše o različitim strategijama ulaganja u market. Zanimljiv je način na koji započinje svoju knjigu, gde govori da ekspert u trejdovanju deonica nema nikakvu prednost u odnosu na majmuna sa povezom na očima koji baca strelice na tablu za pikado, pomoću koje bira deonice u koje će ulagati. Baš iz ovog razloga nastaje hipoteza poznata kao: „**Random Walk Hypothesis**” [7]. Ovaj koncept je u matematici poznat još od ranije i predstavlja specijalan slučaj Markovljevog procesa, dok je konkretno hipoteza specifična baš za finansije i konkretno za cene deonica. Zbog ove hipoteze, 70-ih godina prošlog veka popularizuje se i „**Efficient Market Hypothesis**” [8], a obe hipoteze dovode do istog zaključka. Tržište se ne može pobediti, ili drugim rečima, iz perspektive ovog rada: Tržište je u potpunosti nasumično. Ova hipoteza je bila toliko popularna da je 1988. održan eksperiment od strane časopisa „**Wall Street Journal**”.[9] Sa jedne strane su okupljeni profesionalni trejderi deonica, dok su sa druge strane bili potpuni nepoznavaoci materije koji su birali, deonice u koje će da ulože gađajući strelicama u tablu za pikado. Profesionalni trejderi su dobijali u 60% slučajeva. To bi značilo da ipak treba verovati profesionalnim trejderima i ulagati novac kod njih. Međutim oni su nadmašili tržište tek u 51% slučajeva. Baš iz ovoga razloga je Burton predlagao da se ne koriste aktivni fondovi, jer ako se uračunaju još i naknade za aktivno trgovanje, taj broj postaje nešto niži od 50%. Iz tog razloga, ključna lekcija koju je Burton izneo u svojoj knjizi jeste prednost pasivnog ulaganja nad aktivnim.

Iako su ove hipoteze teške za osporiti, i dalje su hipoteze, a ne činjenice pa 1999. godine Andrew W. Lo i A. Craig MacKinlay [9] objavljuju knjigu „**A Non-Random Walk on Wall Street**” [10], koja je u direktnoj koliziji sa predašnje pomenutom knjigom koju je napisao Burton. Logičan zaključak je to da su obe ideje delimično tačne i netačne. U nekim slučajevima će Random Walk verovatno davati bolje rezultate od kompleksnih modela, dok će nekad ti modeli biti u prednosti. Zadatak ovog rada je da se pokušaju napraviti što bolji modeli, da se uporedi koliko su bolji ili lošiji od naivnog modela Random Walk-a, i kada i koliko je bolje njih koristiti za predikciju cena deonica, a u slučaju ovog rada kriptovaluta.

Ovde će se diskutovati o predviđanju vremenskih serija, ali i o metodama koje više spadaju u tehničku analizu. Pomoću dobijenih rezultata će moći da se zaključi, da li kompleksnost linearnih i nelinearnih modela koje se koriste za predikciju vremenskih serija automatski garantuje bolje performanse od mnogo jednostavnijih tehnika, jer bi intuitivno trebalo da je tako. U skladu sa tim, uvek ćemo se vraćati na najbitnije pitanje: Koliko je najbolji model bolji od naivnog predviđanja?

## Predikcija na dnevnom nivou

Mnogi radovi koji se bave predviđanjem cena deonica najčešće rade predviđanje na dnevnom nivou. Jednodnevna predikcija znači da je na raspolaganju pređašnjih  $v_i, i \in \{1, 2, \dots, k\}$  vrednosti pomoću kojih se predviđa sledeća vrednost  $v_{k+1}$ . Najjednostavnija metoda, i ona od koje se započinje predikcija vrednosti ETH-a za jedan dan unapred je *naivno* predviđanje.

### Naivno predviđanje

*Naivno* predviđanje na jednodnevnom nivou, je zapravo pretpostavka da se cena deonice narednog dana neće promeniti u odnosu na prethodni. Ma koliko ovo delovalo naivno, za jednodnevno predviđanje nije nimalo besmisleno. Ukoliko se uzme da je vrednost ETH-a  $k$ -tog dana  $v_k$  i vrednost narednog dana  $v_{k+1}$ , može se reći da je njihova relacija opisana izrazom:  $v_{k+1} = v_k + e$ , gde se  $e$  može nazvati promenom vrednosti, koja je kod naivnog predviđanja 0.

Iako ova metoda zadržavanja prethodne vrednosti izgleda možda previše naivno, za jednodnevno predviđanje nije loša, a osim toga ima smisla. Ako se razmatra da promena vrednosti treba da bude različita od 0, treba se zapitati kako je onda treba birati. Najlogičniji je izbor, na osnovu teorije ekonomije i statistike, izabrati  $e$  tako da je  $e \sim \mathcal{N}(0, \sigma^2)$ .

Pošto je  $v_{k+1} = v_k + e$ , onda će:  $v_{k+1} \sim \mathcal{N}(v_k, \sigma^2)$ . Iz toga se vidi da je  $v_{k+1}$  u potpunosti određeno normalnom raspodelom centriranom u  $v_k$  sa varijansom  $\sigma^2$ . Vrednost narednog dana ne zavisi ni od jedne prošle vrednosti te vremenske serije. Zbog toga **Gauss-ov Random Walk** [11] formira **Markovljev lanac**. Pošto je pomeraj nasumičan, najbolje što se može uraditi je naći očekivanu vrednost. Očekivana vrednost Gausove normalne raspodele centrirane oko  $v_k$  je upravo ta vrednost, a to je baš ono što je i navedeno na početku. Ovo govori da, ako vrednost ETH-a prati Random Walk, najbolja predikcija koja se može napraviti za tekući dan je trenutna vrednost ETH-a.

## Moving Average tehnike

Za razliku od naivne predikcije sledeće dve tehnike predstavljaju računske metode, a ne tehnike predikcije. One se kasnije koriste kao bitna osnova za *Simple Exponential Smoothing (SES)*, koja je zaista tehnika predviđanja, iako suštinske razlike u odnosu na drugu tehniku od naredne dve nema. Ove dve tehnike se razlikuju u načinu na koji se računa, tj. ponderiše niz pomoću kojeg se računa srednja vrednost *MA* predikcije i prema tome se dele na dve podvrste. Osnovnu nazivamo *Simple Moving Average (SMA)* i *Exponentially-Weighted Moving Average (EWMA)*.

## Simple Moving Average (SMA)

Najjednostavnija vrsta MA tehnike je SMA. U ovom slučaju ponder ne postoji, tj. ponder je niz jedinica iste dužine kao i dužina prozora koji se koristi. U opštem slučaju, ako je dužina prozora  $n$ , onda je:

$$v_k = \frac{\sum_{i=k-n}^k v_i}{n}.$$

Ukoliko postoji niz:

2	0	8	3	6	1	1	6	5	5
---	---	---	---	---	---	---	---	---	---

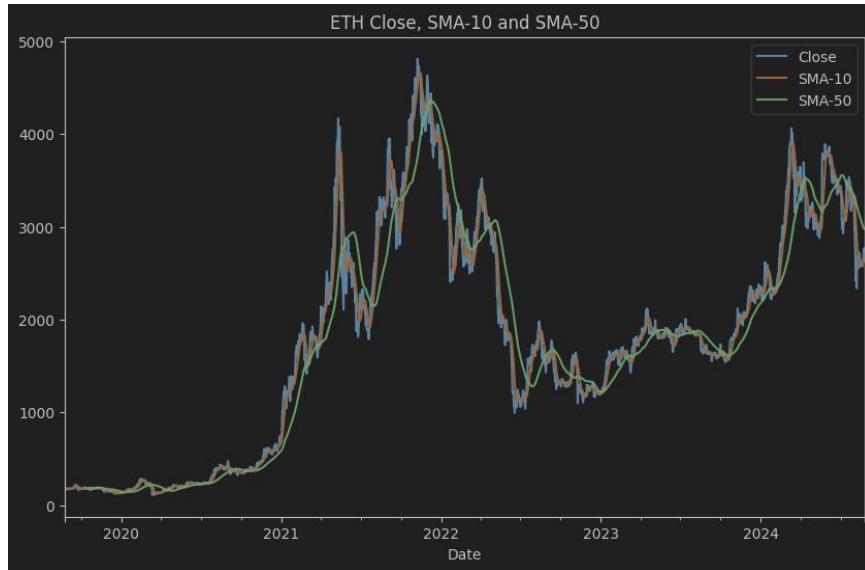
Tabela 1: Arbitražni niz

Ako je dužina prozora 3, usrednjena vrednost za dan  $k$  bi po dатој формулі била рачуната као  $v_k = \frac{v_{k+1} + v_{k-1} + v_{k-2}}{3}$ . Прве две вредности не могу бити израчунате, јер су на индексима мањим од дужине прозора, па им се додељују NaN вредности. SMA низ, који се налази испод оригиналног низа изгледа овако:

2	0	8	3	6	1	1	6	5	5
NaN	NaN	3.33	3.67	5.67	3.33	2.67	2.67	4	5.33

Tabela 2: Arbitražni niz i SMA низ добијен помоћу њега

Концепт MA-а има више смисла од целокупног proseка, јер догађаји од пре 6 месеци нису битни за цене након 6 месеци, већ су много већи свежије вредности. Дужином прозора одређује се број свежих вредности које се жеље узети у обзир.

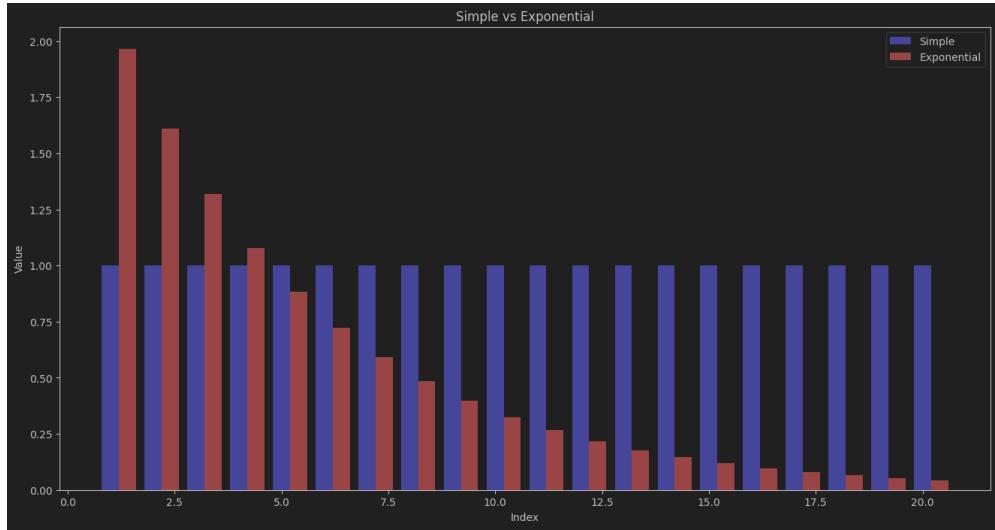


Grafik 1: Vremenska serija ETH-a, SMA-10 i SMA-50 применете над њом

На графику се види да са повећањем ширине прозора прелази постапају глаткији и више касне у односу на реалне вредности.

## Exponentially–Weighted Moving Average (EWMA)

Ovo je druga vrsta MA tehnike. Drugi naziv koje se koristi je Exponential Smoothing, a može se interpretirati kao vrsta low-pass filtra.



Grafik 2:SMA vs EWMA

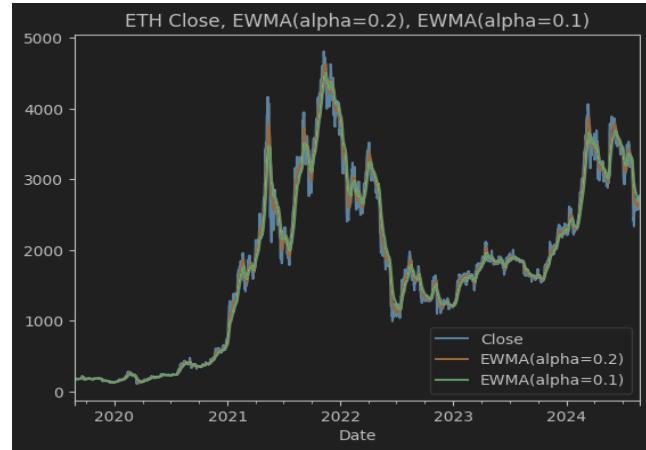
Ovde se može videti razlika u nizu pondera kojima se množi niz. Kod SMA je to niz jedinica širine prozora, a kod EWMA je to niz koji eksponencijalno opada od 2 pa sve do 0. To omogućava da se značaj daje svežijim vrednostima u odnosu na one koje su manje sveže.

EWMA se može predstaviti kao low-pass filter preko formule:

$$\hat{v}_k = \alpha v_k + (1 - \alpha)\hat{v}_{k-1}, \text{ gde } \alpha \in [0, 1], \text{ a} \\ \text{najčešće } \alpha \in [0.1, 0.2].$$

Što je  $\alpha$  bliže 1, to će nova srednja vrednost više težiti najsvežijoj vrednosti, a što je  $\alpha$  bliže 0 to će se više težiti trenutnoj srednjoj vrednosti.

Na slici desno, se može primetiti da EWMA sa  $\alpha = 0.2$  više kasni za vremenskom serijom ETH-a, dok je za  $\alpha = 0.1$  to kašnjenje manje.



Grafik 3:Vremenska serija ETH-a, EWMA( $\alpha=0.1$ ) i EWMA( $\alpha=0.2$ )

## Simple Exponential Smoothing (SES)

SES je zapravo koncipiran kao EWMA, ali tako da daje predikciju. Matematički je skoro identičan, dok sa aspekta izlaza koji se dobija nije. Ovde se umesto:  $\hat{v}_k = \alpha v_k + (1 - \alpha)\hat{v}_{k-1}$  nalazi:  $\hat{v}_{k+1|k} = \alpha v_k + (1 - \alpha)\hat{v}_{k|k-1}$ . Uveden je operator  $|$  pa se leva strana jednačine tumači kao: eksponencijalno glatko predviđanje u trenutku  $k + 1$  ako je poznata vrednosti u trenutku  $k$ .

Može se videti da je optimalno  $\alpha = 0.9386$ , što znači da SES model zadržava vrednost od prošlog dana- $v_k$  skoro 94%, dok  $\hat{v}_{k|k-1}$  utiče na predikciju sa samo 6%, pa se ovo može nazvati devedesetčetvoroprocentna naivna predikcija. Ovo je još jedna potvrda da naivna predikcija na jednodnevnom nivou i te kako ima smisla.

```
{'smoothing_level': 0.9386755399024236,
'smoothing_trend': nan,
'smoothing_seasonal': nan,
'damping_trend': nan,
'initial_level': 186.842392,
'initial_trend': nan,
'initial_seasons': array([], dtype=float64),
'use_boxcox': False,
'lambda': None,
'remove_bias': False}
```

Slika 1:Parametri SES modela

## Autoregressive Integrated Moving Average (ARIMA)

Razlika između SES i ARIMA modela je u tome što je prvi namenjen za specifične vrste podataka sa linearnim trendovim i sezonskim ponavljanjem, dok je ARIMA sličniji modelima mašinskog učenja. On prilagođava model u odnosu na podatke, bez obzira na njihovu prirodu. ARIMA model se sastoji iz tri dela: AR (Autoregressive), I(Integrated) i MA(Moving Average).

AR(p)

Autoregresivni modeli su u osnovi modeli linearne regresije gde je ulaz prethodna vrednost vremenske serije. Kod klasične linearne regresije model je dat po ovoj formuli:

$$\hat{y} = mx + b$$

, gde je  $x$  ulaz, a  $m$  i  $b$  se pronalaze minimizacijom greške.

Za razliku od toga,  $AR(p)$  - autoregresivni model reda  $p$ , se može interpretirati kao model linearne regresije višeg reda, i zapisuje se u ovom obliku:

$$\hat{y}_t = b + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p}$$

, gde su  $y_{t-1}, \dots, y_{t-p}$  prethodne vrednosti niza vremenske serije.

Drugi način zapisa ovog modela je:

$$y_t = b + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t$$

, gde je sada na levoj strani znaka jednakosti umesto predviđane vrednosti prava vrednost, a da bi jednakost bila očuvana sa desne strane se dodaje šum:  $\varepsilon_t \sim N(0, \sigma^2)$ .

Takođe ukoliko se primeni operacija očekivane vrednosti na obe strane jednakosti  $\hat{y}_t = y_t \pm \varepsilon_t$  ( $\pm$  jer predznak nije bitan), dobija se:  $\hat{y}_t = E(y_t)$ , što se može interpretirati kao: predikcija  $y_t$  će biti očekivana vrednosti  $y_t$ , sa pretpostavkom da je očekivana vrednost šuma 0.

MA(q)

Za početak treba navesti da  $MA(q)$  ne spada u istu kategoriju kao i SMA ili EWMA. Oblik MA modela reda  $q$  je takođe linearna funkcija, ali ovog puta linearna funkcija prošlih grešaka, a ne vrednosti vremenske serije. Zapis je sličan kao kod autoregresivnog modela:

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Ako se i ovde primeni operacija očekivane vrednosti na obe strane jednakosti dobija se:

$$E(y_t) = E(c + \varepsilon_1 y_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t) = c$$

Ako se greške tretiraju kao:  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ , samim tim:  $E(\varepsilon_i) = 0$  i  $E(y_t) = c$ . Zbog toga se može smatrati bias  $c$  srednjom vrednošću, a greške fluktacijama zbog kojih će  $y_t$  biti veće ili manje od  $c$ .

ARMA(p,q)

$ARMA(p,q)$  predstavlja samo kombinaciju, tj. zbir AR(p) i MA(q) modela, tako da ovaj model treba koristiti ukoliko se smatra da je svaka tačka u vremenskoj seriji u linearnej korelaciji sa pređašnjim tačkama i greškama. ARMA(p,q) model se može zapisati kao:

$$y_t = b + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

Da bi se došlo do ARIMA modela nedostaje još / deo. On se naziva integrirani deo i funkcioniše drugačije u poređenju sa ostalima, pa je zato ostavljen za kraj.

I(d)

Da bi se razumelo kako integrirani deo funkcioniše prvo je potrebno razumeti koncept diferenciranja. Ukoliko postoji vremenska serija  $\{y_t\} = \{y_1, y_2, \dots, y_T\}$ , da bi se dobila nova, diferencirana vremenska serija potrebno je primeniti  $\Delta$  operaciju na celu seriju tako da:  $\Delta y_t = y_t - y_{t-1}$ . Diferenciranje se radi da bi podaci bili što više stacionarni, a diferenciranje je jedna od metoda koje mogu napraviti od nestacionarne vremenske serije stacionarnu.

$I(d)$  proces je proces koji je stacionaran posle diferenciranja  $d$  puta. Onda se može reći da je integriran do reda  $d$ . Može se definisati ARIMA model tek kada je taj uslov ispunjen.

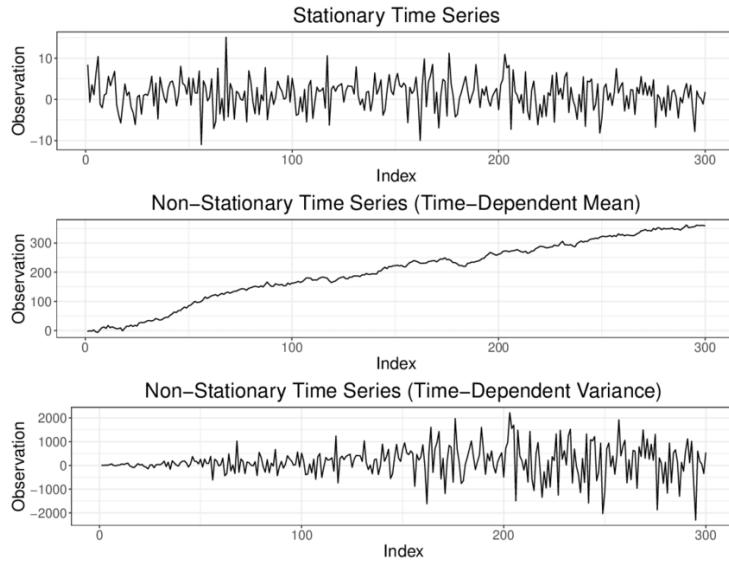
ARIMA(p, d, q)

$ARIMA(p, d, q)$  je ARMA(p, q) model koji radi sa  $d$  puta diferenciranom originalnom vremenskom serijom. U zavisnosti od reda parametra ARIMA modela, može se doći do specijalnih slučajeva ARIMA modela kao npr.  $ARIMA(p, 0, 0)$  koji je AR(p),  $ARIMA(0, 0, q)$  koji je MA(q),  $ARIMA(0, d, 0)$  koji je I(d).

Jedan zanimljiv specijalan slučaj je  $ARIMA(0, 1, 0)$  tj.  $I(1)$ , što je zapravo Random Walk. Može se zapisati kao:  $\Delta y_t = \varepsilon_t$ . Ovde ne postoji, ni AR, ni MA deo, već samo ostaje šum. Drugačije se može zapisati u osnovnoj formi Random Walka:  $y_t = y_{t-1} + \varepsilon_t$ . Ukoliko bi se desilo da povratna vrednost modela posle treniranja bude takva, da model koji najbolje odgovara vremenskoj seriji bude ARIMA(0, 1, 0), moglo bi se zaključiti da vremenska serija ETH-a prati Random Walk i da nije moguće predviđanje korišćenjem prethodnih vrednosti vremenske serije ARIMA modelom.

Stacionarnost i selekcija hiperparametara

Stacionarnost, u širem smislu, znači da se distribucija podataka ne menja tokom vremena, odnosno da su srednja vrednost i varijansa konstantne tokom vremena. To znači da, ukoliko postoji bilo kakav trend ili varijansa zavisna od vremena, vremenska serija nije stacionarna.

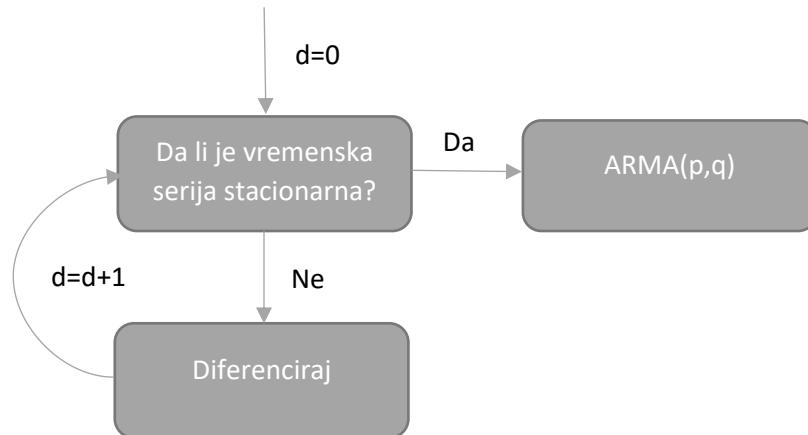


Slika 2: Stacionarna, nestacionarna (sa vremenski zavisnom srednjom vrednošću) i nestacionarna (sa vremenski zavisnom varijansom) serija [12]

Srećom da bi se izbeglo intuitivno ili grafičko određivanje stacionarnosti vremenske serije, postoji test napravljen baš za to. Zove se *ADF Test (Augmented Dickey-Fuller Test)* [13] i radi tako što se kao ulaz prosledi vremenska serija, a kao izlaz dobija *p-vrednost* u intervalu od 0 do 1. U slučaju 0, potpuno je stacionarna, a u slučaju 1 u potpunosti nije stacionarna. Na osnovu postavljene granice, 0,05 najčešće, određuje se da li je vremenska serija stacionarna ili ne. Upravo je to ključni korak u proceni koji hiperparametar  $d$  treba izabrati.

#### Hiperparametar $d$

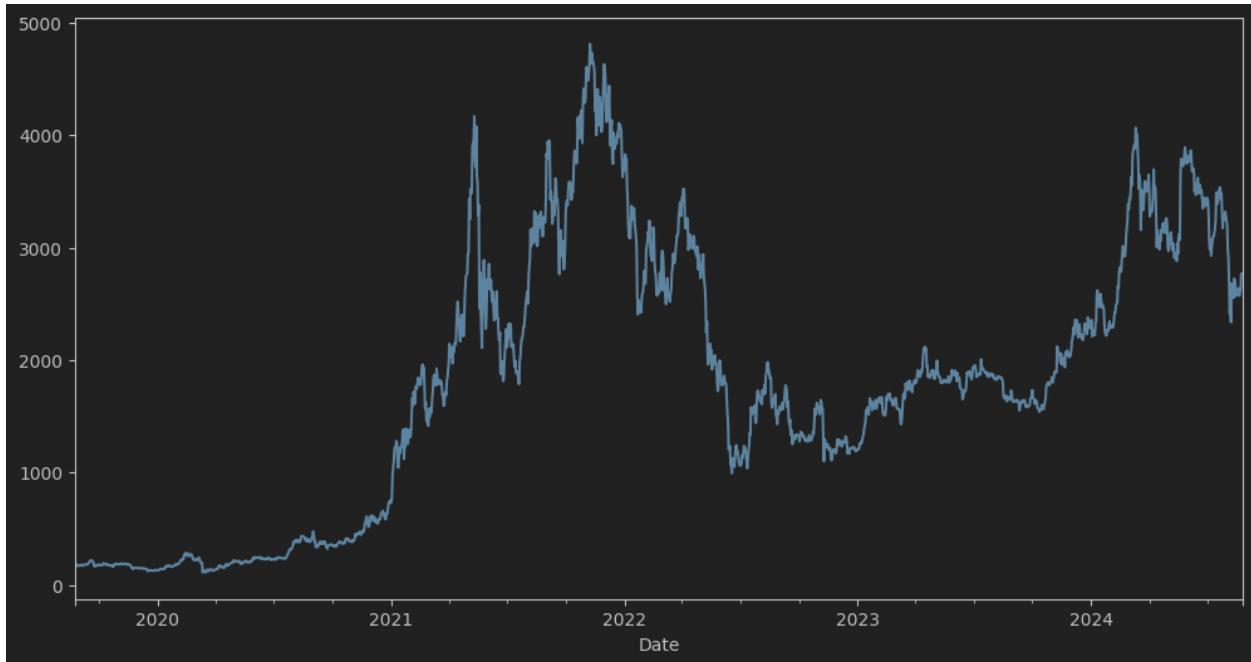
Algoritam rada biranja hiperparamtera  $d$  izgleda ovako:



Ilustracija 1: Algoritam rada biranja parametra  $d$

Stacionarnost vremenske serije je izrazito bitna, jer bi u suprotnom bili potrebni različiti modeli za različite momente u vremenu, zato se može koristiti jedan model ako je vremenska serija stacionarna.

Na slikama ispod se može videti kako izgleda vremenska serija vrednosti ETH-a i kakva je njena stacionarnost.



Grafik 4: Vremenska serija ETH-a

```
adf(eth_close['Close'])
Executed at 2024.09.02 15:02:45 in 160ms

Test-Statistic: -1.714020906597467
P-Value: 0.4239033469394924
Non-Stationary
```

Slika 3: Rezultati ADF testa primjenjenog nad vremenskom serijom ETH-a

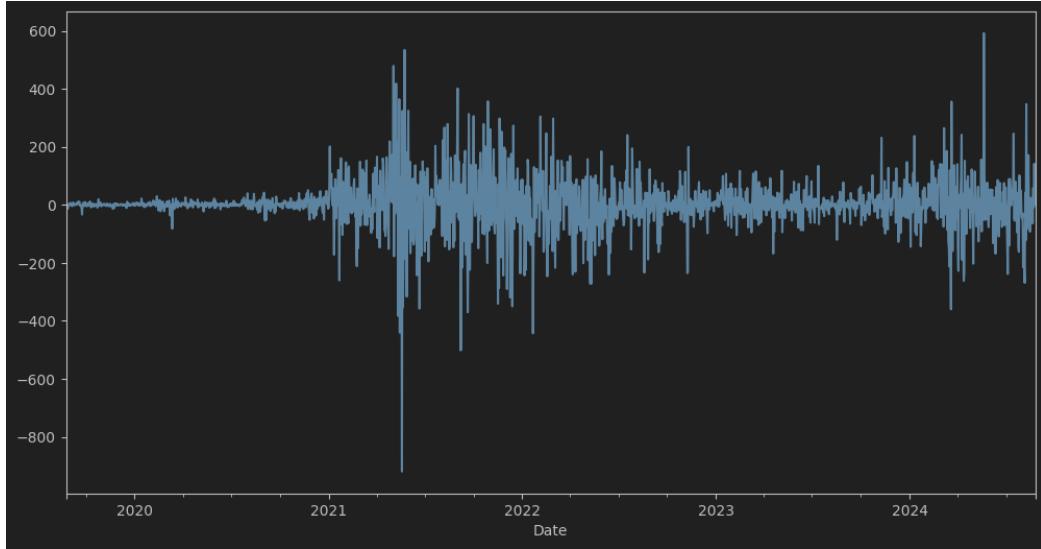
Kao što se može videti iz rezultata testa na slici iznad, vremenska serija u ovom obliku nije stacionarna, a da bi se proverilo kako test radi, pre diferenciranja originalne vremenske serije, kao što nalaže algoritam za pronalaženje  $d$ , test se može isprobati na vremenskoj seriji za koju se pouzdano zna da je u potpunosti stacionarna. Primer takve serije bi bila serija od 1000 nasumičnih brojeva i kada se uradi ADF test očekivano se dobija vrednost 0.

```
adf(np.random.randn(1000))
Executed at 2024.09.02 15:04:42 in 76ms

Test-Statistic: -30.198710860495904
P-Value: 0.0
Stationary
```

Slika 4: ADF test primjenjen nad serijom od 1000 nasumičnih brojeva

Sada, prema algoritmu, treba diferencirati vremensku seriju, povećati vrednost hiperparametra  $d$  za 1 i ponovo proveriti stacionarnost vremenske serije. Grafik diferencirane vremenske serije sada izgleda ovako:



Grafik 5:Diferencirana vremenska serija ETH-a

A rezultat testa stacionarnosti izgleda ovako:

```
adf(eth_close['Ret'].dropna())
Executed at 2024.09.02 15:14:20 in 148ms

Test-Statistic: -10.684450906431993
P-Value: 3.865316356588528e-19
Stationary
```

Slika 5:ADF test primjenjen nad diferenciranom vremenskom serijom ETH-a

Ovo znači da je model tipa  $ARIMA(p, 1, q)$ , tj. da sada treba primeniti  $ARMA(p, q)$  model na diferenciranim podacima vremenske serije.

Hiperparametar  $q$

Sledeći hiperparametar koji treba odrediti je  $q$ . Za određivanje  $d$  se koristi ADF, a za određivanje  $q$  se koristi ACF ili funkcija autokorelacijske. [14] Da bi se razumelo šta je autokorelacija, prvo treba razumeti šta je korelacija, a pre toga šta je kovarijansa. Kovarijansa je očekivana vrednost između dve slučajne promenjive ili matematički:

$$cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\}$$

Autokovarijansa je očekivana vrednost izmedju dve slučajne promenjive, ali iz iste vremenske serije, pa se zbog toga umesto  $cov(X, Y)$  piše  $cov(Y_{t1}, Y_{t2})$ .

Korelacija je skalirana verzija kovarijanse i uvek je između -1 i 1, i samim tim je autokorelacija korelacija između dve slučajne promenjive iz iste vremenske serije. ACF je definisana ako je vremenska serija stacionarna, što znači da autokorelacija zavisi samo od razlike u vremenu ili matematički:

$$\rho(Y(t_1), Y(t_2)) = \rho(t_1, t_2) = \rho(\tau)$$

Tako da je formula za računanje autokorelacije:

$$\hat{\rho}(\tau) = \frac{1}{(T-\tau)\hat{\sigma}^2} \sum_{t=1}^{T-\tau} (y_t - \hat{\mu})(y_{t+\tau} - \hat{\mu})$$

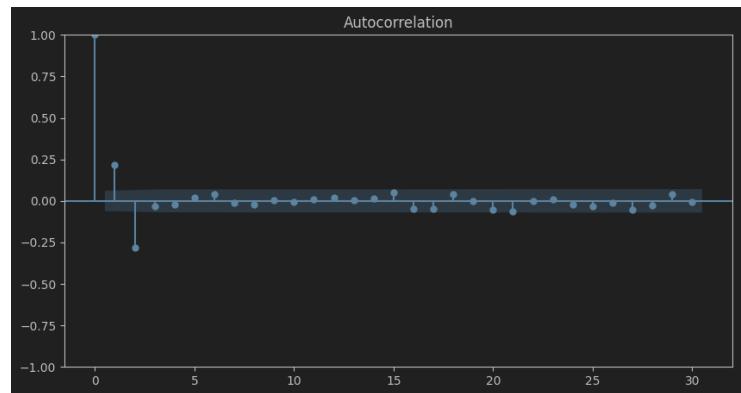
Može se zaključiti da je ACF funkcija koja zavisi samo od kašnjenja  $\tau$ . Ovo znači da će ACF ukazati na pojavu kašnjenja nekog dela vremenske serije u odnosu na drugi deo iste te vremenske serije za  $\tau$ , tako što će vrednost ACF biti upravo broj koraka kašnjenja  $\tau$ . Prirodno je očekivati da će se javiti autokorelacija za  $\tau = 0$ , jer je autokorelacija vremenske serije sama sa sobom potpuna. Grafik ACF će takođe prikazati interval poverenja, što znači vrednosti ACF-a bi trebalo da iskoče iz njega, da bi se smatrале bitnim, dok se one koje su u njemu ili u neposrednoj spoljašnosti zanemaruju.

Kako radi ACF grafik se može proveriti sa diferencnim jednačinama, jer se za njih uvek može prepostaviti korelacija. To je tako, jer one u svojoj formi praktično daju izraz koliko  $y_{k-i}$  utiče na  $y_k$ . Ako se ispita vremenska serija zadana kodom ispod, pod nazivom *ma*:

```
errors = 0.1 * np.random.randn(1000)
ma = []
for i in range(1000):
    x = errors[i] + 0.5 * errors[i-1] - 0.3 * errors[i-2]
    ma.append(x)
ma = np.array(ma)
```

Slika 6: Kod vremenske serije *ma* zadate diferencnom jednačinom

Dobija se ACF na slici desno. Ovo jeste očekivano s obzirom da se vremenska serija formira pomoću pomoćnog niza, uzimajući njegove vrednosti na indeksima  $i, i-1, i-2$  da bi se dobio novi element niza. Zbog toga imamo vrednosti ACF-a koje se značajno razlikuju od nule na kašnjenju 1 i 2.



Grafik 6: Autokorelacija za vremensku seriju zadatu kodom na slici iznad

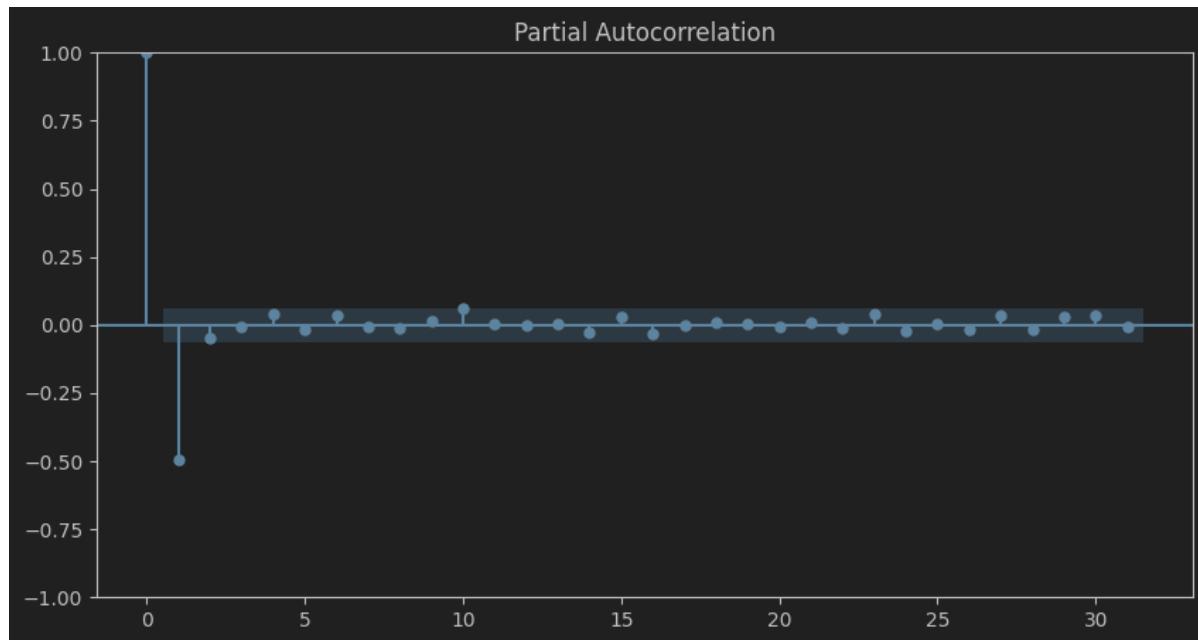
### Hiperparametar $p$

Za određivanje poslednjeg hiperparametra  $p$  koristi se treća funkcija: *PACF* ili funkcija parcijalne autokorelacije. [15] Formula je nešto složenija, ali princip za nalaženje  $p$  je identičan principu koji se koristi za  $q$ , samo što se umesto ACF sada koristi PACF. Za funkciju *x1* zadatu kodom:

```
x1 = [0]
for i in range(1000):
    x = -0.5 * x1[-1] + 0.1 * np.random.randn()
    x1.append(x)
x1 = np.array(x1)
```

Slika 7: Kod nove vremenske serije *x1* zadate diferencnom jednačinom

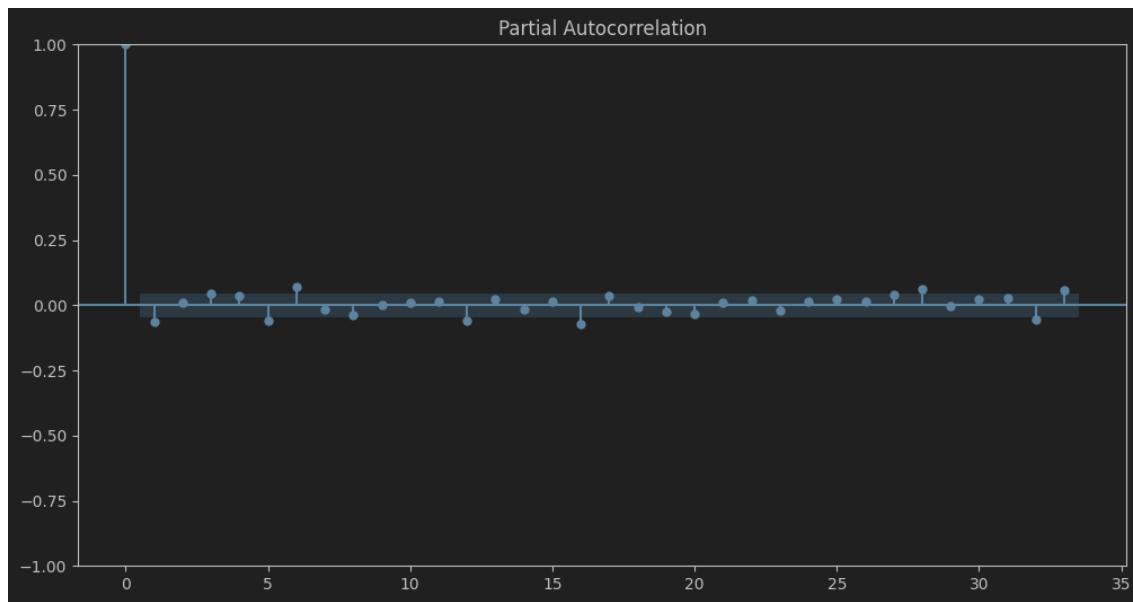
Dobija se grafik PACF na slici ispod.



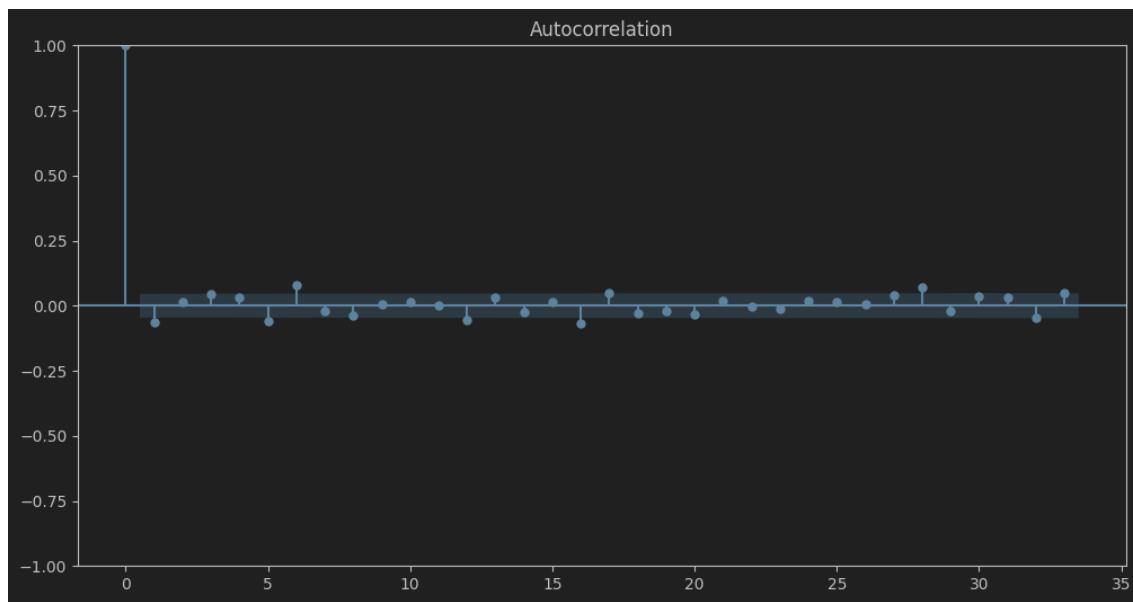
Grafik 8: Parcijalna autokorelacija za vremensku seriju *x1* zadatu kodom na slici iznad

Zbog ovakvog izgleda grafika vrednost  $p$  se bira kao 1.

Sada ostaje da se vidi kakve su ACF i PACF za diferenciranu vremensku seriju vrednosti ETH-a. Grafici izgledaju ovako:



Grafik 9:Autokorelacija za diferenciranu vremesku seriju ETH-a



Grafik 10:Parcijalna autokorelacija za diferenciranu seriju ETH-a

Iz ova dva grafika se može zaključiti da su  $p=q=0$ , što znači da je model  $ARIMA(0, 1, 0)$ . On je već spominjan kao specijalan slučaj  $ARIMA$  modela. Iz ovoga se može zaključiti da diferencirana vremenska serija prati Random Walk. Pošto ovaj zaključak ne zadovoljava cilj ovog rada, da se pobedi naivna predikcija i pod pretpostavkom da ACF i PACF neće uvek dati idealne rezultate, treba se prepustiti tehnikama koje su više nalik mašinskom učenju, tj. modelu pod nazivom *Auto ARIMA*.

## Auto ARIMA

Biranje hiperparametara pomoću ACF i PACF je direktni pristup, koji nažalost ne ispunjava očekivanja za vremensku seriju ETH-a na željeni način. Umesto toga pomoću *Auto ARIMA* modela se bira najbolji set hiperparametara, koristeći metodu pokušaja i grešaka. Jedan od najčešćih metoda koje koristi tu tehniku je Grid Search, gde se pretražuje svaka opcija hiperparametara, ali složenost Grid Search-a raste eksponencijalno s povećanjem broja hiperparametara. Umesto toga će se koristiti Stepwise Search smišljen od strane Hyndmana i Khandakara 2008. godine.[16] Kada Auto ARIMA odradi Stepwise Search dobija se model čiji parametri izgledaju ovako:

Dep. Variable:	y	No. Observations:	1827			
Model:	SARIMAX(2, 1, 2)	Log Likelihood	-10781.878			
Date:	Mon, 02 Sep 2024	AIC	21573.757			
Time:	20:38:55	BIC	21601.306			
Sample:	08-25-2019 - 08-24-2024	HQIC	21583.919			
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	-0.2922	0.113	-2.591	0.010	-0.513	-0.071
ar.L2	-0.6259	0.082	-7.649	0.000	-0.786	-0.466
ma.L1	0.2072	0.111	1.860	0.063	-0.011	0.425
ma.L2	0.6470	0.079	8.172	0.000	0.492	0.802
sigma2	7934.7378	111.458	71.190	0.000	7716.284	8153.191
Ljung-Box (L1) (Q):	0.84	Jarque-Bera (JB):	9854.69			
Prob(Q):	0.36	Prob(JB):	0.00			
Heteroskedasticity (H):	3.48	Skew:	-0.49			
Prob(H) (two-sided):	0.00	Kurtosis:	14.34			

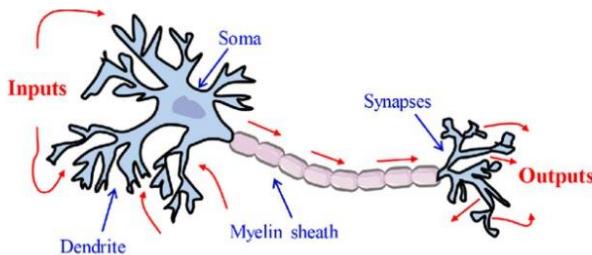
```
{'maxiter': 10,
'method': 'lbfgs',
'order': (2, 1, 2),
'out_of_sample_size': 0,
'scoring': 'mse',
'scoring_args': {},
'seasonal_order': (0, 0, 0, 0),
'start_params': None,
'suppress_warnings': True,
'trend': None,
'with_intercept': False}
```

Slika 8:Parametri Auto ARIMA modela

Ovde se radi o *SARIMAX* modelu, ali se može videti da je *seasonal\_order*(0, 0, 0, 0) tako da sezonska komponenta nije prisutna. Najbolje prilagođen model, ako se koriste podaci iz prethodnih 5 godina je oblika *ARIMA*(2, 1, 2). Što se tiče jednodnevne predikcije, on je predvideo vrednost deonice od 2773.60\$ dok je stvarna vrednost bila 2771.08\$ za poslednji dan vremenske serije.

Kako se i dalje radi o jednodnevnim predviđanjima nema smisla gledati grafike, već se pomocu Auto ARIMA prilagođava model koristeći proteklih 10, 50 i 100 dana, pa se na osnovu toga beleže rezultati i radi evaluacija pomoću koje će se kasnije proceniti šta je najbolje koristiti za predikciju.

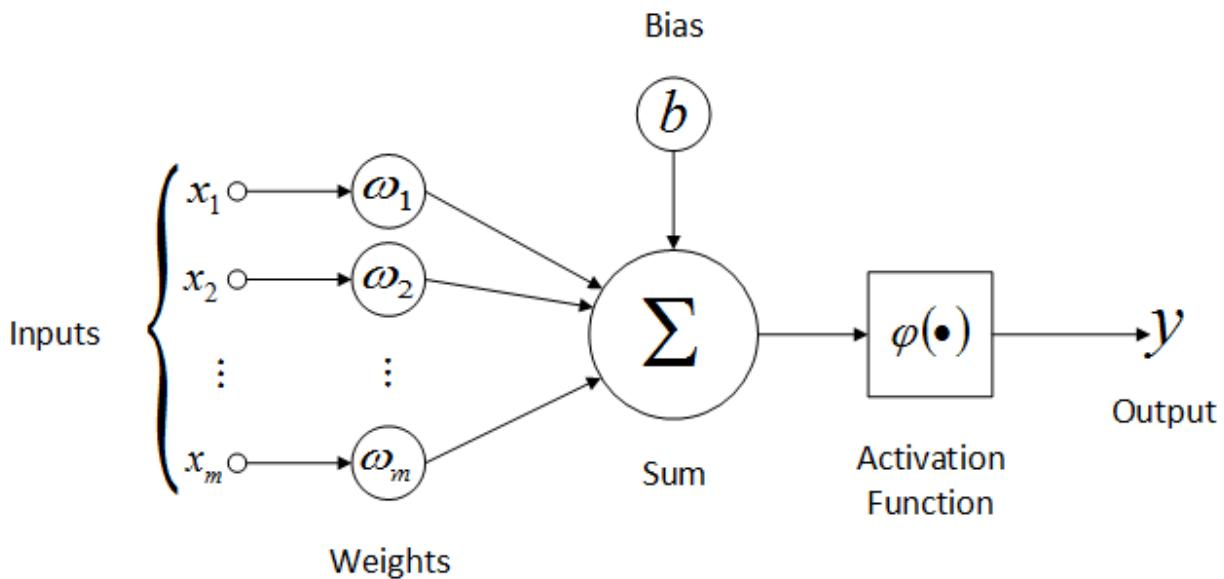
## Duboko učenje



Slika 9: Neuron [17]

Duboko učenje predstavlja granu mašinskog učenja koja se zasniva na modelovanju višeslojnih veštačkih neuronskih mreža. One u svojoj osnovi imaju strukturu biološke neuronske mreže. Ova tehnologija omogućava mnoge stvari, između kojih su obrada i analiza velikih količina podataka, pa je zbog toga postala ključna tehnologija u mnogim oblastima, uključujući predikcije cena deonica, kao i kriptovaluta.

Artificial Neural Network (ANN)



Slika 10: Grafička reprezentacija veštačkog neurona [18]

Osnova svake neuronske mreže je veštački neuron, čiji je matematički model prikazan na slici iznad. Pri modelovanju po uzoru na biološki neuron, moraju se uzeti u obzir tri osnovne komponente koje ga definišu.

Prva komponenta su sinapse, modelovane kao težine (weights), koje povezuju neuronsku mrežu. Sa jedne strane negativne težine odravljaju inhibitorne veze (inhibitory interactions), odnosno one veze koje će manje uticati na akcioni potencijal neurona. Sa druge strane pozitivne vrednosti označavaju ekscitatorne veze (excitatory interactions), one veze koje će više uticati na akcioni potencijal veze.

Sledeća komponenta predstavlja ukupnu aktiviranost neuronske ćelije. Ona se matematički modeluje kao zbir proizvoda svih ulaza i njihovih korespondentnih težina, i biasa koji nema težinu. To se može opisati formulom na sledeći način:

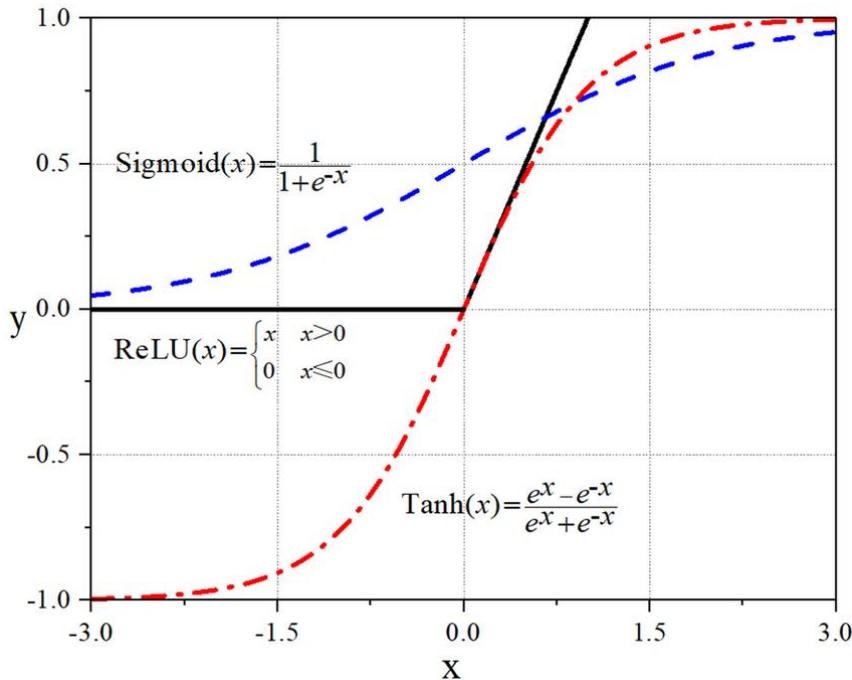
$$v = \sum_{i=0}^n w_i x_i + b$$

Ukupna aktiviranost neuronske ćelije- $v$  predstavlja ulaz u aktivacionu funkciju. Izlaz neurona- $y$  je rezultat primene aktivacione funkcije na vrednost  $v$ :

$$y = \varphi(v)$$

Notacija:  $x_i$  su ulazni signali,  $w_i$  su težine povezane sa svakim ulaznim signalom,  $b$  je bias, tj. konstanta koje se dodaje, omogućavajući modelu da se bolje prilagodi asimetrično raspoređenim podacima i  $\varphi$  je aktivaciona funkcija.

Aktivaciona funkcija deluje kao funkcija sabijanja tako da je izlaz iz neurona obično između fiksiranih vrednosti. Praksa je da  $y \in [0, 1]$  ili  $y \in [-1, 1]$  (slika ispod), jer su performanse ANN najbolje u tom slučaju, ali to nije neophodno sa funkcionisanje neuronskih mreža, već se aktivacione funkcije mogu praviti i na svoju ruku.



Slika 11: Različite aktivacione funkcije neurona [19]

Obuka neuronske mreže najjednostavnije rečeno predstavlja podešavanje njenih težina pomoću optimizatora, tako da mreža što bolje predviđa izlaze za nove ulazne podatke, kojima nije imala priliku da pristupi. Kada se priča o analizi berze, ANN se trenira na velikom skupu podataka vremenske serije, da bi nakon toga, na osnovu prethodno naučnih obrazaca i prilagođenih težina mogla što preciznije predvideti cene deonica u određenoj budućnosti.

Što se tiče predikcije cene ETH-a, algoritam rada programa za svaku vrstu ANN ostaje praktično isti, razlika je samo u arhitekturi modela i skupu podataka za učenje sa nadzorom.



*Ilustracija 2:Algoritam rada koda ANN*

#### Izdvajanje i skaliranje podataka

Za treniranje i predikciju se isključivo koristi „Close“ kolona, tj. „završna cena“ svakog dana. Kada se izdvoji kolona od interesa ona podleže transformacijama kako bi ANN dala što bolje rezultate. Korišćene transformacije su: logaritmovanje, diferenciranje i potom skaliranje StandardScalerom, koji svakom elementu oduzima srednju vrednost vremenske serije i potom svaki element deli sa standardnom devijacijom iste.

#### Kreiranje skupa podataka za učenje sa nadzorom

Ovaj deo predstavlja suštinsku razliku u radu između jednodnevne i višednevne predikcije, kao i između klasične i rekurzivne ANN. Pošto je cilj, u slučaju jednodnevног predviđanja, obučiti ANN da na osnovu prethodnih  $T$  dana predvidi cenu ETH-a  $T+1$ -og dana, ANN mora biti obezbeđena odgovarajuće forma podataka.

Kod se nalazi na slici desno.

U  $X$  su smešteni svakih  $T$  uzastopnih dana od početka do kraja vremenske serije za koju se radi predikcija, dok su u  $Y$  smešteni svi korespondentni  $T+1$  dani. Na kraju će se dobiti da je dimenzija  $X$ : (dužina vremenske serije,  $T$ ), a dimenzija  $Y$  (dužina vremenske serije). Nakon toga se dele podaci na one za trening i test, pa se kreće u izradu modela.

```

T = 10
X = []
Y = []
for t in range(len(series) - T):
    x = series[t:t + T]
    X.append(x)
    y = series[t + T]
    Y.append(y)
X = np.array(X).reshape(-1, T)
Y = np.array(Y)

Xtrain, Ytrain = X[:-Ntest], Y[:-Ntest]
Xtest, Ytest = X[-Ntest:], Y[-Ntest:]
  
```

*Slika 12:Kod za kreiranje skupa podataka za učenje sa nadzorom za jednodnevnu predikciju*

## Kreiranje ANN modela

Arhitektura ANN modela je data na slici ispod.

```
# Basic ANN
i = Input(shape=(T,))
x = Dense(64, activation='relu')(i)
x = Dropout(0.2)(x)
x = Dense(64, activation='relu', kernel_regularizer=l2(0.01))(x)
x = Dense(1)(x)
model = Model(inputs=i, outputs=x)
Executed at 2024.09.04 21:02:49 in 105ms

model.summary()
Executed at 2024.09.04 21:30:24 in 163ms

Model: "functional_31"



| Layer (type)                | Output Shape | Param # |
|-----------------------------|--------------|---------|
| input_layer_15 (InputLayer) | (None, 100)  | 0       |
| dense_35 (Dense)            | (None, 64)   | 6,464   |
| dropout_5 (Dropout)         | (None, 64)   | 0       |
| dense_36 (Dense)            | (None, 64)   | 4,160   |
| dense_37 (Dense)            | (None, 1)    | 65      |


```

Slika 13:ANN model

Ovakav dizajn predstavlja balans između kompleksnosti i robusnosti protiv preprilagođavanje modela. Na početku se nalazi ulazni sloj koji je neophodan za definisanje oblika podataka koji ulaze u model, gde je  $T$  broj dana koji model koristi da bi napravio predikciju za idući dan. Posle njega se nalazi prvi skriveni sloj koji je tipa *Dense (Fully Connected)*. On se koristi za modelovanje složenih obrazaca u podacima. Nakon njega nastavlja se *Dropout* sloj koji nasumično isključuje procenat neurona (na slici iznad 0.2 odnosno 20%) tokom svake iteracije treniranja kako bi model bio robusniji tj. otporniji na preprilagođavanje. Zatim ponovo *Dense* sloj, radi povećanja kapaciteta modela, kod koga se u odnosu na prvi *Dense* sloj uvodi *L2* regularizacija. Ona kažnjava velike težine kao još jedna mera zaštite od preprilagođavanja. Na samom kraju se nalazi izlazni sloj koji ima samo jedan neuron, jer izlaz ANN treba da bude samo jedna vrednost, odnosno cena deonice u narednom vremenskom koraku. Što se tiče hiperparametara (na slici iznad), oni su optimalni za  $T = 100$ , ali naravno u zavisnosti od  $T$  oni se razlikuju, a slika sa konkretnim hiperparametrima je data kao primer.

## Podešavanje hiperparametara i predikcija

Pomoću Grid Search-a se traži optimalni skup hiperparametara od onih koji se nalaze u rečniku na slici ispod.

Količina hiperparametara, kao i njihove opcije moglo bi biti znatno veće, ali treba uzeti u obzir da je preko Grid Search-a broj pokrenutih ANN jednak broju svih mogućih opcija. U ovoj „kraćoj“ verziji je to:  $2^6 = 64$ , plus cv parametar koji iznosi 2, pa se dolazi do broja od 128 prilagođavanja ANN. Samim tim vreme potrebno za treniranje svih tih ANN proporcionalno raste. Zbog želje da se kod izvrši u relativno normlanom vremenskom roku rečnik *param\_grid* izgleda ovako. Nakon izbora najboljih hiperparametara se vrši predikcija i dobijaju se rezultati.

```
param_grid = {
    'model__units': [32, 64],
    'model__activation': ['relu', 'tanh'],
    'model__optimizer': ['adam', 'rmsprop'],
    'model__dropout_rate': [0.1, 0.2],
    'model__regularization_rate': [0.01, 0.001],
    'epochs': [50, 100],
}
```

Slika 14: Skup hiperparametara za ANN

## Sređivanje podataka i evaluacija modela

Da bi bili postignuti što bolji rezultati izlaz iz ANN su podaci koji su prethodno bili logaritmovani, diferencirani i skalirani. Sada da bi se dobili konkretni rezultati moraju se odraditi inverzne operacije u odnosu na sve prvobitno primenjene transformacije. Tek kada su podaci sređeni može se preći na evaluaciju modela. Za evaluaciju se koriste metrike poput: koeficijenta determinacije ( $R^2$  score), srednje apsolutne procentualne greške (MAPE), srednje apsolutne greške (MAE), srednje kvadratne greške (MSE) i korena srednje kvadratne greške (RMSE) pomoću kojih se donosi sud koji model je najbolji.

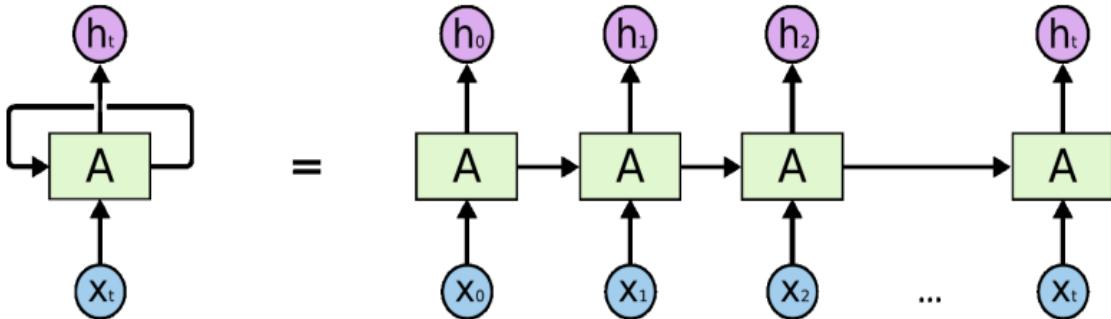
Da bi se očuvala konzistentnost i da bi estimatori bili uporedivi za evaluaciju svih modela jednodnevne predikcije koristilo se jednodnevno predviđanje od 318. do 328., od 818. do 828., od 1318. do 1328. i od 1818. do 1828. dana vremenske serije uzimajući u obzir da je vremenska serije ima 1828 dana.

Pošto neće svi rezultati biti najbolji za jedan model, najlogičnije je dati veću težinu MAPE estimatoru nego ostalim iz dva razloga. Prvi je to što je korišćen kao funkcija gubitka prilikom kompajliranja modela, a drugi je jer on jedini daje uvid u procentualnu grešku, a to je ono što može da objedini evaluaciju podataka iz 4 dela vremenske serije.

## Recurrent Neural Network (RNN)

Kod klasičnih ANN se ne dešava da izlazi iz neurona, postaju ulazi u neurone iz prethodnih slojeva. Kada se malo razmisli o dešavanjima u realnom svetu, može se primetiti da finalna odluka odnosno izlaz ne zavisi samo od spoljasnjih ulaza već i od prethodnih izlaza. Na primer, prilikom čitanja knjige, razumevanje rečenice ne zavisi samo od kombinacije reči pročitanih u tom trenutku, kao i nekoliko trenutaka pre, već i od konteksta dobijenog iz određenog broja prethodnih rečenica. Ljudi ne počinju da razmišljaju od početka svake sekunde, već njihovo razmišljanje uveliko zavisi od prethodnih razmišljanja koji su dovele upravo do tog trenutka. Koncept „konteksta“ nije dostupan klasičnoj ANN i to postaje njihovo najveće ograničenje za odrađene primene. RNN su napravljene da bi se preskočila ova prepreka. Zbog povratne sprege informacija, RNN uspevaju da se izbore sa ovom preprekom.

### Otpakovana RNN i Nestajući(Eksplodirajući) gradijent



Slika 15:Otpakovana RNN [20]

Ako se posmatra kauzalni sistem u nekom trenutku  $t$  za neki ulaz  $X_t$  RNN kao izlaz daje  $h_t$ . U sledećoj iteraciji,  $t + 1$  RNN će imati dva ulaza:  $X_{t+1}$  i  $h_t$  i daće izlaz  $h_{t+1}$ . Ovaj krug informacija dozvoljava informacijama da se šire od jednog neurona do drugog.

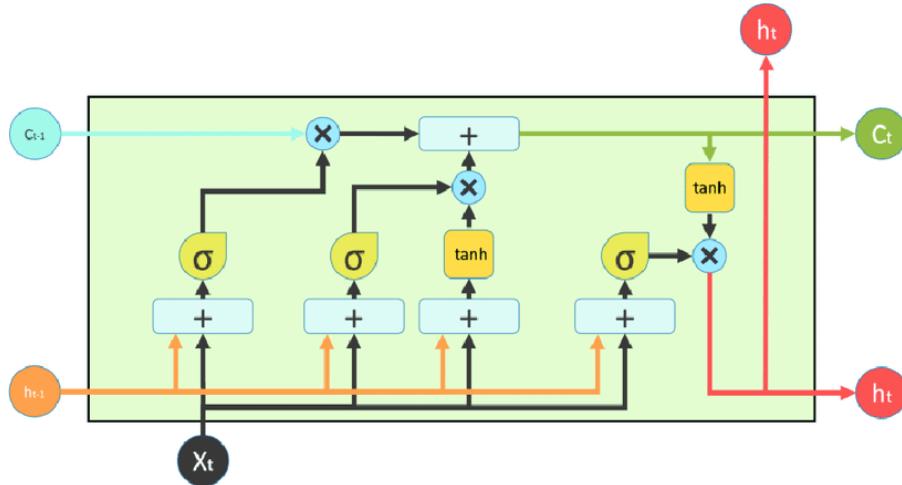
RNN naravno ima svoje limite. Kada je kontekst iz bliske prošlosti dobićemo dobar izlaz, međutim ako je kontekst iz daleke prošlosti RNN daje loše rezultate. To je razlog zašto RNN verovatno neće raditi na za probleme gde je potrebno pamćenje iz duge prošlosti. Ovaj problem se takođe naziva nestajući (eksplodirajući) gradijent. Zbog ovog problema RNN se u svojoj osnovnoj formi retko koriste.

Neka se uzme hipotetički slučaj sa težinom  $w_a$  gde je  $|w_a| < 1$  i brojem dana  $T$  koji se koriste za obučavanje RNN za predikciju narednog dana, a u kudu je korišćeno  $T \in \{10, 50, 100\}$ . To znači da će težina, zbog svojstava klasične RNN biti u krajnjem trenutku pomnožena  $T$  puta sa samom sobom, tj.  $w_{a+T} = w_a^T$ . Zbog toga što je  $|w_a| < 1$  će  $w_{a+T} \sim 0$ . U slučaju eksplodirajućeg gradijenta se dešava isti mehanizam, samo što je  $|w_a| > 1$ , pa  $w_{a+T} \sim \infty$ .

Gradijent se nalazi u nazivu zbog algoritama poput algoritma najbržeg pada (Gradient Descent) i ostalih njegovih ekspanzija poput adam-a i rmsprop-a pomoću kojih se minimizuje funkcija gubitka (loss function). Svaki korak u algoritmu je proporcionalan gradijentu te težine, što znači da će se u prvom slučaju, kada nestane, zaglaviti u tački iz koje neće moći da se pomeri, a u drugom slučaju, kada eksplodira, će najverovatnije prekočiti minimum i verovatno dobiti loše rezultate. Zbog korišćenja veće količine dana vremenske serije cena ETH-a za obuku modela, se klasična RNN zaobilazi i prelazi na LSTM RNN.

## Long Short-Term Memory Neural Network (LSTM)

Ključna razlika u odnosu na prošle ANN je u samom izgledu neurona. LSTM neuron izgleda ovako:

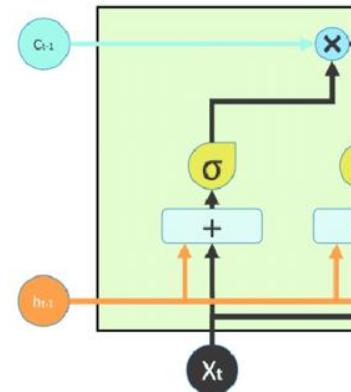


Slika 16:LSTM neuron [21]

Linija koja povezuje  $c_{t-1}$  i  $c_t$  prolazi celom dužinom neurona i naziva se **stanje čelije** i predstavlja **dugotrajnu memoriju** neurona. Iako se može menjati operatorima sabiranja i množenja, ona nema, niti težinu, niti bias koji je direktno mogu menjati. Upravo odsustvo tih parametara omogućava dugotrajanu memoriji da teče kroz seriju neurona, bez brige o nestajućem (ili eksplodirajućem) gradijentu.

Narandžasta linija poseduje težine, iako nisu prikazane na slici iznad. Ona se naziva **skriveno stanje čelije** i predstavlja **kratkotrajanu memoriju**.

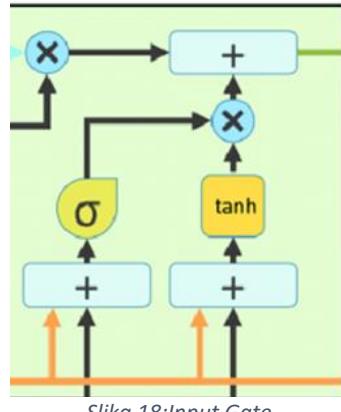
Može se primetiti da je ulaz u prvu aktivacionu funkciju, gledajući sa leva na desno, suma ulaza i kratkotrajne memorije, koje su pomnožene odgovarajućim težinama (što se ne vidi na slici iznad). Izlaz iz aktivacione funkcije *sigmoide* je između 0 i 1, a množi se sa dugotrajnom memorijom. To znači da izlaz iz aktivacione funkcije predstavlja procenat dugotrajne memorije koji se zadržava. Baš iz tog razloga ovaj deo LSTM neurona se naziva **Forget Gate**.



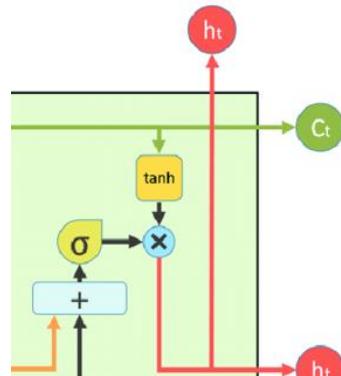
Slika 17:Forget Gate

Desna aktivaciona funkcija *tanh*, kombinuje ulaz i kratkotrajnu memoriju kako bi se dobila potencijalna dugoročna memorija. Leva aktivaciona funkcija *sigmoida* takođe kombinuje ulaz i kratkotrajnu memoriju kako bi se dobio procenat potencijalne dugotrajne memorije koji će ostati zapamćen, pa se baš iz tog razloga *sigmoida* koristi. Onda se one množe i sabiraju sa prethodnom dugoročnom memorijom kako bi se dobila nova dugoročna memorija. Ovaj deo LSTM neurona se naziva **Input Gate**.

Sada kada je formirana nova dugoročna memorija, ostaje poslednji deo LSTM neurona, koji služi za formiranje nove kratkoročne memorije.



Slično kao i kod Input Gate-a, postoji par *tanh* i *sigmoide*, koje služe da odrede potencijalnu kratkoročnu memoriju i procenat kratkoročne memorije koje treba da bude upamćen. Proizvod izlaza te dve aktivacione funkcije formira novu vrednost kratkoročne memorije. Ovo označava kraj LSTM neurona i naziva se **Output Gate**.



Pošto je algoritam koda identičan u odnosu na klasičnu ANN, nema potrebe za ponavljanjem istih stvari već samo za isticanjem onih koje se razlikuju. Jedina stvar koja se suštinski razlikuje je sama arhitektura modela koja se može se videti na slici desno.

```
i = Input(shape=(T,1))
x = LSTM(64, activation='relu', kernel_regularizer=l2(0.001),
return_sequences=True)(i)
x = Dropout(0.1)(x)
x = LSTM(64, activation='relu', kernel_regularizer=l2(0.001), return_sequences=False)(x)
x = Dropout(0.1)(x)
x = Dense(1)(x)
model = Model(inputs=i, outputs=x)
Executed at 2024.09.14 02:04:42 in 48ms

model.summary()
Executed at 2024.09.05 20:43:04 in 58ms

Model: "functional_3"
```

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 100, 1)	0
lstm_2 (LSTM)	(None, 100, 64)	16,896
dropout_2 (Dropout)	(None, 100, 64)	0
lstm_3 (LSTM)	(None, 64)	33,024
dropout_3 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

*Slika 20:LSTM model*

Zaključno sa LSTM ANN, urađene su jednodnevne predikcije za sve vrste modela koje su bile zamišljene. Ostaje samo da se pogledaju rezultati i odredi koji model najbolje može da predvidi šta se dešava sa cenom ETH-a narednog dana.

### Rezultati jednodnevne predikcije

	T=10d	T=50d	T=100d
SES	0.9961, 2.09, 58.2, 8215, 90.64	0.9962, 1.92, 53.71, 7912, 88.95	0.9962, 1.95, 55.3, 7960, 89.22
ARIMA	<b>0.9966</b> , 2.24, 55.01, 7080, 84.14	0.9962, <b>1.83</b> , 55.08, 7849, 88.59	0.9963, 1.97, 55.29, 7813, 88.39
ANN	0.9962, 1.89, 55.02, 7714, 83.43	0.9964, 1.86, <b>53.15</b> , <b>7504</b> , 82.14	0.9961, 1.89, 55.1, 7929, 83.73
LSTM	0.9963, 1.88, 53.95, 7604, 81.97	0.9963, 1.89, 54, 7611, 81.97	0.9963, 1.88, 53.8, 7614, <b>81.83</b>

Tabela 3:Rezultati jednodnevne predikcije

Može se primetiti da najbolji  $R^2$  postiže ARIMA sa  $T = 10$ , najbolju MAPE postiže ARIMA sa  $T = 50$ , najbolju RMSE postiže LSTM za  $T = 100$  i najbolje MAE i MSE postiže ANN za  $T = 50$ . Što se tiče kombinacija svih metrika, jer to je precizniji pokazatelj nego maksimumi pojedinačnih metrika sami za sebe, najbolje se pokazuju ANN i LSTM koji približno daju iste rezultate nezavisno od broja korišćenih prethodnih dana.

Neke velike razlike među modelima se ne vide toliko za jednodnevnu predikciju, ali jako bitna stvar koja se može zaključiti, na osnovu ovih rezultata, je to da Random Walk hipoteza nije tačna. Rezultati naivne predikcije za  $R^2$ , MAPE, MAE, MSE i RMSE su: 0.9962, 1.95, 55.38, 7850 i 88.6 , tako da skoro svaki model za najbolji izbor svog  $T$  daje bolje rezultate, što je i bio cilj da se ispita.

Treba napomenuti da je ovo daleko od najboljeg modela ANN za predikciju cena deonica ili kriptovaluta. Sa sigurnošću se mogu dobiti bolji rezultati sa nekim još novijim i kompleksnijim ANN, kao i korišćenjem većeg skupa podataka, kao što je broj transakcija koje su se obavile u toku dana, što bi bila veoma zanimljiva stvar za proširenje ovog rada.

### Desetodnevna predikcija

Logika kreiranja modela za desetodnevnu predikciju se suštinski ne razlikuje u odnosu na jednodnevnu predikciju. Što se tiče SES i ARIMA modela promenjen je samo parametar koji govori koliko dana unapred se predviđa, a što se tiče ANN i LSTM ima nekoliko razlika. Ključna stvar koju jedino i vredi napomenuti je u sređivanju podataka koji se prosleđuju ANN. Umesto ulaza koji je dobijala ANN za jednodnevnu predikciju sada ulaz u ANN koja predviđa 10 dana unapred se može videti na slici ispod.

$T_x$  predstavlja broj pređašnjih dana koji se koristi za predikciju, dok  $T_y$  je broj dana koji se predviđaju.  $X$  i  $Y$  predstavljaju pomoćne promenjive, gde se u  $X$  smešta svakih  $T_x$  dana, a u  $Y$  smešta narednih  $T_y$  dana. One se kasnije dele u  $X_{train\_m}$ ,  $Y_{train\_m}$ ,  $X_{test\_m}$  i  $Y_{test\_m}$  kako ANN ne bi mogla da dopre do podataka koje zapravo treba da predvidi i time poništi smisao celog postupka predviđanja.

Jedina razlika osim ove u odnosu na jednodnevno predviđanje je u nešto složenijem sređivanju podataka nakon predikcije, ali ta složenost nije dovoljno značajna da bi zahtevala ozbiljnije razmatranje. Iz tog razloga se prelazi na rezultate i evaluaciju modela.

```
series = temp['ScaledLogReturn'].dropna().to_numpy()

Tx = 10
Ty = Ntest
X = []
Y = []
for t in range(len(series) - Tx - Ty + 1):
    x = series[t:t+Tx]
    X.append(x)
    y = series[t+Tx:t+Tx+Ty]
    Y.append(y)

X = np.array(X).reshape(-1, Tx)
Y = np.array(Y).reshape(-1, Ty)

Xtrain_m, Ytrain_m = X[:-Ntest], Y[:-Ntest]
Xtest_m, Ytest_m = X[-Ntest:], Y[-Ntest:]
```

Slika 21: Kod za kreiranje skupa podataka za učenje sa nadzorom za višednevnu predikciju

## Rezultati desetodnevne predikcije

	T=10d	T=50d	T=100d
SES	0.9773, 6.15, 149.29, 47360, 217.62	0.9824, 5.43, 129.89, 36729, 191.65	0.9903, 3.65, 92.71, 20343, 142.63
ARIMA	0.9913, 4.02, 90.39, 18073, 134.43	0.9949, 3.18, 72.54, 10620, 103.05	0.9949, 3.73, 72.96, 10631, 103.11
ANN	0.9948, 2.2, 57.36, 10204, 82.41	<b>0.9953, 2.19, 55.68, 9074, 79.58</b>	0.995, <b>2.12</b> , 56.5, 9850, 81.37
LSTM	0.9947, 2.24, 57.88, 10374, 83.11	0.9948, 2.21, 57.03, 10088, 81.88	0.9947, 2.22, 57.55, 10260, 82.65

Tabela 4: Rezultati desetodnevne predikcije

Može se primetiti da najbolje  $R^2$ ,  $MAE$ ,  $MSE$  i  $RMSE$  postiže ANN sa  $T = 50$ , najbolju  $MAPE$  postiže ANN sa  $T = 100$ . Što se tiče kombinacija svih metrika, tu ANN definitivno pobeđuje, iako ni LSTM ne zaostaje puno, nezavisno od broja korišćenih prethodnih dana.

U odnosu na jednodnevnu predikciju, može se videti da se rezultati dosta više razlikuju. Kada se to dodatno uporedi sa naivnom predikcijom čiji su rezultati za  $R^2$ ,  $MAPE$ ,  $MAE$ ,  $MSE$  i  $RMSE$ : 0.9829, 5.59, 135.45, 35621 i 188.74 može se videti da je svaki model značajno nadmašuje, a ako bi se trebalo odlučiti za jedan model desetodnevne predikcije to bi bila klasična ANN koja koristi 50 prethodnih dana.

## Tridesetodnevna predikcija

U odnosu na desetodnevnu predikciju nema nikakve razlike u kodu i logici, izuzev menjanja broja 10, sa brojem 30 na određenim mestima. Zbog toga se može odmah preći na rezultate.

## Rezultati tridesetodnevne predikcije

	T=10d	T=50d	T=100d
SES	0.7915, 16.41, 440.81, 487878, 698.48	0.9018, 13.09, 332.52, 229912, 479.49	0.9459, 7.37, 212.72, 126588, 335.79
ARIMA	0.9048, 14.68, 239.12, 181717, 396.32	0.9528, 9.37, 303.86, 288019, 446.96	0.9387, 9.23, 192.69, 107938, 306.58
ANN	0.9824, 5.51, 134.28, 42031, 180.9	0.9823, 5.19, 133.87, 42132, 181.01	0.9823, 5.51, 134.56, 42049, 181
LSTM	0.9823, 5.13, 133.77, 42110, 181	<b>0.9824, 5.12, 133.45,</b> <b>41980, 180.64</b>	0.9821, 5.17, 134.35, 42435, 181.81

Tabela 5: Rezultati tridesetodnevne predikcije

U slučaju tridesetodnevne predikcije LSTM sa  $T = 50$ , pobeduje u svim metrikama, iako klasičan ANN ne zaostaje mnogo. Ako se uzmu u obzir rezultati naivne predikcije za  $R^2$ , MAPE, MAE, MSE i RMSE: 0.91, 12.98, 315.8, 187895 i 433.47, za tridesetodnevnu predikciju se najbolje može videti postepeni prelaz od najjednostavnijeg modela do LSTM po pitanju sve boljih rezultata.

## Zajednički rezultati

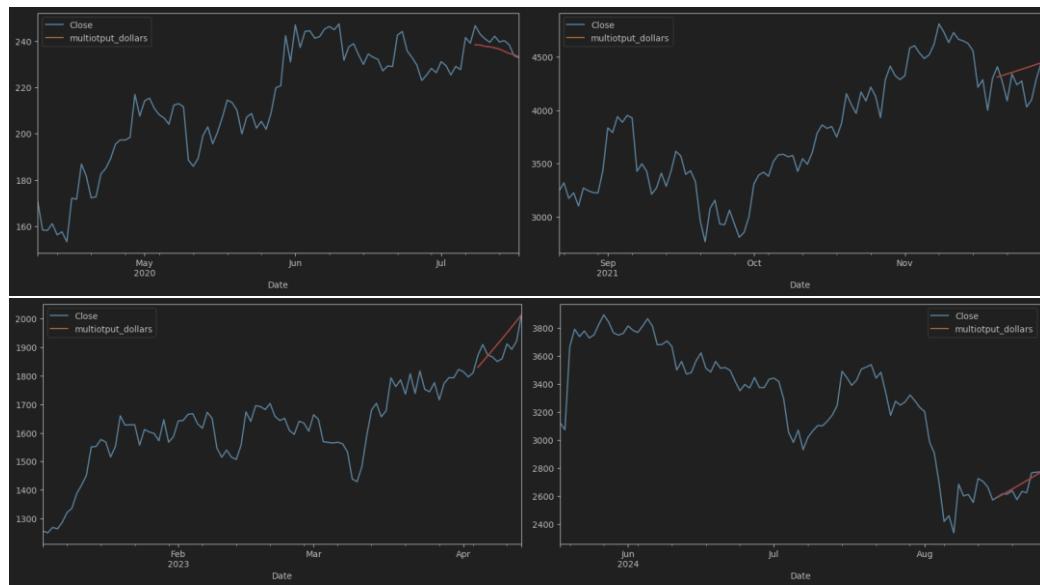
Modeli	Dužina predviđanja	1d			10d			30d		
	Korišćeno dana unazad	T=10d	T=50d	T=100d	T=10d	T=50d	T=100d	T=10d	T=50d	T=100d
Naive ( $R^2$ , MAPE, MAE, MSE, RMSE)		0.9962, 1.95, 55.38, 7850, 88.6			0.9829, 5.59, 135.45, 35621, 188.74			0.91, 12.98, 315.8, 187895, 433.47		
SES	$R^2$	0.9961,	0.9962,	0.9962,	0.9773,	0.9824,	0.9903,	0.7915,	0.9018,	0.9459,
	MAPE	2.09,	1.92,	1.95,	6.15,	5.43,	3.65,	16.41,	13.09,	7.37,
	MAE	58.2,	53.71,	55.3,	149.29,	129.89,	92.71,	440.81,	332.52,	212.72,
	MSE	8215,	7912,	7960,	47360,	36729,	20343,	487878,	229912,	126588,
	RMSE	90.64	88.95	89.22	217.62	191.65	142.63	698.48	479.49	335.79
ARIMA	$R^2$	<b>0.9966</b> ,	0.9962,	0.9963,	0.9913,	0.9949,	0.9949,	0.9048,	0.9528,	0.9387,
	MAPE	2.24,	<b>1.83</b> ,	1.97,	4.02,	3.18,	3.73,	14.68,	9.37,	9.23,
	MAE	55.01,	55.08,	55.29,	90.39,	72.54,	72.96,	239.12,	303.86,	192.69,
	MSE	7080,	7849,	7813,	18073,	10620,	10631,	181717,	288019,	107938,
	RMSE	84.14	88.59	88.39	134.43	103.05	103.11	396.32	446.96	306.58
ANN	$R^2$	0.9962,	0.9964,	0.9961,	0.9948,	<b>0.9953</b> ,	0.995,	0.9824,	0.9823,	0.9823,
	MAPE	1.89,	1.86,	1.89,	2.2,	2.19,	<b>2.12</b> ,	5.51,	5.19,	5.51,
	MAE	55.02,	<b>53.15</b> ,	55.1,	57.36,	<b>55.68</b> ,	56.5,	134.28,	133.87,	134.56,
	MSE	7714,	<b>7504</b> ,	7929,	10204,	<b>9074</b> ,	9850,	42031,	42132,	42049,
	RMSE	83.43	82.14	83.73	82.41	<b>79.58</b>	81.37	180.9	181.01	181
LSTM	$R^2$	0.9963,	0.9963,	0.9963,	0.9947,	0.9948,	0.9947,	0.9823,	<b>0.9824</b> ,	0.9821,
	MAPE	1.88,	1.89,	1.88,	2.24,	2.21,	2.22,	5.13,	<b>5.12</b> ,	5.17,
	MAE	53.95,	54,	53.8,	57.88,	57.03,	57.55,	133.77,	<b>133.45</b> ,	134.35,
	MSE	7604,	7611,	7614,	10374,	10088,	10260,	42110,	<b>41980</b> ,	42435,
	RMSE	81.97	81.97	<b>81.83</b>	83.11	81.88	82.65	181	<b>180.64</b>	181.81

Tabela 6: Zajednički rezultati

## Grafici modela višednevnih predikcija sa najboljim performansama

### Desetodnevna predikcija

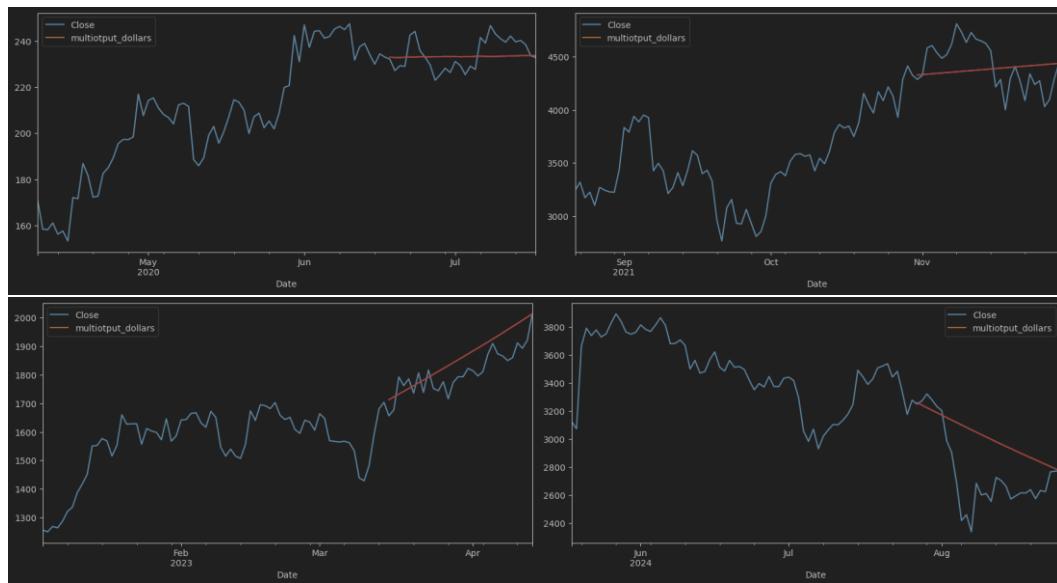
Kako ANN model sa  $T=50$  pobeduje u gotovo svim metrikama za desetodnevnu predikciju, na slikama ispod će biti prikazana predviđanja koja je on uspeo da postigne.



Grafik 4: Desetodnevna predikcija pomoću ANN( $T=10$ )

### Tridesetodnevna predikcija

Pošto je LSTM model sa  $T=50$  pobedio u svim metrikama, na slikama ispod će biti prikazana predviđanja koja je on uspeo da postigne.



Grafik 12: Tridesetodnevna predikcija pomoću LSTM( $T=50$ )

## Zaključak

U ovom diplomskom radu analizirani su različiti modeli za predviđanje vremenske serije kriptovaluta, konkretno ETH-a, uključujući *SES*, *ARIMA*, klasične *ANN* i *LSTM*. Cilj istraživanja bio je utvrditi njihovu efikasnost u odnosu na naivnu predikciju zasnovanu na Random Walk hipotezi. Ispitivani su modeli koji su koristili različit broj prethodnih dana za jednodnevnu, desetodnevnu i tridesetodnevnu predikciju.

Rezultati su potvrdili da složeniji modeli poput *ANN* i *LSTM* pružaju značajno bolje rezultate u predviđanju vremenske serije ETH-a, posebno kako se povećava broj dana za koje se radi predikcija u poređenju sa naivnom predikcijom i jednostavnijim modelima. Efikasnost ovih modela varirala je u zavisnosti od dužine predikcije i broja prethodnih dana korišćenih za treniranje, ali generalno *ANN* i *LSTM* su se istakli kao najpouzdaniji izbori za sve horizonte predviđanja.

Ovi rezultati imaju praktične aplikacije za finansijske analitičare i investitore, pružajući im alate za preciznije predviđanje kretanja kriptovaluta i donošenje informisanijih odluka. Upotreba naprednih modela mašinskog učenja može značajno poboljšati tačnost predviđanja, što je od suštinskog značaja u dinamičnom svetu kriptovaluta.

## Reference

- [1] - <https://finance.yahoo.com/quote/ETH-USD/> , datum pristupa:25.08.2024.
- [2] - <https://ethereum.org/en/> , datum pristupa:10.07.2024.
- [3] - Liliana Gonzalez, Philip Hoang, John G. Powell, Jing Shi „Defining and Dating Bull and Bear Markets: Two Centuries of Evidence“
- [4] - Jonghyeon Min „Financial Market Trend Forecasting and Performance Analysis Using LSTM“
- [5] - <https://jrc.princeton.edu/people/burton-g-malkiel> , datum pristupa:10.07.2024.
- [6] - Malkiel, B. G. (1973.) „A random walk down Wall Street: The time-tested strategy for successful investing (1st ed.)“
- [7] - Malkiel, B. G. (2019.) „A random walk down Wall Street: The time-tested strategy for successful investing (12th ed.)“
- [8] - Samuelson, P. A. (1965.) „Proof that properly anticipated prices fluctuate randomly“ Industrial Management Review, 6(2), 41-49
- [9] - Fama, Eugene F. „Efficient Capital Markets: A Review of Theory and Empirical Work“ Journal of Finance, vol. 25, no. 2, 1970, pp. 383-417
- [10] - The Wall Street Journal, datum pristupa:10.07.2024.
- [11] - Lo, A. W., & MacKinlay, A. C. (1999.) „A non-random walk down Wall Street. Princeton University Press“
- [12] - André Bauer „Automated Hybrid Time Series Forecasting: Design, Benchmarking, and Use Cases“
- [13] - Dickey, D. A., & Fuller, W. A. (1979.) „Distribution of the estimators for autoregressive time series with a unit root“
- [14] - Box, George E. P., and Gwilym M. Jenkins „Time Series Analysis: Forecasting and Control“
- [15] - Seabold, S., & Perktold, J. (2010.) „Statsmodels: Econometric and statistical modeling with Python“
- [16] - Hyndman, R. J., & Khandakar, Y. (2008.) „Automatic time series forecasting: The forecast package for R. Journal of Statistical Software“, 27(3), 1-22
- [17] - Ahmed Nasser B. Alsammak, Ibrahim Ismael Alnaib „Advance Artificial Neural Networks“
- [18] - Alexander Hasse, Lucio Flavio Campanile, Alex Halle „An AI-Assisted Design Method for Topology Optimization Without Pre-Optimized Training Data“

[19] - Dingli Tian, Yu Gong, Luohuan Zou „An Artificial Neural Network-based Approach to Predict the R-curve of Composite DCB Multidirectional Laminates“

[20] - Navid Rajabi, Jihad Qaddour „LSTM Deep Recurrent Neural Network Model for Voltage Abnormality Detection at IoT Gateway“

[21] - Kecheng Peng, Liu Bainian, Xiaoqun Cao „El Niño Index Prediction Using Deep Learning with Ensemble Empirical Mode Decomposition“

[22] - Boris Čuljak „Istraživanje tehnika augmentacije podataka za detekciju žbunova u plantažama borovnica“

[23] - Mladen Blizanac „Detekcija pneumonije upotrebom mašinskog učenja“

[24] - Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P „Stock Price Prediction Using RNN, LSTM and CNN – Sliding Window Model“

[25] - Kriti Pawar, Raj Srujan Jalem and Vivek Tiwari „Stock Market Price Prediction Using LSTM RNN“

[26] - S. Hochreiter and J. Schmidhuber, „LSTM can solve hard long time lag problems in Advances in neural information processing systems“

[27] - Y. Bengio, P. Simard, P. Frasconi and others, „Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks“

## Podaci o kandidatu



Danilo Kaćanski rođen je u Novom Sadu 5. novembra 2001. godine. Prvih šest razreda osnovne škole pohađao je u Osnovnoj školi „Žarko Zrenjanin“, dok je sedmi i osmi razred, kao i srednjoškolsko obrazovanje, završio u Gimnaziji „Jovan Jovanović Zmaj“. Nakon završene gimnazije, 2020. godine upisuje Fakultet tehničkih nauka u Novom Sadu, na smeru Računarstvo i automatika, gde se dve godine kasnije opredeljuje za podsmer Računarski upravljački sistemi.