



**REPUBLIKA E SHQIPËRISË**  
**UNIVERSITETI I TIRANËS**  
**FAKULTETI I SHKENCAVE TË NATYRËS**  
**DEGA: INFORMATIKË**

**Mikrotezë për mbrojtjen e gradës “Master i Shkencave” (MSC)**

**Tema:**

*“Parashikimi i Çmimit të Bitcoin përmes Rrjetave Neural”*

<p><b>Punoi</b> <b>Emër Mbiemër</b> (<u><i>Kejsi Struga</i></u>)</p>	<p><b>Udhëhoqi</b> <b>Grada. Emër Mbiemër</b> (<u><i>Dr.Olti Qirici</i></u>)</p>
--	--

**Tiranë, më 10/10/2018**

## Abstract

In this thesis a neural network model is constructed for predicting the price of Bitcoin. Firstly, the Bitcoin technical mechanisms are analyzed, along with the Cryptocurrency market. Secondly, a fundamental economic analysis is constructed with a focus in Macroeconomics, influencers, and the plausible future. Furthermore, the aforementioned factors and analysis are considered for selecting features deemed appropriate. The Machine Learning problem is considered as regression, and the whole pipeline is implemented where the focus is in the LSTM neural network. The network is explained theoretically and then used for the problem of predicting Bitcoin price. The price is predicted for the future 30 and 60 days. A crucial point is the Hyperparameter Tuning phase to evaluate the network hyperparameters. Finally, the expected conclusion is stated, whereby, the inherent difficulty of predicting a price-related variable persist even in the case of the universal[1] approximators as Neural Networks.

## Abstrakt

Qëllimi i tezës është nxjerrja e nje modeli rrjeti neural që mund të parashikojë cmimin e Bitcoin. Fillimisht kryhet një analizë mbi monedhën në fjalë, duke përfshirë mekanizmat teknik që e mundësojne atë, ekonominë e Bitcoin, ku më tepër ndalemi te faktorët makroekonomik, influencuesit, si dhe të ardhmen e mundëshme. Më tej, mundohemi të nxjerrim një lidhje mes faktorëve të mësipërm, në mënyrë që të dhënat hyrëse të jenë sa më përfaqësuese dhe të koreluara me cmimin. Me njohuritë e mbledhura nga fazat e mëparshme, kryejme të gjithë aktivitet për modelimin dhe implementimin e arkitekturës LSTM për parashikimin e cmimit për 30 ditët e ardhëshme. Problemi konsiderohet si problem regresi, ku mundohemi të parashikojmë një sekuencë vlerash. Gjatë kësaj faze vlerësohen parametra të ndryshëm të rrjetit neural përmes procesit të *Hyperparameter tuning*. Së fundmi arrijmë në konkluzionin e pritur, se parashikimi i cmimit të cdo variabli ekonomik mbart vështirësi në modelimin e problemit, që jo gjithmonë mund të adresohen nga rrjetat neural, pavarësisht universalitetit të këtyre të fundit[1].

# Përbajta

<b>Abstract</b>	i
<b>Abstrakt</b>	ii
<b>Tabelat</b>	vi
<b>Figurat</b>	vii
<b>1 Hyrje</b>	1
1.1 Përshkrimi i problemit . . . . .	1
1.2 Pyetjet e kërkimit . . . . .	2
1.3 Cfarë nuk përfshihet në studim . . . . .	2
1.4 Fjalor . . . . .	2
1.5 Shkurttime . . . . .	3
1.6 Organizmi i tezës . . . . .	3
<b>2 Protokolli Bitcoin</b>	4
2.1 Cfarë është Bitcoin? . . . . .	4
2.2 Historiku dhe zhvillimi i Bitcoin . . . . .	6
2.2.1 Monedhat elektronike përpala Bitcoin . . . . .	6
2.2.2 Përdoruesit dhe investitorët filletar . . . . .	10
2.3 Mekanizmat e Bitcoin . . . . .	16
2.3.1 Funksionet Hash . . . . .	17
2.3.2 Nënshkrimet dixhitale . . . . .	21
2.3.3 Çelësat publik kundrejt identiteteve . . . . .	22
2.3.4 Sistemet e Decentralizuar . . . . .	23
2.3.5 Konsensusi i shpërndarë . . . . .	24

2.3.6	<i>Incentives</i> dhe <i>Proof of Work</i> . . . . .	29
2.3.7	Rrjeti i Bitcoin dhe sfidat . . . . .	34
2.3.8	Transaksionet në Bitcoin . . . . .	35
2.3.9	BTC Exchanges . . . . .	40
2.3.10	Aktivitetet e mining . . . . .	41
2.3.11	Sfidat teknike të Blockchain-eve . . . . .	43
2.4	Ethereum dhe Smart Contracts . . . . .	45
<b>3</b>	<b>Tregu dhe ekonomia e Bitcoin</b>	<b>48</b>
3.1	Si t'i klasifikojmë Cryptocurrency-të? . . . . .	48
3.1.1	Besimi dixhital . . . . .	48
3.1.2	Gjenerata e ndryshimit . . . . .	48
3.2	Forcat e Kërkesë-Ofertës . . . . .	50
<b>4</b>	<b>Rrjetat e Thëllë Neural</b>	<b>53</b>
4.1	Rrjetat neural me shumë shtresa (MLP) . . . . .	53
4.1.1	Neuronet jo-linear . . . . .	54
4.1.2	Perceptroni me shumë shtresa . . . . .	56
4.1.3	Backpropagation . . . . .	57
4.1.3.1	Algoritmi i back-propagation . . . . .	58
4.2	Rrjetat neural me përsëritje . . . . .	60
4.2.1	Arkitekturat . . . . .	61
4.3	LSTM . . . . .	62
4.3.1	Arkitektura e LSTM . . . . .	63
<b>5</b>	<b>Parashikimi i Serive Kohore</b>	<b>66</b>
5.1	Përpunimi dhe marrja e të dhënave për rrjetat neural . . . . .	68
5.1.1	Ekstraktimi i të dhënave . . . . .	68
5.1.2	Përpunimi i të dhënave . . . . .	70
5.1.3	Normalizimi të dhënave . . . . .	71
5.2	Përshtatja e të dhënave të serive kohore për rrjetin neural . . . . .	72
5.2.1	Ndarja e të dhënave në bashkësi trajnimi dhe testimi . . . . .	72
5.2.1.1	Ndarja të dhënave në inpute dhe outpute . . . . .	72
5.2.2	Modeli LSTM . . . . .	73
5.2.2.1	Hiperparametrat e modelit . . . . .	73

5.3 Rezultate . . . . .	76
<b>6 Konkluzione</b>	<b>79</b>
<b>Bibliografia</b>	<b>81</b>
A Software dhe tools te përdorura	84
B Implementimi dhe shtesa të tjera	85

# Tabelat

4.1 XOR . . . . .	53
-------------------	----

# Figurat

2.1	Hash rate i hardware-it të miner-ave i matur në Tera Hashes për sekond (triliona hash-e për sekond) . . . . .	5
2.2	Vështirësia e thyerjes së hash-it ( <i>Difficulty</i> ) . . . . .	5
2.3	Studim nga Pew Research Center. Besimi i publikut në qeverinë e Sh.B.A, nën udhëheqjen e presidentëve të ndryshëm [17] . . . . .	15
2.4	Cmimi historik i Bitcoin . . . . .	17
2.5	Perplasje mes 2 inputeve në të njejtin digest. . . . .	18
2.6	Struktura e një Hash pointer . . . . .	19
2.7	Struktura e një Blockchain . . . . .	20
2.8	Struktura e një Merkle Tree . . . . .	21
2.9	Skema klasike e shpenzimit të dyfishte në blockchain . . . . .	27
2.10	Shperblimi për bllok, per gjysmohet cdo 4 vite duke dhene një total prej 21 milion Bitoin . . . . .	31
2.11	Menyra se si lidhen transaksionet përmes hash-it (logjike e njejte me blockchain-in)[6] . . . . .	35
2.12	Shembull i skriptit më të zakonshem: Pay-to-PubkeyHash . . . . .	37
2.13	Shpërndarja e fuqisë së hash-it në botë . . . . .	43
3.1	5 kompanite teknologjike në S&P 500, vlejne me shumë se 282 kompani të tjera, në baze të market capitalization . . . . .	49
3.2	Studim i Pew Research Center, me pyetje: <i>Sa besoni se qeveria benignë e duhur për shtetin?</i> . . . . .	49
3.3	Market cap dhe oferta . . . . .	52
4.1	Pse XOR solli MLP . . . . .	53
4.2	Zgjidhja e XOR me një shtresë të fshehur mes inputit $(x_1, x_2)$ dhe outputit . . . . .	54

4.3	Funksioni i aktivizimit Sigmoid . . . . .	56
4.4	Funksioni i aktivizimit Tanh . . . . .	56
4.5	Rrjeti neural me 2 shtresa të fshehura, 3 të dhëna hyrese, 4 neurone në secilen shtresë të fshehur, dhe 2 vlera në dalje . . . . .	57
4.6	Grafi funksionit të regresit linear: $\hat{y} = \sigma(w^\top x + b)$ . . . . .	59
4.7	Rrjeti më i thjeshtë me përsëritje, ku ka vetëm një shtresë të fshehur	61
4.8	Shpalosja e rrjetit neural me përsëritje, ku një input varet nga 5 inputet e mëparshme, pra <i>timestep</i> -i është 5 Rreshti i parë i neuroneve përfaqëson shtresën e inputeve, i dyti shtresën e fshehur, dhe në fund shtresa e outputit. . . . .	62
4.9	Pamje e detajuar e njesise LSTM, se bashku me qelizen e gjendjes dhe njesite e perditesimit, outputit dhe inputit [31] . . . . .	64
5.1	Grafiku i cmimit të Bitcoin, dhe ndryshimet e thella në cmim . . . . .	67
5.2	Bashkesia e të dhënavë për parashikimin e cmimit, ketu paraqitet vetëm cmimet e exchange-it të Bitcoin . . . . .	70
5.3	Korrelacioni i Pearson për atributet e datasetit. Vlera 1 është maksimumi	71
5.4	Ndarja bashkesise se të dhënavë fillestare . . . . .	73
5.5	Bias-variance (underfitting vs overfitting), në kontekstin e regularizimit, duam dicka midis. . . . .	75
5.6	Grafiku që tregon se si errori zvogëlohet gjatë trajnimit të modelit . .	76
5.7	Parashikimi në bashkësinë e trajnimit dhe errori. Parashikimi është për 30 dite . . . . .	77
5.8	Parashikimi në bashkësinë e testimit dhe errori. Parashikimi është për 30 dite . . . . .	77
5.9	Parashikimi në bashkësinë e testimit. Parashikimi është për 60 ditet e fundit . . . . .	78
B.1	Veprimet në rrjetat neural . . . . .	85
B.2	Ilustrim i një <i>candlestick</i> , ku paraqiten cmimet e ndryshme të një stoku, në terminologjinë e analizës teknike . . . . .	102

# Kapitulli 1

## Hyrje

### 1.1 Përshkrimi i problemit

Bitcoin është një fenomen i cili nuk mund të anashkalohet, jo vetëm për shkak të gjenialitet që mbart zgjidhja për të eliminuar palët e treta financiare, por gjithashtu për arsyen e rritjes pothuajse pingul të cmimit, gjatë vitit 2017. Nga ana tjetër, Bitcoin i hapi dyert një tregu të ri, atij të Cryptocurrency-ve, por jo vetëm kaq, Distributed Ledger Technologies, nën të cilat perfshihen një gamë implementimesh të ndryshme të Blockchain, janë adoptuar në industri, duke ekzistuar si *Proof of Concepts* apo janë plane për të ardhmen [2]. Duke marrë parasysh këtë impakt të Bitcoin në ekonomi dhe teknologji, lind pyetja se si do të jetë e ardhmja e tij? Është thjeshtë një flusk, e përkrahuar nga spekulatore? Është komuniteti i Bitcoin i aftë të ndryshojë rrënjosish sistemin monetar, apo do të përbëjë një pjesë të vogël të tij? Cilat janë përfitimet që marrim nga përdorimi i Bitcoin, a varet kjo nga shteti në shtet? Këto janë disa nga problemet që shtrohen nga ekonomistët, informaticienet, kriptografët, përdoruesit e sistemit, etj. Parashikimi i cmimit na lejon të krijojmë një opinion mbi ekonominë e tij, si dhe të mund t'i japim përgjigje pyetjeve me lartë. Akoma më tej, ballafaqohemi me problemin ekzistues në sëritë kohore të cmimit të aseteve ekonomik, në të cilat parashikimi i të ardhmes varet shumë nga vetë ajo, ku kjo e fundit mund të ndikohet nga faktor të pa dukshëm apo që nuk ekzistojnë në të dhënrat historike. Përvécse këto janë probleme që mundojnë ekspert të fushave të lartpërmendura, Bitcoin-i duke qenë një monedhë, prek cdo shtresë të popullsisë, kështu që problemi mbi cmimin i përket cdo pjesmarrësi të ekonomisë ekzistente.

Një tjetër problem, është se shumë studime në lidhje me cmimin e Bitcoin-it kryhen në per gjithësi nga ekonomistë, të cilët e analizojnë atë duke lënë mënjean detaje të mekanizmave të Bitcoin që e bëjne atë unik [3]. Kështu, avantazhi ynë është se mund ta analizojmë problemin jo vetëm nga ana ekonomike por edhe nga ajo informatike.

## 1.2 Pyetjet e kërkimit

Disa pyetje që shtrojmë për t'ju përgjigjur gjatë tezës janë si mëposhtë:

1. Me cfarë saktësie mund të parashikohet cmimi i Bitcoin?
2. Cilat atribute mund të ndikojnë në cmimin e tij?
3. Cilat janë disa udhëzime që mund të ndiqen kur parashikojmë seri kohore me shumë volatilitet, përmes rrjetave neural?

## 1.3 Cfarë nuk përfshihet në studim

Ndërkohë që vizioni i Bitcoin është të zëvëndësojë sistemet monetare që operojnë me monedha *fiat* [4], një pjesë e Cryptocurrencieve të tjera, cmimi i të cilave është afër Bitcoin-it, kanë qëllime që mund të mos jenë të lidhura me ekonominë. Si rrjedhojë, fokusi mbahet tek Bitcoin, në mënyrë që analiza dhe faktorët që merren në konsideratë të janë sa më të sakta nga ana logjike.

## 1.4 Fjalor

**Miner** Nyje në rrjet që gjen zgjidhjen e *proof-of-work* për bllokun e ri në Blockchain, duke provuar inpute të ndryshme për funksionin hash

**Wallet** Software që ka si qëllim të ruaj celësat publik-privat dhe të lehtësoj kryerjen e transaksioneve

**Bitcoin Core** Bitcoin client dhe bitcoin wallet

**Proof-of-work** Output-i i gjetjes së vlerës së hashuar

**Vështirësia / Difficulty** Variabli që determinon sa e vështirë do jetë gjetja e një hash-i. Blloqet e validuara, duhet të kenë një hash poshtë vlerës së këtij variabli.

**51% Attack** Një miner, që zotëron më shumë se 50% të fuqisë kompjuterike të rrjetit (hash power), mund të modifikojë dhe ndryshojë rradhën e transaksioneve, duke bërë që monedhat të përdoren më shumë se njëherë.

**Nonce** Në kriptografi (dhe web security) *nonce* është një vlerë random që përdoret vetëm njëherë.

**Market Capitalization** Numri i stokeve shumëzuar me cmimin e tyre , e tillë që këto stoke zotërohen nga të gjithë pronarët, duke përfshirë investitorët e jashtëm (*shares outstanding*) [5]

## 1.5 Shkurtime

**Tx** Transaksion

**BTC** Bitcoin

**DLT** Distributed Ledger Technologies

**LSTM** Long Short-Term Memory

**RNN** Recurrent Neural Network

## 1.6 Organizmi i tezës

Në vijim, teza strukturohet në 5 kapituj. Fillimisht, diskutohet zhvillimi dhe evolimi i Bitcoin-it që nga publikimi i tij. Së dyti, flasim për Bitcoin-in nga një pikpamje teknike. Këtu prezantojmë ato vecori që e bënë atë të rezistojë përgjatë 10 viteve, dhe cilat janë konceptet e reja (por jo dhe aq) më të cilat Bitcoin-i na njeh. Në kapitullin e tretë shtrohet problemi i ekonomise së këtij sistemi, si ndryshon Bitcoin-i nga sistemet aktuale ekonomike, dhe cilat janë tiparet makroekonomike të Bitcoinit. Në fund tregojmë si mund të përdoren rrjetat neural me seritë kohore, ekstraktimi i të dhënavë përmes Web API nga *end-point*-et përkatëse, përshtatja e të dhënavë në formë të pranueshme nga një algoritëm i të mësuarit të supervizuar, implementimi në TensorFlow dhe Keras. Në fund e mbyllim me rezultatet e këtij parashikimi.

# Kapitulli 2

## Protokolli Bitcoin

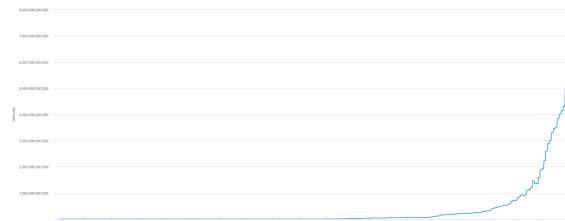
### 2.1 Cfarë është Bitcoin?

Bitcoin-i është një Cryptocurrency, në sensin që të gjitha veprimet monetare, si p.sh. transaksione dhe krijimi monedhave të reja sigurohen përmes primitivave kriptografike. Bitcoin-i është një rrjet *Peer-to-Peer* (i ngjashëm me BitTorrent), ku pjestarët e ekonomisë janë nyjet e rrjetit dhe kryejnë transaksione me njëri-tjetrin. Rrjeti i Bitcoin është i decentralizuar dhe i shpërndarë. I decentralizuar, meqë nuk ka një autoritet qëndror, dhe i shpërndarë meqë procesimi i transaksioneve nuk bëhet vetëm nga një nyje. Përdoruesit e Bitcoin përdorin protokollin e Bitcoin-it për të komunikuar përmes internetit. Ndryshe nga Web-i, rrjetin e Bitcoin mund ta konsiderojmë si *standalone*, pasi nuk përbehet nga protokolle dhe second layer technologies të cilat bëhen update në mënyrë të pavarur nga njëri-tjetri. Bitcoin-i është open-source software, dhe mund të ekzekutohet në smartphones, laptops, PCs, etc. Përdoruesit në rrjetin P2P, mund të kryejnë të gjitha veprimet e mundëshme me monedhat e sotme, me avantazhin që komunikimi është i sigurtë, nuk njeh kufinj gjeografik, dhe veprimi nuk kthehet mbrapsht. Bitcoin-et krijohen përmes procesit të *mining*, i cili përfshin gjetjen e inputit të një funksioni hash. Cdo pjestar në rrjetin e Bitcoin mund të sillet si miner, me kushtin që të ketë një nyje të plotë. Pothuajse cdo 10 minuta, gjendet një zgjidhje e hash-it, dhe si rrjedhojë transaksionet e 10 minutave më të fundit validohen, miner-i shperblehet për energjinë e harxhuar përmes BTC të reja. Procesi i mining është një ndër pjesët më kryesore, duke qenësë është pikërisht konkurenca mes miner-ave për të gjetur sa më shpejt Proof-of-Work, mekanizma që zëvendëson

bankën. Vështirësia e thyerjes së hash-it është një funksion i fuqisë se hardware-it të miner-it. Ajo ndryshohet dinamikisht në menyrë që mesatarisht hash-it t'i jepet zgjidhje cdo 10 minuta, dhe jo me shpejt (Fig. 2). Protokoli gjithashtu per gjysmon sasinë e Bitcoin-eve të krijuar cdo 4 vite, si rrjedhojë numri total i Bitcoin-eve është afërsisht 21 milion. Duke marrë parasysh këto dy fakte, numri total prej 21 milion do të arrihet në vitin 2140. Ky prodhim regresiv i Bitcoin-eve ka si qëllim të eliminojë inflacionin. Akoma më tej, Bitcoin-i ndalon inflacionin, duke qenëse nuk mund të "printohet" monedhë përtej sasisë së përcaktuar nga protokolli. Bitcoin është gjithashtu emri i protokollit, një rrjet nyjesh të shpërndarë. Monedha Bitcoin ishte aplikacioni i parë i këtij rrjeti, dhe më vone kemi inovacione të tjera, si Ethereum, ku ideja e monedhës, zgjerohet, duke përfshire ato cka i quajmë *Smart Contracts*, përmes së cilave jo vetëm monedhat mund të transferohen në një Blockchain, por cdo aset, ide, aplikacion, etj. (mjafton që programohet)



**Figura 2.1:** Hash rate i hardware-it të miner-ave i matur në Tera Hashes për sekond (triliona hash-e për sekond).



**Figura 2.2:** Vështirësia e thyerjes së hash-it (*Difficulty*)

## 2.2 Historiku dhe zhvillimi i Bitcoin

### 2.2.1 Monedhat elektronike përpara Bitcoin

Duke konsideruar sfidat që lidhen me paraqitjen e monedhave në formë bitesh, krijimi i skemave të parave dixhitale lidhet ngushte me zhvillimet kriptografike. Dy pyetjet krysore që lindin nga kushdo që pranon para dixhitale janë [6]:

1. A mund të besoj se paraja është autentike dhe jo e falsifikuar?
2. A mund të sigurohem se këtë para e zotëroj vetëm unë, dhe askush tjetër nuk mund të pretendoj se i përket atyre? (i njohur dhe si problemi i "shpenzimit të dyfishtë" ose "*Double Spend problem*")

Në rastin e qeverive që prodhojnë para, të dyja problemet kanë zgjidhje, për të parën ndihmon teknologja e printimit dhe perdonimi i materialeve të sofistikuara për të krijuar letrën. Problemi i dytë, nuk është fare ekzistent në rastin e parasë leter, duke qenë ka formë fizike dhe nuk mund të ndodhet në dy vende njekohësisht. Në rastin kur paraja konvencionale ruhet dhe transferohet elektronikisht, të dy problemet adresohen nga autoritetet qëndrore, të cilat duke qëne të centralizuara kanë një pamje globale të parave në qarkullim. Për paranë elektronike, nënshkrimet dixhitale përbëjnë bazën e verifikimit se aseti nuk është fallco. E njëjtë primitivë kriptografike, më pak ndryshime përdoret edhe për të zgjidhur problemin e shpenzimit të dyfishtë. Rreth fundit të vitit 1980 [6, p. 2], kriptografia filloj të përhapej dhe kuptohej më

mirë, si rrjedhojë studiuesit filluan ta shihnin atë si një mundësi për të krijuar monedha dixhitale të implementuara përmes kriptografisë. Disa sisteme të tillë janë:

1. **Digicash** (Ecash) 1991 - David Chaum [7] Në skemën e Chaum klientët janë anonim, kështu që bankat nuk gjurmojnë dot mënyrën se si shpenzohen paratë. Por tregtarët nuk janë anonim, ata duhet t'i kthejnë monedhat që në momentin që arrijnë nga klientët, në menyrë që banka të dijë sa po fitojnë, kohën kur ndodhin pagesat, etj. Protokolli për të mbajtur klientët anonim përdoret akoma për blerje online dhe njihet si *Chaum's Anonymity*. Në DigiCash, vlera e monedhës determinohet nga banka që lidhet me sistemin, pra nëse përdoruesi dëshiron të ketë \$50, atij do i duhej të tërhiqte \$50 nga llogaria e bankës së lidhur me

ECash. Por ekzistonin propozime të ndryshme mbi mënyren se si kjo mund të arrihej, dhe cdo kompani ofronte qasjen e vetë, disa prej të cilave (NetCash, Digicash) ofronin Ar në këmbim të parasë elektronike. Pra, të gjitha conin në përdorimin e dollarit ose ndonjë komoditeti tjetër për krijimin e monedhës në Digicash. Nëse vlera e dollarit bie, e njëjtë ndodh me vlerën e parasë së Digicash. Një qasje komplet ndryshtë e lejosh paranë elektronike të jetë edhe monedhë, e krijuar dhe vlerësuar në menyrë të pavarur nga monedhat e tjera. Dhe për të krijuar dicka të tillë, duhet dicka që e eshtë e limituar. Limiti në sasi, e eshtë arsyja sepse diamantet apo ari janë përdorur si objekte që përfaqësojnë paranë. Në sistemet elektronik, një mënyrë modelimi për një sistem të tillë e eshtë që krijimi i parasë të bëhet kundrejt zgjidhjes së një problemi të veshtirë (që merr shumë kohe) matematike, si pasojë:

- Jo cdokush mund të krijoj para
- Koha për të zgjidhur problemin ben që mos të jete e lehte të krijohen monedha të reja

2. **Hashcash** 1997 - Adam Back [8] Sistemi për ndalimin e email-eve spam e eshtë i famshëm nga Adam Back, dhe kjo ishte ideja ku u bazua HashCash, dhe me pas algoritmi i mining në Bitcoin.
3. **B-Money** 1998 - Wei Dai [9] B-money dhe autor i saj, janë dy nga influencuesit më të forte në skemen e Bitcoin. Sipas Web Dai në një shoqeri kripto-anakrike<sup>1</sup>, qeveria nuk e eshtë e nevojshme, pasi dhuna në komunitet e eshtë e pamundur pasi pjestarët nuk lokalizohen dot meqë janë anonim. Akoma më tej, sipas Dai-të problemi me sistemin financier të centralizuar që kemi sot, e eshtë se kontrrolli i parasë e eshtë në një dore të vetme, atë të qeverise, dhe gjithë menaxhimi i dickaje që i perket të gjitheve realizohet nga një pale e vetme. Qellimi i Dai-të ishte krijimi i ekonomive online që janë totalisht vullnetare, pa taksa apo menaxhim të centralizuar [11]. Dy zgjidhjet që dha Dai janë si mëposhtë:

- (a) Në zgjidhjen e parë, ledger-i ku ruhen të gjitha transaksionet nuk janë të centralizuara në një entitet të vetëm, por secili pjestar i sistemit mban

---

<sup>1</sup>Kripto-anarkizmi e eshtë ideja themelore e levizjes *Cypherpunk* [10], ku pjestarët sugjerojnë se interneti së bashku me software dhe kriptografi duhet të përdoren për të mbrojtur lirinë politike, ekonomike, dhe sociale të individit duke i mbajtur ato anonim.

një e të ledger-it. Sa here që një transaksion i ri ndodh kopja e secili behet update. Keto kopje konsistojne në celsa publik të shoqeruar nga sasi monedhash. Kjo mënyrë decentralizimi nuk lejon ekzistencen e një autoriteti qendror që bllokon transaksionet, njëkohësisht ofron privatesi për përdoruesit. Për ta ilustruar me një shembull, supozojme se Alice dhe Bob janë të dy pjestare të B-money, Alice ka celesin publik "A" dhe Bob "B", gjithashtu kanë dhe celesat korrespondues privat. Supozojme se secili ka 3 b-money (emri i monedhës, është i njejtë me atë të sistemit). Nëse Bob deshiron të marrë 2 b-money nga Alice, ai i dergon asaj celesin publik B. Alice me pas formon një transaksion ku nxjerr 2 b-money nga llogaria e saj për tek celësi publik që mori. Më pas e nënshkruan këtë transaksion me celesin e saj privat. Transaksiioni se bashku me nenshkrimin kriptografik i dergohen *të gjithë* përdorueseve të sistemit b-money. Mesazhi i nenshkruar sherben si prove që entjeti me celës publik A dergon 2 b-money tek përdoruesi B. Si rrjedhojë, të gjithë bëjnë update ledger-at e tyre, e tille që Alice ka 1 b-money më pak, dhe Bob ka 1 b-money me shumë. Problemi i kësaj qasje ishte se monedha mund të shpenzohej dyhere, pasi asgje nuk e ndalonte. A mund t'i dergonte b-money-të tek B dhe C njekohësisht, duke i derguar transaksionet në ane të ndryshme të rrjetit. të dy, B dhe C do i pranonin monedhat duke menduar se cdo gje është në rregull, dhe vetëm me vone do kuptonin se balanca e tyre nuk do të pranohej nga nyjet e tjera në rrjet. Gjithësesi, kjo skemë u përdor 10 vite me vone, në Bitcoin, ku u eleminua problemi i shpenzimit të dyfishte përmes miner-ave.

- (b) Në zgjidhjen e dyte, në vend që cdo përdorues të mbaj një kopje të ledger-it, sistemi njeh 2 tipe përdoruesish: përdorues të rregullt, dhe "servera". Vetëm serverat lidhen me rrjetin Usenet<sup>2</sup>, për të mirembajtur ledger-in. Për të verifikuar se një transaksion u krye me sukses, përdorueseve të rregullt si Alice dhe Bob, do i duhej të verifikimin me një nengrup të rastisishem të këtyre serverave. Në rastin e transaksiointit me 3 paleve të permendur me lartë, B dhe C nuk e prandojne transaksiionin me zgjidhjen e dyte. Dhe pse Dai nuk e thotë në mënyrë implicite, mund të themi se cdo

---

<sup>2</sup>UseNet është një rrjet serverash ku mesazhet shperndahen në newsgroups. Serverat, ndryshe nga WWW ku janë standalone, lidhen me njeri-tjetrin, dhe përdoruesit lidhen me keta të fundit, përmes TCP/IP, mesazhet shkembehen përmes protokollit NNTP

entitet kishte mundesi të behej server, por cdo server duhet të depozitoje një sasi parash në një llogari të vecante për t'u përdorur si shperblim për vertetimin e shpenzimit të dyfishte. Cdo pjestare duhet të verifikoje se balancat në llogarite e tije nuk janë me shumë se totali i parave të krijuara. Kjo parandalon serverat të krijojne para në mënyrë të pakonrrolluar. Ky mekanizem, është i ngjashem jo me Bitcoin-in por me algoritmin e mining që përdor Ethereum: proof-of-stake. Gjithësesi, aty ku Bitcoin do të

ndryshonte nga B-money, janë politikat monetare. Në Bitcoin, sasia e krijuar e parave, filloi me 50 BTC për bllok, dhe sot është 12.5 BTC. Satoshi nuk merr parasysh gjëtjetër për këtë skemë, vecse ajo parandalon deflacionin dhe inflacionin. Ndërkohë, Dai, dha idene e një monedhë stabël, ku vlera e një b-money do të shoqerohej me vleren e një shporte produktesh (teorikisht). Pershembull, 100 b-money do ishin ekuivalent me një shport produktesh. Kjo duhet ti jape një vlere stabël monedhës, të pakten në lidhje me shporten: të njejtat 100 b-money do të blenin të njejten shporte në të tashme, të shkuar, dhe të ardhme. Për të krijuar monedha të reja, përdoruesit duhet të determinonin se sa do të kushtonte një shporte në krahasim me zgjidhjen e një problemi matematik: "proof-of-work". Nëse, në një kohe të caktuar, një shporte prodhimesh kushton \$80, do duhej të behej match me nga një proof-of-work që kushton \$80 për t'u prodhuar. Duke përdorur këtë indikator, personi i parë që do të prodhonte një proof-of-work të vlefshme do të shperblehej me 100 b-money të reja nga të gjithë përdoruesit e serverave. Si rrjedhojë, askush nuk do të motivohej për të prodhuar proof-of-works vec rastit kur do donin të perdornin b-money, duke limituar inflacionin e monedhës.

Fatkeqesisht, B-money ngeli në nivelin e një "thought experiment" pa u implementuar kurrë. Fatmiresisht, puna e Dai-të është referenca e parë në paper-in e Bitcoin.

4. **BitGold** 2005 - Nick Szabo [12] Kjo cryptocurrency, përdor timestamping, dhe për krijimin e programit përdoret Hashcash. Ajo cka i mungonte BitGold-it ishte një mekanizem për ti mbajtur nyjet e ndershme. Tjeter problem i madh,

ishte se i njejti token mund t  krijohej me veshtir si t  ndryshme.

5. **Bitcoin** 2008 - Satoshi Nakamoto [13] Sic dihet Bitcoin p rdor:

- Cel sat publik p r t  siguruar pseudoanonimitetin
- Timestamping
- Krijimin e Bitcoin-eve p rmes Hashcash
- Role t  ndryshme p r nyjet: miner-at jan  t  ndershem (veshtiresia varion n  baze t  fuqise se Hardware-it t  miner-it)
- Pemet Merkle p rdoren p r t  grumbulluar transaksionet brenda nj  bllokut

Pra sic duket, Bitcoin ka filluar shum  m  par  se viti 2008. Bitcoin-i dhe si rrjedhoj  t  gjith  Blockchain-et sot, jan  thjesht  vazhdimi i ideve t  Cypherpunk. Ato ofrojn  nj  zgjidhje teknologjike p r problemet sociale dhe politike.

### 2.2.2 P rdoruesit dhe investitor t fillestar

Levizja Cypherpunk, e cila nise n  1993 me nj  paper t  publikuar nga matematiaci Eric Hughes, fjalet e para t  se ciles ishin: "Privatesia n  boten dixhitale  sht  e nevojshme p r nj  shoqeri t  hapur". Mosbesimi ndaj qeverise  sht  dicka normale p r persona t  lidhur me informatike, t  cilet shpenzojne pjes n m  t  madhe duke u mbeshtetur n  kod, ku praktikisht je totalisht i pavarur, dhe kjo  sht  nj  nd r avantazhet e programimit, fuqia q  i jep individit. Sic u tha dhe me lart , lveizjet kriptografike ben t  mundur q  k to ide t  mbroheshin. Nga historia e dim  se teknologjite e enkriptimit kan  qen  privilegji i vet m disa prej institucioneve t  pushtetshme. Individet mund t  enkriptonin komunikimin, por ajo mund t  deshifrohej shum  lehte nga qeveria. Gjat  viteve 1907 deri n  1980 [14], matematicienet n  Stanford dhe MIT arriten t  krijonin kriptografine me cel s publik, si m nyr  q  lejonte njer zit e zakonshem t  shifronin mesazhe t  cilat nuk do mund t  deshifrohesin as nga superkompjuterat e atehershem, pervecse nga marresit e mesazhit (PGP  sht  nj  nga aplikacionet m  t  hereshme t  k tyre ideve, e drejtuar nga David Chaum).

Idealet ishin t  shumta asokohe, por paraja ishte q  nga fillimi, aty ku mblidheshin t  gjith  forcat p r t  rikrijuar t  ardhmen. Paraja  sht  p r cdo treg ekonomik nj lloj si ajri dhe uji p r qenien njerezore. N  at  kohe, p r programuesit, monedhat ekzistuese, t  cilat ishin t  vlefshme vet m brenda p r brenda kufinjeve shteteror,

dhe ishin vulnerabel ndaj teknologjive që banka zgjidhte të perdorte, dukeshin të kushtezuara pa shkak. Pa cka se paraja fizike ofron anonimitet në kryerjen e transaksioneve, kjo nuk është e vertet në sistemet E-banking që sot transferojne para dixhitale. Cypherpunk po kerkonte mënyra përmes të cilave të perdorej paraja në formë dixhitale pa sakrifikuar privatesine e perdonuesve. Por për pjesën më të madhe të njerezve modern, paraja ka lidhje të ngushte me qeverinë. E drejta për të krijuar para përkufizon fuqitë e një shteti. Por edhe po të shkojme më parë në histori, deri gjatë Luftes Civile pjesa më e madhe e parave në qarkullim në US krijohej nga bankat private, si rrjedhojë, nëse bankat rrezoheshin paraja thjeshtë nuk kishte me vlere. Kjo ishte thjeshtë rrjedha normale e njerëzimit për të gjetur mënyra më të mira se sa metalet, për të perfaqesuar vlere. Kërkimi për një formë më të mirë paraje, ka qenë gjithmonë për të gjetur gjera të cilave mund t'i besojme dhe vleresohen në mënyrë uniforme - një metrike që lejon krahasim mes vlerave të objekteve të ndryshme. Sic thotë edhe Nigel Dodd - "paraja e mirë është e afte të konverteje ndryshimet kualitative mes produkteve, në ndryshime sasiore, në mënyrë që të mundesoje shkembimin." Paraja e imagjinuar nga Cypherpunks kerkonte ta standartizonte karakterin e parasë në ekstremet logjike, duke lejuar kështu një para universale që mund të shpenzohej kudo, ndryshe nga monedhat e sotme që janë specifike në nivel shteti. Paraja e mirë, ka kualitetet e mëposhtë:

- Rezistente
- E transferueshme
- E ndashme në njesi më të vogla
- Uniforme
- E limituar në prodhim (scarce)
- E besueshme nga përdoruesit e saj

Kualiteti i fundit, natyrisht që është më pak e prekshme (tangible). Nëse një fermer do të pranoj 1000 Lek, ai duhet të besoje se ky 1000 Lek, ka vlere në të ardhme. Ky kualitet fundamental që një para ta ruaj suksesin përgjatë kohes, nuk ka lidhje me entitetin që e prodhon atë - apo sa rezistente është - ka thjeshtë lidhje me njerëzit, dhe sa të gatshem janë ata për ta përdorur atë. Gjatë shekullit të 20, dollarri ishte

monedha globale, pasi shumica e njerezve *besonin* se Shtetet e Bashkuara dhe sistemi i tyre finansiar kishte me tepër prospekt se alternativa të tjera. Dhe kjo shpjegon pse njerëzit konvertonin dhe shisnin monedhat e tyre lokale për dollar.

Gjatë një fjalimi në 1996, Alan Greenspan, dikur udhehëques bordi (chairman) i Federal Reserve në US, shprehu idene e tij se paraja ishte dicka që jo vetëm bankat qëndrore mund ta krijonin. Ai imagjinonte se revolucioni teknologjik mund të risillte potencialin për para private dhe se kjo mund të ishte dicka pozitive: "Mund të imagjinojmë propozime të ardhshme nga prodhues të parave elektronike për të vendosur korporata të specializuara me balanca dhe histori të fuqishme krediti". [15].

Vetëm disa vite pas fjalimit të Greenspan, në 1997, studiuesi Britanik për të cilin folem në seksionin e meparshëm, Adam Back, publikoi në Cypherpunk mailing list, planin e tij për atë cka e quajti hashcash, dhe zgjidhte problemin e famshëm, atë të mos kopjuarit<sup>3</sup> pafundësish të file-ve elektronike. Ideja ishte shumë e zgjuar, dhe për me tepër është pjesë edhe e Bitcoin. Nga vetë emri, e kuptojmë se hashcash përdor funksionet hash, të cilet sic e dimë ndryshojne nga funksionet e enkriptimit duke qenëse të dhënën në input nuk e marrim dot mbapsht (praktikisht). Ajo cka ngelet për të gjetur input-in është të provosh vlera të ndryshme rastësore dhe të presesh se cila e verteton ekuacionin. Kur kompjuteri të gjeje këtë vlere atëherë ai fiton hashcash. Krijimi i hashcash në këtë mënyrë, ishte i vlefshëm duke konsideruar se monedha nuk kopjohej pa fund. Kompjuterit i duhej të performonte shumë pune, për të krijuar njesi të reja hashcash-i, si rrjedhojë mori emrin - "proof-of-work- që me vone u konsiderua si inovacioni që mundesonte Bitcoin-in. Problemi kryesore me sistemin monetar të Back, ishte se cdo njesi hashcash mund të perdorej vetëm njehere dhe cdo njeriu në sistem i duhej të krijonte njesi të reja sa here që i duheshin të tilla. Tjeter problem ishte se sistemi nuk merrte parasysh se cka ndodh në rastin kur një pjestar në hashcash ka një fuqi shumë të lartë procesimi. Natyrisht, BitGold erdhi pak me vone në skene, nga Nick Szabo, ekspert siguri dhe pjestare i Cypherpunk. Me kalimin e kohes, pjestarët e Cypherpunk, nuk kishin ngelur thjeshtë tek privatesia e transaksioneve, ata kishin një ambicie më të madhe, duke kundershtuar kostot e larta të transfertave dhe tarifat që vendoseshin nga bankat, duke veshtiresuar kështu, transaksionet internacionale [16]. Sipas Back: Čypherpunks kishin në mend, njesi shkembimi që siguronin anonimitet të plote, me kosto të ulet transaksi i dëtihet dhe të transferushme". Por praktikisht, sistemet të cilet po ndertohen e kishin gjithmonë

---

<sup>3</sup>Kopimi i file-eve në boten dixhitale, është analogu i printimit të parave në monedhat fiat

të pamundur heqjen e institucionit qendror. Akoma me tej, eksperimentet po vuanin nga një veshtirësi edhe me fundamente, si do të bindeshin njerëzit pe të pranuar këto monedha dixhitale? Adam Back ishte personi i parë më të cilin Satoshi komunikoje, në 2008 menjehere pasi publikoje paper-in e Bitcoin. Për Satoshin ajo cka kishte rendesi ishte ideja dhe jo personi. Satoshi bashkoi të gjithë këto inovacione të hershme për të krijuar sistemin me unik nga të gjithë atë të meparshmit. Në vend që të bazohej në një banke qëndrore apo kompani për të krijuar para - si DigiCash i Chaum - ky sistem ishte i tille që cdo transaksi me Bitcoin, dhe sasia që cdo person zoteronte, do të gjurmoheshin nga cdo nyje në rrjet, në një databaze që si rrjedhojë mirembahej nga të gjithë pjestarët e sistemit, dhe që me vone u quajt Blockchain. Sipas paper-it revolucionar, cdo user do kishte një ose me shumë cifte celesash publik-privat, që sherbente ekuivalentin e një llogarie bankare. Monedhat e shoqeruara me një adresë mund të shpenzoheshin vetëm nga personi me celesin privat. Natyrisht që celësi private ruhej nga individi, dhe jo nga një autoritet qendror. Njehere që Alice nenshkruante një transaksion me celesin e saj privat, transaksioni behet broadcast në gjithë nyjet e rrjetit. Keto kompjutera kontrrollonin nëse Alice, kishte aq monedha sa donte të shpenzonate. Kjo ishte e mundur duke qenëse përdoruesit mbatin Blockchain-in ku ruhej historiku i cdo transaksi. Pjesa më e nderlikuar e zgjidhjes ishte se si do të shtoheshin bloqet në Blockchain. Satoshi u kthye nga ekonomia, duke u mbeshtetur në modelimin e mekanizmave (incentives), në një nivel shumë të lartë, zgjidhja e Satoshit konsistonte në një gare kompjuterike mes nyjeve në rrjet, e modeluar sipas "proof-of-work" të Adam Back. Kompjuteri që fitonte garen, ishte perjegjes për të shtuar bllokun e tij me transaksione në Blockchain. Në të njejtën kohe ai shperblehej për punen e tij me 50 BTC (në fillimet e Bitcoinit). Ky shperblim në Bitcoin-e, ndihmoi përdoruesit e Bitcoin-it, për të marrë pjesë në aktivitetin e gjurmuarit të transaksioneve. Nëse kishte keqkuptime në lidhje mbi kompjuterin që gjeti i pari zgjidhjen, fitonte ai bllok mbi të cilin ndertonte blloqet pasardhese pjesa më e madhe e rrjetit (majority rule). Transaksionet në bllokun humbes do të beheshin discard dhe do futeshin me vone, përmes ndonje blloku tjetër në Blockchain. Për ta ilustruar një transaksion japim 5 hapat e meposhtem:

- Alice krijon një mesazh ku deklaron se do t'i dergoje Bob-it Bitcoin-e. Mesazhin e nënshkruan me celesin privat, dhe i ben broadcast në gjithë rrjetin.
- Perdoruesit e tjere sigurohen se Alice ka mjaftueshem Bitcoin-e për të kryer

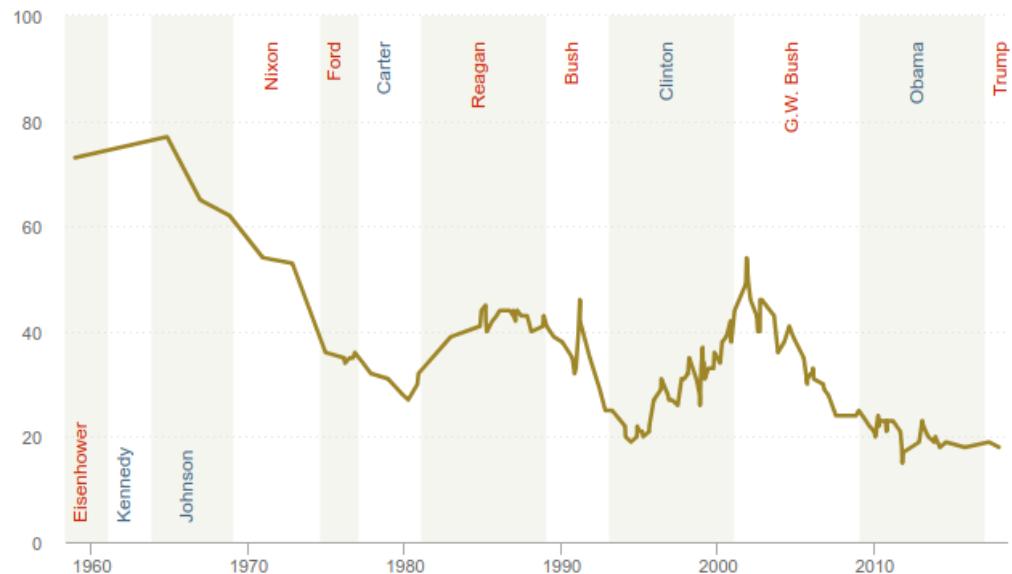
transaksionin. Nëse po, krijojne një bllok dhe e shtojne aty.

- Miner-at e sistemit, fillojne garen për të zgjidhur POW. Kush e zgjidh i pari, krijon Bitcoin-e të reja dhe shpallet si zoterues i tyre.
- Perfundimi i kësaj gare, do të thotë që gara e rradhës ka nisur, dhe miner-at fillojne procesin nga fillimi

Rezultati i këtij procesi (që do e shpjegojme me vone me detaje), është një sistem finansiar që krijon dhe leviz para pa autoritet qendror. Sistemi është i tille, që monedhat mund të shpenzohen vetëm nga ai që ka celesin privat, që i korrespondon një adresë në Bitcoin. Pra cdo përdorues i sistemit është konfident se në cdo moment, ekziston një dhe vetëm një rekord i pandryshueshem që tregon monedhat që cdo përdorues zoteron. Për ta besuar këtë, përdoruesit nuk kishin pse t'i besonin Satoshi, sic ndodhë tek skema e David Chaum, apo sic ndodh me bankat, ata duhet t'i besonin thjeshtë nyjes se tyre, kur ekzekutohej klini Bitcoin-it. Prandaj dhe sistemi quhet *trustless*, pasi t'i verteton për vetë, nuk ke nevojë për ndermjetes që kryejne kontrollle mbi vlefshmerine e transaksioneve. Nëse përdoruesit nuk pëlqenin rregullat e sistemit, ata mund t'i ndryshonin rregullat, pra në njefare mënyrë, përdoruesit janë edhe autoriteti që verifikon edhe klientet që përdorin sistemin për transaksione. Një javë pas publikimit të paper-it, vetëm 2 persona kishin dhene feedback. I pari ishte patjeter Hal Finney dhe i dyti John Levine, një ekspert në sigurine kompjuterike. Ai ishte i mendimit se sistemi Blockchain-i i Bitcoin do të vuante nga sulme sigurie, dhe kjo do conte në krijimin e versioneve të ndryshme të kësaj databaze. Argumenti i Levine ishte se hackers, apo persona të tjere keqdashës kanë gjithmonë me shumë fuqi kompjuterike, kundrejt persona me qellime pozitive. Patjeter që kishte të drejte, duke qenëse siguria arrihej nga konsensusi që arrinte maxhoriteti i sistemit (majority rule). Në fillimet e Bitcoin, ku kishte fare pak kompjutera, ishte e thjeshtë për dike "të behj maxhoritet", duke qenëse konkurenca nuk ishte shumë e madhe, dhe Bitcoin-et beheshin mine edhe me një CPU që zoteron shumica sot, si rrjedhojë, mjaftonte për 1 miner me GPU që zoteronte shumicën e fuqise kompjuterike të behj maxhoritet dhe të merrte nën zoterim 51% të Blockchain-it. Satoshi, dhe kur e modeloj sistemin, ishte me shpresen se askush nuk do kishte incentive për të kryer një sulm packa lehtesise në të. Nëse do kishte incentive për të sulmuar sistemin, kjo duhet të ishte për shkak të veshtiresise për të bërë mine (e cila vjen nga rritja e kerkeses). Pasi Satoshi publikoi

kodin, në maj 2009, Martti Malmi, një student informatike, i cili kishte njoburi në C dhe Java, doli vullnetar për të ndihmuar Satoshi me kodin. Falë entuziasmit dhe ideve të ngjashme me Satoshi, Martti filloi të programonte në Bitcoin Core.

Një tjetër problem që nuk diskutohet nga Cypherpunks, por Satoshi e prek, është zhvlerësimi i monedhiës, si në sistemin e arit, ashtu dhe në atë të monedhave *fiat*. Kriza e viteve 2007-2008, e kishte treguar hapur mos efektivitetin e sistemit të centralizuar. Lëvizja nga standarti i arit, në fiat, dhe pse konsiderohet si zgjedhja e duhur për simulimin e ekonomisë gjatë Depresionit të Madh, përbën një ndër faktorët kryesore, për krizen e permendur me lartë. Ky lloj keq-menaxhimi i parasë, beri që shumica ta shihnin bankën qëndrore, nën qeverisjen e Nixon, si të padenjë në detyrën e tyre (Fig. 2.3).



**Figura 2.3:** Studim nga Pew Research Center. Besimi i publikut në qeverinë e Sh.B.A, nën udhëheqjen e presidentëve të ndryshëm [17]

Një tjetër personazh, influencues është Laszlo Hanecz, pasi ky ishte personi që mund të kryente një double spend duke patur në dispozicion 51% të fuqise se hash-it në rrjet. Lazlo ishte një Software Architect, i cili kishte degjuar mbi Bitcoin në Internet Relay Chat<sup>4</sup>. Laszlo pervecse kontribuoi në versionin MacOS për Bitcoin-in,

<sup>4</sup>Internet Relay Chat (IRC) është një chat për komunikim në gjithë boten. Konsiston në bashkësi

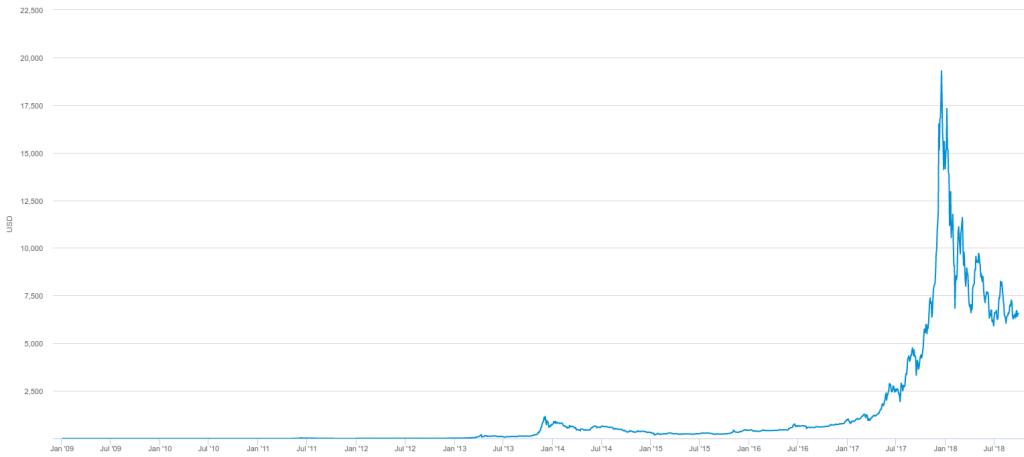
ai ishte edhe personi i parë që beri një shkembim malli me Bitcoin, duke blerë 2 pizza për 10.000 BTC. Arsyet tjetër pse është i rëndësishëm, është se duke marrë parasysh profesionin e tij, Laszlo ishte i interesuar në bug-e të mundëshme në sistem. Kur kuptoi se si mining funksionon, ai vendosi të përdorë GPU për të krijuar Bitcoin-e të reja. Kjo nuk u prit mirë nga Satoshi, pasi Laszlo fare mirë mund të kryente një sulm 51%. Fatmiresisht, Laszlo e pa se nuk kishte ndonje fitim me këtë veprim, dhe si rrjedhojë nuk pati sulm.

Histori të tjera që duhet të permenden në lidhje me Bitocin janë sulmet e shumta që ka pësuar. Në 2013 FBI konfiskon 171.955 BTC (\$270.000.000 në atë kohe), për aktivitetet narkotike të Silk Road. Pak me vone, në 2014, exchange-i i famshëm MtGox, i ngritur nga një person i vetëm Jed McCaleb, i cili me vone do të ishte co-founder i Ripple, se bashku me Chris Larsen, sulmohet dhe vidhen 850.000 BTC. MtGox, nuk mund të ruante dot celesat privat që ruheshin neper wallets të klienteve. Në korrik të po të njejtit vit 300.000 BTC vidhen nga exchange-i Cryptsy. Më pas në dhjetor, vidhen 3894 BTC në exchange-in Mintpal. Në Janar të 2015 19.000 BTC vidhen nga exchange-i Bitstamp, ku një nga punonjesit nuk ndoqi rregullin për të mos hapur file nga persona të huaj. Në Gusht të 2016, 120.000 BTC vidhen nga exchange-i i famshëm Bitfinex. të gjitha këto në fakt, mund të korelohen dhe me ngritjet e larta në grafikun e cmimit mesatar të Bitcoin.

## 2.3 Mekanizmat e Bitcoin

Në këtë seksion flitet kriptografine, dhe mekanizmat që bëjnë të mundur Bitcoin-in, si funksionojne dhe si nderveprojne se bashku.

të gjitha monedhat kanë nevojë për një mënyrë se si të kontrollojne prodhimin e monedhës dhe si ta mbajne atë të sigurte ndaj mashtrimeve. Ndryshe nga sistemet fiat, ku banka apo qeveria vendosin rregulla mbi këto të mesipermet, në boten e Kriptoekonomise, është teknologjia ajo përmes se ciles vendosen rregullat e sigurisë. Duhet theksuar gjithashtu, se sistemet e Cryptocurrency-ve, nuk janë njëlloj me E-banking. Ndersa në E-banking ekzistojnë certifikatat që bëjnë të mundur identifikimin e individit, kjo nuk është e vertete në boten e kriptove, duke qenë se, sic do shohim dhe më poshtë ato, sistemi është pseudoanonim, dhe celesat publik janë ekuivalent rrjetash serverash IRC të cilet lejojne lidhjen me rrjetin. Ndryshon nga forma e komunikimit me mesazhe, pasi qellimi është komunikimi në grup.



**Figura 2.4:** Cmimi historik i Bitcoin

me identitetin.

Cryptocurrency-të, sic kuptohet dhe nga emri, përdorin gjerësisht kriptografine. Ajo ofron mekanizmat për të siguruar rregullat e sistemit të kriptove brenda sistemit, pa patur nevojë për një autoritet të trete. Në Bitcoin përdoret për të parandaluar modifikimet e parasë si, për të siguruar përdoruesit e saj, si dhe për të ndertuar rregullat e krijimit të njesive të reja përmes një protokoli matematikor. Kriptografia është një fushe shumë e thellë kerkimore, por ndryshe nga cka beson një person në përgjithësi, Bitcoin-i përdor një kriptografi shumë bazike, maksimumi shkon tek Elliptic Curve Cryptography, natyrisht këto primitiva me shumë mundesi do të ndryshojne se shpejti.

### 2.3.1 Funksionet Hash

Primitiva e parë dhe më e përdorura në Bitcoin janë funksionet hash. Një funksion hash gëzon vjetitë e mëposhtme:

- Inputi është një varg karakteresh me gjatesi cfaredo
- Outputi është fiks. Në Bitcoin është 256-bite, duke qenë se përdoret SHA-256.
- Llogaritja e hash-it është eficent, pra sasia e kohes që hash-it i duhet për të nxjerre outputin është e arsyeshme. Specifisht, llogaritja e një hash-i (*digest*)<sup>5</sup>,

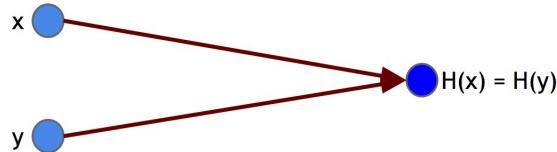
---

<sup>5</sup>Digest-i është outputi i funksionit hash, kështu që i perdorim të dy termat në vazhdim

prej  $n$ -bitesh ka kompleksitet  $\mathcal{O}(n)$ .

Vecorite me lartë, specifikojne funksionet e per gjithshme hash, për të shkuar një nivel me lartë, në ato funksione që janë të sigurte nga ana kriptografike janë edhe tre vecori të tjera:

- **Rezistent ndaj përplasjeve** (*Collision resistant*) Një perplasje ndodh vetëm në rastet kur 2 inpute të ndryshme kanë të njejin output (Fig. 2.5). Pra, nuk është *praktikisht* e mundur të gjejme  $x, y$  e tille që,  $x \neq y$ . Arsyja se pse na duhen funksionet hash që nuk kanë perplasje është se: nëse dimë inputin  $x$  dhe  $y$  për një funksion hash  $H$  janë të ndryshem, atëherë mund të implikojme se edhe output-et e hash-it janë të ndryshme. Kjo vecori, na lejon të perdorim outputin e hashit si *message digest*. Packa se gjetja e një perplasje të tille nuk është praktike, fakti se këto perplasje nuk ekzistojnë nuk është e vertetuar, në fakt ajo për cka jemi të sigurte është e kunderta, në funksionet hash gjithmonë ka perplasje (*pigeonhole principle*[18]), duke marrë parasysh se hapësira e inputit është e pafundme ndër kohë që hapësira e outputit është e fundme, meqë gjatësia e tij është fiksë, si rrjedhojë ka input-e që lidhen më të njejin output. Problemi, sic ndodh shpesh në kriptografi, është se koha e llogaritjes për të gjetur një perplasje, është shumë e madhe, dhe si rrjedhojë jo praktike. Ajo për të cilën jemi të sigurte, është se për asnjë funksion hash nuk është vertetuar të jene pa perplasje.



**Figura 2.5:** Perplasje mes 2 inputeve në të njejin digest.

- **Hiding** Logjikisht, duhet që hash-i ta "fshehe" hapesiren e inputeve të mundshem. Kur na jepet hash-i  $y = H(x)$  një sulmues duhet të gjeneroje hashin e cdo  $x$ 'në hapesiren e inputeve (*brute force*) në mënyrë që të gjeje  $x$  që gjeneroje  $y$ . Kjo e ben të pamundur parashikimin mbi input-in. Pershembull, nëse informacioni që po hashojme është "Alice dergon 1BTC tek Bob", apo "Bob

dergon 1BTC tek Eve", atëherë mund të gjendet lehte se hapësira e inputit ka formen: "Perdoruesi1 dergon BTC tek perdoruesi2". Kjo është situata që duam të evitojme, dhe e arrijme këtë përmes zgjedhjes se një stringu  $r$  me gjatesi fikse prej 256 bitesh nga një shpërndarje probabilitare që është e rrallë (me entropi-minimale të lartë). Në Bitcoin, numri  $r$  është *nonce* i famshëm. Pra min-entropy, është metrika që mat sa i parashikueshem është një output, nga ana tjetër, high min-entropy, sugjeron që shpërndarja të jete e rrallë.

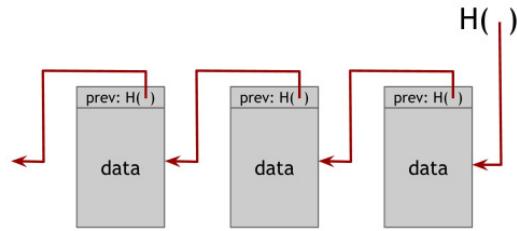
- **Puzzle Friendliness** Themi se një funksion hash është puzzle-friendly, nëse për cdo output të mundshem  $y$  prej  $n$ -bitesh, nëse zgjedhim k nga një shpërndarje me entropi-minimale të lartë, atëherë gjetja e një  $x$  e tille që  $H(k \parallel x) = y$  nuk është praktike dhe koha e kerkuar është minimalisht  $2^n$ . Pra, është e pamundur që dikush të modifikoje funksionin hash, me qellim nxjerrjen e një vlere të caktuar  $y$ , duke patur nën kontrroll vetëm inputin  $x$ . Si rrjedhojë, mënyra më e mirë për të gjetur  $x$  është *brute force* Pra nëse një problem gëzon vetinë e të qenit puzzle-friendly, nuk ka strategji më të mirë se sa të provojme në mënyrë random vlera të ndryshme të inputit. (Kjo vecori nuk është e nevojshme në cdo funksion hash kriptografik, në Bitcoin, duhet për procesin e mining)

**Hash pointers** Një hash pointer, nga vetë emri, është një struktura të dhenash që tregon se ku ruhet një informacion se bashku me një hash kriptografik të atij informacioni. Ndërkohë që një pointer normal jep një mënyrë për të bërë retrieve informacionit, një hash pointer jep edhe një mënyrë për të vertetuar se ai informacion nuk ka ndryshuar.



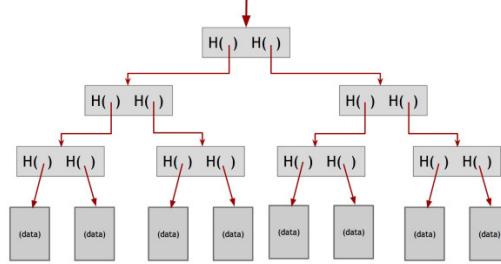
**Figura 2.6:** Struktura e një Hash pointer

**Blockchain** Blockchain, perseri nga vetë emri, është një liste e lidhur mbërapsht, ku të dhënrat janë të hashuara. Një use-case për njew block chain, është një tamper-evident log. Pra një strukture të dhenash ku ruajme informacion, dhe na lejon të shtojmë të dhëna në fund të log-ut. Por nëse dikush kerkon të modifikoje të dhënrat e mëparshme në log, sistemi e detekton. Arsyja pse funksionon, duket edhe nga figura, është se ndryshimi i të dhënave në bllokun e mesit  $k$ , si rrjetohojë hash-i i këtij bllokut do të ndryshoje, meqë bllokut  $k+1$  por ky bllok lidhej me hashin që ndodhej më parë, dhe jo me këtë të ndryshuarin, pra një sulm ka ndodhur. Pra, kjo inkonsistence, duket mes bllokut  $k$  dhe  $k+1$ . Sulmuesi mund të vazhdoje ta beje këtë gjëderi në bllokun e parë në blockchain (në Bitcoin quhet Genesis block). Pra për një sulmues që mundohet të ndryshoje të dhëna në ndonje pjesë të blockchain-it, hash pointer-i në bllokun e ardhshem do të jete i pasakte.



**Figura 2.7:** Struktura e një Blockchain

**Merkle Trees** Një tjetër strukture të dhenash shumë e rëndësishme, janë pemet binare më të dhëna të hash-uara. Një peme binare me hash pointers quhet **Merkle tree**, në ndër të atij që e zbuloi: Ralph Merkle. Ketu, blloqe të dhenash grupohen në cifte dhe hash-i i këtyre cifteve ruhet nga nyja prind. Kjo vazhdon deri sa të arrihet rrenja. Perseri, e njejta gjësi në blockchain aplikohet pra, është tamper-evident.



**Figura 2.8:** Struktura e një Merkle Tree

Pemet e Merkle-it, përdoren për të ruajtur transaksionet brenda një blloku, pasi është më eficiente të ruhet vetëm vlera e hash-it që del në fund të pemes, se sa të gjitha transaksionet.

### 2.3.2 Nënshkrimet dixhitale

Primitiva e dyte kriptografike më e përdorshme në Cryptocurrency, janë nënshkrimet dixhitale (*digital signature*). Nënshkrimet dixhitale duhet të ofrojnë dy vecori:

- Nuk mund të ekzistojnë dy persona më të njejtin nënshkrim dixhital, dhe cdo kush mund të verifikoje që nenshkrimi është i vlefshëm.
- Nënshkrimi është i lidhur me një dokument, në mënyrë që të mos përdoret për të implikuar pronësinë e ndonje dokumenti tjeter. Pra nuk është si celësi publik që cdo kush e sheh dhe është i lire ta përdorë.

**API e nënshkrimeve dixhitale** Një skemë nenshkrimesh dixhitale perbehet nga funksionet e mëposhtme:

- **(sk, pk) := gjeneroCelësa(gjatësiaCelësit)** Kjo metode merr si input një numër që i korrespondon gjatesise se celesit për të gjeneruar ciftin e celesit publik (pk) dhe atij privat (sk). Celësi publik verifikon, ai privati nënshkruan.
- **sig := nënshkruaj(sk, msg)** Funksioni i nënshkrimit, i cili ka si output nenshkrimin për një mesazh.

- **eshteIVlefshem := verifiko(pk, msg, sig)** Metoda e verifikimit merr një mesazh, një nënshkrim, dhe një celës publik si input. Nëse celësi publik i perket celesit privat më të cilin është nenshkruar, output-i është 1, në të kundërt 0.

Me tej, 2 vecori duhet të ekzistojnë:

- Nënshkrimet e vlefshme duhet të verifikohen **verifiko(pk, msg, sign(sk, msg))**  
== 1
- Nënshkrimet nuk mund të fallsifikohen

Bitcoini përdor skemen **ECDSA** (Elliptic Curve Digital Signature Algorithm), vija eliptike e përdorur është "secp256k1", e cila ka të njejtin nivel veshtirësie për t'u thyer sikur të llogarisësh  $2^{128}$  veprime kriptografie me 1 celës, si psh. ekzekutimi i një funksioni hash. Kjo vijë eliptike përdoret rrallë jashtë Bitcoin, dhe tanë është veshtire të ndryshohet duke qenë se është përfshirë nga Satoshi herët në specifikimin e sistemit. Disa vecori që rrjedhin në lidhjen me gjatësinë e celësave janë si mëposhtë:

- Celësi privat; 256 bite
- Celësi publik i pa kompresuar: 512 bite
- Celësi publik i kompresuar: 257 bite
- Mesazhi që do nenshkruhet: 256 bite
- Nënshkrimi: 512 bite

### 2.3.3 Çelësat publik kundrejt identiteteve

Nga skema e nënshkrimit merret celësi publik dhe ai perfaqson identitetin që po kryen transaksionin, dhe kjo është një ndër arsyet pse në Bitcoin nuk ka nevojë për enkriptim, apo certifikata, pasi nuk ka identitete të tipit: një person me emer dhe mbiemer, në fakt cdo entitet mund të kryej transaksion në Bitcoin. Si rrjedhojë, çelësat publik janë aktore në sistem. Në mënyrë që dikush të kryej një transaksion me këtë celës publik (identitet), ai duhet të zoteroje çelësin privat korrespondues. Një pasojë e perdomit të celësave publik si identitet është se mund të krijohen në cdo lloj sasie, ndryshe nga identiteti i një personi në jetë reale, ku ai njihet nga një identitet i vetëm, në Bitcoin, ekzekuton funksionin **gjeneroCelësa** dhe ja tek krijove

një identitet të ri. Në praktike, përdoret hash-i i  $pk$ , meqë na jep celës më të vogel në madhesi. Kjo sjell që, për të verifikuar se një mesazh vjen nga identiteti me një celës publik  $pk$ , duhet të llogaritet hash-i i  $pk$ .

**Menaxhimi i decentralizuar i identiteteve** Krijimi i identiteteve të reja nuk lidhet me ndonje server qendror, apo njesi që leshon celësa publik. Por, protokolli është i tille që njehere që behesh pjesë e rrjetit, lind e drejta për të krijuar celësa publik në numër të pacaktuar. Shumë here në terminologjine e cryptocurrencie-ve permendet fjala **adrese**, dhe kjo nuk është asgje tjetër vecse hash-i i celesit publik. Meqë është kaq e lehte të krijohen celësa publik dhe dy persona mund t'u gjenerohen të njejtat celësa publik, shpesh skema e Bitcoin-it duket pak e "frikshme", por probabiliteti që kjo të ndodhe është shumë shumë i vogel, dhe praktikisht nuk merret parasysh si rrezik. Kjo vjen nga fakti që celesat gjenerohen nga një burim që ofron jo determinizem (rastesi). Rrezikshmeri do të kishte vetëm nëse burimi do ishte determinist.

### 2.3.4 Sistemet e Decentralizuar

E-mail (SMTP) dhe interneti janë disa nga sistemet e decentralizuar, nga ana tjetër rrjetet sociale si Facebook dhe LinkedIn janë të centralizuar. Decentralizimi nuk është i prerë me thikë nga centralizimi, psh. në rastin e Bitcoin, edhe pse protokolli i krijuar nga Satoshi ofron skemë perfekte decentralizimi, shërbimet e exchange-ve, software-et e wallets, si dhe mining pools janë të centralizuara. Gjithësesi mënyra Bitcoin-i në vetevete është i decentralizuar dhe kjo arrihet përmes menyres se si:

- Mirembahet blockchain-i
- Identitetet kanë autoritet mbi transaksionet që janë të vlefshme
- Krijohen Bitcoin-e të reja
- Merren vendime mbi ndryshimin e rregullave të sistemit
- Bitcoin-et marrin vlere në exchange

Rrjeti P2P është i decentralizuar pasi cdo kush mund të ekzekutoje një nyje Bitcoin-i, duke bërë download një klient Bitcoin-i. Mining është pjesa me shqetesuese, për

shkak t  cenrtalizimit t  lart  n  shtete si Kina apo Japonia. Nj  tjet r aspekt, jan  updates t  software-it t  Bitcoin q  kan  nyjet. Meq  sistemi  sht  i decentralizuar dhe nuk ka nj  server q  t  beje forced-updates, n  Bitcoin duhet q  cdo nyje t  marr  patch-et dhe upgrade-et m  t  fundit n  m n yr  t  pavarur, ndonese shumica kan  implementimin q  referohet nga bitcoin.org, ekziston nj  num r i lart  i nyjeve q  kan  implementime t  ndryshme, dhe zhvilluesit e software-it t  Bitcoin kan  nj  fuqi shum  t  madhe mbi komunitetin.

### 2.3.5 Konsensusi i shp rndar 

Termi **konsensus**  sht  kyc n  funksionimin e Bitcoin, n  vecanti, **konsensusi i shperndare**. Problemi kryesor q  duhet t  zgjidhet n  nj  sistem t  shperndar e-cash,  sht  si t  arrihet n  konsensus t  shperndare.

Fokusi kryesor i sistemeve t  shperndare  sht  arritja e qendrueshmerise (*reliability*)<sup>6</sup>, pra nyje t  ndryshme q  zot rojn  t  nejtin informacion, t  jene t  sinkronizuara. Edhe pse Google-i, Facebook etc. zot rojn  arkitektura backend-i prej shum  serverash dhe si rrjedhoj  t  gjith  nyjet duhet t  jene n  t  nejten gjendje, konsensusi i shperndare  sht  dicka tjet r dhe shum  m  i veshtire se kaq. Nj  perfkufizim mbi nj  protokoll konsensusi t  shperndare do ishte si m  posht : Ekzistojne  $n$  nyje e t  tille q  secila ka nj  vlere inputi. Disa prej k tyre nyjeve zoterohen nga sulmues. At her  nj  protokolli i konsensusit t  shperndare do k t  k to dy vecori:

- t  gjith  nyjet duhet t  bien dakort mbi nj  vlere n  fund t  komunikimit
- Vlera duhet t  jete gjeneruar nga nj  nyje e ndershme

Rrjeti i Bitcoin  sht  P2P dhe si rrjedhoj  i shperndare dhe i decentralizuar. Kur Alice deshiron t  paguaj Bob-in, ajo ben broadcast transaksionin n  t  gjith  rrjetin dhe jo vet m n  adresen e Bob-it. Ajo mbi t  cilin nyjet duhet t  arrijne n  konsensus  sht : cilat transaksione u bene broadcast dhe n  cilin rend ndodhen transaksionet. Transaksionet p r arsy  eficence n  komunikim grumbullohen n  nj  *pool* trnnsaksionesh mbi t  cilat akoma nuk  sht  arritur konsensus global. Nyjet mund t  kene transaksione t  ndryshme n  *pool* duke marr  parasysh q  rrjeti P2P nuk  sht  perfekt, dhe disa nyje mund t  kene degjuar p r transaksione t  caktuara, nd rkoh 

<sup>6</sup>Reliability i referohet sakt sis  se t  dh nave, ku sistemi  sht  fault tolerant jo vet m ndaj t  dh nave t  korruptuara, por edhe ndaj nyjeve t  korruptuara

nyjet e tjera kanë degjuar mbi transaksione të tjera. Menyra si arrihet konsensusi është kjo: në intervale prej 10 minutash, cdo nyje në siste, propozon *pool*-in e saj të transaksioneve për të qenë blloku i ardhshem, me pas ekzekutohen rregulla të protokollit, ku input-i i cdo nyjeje është blloku i propozuar. Disa nyjeje mund të mos jene të ndershme dhe vendosin transaksione të pavlefshme në bllok, por në supozojme se nyjet e tjera janë të ndershme. Nëse konsensusi ka sukses atëherë, një bllok i vlefshëm do të zgjidhet si output. Edhe pse blloku propozohet vetëm nga një nyje, është output i vlefshëm për aq kohe sa blloku është i vlefshëm. Bitcoin-i përdor një zgjidhje të ngjashme, por jo fiks. Fillimisht, arritja e konsensusit është një problem i veshtire pasi nyjet mund të mos jene të ndershme ose të behen faulty në momentin që ndodh validimi. Se dyti, specifisht në Bitcoin, rrjeti është shumë larg atij perfekti. Pasi nuk është një graf plotesishte i lidhur, pra jo cdo nyje lidhet me cdo nyje. Qofte edhe lidhja e dobet e internetit<sup>7</sup> paraqet problem, dhe si rrjedhojë ekzekutimi i protokollit të konsensusit në një rrjet ku të gjithë nyjet janë aktive nuk është e mundur. Së fundmi, duke qenë se është i shperndare në të gjithë internetin rrjeti ka shumë latency<sup>8</sup>, duke konsideruar faktin se ora nuk është 100% e sinkronizuar në sisteme të shperndare. Pra, jo të gjithë nyjet mund të bien dakort mbi një renditje globale të eventeve thjeshtë duke parë timestamps (pasi nuk ka një server që të vendos mbi një kohe të perbashket!). Problemi klasik që na kujton kjo skemë është ai i Gjeneralit Bizantin, ku ushtria Bizantine është e ndare në divizione, e tille që secili udhehiqet nga një gjeneral. Gjeneralet komunikojne përmes mesazheve nën mënyrë që të kene një plan të perbashket veprimi. Por, disa gjeneral mund të mos jene të ndershem duke u munduar kështu të terhiqen nga sulmi në mënyrë që gjeneralet e ndershem të mos arrijne në një plan të unifikuar. Qellimi i problemit është që të gjithë gjeneralet e ndershem të arrijne në të njejtin plan pa lejuar gjeneralet e pa ndershem t'i prishin mendjen. Kjo ka zgjidhje vetëm kur  $2/3$  e gjeneraleve janë të ndershem [19] [20]. Por këto probleme në sistemet e shperndare, dalin në pah kur konsiderojmë databazat e shpendara, Bitcoin-i edhe pse ka një Blockchain që sherben si databaze është pak me ndryshe meqë është sistem monetar. Prandaj themi se ai funksionon me mirë në praktike se sa në teori, pasi nuk eskizton një teoreme e cila verteton pse konsensusi në Bitcoin funksionon. E vecanta e Bitcoin-it që con në zgjidhjen e problemit të kon-

---

<sup>7</sup>Lidhja e internetit është e nevojshme për regjistrimin e transaksionit në blockchain, ndërkokë që nenhkrimi i tij mund të ndodhe dhe offline

<sup>8</sup>Psh. 2 minera bëjnë append bllokun e tyre në blockchain në të njejten kohe

sensusit është se Satoshi perfshin një *incentive*, dhe kjo është një risi në protokollet e sistemeve të shperndare. Tjeter risi është ajo cka kemi thene dhe me lartë, që një nyje mund të këtë shumë identitetet meqë nuk ka një autoritet qëndrore që të leshe token-a identifikimi. Termi teknik për këtë është *textbf{Sybil attack}*. Sybils, janë thjeshtë kopjet që krijon një nyje e pandershme për të tentuar të duket sikur paraqet pjestare të ndryshem në rrjet, kur në fakt është vetëm 1. Arsyja tjetër pse nyjet mund të krijojnë shumë identitetë është pasi by-design, Satoshi deshironte që Bitcoin-i të ishte pseudo-anonim. Nëse nyjet do të kishin identitet kjo do e bente implementimin e sistemet më të thjeshtë. Fakti që një nyje mund të gjeneroj identitetet pa kufi, nuk na lejon të llogarisim sa nyje të pandershme ka në rrjet.

**Konsensusi implicit** Në protokoll ka disa round-e, ku secili i korrespondon një blloku të ndryshem në blockchain. Në fund të cdo round-i zgjidhet një nyje (supozojme se zgjidhet në mënyrë të rastesishme) e cila propozon bllokun e ardhshem në blockchain. Po nëse nyja është e pandershme? Ka një proces që e zgjidh këtë: nyjet e tjera në mënyrë implicite e pranojnë ose jo bllokun duke vendosur nëse do të vendosin bllokun e rradhës mbi këtë bllok. Nëse duan ta injorojne bllokun, thjeshtë nuk vendosin bllokun e rradhës mbi të. cdo bllok në strukturen e të dhenes se tij, ka edhe hash-in e bllokut paraardhes. Dhe ky është mekanizmi teknik që lejon nyjet të sinjalizojne se cilin bllok po zgjerojne. Algoritmi do dukej pak a shumë si mëposhtë:

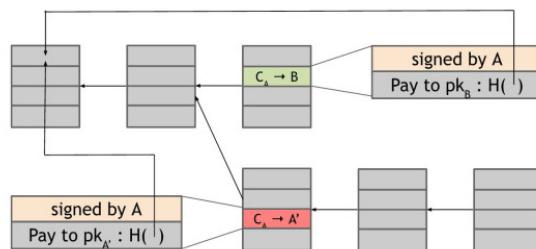
1. Transaksionet e reja behen broadcast në të gjithë rrjetin
2. Cdo nyje grumbullon transaksionet e reja në një bllok
3. Në cdo round një nyje \_ e rastesishme ben broadcast bllokun e saj
4. Nyjet e tjera pranojnë bllokun vetëm nëse të gjitha transaksionet në të janë të vlefshme (shuma e inputit është më e madhe se ajo e output-it, nenshkrimi është i vlefshëm, shuma nuk është e shpenzuar më parë)
5. Nyjet shprehin dakortesine e tyre me këtë bllok duke ndertuar blloqet pasardhese mbi të (perfshijne hash-in e këtij blloku në bllokun e ardhshem që krijojnë)

Disa nga sulmet që mund të konsiderojmë për të parë se si skema e përshkruar me lartë funksionon janë:

**Vjedhja e Bitcoin-eve** A mundet Alice të vjedhe Bitcoin-et e Bob-it pa ditur celesin e tij private? Jo, pasi për këtë do duhej që Alice të krijonte një transaksion të vlefshëm që tenton të shpenzoje atë monedhë, por që të krijosh një transaksion duhet të nenshkruash shumën, dhe që të nenshkruash duhet celësi privat

**Denial of Service** A mundet Alice që të refuzojë transaksionet e Bob-it? Po, por transaksiioni behet broadcast në të gjitha nyjet, dhe me shumë mundesi do të dale një nyje që do ta përfshijë transaksionin e Bob-it në një bllok.

**Shpenzimi i dyfishte** Supozojme se Alice krijon një transaksion për tek Bob-i, dhe e broadcast në rrjet. Supozojme me tej se transaksiioni pranohet në bllokun e një nyjeje. Kështu u krijuar një bllok nga një nyje e ndershme që permbar transaksionin e Alice-Bob. Një transaksion në Bitcoin është thjeshtë një strukture të dhenash që permbar nenshkrimin e Alice, një instruksion për të paguar tek celësi publik i Bob-it, dhe një hash. Ky hash paraqet një pointer<sup>9</sup> për tek transaksiioni i outputit të mëparshëm të Alice. Ky pointer duhet të referojë një transaksion që përfshihej në një bllok të mëparshëm në zinxhirin e konsensusit. Supozojme se pas bllokut që permbar transaksionin e Alice, blloku i rradhës që vendoset është një bllok i formuar nga ajo. Meqë Alice po propozon bllokun e ardhshem, ajo mund të propozooje një bllok që injoron bllokun që permbar pagesën e kryer tek bob dhe e vendos pointer-in tek bllok para atij ku ndodhet transaksiioni me Bob-in.



**Figura 2.9:** Skema klasike e shpenzimit të dyfishte në blockchain

---

<sup>9</sup>Pointer-i i transaksiionit është i ndryshem nga ai i bllokut.

Billoket perfshijne një hash pointer për tek blloku që është vendosur përpëra tyre, transaksiyonet nga ana tjetër, perfshijne një ose me shumë hash pointers për tek output-et e transaksioneve që po shpenzohen

Për me tepër, në bllokun që propozon, Alice perfshin edhe një transaksion që transferon monedhat që ajo po i dergonte Bob-it tek një tjetër adresë që zoteron. Meqë të dy transaksionet shpenzojne të njejtat monedha, vetëm njera prej tyre mund të perfshihet në blockchain. Nëse Alice ka sukses për të përfshirë pagesën në adresen e saj në blockchain, atëherë transaksioni në të cilin ajo i paguan Bob-it është i pavlefshëm meqë nuk mund të perfshihet me vone në blockchain. Cështja është si të kuptojmë nëse një shpenzim i dyfishte do të ketë sukses apo jo. Kjo varet se nga cili bllok do të perfundoj në chain-in më të gjatë të konsensusit - transaksioni i Alice -> Bob apo transaksioni i Alice -> Alice'. Nyjet e ndershme ndjekin rregullin e zgjerimit të zinxhirit më të gjatë, si rrjedhojë nuk ka përgjigje të saktë, pasi në zinxhirin më të gjatë mund të ndodhet transaksioni me shpenzim të dyfishte ose jo, kjo nuk i perket nyjeve të tjera që vijne me pas. Nga pikepamja e moralit, natyrisht që kuptohet se transaksioni Alice -> Bob duhet të perfshihet, por duke u bazuar në historine e blockchain-it nuk ka një diferenca mes këtyre transaksioneve. Nyjet që shohin blockchain-in nga jashte nuk kanë si të determinojne kush nyje është me morale përt'u zgjeruar.

Në praktike, nyjet në përgjithësi ndjekin një heuristik për të zgjeruar një bllok për të cilin kanë degjuar të parin në rrjetin P2P. Por kjo nuk është 100% të hereve e vertete. Dhe në cdo rast, për shkak të vonesave në rrjet, mund të ndodh që blloku për të cilin nyja degjon fillimisht është blloku që është krijuar i dyti. Pra ekziston një mundesi që nyja të zgjedhe bllokun që permban shpenzimin e dyfishte. Nëse kjo ndodh dhe vazhdimi i blloqeve vendosen mbi këtë të fundit, blloku ku ndodhet transaksioni Alice -> Bob perfundon në një bllok jetim (*orphan block*).

Por Bitcoin-i mbrohet nga kjo skemë e shpenzimit të dyfishte duke përdorur konfirmimet. Kur Alice ben broadcast transaksionin që përfaqëson pagesën e saj ndaj Bob-it, Bob-i është duke degjuar në rrjet dhe i vjen një notifikim mbi transaksionin që Alice dergoi, pa u futur transaksioni akoma në një bllok. Nëse Bob-i e pranon transaksionin kjo quhet **zero-proof transaction** pasi nuk priti për asnjë nyje tjetër që të konfirmoj por e mori për të vertete se Alice është nyje e ndershme duke i pranuar monedhat. Ajo cka Bob duhet të beje është të prese për me shumë se sa një konfirmim dhe me pas të pranoj transaksionin. Momentalisht, duhen 6<sup>10</sup> kon-

---

<sup>10</sup>Kur themi se presim 6 konfirmime do të thotë se presim 6 blloqe të mbivendosen mbi bllokun që pret të konfirmohet

firmime që transaksiioni të quhet i sigurte (pra monedhat nuk përdoren me shumë se njehere, bllokun nuk perfundon në chain-in jetim, etj.). Vështirësia për të shpenzuar monedhen 2 here ulet në mënyrë eksponenciale me rritjen e numrit të konfirmimeve.

Për ta përmbyllur, mund të themi se mbrojtja ndaj transaksioneve të pavlefshme është totalisht kriptografike. Por rrjedha e veprimeve bazohet në rregullat e konsensusit, cka do të thotë se nëse një nyje tenton të përfshijë një transaksion të pavlefshëm nga ana kriptografike, arsyeva pse ky i fundit nuk perfundon në chain-in më të gjatë është sepse shumica e nyjeve janë të ndershme dhe nuk do të perfshinin një transaksion të pavlefshëm në blockchain. Nga ana tjetër, mbrojtja ndaj shpenzimit të dyfishte, vjen nga rregullat e konsensusit. Kriptografia nuk ka lidhje me këtë, dhe dy transaksione që perfaqesojnë një shpenzim të dyfishte janë të dyja të vlefshme nga kendveshtrimi kriptografik. Por është konsensusi ai që vendos se cili do të perfundoje në zinxhirin më të gjatë. Asnjehere nuk mund të jesh 100% i sigurte për një transaksion që ai mund të ndodhet në degen e konsensusit. Por, probabiliteti eksponencial është një garanci e mirë. Pas pothuasje 6 transaksionesh, mundesia për shpenzim të dyfishte konvergjon në zero.

### 2.3.6 *Incentives* dhe *Proof of Work*

Bitcoin-i është shumë me shumë se sa thjeshtë një algoritem konsensusi, ajo cka e ben të vecante është pikerisht, mekanizmi ekonomik (*incentive*) i cili i detyron nyjet të jene të ndershme. Me parë supozuam se 50% të hereve procesi i zgjedhjes se një blloku do të zgjidhte një nyje të ndershme. Ky lloj supozim është problematik sidomos kur flitet për vlera monetare. Menyra se si Satoshi e modeloj sistemin i jepte zgjidhje pikerisht këtij problemi, si t'i bëjmë nyjet të jene të ndershme, pa ja detyruar këtë? Pra, cfare incentive t'i japim atyre, në mënyrë që ata vetë ta kërkojnë ndershmerine nga gjithë rrjeti?

Në sulmin e shpenzimit të dyfishte pasi një konfirmimi, a mund të penalizojme nyjen që krijoi bllokun me transaksiionin me shpenzim të dyfishte? Jo, për arsyet me lartë. Por pyetjen duhet ta bëjmë nga ana e kundërt, në vend që të mundohemi të ndryshojme dicka pasi ka ndodhur, duhet të gjejme një mënyrë që ajo mos të ndodhi fare, dhe kjo është fusha e studimit të mekanizmave të modelimit. Dicka që sipas Satoshiit do i bente nyjet të ishin të ndershme do ishte nëse ato do të kishin një përfitim ekonomik, kundrejt këtij sinqueriteti. Meqë nyjet nuk kanë identitet të

botes reale, sistemi nuk i dergon dot para më e-mail, kështu që e vetmja mënyrë do ishte nëse do ekzistonte ndonje monedhë dixhitale, dhe kjo monedhë është pikerisht Bitcoin-i.

Algoritmi i përshkruar deri tani është application agnostic, për të arritur konsensus në një rrjet të shperndare. Në rastin e Bitcoin janë disa pjesë specifike që e bëjnë të vecante, meqë nyjet i mbajme të ndershme duke i shperblyer me monedha.

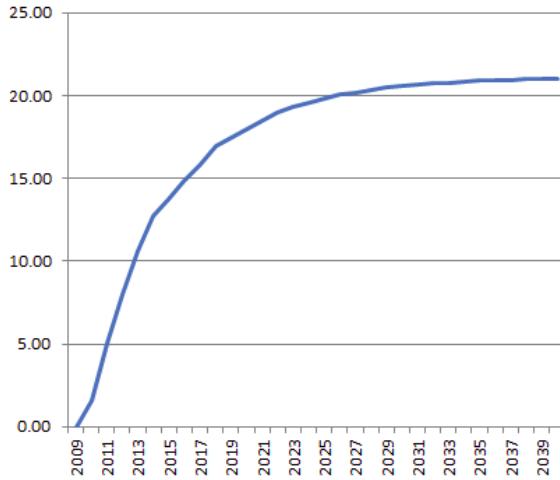
**Shpërblimi për Bllokun** Në Bitcoin ka 2 mekanizma të vecanta motivimi. E para është shpërblimi që merret për bllokun e verifikuar që arrin të behet append në blockchain. Sipas rregullave, nyja që arrin këtë, i lind mundesia të krijoje një transaksion të vecante të quajtur *coinbase* dhe sherben për të krijuar Bitcoin-e të reja.

Kjo vlere ulet mesatarisht cdo 210.000 blloqe. Në momentin e shkrimit të kësaj teze, shpërblimi është 12.5 BTC. Sasia bie cdo 4 vite, dhe tani jemi në periudhen e 3-të. Për 4 vitet e para 2009-2013, shpërblimi për bllok ishte 50 BTC, 2013-2017 ishte 25 BTC dhe tani 12.5 BTC (Fig. 2.10). Kjo percaktohet nga një seri gjeometrike, që paraqitet mëposhtë:

$$S_n = \frac{a (1 - r^n)}{1 - r} = 210000 \times \frac{50 (1 - 0.5)}{1 - 0.5} \approx 21 \times 10^6. \quad (2.1)$$

Por nyja e merr shpërblimin pavarsisht nëse është e ndershme apo jo. Problemi i kësaj qendron se nyja e merr shpërblimin nëse blloku i saj perfundon në chain-in më të gjatë. Ky është inovacioni i kësaj skemë, sepse i ben nyjet të sillen në cfaredo lloj mënyrë që i ben ato të besojne se nyjet e tjera do zgjerojnë bllokun e tyre. Dhe nëse shumica e rrjetit ndjek degen e saktë, kjo i ben gjithë të tjerat të ndjekin atë. Ky është mekanizmi i parë i incentive në Bitcoin si dhe mënyra e vetme për të krijuar monedha të reja.

**Tarifat nga Transaksionet** Ajo cka do i motivoje miners të vazhdojnë të mbrojne Blockchain-in nga blloqet e keqija, pasi të jete arrire limiti prej 21 milion, janë tarifat e transaksioneve. Tarifat zgjidhen nga vetë iniciuese i transakzionit, sa më e lartë tarifa aq me shpejt procesohet nga miner-i, kështu që aq me shpejt behet pjesë e blockchain-it. Ajo cka pritet është që me kalimin e kohes, teksa numeri i Bitcoineve të krijuara ulet, përdoruesit e sistemit do i rrisin vetë tarifat për të patur një quality of



**Figura 2.10:** Shperblimi për bllok, per gjysmohet cdo 4 vite duke dhene një total prej 21 milion Bitoin

service. Kjo është dicka që ka filluar që prej 2 vitesh në Bitcoin, pasi njerëzit donin që t'u perpunohej sa me shpejt transaksioni. Akoma nuk kuptohet mirë se si do shkoje kjo skemë, dhe kjo është e hapur për studime në game theory.

**Mining dhe Proof of Work** Ideja kryesore e Proof-of-Work, është që duam ti zgjedhim nyjet të cilat do procesojne transaksionet në proporcion me një burim të cilin shpresojme asnjë nuk ka mundesi ta monopolizoje. Fuqia kompjuterike është një shembull i tille, dhe në këtë rast themi se sistemi bazohet në PoW. Në Ethereum burimi është pronesia ndaj monedhës Ether, në këtë rast quhet Proof-of-Stake.

Ideja e PoW është që nyjet konkurrojnë me energji kompjuterike. Kjo arrihet me funksionet hash (*hash puzzles*). Në mënyrë që të krijohet një bllok, nyja që propozon atë bllok duhet të gjeje vleren për një *nonce*, e tille që kur të bashkojmë nonce, me vleren e mëparshme të hash-it, se bashku me hashin që ndodhet në rrenjen e pemes merkle të transaksioneve, dhe t'i hashoj të gjitha këto, atëherë kjo vlere duhet të beje pjesë në një hapesire numrash shumë të vogel në krahasim me hapesiren e outputit të funksionit të mesiperme hash. Kete hapesire numrash do e quajme target space [21], dhe mund ta percaktojme si cdo vlere që ndodhet nën një vlere target. Atëherë nonce duhet të plotesoje këtë mosbarazim:

$$H(\text{nonce} \parallel \text{previous\_hash} \parallel tx_1 \parallel tx_2 \parallel \dots \parallel tx_n) < \text{target}$$

Në rast se hash-i që del si output i gjithë bllokut, e gëzon vetinë e të qenë puzzle-friendly atëherë e vjetmja mënyrë për të gjetur nonce është brute force. Miner-at kalkulojne hash-e për të gjetur vleren e nonce gjatë gjithë kohes! Në momentin që gjendet një vlere e nonce, dhe blloku behet pjesë në chain, fillon gara për bllokun tjetër. Dhe kjo është mënyra se si propozohen nyjet, nuk ka asnje entitet qendror të vendose për këtë.

**Vështirësia në Llogaritje** Në fund të vitit 2014 niveli i veshtiresise ishte  $10^{20}$  hash-e të llogaritura për bllok. Ndërkohë që hapësira e target-it është vetëm  $1/10^{20}$  e hapesires se outputit të funksionit hash. Dhe sic e kemi permendur më parë, procesi përgjatë të cilit nyjet llogarisin hash-e për të gjetur nonce, quhet ***Bitcoin Mining*** dhe nyjet që marrin pjesë në këtë proces janë pikerisht ***Miners***. Teorikisht cdo kush mund të jete miner, por duke marrë parasysh sa shumë energji elektrike duhet vetëm një pjesë e vogel e popullsise e kanë mundesine sidomos gjatë 3 viteve të fundit kur cmimi është rritur shumë.

Akoma me tej, probabiliteti që një miner cfaredo, psh Alice, të fitoje bllokun e ardhshem është perafersisht 0.1% e totalit të fuqise se hash-it në rrjet, kështu që do të gjeje perafersisht 1 në 1000 blloqe. Dhe kjo e gjitha për të mbajtur intervalin 10 minutesh bosh nga shtimi i blloqeve të reja në blockchain. Hapesira prej 10 minutash është fillimisht për ceshtje optimizimi, nëse hapësira do ishte psh: 2 minuta, atëherë pak transaksione do ishin brenda një blloku, si rrjedhojë gjatësia e blockchain-it do rritej shumë.

**Kostoja e Parametrizueshme** Vecoria tjetër është se kostoja duhet të jete variabël, dhe jo konstante. Menyra se si arrihet kjo është duke lënë nyjet të rillogarisin target-in cdo 2016 blloqe (2 javë). Target modifikohet në mënyrë të tille që koha mesatare mes blloqeve të njepasnjeshem të jete 10 minuta.

Satoshi nuk ka vendosur kurrë një vlere fikse për 1 Bitcoin, Bitcoin-et janë thjeshtë output i transaksioneve, dhe në rregullat e tanishme ata kanë një vlere arbitrale prej 8 shifresh pas presjes. Vlera më e vogel është  $10^{-8}BTC$ , që është 1 **Satoshi**

Në Bitcoin, ekzistojnë 2 mënyra për të ndryshuar rregullat. *Soft forks* dhe *hard forks*. Një tjetër koncept është ai i *bootstrapping*. Ka tre ide që behen merge me njera-tjetren në Bitcoin:

## 1. Siguria e Blockchain

### 2. Mireqenia e miners

### 3. Vlere e monedhës

Natyrisht që blockchain-i duhet të jetë i sigurte që monedha të këtë vlere. Qe blockchain-i të jetë i sigurte, një kundershtare nuk duhet ta këtë të mundur të thyej rregullat e konsensusit. Kjo do të thotë se kundershtari nuk mund të krijoje shumë mining nodes dhe të marrë mbi 50% të rrjetit ose me shumë se krijimin e një blloku të ri. Kushti i nevojshem që këto rregulla të qendrojnë është komuniteti i singerte i miners, sa me shumë të rritet cmimi i Bitcoin aq me interes kanë dhe aq me shumë duan të blejne, gjithnjë duke parë nëse dalin me fitim (duke marrë parasysh hardware-in e kushtueshem). Ndërkohë që stabilitetin e monedhës e sigurojne vetëm konsumatoret (përdoruesit) të cilet kanë besim në blockchain. Nëse janë të besimit se rrjeti mund të behet hack nga momenti në moment, atëherë Bitcoin nuk do të këtë shumë vlere si monedhë. Dhe kjo është nderlidhja mes sigurisë se blockchain-it, komunitetit të miner-ave të shendetshem dhe një kurs shkembimi. Për shkak të natyres ciklike të kësaj variance, ekzistenca e seciles varet nga ekzistenca e dy të tjerave. Kur Bitcoin u krijuar në fillim, asnje nga këto nuk ekzistonte. Nuk kishte miners pervec Satoshi, Bitcoin nuk kishte shumë vlere si monedhë, dhe Blockchain-i nuk ishte i sigurt pasi nuk kishte shumë aktivitet në mining dhe ishte shumë e lehte të ndodhje sulmi i 51%.

**Sulmi i 51%** Nëse një miner apo grup miners zoteron  $>50\%$  të fuqise se hash-it në rrjet, mund të ndodhin një sere sulmesh:

- Bitcoin-et perseri nuk mund të vidhen nga një adresë ekzistuese. Sepse të vjedhesh Bitcoin do të thotë të fallsifikosh nenshkrimin dixhital, pra të gjesh celesin privat. Sic duket, thjeshtë të thyesh rregullat e konsensusit nuk sjell në vjedhje monedhash nga adresat.
- Ajo që mund të ndodhi është një shpenzim i dyfishte, por nyjet e tjera do e shohin se ky chain permban një transaksion jo të vlefshëm dhe si rrjedhojë nuk do e zgjerojne atë.
- As shperblimin për bllok nuk mund ta ndryshoje dot, pasi secula nyje ka kopjen e vetë të software-it.

Gjithësesi, duke u nisur dhe nga historite e mëparshme, nëse vertete do kishte shenja për një sulm 51% core developers do të nderhynin duke shtuar rregulla në software-in e Bitcoin.

### 2.3.7 Rrjeti i Bitcoin dhe sfidat

Rrjeti i Bitcoin është P2P dhe funksionon si cdo rrjet i tille (psh: BitTorrent). të gjithë nyjet janë të barabarta, nuk ka hierarki apo nyje speciale ose nyje master. Ekzekutohet mbi TCP dhe ka një topologji të rastesishme, ku cdo nyje shoqerohet me nyje të tjera në mënyrë të rastesishme. Nyje të reja mund të bashkohen në cdo moment. Rrjeti ndryshon përgjatë kohes dhe është shumë dinamik për shkak të nyjeve që hyjne e ikin. Nuk ka ndonje mënyrë eksplikite për t'u larguar nga rrjeti, në fakt nëse një nyje nuk është lidhur me ndonje tjetër për disa kohe (në disa klient kjo kohe është 3 ore) nyjet e tjera e harrojnë atë. Në këtë mënyrë rrjeti i nxjerr nyjet offline. Nyjet lidhen në mënyrë të rastesishme me njera-tjetren, dhe nuk ka ndonje topologji gjeografike. Tani le të themi se një nyje e re hyn në sistem. Fillohet me një mesazh në një nyje e cila njihet nga klienti. Zakonisht kjo quhet **seed node**, dhe ka disa mënyra se si mund të skanojmë lista me nyje seed për t'u lidhur me to. Dergohet një mesazh, dhe ky proces perseritet. Më pas zgjidhet një peer, dhe që nga ky moment nyja është pjesë e rrjetit të Bitcoin. Shumë hapa perfshijne faktore të rastesishem, dhe rezultati ideal është të qenit peer me një grup të rastesishem nyjesh. Për t'u lidhur me rrjetin pra, mjafton të dihet si të lidhesh me një nyje të vetme në të.

Rrjeti është i vlefshëm sepse ai miremban blockchain-in. Pra që të publikohet në një transaksion, duhet të publikojme tek i gjithë rrjeti. Kjo ndodh përmes algoritmit *flooding* që shpesh here njihet dhe si *gossip protocol*. Nëse Alice deshiron të paguaj Bob-in, klienti i saj duhet të krijoje për të dhe nyja e saj dergon transaksionin tek të gjitha nyjet që janë të lidhura më të. Secila nga këto nyje e propagon tek nyjet më të cilat lidhet e kështu me rradhe. Secila nga nyjet ekzekuton një sere kontrrollesh për të vendosur nëse transaksioni duhet pranuar apo jo. Nyjet që degjojne për transaksione i shtojne ato në *transaction pool* të transaksioneve për të cilat kanë degjuar por nuk janë ende në blockchain. Nëse një nyje degjon për një transaksion që është tashme në pool, nuk i ben me broadcast. kjo siguron se algoritmi flooding perfundon dhe transaksionet nuk bëjnë loop per gjithnjë në rrjet. Kontrrolli në pool për transaksionin është i shpejte pasi vetëm hash-i kontrollohet.

## 2.3.8 Transaksionet në Bitcoin

Ndryshe nga sistemi ekzistues financierar ku të dhënat mbi transaksionet mbahen në llogari, në Bitcoin nuk ekziston termi "llogari". Bitcoin-i ka një sistem të bazuar në transaksione, cka do të thotë nëse dikush do të kontrrolloj nëse derguesi ka mjaftueshem Bitcoin-e për të kryer një transaksion sheh vetëm transaksionin në fjalë. Ndërkohë, në një sistem të bazuar në llogari do ishte e detyrueshme që të kontrrolloshin të *gjitha* transaksionet ku derguesi ben pjesë për të kuptuar sa para i kanë ngelur. Arsyja pse në Bitcoin është ndryshe është thjeshtë për ceshtje eficence kompjuterike. Në rastin e sistemit të bazuar në transaksione nuk ka pse të kontrollohet i gjithë blockchain-i, gjë që do të ishte e nevojshme në një sistem të bazuar në sistemin me dy hyrje (*double-ledger entry*). Në Bitcoin, transaksiioni ka inpute dhe outpute. Vlera leviz nga inputet tek outputet. Një input është adresa nga ka ardhur monedha (në përgjithësi outputi i transaksionit të mëparshëm). Një output transaksiioni vendos një pronar të ri vleres se transferuar duke e shoqeruar atë me një celës. Celësi destinacion quhet *encumbrance*. Adresat e kusurit (*Change address*) Në Bitcoin, i gjithë output-i i transaksionit duhet të konsumohet nga një tjetër transaksion, nëse ky output është me shumë se vlera që derguesi deshiron të shkembeje me marresin, atëherë duhet krijuar një transaksion i cili i kthen kusurin derguesit. Dhe kjo quhet adresa e kusurit.



**Figura 2.11:** Menyra se si lidhen transaksionet përmes hash-it (logjike e njejte me blockchain-in)[6]

Sa e lehte është të kontrollojme nëse një transaksion në ledger është i vlefshëm?

Kjo nuk është e lehte duke marrë parasysh që po perdomim hash pointers në outputin e transaksionit. Për t'u siguruar se ai nuk është shpenzuar duhet të skanojme blockchain-in nga ky transaksion deri tek blloku më i fundit. Nuk na duhet të shkohje që nga fillimi fare i blockchain-it, dhe nuk ruhen struktura të dhenash shtese. Persa i perket konsolidimit të inputeve, cdo kush mund të bashkoje 2 ose me shumë outpute në një dhe të kryej transaksion. Gjithashtu, pagesat në grup (*joint payment*) krijojnë shpejt. Psh. supozojme se Alice dhe Bob të dy duan ti paguajne Enit. Ato mund të krijojnë një transaksion me dy inpute dhe një output, ku inputet ofrohen nga persona të ndryshem, si dhe transaksioni duhet të këtë dy nenshkrime.

**Sintaksa e transaksioneve:** Një transaksion në Bitcoin ka tre pjesë:

- **Metadata** - Ketu perfshihet madhesia e transaksionit, një seri inputesh, dhe numri i outputeve. Hash-i i të gjithë transaksionit i cili sherben si identifikues unik për transaksionin. Kjo është ajo që na lejon të perdomim hash pointers për të referencuar transaksionet.
- **Inputet** - Inputet e një transaksioni janë në formen e një array, dhe cdo input ka formë të ndryshme. Një input specifikon një transaksion të mëparshëm, kështu që permban një hash të këtij transaksioni, i cili shkerben si hash pointer për të. Inputi gjithashtu permban një index për outputet e transaksionit të mëparshëm. Natyrisht që është dhe një nënshkrim, pasi duhet që të firmosim në mënyrë që të tregojme pronesi për outputet e mëparshme.
- **Outputet** - Outputet janë perseri array, ku secili output ka 2 fusha. Secili prej të cilave ka një vlerë, dhe shuma e të gjithë atyre vlerave outputi duhet të jete më pak ose e barabarte me shumën e të gjitha vlerave të inputit. Nëse shuma e vlerave të outputit është më pak se shuma e vlerave të inputit, diferenca është tarifa e transaksionit për minerin që publikon transaksionin në blockchain.

Kur nyjet degjojne për një transaksion të ri, si vendosin ato nëse duhet ta propagojne atë apo jo? Ekzistojne 4 kontrrolle:

1. Kontrrolli për transaksione të vlefshme - transaksioni duhet të jete i vlefshëm në lidhje me blockchain (main chainin) e atij momenti.
2. Kontrollojne se outputet nuk janë shpenzuar

3. Nuk dergojne/pranojnë një transaksion që e kanë marrë njehere
4. Nyjet By default do të pranojnë vetëm transaksione me scripte që bazohen në një whitelist të skripteve.

Gjithësesi ky nuk është një rregull i prer me thike, nëse një nyje nuk i zbaton prap mund të jete pjesë e rrjetit. Por nyjet e ndershme i zbatojne për të mbajtur blockchainin stabël. Probabiliteti që të ndodhin shpenzime të dyfishta, transaksione jo-standarte etj. ekziston gjithnjë. Meqë në rrjet ka vonesa, mundet që të gjitha nyjet të perfundojne men një paraqitje të ndryshme të transaction pool.

**Skriptet në Bitcoin** Outputet e transaksioneve nuk specifikojnë thjeshtë celësa publik, por specifkojnë skripte. Kjo është një tjetër vecori e menyres se si është ndertuar Bitcoin. Pavec software-it baze i cili është shkruar në C++, Bitcoin përdor një gjuhe të tijen të quajtur *Script*. Transaksi me tipik në Bitcoin është perdonimi i output-it të një transaksi të mëparshëm dhe nenshkrimi i tij me celesin e duhur. Në këtë rast duam që outputi i transaksionit të percoje mesazhin se "ky transaksion mund të përdoret vetëm në pranine e një nenshkrimi nga entiteti i adreses X". Adresa është thjeshtë hash-i i celesit publik. Kështu që thjeshtë specifikimi i adreses nuk na jep dot celesin publik, dhe pa celesin publik nuk verifikojme dot nenshkrimin elektronik. Pra ajo që duhet të na thotë outputi i transaksionit është: "se outputi mund të merret nga një celës publik i cili kur hashohet jep X, se bashku me një nënskrim të pronarit të atij celësi publik"(Fig. 2.11). Pavec outputeve që permbajne skripte, edhe inputet

```

OP_DUP
OP_HASH16069e02e18...
OP_EQUALVERIFY
OP_CHECKSIG

```

**Figura 2.12:** Shembull i skriptit më të zakonshem: Pay-to-PubkeyHash

i permbajne ato në vend të nënskrimeve. Për të verifikuar se një transaksion paraqet outputin sic duhet kombinojme skriptin e inputeve të transaksioneve me skriptin e outputeve të transaksionit të mëparshëm. Thjeshte i bashkojme, dhe rezultati është një skript i cili duhet të ekzekutohet me sukses në mënyrë që transaksioni të quhet

i vlefshëm. Keto dy skripte quhet *scriptPubKey* dhe *scriptSig*, pasi në rastin më të thjeshtë skripti i outputit specifikon thjeshtë celesin publik (ose një adrese tek e cila hashohet celësi publik), dhe skripti i inputit specifikon një nënshkrim me atë celës publik. Një lloj transaksioni interesant në Bitcoin janë ato që i quajme ***Escrow Transaction*** Psh. Alice dershiron të paguaj në Bitcoin përmallra që merr nga Bobi, por Alice nuk preferon të paguaj pepara se malli të arrij, e njejtë vlen dhe për Bobin, ai nuk deshiron të shkembej me Alice pa u siguruar se do të marrë pagesën. Bitcoin-i ofron zgjidhjen duke përfshirë një pale të trete e cila rezulton në një escrow transaction. Teknikisht escrow transactions, implementohen përmes një skripti MULTISIG<sup>11</sup> i cili kerkon 2 ose 3 persona të nenshkruajne në mënyrë që monedhat të përdoren. Në shembullin lartë, 3 njerëzit do jene Alice, Bob, dhe Eni (pala e trete). Kështu që Alice krijon një skript MULTISIG, ku specifikon se monedhat mund të shpenzohen vetëm nëse dy nga pjestarët e grupit Alice, Bob, dhe Eni firmosin. Ky transaksion perfshihet në blockchain, dhe në këtë pike, këto monedha shpenzohen në varësi të grupit me lartë. Tani Bobi mund t'i dergoje mallrat Alice. By default Alice dhe Bob janë të sinqerte. Kështu që Bob dergojn mallrat dhe Alice kur ti marrë ato do të nenshkruaj transaksionin. Në rastin kur të dy janë të ndershme Eni (pala e trete) nuk perfshihet fare.

Por në rast se Bob nuk i dergon mallrat (apo mallrave u ndodh dicka rruges), Alice nuk deshiron të paguaj Bob dhe kerkon monedhat mbapsht, kështu që ajo nuk nënshkruan transaksionin. Nga ana tjeter Bob mund ta mohoje këtë sjellje të tij dhe refuzon nenshkrimin e transaksionit. Eni është ajo që do të vendos kujt do i kthehen monedhat mbapsht. Nëse ajo mendon se Bobi mashtroi atëherë ajo do të nenshkruaj një transaksion me Alice, duke derguar kështu parate nga escrow tek Alice. Nënshkrimet e Alice dhe Eni i plotesojne kushtet për një transaksion MULTISIG, dhe Alice i merr parate e saj. E kunderta qendron gjithashtu, nëse Bobi ka të drejte, Eni bashkon nenshkrimin më të dhe i kalohen leket llogarise se Bob.

Adresat jeshile janë një tjeter aplikacion në Bitcoin. Psh. nëse Alice deshiron të kryej një transaksion me Bobin ndërkokë që ai është offline. Por meqë Bobi është offline ai nuk mund të shohe transaksionin e Alice në blockchain dhe ta prese atë sa të marrë 6 konfirmime (afersisht një ore). Por nëse Bobi ka blerë ushqim për të ngrene nuk mund të prese dot 1 ore. Për të zgjidhur këtë problem (dergiimi i

---

<sup>11</sup>Skriptet MULTISIG kërkojnë me shumë se një nënshkrim në mënyrë që outputi i tyre të shpenzohet.

Bitcoineve kur marresi është offline), duhet të perfshijme një pale të trete, që mund të jete një exchange, një institucion financiar ose një banke. Alice do të flase me bankën e cila do të dergoje një adresë jeshile tek Bobi. Banka jep sigurine që nuk do i shpenzoje 2 here monedhat. Kështu që në momentin që Bob sheh se transaksi është firmosur nga banka, nëse ai e beson bankën, ai i pranon monedhat edhe pa pritur konfirmimet në blockchain. Besimi në këtë set-up nuk sigurohet nga rregullat e protokollit të Bitcoin, por varet nga individi, i cili mund të shohe historikun e bankes, dhe të kuptoje se adresat e saj jeshile përdoren për një kohe të gjatë, dhe nuk janë shpenzuar kurrë me shumë se njehere. Nëse banka e përdor një monedhë dy here, njerëzit thjesht nuk do t'u besojne me adresave jeshile që ajo ofron. Në fakt, Mt.Gox dhe Instawallet (2 shërbime online, Mt.Gox ishte exchange), u mbyllen, pikerisht sepse njerëzit shpenzonin dyhere monedhat. Në kohen e shkrimit të këtij dokumenti, adresat jeshile nuk janë shumë të përdorur. Një tjetër perdonim i skripteve në Bitcoin, janë micro-payments. Psh. Alice është klienti dhe deshiron ta paguaj Bobin vazhdimesht në shuma të vogla parash, me tej, supozojme se Alice deshiron t'i beje pagesat cdo 1 minute. Krijimi i një transaksi cdo një minute, natyrisht që nuk do të funksionoje, pasi krijojen shumë transaksione dhe si rrjedhojë shumë tarifa. Ajo cka duhet është bashkimi i të gjithë pagesave në një të vetme. Kete mund ta krijojmë duke filluar me një transaksion MULTISIG që paguan një shumë maksimum që Alice mund të paguaje (limit i siperm). Pas minutit të parë që Alice ka përdorur sherbimin, ose në momentin e parë që Alice deshiron të kryej një micro-payment ajo nënshkruan një transaksion që shpenzon monedhat që ishin derguar në adresen MULTISIG, duke derguar një njesi pagese tek Bob dhe duke kthyer pjesën tjetër tek Alice. Pas minutit të parë të perdonimit të sherbimit, Alice nënshkruan edhe një transaksion tjeter, kësaj here duke paguar 2 njesi tek Bob dhe duke e derguar pjesën tjetër mbrapsht tek vetja. Keto transaksione nenshkruhen vetëm nga Alice, dhe nuk kanë qenë nenshkruar më parë nga Bob, dhe as nuk janë publikuar në blockchain (prandaj nuk ka tarifim)! Pasi Alice të këtë mbaruar pune, i thotë Bobit se nuk mund ta nderprese ofrimin e sherbimit. Kështu që ajo nuk nënshkruan me transaksione. Ndërkohë Bob pret që Alice të nenshkuaj transaksionin e fundit dhe t'ja dergoje atij, në mënyrë që të publikohet në blockchain.

### 2.3.9 BTC Exchanges

Për të kuptuar exchange-et e Bitcoin duhet që fillimisht të kuptohet se si funksionon një banke. Bankes i jepen para - depozite - dhe banka premton se do i kthej parate me vone. Banka i merr parate dhe i investon. Pervec kësaj, banka ruan një rezerve për raste emergjente e cila në përgjithësi quhet *fractional reserve*. Exchange-et e Bitcoin janë biznese që të pakten nga perspektiva e ndërsaçës së përdoruesit, funksionojnë si një banke. Ato pranojnë depozita bitcoinesh, dhe si cdo bankë, premtojne se do i kthejne mbrapsht. Pervec bitcoineve mund të transferohet edhe para konvencionale - në përgjithësi Euro dhe Dollar - duke bërë transferta nga llogaria bankare. Exchange premton t'i kthejë mbrapsht njëren ose të dyja tipet e depozitave. Veprimet të tjera tipike të bankava janë po ashtu të mundura, psh. mund të konvertosh nga fiat në bitcoin. Dicka tjetër e rëndësishme është se kur bëhen shkembime në këtë mënyrë, nuk krijohet transaksion në blockchainin e Bitcoin. Pra është thjeshtë një tjetër premtim që ben exchange, nuk ka levizje as në blockchain, as në tregun e dollarit. Përdorimi i exchange-eve ka avantazhet dhe disavantazhet e veta. Një nga avantazhet më të mëdha është se ndihmon individet që nuk mund të bëjnë dot mine, të hyjne në ekonomine e Bitcoin. Por disavantazhi qendron tek rreziku që lind në këtë rast, i cili është i njejtë me rrezikun që merr kur merr para në banke. Më konkretisht tre lloje rreziqesh janë:

1. Shumë njerëz kërkojnë të bëjnë térheqje njëkohësisht
2. Exchange-i mund të jete thjeshtë një skemë mashtrimi
3. Punonjësit e bankës mund të dëmtojnë sigurinë

Përgjatë historisë së Bitcoin-it të treja rreziqet kanë ndodhur, madje që në 2013 një studim tregoi se si 18 nga 40 exchange Bitcoin (se bashku me Mt.Gox) perfunduan duke u mbyllur për shkak të pazotesise për të kthyer para ose Bitcoin-e që kishin prentuar [22]. Arsyja se pse me bankat në vendet e zhvilluara nuk ndodh dicka e tille, është se ekziston qeveria e cila e menazhon bankën në mënyra të ndryshme (dhe sipas situatave).

**Regulacioni Bankar** Gjeja e parë që ben një qeveri është të vendos një rezerve minimum si kusht. Në Sh.B.A, likuiditeti që i kerkohet një banke është në përgjithësi 3-10%. Se dyti, qeveritë vendosin rregulla edhe mbi tipet e investimeve që mund të

kryej një banke, dhe menyrat e menaxhimit që ajo do përdorë. Qellimi fundore është që asetet e bankes të vendosen aty ku ka një risk të ulet, pasi ato janë në njefare mënyrë asetet e depozitoreve.

Në kembim të kësaj qeveritë ndihmojne bankën ose depozitorët e tyre. Fillimisht, qeveria vendos deposit insurance. Cka do të thotë, se qeveria premton depozitoret që nëse banka që ndjek këto rregulla bie, qeveria do të marrë një pjesë të atyre depozitave. Shpesh qeveritë sillen edhe si "lender of last resort". Nëse një bank nuk ecen dot para, por është dicka e zgjidhshme, qeveria jep borxh, derisa banka "të marrë veten". Por exchange-et e Bitcoin nuk menaxhohen në këtë mënyrë.

### 2.3.10 Aktivitetet e mining

Satoshi Nakamoto e modeloi Bitcoinin duke patur në mendje arin, kështu Bitcoin mining ka shumë ngjashmeri me gold mining. Shumë pak veta dalin me përfitim, duke lënë shumicen me humbje të mëdha. Cdo miner në Bitcoin lidhet me nyjet e tjera në rrjetin P2P. Kur lidhet një miner aktivitet e mëposhtë ndodhin:

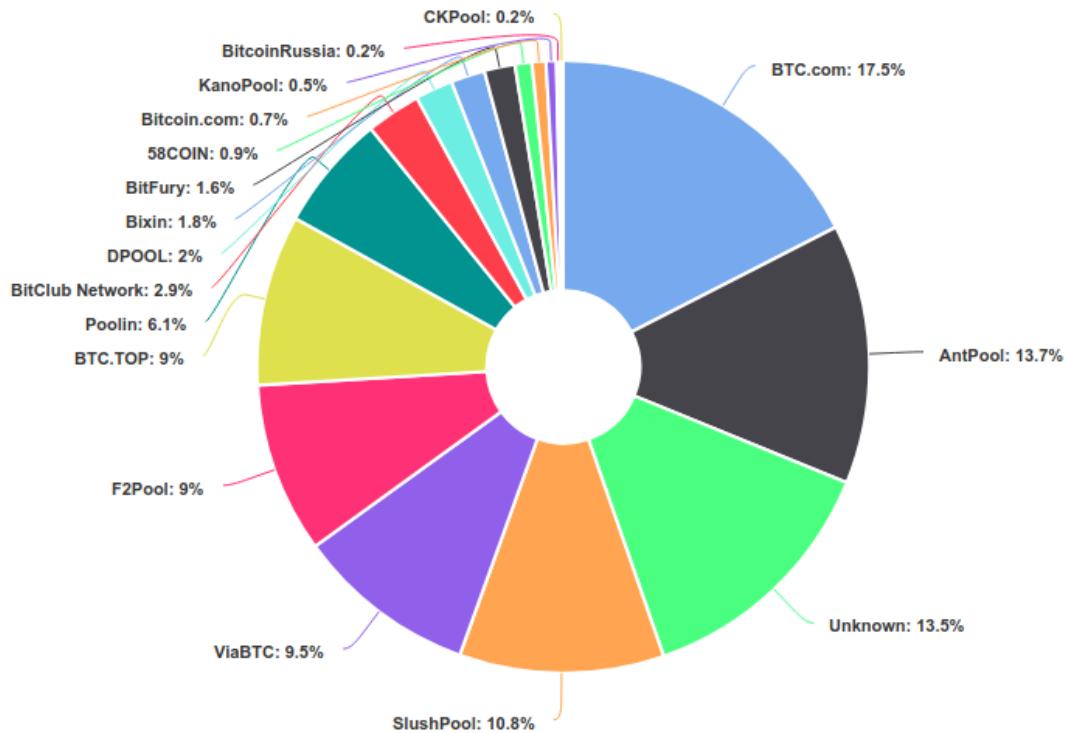
1. *Degjohet për transaksione.* Fillimisht, miners degjojne për transaksione në rrjet dhe i verifikojne ato duke kontrrolluar korrektesine nenshkrimet dixhitale dhe se outputi i transaksioneve nuk është shpenzuar më parë.
2. *Mirembahet blockchain dhe degjohet për blloqe të reja.* Një miner, fillon duke pyetur nyjet e tjera për të gjithë historikun e blloqeve që janë pjesë e blockchain-it përpala se miner-i ti bashkohej rrjetit. Më pas, degjohet për blloqe të rinj që behen broadcast në rrjet. Cdo bllok që vjen duhet të validohet - duke validuar cdo transaksion në bllok dhe duke kontrrolluar vleren e nonce.
3. *Krijohen blloqe kandidat.* Njehere që mineri ka kopjen e fundit të blockchain-it, mund të filloj ndertimin e blloqeve të reja. Për ta bërë këtë, transaksionet grupohen dhe fillojnë llogaritjet për PoW. Cdo transaksion që hyn në një bllok duhet të jete i vlefshëm.
4. *Gjehet vlera e nonce që e ben bllokun të vlefshëm.* Pjesa më e veshtire e procesit të mining. Gjeja e parë që ben një miner është të kompiloje një bashkësi transaksionesh të vlefshme që ka në pending transactions pool në Merkle tree. Mineri mund të zgjedhe edhe numrin e transaksioneve që deshiron të shtoje në

Merkle tree duke patur parasysh limitin prej 1 MB të bllokut. Më pas krijohet një bllok me një header që shenjon tek blloku i mëparshëm. Në header-in e bllokut, ndodhet fusha e nonce prej 32 bitesh, dhe fillohen llogaritjet për një nonce e tille që vlera e nonce e bashkuar me vleren e hashit të bllokut të jetë më e vogel se sa një target (difficulty). Miner mund të filloj edhe thjeshtë me një vlere 0 dhe ta rrise me nga 1, derisa të gjeje nonce-in e duhur. Në shumicen e rasteve miner mund të provoje cdo vlere për nonce dhe asnje mund të mos jetë nën target. Por pervec nonce-it të bllokut kemi dhe nonce-in e transaksionit të coinbase-it. Pasi është provuar cdo vlere e mundeshme për nonce-in me headerin e bllokut, do të duhet të ndryshohet vlera e nonce-it të transaksionit coinbase. Kur ndryshohet nonce-i në coinbase, si rrjedhojë duhet të ndryshoje e gjithë pema Merkle e transaksioneve. Meqë ndryshimi i nonce në coinbase prek të gjithë pemën Merkle, kjo e ben ndryshimin e nonce-it në coinbase një instruksion shumë më të shtrenjte, se sa ndryshimi i nonce në headerin e bllokut. Kështu që shumicen e rasteve, miners fokusohen të provojne të  $2^{32}$  mundesite për nonce në headerin e bllokut pa gjetur një bllok të vlefshëm. Është e rëndësishme të theksohet, se jo cdo miner po mundohet të gjeje të njejten vlere të nonce, pasi transaksionet që një miner grumbullon me shumë mundesi janë ndryshe nga ato të një mineri tjeter. Por edhe nëse miners do të kishin të njejten bashkësi transaksionesh, kanë transaksion coinbase-i të ndryshme (pasi kanë celësa publik të ndryshem), kjo ben që rrenja e pemes merkle, që perfshihet në hash të jetë e ndryshme për cdo miner.

5. *Pritet që miners të tjere të pranojnë bllokun e propozuar* Edhe nëse vlera e nonce është gati, mund të ndodhe një fork, dhe kemi me shumë se një chain të saktë, në këtë rast fati i minerit varet nga minerat e tjere dhe ke bllok do zgjerojne ata.
6. *Perfitohen Bitcoin-e.* Nëse të gjithë miners pranojnë bllokun atëherë, protokolli shperblen me bitcoine të reja.

Veprimet e minerave mund ti klasifikojme në dy kategori. Një pjesë e detyrave - validimi i transaksioneve dhe i blloqeve - ndihmojne rrjetin e Bitcoin dhe janë fundamental për ekzistencen e tij. Keto pune janë arsyaja se pse Satoshi futi konceptin e miners, sepse perndryshe do duhet një central authority për të validuar transaksionet. Pune të tjera - gara që ndodh për të gjetur i pari vleren e nonce - nuk janë

kryesor në ekzistencen e rrjetit, por sherbejne për të motivuar miners të kryejne hapat e permendur me lartë. Natyrisht, që së bashku këto pune janë të domosdoshme për funksionimin e Bitcoin si monedhë, meqë miners duan një shperblim për pune e lodhshme të verifikimit dhe mbajtjes paster se blockchain-it.



**Figura 2.13:** Shpërndarja e fuqisë së hash-it në botë

### 2.3.11 Sfidat teknike të Blockchain-eve

Duke qenëse Bitcoin-i është një protokoll open-source, implementime të ndryshme ekzistojnë të cilat nderveprojne me njera-tjetren. Si rrjedhojë, edhe nëse ekziston një bug tek njera prej tyre, me siguri nuk do të jete tek të tjerat. Edhe pse protokolli është ri-implementuar (C++ dhe Go), shumica e nyjeve në rrjet vazhdojnë të veprojne nën librarin *bitcoind*, e mirembajtur nga Bitcoin Core developers dhe disa prej nyjeve

ekzekutojne versione të cilat nuk janë bërë upgrade. Limitime të tjera janë rregullat e hard-koduara në software të cilat janë aty që kur Bitcoin u propozua në 2009, përpara se të merrte famë si monedhë globale. Disa rregulla të tilla janë:

- Koha mesatare për bllok (10 minuta)
- Madhësia e bllokut (1MB)
- Numri i nënshkrimeve në një bllok
- Pjestueshmëria e monedhës
- Numri total i Bitcoineve
- Skema e shpërblimeve për bllok

Limitet mbi numrin total të Bitcoin-eve si dhe struktura e shpërblimeve për mining mund të mos ndryshojne kurrë duke marrë parasysh që prekin incentive që kanë miners dhe investitoret, dhe implimkimet ekonomike në këtë rast do ishin të mëdha. Nga ana tjetër, disa zgjedhje të Satoshiit mbi modelimin fillestar nuk bëjnë shumë kuptim. Limiti në throughput-in e sistemit është një prej tyre. Sa transaksione do të procesohen për sekond? Limitimet vine nga limiti i hardkoduar mbi madhesine e bllokut.

- Cdo bllok është i limituar në 1MB = 1.000.000 byte
- Cdo transaksion është të pakten 250 byte =>  $1.000.000 / 250 = 4000$  transaksione për bllok
- Blloqet gjenden cdo 10 minuta =>  $4000 / 600 \text{ (s)} = 7$  transaksione për sekond, ndërkohë që VISA kryen 2000-10.000 transaksione për sekond
- Algoritmi i nënshkrimit ECDSA nën vijën eliptike *secp256k1*, kur se fundmi NSA e ka hequr ECDSA nga perdorimi.
- Funksionet Hash të familjes SHA, pervec SHA-256 që konsiderohet më i sigurte (edhe pse mendohet të zevendesohet me SHA-3 kur ai të dale), Bitcoin-i për gjuhen e transaksioneve Script, përdor SHA-1 për të cilin kriptoanalistet dinë shumë dobesi, kjo do kerkonte zgjerim në të gjithë gjuhen e programimit të Bitcoinit për të suportuar algoritme të reja kriptografike.

të besh ndryshime në një sistem të decentralizuar është mjaft sfiduese, dhe një nga arsyet që e ben implementimin e blockchain-eve të veshtire. Gjithësesi, ekzistojnë dy mënyra për të sjelle ndryshime në protokoll:

- *Hard fork*: Software-i i ri do të njihte blloqet si të vlefshme edhe pse versioni tjetër i software-it i njeh si të pavlefshme. Është pothuasje e frikshme të mendosh se si do jete rrjeti nëse disa nyje pranojnë të behen upgrade dhe të tjera jo. Se shpejti main chain, do të permbante blloqe që konsideroheshin të pavlefshme nga nyjet e vjetra. Kështu që nyjet e vjetra do vazhdonin në branchin që ata mendojne se është i saktë, duke zgjeruar në këtë mënyrë branch-in më të shkurter (jo main branchin). Quhet hard fork, meqë e ben blockchainin të ndahet në dy pjesë. Cdo nyje në rrjet do jete në një nga branchet, në varësi të protokollit që po ekzekuton. Branch-et nuk do të bashkohen kurrë bashke, dhe për këtë arsyе kjo konsiderohet e papranueshme nga komuniteti meqë nyjet e vjetra do të dilnin direkt nga blockchain-i i ri, nëse do të benin upgrade software-in.
- *Soft fork*: Një lloj i dyte ndryshimi që mund të bëjmë në Bitcoin është të shtojmë vecori të cilat i bëjnë rregullat e validimit me strikte. Si psh. e bëjnë validimin e transaksioneve më të veshtire. Në këtë rast mund të ndodhe që miners me rregullat e vjetra, të bëjnë mine blloqe të pavlefshme, pasi ato perfshijne transaksione, të cilat nën rregullat e reja nuk duhet të pranohen. Por në këtë rast miners do e shohin se blloqet e tyre po refuzohen maxhoriteti dhe mund ta kuqtojne kështu se duhet bërë upgrade. Si dhe nëse dega e tyre merret nga miners me versionin e ri, ato shkojne tek kjo dege, dhe nuk krijojne hard fork, thjeshtë do këtë forks të njepasnjeshme por që do zhduken shpejt.

## 2.4 Ethereum dhe Smart Contracts

Escrow transactions në Bitcoin, janë një aplikacion interesant, por gjuha Script e tij është disi e limituar duke marrë parasysh se është Turing-incomplete. Si rrjedhojë, disa altcoins kanë propozuar shtimin e funksionaliteteve application-specific. Namecoin ishte e para ndër to, por plot të tjera kanë propozuar cryprocurrencies si Bitcoin që suportojne baste, parashikimin e tregut, dergim stock-u etj.

Problemi më të gjitha është se nuk kanë një gjuhe të perbashket, dhe asnjerë nuk suportan cdo aplikacion. Në një nivel teknik ajo cka duhet është një gjuhe Turin-complete, dhe është po e njejtë histori si në 1940, ku ndertoheshin makineri të ndryshme për qellime të ndryshme, Enigma ishte një prej tyre, kemi makina për të determinuar trajektorët e armeve të luftes etj. Kjo beri që Alan Turing të vinte me gjuhen e parë general-purpose që mund të perdorej për cdo aplikacion (Është Vitalik Butnik, Alan Turing?).

Ethereum është një platform e cila ofron një gjuhe Turing-complete për të programuar skripte ose ato cka i quajme "kontrata".

Termi *smart contract* është përdorur për here të parë në kontekstin e perdomit të programimit për të detyruar kontratat. Pra kompjuterat i mendojme si makineri që detyrojne 2 pale të bashkepunojne për të shkembyer shërbime. Në Ethereum një kontrat është një program që qendron në blockchain. Kushdo mund të krijoje një kontrate Ethereum kundrejt një tarife të vogel, duke bërë upload programin përmes një transaksi. Kjo kontrate është shkruar në bytecode dhe ekzekutohet në EVM (Ethereum virtual machine). Në momentin që kontrata behet publish në blockchain nuk mund të hiqet me (meqë sipas perkufizmit blockchain-i është immutable). Kontrata ka fondet e veta, përdorues të tjere mund ta aksesojne kontraten përmes API që ekspozon Ethereum, dhe smart-contract mund të dergoje dhe marrë para.

Kontratat në Ethereum implementohen përmes Solidity (Ethereum's high-level language). Përmes Ethereum, arrihet të programohen logjika të ndryshme, psh. edhe një cryptocurrency si Bitcoin mund të programohet në Ethereum. Meqë Ethereum suporton ciklet, kjo do të thotë se një kontrate mund të ekzekutohet pafundësisht. Për ta ndaluar këtë, përdoret një mekanizem i quajtur *gas*. Cdo instruksion që ekzekutohet në Ethereum kushton një sasi të vogel parash që perseri, quhen *gas*. Operacione të ndryshme kushtojne sasi të ndryshme. Psh. instruksioni i mbledhjes kushton 1 gas, ndërkohë që llogaritje e një hash-i me SHA-3 kushton 20 gas dhe shkrimi i një fjalë 256-bit në hard-drive kushton 100 gas. Cdo transaksion kushton 21.000 gas. Pra cdo veprim që ndodh në blockchain-in e Ethereum kushton. Kostoja e cdo veprimi është konstante. Ndryshimi i këtyre do të kerkonte një hard-fork, dhe do ishte analoge me ndryshimin e rregullave në protokollin e Bitcoin-it ose në gjuhen Script të këtij të fundit. Gas paguhet për perdomin e cryptocurrency-të të përdorur nga Ethereum të quajtur ether. Pra monedha e Ethereum quhet ether, por kur përdoret për të ekzekutuar kontratat në blockchain quhet gas. Cdo transaksion mund të specifikoje

gas price-in për të cilin është gati të paguaj (sa ether do të paguaj për njesi gas të konsumuar). Cmimi i gas-it që ofrohet është ekuivalenti i tarifave të transaksioneve në Bitcoin: miners janë të lire të publikojne transaksionet me cdo cmim gas-i, dhe cdo miner në mënyrë të pavarur vendos strukturen e këtyre tarifave. Kjo rezulton në një cmim tregu për gas-in, duke paraqitur kështu kërkesën dhe oferten në ekonomi.

**Balancat në llogarite e Ethereum** Bitcoin-i ka një transaction-based ledger ndryshe nga tipikja account-based. Në këtë rast, blockchain-i ruan vetëm transaksionet (se bashku me metadata në block headers). Për ta lehtësuar validimin e transaksioneve, Bitcoin-i i konsideron monedhat si immutable, dhe outputi i transaksioneve duhet të shpenzohet i gjithi, duke përfshirë një llogari kusuri nëse është e nevojshme. Transaksionet operojne në një gjendje globale që është një list UTXOs, por në protokollin e Bitcoin, gjendja e kësaj liste nuk jepet në mënyrë eksplikite nga protokolli i Bitcoin, dhe është thjeshtë dicka që krijohet nga miners për të shpejtuar verifikimin. Nga ana tjetër, Ethereum përdor një model të bazuar në llogari dhe jo transaksione. Meqë Ethereum-i i ruan strukturat e të dhënave që lidh adresat e kontratave me gjendjen e tyre, është e natyrshme që të ruaj edhe balancat e llogarive të cdo adrese të rregullt (ndryshe quhen *owned address*). Kjo do të thotë se në vend që pagesat të paraqiten përmes një grafi transaksionesh pa cikle, ku cdo transaksion shpenzon inpute dhe krijon outpute, Ethereum ruan balancat e cdo adrese njëloj sic bankat ruajn balancat e cdo llogarie.

# Kapitulli 3

## Tregu dhe ekonomia e Bitcoin

### 3.1 Si t'i klasifikojmë Cryptocurrency-të?

#### 3.1.1 Besimi dixhital

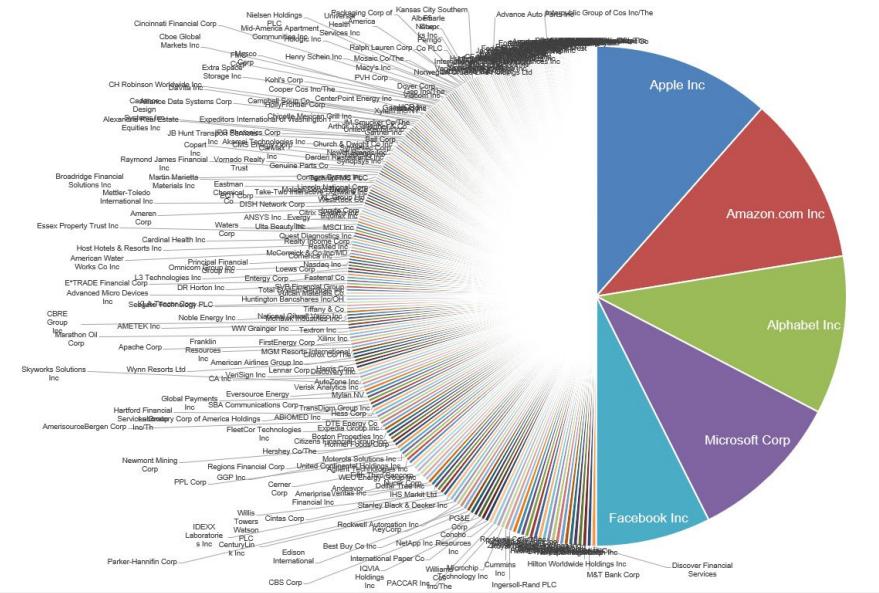
Në vendet e zhvilluara, monedhat dhe bankat funksionojne mjaft mirë, duke i ofruar siguri qytetareve, gjithësesi, në bote dixhitalizimi po rritet jashtezakonisht shpejte (Fig. 3.1) dhe një produkt i këtij zhvillimi janë edhe teknologjite e decentralizuara. Në këtë rast link pyetja nëse paraja duhet të vazhdoje të ekzistoje në formë fizike (leter), apo të kthehet në një token dixhital (*tokenization*) dhe mbi të gjitha të sistemi monetare të bazohet në cryptocurrency? By default, sistemet e centralizuar kanë qenë përgjatë gjithë historise moderne financiare me probleme [23] [24] [25] [26]. Kështu që sistemet e centralizuar nuk funksionojne mirë. Dhe ky është një problem që prek pothuajse cdo shtet ndonese për arsyet e ndryshme (Fig. 3.2 )

#### 3.1.2 Gjenerata e ndryshimit

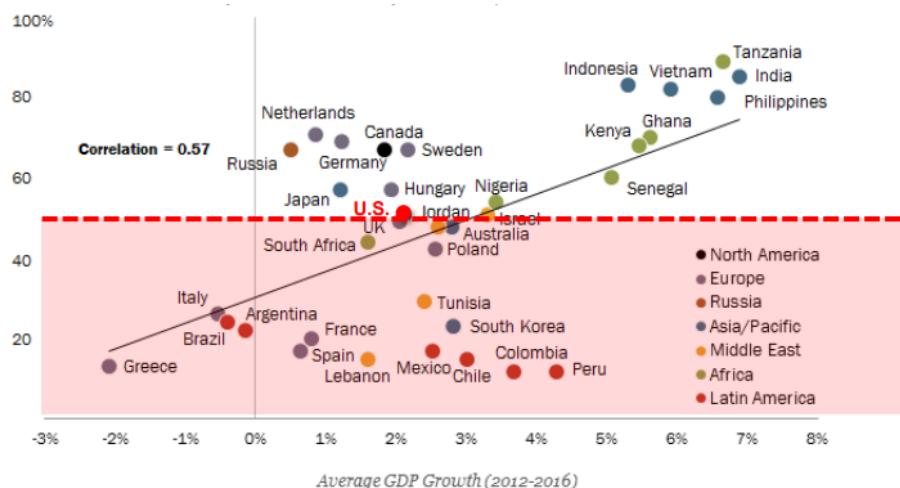
Cryptocurrency-të janë një fenomen i 2 gjeneratave të fundit <sup>1</sup>. Afersia e tyre me teknologjine është një ndër faktorët kryesore se pse cryptocurrencieset dhe forma të ndryshme decentralizimi do mbijetojnë pavaresisht sfidave teknologjike. Dhe sic kemi parë tek Figura 2.3, ata nuk kanë shumë besim në qeveri dhe sistemet e centralizuar. Ndër vite mund të shihet qarte se si është pranuar inovacioni teknologjik. Fillon me skepticizem derisa arrin të stabilizohet, pasi është nevoja ajo që shtyn. Permendim

---

<sup>1</sup>Gjenerata Z e lindur në 1995-2012 dhe gjenerata e lindur në 1980-1995 ose ndryshe *millennials*



**Figura 3.1:** 5 kompanite teknologjike në S&P 500, vlejne me shumë se 282 kompani të tjera, në baze të market capitalization



**Figura 3.2:** Studim i Pew Research Center, me pyetje: *Sa besoni se qeveria ben gjene e duhur për shtetin?*

ketu, levizjet informatike nga nderfaqet grafike, PC apo Apple, me pas tek interneti ku në fillim shpejtesia ishte në nivel kilobitesh, deri tek ditet e sotme të medias sociale. Tani se fundmi, është shtuar në liste edhe blockchain-i, e cila premton me shumë se

cfare kemi parë deri tani, duke qenëse prek financat.

## 3.2 Forcat e Kërkesë-Ofertës

Si cdo treg, edhe ai i shkembimit të Bitcoineve lidh shitesit me bleresit. Tregu arrin qindra biliona USD (Fig. 3.1), dhe pse jo aq i madh sa NYSE (New York Stock Exchange) apo kursi i USD-EUR, por është aq i gjere sa të këtë një konsensus për cmimin. Cmimi i tregut vendoset nga forcat e kerkese-ofertes. Pra potenciale që bitcoinet të shiten (oferta) dhe potenciali që ajo të kerkohet (kërkesa) nga njerëzit që kanë USD. Sipas këtij mekanizmi do të vendoset nga nivelet ku perputhen kërkesa me oferten.

Oferta për bitcoine është numri i bitcoineve që mund të blihen në një nga këto tregje, dhe është i barabarte me oferten e bitcoineve që janë në qarkullim. Ekziston një numër fiks bitcoinesh në qarkullim, për momentin është afersisht 17 milion, dhe ky do vijë duke u rritur deri sa të arrihet 21 milion.

Nga ana tjetër, kërkesa ka 2 burime kryesore për bitcoin. Ekziston një kerkese për bitcoinin si mjet për shkembim mallrash (jo aq shumë sot) dhe kërkesa për bitcoinet si investime. Për rastin e parë, le të supozojme se Alice deshiron të blej dicka nga Bob i cili ndodhet në një tjetër kontinent. Për të evitar tarifat e institucioneve financiare, Alice vendos të blej me bitcoine. Supozojme se as Alice dhe as Bob, nuk mendojne për ti mbajtur bitcoinet për investim. Gjeja kryesore në këtë kendveshtrim është se nga pikepamja e kerkeses për bitcoine, bitcoinet ndodhen në një transferte kështu që po hiqen nga qarkullimi gjatë kohes që transaksi i ndodh. Kjo krijon kerkese për bitcoinin. Burimi i dyte është ai për investime. Pra nëse dikush deshiron të bleje bitcoine dhe ti mbaj ato me shpresen se cmimi i bitcoineve do të rritet, dhe në të ardhme do mund t'i shese me cmim më të lartë. Kur njerëzit blejn dhe nuk përdorin, do të thotë se bitcoinet hiqen nga qarkullimi. Kur cmimi i bitcoinit të bie, mund të pritet që shumë njerëz të blejne bitcoinin për investim, por nëse cmimi shkon shumë lartë, atëherë kërkesa për investim nuk do jete aq e lartë.

**Sjellja e tregut** Për të kuptuar sjelljen e tregut, mund të ndertojmë një model. Disa parametra do të ishin si mëposhtë:

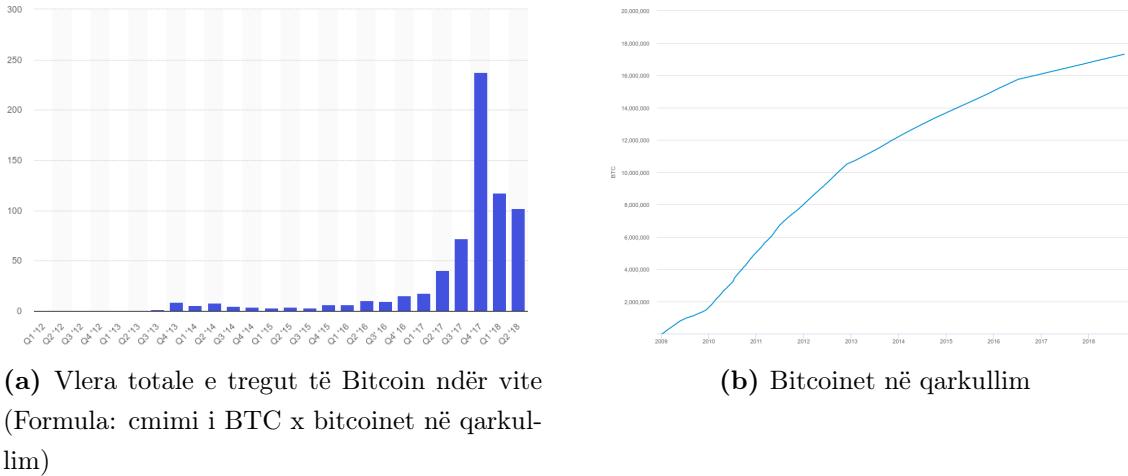
- $T \rightarrow$  vlera totale e transaksioneve nga të gjithë pjestarët e tregut (i matur në

USD për sekonde)

- D -> koha që i duhet bitcoineve për t'u transferuar, që sjell në nxjerrjen e bitcoineve nga tregu, që është koha nga momenti që bleresi blen bitcoine, deri kur marresi është i afte ti shese në treg (i matur në sekonda)
- S -> oferta totale e bitcoineve që janë gati për t'u blerë - për momentin pak a shumë 17 milion or 21 milion minus ato që mbahen nga njerëzit si investime. Në fjalë të tjera, po flasim për Bitcoine të cilat përdoren për transaksione, jo ato që ruhen.
- P -> cmimi i Bitcoin, (i matur në USD për bitcoin)

Fillimisht, mund të llogarisim se sa bitcoine janë gati për t'u përdorur në transaksione cdo sekonde. Meqë kemi S bitcoine që hiqen nga qarkullimi cdo D sekonda, atëherë mesatarisht cdo sekond  $S/D$  e këtyre bitcoineve është e disponueshme, pasi dalin nga gjendja e bitcoineve që janë jashte qarkullimit. Nga ana e kerkeses - numeri i bitcoineve për sekonde që duhen për të krijuar transaksione - po të kemi gjithesej transaksione që vlejne një total prej  $T$  USD dhe në mënyrë që të marrim 1 USD për transaksion na duhen  $1/P$  bitcoine. Kështu që  $T/P$  është numri i bitcoineve për sekonde që duhen për t'u sherbyer transaksioneve për të cilat ka kerkese. Me tej, nëse shohim tregun në një sekonde të caktuar, do tekete një oferte prej  $S/D$  dhe një kerkese prej  $T/P$ . Në këtë treg, si në shumicen e tregjeve, cmimi ndryshon në mënyrë që ta sjelle oferten në të njejtën linje me kérkesën. Nëse oferta është më e lartë se sa kérkesa, atëherë ka bitcoine që nuk po shiten, kështu që njerëzit që shesin bitcoine do ulin cmimin në mënyrë që ti shesin ato. Dhe sipas formules  $T/P$  për kérkesën, kur cmimi të ulet kérkesa do rritet, në mënyrë që kérkesa dhe oferta të jene në ekuliber.

Nga ana tjetër, nëse oferta është shumë më e vogel se sa kérkesa atëherë do të thotë se ka njerëz që duan të përdorin bitcoine për transaksione por nuk i marrin dot pasi nuk ka mjaftueshem në qarkullim. Si rrjedhojë, keta njerëz duhet të rrisin cmimin që janë gati të paguajne për bitcoine (bid), pasi do të këtë shumë konkurrence për oferten e limituar të bitcoinit. Kjo i con cmimet lartë, po t'i referohemi formulave, kjo do të thotë se kérkesa do ulet deri sa të këtë një ekuliber. Në ekuliber, oferta duhet të jete e barabarte me kérkesën:  $S/D = T/P$  cka na jep:  $P = TD/S$ . Mund ta mendojme se  $D$ , koha e transaksionit të bitcoinit, nuk ndryshon. Oferta totale  $S$  ndryshon shumë pak ose pothuajse fare. Kjo na thotë se cmimi është proporcional



**Figura 3.3:** Market cap dhe oferta

me kérkesën për transaksione në USD. Pra nëse kérkesa për transaksione në USD dyfishohet atëherë cmimi i bitcoineve duhet të dyfishohet.

Gjithësesi në këtë rast, S perfshin vetëm bitcoinet që nuk mbahen si investim. Pra nëse me shumë njerëz po blejne bitcoine për investim, S do ulet, dhe formale na thotë se P rritet. Dhe ka kuptim, sepse nëse ka me shumë kerkese nga ana e investimit atëherë cmimi që duhet të paguhet shkon lartë.

Një model i plote i tregut duhet të përfshijë dhe investitoret, pra ato që kérkojnë bitcoine sepse mendojne se cmimi do të këtë rritje në të ardhme, kështu që duhet të marrim parasysh se cfare mendimi kanë investitoret. Keto mendime, lidhen me kérkesën që pritet të këtë bitcoinin në të ardhme.

Ideja kryesore është se ekziston një treg mjaft likuid mes BTC dhe USD apo monedhave të tjera. Së fundmi, është e mundur që të krijohen modele ekonomik për të krijuar një perspektiv se si oferta dhe kérkesa nderveprojne me këtë treg dhe të parashikohet se cfare mund të ndodh në treg, për aq kohe sa ekziston një mënyrë për të matur faktore që nuk dihen si psh. sa njerëz do e duan bitcoinin për të kryer transaksione në të ardhme.

$$x' = \frac{a(1 - r^n)}{1 - r} = 210000 \times \frac{50(1 - 0.5)}{1 - 0.5} \approx 21 \times 10^6 \quad (3.1)$$

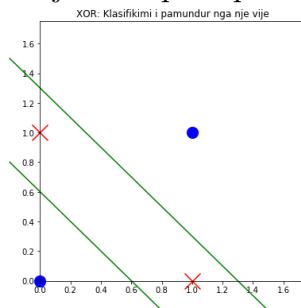
# Kapitulli 4

## Rrjetat e Thëllë Neural

Në këtë kapitull diskutohet arkitektura e rrjetit neural LSTM. Duke shpjeguar se cfare problemi zgjidhin ato dhe si janë përdorur në evaluimin e cmimit të Bitcoin, duke përfshirë të gjithë hapat e ndertimit të një modeli rrjeti neural. Në fund disktohet hyperparameter tuning si një nga fazat më të rëndësishme për të validuar modelin.

### 4.1 Rrjetat neural me shumë shtresa (MLP)

Rrjetat neural me shumë shtresa të quajtura (*multilayer perceptron*) formojne bazen e rrjetave neural që përdoren dhe neper aplikacione komerciale. MLP është përdorur për here të parë si zgjidhje për problemin e XOR, i cili nuk pershkruhej dot nga modeli i thjeshtë i perceptronit (Fig. 5.1).

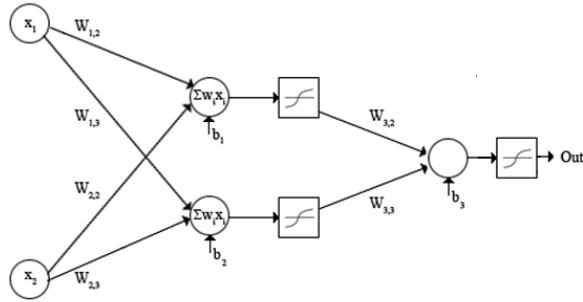


**Figura 4.1:** Pse XOR solli MLP

$x_1$	$x_2$	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

**Tabela 4.1:** XOR

Rrjetat me shumë shtresa kanë si qellim perafrimin e një funksioni  $f^*$ . Një arkitekturë e tille, percakton një funksion  $\mathbf{y} = \mathbf{f}(\mathbf{x}; \theta)$  dhe permireson vlerat e parametrit  $\theta$  që rezultojne në perafrimin më të mirë të funksionit. Keto modele në formen e tyre më të thjeshtë, i propagojne të dhënrat vetëm përpëra, nga inputet tek funksionet e



**Figura 4.2:** Zgjidhja e XOR me një shtresë të fshehur mes inputit ( $x_1, x_2$ ) dhe outputit

shtresave të mesit me rradhe, pa u kthyer mbrapa. Në rastin e serive kohore, do të na duhet një mekanizem për t'u rikthyer pas, dhe do shohim si ndryshon arkitektura me lartë, nga ajo e Recurrent Neural Network.

Arsyeja pse këto perceptrone formojne rrjeta është lidhja që ekziston mes funksioneve. Për shembull, nëse do kishim 3 funksione  $f^{(1)}, f^{(2)}$  dhe  $f^{(3)}$  të lidhura në zinxhir, do formonin  $f(x) = f^{(3)}(f^{(2)}(f^{(1)}))$ . Kështu,  $f^{(1)}$  përbën shtresën e parë,  $f^{(2)}$  atë të dyten, etj. Gjatesia e këtij zinxhiri funksionesh, na jep thellsinë e rrjetit, nga ku rrjedh dhe emri *mesimi i thelluar (deep learning)*. Shtresa e fundit, është shtresa e outputit. Përgjatë trajnimit të rrjetit (ndryshimit të parametrave), mundohemi që  $f(x)$  të perputhet me  $f^*(x)$ . të dhënat e trajnimit nuk janë perfekte, kështu që vetëm një perafkim mund të arrihet me funksionin e vertete.

Së fundmi, duke qenëse roli i secilit perceptron në rrjet është analog me një neuron në tru, rrjetin e quajme rrjet neural dhe nyjet neurone. Numri i neurone në shtresat e fshehura na jep gjeresine e modelit.

### 4.1.1 Neuronet jo-linear

Për të zgjeruar një model linear si regresi apo sigmoid për të përshkruar funksione jo lineare, mund të aplikojmë një model linear jo tek inputi  $x$  por tek një input i transformuar  $\phi(x)$ , e tille që  $\phi$  është një transformim jolinear. Në mënyrë të ngjashme, mund të aplikojmë një kernel (dritare ose konvolucion), për të përafruar një algoritem që meson funksione jolineare, duke u bazuar në thjeshtesine e  $\phi$ . Ky funksion nuk dihet paraprakisht kështu që na duhet ta mesojme. Një mënyrë për ta riformuluar modelin do ishte si:  $\hat{y} = f(x; \theta, w) = \phi(\mathbf{x}; \theta)^w$ , ku parametri  $\theta$  përdoret për të mesuar

$\phi$  dhe parametri  $w$  për të shprehur lidhjen e  $\phi(x)$  me outputin e vertete [27]. Nëse marrim si shembull, funksionin XOR të diskutuar me lartë, i cili merr si input një mënyrë binar  $x = (\mathbf{x}_1, \mathbf{x}_2) \in \{0, 1\}$  dhe outputi është si në Tabelen 5.1.

Për të përafruar  $f^*$ , na duhet të mesojme parametrin  $\theta$  të funksionit  $\hat{y} = f(x; \theta)$ . Për të klasifikuar saktë inputet, perceptron i Rosenblatt nuk mund të përdoret, meqë ai përdoret një kombinim linear,  $g : x \mapsto \mathbf{w}x + b$ , me threshold  $x \mapsto 1_{x \geq 0}$ . Duke marrë  $w = (w_1, w_2)$  si vektor për peshat e perceptronit dhe  $b$  si vlerë për vektorin e *bias*, parametrat e modelin janë  $\theta = (w, b)$ . Perceptroni lidh inputet  $x$  me outputet  $\hat{y}$  si mëposhtë:

$$\hat{y} = f(x; w, b) = 1_{\mathbf{w}x+b \geq 0} \quad (4.1)$$

Dhe sic duket me lartë, ky është një klasifikues linear që cdo  $x$  e shoqëron me një nga dy klasat  $C_0$  dhe  $C_1$ , kur outputi  $\hat{y}$  është 0, dhe 1 respektivisht.

**Funksioni i Aktivizimit** Një funksion aktivizimi në një neuron të fshehur aplikon funksionin  $h_i = g(\mathbf{W}_{:,i}x + b_i)$ , e tille që  $\mathbf{W}_{:,i}$  është vektori i peshave të lidhura me neuronin  $i$  dhe bias-in përkatës  $b_i$ .

Në varësi të problemit ekzistojnë funksione të ndryshme aktivizimi, në rastin e regresit një nga më të perdorshmit sidomos në rrjetat me përsëritje është *tanh - Hyperbolic Tangent* (i cili është edhe vlera default në Keras). Gjithësesi para *tanh* duhet të shohim funksionin *sigmoid* (Fig. 5.3) meqë përdoret gjërësisht në disa njesi të rrjetit LSTM.

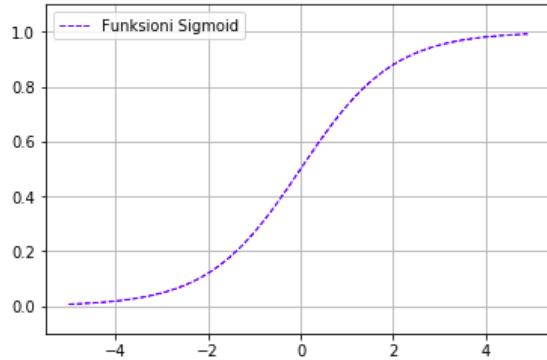
Funksioni percaktohet si:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.2)$$

, dhe arsyja kryesore e perdorimit është se percaktohet në intervalin  $[0; 1]$ . Kështu, përdoret gjërësisht në modele ku duhet të parashikojme një probabilitet si output, dhe meqë në përgjithësi vlerat i normalizojme nga 0 në 1, zgjidhet sigmoid. Funksioni është i diferençueshem<sup>1</sup>, por shpesh mund të shkaktoj që rrjeti neural të ngece në minimume lokale gjatë kohes se trajnimit. Funksioni softmax është më i gjeneralizuar se ky i fundit, dhe përdoret me gjërësisht në shtresën e fundit në problemet e klasifikimit.

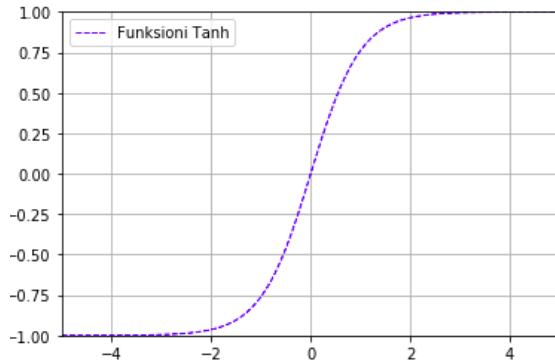
---

<sup>1</sup>Arsyeja se pse derivimi është kaq i rëndësishëm në , është pasi na lejon ti extrapolojme apo interpolojme vlerat sipas nevojes, pa dale shumë nga vlerat minimum apo maksimum të inuteve.



**Figura 4.3:** Funksioni i aktivizimit Sigmoid

*Tanh* nga ana tjetër leviz në intervalin  $[-1; 1]$ , dhe ka po të njejten formë si sigmoid.



**Figura 4.4:** Funksioni i aktivizimit Tanh

Funksioni i *tanh* percaktohet nga:

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (4.3)$$

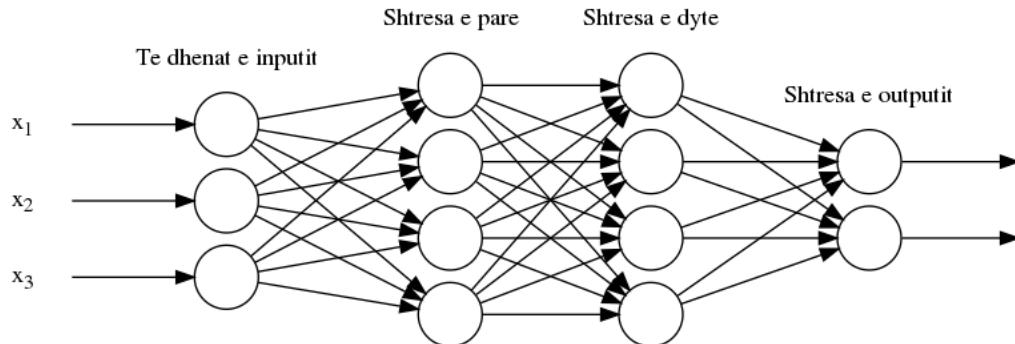
#### 4.1.2 Perceptroni me shumë shtresa

Vendosja e arkitekturese se rrjetit neural ka rendesi shumë të madhe pasi percaktohet struktura e thellisise, gjeresise, dhe lidhjeve mes neuroneve. Shumica e rrjetave neural kombinojne shtresat e fshehura në një strukture zinxhire (sic pame me lartë),

ku outputi i shtreses se mëparshme është input për atë pasardhesen:

$$\begin{cases} h^{(1)} = g^{(1)}(W^{(1)T}x + b^{(1)}) \\ h^{(2)} = g^{(2)}(W^{(2)T}h^{[1]} + b^{(2)}) \\ \dots \\ h^{(d)} = g^{(d)}(W^{(d)T}h^{(d-1)} + b^{(d)}) \end{cases} \quad (4.4)$$

Kete konfigurim rrjeti e quajme *perceptron me shumë shtresa* ose *rrjet neural me shumë shtresa* (Fig. 5.5). Vendimi për thellsine e rrjetit është shumë i rëndësishëm në modelim. Në përgjithësi, një shtresë e fshehur mjafton në shumicen e problemeve [28], kur i jepen funksionet e duhura të aktivizimit, dhe numër i kenaqshem neuronesh. Gjithësesi teorema nuk thotë se cdo funksion mund të mesohet, thjeshtë që MLP mund të paraqese cdo funksion, pasi rrjetat shumë të mëdha (shumë neurone) janë shumë të ngadalta rrreth trajnimit, dhe prandaj preferojme ata të thellët kundrejt atyre të medhenj.



**Figura 4.5:** Rrjeti neural me 2 shtresa të fshehura, 3 të dhëna hyrese, 4 neurone në secilen shtresë të fshehur, dhe 2 vlera në dalje

### 4.1.3 Backpropagation

Rrjetat neural sic pame, mund të perafrojne cdo funksion nga inputet tek outputet, por duhet një mënyrë që rrjeti të gjeje parametrat më të pershtateshme duke u bazuar në të dhënrat e trajnimit dhe outputet e dhënrat (në rastin e të mesuarit të supervizuar).

Parametrat e modelit duhet të jepin gabimin minimum, për të marrë një parashikim sa më të mirë të të dhënave hyrese. Kete gabim e quajme si funksioni i humbjes (apo *loss function* sic gjendet në literatur), të parametrave,  $Q(\theta)$ , që duam të minimizojme në lidhje me parametrin  $\theta$ . Perderisa jo cdo problem në machine learning ka të njejtin qellim (psh. regres vs. klasifikim), kemi funksione të ndryshme që percaktojne gabimin e peshave në një moment të caktuar. Konsiderojme bashkësinë e të dhënave të trajnimit si  $(x_1, y_1), \dots, (x_n, y_n)$  ku  $x_i$  dhe  $y_i$  janë vektoret respektiv. Outputi  $y$  perafrohet nga rrjeti neural sipas  $\hat{y} = f(x; \theta)$  dhe parametri  $\theta$  i tille që  $Q(\theta)$  është minimum, falë algoritmeve të optimizimit si *gradient descent*, *adam*, etj. Gradienti<sup>2</sup> i funksionit të kostos, i quajtur dhe si gradient-i i gabimit, llogaritet për cdo shtresë, në mënyrë që të kutpohet se si ndryshimet në pesha në shtresën e inputeve afektojne funksionin e humbjes. Llogaritje e gradientit me metoda analitike është e thjeshtë, por metodat numerike janë mjaft të kushtueshme duke kerkuar shumë llogaritje, prandaj përdoret algoritmi i back-propagation meqë na jep një mënyrë më të thjeshtë dhe të lire.

#### 4.1.3.1 Algoritmi i back-propagation

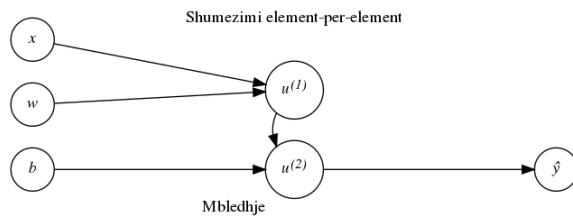
- **Forward pass:** Në rrjetat feedforward, informacioni rrjedh nga shtresa e parë, neper shtresat e fshehura, deri tek ato të fundit. Ky veprim vazhdon përgjatë të gjithë të dhënave testim, derisa të merret humbja (skalar ose vektor)  $Q(\theta)$ . Në mënyrë që të mund t'i modifikojme peshat e rrjetit, duhet ta kthejme mbërapsht gabimin në hapin që njihet si *backward pass*.
- **Backward pass:** Ndryshe i njohur dhe si *backprop*, lejon llogaritjen e gradientit. Backprop nuk është algoritem i të mesuarit, është thjeshtë metode llogaritje që lejon rrjetin të mesoje një algoritem optimizimi si *gradient descent* apo *adam*.

Për të kuptuar se si funksionon back-propagation, mund të paraqesim një rrjet neural në formen e një grafi, e tille që cdo nyje aplikon funksionin jolinear të perceptorit (Fig. 5.6)

Bakcpropagation është një algoritem i cili shpreh errorin e gradientit në lidhje me vlerat që i jepen një neuroni si funksion i outputeve të neuroneve që dalin nga

---

<sup>2</sup>Gradienti është derivati i funksioneve me shumë variabla



**Figura 4.6:** Grafi funksionit të regresit linear:  $\hat{y} = \sigma(w^\top x + b)$

ai. Kjo është e mundur falë rregullit zinxhir për llogaritjen e derivatit në funksionet kompozite, ku derivati i këtyre të fundit është i njojur. Psh. marrim numrin real \$x\$, dhe funksionet \$f\$ dhe \$g\$ të përcaktuar po në bashkësinë e numrave real. Marrim kompozimin e \$f\$ dhe \$g\$ duke perfshuar \$z = f(g(x))\$, rregulli mbi gjetjen e derivateve të funksioneve të përbërë thotë se:

$$\left(\frac{z}{x}\right)' = \left(\frac{z}{y}\right)' \left(\frac{y}{x}\right)' \quad (4.5)$$

Nga ky graf mund të shkruajm mbrapsht errorin duke përdorur derivatet e pjesëshme:

$$\left(\frac{Q(\theta)}{u_i}\right)' = \left(\frac{z_i}{u_i}\right)' \sum_j \left(\frac{Q(\theta)}{u_j}\right)' \left(\frac{u_j}{z_i}\right)' \quad (4.6)$$

ky është ekuacioni kur kthehem i mbrapsht pasi është llogaritur humbja në shtresën e fundit:

$$\left(\frac{Q(\theta)}{u_i}\right)' = g'(u_i) \sum_j \left(\frac{Q(\theta)}{u_j}\right)' w_{ij} \quad (4.7)$$

Më pas mund të llogarisim parametrat e gradientit si në vijim:

$$\frac{Q(\theta)'}{w'_{ij}} = \frac{Q(\theta)'}{u'_j} \frac{a'_j}{w'_{ij}} \text{ dhe } \frac{Q(\theta)'}{b'_{ij}} = \frac{Q(\theta)'}{u'_j} \frac{u'_j}{b'_j} \quad (4.8)$$

që na jep:

$$\frac{Q'(\theta)}{w'_{ij}} = \frac{Q'(\theta)}{u'_j} z_i \text{ dhe } \frac{Q'(\theta)}{b'_{ij}} = \frac{Q'(\theta)}{u'_j} \quad (4.9)$$

Implementimi i backprop mund të gjeneralizohet për cdo rrjet neural me një shtresë të fshehur (MLP), duke përdorur formen matricore të tij.

**Backpropagation në formë matricore** Po të konsiderojmë një MLP me parametra si mëposhtë:

$\mathbf{z}_l$  : vektori i neuroneve të aktivizimit në shtresën  $l$

$\mathbf{W}_{l,l+1}$  : matrica e peshave që lidhin shtresën  $l$  me shtresën  $l+1$  (4.10)

$\mathbf{b}_l$  : vektori i njesive bias që shtojmë në shtresën  $l$

$\mathbf{g}_i(\cdot)$  : funksioni që aplikon funksionin e aktivizimit në njesine  $i$

**Ekuacionet nga të dhënat e inputit në neuronet e outputit (Forward pass):**

$$\begin{aligned} u_{l+1} &= \mathbf{W}_{l,l+1}^\top \cdot \mathbf{z}_l + b_{l+1}^3 \\ g_{l+1} &= \mathbf{g}(\mathbf{u}_{l+1}) \end{aligned} \quad (4.11)$$

**Ekuacionet e algoritmit të kthimit mbroqshët:**

$$\begin{aligned} \frac{Q'(\theta)}{z'_l} &= \mathbf{W}_{l,l+1} \frac{Q'(\theta)}{u_l + 1'} \\ \frac{Q'(\theta)}{u'_l} &= \mathbf{g}'(\mathbf{u}_l) \odot \frac{Q'(\theta)}{z'_l} \end{aligned} \quad (4.12)$$

**Perditesimi i peshave:**

$$\begin{aligned} \frac{Q'(\theta)}{\mathbf{W}'_{l,l+1}} &= z_1 \frac{Q'(\theta)^\top}{u'_{l+1}} \\ \frac{Q'(\theta)}{b'_l} &= \frac{Q'(\theta)}{u'_l} \end{aligned} \quad (4.13)$$

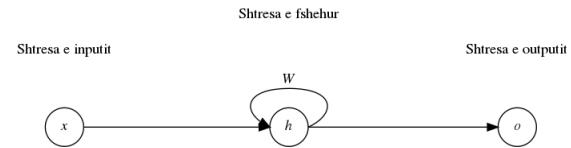
## 4.2 Rrjetat neural me përsëritje

Me lartë, u konsideruan rrjetat MLP pa lidhje ciklike, gjë që i ben ata një model statik ku ciftet e input-outputit janë të pavarur. Kjo skemë nuk është eficiente në problemet ku duhet të bazohemi apo parashikojme të dhëna sekuencore, dhe atributet janë të nderlidhur. Nëse lejojme këto lidhje ciklike, atëherë marrim një *rrjet neural me përsëritje* (RNN), i cili mund të modeloje procese dinamike, si psh. serite kohore. Dhe është e vertete se kur duam të modelojme një proces të tille, na duhet një model i cili paraqet varesite mes të shkuarës dhe të ardhmes, dhe outputi është një funksion i outputit të perparshem. Pra modeli duhet të perpunoje shembujt një nga një dhe

të ruaj ato të meparshmet në një memorje që përfaqëson kontekstin dhe mund të përdoret në hapin e rradhës. Falë këtij informacioni të perseritur, duke e krahasuar me MLP, një rrjet me përsëritje ndan të njejtat pesha përgjatë disa hapave [27]. RNN përdoren në vijim për parashikimin e cmimit të Bitcoin.

#### 4.2.1 Arkitekturat

Arkitekturen e një RNN mund ta mendojme si një superset të rrjetave neural pa përsëritje (*feedforward*), meqë permbajne një apo me shumë cikle. Secili cikel bën të mundur që cfare del nga një neuron të kthehet prap tek ai, duke lejuar *feedback information*. Keto cikle, të lidhjeve të perseritura, lejojnë që neuronet e fshehura të shohin outputin e mëparshëm të tyre, pra sherbejnë si mekanizem memorje dhe na paraqesin nocionin e kohes në model. Neuronet me përsëritje shpesh njihen si neurone me kontekst apo neurone me gjendje. Struktura e një rrjeti të thjeshtë të tille është si mëposhtë (Fig. 5.6):

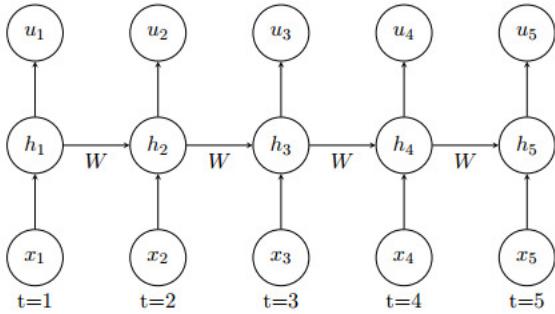


**Figura 4.7:** Rrjeti më i thjeshtë me përsëritje, ku ka vetëm një shtresë të fshehur

Informacioni në output formohet nga inputi dhe neuronet që kanë akumuluar memorie përgjatë njesive të fshehura. Së fundmi, lidhjet me përsëritje, nga njesite me gjendje për tek vetvetja dhe nga njesite e outputit tek njesite me memorie lejojnë që informacioni që hyn njehere në rrjet të kthehet preseri mbapsht përgjatë 1 timestep. Duke supozuar se kemi një funksion aktivizimi  $g(\cdot)$ , ekuacionet për figuren me lartë do të ishin:

$$\begin{aligned} h^{(t)} &= g_h(W_1 x^{(t-1)} + b_h) \\ o^{(t)} &= g_o(W_o h^{(t)} + b_o) \end{aligned} \tag{4.14}$$

Një mënyrë shumë përmirë e vizualizuar do të ishte shpalosja e neuronit që perfshin cikel, atëherë rrjeti neural do dukej si në Fig : Gjithësesi, një veshtirësi që lind me këtë arkitekturë është zvogelimi i madh i gradientit, ndërkohë që zbatohet



**Figura 4.8:** Shpalosja e rrjetit neural me përsëritje, ku një input varet nga 5 inputet e mëparshme, pra *timestep*-i është 5 Rreshti i parë i neuroneve përfaqëson shtresën e inputeve, i dyti shtresën e fshehur, dhe në fund shtresa e outputit.

algoritmi i backpropagation. Kete problem e quajme *vanishing gradient*, e anasjellta e të cilit është *exploding gradient*, ku peshat pesojne shumë rritje. Në rastin e zhdukjes se gradientit, peshat konvergjojne drejt zeros, kështu që edhe pse algoritmi backprop, vazhdon të trajnoje modelin peshat nuk po permiresohen. Në vitin 1997, Jürgen Schmidhuber se bashku me Sepp Hochreiter [29], i dhane zgjidhje këtij problemi përmes krijimit të arkitektures LSTM. Ata e testuan deri me një *timestep* prej 100 rekorde, prandaj shpesh për LSTM nuk keshillohet të konsiderohen me shumë se kaq rekorde.

### 4.3 LSTM

Në këtë pjesë të kapitullit fokusohemi në një nga modelet më të përdorura për të zgjidhur problemet e RNN në lidhje me varesite që kanë rekordet e sekuençave, *njesite LSTM*. Arsyja pse zgjidhet LSTM për parashikimin e cmimit të Bitcoin, janë avantazhet që ai ka kundrejt RNN. Koncepti pas LSTM është krijimi i lidhjeve përgjatë kohes, ku errori të qendroje konstant në mënyrë që gradienti të mos rritet apo zvogëlohet shumë. Rrjetat LSTM janë krijuar specifisht për të eleminuar problemin e varesive në sekuençat e gjata. Krijimi i kujteses për informacion është sjellja e tyre normale, prandaj konsiderohen dhe si modeli më i perdorshem për mesimin e sekuençave.

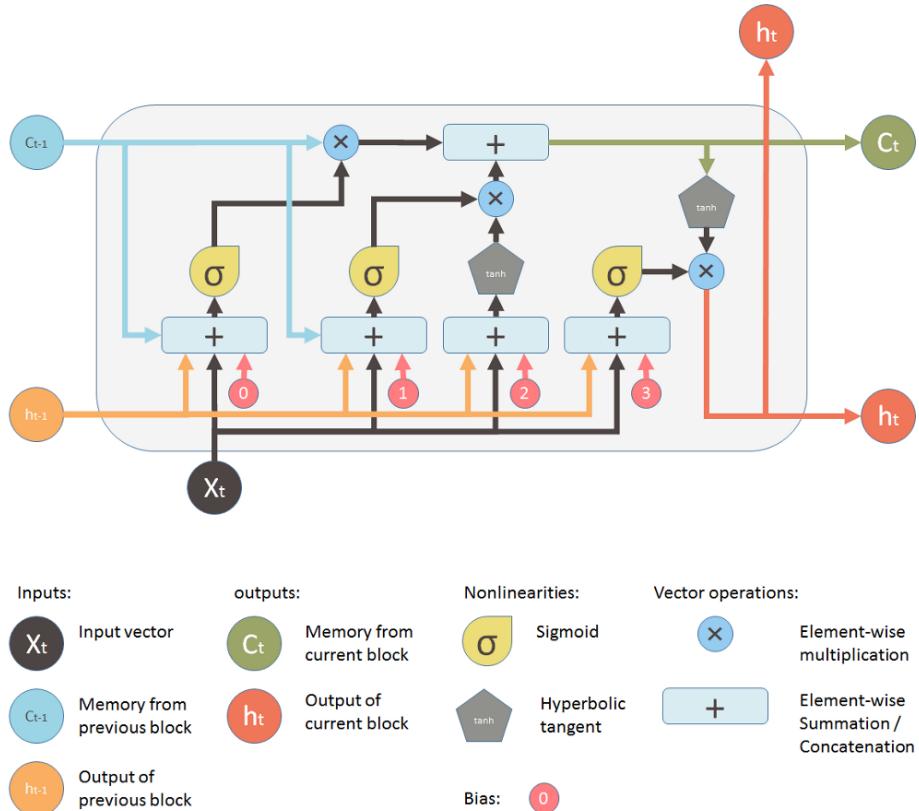
### 4.3.1 Arkitektura e LSTM

Arkitektura e LSTM është e ngjashme me RNN, dhe e konsiderojmë si një superset të RNN, njëloj sikurse RNN e mendonim superset të MLP. LSTM janë bloqe memoriesh (prandaj njihen dhe si *gated networks*). Blloku më i thjeshtë ka tre shtresa rrjetash neural, që nderveprojne me njera-tjetren, perkundrejt një shtresë që ndodhet në RNN:

- Kane një ose me shumë qeliza kujtese  $s_c$ , këto perbejne dhe gjendjen e qelizes (*cell state*). Kjo qelize prodhon vetëm disa transformime të vogla lineare, duke arritur kështu një error pothuajse konstant në bllokun e memories. Është një njesi lineare, me një cikl tek vetja. Qelizat e tjera mblidhen me këtë qelize.
- Një njesi inputi e cila mbron memorien e  $s_c$
- Një njesi outputi për të mbrojtur neuronet e aferta nga të dhënrat e memories që nuk janë shumë të nevojshme

Keto njesi i jatin LSTM-se aftesine për të kontrrolluar rrjedhen e informacionit në qelize dhe kanë perbehen nga funksioni i aktivizimit sigmoid, i cili vendos mbi sasine e informacionit që do të kaloje në këto njesi. Ato janë të myllura kur aktivizimi është afer zeros, dhe hapen kur aktivizimi është afer 1. Si rrjedhojë, njesia e inputit vendos kur ti mbishkruaj apo jo të dhënrat në qelizen e memories. Me këtë arkitekturë, qeliza që mban gjendjen,  $s_c$ , modifikohet në baze të gjendjes që ndodhet rrjeti gjatë trajnimit, dhe tre burime inputi:  $net_c$ , inputi në qelize vjen nga lidhja ciklike,  $net_{in}$  dhe  $net_{out}$  janë njesite e inputeve dhe outputeve [30]. Në cdo hap, gjatë propagimit të të dhënavë të inputit, të gjitha njesite perditesohen dhe sinjalët e errorit për të gjithë peshat llogariten gjatë ekzekutimit të algoritmit backpropagation.

Në pamje të parë LSTM duket shumë e komplikuar, por në nuk do e trajtojme si blackbox, kështu që në vijim do të shpjegojme ekuacionet e njesive perberese. Në pjesën e siperme të figures 5.10 (bija nga  $C_{t-1}$  deri në  $C_t$ ) formojne qelizen që mban gjendjen e LSTM-se, ku  $C_{t-1}$  (vektor) përbën memorien e vjeter. Operacioni i parë  $X$  neper të cilin kalon është njesia e harreses (*forget gate*). Kështu që nëse shumezojmë memorien e vjeter  $C_{t-1}$  me një vektor që është pothuajse zero, atëherë do të thotë se duhet të harrojme memorien e vjeter. Në të kundërt, kur vektori është pothuajse 1, e lejojme të kaloje përmes njesise se qelizes. Kur kalon ketu, informacioni i memories se vjeter bashkohet me atë të memories se re. Sasia e informacionit që kalon vendoset



**Figura 4.9:** Pamje e detajuar e njesise LSTM, se bashku me qelizen e gjendjes dhe njesite e perditesimit, outputit dhe inputit [31]

nga operacioni  $X$  që ndodhet poshte  $+$ . Pas këtyre dy veprimeve, memoria e vjeter  $C_{t-1}$  ka ndryshuar dhe është kthyer në memorien e re  $C_t$ .

Me tej, hapi i parë i LSTM-se është të vendose se cfare informacioni të mbaje në memorie dhe cfare jo. Ky vendim merret nga shtresa e parë sigmoid, që është *njesia e harreses* (Ek. 5.15). Ajo sheh  $h_{t-1}$  dhe  $x_t$ , dhe kthen një numër mes 0 dhe 1 për cdo numër në qelizen e gjendjes  $C_{t-1}$ . Hapi i rradhës është marrja e vendimit se cfare informacioni do të ruajme në qelizen e gjendjes. Këtu ka dy hapa, fillimisht shtreses se sigmoid të quajtur *njesia e inputit* (Ek. 5.16), do i perditesohen vlerat. Së dyti, shtresa me funksion aktivizimi *tanh* do të krijoje një vektor të ri vlerash kandidat,  $\tilde{C}_t$ , (Ek. 5.17) që do t'i shtohet gjendjes. Në hapin e ardhshem, do të kombinohen këto të dyja për të krijuar një perditesim të gjendjes. Pra zevendesohet ajo cka u harrua nga njesia e harreses (Ek. 5.18). Së fundmi duhet të vendosim mbi

outputin. Ky output do të bazohet në gjendjen e qelizes tone, por duhet që fillimisht të aplikojmë një transformim. Fillimisht, do të aplikojmë një shtresë sigmoid, e cila vendos mbi pjeset që e qelizes që do të jene pjesë e outputit. Më pas, aplikohet **tanh** dhe shumezohet outputi me një njesi sigmoid, në mënyrë që të kthehen vetëm pjeset për të cilat vendosim (Ek. 5.19). Outputi perfaqesohet nga Ekuacioni 5.20 [32].

$$f_t = \sigma(W_f S_{t-1} + W_f S_t) \quad (4.15)$$

$$i_t = \sigma(W_i S_{t-1} + W_f S_t) \quad (4.16)$$

$$\tilde{C} = \tanh(W_c S_{t-1} + W_c X_t) \quad (4.17)$$

$$c_t = (i_t * \tilde{C}_t) + (f_t * c_{t-1}) \quad (4.18)$$

$$o_t = \sigma(W_o S_{t-1} + W_f S_t) \quad (4.19)$$

$$h_t = o_t * \tanh(c_t) \quad (4.20)$$

## Kapitulli 5

# Parashikimi i Serive Kohore

Parashikimi është shumë i veshtire,  
sidomos kur flitet për të ardhmen

---

Niels Bohr

Parashikimi i të ardhmes ka qenë dhe është pjesë e njerëzimit. Ekonomia është një shkence sociale, si rrjedhojë shumë parashikime nuk janë të sakta, pasi saktësia nuk ka një perkufizim, të dhënat jo gjithnjë janë të pastra, shumë variabla varen nga vetë e ardhmja, etj. të gjitha këto veshtirsojne analizen për të krijuar një model që mund të mendohet si i saktë. Por të pakten sot jemi më të zgjuar dhe e dimë se nuk mund të shprehesh kurrë bindje të plote mbi një event që ndodh në të ardhmen. Ndryshe nga sot, njerëzit më parë kanë bërë plot parashikime të cilat koha i vertetoi totalisht të gabuara:

- *Një PC nuk do këtë kurrë nevojë për me shumë se 637KB. 640KB do jetë e mjaftueshme për kedo* - Bill Gates
- *Telefonia celular nuk do të zevendesoj kurrë sistemin telefonik* - Marty Cooper
- *Interneti do pesoje një boom dhe në 1996 do shkaterrohet plotësisht* - Robert Metcalfe
- *Nuk ka mbetur asgje për t'u zbuluar në Fizike, pervec se të kryhen matje më të sakta* - Lord Kelvin

Për të ndertuar produkte dhe shërbime të suksesshme duhet që të kemi një ide para-prake të se ardhmes. Ka fenomene që mund të parashikohen thjeshtë, dhe ka të tjera

për të cilat është shumë shumë e veshtire të jepet një mendim, kjo pasi parashikimi i një variabli apo eventi varet nga disa faktore si psh:

- Sa mirë njihen faktorët që ndikojnë mbi të?
- Sa të dhëna kemi në dispozicion?
- A varet parashikimi nga parashikimi që po bëjmë?

Për shembull, kërkesa për elektricitet mund të parashikohet me saktesi të lartë pasi të tre kushtet e siperme plotësohen, kërkesa për elektricitet percaktohet pjeserisht nga temperaturat, sa me shumë të dhëna për kërkesën e elektricitetin dhe kushtet atmosferike, aq më të lehte e kemi parashikimin e tij.

Nga ana tjeter, parashikimi i cmimit ekonomik, vetëm kushti i të dhëna të disponueshme plotësohet. Por jo vetëm kaq, sepse historiku i të dhënave të cmimit pothuajse asnjehere nuk është i mjaftueshem për të modeluar një parashikim të sukseshem. Për me tepër, pothuajse gjithmonë, të dhënat nuk janë stacionare<sup>1</sup>, sidomos në rastin e Bitcoin (Fig. 4.2). Tjeter faktor i paevitueshem, është se ekonomia është e tille



**Figura 5.1:** Grafiku i cmimit të Bitcoin, dhe ndryshimet e thella në cmim

që prodhimi qarkullon dhe cmimi leviz. Sic pame dhe në modelin e thjeshtezuar të tregut të Bitcoin-it, edhe sikur të parashikojme saktë një rritje në cmim, njerëzit do vepronin duke e ulur cmimin. Kjo na tregon dy gjera:

---

<sup>1</sup>Një sekuence është stacionare nëse vecorite statistikore si mesatarja, varianca, autokorrelacioni etj. nuk ndryshojne përgjatë kohes

- Cmimi i të ardhmes varet (pjeserisht) nga e ardhmjë
- Mund të jemi më të sigurt për parashikimin afatshkurtër të cmimit se sa atë afatgjate

Dhe në fakt, persa i perket pikes se dyte, është dhe një ndër arsyet <sup>2</sup> se pse high frequency forecasting ka marrë kaq hov. Ndersa për pikën e parë, mund të themi se në një fare mënyrë, parashikimi ndikon në vetë cmimin, sidomos kur parashikimet publikohen para një audience të gjere. Ky është një shembull i *Efficient Market Hypothesis*. Gjithësesi në mendojme se nëse faktorët që merren në konsideratë luajne një rol mjaft të rëndësishëm dhe mund të bëjnë diferençen, si dhe perfshirja e variabave të se ardhmes, është shpesh e nevojshme për rezultate më të mira.

Lloje parashikimi ka plot, dhe variojnë nga më i thjeshti tek ai me kompleksi, duke filluar nga metodat naive që konsiderojnë vetëm rekordin e mëparshëm, deri tek sistemet ekonometrike dhe rrjetat neurale. Zgjedhja e metodes për parashikim varet nga të dhënat e disponueshme dhe tipi i variablit që to parashikohet (vlera, klase, etc.) Një tjetër klasifikim që mund t'i bëjmë parashikimiit në varësi të sasise se të dhënavë që kemi është në atë kualitative dhe kuantitativ. Parashikimi kuantitativ mund të kryhet nëse 2 kushtet e mëposhtme plotësohen:

1. Informacioni i të shkuarës është në formë numerike (ose mund të transformohet në të tille)
2. Është e arsyeshme të mendojme se një pjesë e trendit do të ruhet edhe në të ardhmen (psh. mund ta themi se në cmimin e Bitcoin do të këtë perseri levizje të thella cmimi, meqë sasia e Bitcoin-eve që marrin miners sa vjen e ulet)

## 5.1 Përpunimi dhe marrja e të dhënavës për rrjetat neural

### 5.1.1 Ekstraktimi i të dhënavës

<sup>3</sup> Nga analiza ekonomike dhe teknike e Bitcoin, është arritur në perfundimin se disa të dhëna fundamente që i kemi përfshirë si input për modelin janë:

---

<sup>2</sup>Krahas zhvillimeve në Machine Learning dhe rritjes se fuqise se Hardware-it

<sup>3</sup>të gjithë të dhënat merren në interval ditor

- të dhënat historike të cmimit** - Përdoret API publike e Coinmarketcap. Arsyja pse merret ky website është pasi ai na jep një mesatare të cmimit të exchange-eve të Bitcoin. të dhënat që merren janë të dhënat tipike për stoket në exchange në interval ditor B.2:
  - Cmimi i hapjes (*Open*)
  - Cmimi i mbylljes (*Close*)
  - Cmimi më i lartë (*High*)
  - Cmimi më i ulet (*Low*)
  - Volumi (*Volume*)
  - Market Capitalization (*Market Cap*)
- të dhëna nga Blockchain-i i Bitcoin** - Perdoren dy API publike statistikor, ajo e website-it Blockchain, dhe ajo e Quandl. të dhënat që marrim nga to janë:
  - Fitimi i miners
  - Numri i transaksioneve të konfirmuara
  - Madhësia mesatare e bllokut
  - Numri i llogarive në rrjet
- të dhëna sentimenti nga Google Trends** - Përdoret libraria PyTrends për të marrë shkallën e interesit (*Interest over time*) për kerkimet që permajne fjalen "bitcoin".
- të dhëna makroekonomike si indekset financiare** <sup>4</sup> - Përdoret API financiare e Yahoo. Indekset që kemi konsideruar janë:
  - Standard & Poor's 500 (S&P 500) <sup>5</sup>
  - Dow and Jones Industrial Average (DIJA) <sup>6</sup>

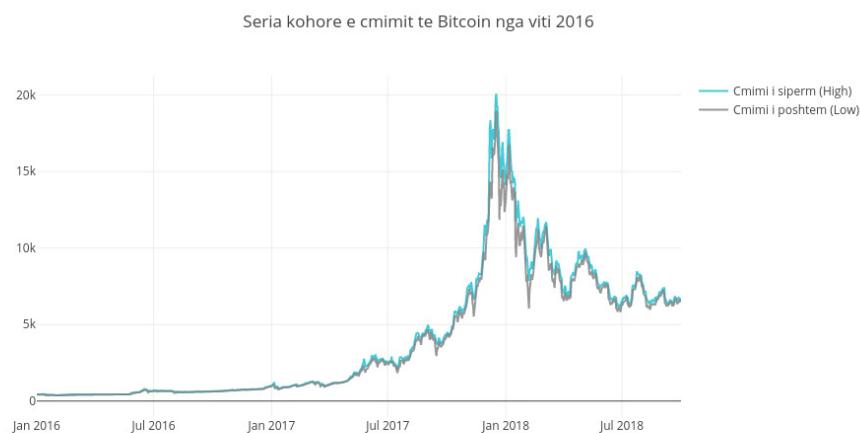
<sup>4</sup>Në përgjithësi, indeksi i referohet një indikatori statistikor që mat ndryshimet në tregun e instrumentave financiar të tregtueshem (stoket janë një klase e këtyre instrumentave)

<sup>5</sup>S&P 500 është një indeks për tregun e stokeve i përbërë nga 500 kompanite që zotërojnë stoke në New York Stock Exchange (NYSE) ose Nasdaq

<sup>6</sup>Dow Jones Industrial Average është një ndër indekset më të vjeter, dhe perbehet nga 30 kompanite më të mëdha në Sh.B.A.

### 5.1.2 Përpunimi i të dhënavë

Gjatë përpunimit të dhënat do të transformohen në një format të përshtatshëm për për algoritmet e të mesuarit të supervizuar, dhe me konkretisht për rrjetin neural LSTM. Disa hapa perfshijne: kthimin e të dhënavë në vektor, zevendesimin e të vlerave që mungojne, normalizimi. të dhënat që konsiderojmë merren nga viti 2016 deri në fund të shtatorit të 2018 (Fig. ), dhe na ngelet tes kemi gjithesej 1008 dite të serive kohore të permendura me lartë. Mbi të dhënat kryhen përpunime të



**Figura 5.2:** Bashkesia e të dhënavë për parashikimin e cmimit, ketu paraqitet vetëm cmimet e exchange-it të Bitcoin

tipit, ndryshim i emrit të fushave, në varësi të datasetit që i perkasin. Psh. kolonat e të dhënavë të blockchain paraprihen nga parashtesa *bch*, ato të cmimit historik (exchange-it) kanë parashtesen *btc*, të dhënat e sentimentit nga *google\_trends*, si dhe DIJA me S&P paraprihen nga *dj* dhe *sp* respektivisht. Në Figuren 5.3 tregojme si nderlidhen këto variabla duke përdorur korrelacionin e Pearson, ku ajo që bie në sy është koeficenti i lartë që ekziston mes të dhënavë të sentimentit me numrin e transaksioneve, e volumin. Apo se si transaktionet lidhen me madhesine e bllokut.

të dhënat pasi mblidhen vec e vec bashkohen përgjatë dimensionit të kohes. Ato ruhen në Dataframe në Pandas.

	btc_high	btc_close	btc_volume	btc_market_cap	bch_avg_block_size	bch_transactions	bch_mining_revenue	bch_accounts	sp_close	dj_close	google_trends_bitcoin
btc_high	1	0.92	0.37	0.95	0.11	0.074	0.37	-0.038	-0.089	-0.072	0.53
btc_close	0.92	1	0.31	0.89	0.13	0.074	0.37	-0.068	-0.1	-0.084	0.33
btc_volume	0.37	0.31	1	0.32	0.12	0.099	0.19	0.065	-0.096	-0.094	0.37
btc_market_cap	0.95	0.89	0.32	1	0.11	0.078	0.43	0.017	-0.11	-0.09	0.41
bch_avg_block_size	0.11	0.13	0.12	0.11	1	0.65	0.1	0.029	-0.088	-0.084	0.14
bch_transactions	0.074	0.074	0.099	0.078	0.65	1	0.084	0.12	-0.092	-0.1	0.23
bch_mining_revenue	0.37	0.37	0.19	0.43	0.1	0.084	1	0.2	-0.21	-0.16	0.17
bch_accounts	-0.038	-0.068	0.065	0.017	0.029	0.12	0.2	1	-0.085	-0.018	0.032
sp_close	-0.089	-0.1	-0.096	-0.11	-0.088	-0.092	-0.21	-0.085	1	0.95	-0.043
dj_close	-0.072	-0.084	-0.094	-0.09	-0.084	-0.1	-0.18	-0.018	0.95	1	-0.045
google_trends_bitcoin	0.53	0.33	0.37	0.41	0.14	0.23	0.17	0.032	-0.043	-0.045	1

**Figura 5.3:** Korrelacioni i Pearson për atributet e datasetit. Vlera 1 është maksimumi

### 5.1.3 Normalizimi të dhënavë

Rrjetat neural janë shumë të ndjeshëm ndaj ndryshimeve të konsiderueshme në vlerat e të dhënavë, si rrjedhojë vendosja e të dhënavë në një interval të vogel (në përgjithësi nga 0 në 1) dhe uniform (të variojnë në pothuajse të njejtin interval) është proces që kryhet pothuajse gjithmonë [33]. Në machine learning ekzistojnë tipe të ndryshme të normalizimit. Disa janë:

- **Shkallezim sipas vlerave minimum dhe maksimum (Min-Max Scaling),** të dhënat e inputit variojnë nga 0 në 1:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \quad (5.1)$$

- **Normalizim sipas mesatares,** të dhënat marrin vlera nga -1 në 1, në përgjithësi, kur përdoret ky lloj normalizimi funksioni i aktivizimit do jete  $\tanh$ :

$$x' = \frac{x - \bar{X}}{\max(X) - \min(X)} \quad (5.2)$$

- **Standartizim (z-Score),** ku atributet rishperndohen e tille që mesatarja të jetë zero dhe devijimi standart 1. Një kusht për aplikimin e tij është se dataset-i duhet të shfaq një shpërndarje Gaussiane:

$$x' = \frac{x - \bar{X}}{\sigma} \quad (5.3)$$

Nga këto mënyra, është zgjedhur normalizimi min-max, i cili kryhet nga moduli `sklearn.preprocessing` i scikit-learn.

## 5.2 Përshtatja e të dhënave të serive kohore për rrjetin neural

Serite kohore janë sekuencia vlerash numerike përgjatë kohes. Rrjeti neural LSTM për parashikimin e serive kohore sillet si një algoritem machine learning i supervizuar. Kështu që, të dhënat duhet të ndahen në të dhëna inputi dhe outputi. Për parashikimin e cmimit të Bitcoin, është parë e arsyeshme të parashikohen cmimet për 30 ditet e ardhshme. Fillimisht duhet të vendoset se në sa dite të mëparshme do të bazohet parashikimi, nga testimet e shumta u arrit në rezultatin se 32 rekorde të mëparshme janë të mjaftueshme, duke qenë se cmimi Bitcoin-it ka shumë luhatje të thella, dhe rritja e drifteve që parashikon të ardhmen, e ul saktesine. Si rrjedhojë inputi për LSTM-në do të jete një tensor i përbërë nga matrica me dimesion 32x11 (meqë kemi 11 atribut). Një tjetër arsyë për të zgjedhur këtë madhesi drifteje është pasi po të zgjdedhim më pak të dhëna të shkuara algoritmi nuk arrin të detektoje trendet e cmimit që mund të vihen re vetëm në sekuencia të gjata. Outputi i rrjetit LSTM do të jete perseri një tensor, i përbërë nga matrica me gjatesi 30x11, meqë parashikojme 30 ditet në vijim.

### 5.2.1 Ndarja e të dhënave në bashkësi trajnimi dhe testimi

Ndarja e bashkesise se të dhënave totale është një ndër hapat më të rendësishëm, në fillim u provua të parashikohej i gjithë viti 2018, por trendi i krijuar në 2017, ku cmimi ka një rritje jashtezakonisht të vrullshme beri që nga fillimi i 2018 ky cmim të biente, natyrisht kjo është dicka që algoritmi yne nuk ka si ta parashikoje, kështu që parashikimi 1 vjecar u la menjeane. Për momentin është zgjedhur data 2018-04-25, dhe kemi 845 rekorde për trajnimi kundrejt 163 të dhenash për testim. Ky raport korrespondon perafersisht me një ndarje 80% me 20% të dhenash trajnimi dhe testimi respektivisht (Fig. ).

#### 5.2.1.1 Ndarja të dhënave në inpute dhe outpute

Sic u tha dhe me lartë, ndryshe nga datasetet më të zakonshme, serite kohore nuk i kanë të paracaktuara outputet (pra varet nga ne sa me përpara në kohe duam të parashikojme). Bashkesia e trajnimit ndahet në 2 vektor numpy, njeri që mban



**Figura 5.4:** Ndarja bashkesise se të dhënave fillestare

inputet dhe tjetri për outputet korresponduese (e njëjtë procedure kryhet edhe për të dhënat e testimit). Pra do të kemi bashkësitet e mëposhtme:

1. *training\_inputs* - dimesione (813, 32, 11)
2. *training\_outputs* - dimesione (783, 30)
3. *test\_inputs* - dimesione (131, 32, 11)
4. *test\_outputs* - dimensione (101, 30)

## 5.2.2 Modeli LSTM

Gjatë fazes se trajnimit janë provuar konfigurime të ndryshme të LSTM-se, më poshtë do të permendim se cilat ishin gjetjet mbi pjesën e dyte më të rëndësishme pas perzgjedhjes se të dhënave të inputit, që është ajo e zgjedhjes se hiperparametrave <sup>7</sup>.

### 5.2.2.1 Hiperparametrat e modelit

---

<sup>7</sup>Një hiperparameter (*hyperparameter*) është një parameter i algoritmit dhe jo i modelit, pra nuk ndikohet nga algoritmi i të mesuarit sic bëjnë peshat. Gjithashtu ato janë konstant gjatë trajnimit.

- Numri i neuroneve në shtresat e fshehura:** Është vendosur të përdoren 100 neurone në shtresat e fshehura, ndonse janë provuar konfigurime me 20, dhe 50. Sa më i madh të jetë ky numër aq më i ngadalte është trajnimit, dhe duket qarte që duhet bërë një tradeoff mes kohes dhe saktësisë. Meqë të dhënë historike të Bitcoin nuk ishin në rang milionash, u pa e arsyeshme se 100 neurone nuk ndikojnë aq keq.
- Funksioni aktivizimit:** Duke marrë parasysh faktin që të dhënët e normalizuar i kemi nga 0 në 1, funksioni *sigmoid* mund të mendohet si default, por ReLU (Rectified linear units) është një funksion tjetër aktivizimi i cili dha rezultate më të mira se sigmoid. ReLU llogarit një funksion linear të inputeve dhe nëse rezulati është më i madh se zero atëherë kthen këtë kombinim në të kundërt kthen 0 9 (Ekuacioni. 5.4) [33]. Në rastin konkret, sigmoid dha një error prej 0.23 ndërkohë që ReLU prej 0.2.

$$h_{\mathbf{w},b}(X) = \max(X \cdot \mathbf{w} + b, 0) \quad (5.4)$$

- Optimizuesi:** Pasi algoritmi i backpropagation llogarit gradientet e funksionit të humbjes (MAE) në lidhje me cdo parameter të rrjetit, duhet që këto të fundit të modifikohen. Kjo detyrë kryhet nga një funksion optimizimi.

Ndër më të perdorurit është gradient descent ose versione të tij - stochastic gradient descent apo mini-batch gradient descent. Disavantazhi i tyre është ngadalesia për të konvergjuar në një zgjedhje të kenaqshme. Teknikat e quajtura ***momentum optimization***<sup>8</sup>, janë shumë më të shpejtë se cilido nga tipet e gradient descent, një nga keta optimizues është Adam (Adaptive Moment Estimation), i cili llogarit *shkallet adaptive të mesuarit* për secilin parameter. Meqë impulsi në fizike është  $p = mv$ , ketu mund të mendojme se rolin e mases e luan mesatarja dhe atë të shpejtesise varianca [34]. Adam, ruan një mesatare të gradienteve të mëparshëm pervec impulsit. Variante të tjera që u provuan janë RMSProp, AdaGrad, dhe Nadam. të gjithë përdorin iden e impulsit, por adam është deri diku default pasi në shumë pune kërkimore ka rezultuar si më performanti.

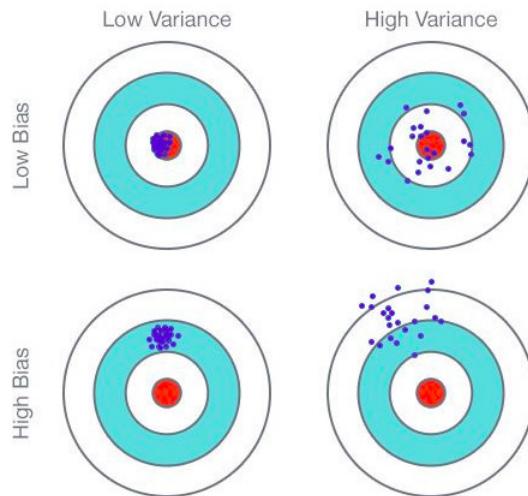
---

<sup>8</sup>Emri momentum (impuls në shqip), analog me termin ne fizikë, ku një gradient negativ është një force që leviz një grimce përgjatë hapesires se parametrave [27].

4. **Funksioni i humbjes:** Si funksion humbje është përdorur mesatarja e errorit absolut ose shkurt MAE (Mean Absolute Error) (Ekuacioni 5.5).

$$MAE = \frac{1}{n} \sum_{i=1}^n |(h(\mathbf{x}^i) - y^i)| \quad (5.5)$$

5. ***Dropout rate:*** Dropout është një metode *regularizimi*. Regularizimi ndihmon modelin të gjeneralizoje (ul mundesite për overfitting). Zakonisht shtresa e Dropout nuk vendoset në fund pasi rrjeti nuk ka me mundesi të korrigjoje erroret e dropout në shtresat e mëparshme. Duke u nisur nga problemi i tradeoff-it mes bias dhe variances, regularizimin mund ta mendojme si një mënyrë e cila ndodhet në mes të të dyjave. Në cdo hap të trajnimit, secili neuron (duke përfshirë neuronet e inputit por pa ato të outputit) kanë një probabilitet  $p$  për t'u hequr (behen drop out), pra injorohen totalisht gjatë trajnimit, por në hapin pasardhes neuroni mund të jete aktiv. Hiperparametri  $p$  quhet *dropout rate*. Regularizimi ndodh vetëm gjatë trajnimit [33]. Pasi janë provuar vlera të ndryshme për dropout-in është zgjedhur të përdoret një vlere prej 0.25.



**Figura 5.5:** Bias-variance (underfitting vs overfitting), në kontekstin e regularizimit, duam dicka midis.

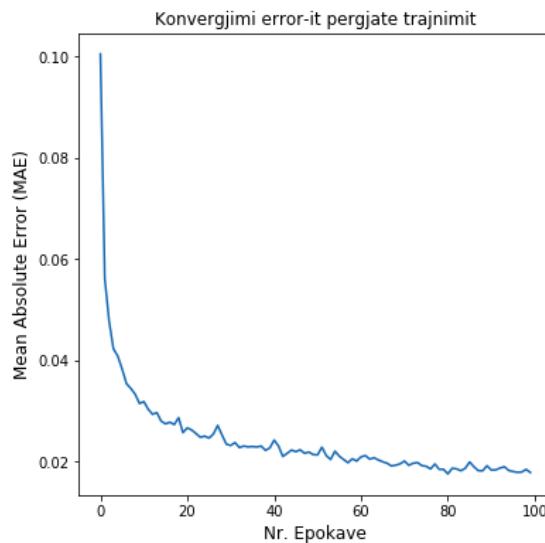
6. **Numri i epokave:** Nga testimet e kryera arritëm në rezultatin se një numër epokash prej 100 është i mjaftueshem, si dhe kur tentonim me me shumë (psh.

200 epoka), errori zmadhohej konsiderueshëm.

7. **Madhësia e batch-eve:** Duke qenëse algoritmi i merr një nga një inputet (në batch-e) *batch size* është një tjetër hiperparameter i rëndësishëm. Në këtë rast, *batch size* është lënë njëloj sa gjatësia e dritares, pra 32.

### 5.3 Rezultate

Rezultatet e trajnimit dhe testimit jepen më poshtë:



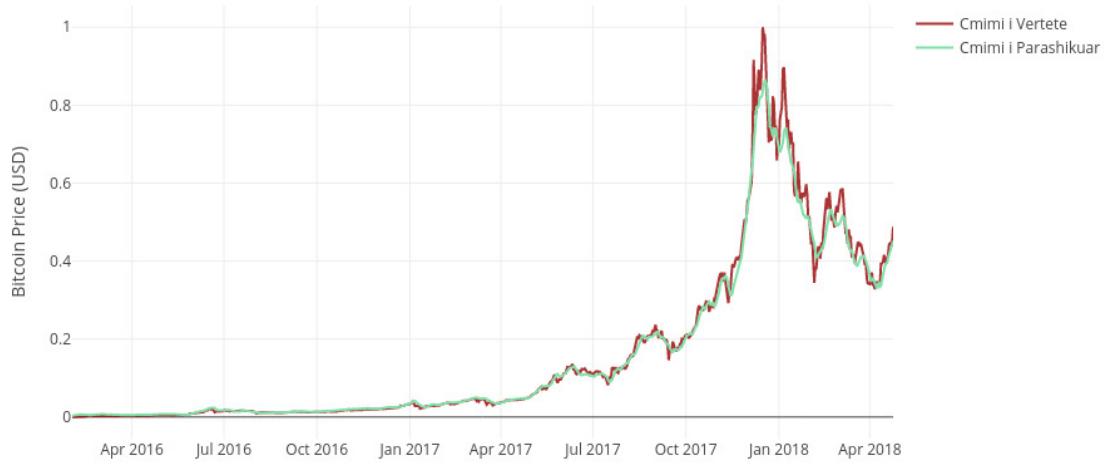
**Figura 5.6:** Grafiku që tregon se si errori zgjedhet gjatë trajnimit të modelit

**Permbledhje e modelit :** Dhe pse hiperparametrat i përshkruam më lartë, tanë do japim një permbledhje të modelit perfundimtar.

1. **Numri i shtresave:** 4

- Shtresa LSTM
- Shtresa Dropout
- Shtresa Dense
- Shtresa Outputit

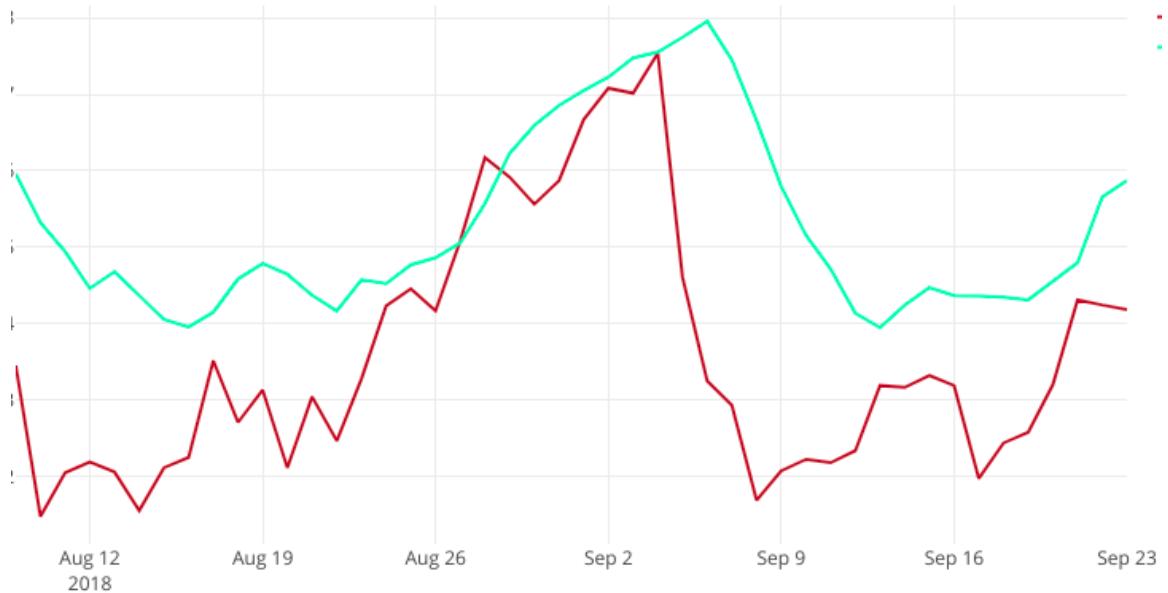
Parashikimi ne Bashkesine e Trajnimit, MAE: 0.0143



**Figura 5.7:** Parashikimi në bashkësinë e trajnimit dhe errorri. Parashikimi është për 30 dite



**Figura 5.8:** Parashikimi në bashkësinë e testimit dhe errorri. Parashikimi është për 30 dite



**Figura 5.9:** Parashikimi në bashkësinë e testimit. Parashikimi është për 60 ditet e fundit

2. **Numri neuroneve:** Në shtresat LSTM dhe Dropout numri i neuroneve vendlodset nga ne dhe sic u tha ai është 100, në shtresa Dense dhe atë të outputit ai është 30 ose 60 (kur parashikimi është për 30 ose 60 ditet në vazhdim)
3. **Funksioni i humbjes:** Mesatarja gabimit absolut *mean absolute error* (MSE)
4. **Funksioni aktivizimit për shtresat e fshehura:** ReLU (*Rectified linear units*)
5. **Optimizuesi:** Adam
6. **Dropout rate:** 0.25

# Kapitulli 6

## Konkluzione

Gjatë tezës shtruam problemin e parashikimit të cmimit të Bitcoin përmes rrjetit neural Long Short-Term Memory (LSTM). U përshkrua algoritmi i backpropagation, vecoritë e funksioneve jo-linear, si dhe njësite e rrjetit LSTM dhe perdonimi i tij në probleme që shprehen në formë sekuencore.

Rezultati i implementimit të parashikimit, na tregon se tregu i Cryptocurrency-ve është mjaft dinamik, si rrjedhojë, parashikimet afatgjata janë jo-realiste. Gjithësesi, testimi i hiperparametrave të ndryshëm dha rezultat të mirë për 30 ditët që u parashikan në rastin konkret. Kështu që vlerësimi i parametrave të ndryshëm përmës hyperparameter tuning dhe perdonimi i rrjetave me përsëritje (si LSTM) janë dy udhëzime për parashikimin e sekuencave me volatilitet të lartë.

U pa gjithashtu se faktorët e sentimentit së bashku më të dhëna nga blockchain ishin të nevojshme duke konsideruar popullaritetin e Bitcoin si mënyrë e re për të shkëmbyer vlerë, si dhe sfidat teknike që blockchain ka si teknologji që akoma nuk e ka vërtetuar veten.

Nga analiza e Bitcoin-it si inovacion, por dhe si një premtues për një sistem monetar (më) të decentralizuar, kuptuam se studimi i kripto-ekonomisë është i nevojshem për të qenë sa më të hapur ndaj të ardhmes.

Në një punë të ardhshme, mund të konsiderohet shtimi i të dhënave mbi investitorët, si dhe të testohet varianti GRU (Gated Recurrent Unit) i rrjetave me përsëritje.

Së fundmi, ndonëse për cmimin e Bitcoin është e vështirë, dhe deri diku spekulative të japësh mendim, fuqia e decentralizimit është e padiskutueshme dhe në mos Bitcoin, një tjetër Cryptocurrency, apo një klasë prej tyre, do të kenë impakt në ekonomi. Kështu që, këtyre teknologjive duhet t'u kushtohet më tepër vëmendje pasi njohuritë

që fitohen janë multidisiplinare dhe jo vetëm teknike.

# Bibliografia

- [1] Wikipedia. *Universal approximation theorem*. URL: [https://en.wikipedia.org/wiki/Universal\\_approximation\\_theorem](https://en.wikipedia.org/wiki/Universal_approximation_theorem).
- [2] McKinsey. *McKinsey Blockchain Adoption*. URL: [https://www.treasury.gov/initiatives/fio/Documents/McKinsey\\_FACI\\_Blockchain\\_in\\_Insurance.pdf](https://www.treasury.gov/initiatives/fio/Documents/McKinsey_FACI_Blockchain_in_Insurance.pdf).
- [3] Forbes. *Forbes, Warren Buffet on Bitcoin*. URL: <https://www.forbes.com/sites/panosmourdoukoutas/2018/05/07/warren-buffett-is-wrong-about-bitcoin/#30ad8390379c>.
- [4] *Fiat Money*. URL: [https://en.wikipedia.org/wiki/Fiat\\_money](https://en.wikipedia.org/wiki/Fiat_money).
- [5] Investopedia. *Outstanding Shares*. URL: <https://www.investopedia.com/terms/o/outstandingshares.asp>.
- [6] Andreas M. Antonopoulos. *Mastering Bitcoin*. URL: <https://ungleit-files.s3.amazonaws.com/ebf/05db7df4f31840f0a873d6ea14dcc28d.pdf>.
- [7] David Chaum. *Untraceable Electronic Cash*. URL: [http://blog.koehntopp.de/uploads/chaum\\_fiat\\_naor\\_ecash.pdf](http://blog.koehntopp.de/uploads/chaum_fiat_naor_ecash.pdf).
- [8] Adam Back. *Hash cash postage implementation*. URL: <http://www.hashcash.org/papers/announce.txt>.
- [9] Wei Dai. *B-Money*. URL: <http://www.weidai.com/bmoney.txt>.
- [10] Eric Hughes. *A Cypherpunk's Manifesto*. URL: <https://www.activism.net/cypherpunk/manifesto.html>.
- [11] *Web Dai's B-Money*. URL: <https://bitcoinmagazine.com/articles/genesis-files-if-bitcoin-had-first-draft-wei-dais-b-money-was-it/>.

- [12] Nick Szabo. *Bit Gold*. URL: <https://unenumerated.blogspot.nl/2005/12/bit-gold.html>.
- [13] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Digital Cash System*. URL: <https://bitcoin.org/bitcoin.pdf>.
- [14] Ronald L. Rivest. *The growth of cryptography*. URL: <https://courses.csail.mit.edu/6.857/2018/files/L02-Killian-lecture-slides-on-growth-of-cryptography.pdf>.
- [15] Nathaniel Popper. *Digital Gold*. URL: %7B<https://www.amazon.com/Digital-Gold-Bitcoin-Millionaires-Reinvent/dp/006236250X%7D>.
- [16] OFX. *How much does it cost to send money internationally?* URL: <https://www.ofx.com/en-au/faqs/how-much-does-it-cost-to-send-money-internationally/>.
- [17] Pew Research. *Public Trust in Government: 1958-2017*. URL: <http://www.people-press.org/2017/12/14/public-trust-in-government-1958-2017/>.
- [18] Geeks for Geeks. *Pigeonhole Principle*. URL: <https://www.geeksforgeeks.org/discrete-mathematics-the-pigeonhole-principle/>.
- [19] Medium. *Byzantine Fault Tolerance*. URL: <https://medium.com/loom-network/understanding-blockchain-fundamentals-part-1-byzantine-fault-tolerance-245f46fe8419>.
- [20] Satoshi Nakamoto. *Bitcoin P2P e-cash paper*. URL: <https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html>.
- [21] Bitcoin Wiki. *Target Space*. URL: <https://en.bitcoin.it/wiki/Target>.
- [22] Princeton university via Coursera. *Bitcoin and Cryptocurrency technologies*. URL: <https://www.coursera.org/learn/cryptocurrency>.
- [23] Reuters. *Malaysia central bank cyberattack*. URL: <https://uk.reuters.com/article/uk-philippines-cenbank-cybersecurity/philippine-banks-on-alert-after-cyber-attack-at-malaysia-central-bank-idUKKBN1H70GR>.
- [24] Investopedia. *The 2007-08 Financial Crisis In Review*. URL: <https://www.investopedia.com/articles/economics/09/financial-crisis-review.asp>.

- [25] The Guardian. *Banking inquiry has already exposed shocking corruption – but it needs more time*. URL: <https://www.theguardian.com/australia-news/commentisfree/2018/mar/22/banking-inquiry-has-already-exposed-shocking-corruption-but-it-needs-more-time>.
- [26] Wikipedia. *List of corporate collapses and scandals*. URL: [https://en.wikipedia.org/wiki/List\\_of\\_corporate\\_collapses\\_and\\_scandals](https://en.wikipedia.org/wiki/List_of_corporate_collapses_and_scandals).
- [27] Aaron Courville Ian Goodfellow Yoshua Bengio. *Deep Learning*. URL: <https://github.com/janishar/mit-deep-learning-book-pdf>.
- [28] White H. Hornik K. Stinchcombe M. *Multilayer Feedforward Networks are Universal Approximators*. URL: [http://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick\\_et\\_al.pdf](http://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick_et_al.pdf).
- [29] Sepp Hochreiter Jürgen Schmidhuber. *Long short-term memory*. URL: <https://www.bioinf.jku.at/publications/older/2604.pdf>.
- [30] Gers Schmidhuber Cummins. *Learning to forget: Continual prediction with lstm, Neural Computation*. URL: <https://pdfs.semanticscholar.org/1154/0131eae85b2e11d53df7f1360eeb6476e7f4.pdf>.
- [31] Shi Yan. *Understanding LSTM and its diagrams*. URL: <https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714>.
- [32] Ch. Olah. *Understanding LSTMs*. URL: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [33] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. URL: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781491962282/>.
- [34] Towards Data Science. *Types of optimization algorithms used in neural networks and ways to optimize gradient descent*. URL: <https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f>.

# Shtesa A

## Software dhe tools te përdorura

- **Pandas:** Librari për manipulim performant të dhënash
- **Scikit-learn:** Librari që automatizon pothuajse cdo funksion në lidhje me hapat e machine learning
- **Numpy:** Librari për veprime performante vektorësh dhe matricash
- **Keras:** Librari (shumë high level) për të ndërtuar modele deep learning, përdor TensorFlow si backend
- **Tensorflow:** Librari për deep learning, kryen llogaritjet e gradientëve dhe është shumë performantë pasi është implementuar në C++
- **Plotly:** Fork i D3.js, përdoret për të shfaqur grafiket me cmimin e bitcoin
- **Matplotlib:** Librari për të shfaqur grafikë
- **Graphviz :** Përdoret për të vizatuar rrjetat neural në këtë dokument

# Shtesa B

## Implementimi dhe shtesa të tjera

Pjesë implementi dhe shtesa të tjera B.

```
In [20]: import numpy as np
"""
Veprimet matricore ne rrjeta neurale
"""

a = np.array([[1,2,3], [1,2,3]])
b = np.array([[1,1], [2,2], [3,3]])

c = np.array([[4,5,6], [4,5,6]])
d = np.array([[4,5,6], [4,5,6]])

print( "Prodhimi element-per-element i Hadamar-it: \n" +
      str(c*d) +
      "\n")
print( "Dot product: " + str(np.dot(a,b)))

Prodhimi element-per-element i Hadamar-it:
[[16 25 36]
 [16 25 36]]

Dot product: [[14 14]
 [14 14]]
```

**Figura B.1:** Veprimet në rrjetat neural

Mbledhja te dhenave B.

Modeli LSTM, dhe perqatitja te dhenave per algoritmin B.

Trading B.

## Ngarkimi i te dhenave

---

### Funksione ndihmese

```
# import seaborn as sns
# from cryptory import Cryptory
# import datetime
# import matplotlib.pyplot as plt
# import urllib.request as urllib
# import time
# import numpy as np
# import pandas as pd
# from datetime import timedelta
# import numpy as np

from date="2013-04-28"
to_date="2018-10-30"
util_extractor = Cryptory(from_date=from_date, to_date=to_date)

def rename_columns(prepend_to_name, df):
    df.columns = [df.columns[0]]+ \
        [str(prepend_to_name)+i for i in df.columns[1:]]

"""
Funksioni merr ekstraktion te dhenat nga API-te publike perkatese

start_date: te dhenat qe kane date me te vogel se start_date do te fshihen
new_col_name: specifikon emrin e ri te kolones (meqe cdo dataset qe perdon kete funksion ka vetem 2 kolona,
            ate te dates dhe nje kolone specifique te datasetit)
"""

def drop_initial_rows(api, new_col_name, start_date="2013-04-28"):
    # Read data
    api_range_format= api
    data = pd.read_csv(urllib.urlopen(api_range_format))

    data = pd.DataFrame(data.values, columns=['date', new_col_name])

    data['date'] = pd.to_datetime(data['date'], format='%Y-%m-%d')

    data['date'] = pd.DataFrame(data=data['date'], columns=['date'])
```

```

# Disa seri kohore nga blockchain.com i llogarisin te dhenat cdo 2 dite dhe jo perdite
# sic kemi datasetin, keshtu qe datat per te cilat nuk ka vlore, u vendosim vleren e dates me pare
data = data.set_index('date').resample('D').ffill()

# Reset index
data.index.name='date'
data = data.reset_index()
data = data
data = data.drop(data[data['date'] < start_date].index)
data = data.reset_index(drop=True)

return data

```

### Ngakimi i te dhenave nga API-te perkatese

#### *Ekstraktimi i te dhenave historike*

```

# Marrja e te dhenave historike ne Coinmarketcap
exchange_data = pd.read_html("https://coinmarketcap.com/currencies/bitcoin/historical-data/?start=20130428&end=")

exchange_data.rename(columns={'Date': 'date', 'Open': 'open', 'High': 'high', 'Low': 'low', \
                             'Close**': 'close', 'Volume': 'volume', \
                             'Market Cap': 'market_cap'}, inplace=True)

# Konvertimi i dates nga string ne datetime
exchange_data = exchange_data.assign(date=pd.to_datetime(exchange_data['date']))

# Nese kolona volumit permban '-' konvertohet ne 0
exchange_data.loc[exchange_data['volume']=="-", 'volume']=0

# Konvertohet ne int
exchange_data['volume'] = exchange_data['volume'].astype('int64')
# Heq "*" qe mund te ndodhet tek te dhenat e coinmarketcap
exchange_data.columns = exchange_data.columns.str.replace("*", "")

```

Type Markdown and LaTeX:  $\alpha^2$

#### *Ekstraktimi i te dhenave te blockchain-it*

```

# Mesatarja e madhesise se bllokut nga blockchain.info; max size eshte 1 MB
block_size_api = "https://blockchain.info/charts/avg-block-size?timespan=all&format=csv"
avg_block_size_data = drop_initial_rows(block_size_api, 'avg_block_size')

# Numri i transaksioneve
txs_api = "https://api.blockchain.info/charts/n-transactions?timespan=all&format=csv"
txs_data = drop_initial_rows(txs_api, 'transactions')

# Fitimi i Miners ne $
bchain_mirev_api = "https://www.quandl.com/api/v3/datasets/BCHAIN/MIREV.csv?api_key=55AcwGQK3qwgy8J3K4Pw"
bchain_mirev_data = drop_initial_rows(bchain_mirev_api, "mining_revenue")

# Numri i perdoruesve te wallet
bch_accounts_api = "https://api.blockchain.info/charts/my-wallet-n-users?timespan=all&format=csv"
bch_accounts_data = drop_initial_rows(bch_accounts_api, 'accounts')

# Bashkimi i te dhenave per gjate dimensionit te kohes
blockchain_data = avg_block_size_data.merge(txs_data, on='date', how='inner'). \
    merge(bchain_mirev_data, on='date', how='inner'). \
    merge(bch_accounts_data, on='date', how='inner')

```

#### *Ekstraktimi i te dhenave makroekonomike*

```

# S&P 500 price index nga Yahoo Finance
s_and_p_stock = util_extractor.get_stock_prices(market="%5EGSPC")
s_and_p_stock = s_and_p_stock.loc[:, ['date', 'close']]
s_and_p_stock = s_and_p_stock.rename(columns={'close': 'sp_close'})

# Dow and Jones nga Yahoo Finance
dow_jones_stock = util_extractor.get_stock_prices(market="%5EDJI")
dow_jones_stock = dow_jones_stock.loc[:, ['date', 'close']]
dow_jones_stock = dow_jones_stock.rename(columns={'close': 'dj_close'})

macro_econ_data = s_and_p_stock.merge(dow_jones_stock, on='date', how='inner')

```

### Ekstraktimi i te dhenave te Sentimentit

```
| # Te dhenat nga Google Trends  
| btc_google_trends = util_extractor.get_google_trends(kw_list=['bitcoin'])  
| sentiment_data = btc_google_trends
```

### Riemertim i kolonave

```
| exchange_data.columns =[exchange_data.columns[0]]+['btc_'+i for i in exchange_data.columns[1:]]  
| blockchain_data.columns =[blockchain_data.columns[0]]+['bch_'+i for i in blockchain_data.columns[1:]]  
| sentiment_data.columns = [sentiment_data.columns[0]] + ['google_trends_'+i for i in sentiment_data.columns[1:]]
```

### Vizualizimi i te dhenave te cmimit

```
| import plotly.plotly as py  
| import plotly.graph_objs as go  
  
| import pandas as pd  
  
| btc_trace_high = go.Scatter(  
|     x=exchange_data.date,  
|     y=exchange_data['btc_high'],  
|     name = "Cmimi i siperm (High)",  
|     line = dict(color = '#17BECF'),  
|     opacity = 0.8)  
  
| btc_trace_low = go.Scatter(  
|     x=exchange_data.date,  
|     y=exchange_data['btc_low'],  
|     name = "Cmimi i poshtem (Low)",  
|     line = dict(color = '#7F7F7F'),  
|     opacity = 0.8)  
  
| data = [btc_trace_high,btc_trace_low]  
| # d.strftime('%Y-%m-%d')  
| start = '2016-01-01'  
| end = '2018-10-01'  
  
| layout = dict(  
|     title = "Seria kohore e cmimit te Bitcoin nga viti 2016",
```

```

        xaxis = dict(
            range = [start,end]
        )
    )

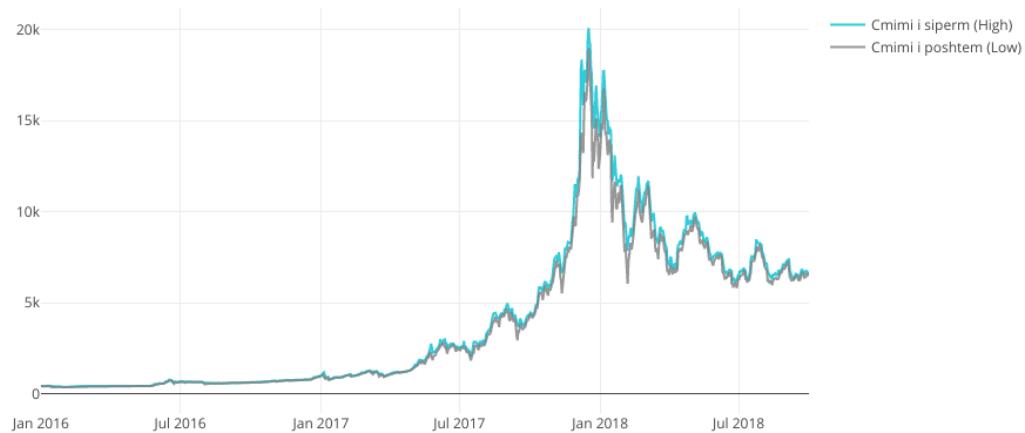
fig = dict(data=data, layout=layout)
py.iplot(fig, filename = "Seria kohore e cmimit te Bitcoin nga viti 2016")

```

High five! You successfully sent some data to your account on plotly. View your plot in your browser at <http://plot.ly/~kstruga/0> or inside your plot.ly account where it is named 'Seria kohore e cmimit te Bitcoin nga viti 2016'

:

Seria kohore e cmimit te Bitcoin nga viti 2016



[EDIT CHART](#)

Ruajme dataset-et e vecante

```
: ⚡ exchange_data.to_csv('exchange_data.csv')
blockchain_data.to_csv('blockchain_data.csv')
macro_econ_data.to_csv('macro_econ_data.csv')
sentiment_data.to_csv('sentiment_data.csv')

: ⚡ model_data = exchange_data.merge(blockchain_data, on='date', how='inner'). \
    merge(macro_econ_data, on='date', how='inner'). \
    merge(sentiment_data, on='date', how='inner')
```

Heqim kolonat qe nuk na interesojne: cmimi i hyrjes, cmimi i hapjes

```
: ⚡ model_data=model_data.drop(columns=['btc_open', 'btc_low'])
```

Fshij rekordet qe u perkasin te dhenave para 2014

```
: ⚡ model_data=model_data[model_data['date']>='2016-01-01']

: ⚡ model_data.head()
```

```
[75]:
```

	date	btc_high	btc_close	btc_volume	btc_market_cap	bch_avg_block_size	bch_transactions	bch_mining_revenue	bch_accounts	sp_close
0	2018-10-04	6603.31	6576.69	3838410000	112435991226	0.838226	242303	1.18658e+07	2.89648e+07	2901.610107 26
1	2018-10-03	6571.46	6502.59	3887310000	113392236466	0.838226	242303	1.14073e+07	2.89464e+07	2925.510010 26
2	2018-10-02	6611.84	6556.10	3979260000	114062551875	0.857106	242820	1.3161e+07	2.89234e+07	2923.429932 26
3	2018-10-01	6653.30	6589.62	40009720000	114509724600	0.857106	242820	1.35352e+07	2.88884e+07	2924.590088 26
4	2018-09-30	6643.78	6625.56	4002280000	114234369232	0.693384	218339	1.25711e+07	2.88698e+07	2913.979980 26

#### Vizualizojme korrelacionin e Pearson-it mes atributave

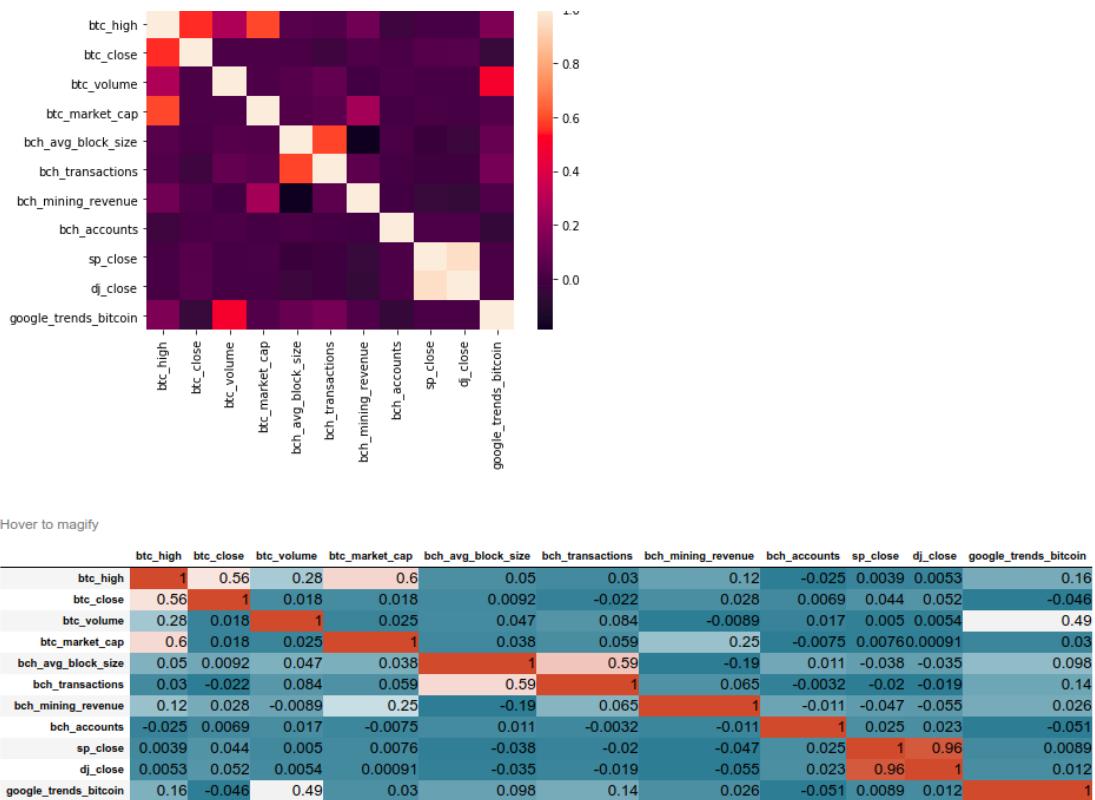
```
# Pearson correlation on all attributes
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

""" Nuk marrë parasysh kolonen e dates """
all_features_df = model_data.loc[:, model_data.columns != 'date']

# Ndryshimi ne perqindje =>
# fillimisht llogaris: rritjen = y_t - y_(t-1)
# me pas: %rritje = (Increase-y_(t-1))*100
corr = all_features_df.pct_change().corr(method='pearson')
fig, ax = plt.subplots(figsize=(7,5))
sns.heatmap(corr,
            xticklabels=[col.replace("_price", "") for col in corr.columns.values],
            yticklabels=[col.replace("_price", "") for col in corr.columns.values],
            annot_kws={"size": 16})
plt.show()

# Paraqitja ne forme tabelje
cmap=sns.diverging_palette(220, 20, sep=20, as_cmap=True)

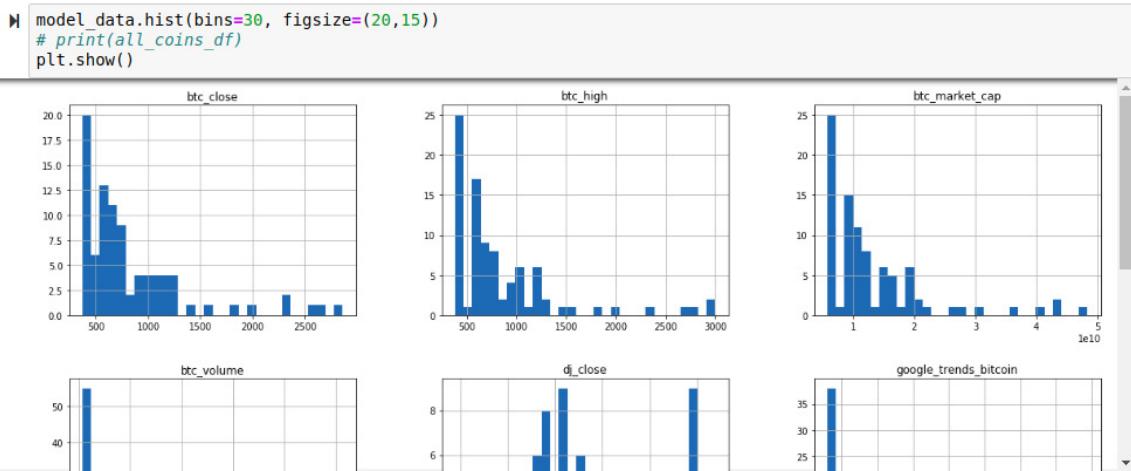
def magnify():
    return [dict(selector="th",
                 props=[("font-size", "7pt")]),
            dict(selector="td",
                 props=[('padding', "0em 0em")]),
            dict(selector="th:hover",
                 props=[("font-size", "12pt")]),
            dict(selector="tr:hover td:hover",
                 props=[('max-width', '200px'),
                        ('font-size', '12pt')])]
```



]:

Hover to magify

	btc_high	btc_close	btc_volume	btc_market_cap	bch_avg_block_size	bch_transactions	bch_mining_revenue	bch_accounts	sp_close	dj_close	google_trends_bitcoin
btc_high	1	0.56	0.28	0.6	0.05	0.03	0.12	-0.025	0.0039	0.0053	0.16
btc_close	0.56	1	0.018	0.018	0.0092	-0.022	0.028	0.0069	0.044	0.052	-0.046
btc_volume	0.28	0.018	1	0.025	0.047	0.084	-0.0089	0.017	0.005	0.0054	0.49
btc_market_cap	0.6	0.018	0.025	1	0.038	0.059	0.25	-0.0075	0.0076	0.0091	0.03
bch_avg_block_size	0.05	0.0092	0.047	0.038	1	0.59	-0.19	0.065	-0.0032	-0.02	-0.035
bch_transactions	0.03	-0.022	0.084	0.059	0.59	1	0.065	-0.0032	-0.02	-0.019	0.14
bch_mining_revenue	0.12	0.028	-0.0089	0.25	-0.19	0.065	1	-0.011	-0.047	-0.055	0.026
bch_accounts	-0.025	0.0669	0.017	-0.0075	0.011	-0.0032	-0.011	1	0.025	0.023	-0.051
sp_close	0.0039	0.044	0.005	0.0076	-0.038	-0.02	-0.047	0.025	1	0.96	0.0089
dj_close	0.0053	0.052	0.0054	0.00091	-0.035	-0.019	-0.055	0.023	0.96	1	0.012
google_trends_bitcoin	0.16	-0.046	0.49	0.03	0.098	0.14	0.026	-0.051	0.0089	0.012	1



Type *Markdown* and *LaTeX*:  $\alpha^2$

Normalizimi te dhenave duke perdorur metoden Min-Max scaling

```
# Perdor scikit-learn per normalizim
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.preprocessing import Imputer, StandardScaler

model_data.fillna(model_data.mean(), inplace=True)

mean_imputer = Imputer(missing_values='NaN', strategy='mean', axis=0)

model_data_without_date = model_data.loc[:, model_data.columns != 'date']

# Fit -> do te thote qe scikit gjen vlerat min dhe max
mean_imputer = mean_imputer.fit(model_data_without_date)

imputed_df = mean_imputer.transform(model_data_without_date)

imputed_df = pd.DataFrame(imputed_df, columns = model_data_without_date.columns)

imputed_df[['btc_high', 'btc_close', 'btc_volume', 'btc_market_cap', \
            'bch_avg_block_size', 'bch_transactions', 'bch_mining_revenue', \
            'bch_accounts', 'sp_close', 'dj_close', 'google_trends_bitcoin']] = \
            MinMaxScaler(imputed_df[['btc_high', 'btc_close', 'btc_volume', \
            'btc_market_cap', 'bch_avg_block_size', \
            'bch_transactions', 'bch_mining_revenue', 'bch_accounts', 'sp_close', 'dj_close']])

imputed_df['date'] = model_data['date']

imputed_df["date"] = imputed_df["date"].values[::-1]

model_data=imputed_df
```

Rirrendita e kolonave duke vendosur 'daten' si kolonen e pare

```
model_data=model_data[['date','btc_high', 'btc_close', 'btc_volume', 'btc_market_cap', \
                      'bch_avg_block_size', 'bch_transactions', 'bch_mining_revenue', \
                      'bch_accounts', 'sp_close', 'dj_close', 'google_trends_bitcoin']]
```

Ruajtja e model\_data per tu perdorur gjate trajnimit/testimit

```
# I ruaj ne csv
model_data.to_csv('model_data.csv')
```

Leximi i te dhenave

```
model_data.head()
len(model_data)
```

79]: 1008

Rirrendita e kolonave duke vendosur 'daten' si kolonen e pare

```
model_data=model_data[['date','btc_high', 'btc_close', 'btc_volume', 'btc_market_cap', \
                      'bch_avg_block_size', 'bch_transactions', 'bch_mining_revenue', \
                      'bch_accounts', 'sp_close', 'dj_close', 'google_trends_bitcoin']]
```

Ruajtja e model\_data per tu perdorur gjate trajnimit/testimit

```
# I ruaj ne csv
model_data.to_csv('model_data.csv')
```

Leximi i te dhenave

```
model_data.head()
len(model_data)
```

79]: 1008

```

In [1]: model_data = model_data[model_data['date'] >= "2016-01-01"]

In [2]: len(model_data)

Out[2]: 1008

In [3]: split_date='2018-04-25'
# Hes kolonen e dates meje nuk na duhet me
training_set, test_set = model_data[model_data['date'] < split_date], model_data[model_data['date'] >= split_date]
training_set = training_set.drop('date', 1)
test_set = test_set.drop('date', 1)

In [4]: print(len(training_set))
print(len(test_set))
print(len(model_data))

845
163
1008

```

```

In [1]: import plotly.plotly as py
import plotly.graph_objs as go

import pandas as pd

training = go.Scatter(
    x=model_data[model_data['date'] < split_date]['date'].astype(datetime.datetime),
    y=model_data[model_data['date'] < split_date]['btc_close'],
    name = "Trajnim",
    line = dict(color = '#6D13C1'),
    opacity = 0.8)

test = go.Scatter(
    x=model_data[model_data['date'] >= split_date]['date'].astype(datetime.datetime),
    y=model_data[model_data['date'] >= split_date]['btc_close'],
    name = "Testim",
    line = dict(color = '#17BECF'),
    opacity = 0.8)

data = [training, test]

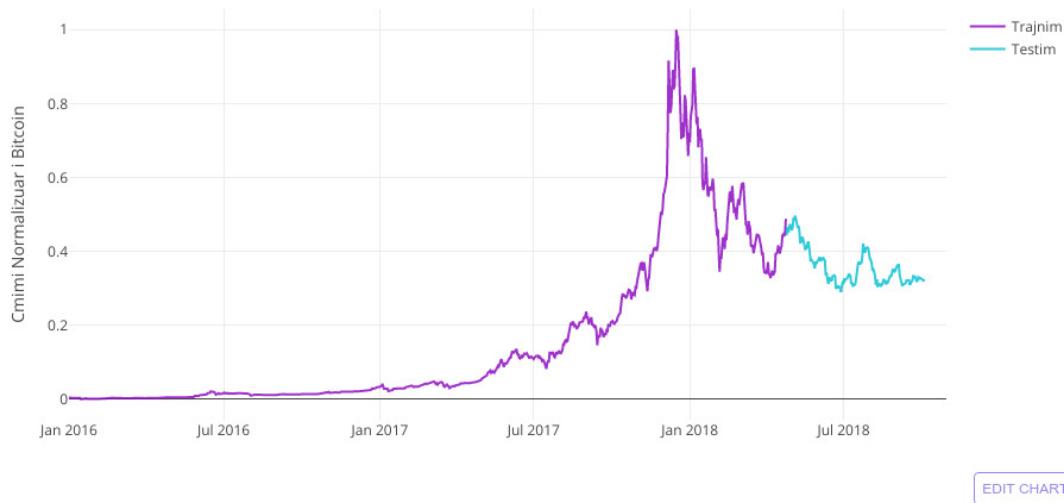
layout = dict(
    title = "Ndarja e te dhenave ne bashkesi per trajnim (80%) dhe testim (20%)",
    xaxis = dict(
        range = ['2016-01-01', '2018-10-30']),
    yaxis=dict(
        title='Cmimi Normalizuar i Bitcoin')
)

fig = dict(data=data, layout=layout)
py.iplot(fig, filename = "Manually Set Range")

```

Out[1]:

Ndarja e te dhenave ne bashkesi per trajnim (80%) dhe testim (20%)



```
window_len=32
pred_range=30

training_inputs = []
for i in range(len(training_set)-window_len):
    temp_set = training_set[i:(i+window_len)].copy()
    training_inputs.append(temp_set)
```

```
print(len(training_inputs))
```

813

```

M test_inputs = []
for i in range(len(test_set)-window_len):
    temp_set = test_set[i:(i+window_len)].copy()
    test_inputs.append(temp_set)

# print(test_inputs[0])
test_outputs = test_set['btc_close'][window_len:].values
print(len(test_outputs)) # predicting 45 points in the future

```

131

Type Markdown and LaTeX:  $\alpha^2$

```

M training_inputs = [np.array(training_inputs) for training_inputs in training_inputs]
training_inputs = np.array(training_inputs)

test_inputs = [np.array(test_inputs) for test_inputs in test_inputs]
test_inputs = np.array(test_inputs)

```

```

M training_outputs = []
for i in range(window_len, len(training_set['btc_close'])-pred_range):
    training_outputs.append(training_set['btc_close'][i:i+pred_range].values)

training_outputs = np.array(training_outputs)

# testing outputs, which is needed to evaluate/predict the model
testing_outputs = []
for i in range(window_len, len(test_set['btc_close'])-pred_range):
    testing_outputs.append(test_set['btc_close'][i:i+pred_range].values)

testing_outputs = np.array(testing_outputs)

```

```

M print(training_inputs.shape)
print(training_outputs.shape)

print(test_inputs.shape)
print(testing_outputs.shape)

```

```

# importujme modulet e Keras
from keras.models import Sequential
from keras.layers import Activation, Dense
from keras.layers import LSTM, GRU
from keras.layers import Dropout

def lstm_model(inputs, output_size, neurons, activ_func="relu",
               dropout=0.25, loss="mae", optimizer="adam"):
    model = Sequential()
    model.add(LSTM(neurons, input_shape=(inputs.shape[1], inputs.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(units=output_size))
    model.add(Activation("linear"))

    model.compile(loss=loss, optimizer=optimizer)
    return model

# random seed for reproducibility
np.random.seed(202)

# initialise model architecture
bt_model = lstm_model(training_inputs, output_size=pred_range, neurons = 100)
# bt_model = denser_model(LSTM_training_inputs, output_size=pred_range, neurons = 100)

# print(bt_model.get_weights())
# train model on data
bt_history = bt_model.fit(training_inputs[:-pred_range], training_outputs,
                           epochs=100, batch_size=32, verbose=1, shuffle=True)

loss_fn=str(np.mean(bt_history.history['loss']))

Epoch 1/100
783/783 [=====] - 1s 2ms/step - loss: 0.1005
Epoch 2/100
783/783 [=====] - 0s 610us/step - loss: 0.0562
Epoch 3/100
783/783 [=====] - 0s 618us/step - loss: 0.0478
Epoch 4/100
783/783 [=====] - 1s 1ms/step - loss: 0.0423
Epoch 5/100
783/783 [=====] - 0s 618us/step - loss: 0.0409
Epoch 6/100
783/783 [=====] - 1s 860us/step - loss: 0.0383
Epoch 7/100
783/783 [=====] - 1s 805us/step - loss: 0.0354

```

```

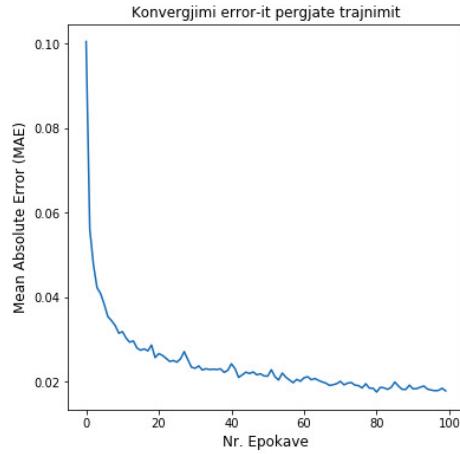
from pylab import rcParams
rcParams['figure.figsize'] = 6, 6

# fig.clear()
fig, ax1 = plt.subplots(1,1)

ax1.plot(bt_history.epoch, bt_history.history['loss'])
ax1.set_title('Konvergjimi error-it per gjate trajnimit')

if bt_model.loss == 'mae':
    ax1.set_ylabel('Mean Absolute Error (MAE)', fontsize=12)
    # just in case you decided to change the model loss calculation
else:
    ax1.set_ylabel('Model Loss', fontsize=12)
ax1.set_xlabel('Nr. Epokave', fontsize=12)
plt.show()
bt_model.loss

```



4]: 'mae'

```

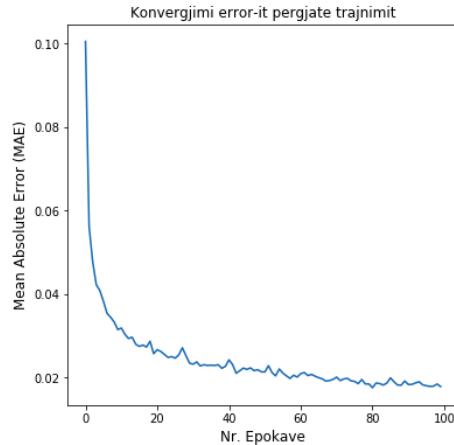
from pylab import rcParams
rcParams['figure.figsize'] = 6, 6

# fig.clear()
fig, ax1 = plt.subplots(1,1)

ax1.plot(bt_history.epoch, bt_history.history['loss'])
ax1.set_title('Konvergjimi error-it per gjate trajnimit')

if bt_model.loss == 'mae':
    ax1.set_ylabel('Mean Absolute Error (MAE)', fontsize=12)
# just in case you decided to change the model loss calculation
else:
    ax1.set_ylabel('Model Loss', fontsize=12)
ax1.set_xlabel('Nr. Epokave', fontsize=12)
plt.show()
bt_model.loss

```



```
4]: 'mae'
```

```

import plotly.plotly as py
import sklearn.metrics
import plotly.graph_objs as go

mae_test_error = sklearn.metrics.mean_absolute_error(training_outputs, bt_model.predict(training_inputs[:-pred_len]))

# Create a trace
real_price = go.Scatter(
    x = model_data[model_data['date'] < split_date]['date'][window_len:].astype(datetime.datetime),
    y = training_set['btc_close'][window_len:],
    name = "Cimi i Verte",
    marker = dict(
        size = 10,
        color = 'rgba(152, 0, 0, .8)'
    )
)

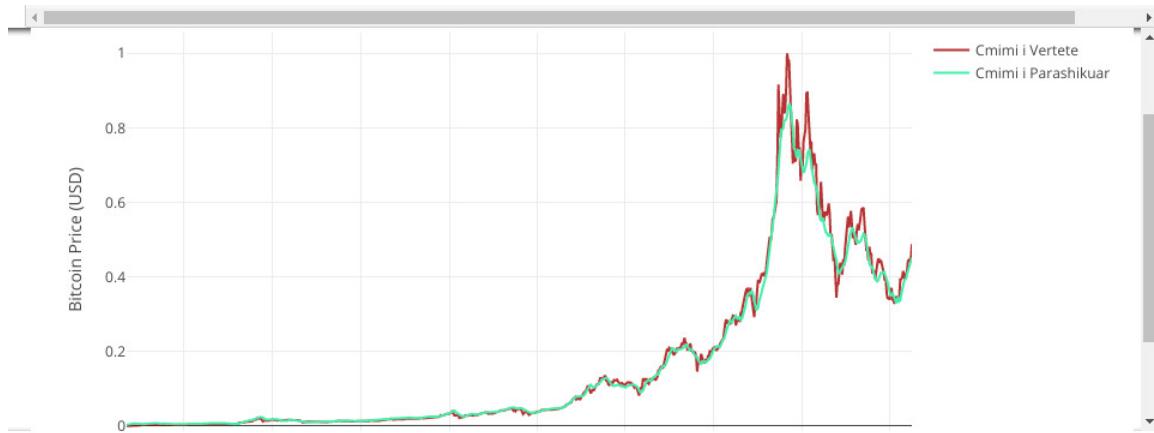
predicted_price = go.Scatter(
    x = model_data[model_data['date'] < split_date]['date'][window_len:].astype(datetime.datetime),
    y = ((np.transpose(bt_model.predict(training_inputs)))[0]),
    name = "Cimi i Parashikuar",
    marker = dict(
        size = 10,
        color = "#82E0AA"
    )
)

layout = dict(title = 'Parashikimi ne Bashkesine e Trajnimit, MAE: %.4f' % mae_test_error,
              yaxis = dict(title = 'Bitcoin Price (USD)'),
              )

data = [real_price,predicted_price]
fig = dict(data=data, layout=layout)

py.iplot(fig, filename='bitcoin-prediction')

```



```

import plotly.plotly as py
import sklearn.metrics
import plotly.graph_objs as go

# llogaritja errorit me scikit
# mae_test_error = sklearn.metrics.mean_absolute_error(testing_outputs, bt_model.predict(test_inputs[:-pred_range]))
# ne menyre alternative mund ta llogarsim edhe vete:
mae_test_error=np.mean(np.abs((np.transpose(bt_model.predict(test_inputs)))-(test_set['btc_close'].values[window_len:])))

# Create a trace
real_price = go.Scatter(
    x = model_data[model_data['date']>=split_date]['date'][window_len:].astype(datetime.datetime),
    y = test_set['btc_close'][window_len:],
    name = "Cmimi i Vertete",
    marker = dict(
        size = 10,
        color = 'rgba(152, 0, 0, .8)'
    )
)

predicted_price = go.Scatter(
    x = model_data[model_data['date']>= split_date]['date'][window_len:].astype(datetime.datetime),
    y = ((np.transpose(bt_model.predict(test_inputs))))[0],
    name = "Cmimi Parashikuar",
    marker = dict(
        size = 5,
        color = "#82E0AA"
    ),
    mode = 'lines+markers',
)

layout = dict(title = 'Parashikimi me te dhenat test, MAE: %.4f'% mae_test_error,
              yaxis = dict(title = 'Cmimi Bitcoin '))

data = [real_price, predicted_price]
fig = dict(data=data, layout=layout)
py.iplot(fig, filename='bitcoin-prediction')

```

Parashikimi me te dhenat test, MAE: 0.0967



EDIT CHART

```
import plotly.plotly as py
import sklearn.metrics
import plotly.graph_objs as go

# mae_test_error = sklearn.metrics.mean_absolute_error(testing_outputs, bt_model.predict(test_inputs[:-pred_range]))

# Create a trace
real_price = go.Scatter(
    x = model_data[model_data['date']>=split_date]['date'][window_len:].astype(datetime.datetime),
    y = test_set['btc_close'][window_len:],
    name = "Cmimi i Vertete",
    marker = dict(
        size = 10,
        color = 'rgba(152, 0, 0, .8)'
    )
)

predicted_price = go.Scatter(
    x = model_data[model_data['date']>= split_date]['date'][window_len:][:pred_range].astype(datetime.datetime),
    y = ((np.transpose(bt_model.predict(test_inputs)))[0],
    name = "Cmimi Parashikuar, Muaji Qershorr",
    marker = dict(
        size = 10,
        color = "#82E0AA"
    )
)

predicted_price1 = go.Scatter(
    x = model_data[model_data['date']>= split_date]['date'][window_len+pred_range:][:pred_range].astype(datetime.datetime),
    y = ((np.transpose(bt_model.predict(test_inputs[window_len+1:])))[0],
    name = "Cmimi Parashikuar, Muaji Korrik",
    marker = dict(
        size = 10,
        color = "#B60BE5"
    )
)

predicted_price2 = go.Scatter(
    x = model_data[model_data['date']>= split_date]['date'][((window_len+pred_range)*pred_range):][:pred_range].astype(datetime.datetime),
    y = ((np.transpose(bt_model.predict(test_inputs[window_len*2:])))[0],
    name = "Cmimi Parashikuar, Muaji Gusht",
    marker = dict(
        size = 10,
        color = "#1739d1"
    )
)
```

```

:   """
    EVALUATING ON TEST SET; MAE: 0.0301
"""

print(bt_model.evaluate(test_inputs[:pred_range], testing_outputs, batch_size=window_len))
print(bt_model.evaluate(training_inputs[:pred_range], training_outputs, batch_size=200))

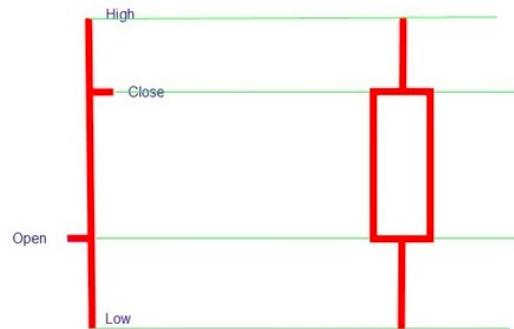
102/102 [=====] - 1s 8ms/step
0.10037814672378932
784/784 [=====] - 0s 114us/step
0.012127881615460679

:   # bt_model
bt_model.summary()
bt_model.save('bt_model_dense.h5')
bt_model.get_weights()
# bt_model.save_weights('bt_model_weights')

Layer (type)          Output Shape         Param #
=====
lstm_9 (LSTM)        (None, 100)          44800
dropout_9 (Dropout)  (None, 100)          0
dense_9 (Dense)      (None, 30)           3030
activation_9 (Activation) (None, 30)          0
=====
Total params: 47,830
Trainable params: 47,830
Non-trainable params: 0

.62]: [array([[-0.09414479, -0.03781071,  0.09314576, ...,  0.02147012,
   -0.05938193, -0.05666057],
   [ 0.00467169,  0.03169158, -0.06160763, ..., -0.08063728,
   0.05194542, -0.12420416]

```



**Figura B.2:** Ilustrim i një *candlestick*, ku paraqiten cmimet e ndryshme të një stoku, në terminologjinë e analizës teknike